SESSION

REAL-WORLD DATA MINING APPLICATIONS, CHALLENGES, AND PERSPECTIVES

Chair(s)

Mahmoud Abou-Nasr
Robert Stahlbock

Three Different Paradigms for Interactive Data Clustering

Terje Kristensen and Vemund Jakobsen

***Abstract*—In this paper we present three different algorithms for data clustering. These are the Self-Organizing Map (SOM), Neural Gas (NG) and Fuzzy Means (FCM) algorithms. SOM and NG algorithm are based on competitive learning, and an important property of these algorithms is that they preserve the topological structure of data. This means that data that is close in input distribution is mapped to nearby locations in the network. The FCM algorithm is an algorithm based on soft clustering which means that the different clusters are not necessarily distinct, but may overlap. This clustering method may be very useful in many biological problems, for instance in genetics, where a gene may belong to different clusters. The different algorithms are compared in terms of their visualization of the clustering of proteomic data.**

I. INTRODUCTION

An important step in the data analysis process is to organize the data into meaningful structures to uncover their natural grouping(s). Once the data has been organized, it can be used in subsequent steps of analysis such as hypothesis forming or decision making. This method of exploring and organizing data can be done automatically by using an algorithmic approach known as *data clustering* (or cluster analysis). Data clustering has been studied extensively over the years and has several areas of application, including market research, pattern recognition, image analysis and machine learning [4]. In these areas, one often has to do analysis without a priori information about the data. Data clustering can in such cases be useful in organizing and uncovering relationships hidden in the data before doing further analysis.

Analyzing large amounts of data can be difficult and a time-consuming task if done manually. It is therefore necessary to develop tools to support analysis and visualization of large multi-dimensional data sets. These tools may provide structured view of the data, and potentially reveal previously unknown information. Although there are several algorithms that can be used for this purpose, they are not always equally suited for the problem at hand. Using different algorithms on the same problem will often produces different results and the best way to analyze data is therefore to compare the different algorithmic approaches and their solutions.

The three algorithms and their visualization are:

- Self-organizing Map (SOM), which is proven to be a powerful tool for visualizing high-dimensional data
- Neural Gas (NG) [2,3], that can be a less constrained version of SOM
- Fuzzy C-means (FCM), which opposed to the other algorithms, allow objects to belong to several clusters

There are many systems that support analysis and visualization of clusters based on SOM, NG and FCM, but not in one combined system. In the paper we describe a system that explores data by using these clustering methods and their visualizations. The data to be clustered is high-dimensional with more than three dimensions. To be able to interpret the data, the dimension has to be reduced to either two or three dimensions. Finally, the algorithms have been applied to interpret the clustering of protein data found in blood serum.

II. MAIN TYPES OF CLUSTERING

A. Hard clustering

In hard clustering (or crisp clustering) the objects belong to one and only one cluster, and each cluster contains at least one object. In mathematical terms it can be defined as follows [7]:.

Let D be the set of data consisting of n vectors, that is:

$$D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \quad (1)$$

We now define the k-clustering of D as the partition of D into k sets (clusters) $C_1, \ldots, C_k$ such that the following conditions are satisfied

- $C_i \neq \emptyset, \; i = 1, \ldots, k$ (2)
- $\cup C_i = D$ (3)
- $C_i \cap C_j = \emptyset, \; i \neq j, \; i, j = 1, \ldots, k$ (4)

B. Soft clustering

In soft clustering or Fuzzy clustering an approach based on fuzzy logic is used. The objects may then belong to several objects with a certain degree of membership. This means for instance that an object in the center of a cluster may have a higher probability of belonging to that cluster than an object on the edge of the cluster. It can be defined as follows: let D be the set of data consisting of n vectors. A fuzzy clustering of D into k clusters is characterized by k functions u_j where

Terje Kristensen is with the Department of Computer Engineering, Bergen University College, Nygaardsgaten 112, N-5020 Bergen, Norway (e-mail: tkr@hib.no).

Vemund Jakobsen is with the Company Pattern Solutions AS in Bergen, Norway, Nygaardsgaten 112, N-5020 Bergen, Norway (e-mail: vemund.jakobsen@gmail.com)

$u_j: D \rightarrow [0, 1],\ j = 1, 2, \ldots, k,$ and (5)

$$\sum_{j=1}^{k} u_j(x_i) = 1,\ i = 1, 2,, \ldots.,n, \quad 0 < \sum_{j=1}^{k} u_j(x_i) < n \qquad (6)$$

The functions u_j are the membership functions. Vectors with membership values close to unity have a high degree of membership in the corresponding cluster, and vectors with values close to zero have a low membership in the corresponding cluster.

III. The Self-Organizing Map algorithm

The Self-Organizing Map (SOM) network consists of two fully connected layers: the input layer (representing input vectors) and the output layer. The output layer consists of nodes that are arranged in a map. Moreover, each node has a specific position in the map and also an associated weight vector of the same dimension as the input vector.

An important feature of the SOM algorithm is that it preserves the topological structure of the input space. This means that data items that are close in the input space are mapped to nodes that are close in the map [8]. The SOM algorithm accomplishes this by calculating a neighbourhood function within the map. All nodes within this neighbourhood are adjusted towards the input vector.

The idea of SOM is to adjust the nodes until they represent the input distribution. The nodes represent clusters that reflect how the data is distributed. The SOM algorithm begins by setting the weights to random values. It then proceeds to the three processes given below [7].

A. Competitive process

Let $\mathbf{x} = (x_1, x_2, \ldots, x_d)^T$ be an object selected at random from the data set, where d is the dimension of the data set. Let $\mathbf{w_i} = (wi_1, wi_2, \ldots, wi_d)^T$ be the weight vector of node i. The node that is most similar, or closest according to some distance measure, to the input vector is then determined. This node is referred to as the *Best Matching Unit* (BMU) [11].

B. Cooperative process

In this process a topological neighbourhood is defined where the BMU locates the center of the neighbourhoood. The neighbourhood is usually an exponential function, typically a Gaussian function defined as

$$h(t) = \exp(-d^2_{c,j}/2\sigma^2(t)) \qquad (7)$$

where t is the current iteration, $d_{c,j}$ is the lateral distance at node j and σ is the radius or with of the neigbourhood function specified by

$$\sigma(t) = \sigma_0 \exp(-t/\lambda_1) \qquad (8)$$

where λ_1 is a time constant that allows the exponential function to decay with increasing time, and σ_0 is the radius at time t_0. The lateral distance d in the two-dimensional case is defined as:

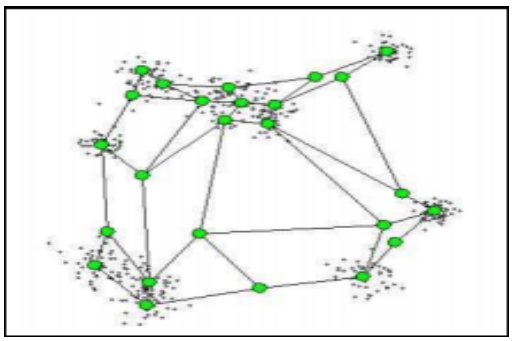


Fig.1. SOM network approximating the input distribution

$$d_c = \|\mathbf{x} - \mathbf{w_c}(t)\| = \min\{ \|\mathbf{x} - \mathbf{w_i}(t)\| \},\ 1 \le i \le k \qquad (9)$$

C. Adaptive process

In the adaptive process the weights of all nodes within the neigbourhood, as well as BMU, are adjusted according to the rule:

$$\mathbf{w_i}(t+1) = \mathbf{w_i}(t) + \alpha(t)h(t)[\mathbf{x}(t) - \mathbf{w_i}(t)] \qquad (10)$$

where $\mathbf{w_i}(t)$ and $\mathbf{x}(t)$ are the weight and input vector at time t. The learning rate α(t) is defined as

$$\alpha(t) = \alpha_0 \exp(-t/\lambda_2) \qquad (11)$$

where λ_2 is a time constant that brings the learning rate close to zero with increasing time t, and α_0 is the learning rate at time t_0.

D. Choice of Parameters

SOM leads to an organized representation of input objects provided that the parameters of the algorithm are selected properly. The adaption process may be decomposed into two phases; the ordering phase and the convergence phase. In the ordering phase the topological ordering of the weight vectors takes place. The topological ordering produces a rough ordering of weight vectors when the neighbourhood is big and the learning rate high. In the convergence phase the map is fined-tuned and provides an accurate statistical quantification of the input space. During this phase the neighbourhood is small and the learning rate low. Some guidelines for selecting the parameters are given below:

- The learning rate is often set initially close to 0.1 and decreases gradually to above 0.01
- The neigbourhood h(t) starts with a large radius covering most of the nodes in the grid. The initial radius σ_0 can be set equal to the "radius" of the lattice
- The number of iterations are difficult to set and depends on both the size and dimensionality of the data set

In general the topological ordering of the weights may require as many as 10 000 iterations or more [15]. After the weights have been topological ordered , the map needs to be fine-tuned. The number of iterations constituting the convergence phase should be at least 500 times the number of nodes in the lattice [6]. The algorithm in pseudo code is given by:

Algorithm 1 : Pseudocode of the SOM algorithm.

Input: A set of input vectors $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$

Output: A set of weight vectors $W = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k\}$

```
Set parameters α0, σ0, τ1, τ2, k, and t_max
Initialize all w ∈ W to random values
for t = 1 to t_max do
    Select random x ∈ D
    Find w such that d(x, w) = min{d(x, w) | x ∈ D}
    for all w in neighbourhood h do
        Update the weights: w = w + αh(x − w)
        Reduce learning rate α
    end for
end for
```

Fig.2. The SOM algorithm in pseudo code

IV. THE NEURAL GAS ALGORITHM

The NG algorithm is, similar to SOM, an algorithm that uses unsupervised competitive learning. It starts by initializing a set W containing N nodes, each with weights set to random values. It then proceeds, in a similar way as the SOM, to the three processes given below.

Like SOM, NG aims to preserve the topological structure of the input space. It also uses a neighbourhood function. However, instead of computing the neigbourhood within a lattice, it computes the neighborhood within the input space. The nodes are therefore ranked by their distance from an input object. After ranking, the node weights are adjusted according to an adapted rule, with the closest node being adjusted most. As a consequence, similar input objects are more likely to be mapped to nodes that are close (i.e. similar) in the NG network (output space). In addition, the NG adaption rule minimizes a global cost function. Such a cost function does not exist for the SOM adaption rule [12].

Since NG lacks a fixed output space, it can achieve better results that SOM. However, this also limits the applications of NG to data projection and visualization, and therefore only a few visualization schemes have been developed.

A. Competitive process

Let $\mathbf{x} = (x_1, x_2, \ldots, x_d)^T$ be an object selected at random from the data set, where d is the dimension of the object. Sort all nodes of the weights space W according to their distance to x, with the nearest node coming first and the farthest as the last. That is, for sorted $W = \{C_m, C_o, C_k, \ldots\}$ with corresponding weights $\mathbf{w}_m$, $\mathbf{w}_o$, $\mathbf{w}_k$, . . ., the following relation is given

$$\|\mathbf{x} - \mathbf{w}_m\| \leq \|\mathbf{x} - \mathbf{w}_0\| \leq \|\mathbf{x} - \mathbf{w}_k\| \ldots.$$

where the norm is usually the Euclidean norm.

B. Cooperativ processs

In this process the neigbourhood function could be an exponential function given by:

$$h(k_i) = \exp(-k_i / \lambda(t)),\ t = 1, 2, 3 \ldots, \qquad (12)$$

where k_i is the rank index in W and t is the current iteration. $k_i = 0$ means the closest node **w**i to the input vector **x** and k_i = N-1 the least closed node to the input vector **x**.

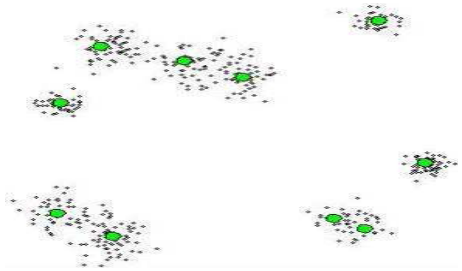


Fig. 3. NG network approximating the input distribution. The network consists of green points and the input data of black points

In the equation above λ(t) is defined by

$$\lambda(t) = \lambda_0(\lambda_f / \lambda_0)^{(t / t_{max})} \qquad (13)$$

where λ_0 and λ_f are the initial and final with of the neigbourhood, respectively, and t_{max} the maximum number of iterations.

C. Adaptive processs

In the adaptive process the weights of all the nodes in W are adjusted according to the rule:

$$\mathbf{w}_i(t + 1) = \mathbf{w}_i(t) + \varepsilon(t)h(t)[\mathbf{x}(t) - \mathbf{w}_i(t)] \qquad (14)$$

where ε(t) is given by.

$$\varepsilon(t) = \varepsilon_0(\varepsilon_f / \varepsilon_0)^{(t/t_{max})} \qquad (15)$$

The algorithm in pseudo code is given by:

Algorithm 2 : Pseudocode of the NG algorithm.

Input: A set of input vectors $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$

Output: A set of weight vectors $W = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N\}$

```
Set parameters λ0, λf, ε0, εf, hλ, N and t_max
Initialize all w ∈ W to random values
for i = 1 to t_max do
    Select random x ∈ D
    for j = 1 to N do
        Sort the weight vectors in W according to their distance to x
        Update the weights: w = w + εhλ(k(x, W))(x − w)
        Reduce learning rate ε
    end for
end for
```

Fig.4.The NG algorithm in pseudo code

V. THE FUZZY MEANS ALGORITHM

In traditional clustering the clusters are disjoint. Fuzzy C-means [14] is a fuzzy-based clustering algorithm that allows objects to belong to several clusters with different degrees of membership. In many practical situations fuzzy clustering is more natural to use than hard clustering.

In FCM the clusters are represented by the cluster centers. Each of the data has a degree of membership in each cluster center. In a plotting the vertical axis represents the membership function value u_j (x) corresponding to the cluster j and the horizontal axis represents the data items **x** to be clustered. In figure 5 there are three membership functions corresponding to the three clusters, denoted as low, medium and high.

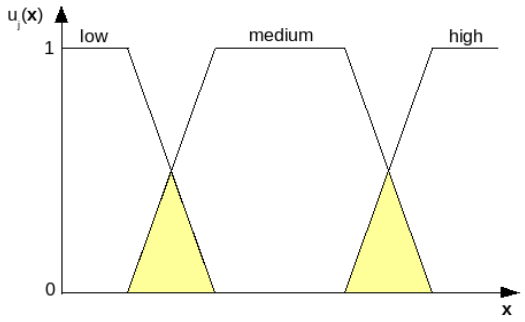


Fig.5.The membership functions of three clusters

FCM is based on fuzzy-partition and can be described as follows: Let $C = \{c_1,c_2,...,c_k\}$, be a set of cluster centers and $D = \{\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_n\}$ the set of given data where each object $\mathbf{x}_i = \{\mathbf{x}_{i1},\mathbf{x}_{i2},...,\mathbf{x}_{id}\}$ is a d-dimensional vector and U a k x k matrix. The matrix U is called a fuzzy k-partition and has the following constraints:

$$u_{ji} \in [0.1],\ 1 \le j \le k,\quad 1 \le i \le n \tag{16}$$

$$\sum_{j=1}^{k} u_{ji} = 1,\ 1 \le i \le n \tag{17}$$

$$\sum_{i=1}^{n} u_{ji} > 0,\quad 1 \le j \le k \tag{18}$$

where u_{ji} is the membership value of object $\mathbf{x}_i$ in cluster j. The constraint (17) implies that every object has some degree of membership in at least one cluster and that the sum of the objects' membership values is 1. The last constraint implies that each cluster contains at least one object to some degree, i.e. there are no empty clusters.

The aim of the FCM algorithm is to find an optimal fuzzy k-partition and corresponding cluster centers by minimizing the objective function [16].

$$J(U,C) = \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ji}^{\,m} \|x_i - c_j\|^2,\quad 1< m < \infty \tag{19}$$

In (19) m is a fuzzification constant that influences the membership values, c_j is a d-dimensional cluster center and ||.|| is the norm expressing the similarity between objects and cluster centers. The value of m determines the amount of fuzzification. The larger m is, the fuzzier the cluster membership is. An m value close to 1 expresses hard clustering.

The algorithm starts by generating randomly a fuzzy k-partition U^0. It then proceeds to update the cluster centers c_j by calculating

$$\mathbf{c_j}(t) = \frac{\sum_{i=1}^{n} (u_{ji}(t))^m \mathbf{x_i}}{\sum_{i=1}^{n} (u_{ji}(t))^m},\quad t = 1, 2, 3, \ldots, \tag{20}$$

where t is the current iteration step. Given the new cluster centers, the membership values are updated according to:

$$u_{ji}(t+1) = \frac{1}{\sum_{h=1}^{k} \left(\frac{\|\mathbf{x_i} - \mathbf{c_j}(t)\|}{\|\mathbf{x_i} - \mathbf{c_h}(t)\|} \right)^{\frac{2}{m-1}}} \tag{21}$$

The process stops when $\|U(t+1) - U(t)\| \le \varepsilon$ where ||.|| is the matrix norm and ε is a termination criterion between 0 and 1, or the maximum number of iterations t_{max} is reached. The algorithm in pseudo code is given by:

Algorithm 3 : Pseudocode of the FCM algorithm.
Input: A set of input vectors $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$
Output: The fuzzy partitioning of U and corresponding cluster centers

```
Set parameters ϵ, m, k and t_max
Initialize U to random values such that Σ_{j=1}^{k} u_ji = 1,  i = 1,2,...,n
repeat
  Calculate the cluster centers c_j = Σ_{i=1}^{n} (u_ji)^m x_i / Σ_{i=1}^{n} (u_ji)^m,  j = 1,2,...,k
  for i = 1 to n do
    for j = 1 to k do
      Calculate d(x_i, c_j) = ||x_i − c_j||
      if d(x_i, c_j) > 0 then
        u_ji = 1 / Σ_{h=1}^{k} ( d(x_i,c_j) / d(x_i,c_h) )^{2/(m−1)}
      else
        u_ji = 1
      end if
    end for
  end for
until ||U(t+1) − U(t)|| ≤ ϵ OR the maximum number of iterations t_max
```

Fig.6.The FCM algorithm in pseudo code

VI. COMPLEXITY

A. *Complexity of SOM*

Within the neigbourhood the weights of the SOM algorithm are adjusted by the updating procedure. This requires updating of every component of the weight vector and has O(d) complexity where d is the dimension of the weight vector. Adjusting the SOM map requires looping through the whole map where k is the number of nodes. Finding the BMU and adjusting the map are operations performed in every iteration. For a total of n iterations the SOM complexity is O(nkd).

B. *Complexity of NG*

The NG training process is very similar to the training process of SOM. To implement NG we have to compute the Euclidean distance and current neighbourhood, that means

- Compute the Euclidean distance between the input vector and every node
- Sort the nodes in ascending order according to the computed distance
- Adjust the weights. The closer a node is to an input vector the more the weight is adjusted.

With respect to the points above the complexity of computing the distances d is O(kd), where k is the number of nodes and is the dimension of the input vector. The nodes are then sorted according their distances. In an actual implementation of NG we may use *mergesort* with a complexity of O(klogk), where k is the number of elements. Adjusting the node weights requires looping through the list of nodes adjusting the weights. The gives a complexity of O(kd) as in SOM. For a total number of n iterations the total complexity becomes O(nkd + nklogk).

C. Complexity of FCM

The FCM algorithm is very different from SOM and NG algorithm. To implement FCM algorithm one has to

- Select an input sample from the data set
- Update the cluster centers
- Update the membership values
- Compute the distance between the current membership matrix and the previous and check for termination
- Copy the membership matrix for use in the next iteration

The updating of cluster centers and membership values are the most computationally expensive operations. Each cluster center is the mean of all input vectors weighted by their degree of belonging to the cluster. Updating one cluster center takes O(nd), where n is the number of input vectors and d is the dimensionality of input vector (or cluster center). Since this is done for every cluster center, the total complexity becomes O(knd), where k is the number of cluster centers.

The complexity of finding the Euclidean distance between cluster centers and input vector is O(d), where d is the dimensionality of the input vector. We need to compute the distance between the input value and every cluster centers which takes O(kd) time, where k is the member of cluster centers. For n input vectors and every cluster center the total complexity is $O(k^2nd)$ in the worst case.

The complexity of computing the matrix distance is done by minimizing their distance for every time the cluster centers and membership values have been updated. When this time is lower than a threshold value, the algorithm stops. The distance between the current membership matrix and the one on the previous iteration is computed in the same way as the Euclidean distance. This matrix norm is known as Frobenius norrn [13] and has a complexity of O(kn), where k is the number of cluster centers and n is the number of input vectors. The total complexity of FCM for a total of t iterations is therefore $O(tk^2nd)$ in the worst case, and thus more computationally heavy than both SOM and NG.

VII. THE SYSTEM

In the system developed we want that the SOM, NG and FCM algorithms to be run simultaneously in background while exploring and revealing cluster relationships hidden in the data. By using threads in java, for instance, one is able to run the different algorithms simultaneously.

By comparing the different clustering methods the user can get new insights into analysis of the data at hand. The system mainly performs two tasks:

1. Run the SOM, NG and FCM algorithm on a data set
2. Visualize the SOM, NG and FCM clustering of data

The first task involves the following steps:

- Importing of a data set into a data structure
- Choosing an algorithm and setting the parameters
- Running the algorithm on the data set

The second task involves:

- Translating the trained network to a graphical display such that the cluster relationships are revealed
- Displaying additional tools on request

A. Functional requirements

The input to the system is supposed to be a text file containing input objects. The column represents the attributes of the input objects. We assume that the attribute at the moment have numerical values. The attribute may be separated by a delimiter which can be a any character or just a blank.

In the visualization step we have to translate the internal data structure onto a graphical display by mapping each weight vector to a colour such that similar vectors are reflected in similar colours. Each algorithm is visualized in a separate window to make it easier to compare the different types of clustering. The system is scaled well and is able to handle data sets of different size (number of items) and dimensionalities. The parameter values (iterations, nodes, etc.) are not fixed, but depend on the size of dimensionality of the data set.

VIII. THE COLOR MODEL

The mapping of similar weights to similar colours is rather difficult since the weights are of higher dimension than the colour space. To be able to map objects of an arbitrary dimension onto colours we need to use an approach proposed by Schatzman [16]. In this approach an HSB model was developed where H = hue, S = saturation and B = brightness. This model describes the points inside an inverted cone (see figure 7). In a representation the hue, saturation and brightness are floating point values between 0 and 1. In the mapping of node weights to colours, the attribute values must be in the range from 0 to 1. To map a set of d-dimensional weight vectors to colours, one first has to divide the HSB colour circle at d equally spaced angles, given d colours

$$c_1, c_2, \ldots, c_d \tag{22}$$

Each of these colours is a three-dimensional vector with red, green and blue component.

$$\mathbf{c_i} = (c_{ir}, c_{ig,}, c_{ib})^T \tag{23}$$

Any d dimensional vector $\mathbf{w_i} = (w_1, w_2, ..., w_d)^T$ can now be mapped to a colour c_w by computing:

$$cw_r = \sum c_{ir} w_i / \sum c_{ir} \quad (24)$$

$$cw_g = \sum c_{ig} w_i / \sum c_{ig} \quad (25)$$

$$cw_b = \sum c_{ib} w_i / \sum c_{ib} \quad (26)$$

The mapping of every node in the map to a colour does not increase the overall complexity of the algorithm. In the SOM for instance to make d colours give a complexity of *O*(d). The mapping of k nodes to colours requires *O*(kd) which is less than the overall complexity of SOM.

The SOM, NG and FCM algorithms require different visualization regimes that reflect their behavior. Furthermore, implementation of NG and FCM, as opposed to SOM require dimension reduction to be mapped to the nodes (or cluster centers) onto a two-dimensional surface and therefore requires more computation.

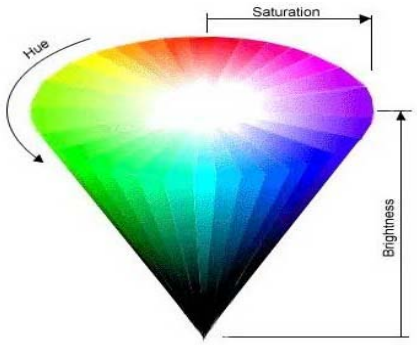


Fig. 7. Color mapping

IX. PROTEIN CLUSTERING

The three algorithms have been applied on clustering of protein mass spectrometry (MS) data [8,10] where the aim is to describe large-scale sets of proteins, their structure and function. MS has now become the platform for both identification and quantification of proteins. The basic principle of MS is to generate ions separated by their mass-to-charge ratio (m/z) which can be detected qualitatively and quantitatively by their m/z value and abundance.

MS produces large amounts of high-dimensional data that is difficult to analyze [5]. In this paper we use the three clustering algorithms to group the MS protein data in order to detect if a change in the environment causes a change in the clustering of the proteins of the human blood serum.

A. The data Set

The entire data set is made of ten smaller data sets containing MS data of proteins (or protein fragments) in the human serum. where each measurement is described by many attributes. These 10 data sets are further divided into two groups where five groups contain MS data representing proteins in the blood serum when the individual is sitting (S) and the other five while the individual is lying down (L). Our goal is to detect differences of these samples of data by comparing the clustering of S and L [9].

B. The Clustering of the Data Set

The difference in magnitude between the entries in the data sets were large, so they were preprocessed using a procedure known as z-score transformation [1].

The SOM clustering also contains the density of the input distribution. This is based on an idea developed by Simula [17] and aims to show the density of the input distribution in terms of map nodes. This is done by first locating the input vector's BMU after training. Then, the BMU's "hit" counter is incremented. Inside each square, the hit counter of the corresponding node is printed. If a node has zero hits the square remains entirely blank. This makes it easier to detect borders.

The following parameters were used for clustering of the different clustering algorithms:

SOM	**NG**	**FCM**
#iter:50 000	#iter:50 000	#iter:100
map size:10 x 12	# nodes:120	#cluster centers:3
Initial radius: 6	initial radius 60	fuzzification:2
Initial learn_ rate: 0.1	initial learn_ rate:0.5	stop crit: 0.01
	final radius: 0.01	
	final learn_rate: 0.005	

C. Visualization of the Sets L and S

The figures below show us the SOM, NG and FCM clustering of the L and S data set.

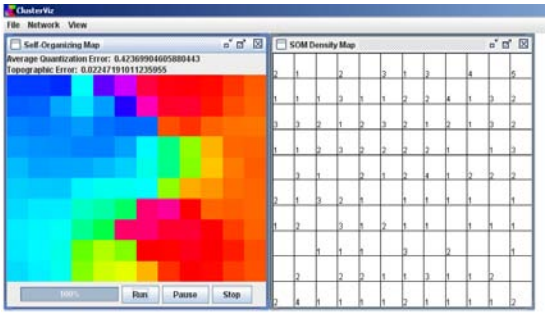


Fig. 8 SOM clustering of L data set

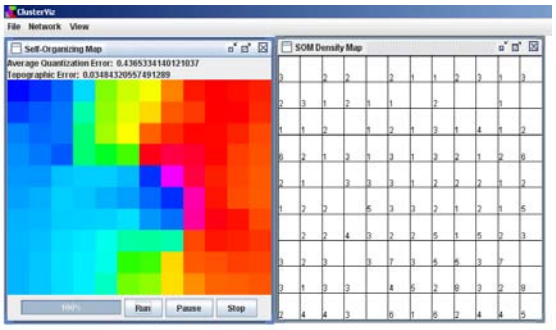
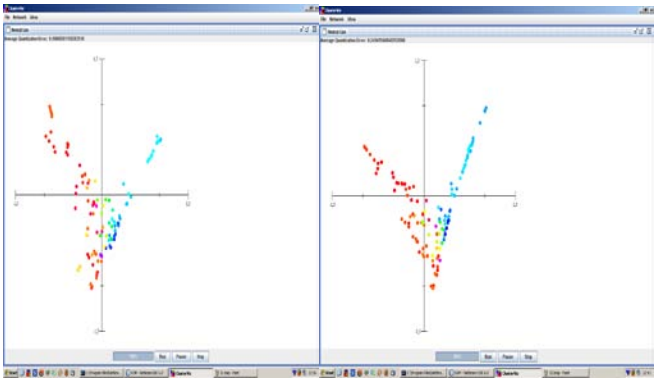


Fig. 9 SOM clustering of S data set

In the SOM clustering there is a red cluster in L that is larger than the corresponding one in S, but the red cluster in S seems to be more compact.

For NG we notice that the blue cluster of S is a little more compact than the blue cluster of L, and the blue cluster of S is larger. In addition, NG also contains a red cluster

which is similar to the SOM cluster found in both L and S.



a) b)

Fig. 10 a) NG clustering of L data set. b) NG clustering of S data

For the FCM we observe in figure 11 that L contains two red clusters represented by the cluster centers denoted by c_2 and c_3. On the contrary, in S in figure 12, the FCM clustering contains only one red cluster represented by c_3. The corresponding c_3 red cluster of L has a low hit number, so most of the data is contained in the c_2 and c_3 clusters of L. In S the situation is opposite, where the blue clusters c_2 and c_3 contain most of the data.

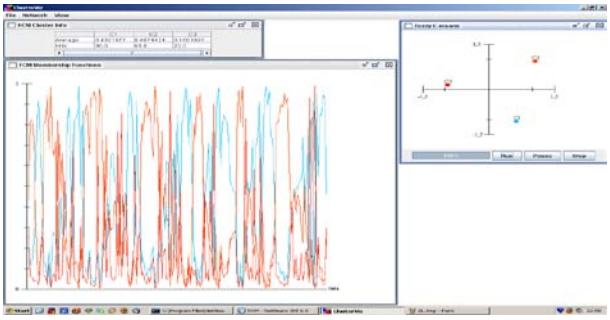


Fig.11. The FCM clustering of L data

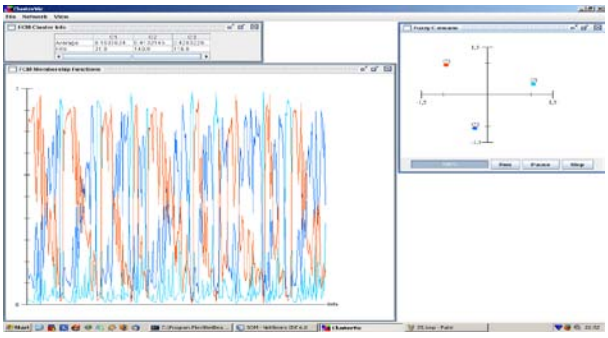


Fig12. The FCM clustering of S data

From all this we may conclude that the distribution of points seems to be clearly different in L and in S, and thus indicates a structural difference in the protein structure of the two groups.

X. Conclusion

A clustering system has been developed for multi-dimensional data. The system provides functionality for visualizing SOM, NG and FCM algorithms and for viewing them side by side to identify similarities in their structures. The system has been demonstrated on proteomic data of the blood serum. Use of all the different methods indicates that there exist structural differences in the protein structure when blood tests are taken in lying or sitting positions. This may be an important aspect to remember, for instance in connection with early determination of Parkinson disease, based on microarray gene expression data. The gene expressions will be different for lying and sitting positions.

References

[1] C. Cheadle, M.P. Vawter, W.J. Freed and K.G. Becker, "Analysis of Micro array data using z-transformation," in. Diagnostics, volume 5, no. 2, 2003.

[2] P.A Estevez,. "Data projection and visualization using self-organizing neural networks," EVIC 2005.

[3] P.A Estevez,, .C.J, Fiuer, K.Saito,. "Cross-entropy embedding of High-dimensional data using Neural Gas Model. Neural Networks," Vol 18, pp. 727-737, 2005.

[4] P.A Estevez, W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, no. 4, April 1965.

[5] J.H. Gross, "Mass Spectrometry," *Springer*, 2004.

[6] G. Gan, C. Ma, J. Wu. "Data Clustering Theory, Algorithms and Applications," Society for Industrial and Applied Mathematics, 2007

[7] S. Haykin, "Neural Networks and Learning Machines," *Third Edition. Pearson,* 2009.

[8] V. Jakobsen, T. Kristensen,." Interactive Clustering of Proteomic Data – a Comparison between Self-Organizing Map and Neural Gas,". in Proceedings of 6th Conference on Data Mining, DMIN'10, July 12-15, Las Vegas. CSCREA Press**,** pp. 350-356**,** 2010, USA.

[9] V. Jakobsen. "Data clustering with Visualization," Master thesis in Software Engineering," Department of Informatics, University of Bergen, 2010, Bergen, Norway.

[10] T. Kristensen,, V. Jakobsen , A.A Bjørkum,," Different Visualization Schemes of Protein Data Clustering, " in Proceedings of Fourteenth International Conference on Cognitive and Neural Systems (ICCNS 2010)," May 19-22, Boston, USA.

[11] T. Kohonen, O. Simula," Engineering Applications of Self-Organizing Map," IEEE volume 84, pp. 1368-1372, 1996.

[12] T.M.Martinetz, S.G. Berlkovich,, K.J. Schulten," Neural Gas network for vector quantization and its application to time-series," IEEE Transaction on Neural Networks, 4:559, 1993.

[13] C.D.Meyer," Matrix analysis and Applied Linear Algebra." Society for Industrial and Applied mathematics," 2001.

[14] S Nasciento, B. Mirkin, F.Moura-Pires, " a Fuzzy clustering model of data and fuzzy-means," Proceedings of IEEE Conference on Fuzzy Systems, 1:302-307, 2000.

[15] S. Obayashi, D. Saisaki, "Visualization and data mining of pareto solutions using self-organizing map," Second International Conference on Evolutionary Multi-Criterion Optimization (EMO 2003), Faro, Portugal, LNCS 2632, Springer-Verlag Berlin Heidelberg 2003, pp. 796–809, April 2003.

[16] J. Schatzmann, "Using self-organizing maps to visualize clusters and trends in multidimensional datasets," Master Thesis, Imperial College, London, U.K., pp. 17–24, 2003.

[17] O.Simula, P. Vasara, J. Vesanto, R.-R. Helminen, "Self-organizing map in industry analysis. " Intelligent Techniques in Industry, pp. 87–112, 1999.

[18] S. Theodoris, K. Kouuntroubas. Pattern Recognition. Academic Press, Third edition, 2006.

Noise-Tolerant Active Learning

Tsutomu Osoda[1], **and Satoru Miyano**[1]
[1]Information Science and Technology, The University of Tokyo, Shiroganedai 4-6-1, Minato-ku, Japan

Abstract— *Selection of unlabeled instances seriously affects the efficiency of active learning. Noise in data degrades efficiency because noise causes a predictive model to be inaccurate. To analyze this effect, we introduce a noise parameter to the expected log likelihood and formulate the objective function. Then, we define the density around a labeled instance to approximate the expected log likelihood with noise more precisely. Empirical experiments performed on the UCI data set show that the accuracy of noise-tolerant active learning is always the highest among several approaches under various noise conditions. Therefore, the noise-tolerant active learning algorithm is stably effective regardless of whether the data set includes considerable noise or not.*

Keywords: noise-tolerant; active learning; pool-based; expected log likelihood

1. Introduction

In an active learning algorithm [1], [2], [3], instances are selected from the unlabeled data set after a model for predictive classification has been built. These two steps are alternatively repeated until a good model [4], [5] is obtained. Selection of next instances is one aspect that determines the efficiency of an active learning algorithm [6], [7], [8]. For the analysis of such a selection, many researchers have formulated models for predictive classification mathematically [9], [10], [11].

The expected log likelihood of the labeled and unlabeled data measures the quality of a model for classification [12], [13], [14]. This is sometimes a starting point for selecting good instances theoretically. In discriminative batch mode active learning [15], an instance parameter is introduced to the expected log likelihood. In an entropy-based approach [16], instances with large variances are selected as next instances. In a density-based approach [17], [18], the density around an instance is defined and instances in the high-density area are selected. A combinatorial approach involving entropy-based and density-based approaches has also been proposed for text classification [19].

Numerical analysis shows that the density of a probability function strongly affects the formation of a model [20]. A model is built with a sharp change at the low-density region and with a mild change at the high-density region. Noise in data affects a model's predictive classification capabilities and sometimes degrades the efficiency of an active learning algorithm [21], [22].

We do not need to consider the noise at the high-density region because a model function with a mild change is less affected by noisy data. However, noise at the low-density region seriously affects a model because a model flexibly changes according to the noise in those instances.

2. Formulation and Methodology

2.1 Definition of Parameters

Before formulation, we defined the parameters as follows:

- L: Labeled data set
- U: Unlabeled data set
- y: Value of instances
- S: Set of unlabeled instances
- $f(S)$: Objective function

2.2 Formulation of the Objective Function

The strategy of noise-tolerant active learning is to select next instances with the highest score. The expected log likelihood is as follows:

$$\sum_{i \in L} \log P(y_i|x_i, w) + \alpha \sum_{j \in U} \sum_{y=\pm 1} P(y|x_j, w) \log P(y|x_j, w) \quad (1)$$

This function is defined as a binary classification. It is easy to expand the formulation to a multiclass classification. The weight parameter w changes to value w^t after t cycles. Then, the objective function 2 is formulated based on expected log likelihood 1.

$$f(S) = \max\Big(\sum_{i \in L^i \cup S} \log P(y_i|x_i, w^t) - \alpha \sum_{j \in U \setminus S} H(y|x_j, w^t) \Big) \quad (2)$$

Here, the function $H(y|x_j, w^t)$ is defined as the next entropy function.

$$H(y|x_j, w^t) = -\sum_{y=\pm 1} P(y|x_j, w^t) \log P(y|x_j, w^t)$$

Solving eqn. 2 as an integer problem is NP hard; therefore, we relax the constraints of a parameter by regarding the integer parameters as continuous.

$$\sum_{j \in U \setminus S} H(y|x_j, w^t) = \int_{x \in U \setminus S} H(y|x, w^t) g(x) dx$$

This function $g(x)$ represents the probability density function of x. Then, eqn. 2 is rewritten as eqn. 3.

$$f(S) = \max\Big(\sum_{i\in L^i\cup S} \log P(y_i|x_i,w^t) - \alpha\int_{x\in U\setminus S} H(y|x,w^t)g(x)dx\Big) \quad (3)$$

When a training data set includes noise, the noise influences the weight parameter. To analyze the influence of noise, we introduce a noise parameter to the expected log likelihood. The weight parameter w^t is divided into a noise-free part w^t and a noise-related part Δw^t.

Then, eqn. 3 can be rewritten as eqn. 4.

$$f(S) = \max\Big(\sum_{i\in L^i\cup S} \log P(y_i|x_i,w^t+\Delta w^t) - \alpha\int_{x\in U\setminus S} H(y|x,w^t+\Delta w^t)g(x)dx\Big) \quad (4)$$

We then suppose that the logistic regression model represents a model function [10].

$$P(y|x,w^t) = \frac{1}{1+e^{y_i w^t x_i}} \quad (5)$$

We approximate eqn. 4 with a second-order Taylor expansion.

$$\begin{aligned} f(S) = \max\Big(&\sum_{i\in L^i\cup S} \log P(y_i|x_i,w^t+\Delta w^t) \\ &-\alpha\int_{x\in U\setminus S} H(y|x,w^t)g(x)dx \\ &-\alpha\int_{x\in U\setminus S} \frac{\Delta w^t}{2} J(x)g(x)dx \\ &-\alpha\int_{x\in U\setminus S} \frac{\Delta w^t}{2} O((\Delta w^t)^2))g(x)dx\Big) \end{aligned} \quad (6)$$

Here, the function $J(x)$ is defined as follows:

$$J(x) = -\sum_{y=\pm 1} P(y|x_i,w^t)(1-P(y|x_i,w^t))(\log P(y|x_i,w^t)+1)$$

The first and second terms of eqn. 6 are associated with the noise-free part. The third and fourth terms of eqn. 6 are associated with the noise-related part. The noise-free part is equivalent to the original expected log likelihood. If the function $H(y|x,w^t)$ is approximated to a constant value, the information gain is constant for all instances. In this case, instances at the high density have a high score; hence, they are selected as the next training data. This strategy is referred to as density-based active learning. If the density function $g(x)$ is uniform, instances with high information gain $H(y|x,w^t)$ are selected as the next training data. This strategy is referred to as entropy-based active learning. When both density functions $g(x)$ and entropy functions $H(y|x,w^t)$ are variable, the strategy is to select instances obtained by the multiplication of the density function $g(x)$ and the entropy function $H(y|x,w^t)$ [23], [24]. On the other hand, discriminative batch mode active learning [15] also has been proposed for selecting multiple instances. In this method, the instance selection variable is introduced to the expected log likelihood. In the abovementioned methods, the influence of noise is not taken into account.

In noise-tolerant active learning, a noise parameter Δw^t is introduced. According to the paper [20], the change in a probability function $P(y|x_i,w^t)$ correlates with the density $g(x)$ of data. The value of the function $P(y|x_i,w^t)$ sharply changes at a low-density region and mildly changes at a high-density region. When a probability function is given as a logistic regression, the weight parameter determines the properties of the probability function. Therefore, the density correlates with this weight parameter. The weight parameter becomes larger as the change in the function value becomes sharper, that is, as the density of instances becomes lower. Conversely, the weight parameter becomes smaller as the density of data becomes higher. This relationship between the density function and the change in the weight parameter is maintained, regardless of whether noise causes this change or not. Here, suppose that this relationship is approximately represented as an inverse proportion.

$$w^t + \Delta w^t \propto \frac{1}{h(x)}$$

Then, eqn. 6 can be rewritten in eqn. 7.

$$\begin{aligned} f(S) = \max\Big(&\sum_{i\in L^i\cup S} \log P(y_i|x_i,w^t+\Delta w^t) \\ &-\alpha\int_{x\in U\setminus S} H(y|x,w^t)g(x)dx \\ &-\beta\int_{x\in U\setminus S} (\frac{g(x)}{h(x)} - w^t)(H(y|x,w^t)+J(x))dx \\ &-\gamma\int_{x\in U\setminus S} (\frac{g(x)}{h(x)} - w^t)O((\Delta w^t)^2))dx\Big) \end{aligned} \quad (7)$$

The parameters β,γ are constant. Here, suppose that the function $J(x)$ is constant C.

$$\begin{aligned} f(S) = \max\Big(&\sum_{i\in L^i\cup S} \log P(y_i|x_i,w^t+\Delta w^t) \\ &-\alpha\int_{x\in U\setminus S} H(y|x,w^t)g(x)dx \\ &-\beta\int_{x\in U\setminus S} (\frac{g(x)}{h(x)} - w^t)(H(y|x,w^t)+C)dx \\ &-\gamma\int_{x\in U\setminus S} (\frac{g(x)}{h(x)} - w^t)O((\Delta w^t)^2)dx\Big) \end{aligned}$$

The parameter w^t is also constant. Then, we redefine the

constant parameters β, γ.

$$\begin{aligned} f(S) &= \max(\sum_{i \in L^i \cup S} \log P(y_i|x_i, w^t + \Delta w^t) \\ &\quad -\alpha \int_{x \in U \setminus S} H(y|x, w^t) g(x) dx \\ &\quad -\beta \int_{x \in U \setminus S} \frac{g(x)}{h(x)} (H(y|x, w^t) + C) dx \\ &\quad -\gamma \int_{x \in U \setminus S} \frac{g(x)}{h(x)} O((\Delta w^t)^2) dx) \end{aligned} \quad (8)$$

The strategy of noise-tolerant active learning is to select instances that allow maximization of the objective function 8.

2.3 Definition of Density

We define the density around an instance as the number of instances within some distance from it. In the case of unlabeled instances, the density of an instance is defined as follows:

$$density(x, h) = \#x_i \ s.t. \ |x - x_i| \leq h, \ x, \ x_i \in U$$

Similarly, in the case of labeled instances, the density of an instance is defined as follows:

$$density(x, h) = \#x_i \ s.t. \ |x - x_i| \leq h, \ x, \ x_i \in L \quad (9)$$

Distance $|x_i - x|$ is discrete metric. To avoid zero division, the function $h(x)$ is defined as $density(x, h) + 1$:

3. Empirical Experiments

3.1 Data Set for Evaluation

Four kinds of data sets from UCI [25] are used for evaluation. Details of the evaluation data sets are as follows.

- kind(data, #descriptor, #kinds of value (number of class instances))
- car (1728, 6, 4 (1219, 384, 69, 65))
- crx (690, 15, 2 (307, 383))
- flare (1369, 10, 8 (884, 112, 33, 20, 9, 4, 3, 1))
- australian (690, 14, 2 (383, 307))

Each data set is uniformly separated into five groups for cross validation. One data set contains labeled data, one data set contains validation data, and the other data sets contain unlabeled data. Active learning starts with the labeled data set and then selects next instances from the unlabeled data sets. We select five-fold cross validation to reduce the effect of the initial data set. To evaluate the accuracy of prediction, the validation data set is used at each cycle. The active learning algorithm used is a query by the bootstrap aggregation method [26], [2]. The classification algorithm used is a decision tree [27]. Accuracies of predictions are found to be different (Table 1). The "car" problem is the most difficult, and the "crx" problem is the easiest among these four problems (Table 1). The difficulty of a problem is mainly determined by the power of the algorithm, descriptors of the data, and noise in the data. When the algorithms and descriptors are fixed, the noise in the data is the main difficulty. Therefore, the "car" problem includes the noisiest data among these four problems. For a problem with less noise, the final accuracy is higher. Accuracy of prediction increases as the number of iterations becomes larger (Figure 1). An entropy-based approach is better than a random-based approach when the final accuracy is high. In the case of a logistic regression model, the information gain of entropy-based sampling is greater than that of random-based sampling [1]. This theory supports this experimental result. For example, in the case of the "Australian" problem, accuracy of prediction is relatively high (Table 1). Accuracy of entropy-based active learning is higher than that of random-based active learning (Figure 1). Conversely, an entropy-based approach is worse than a random-based approach when the final accuracy is low. The volume of noise causes the differences in accuracy among these four problems. Entropy-based active learning selects instances with a large variance among decision trees. The variance of an instance around a high-noise area is large because the designed model is flexible around a noisy area. When the volume of noise is small, an entropy-based approach is better than a random approach. Conversely, when the volume of noise is high, a random approach is better than an entropy-based approach. Noise-tolerant active learning is efficient for all of the data sets (Figure 1). Therefore, noise-tolerant active learning is stably better regardless of whether the data set includes considerable noise or not.

Table 1: Accuracy rate of prediction on each evaluation problem. Accuracy rate at Car case is 60%. This accuracy is the lowest in these four problems. Accuracy of crx case is 84%. This accuracy is the highest in these four problems.

kind	Accuracy(%)
car	73
crx	85
flare	80
australian	80

4. Conclusion

Noise can sometimes affect a learning model. To analyze the effects of noise, we introduced a noise parameter to the expected log likelihood of the labeled and unlabeled data. Then, we formulated an objective function for noise-tolerant active learning. The density of the training data was defined to approximate the objective function more precisely. Empirical experiments performed on the UCI data sets showed that the accuracy of noise-tolerant active learning was stably higher than that of traditional algorithms.

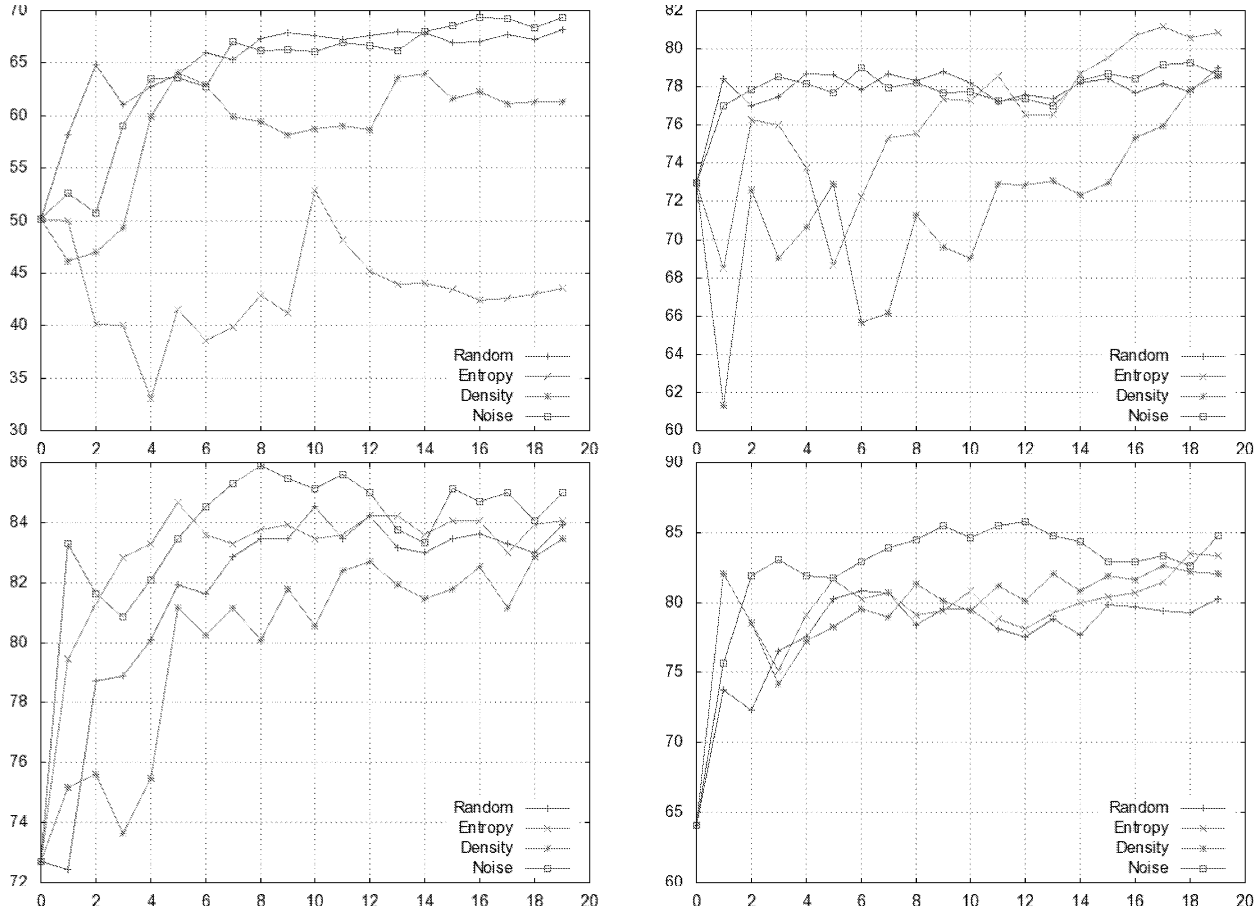


Fig. 1: Accuracy rate at each steps in each problem. Car (Left Upper), Flare (Right Upper), Crx (Left Lower), Australian (Right Lower). The x-axis line shows the number of iterations. The y-axis shows the average accuracy of an algorithm. Every lines show the average of five-folds. Each curve corresponds to random selection, entropy-based selection, density-based selection and noise tolerant AL selection. $+, \times, *, \diamond$ indicate each accuaracy rate, respectively. Accuracy rate becomes higher as number of iteration is larger and converges to last accuracy rate (Table 1).

References

[1] B. Settles, "Active learning literature survey," University of Wisconsin.Madison, Tech. Rep. 1648, 1 2009.

[2] K. U. Nicolas Majeux and H. Mamitsuka, "Prediction of mhc class i binding peptides using an ensemble learning approach," *Genome Informatics*, vol. 14, pp. 687–688, 2003.

[3] J. Y. Jun Long and E. Zhu, "An active learning method based on most possible misclassification sampling using committee," *Modeling Decisions for Artificial Intelligence*, vol. 4617, pp. 104–113, 2007.

[4] H. W. S. L. Shenyang, Liaoning and E. Hovy, "Multi-criteria-based strategy to stop active learning for data annotation," in *Proceedings of the 22nd International Conference on Computational Linguistics*, vol. 1, 2008, pp. 1129–1136.

[5] H. W. Jingbo Zhu and M. del Rey, "Confidence-based stopping criteria for active learning for data annotation," in *ACM Transactions on Speech and Language Processing*, vol. 6, 4 2010.

[6] T. Y. Jingbo Zhu, Huizhen Wang and B. K., "Active learning with sampling by uncertainty and density for word sense disambiguation and text classification," in *Proceedings of the 22nd International Conference on Computational Linguistics*, vol. 1, 2008, pp. 1137–1144.

[7] S. Dasgupta, "Coarse sample complexity bounds for active learning," in *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Scholkopf, and J. Platt, Eds. MIT Press, 2006, pp. 235–242.

[8] N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction," in *Proceedings of the 18th International Conference on Machine Learning*. San Francisco: Morgan Kaufmann, 2001, pp. 441–448.

[9] T. Kanamori, "Pool-based active learning with optimal sampling distribution and its information geometrical interpretation," *Neurocomputing*, vol. 71, no. 1-3, pp. 353–362, 12 2007.

[10] A. I. Schein and L. H. Ungar, "Active learning for logistic regression: an evaluation," *Machine Learning*, vol. 68, no. 3, pp. 235–265, 10 2007.

[11] I. Muslea and S. Minton, "Active learning with multiple views," *Journal of Artificial Intelligence Research*, vol. 27, pp. 203–233, 10 2006.

[12] T. Zhang and Oles, "A probability analysis on the value of unlabeled data for classification problems," 2000, pp. 1191–1198.

[13] J.-T. Huang and M. Hasegawa-Johnson, "On semi-supervised learning of gaussian mixture models for phonetic classification," in *Proceedings of the NAACL*, 2009, pp. 75–83.

[14] T. Zang, "The value of unlabeled data for classification problems," in *Proceedings of the Seventeenth International Conference on Machine Learning*. Morgan Kaufmann, 2000, pp. 1191–1198.

[15] Y. Guo and D. Schuurmans, "Discriminative batch mode active learning," in *Proceedings of Advances in Neural Information Processing Systems*, vol. 20. Cambridge: MIT Press, 2008, pp. 593–600.

[16] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," Cambridge, MA, 2005.
[17] P. Donmez and J. Carbonell, "Paired-sampling in density-sensitive active learning," in *Proceeding 10th International Symposium on Artificial Intelligence and Mathematics*, Florida, 2008.
[18] K. Brinker, "Incorporating diversity in active learning with support vector machines," in *Proceedings of the Twentieth International Conference on Machine Learning*. AAAI Press, 2003, pp. 59–66.
[19] Z. C. Bishan Yang, Tengjiao Wang, "Effective multi-label active learning for text classification," in *KDD '09 Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 917–926.
[20] M. Szummer and T. Jaakkola, "Information regularization with partially labeled data," in *In Advances in Neural Information Processing Systems 15*, vol. 15. MIT Press, 2003.
[21] R. J. Tianbao Yang and A. K. Jain, "Learning from noisy side information by generalized maximum entropy model," in *Proceedings of 27th International Conference on Machine Learning*, 2010.
[22] D. R. Sariel Har-Peled and D. Zimak, "Maximum margin coresets for active and noise tolerant learning," in *Proceedings of the 20th international joint conference on Artifical intelligence*, 2007, pp. 836–841.
[23] A. K. McCallum and K. Nigam, "Employing em and pool-based active learning for text classification," in *In Proceedings of the 15th International Conference on Machine Learning*. Morgan Kaufmann, 1998, pp. 350–358.
[24] R. J. Steven C. H. Hoi and M. R. Lyu, "Large-scale text categorization by batch mode active learning," in *Proceedings of the 15th international conference on World Wide Web*, Edinburgh, Scotland, 5 2006, pp. 633–642.
[25] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml
[26] L. BREIMAN, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.
[27] J. R. Quinlan, "C4.5: Programs for machine learning," *Morgan Kaufmann*, 1993.

Breast cancer risk score: a data mining approach to improve readability

Emilien Gauthier, Laurent Brisson, Philippe Lenca, Stéphane Ragusa

***Abstract*— According to the World Health Organization, starting from 2010, cancer will become the leading cause of death worldwide. Prevention of major cancer localizations through a quantified assessment of risk factors is a major concern in order to decrease their impact in our society. Our objective is to test the performances of a modeling method easily readable by a physician. In this article, we follow a data mining process to build a reliable assessment tool for primary breast cancer risk. A *k*-nearest-neighbor algorithm is used to compute a risk score for different profiles from a public database. We empirically show that it is possible to achieve the same performances than logistic regressions with less parameters and a more easily readable model. The process includes the intervention of a domain expert who helps to select one of the numerous model variations by combining at best, physician expectations and performances. A risk score is made up of four parameters: *age*, *breast density*, *number of affected first degree relatives* and *prone to breast biopsy*. Detection performance measured with the area under the ROC curve is 0.637.**

I. Introduction

As cancer is becoming the leading cause of death worldwide, prevention of major types of cancer through a quantified assessment of risk is a major concern in order to decrease its impact in our society. Physicians have to inform patients about risk factors and have to detect fatal diseases as soon as possible in order to treat them as quickly as possible. Nowadays, this detection is led by prevention programs designed to target highest-risk subsets of the population. For example, women over 50 years old in France and over 40 in USA are recommended to perform a mammography every two years to detect breast cancer; mammography being the primary method for detecting early stage breast cancer which is the first cause of cancer for women [1]. As a consequence, our society could benefit from a widely used risk score in order to give more accurate counseling on how cancer is impacted by risk factors and to target smallest subset of the population with higher risks. For example, using age at first mammogram as an actionable variable, screenings programs for breast cancer could be extended: younger women with high risk profiles could be offered more frequent screenings in order to decrease death risk [2].

Emilien Gauthier, Laurent Brisson and Philippe Lenca are with the Institut Telecom, Telecom Bretagne, UMR CNRS 3192 Lab-STICC, Technopôle Brest Iroise CS 83818, 29238 Brest Cedex 3, France (phone: 33 2 29 00 11 75; fax: 33 2 29 00 10 30; email: {emilien.gauthier || laurent.brisson || philippe.lenca} @telecom-bretagne.eu).

Emilien Gauthier and Stéphane Ragusa are with the Statlife company, Institut Gustave Roussy, 114 rue Edouard Vaillant, 94805 Villejuif Cedex, France (phone: 33 1 42 11 51 84; fax: 33 1 42 11 40 00; email: {emilien.gauthier || stephane.ragusa} @statlife.fr).

Even if some women may have genetic predisposition for breast cancer, environmental factors may have a larger impact on the risk according to Lichtenstein [3]. Because of this impact and due to acquisition cost and easyness-to-use constraints, we have decided to focus on environmental factors as attributes to compute a risk for women who never had breast cancer.

As pointed out by [4], "*information, dialog and more patient involvement in the decision-making process*" are key words in dealing with cancer, therefore a major challenge in the field of medical counseling is to provide physicians and radiologists with adequate tools to help them to assess breast cancer risk of their patients and to show easily how risk factors impact global risk. For many years, risk scores built upon statistical models did not reach to spread in medical counseling domain despite their performances. This may be because end-users of these tools are not oncologist nor clinician and underlying models are too complex and too difficult to use during a medical consultation. Thus, to build a new risk score tool, we need to consider the model readability and the current medical decision process. Moreover, we will have to consider the obligation to use imbalanced datasets with missing data. To the best of our knowledge, no one has been interested in analyzing, with a mining approach, data from women who never had cancer in order to create a risk score with a prevention purpose.

Showing similar cases may improve communication with the patient, therefore increase its involvement in the prevention and decision process. Because core concept of k-nearest-neighbor algorithm is to gather similar profiles using a distance computation, we use it with help of a domain expert in order to build a tool to predict breast cancer risk and measure its performances.

The paper is organized in six sections. Section II provides an overview of related works on risk models. Section III describes source data and Section IV presents our approach of the data mining process we follow. In section V, we present results, discuss them and present future works.

II. Breast cancer risk scores

A. Statistical approaches

We present studies focusing on prevention and the use of environmental factors such as reproductive and medical history. One risk prediction model emerges in the statistical field.

Based on an unstratified, unconditional logistic regression analysis, the most commonly used model was developed by

Gail *et al* [5] using data from the *Breast Cancer Detection Demonstration Program*. Risk factor information was collected during a home interview and the analysis was based on approximately 6,000 cases and controls. Among 15 risk factors obtained through patient interviews, only 5 were chosen: age, age at menarche (first natural menstrual period), number of previous breast biopsies, age at first live birth and number of first-degree relatives with breast cancer. The model lead to the computation of a cumulative risk of breast cancer by multiplying each of the five relative risks. Then, individual risk of breast cancer is obtained by multiplication of the cumulative risk score by an adjusted population risk of breast cancer. Gail's risk score was validated on the population of United States with the *Cancer and Steroid Hormone Study* (CASH) by Costantino *et al* [6] and in Italy on the *Florence-EPIC Cohort Study* by Decarli *et al* [7].

Barlow *et al* [8] also built a risk prediction model using a logistic regression on the *Breast Cancer Surveillance Consortium* (BCSC) database (see Table I and *http://breastscreening.cancer.gov*) which contains 2.4 millions screenings mammograms and associated self-administered questionnaires (see section III). Two logistic regression risk models were constructed with 4 or 10 risk factors depending on the menopausal status. Compared to Gail's model, it gains the use of breast density and hormone therapy. As we will use the same database, it is worth highlighting that reported area under ROC curve (see performance measurement in section IV-D) was 0.631 for premenopausal women and 0.624 for postmenopausal women.

Primary goal of these studies was not readability, but rather highest risk detection performances and impact levels of each risk factors.

B. Data mining approaches and imbalanced data

Most similar data mining approaches dealt with slightly imbalanced data, mostly used to predict a cancer relapse as a result of the *Surveillance, Epidemiology and End Results* (SEER) database use. Here, we present two significant related studies involving both medical data and mining algorithm.

Endo *et al* [9] implemented common machine learning algorithms to predict survival rate of breast cancer patient. This study is based upon data of the SEER program with high rate of positive examples (18.5 %). Since this study aims at classifying examples in two classes, authors did not used ROC curve to assess performances results but accuracy, specificity and sensitivity. Logistic regression had the highest accuracy, artificial neural network showed the highest specificity and J48 decision trees model had the best sensitivity.

Jerez-Aragonés *et al* [10] built a decision support tool for the prognosis of breast cancer relapse. They used similar attributes as Gail (like age, age at menarche or first full time pregnancy, see section II-A) but also biological tumor descriptors. A method based on tree induction was conceived to select the most relevant prognosis factors. Selected attributes were used to predict relapse with an artificial neural network by computing a Bayes *a posteriori* probability in order to generate a prognosis system based on data from 1,035 patients of the oncology service of the Malaga Hospital in Spain .

Such studies show how mining approaches can be used to built classification tools on medical databases while dealing with missing data and business processes. But they do not consider problems (such as readability) encountered by patients who never had cancer nor physicians in their day to day interactions.

TABLE I
BCSC DATABASE PUBLICLY AVAILABLE ATTRIBUTES

Full name	Short name	Description & coding
Menopausal status	menopaus	Premenopausal or postmenopausal
Age group	agegrp	10 categories from 35 to 84 years old
Breast density	density	BI-RADS breast density codes
Race	race	White, Asian/Pacific Islander, Black, Native American, Other/Mixed
Being hispanic	hispanic	Yes or no
Body mass index	bmi	4 category from 10 (underweight) to 35 and more (obese)
Age at first birth	agefirst	Before or after 30 at first live birth or nulliparous (i.e. no children)
First degree relatives	nrelbc	Number of first degree relatives with breast cancer 0, 1 or more than 2
Had breast procedure	brstproc	Prone to breast biopsy, yes or no
Last mammogramm	lastmamm	Last mammogram was negative or false positive
Surgical menopause	surgmeno	Natural or surgical menopause
Hormone therapy	hrt	Being under hormone therapy
Cancer status	cancer	Diagnosis of invasive breast cancer within one year, yes or no

To build a risk score, we have to detect highest risk profiles among general population. It means we are facing highly imbalanced data with a breast cancer incidence rate lower than 1 000 new cases for 100 000 women. Dealing with such imbalanced data can be done at two levels [11], [12], [13].

At the algorithmic level, assuming all errors have a different cost is a solution to guide the data mining process [14], especially in the medical field where detecting an high risk profile is more informative than detecting a low risk profile. At the data level, sampling is another solution. A first way to rebalance data is to decrease the number of negative examples (under-sampling) [15]. And a second way of rebalancing data is to increase number of positive examples (over-sampling) [16].

III. Data source

To ensure result reproducibility, we have to choose a public database with environmental factors. The Breast Cancer Surveillance Consortium (BCSC) makes available a database that fits those major constraints. Each of the 2,392,998 lines match to a screening mammogram for a woman. This

TABLE II
MISSING DATA LEVEL BY ATTRIBUTE

Attribute	Missing data level
Body mass index	55.9 %
Age at first birth	55.5 %
Surgical menopause	52.1 %
Hormone therapy	41.0 %
Breast density	26.3 %
Last mammogramm	23.4 %
Being hispanic	20.3 %
Race	15.9 %
First degree relatives	15.2 %
Had breast procedure	10.5 %
Menopausal status	7.6 %
Age group	0 %
Cancer status	0 %

TABLE III
BREAST CANCER INCIDENCE RATE PER 100 000

Age category	SEER rate (2003-2007)	BCSC rate (1996-2002)
35-39	58.9	142.7
40-44	120.9	168.1
45-49	186.1	250.5
50-54	225.8	360.7
55-59	280.2	436.4
60-64	348.9	478.5
65-69	394.2	512.3
70-74	410.0	575.1
75-79	433.7	632.0
80-84	422.3	709.4
85+	339.2	Unavailable

publicly available database provides 12 attributes to describe the woman including cancer status.

A. BCSC database: data collection

Originally, the consortium was conceived to enhance understanding of breast cancer screening practices [17]. The consortium aims at establishing targets for mammography performance and a better understanding of how screenings affect patients in term of actions taken after the mammography. Domain experts from the surveillance consortium identified critical data elements for evaluating screenings performances reaching a consensus on a standard set of core data variables. Then, from 1996 to 2002, data were collected in seven centers across the United States: mammograms and their detailed analysis were collected and, at the same time, women were asked to complete a self-administrated questionnaire.

BCSC database provides personal factors (see Table I) such as factual factors (age, race, body mass index), reproductive history (age at first birth, menopausal status, hormone therapy) and medical history (number of first degree relatives with breast cancer or type of menopause). In addition, breast density was recorded when the classic Breast Imaging Reporting and Data System (BI-RADS) [18] was used by the radiologist. To ensure good quality of data, exclusion rules were set: for example, women who have undergone cosmetic breast surgery were excluded as well as women with previous breast cancer and women with no known prior mammogram.

Eventually, breast cancer cases were identified by linking cancer registries to BCSC database, i.e. for each record of the database, the class of the example is positive if the corresponding women was diagnosed with breast cancer within one year after the mammogram and completing the questionnaire and negative otherwise.

B. BCSC database: exploratory analysis

Among the 2,392,998 records of the database, 9,314 cases of invasive breast cancer were diagnosed in the first year of follow up. We are facing highly imbalanced data with a positive class accounting for only 0.39 % of all records.

We also observe a high level of missing data (see table II). Two main reasons explain missing data:

- Data were collected in different registries with non-standardized self-reported questionnaire: some questions were not asked and for any question, each woman had the possibility not to answer.
- Collection of some risk factors did not start at the same time. For example, height and weight were added later, explaining such a high rate of missing data for the body mass index.

Last, one has to notice that data of the BCSC are not representative of the USA breast cancer incidence rate (number of new cases during a specified time for a given population). Table III offers a comparison between the BCSC and the SEER incidence rate [19] by age categories.

Indeed, depending on data sources, the breast cancer incidence usually increase slowly from approximatively 60 to 80 years old and starts to decrease after 80 years old. But such a slower increase or decrease does not occur in the BCSC database.

IV. PROCESS TO BUILD A RISK SCORE

A. Main objectives

The main objective of our approach is to provide physicians with a tool to assess a cancer risk score for their patient and to promote dialog between them. As statistical models spread with difficulty in the physician community, we aim to find models with good scoring performance and good readability. In our case, we say a model has a good readability if it allows a physician to explain the risk score to his patient:

- it has to be quickly readable by a physician during a medical appointment
- and has to give access to understanding the score,

Furthermore, we have other constraints: physicians have *a priori* ideas about good attributes of a model, patients need actionable attributes to change their lifestyle, both of them want immediately usable score (i.e. very low cost of data acquisition). In addition, a generic algorithm that can be easily adapted to various pathologies is desirable.

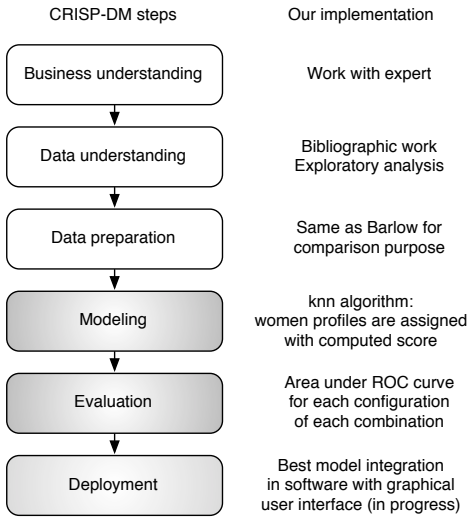


Fig. 1. General process based on CRISP-DM methodology - Gray steps identify our major contributions

B. General process

Our approach follows the CRoss Industry Standard Process for Data Mining (CRISP-DM) [20] data-mining methodology. Figure 1 shows the 6 steps of this process where gray ones identify our major contributions. Business and data understanding steps are not impacted because we want to work on the same data as [8] to be able to compare our results.

1) Business understanding: An expert with knowledge of the needs of physicians help us to prioritize our objectives (see section IV-A) and to assess the situation. We decide to focus on a scoring task (no classification or prediction).

2) Data understanding: Despite limitations described in section III, the BCSC database contains most of the known breast cancer personal factors. It is the largest database publicly available that includes breast density information.

3) Data preparation: To deal with data imbalance, we can apply rebalancing algorithms on this data but it is not the focus of the paper. We do want to minimize modification of data in order to compare our results with Barlow's. The only modification we apply is normalization. It was decided to keep the same split between training and validation set.

4) Modeling: If several data mining algorithms were considered, domain expert suggested to use a k-nearest-neighbor algorithm because it uses a concept of similarity which is easily understandable by end-users without explaining a complex formula. Moreover, such algorithm is able to deal with imbalanced data if there is enough positive examples among neighbors. We generate models and search for the best combination of attributes by performing an exhaustive search (see section IV-C) on a limited set of combinations. The reason is that the expert issued a recommendation of using a restricted number of factors to make the risk score easy to use. Obviously, for large combinations, computation time can increase sharply, but it is not a problem as models are generated offline only once by us, when a physician uses the final software, no computation is necessary.

5) Evaluation: We evaluate generated models with Receiver Operating Characteristic validation (see section IV-D) using Area Under Curve (AUC) in order to sorts models by scoring performance. Then, our expert has to choose the most useful models leveraging on the AUC performance combined with its knowledge of physician needs. ROC evaluation of every generated model is automatized in our software but we have to improve our process to formalize and support expert choice.

6) Deployment: We are currently working to incorporate selected model configuration into a computer software tool for physicians. It will come with a graphical explanation of the concept of nearest neighbor. But it will not embed the database.

C. Focus on k-nearest-neighbor implementation

To provide experts with several interesting models, k-nearest-neighbor algorithm (see [21], [22]) is used with various size of attributes combinations (from 1 to 6 attributes), several Minkowski generalized distance measure ($p = 1$ to 5) and several k values were used (see section V). Performance of each of hundreds generated combinations is tested for each values of k.

We implement the k-nearest-neighbor algorithm in two steps:

- Selection of neighborhood: for a combination of attributes (e.g. *age* and *breast density*), a score value has to be computed for each combination of values (e.g. *age=5* and *breast density=3*). To compute such score value, a neighborhood has to be defined for each values combination. To determine if a profile of the database belong to the neighborhood of a combination of values, an euclidian distance is used to compute the distance between a combination of value and every single record of the database using a normalized version of the coding values of the BCSC database. Thus, at least k of the nearest records of the database are included in the neighborhood. The neighborhood may not have always the same size because for a given group at the same distance, if k is not reached yet, all neighbors at the same distance are added to the neighborhood.
- Scoring function: the score of a combination of values, is the ratio between the number of breast cancer cases (i.e. positive examples) and the size of the neighborhood. In epidemiology, the ratio of individuals having a disease in a population is called prevalence. This ratio was chosen because it is well known by physicians, easily explainable to a patient and it is directly built on the number of patient diagnosed with breast cancer among patients with a similar profile.

To deal with missing data, we keep the same decision as Barlow, i.e. assign a high value when missing. It will prevent a record with a missing value to be integrated in the neighborhood.

TABLE IV
BEST PERFORMANCES BY COMBINATION SIZE

Size	Combinations	AUC Mean	AUC Std Deviation	AUC Median	Best combination (See Table I)	AUC
1	12	0.536	0.030	0.529	agegrp	0.614
2	66	0.563	0.031	0.553	agegrp+density	0.635
3	220	0.581	0.029	0.601	agegrp+density+brstproc	0.641
4	495	0.593	0.026	0.597	agegrp+density+brstproc+lastmamm	0.642
5	792	0.602	0.023	0.586	agegrp+density+brstproc+lastmamm+menopaus	0.642
6	924	0.607	0.019	0.603	agegrp+density+brstproc+lastmamm+hrt+nrelbc	0.637

D. Focus on ROC evaluation

The Receiver Operating Characteristic (ROC) [23] is used to measure performance due to the continuous nature of our classifier: performance has to depict how positive instances are assigned with higher scores than negative ones. The ROC curve allows to measure detection performances using a moving threshold to classify examples of the validation set. Moreover, it allows direct comparison with Barlow's results and epidemiological-based scores in general.

Negative examples labeled as positive by the algorithm are called a false positives whereas positive examples labeled as positives are called true positives. The ROC curve is plotted with the false positive rate on the X axis and the true positive rate on the Y axis [24], both rates being calculated for a given threshold. It can be summarized in one number: the Area Under the ROC Curve (AUC). The area being a portion of the unit square, its value is in then [0,1] interval. The best classifier will have an AUC of 1.0 (i.e. all positive examples are assigned with higher score than negative ones) whereas an AUC of 0.5 is equivalent to random score assignment.

Each k value of each combination of attributes is assigned with a ROC curve and the corresponding AUC in order to help the expert to choose the best model.

V. EXPERIMENTAL RESULTS

A. Scoring performances

An experiment set was designed to test how the k-nearest-neighbor algorithm perform on the BCSC data. As one of our constraint is to build a readable risk score (see section IV-A), we select all combinations with a size s of 1 to 6 attributes among $n = 12$ available attributes, meaning we have $\sum_{s=1}^{6} \frac{n!}{s!(n-s)!} = 2,509$ combinations to test. A first way of assessing results of these combinations is to look at the best combinations by size (see Table IV). These results are obtained in an euclidian space using a 2-norm euclidian distance as they are not significantly better, when improved, using another p-norm measures.

Among one attribute combinations, *agegrp* is by far the best factor to score breast cancer risk in the BCSC database with an AUC of 0.614, while the next best attribute (not shown), *menopaus* for menopausal status, performs only at 0.563. This result confirms expert knowledge since it's widely known that age is a major breast cancer risk factor.

For combinations size from 1 to 3 attributes, mean, median and best AUC rise, whereas for sizes of 4 and 5 attributes, maximal performances level off around 0.64 with a slight decrease with 6 attributes for best combinations. It is interesting to obtain the best results using less possible attributes to improve model readability. Furthermore, our 3 attributes *agegrp, density, brstproc* combination has an AUC of 0.641 while in Barlow's results (see section II-A), at least 4 attributes are needed to achieve an AUC of 0.631 on a subset of data that includes only premenopausal women only.

TABLE V
TOP 15 PERFORMANCE RESULTS BEFORE AND AFTER EXPERT ADVICE

A. **Best combinations before expert advice**	AUC
agegrp, lastmamm, density, brstproc	**0.642**
menopaus, agegrp, lastmamm, density, brstproc	0.642
agegrp, density, brstproc	0.641
menopaus, agegrp, density, brstproc	0.641
bmi, agegrp, density, brstproc	0.640
bmi, agegrp, lastmamm, density, brstproc	0.640
agegrp, hispanic, density, brstproc	0.640
agegrp, density, brstproc, agefirst	0.639
agegrp, hispanic, lastmamm, density, brstproc	0.639
bmi, agegrp, density, brstproc, race	0.638
menopaus, agegrp, hispanic, density, brstproc	0.638
hrt, agegrp, lastmamm, density, brstproc	0.638
agegrp, density, brstproc, race	0.638
agegrp, surgmeno, lastmamm, density, brstproc	0.638
agegrp, lastmamm, density, brstproc, race	0.638

B. **Best combinations after expert advice**	AUC
agegrp, density, brstproc	0.641
menopaus, agegrp, density, brstproc	0.641
bmi, agegrp, density, brstproc	0.640
agegrp, hispanic, density, brstproc	0.640
agegrp, density, brstproc, agefirst	0.639
bmi, agegrp, density, brstproc, race	0.638
menopaus, agegrp, hispanic, density, brstproc	0.638
agegrp, density, brstproc, race	0.638
menopaus, agegrp, surgmeno, density, brstproc	0.638
agegrp, hispanic, density, brstproc, agefirst	0.638
bmi, agegrp, hispanic, density, brstproc	0.638
menopaus, agegrp, density, brstproc, agefirst	0.638
bmi, agegrp, density, brstproc, agefirst	0.637
menopaus, hrt, agegrp, density, brstproc	0.637
agegrp, density, brstproc, nrelbc	**0.637**

A first list of all possible combinations (from 1 to 6 attributes), is produced and sorted by performances (see Table V-A). We observe that with an AUC of 0.642, the *agegrp, density, brstproc, lastmamm* combination perform better than the two specialized regression models obtained on pre- and postmenopausal women by [8].

B. Use of expert knowledge

As stated in section IV-A, besides scoring performances, our main objectives also include readability and integration of *a priori* ideas from physicians. This step of the process involves contribution from a domain expert (see section IV-B).

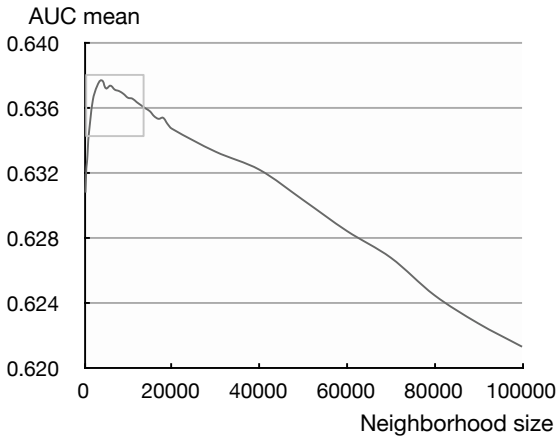


Fig. 2. Performances of top 15 combinations from Table V-B

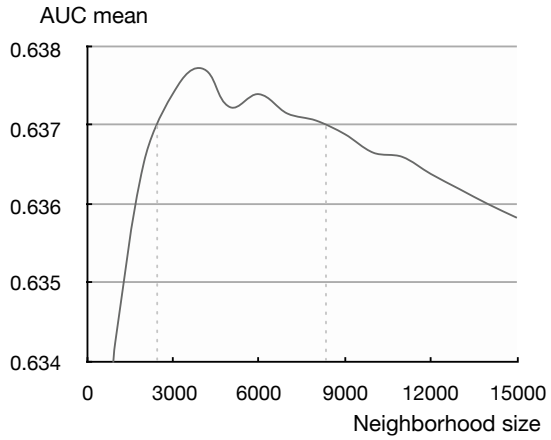


Fig. 3. Zoom on performances of top 15 combinations from Table V-B

From our domain expert point of view, with Table V-A in hand, it appears that the result of the last mammogram is a costly piece of information to obtain from women during a counseling appointment with a physician compared to performance improvement. Domain expert chooses to reduce his choices list to available combinations without *lastmamm*. Top 15 performances measures without *lastmamm* attribute are shown in Table V-B.

Based on his domain knowledge, the expert highlights that the number of first degree relatives affected by breast cancer (*nrelbc*) is widely recognized as an important factor in breast cancer risk whereas other risk factor, like the body mass index (*bmi*), are not that important compared to others. According to this expert, a good candidate for our risk score would be the *agegrp, density, brstproc, nrelbc* combination with an AUC of 0.637, which is a good performance compared with best performances of Barlow's logistic regression model (AUC of 0.624 to 0.631 depending on menopausal status). This combination uses relevant attributes for physicians according to our expert and performance loss, from 0.642 to 0.637, is acceptable.

C. Stability

In order to run a k-nearest-neighbor algorithm, the size of neighborhood has to be set. Since only k closest neighbors are used to compute the ratio healthy vs. diseased, risk score value depends on k value. If the neighborhood is too small, few breast cancer cases are included and if the neighborhood is too large, patient profiles are too different: in both cases the risk score is not reliable. For each of the 2,509 combinations of attributes, we tested the scoring function with 40 values of k from 100 to 100 000.

Using, as an example, the top 15 combinations from Table V-B, we plotted the evolution of the performance (using the AUC mean) depending on the size of the neighborhood (see Fig. 2). With an undersized neighborhood, performances are low but then, as k increases, performances increase with a maximum of 0.638. From 2,500 to 8,400 neighbors (see Fig. 3), performances are always higher than 0.637 meaning that the algorithm is relatively stable depending on k and ultimately on the number of positive examples in the neighborhood. Eventually, as k increases, performances decrease because using a larger neighborhood leads to compute a ratio with increasingly dissimilar profiles and poor targeting.

It means that performance of the combination is not obtained with a local maximum for a single value of k. It rather depicts overall prediction ability of a combination independently of the value of k as long as the size of the neighborhood is large enough to be statistically reliable (according to the law of large numbers) and stringent enough to eliminate too dissimilar profiles.

D. Discussion

As statistical risk scores do not spread in the medical community, we think there is a possibility to improve risk scores to offer both readability in its elaboration and possibility for experts to integrate their knowledge (regarding end users expectations and the disease itself) in the process. A standard methodology called *CRISP-DM* was followed in the process of building such a risk score. The database from the BCSC was selected because a regression-based score was already built upon it and because the database itself was publicly available. We chose to run extensive test with a k-nearest-neighbor algorithm to score profiles with different combinations of attributes. Every combinations with 1 to 6 attributes were tested, each for several values of k neighbors. Thus, we were able to allow experts to establish rules to keep or reject combinations by weighting between performance versus attributes usefulness and risk factors expected by physicians.

Nevertheless, our study has some limitations. First, even if we selected one of the few databases large enough to be representative of the targeted population, findings from database of volunteers require cautious extrapolation to general population. Second, if the concept of similarity

used in the algorithm is easy to understand for everyone, performances may be limited due to imbalanced data and the constraint of not modifying data used in this paper in order to be able to compare results. However, options are available to improve steps of the process. Better performances may be obtained using another algorithm, potentially with balance of data in the data preparation step, or by combining k-nearest-neighbor with another algorithm [25]. Use of expert knowledge could be improved by selecting models which are provided to the expert to avoid complications due to the size of the list of combinations. Performances could also be improved by integrating domain knowledge deeper in the algorithm: for example, introduction of relative risk as a weight in the distance computation may help to deal with the different level of influence of each risk factors.

Since k-nearest-neighbor algorithm gives good results, we think it would be useful to test this process on another database that include continuous attributes that were not discretized. For example age or breast density are one of the most predictive attributes and more specific data should improve performances. Higher risk profiles should be more accurately targeted leading to increased performances.

VI. Conclusion

On a medical dataset, we obtain good results on readability on the modeling method with a k-nearest-neighbor algorithm easy to understand for physicians and patients. In addition, the score is very easy to use for end-users with only four attributes needed. We also allow the expert to choose a combination that has not necessarily the best detection performance, but show qualities like physician acceptance and inclusion of most performant attributes recognized by the community.

Our approach is innovative and successful because we have shown that it is possible to build a simple and readable risk score model for primary breast cancer prevention that performs as good as widely used logistical models.

References

[1] "World Cancer Report," p. 512, 2008. [Online]. Available: http://www.iarc.fr/en/publications/pdfs-online/wcr/index.php

[2] F. C. Teams, "Mammographic surveillance in women younger than 50 years who have a family history of breast cancer: tumour characteristics and projected effect on mortality in the prospective, single-arm, fh01 study," *The Lancet Oncology*, vol. 11, no. 12, pp. 1127–1134, 12 2010.

[3] P. Lichtenstein, N. V. Holm, P. K. Verkasalo, A. Iliadou, J. Kaprio, M. Koskenvuo, E. Pukkala, A. Skytthe, and K. Hemminki, "Environmental and heritable factors in the causation of cancer, analyses of cohorts of twins from sweden, denmark, and finland," *New England Journal of Medicine*, vol. 343, no. 2, pp. 78–85, 07 2000.

[4] P. Testard-Vaillant, "The war on cancer," *CNRS international magazine*, vol. 17, pp. 18–21, 2010.

[5] M. H. Gail, L. A. Brinton, D. P. Byar, D. K. Corle, S. B. Green, C. Schairer, and J. J. Mulvihill, "Projecting individualized probabilities of developing breast cancer for white females who are being examined annually," *J. Natl. Cancer Inst.*, vol. 81, no. 24, pp. 1879–1886, 1989.

[6] J. Costantino, M. Gail, D. Pee, S. Anderson, C. Redmond, J. Benichou, and H. Wieand, "Validation studies for models projecting the risk of invasive and total breast cancer incidence." *J Natl Cancer Inst*, vol. 91, no. 18, pp. 1541–8, 1999.

[7] A. Decarli, S. Calza, G. Masala, C. Specchia, D. Palli, and M. H. Gail, "Gail model for prediction of absolute risk of invasive breast cancer: Independent evaluation in the florence-european prospective investigation into cancer and nutrition cohort," *J. Natl. Cancer Inst.*, vol. 98, no. 23, pp. 1686–1693, 2006.

[8] W. E. Barlow, E. White, R. Ballard-Barbash, P. M. Vacek, L. Titus-Ernstoff, P. A. Carney, J. A. Tice, D. S. M. Buist, B. M. Geller, R. Rosenberg, B. C. Yankaskas, and K. Kerlikowske, "Prospective breast cancer risk prediction model for women undergoing screening mammography," *J. Natl. Cancer Inst.*, vol. 98, no. 17, pp. 1204–1214, 2006.

[9] A. Endo, T. Shibata, and H. Tanaka, "Comparison of seven algorithms to predict breast cancer survival," *Biomedical Soft Computing and Human Sciences*, vol. 13 2, pp. 11–16, 2008.

[10] J. M. Jerez-Aragonés, J. A. Gómez-Ruiz, G. Ramos-Jiménez, J. Muñoz-Pérez, and A.-C. E., "A combined neural network and decision trees model for prognosis of breast cancer relapse," *Artificial Intelligence in Medicine*, vol. 27, pp. 45–63(19), jan 2003.

[11] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[12] S. Visa and A. Ralescu, "Issues in mining imbalanced data sets - a review paper," in *Sixteen Midwest Artificial Intelligence and Cognitive Science Conference*, 2005, pp. 67–73.

[13] G. M. Weiss and F. Provost, "Learning when training data are costly: The effect of class distribution on tree induction," *J. of Art. Int. Research*, vol. 19, pp. 315–354, 2003.

[14] P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," in *Fifth International Conference on Knowledge Discovery and Data Mining*. ACM Press, 1999, pp. 155–164.

[15] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory under-sampling for class-imbalance learning," in *International Conference on Data Mining*, 2006, pp. 965–969.

[16] A. Liu, J. Ghosh, and C. Martin, "Generative oversampling for mining imbalanced datasets," in *International Conference on Data Mining*, 2007, pp. 66–72.

[17] R. Ballard-Barbash, S. Taplin, B. Yankaskas, V. Ernster, R. Rosenberg, P. Carney, W. Barlow, B. Geller, K. Kerlikowske, B. Edwards, C. Lynch, N. Urban, C. Chrvala, C. Key, S. Poplack, J. Worden, and L. Kessler, "Breast Cancer Surveillance Consortium: a national mammography screening and outcomes database," *Am. J. Roentgenol.*, vol. 169, no. 4, pp. 1001–1008, 1997.

[18] V. Reston, Ed., *Breast Imaging Reporting and Data System Atlas (BI-RADS Atlas)*. American College of Radiology, 2003.

[19] S. F. Altekruse, C. L. Kosary, M. Krapcho, N. Neyman, R. Aminou, W. Waldron, J. Ruhl, N. Howlader, Z. Tatalovich, H. Cho, A. Mariotto, M. Eisner, D. R. Lewis, and B. K. Edwards. (2010, August) SEER Cancer Statistics Review, 1975-2007. http://seer.cancer.gov/csr/1975_2007/.

[20] P. Chapman, J. Clinton, R. Kerber, and T. Khabaza, "Crisp-dm 1.0 step-by-step data mining guide," The CRISP-DM Consortium, Tech. Rep., 2000.

[21] E. Fix and J. Hodges, "Discriminatory analysis, non-parametric discrimination: consistency properties," USAF Scholl of aviation and medicine, Randolph Field, Tech. Rep., 1951.

[22] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[23] J. P. Egan, *Signal detection theory and ROC analysis*, ser. Series in Cognition and Perception. Academic Press, 1975.

[24] T. Fawcett, "An introduction to ROC analysis," *Pattern Recogn. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.

[25] N.-K. Pham, T.-N. Do, P. Lenca, and S. Lallich, "Using local node information in decision trees: Coupling a local labeling rule with an off-centered entropy," in *International Conference on Data Mining*, 2008, pp. 117–123.

Soft-sensors for real-time monitoring and control of a black liquor concentration process

Mouloud Amazouz and Radu Platon

CanmetEnergy, Natural Resources Canada, Varennes (QC) J3X 1S6 Canada

Abstract - The control of the black liquor concentration system is an important issue in the pulp and paper manufacturing industry. A high black liquor solid content (BLSC) is desired at the outlet concentration sub-system to maximize chemicals recovery and steam production by black liquor burning while it tends to increase evaporation surfaces fouling. A good control of the concentration system is then necessary to minimize fouling while maximizing the BLSC. This important task relies on accurate continuous measurement of BLSC and the process variables that affect it. Despite the advances in instrumentation in this sector, it is still very difficult to obtain a continuous and accurate BLSC data. It is then proposed to develop soft sensors to estimate the BLSC along the concentration system. This paper presents the development and the implementation of soft-sensors in the black liquor concentration process for a pulp & paper plant. Both PLS and Neural Networks (NN) methods were used to develop accurate soft sensors. A PLS model was first developed using all available variables then a variable contribution analysis was made to reduce the number of variables. A new PLS model is then developed using the reduced set of data. A neural networks based model was also developed using the reduced dataset in order to improve the PLS model. Both models displayed satisfactory predictive performances, but the NN model clearly outperformed the PLS model, both for modeling and validation. An updating tool was also developed to automatically update the model when necessary. The NN-based model was implemented online at a plant to monitor the BLSC, detect deviations and identify the variables responsible for the deviations. The estimated BLSC value can be sent to the existing control system to improve the operation of the concentration sub-system thus reducing the BLSC variability and the energy use. The predictive model can be updated automatically when major changes in equipments or their operation occur.

1. INTRODUCTION

Some manufacturing industries face a problem of lack of real time and accurate measurements necessary for good control of product quality and process performance. This can result in process variability, unscheduled down-time and upsets. This is the case in the pulp and paper industry where the control of the BLSC is an important challenge. BLSC is usually measured by refractometers. Although these instruments provide reliable measurements, they need frequent cleaning because of solid deposits on the probs. Despite frequent cleaning operations, experience showed hat these sensors rapidly get se sensors are importance of the cleaning operation, the probes get rapidly recovered by a layer of black liquor. Samples of black liquor are taken for laboratory analysis every eight hours. When the results from the laboratory analysis show quality degradation, an action is taken to improve the situation. Important product loss could result when time delay between the beginning of a quality degradation and the obtention of the results from laboratory analysis is large. This problem could be solved by using soft sensors that can give a good estimation of the black solid content on a continuous basis. The integration of soft sensors in the control system can help improve the black liquor concentration system operation.

A soft sensor is the correlation from various raw data sources to create a new source of useful information. It is a model that infers process state and product quality variables that are difficult to measure on-line (composition, melt index, molecular distribution, etc.) from readily available process measurements (temperature, pressure, flow, etc.). The first applications of soft sensor technology began in the mid 80's and have matured during the 1990's. They can be developed using three types of modeling techniques:

1. Physical Models
2. Statistical Based Models
3. Black Box (AI) Models

Most processes in the pulp and paper industry are non-linear and too complex to be accurately described with physical models. Black box models such as neural network based models are generally more suited both for desription and prediction modeling. K. Rajesh *et al* [1] made an exhaustive review of applications of neural network to develop soft sensors to solve paper industry problems. In all cited papers in Rajesh et al review, a priori knowledge is used to select input variables prior to the neural network model development. This approach is subject to errors in case of complex processes or lack of knowledge. Using statisctical methods to first determine most influential variables prior to neural nets model development could be the solution.

Statistical methods namely PCA and PLS and their variants are widely used in industry for knowledge discovery in industrial databases and for variable estimation. Unlike black-box methods statistical techniques offer an understanding of the process, by revealing inter-relationships among the different variables [2] - [3] - [4]. The PLS method is based on a linear regression technique. The accuracy of this method will depend on the type of relationship between the variables that govern the process. PLS has the capability of dealing with missing data which makes the method very attractive in process industry. The neural network method does not

cope with missing data. Selection of input variables for PLS model development can be made based on process knowledge when the process is not complex or by using a PCA analysis using all available data. When dealing with a process containing a large number of variables, PCA can be used in order to determine the variables having the most significant influence on the process variability and on the variable to be inferred. This allows for a reduction in the number of variables, without losing significant information. The first advantage of reducing the number of inputs is the reduction of computing time and the second one is a simplified online monitoring of the soft sensor. One should however make sure that the PLS model made with a reduced number of variables is still accurate enough for the application.

Statistical methods are based on the assumption that the process is linear and in steady state. One needs to be very careful when applying these methods to study non-linear and transient processes such as pulp and paper process. It is however possible to apply some simple transformations to process data to extend the use of these methods to non-linear and transient cases. Dayal *et al.* [5] introduced lags in some variables before applying PLS to estimate the amount of fibers in the pulp produced by a digester.

When the PLS model obtained from a reduced data set after the application of lags is not accurate enough, an appropriate method can be used to improve the prediction capacity of the model because of non-linearities. Neural networks can be used in this case.

Despite the advancement of R&D work in this area, there are still very few industrial applications of this technology in the pulp and paper industry. Long term continuous use of soft sensors usually faces the problem of models maintenance and updating especially when the operators do not have the necessary knowledge and expertise to update the models. It is therefore proposed in this work to develop tools that will help make the task of model updating easier and faster especially for complex processes where the number of variables is very important. This paper deals with the development, the implementation and the continuous use of a soft sensor to predict BLSC in a pulp and paper Kraft mill.

This paper is divided into six chapters. After the introduction, the second chapter describes the black liquor concentration process and defines the different measured variables. The deficition of the soft sensor and its development are given in chapter 3. The developed models are analysed in chapter four and model selection and online implementation are explained in chapter five before the conclusion.

2. DESCRIPTION OF THE BLACK LIQUOR CONCENTRATION PROCESS

Black liquor (BL), a by-product of the papermaking process, is an important liquid fuel in the pulp and paper industry. It consists of the remaining substances after the digestive process where the cellulose fibres have been cooked out from the wood. A simple schematic description of the pulping process is shown in figure 1. The pulp and paper making process consists of producing pulp from wood chips. Wood chips and "white liquor", a mixture of sodium hydroxide (NaOH) and sodium sulfide (Na_2S), are combined and cooked in a digester. The cooking process breaks down the lignin, which is the glue that holds the wood fibres together. The spent mixture of white liquor and lignin is known as "black liquor". The black liquor is concentrated in several stages through evaporation, by heating with steam, until it consists of approximately 66% solids. The concentrated black liquor is then burned in the recovery boiler. The combustion of the black liquor produces steam for the process and a residue of molten chemicals or "smelt" consisting of sodium carbonate, sodium sulphate and sodium sulphide. The smelt is now referred to as green liquor due to its color. By adding water and lime, the green liquor is processed through the lime kiln and turned back to fresh white liquor that can be used again in the digester.

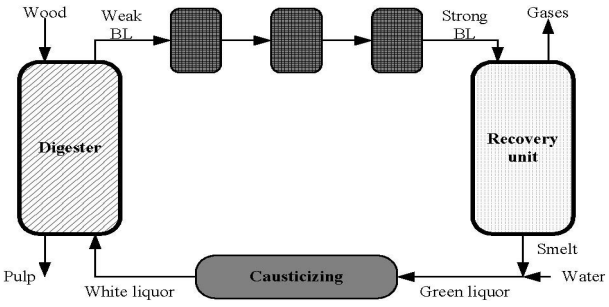


Fig. 1 Simplified diagram of the pulping process

Weak black liquor (solid content of 13%) from the digester passes through several evaporators and concentrator of falling film type where steam is brought in contact with the black liquor (fig. 2). Heat exchange from the steam and the black liquor evaporates the water contained in the black liquor. Live steam is sent to evaporator first effect and vapour produced from the evaporation of the water contained in the black lquor from each effect is directed to the following effect. Black liquor should leave the last concentrator at a concentration of 66%. The concentration system also consists of several flash tanks and a condenser to condense the vapour produced by the evaporation process. The solid content of

the black liquor depends on several variables (pressures, temperatures, flow rates, and the composition of black liquor, vapour and condensates). The final BLSC is controlled by adjusting the live steam flow rate and pressure by using a simple PID controller. One major issue in this process is the fouling in black liquor evaporators and concentrators that result from deposition of sodium salts. This occurs when the total solid concentration in the black liquor exceeds the solubility limit of the Na_2CO_3 and Na_2SO_4 that it contains. Knowing the black liquor solids content at different locations of the evaporators and concentrators with good precision can lead to fouling prevention, reduction of shutdowns due to cleaning, and an overall process stability improvement. Pulp and paper mills use refractometers and samples lab-analysis to measure the BLSC. Both methods are not ideal and can lead to non-precise values and non acceptable delays because refractometres need to be cleaned on a regular basis and lab-testing is long and it is usually made every 8 hours. When the measured solid content is not adequate, operators should act rapidly to bring back the solid content to its desirable value and sometimes they need to manually change the value of the input to the controller. When the measured solid content is too low, operators turn the controller into manual mode and open the steam valve to its maximum to try to quickly recover the process. Such action is costly in terms of energy and can lead to local overheating of the black liquor that can cause excessive deposits on evaporation surfaces. An online and accurate estimation of the BLSC can lead to variability and cost reduction and a more stable operation with limited unscheduled shutdowns. This can be achieved by using soft sensors.

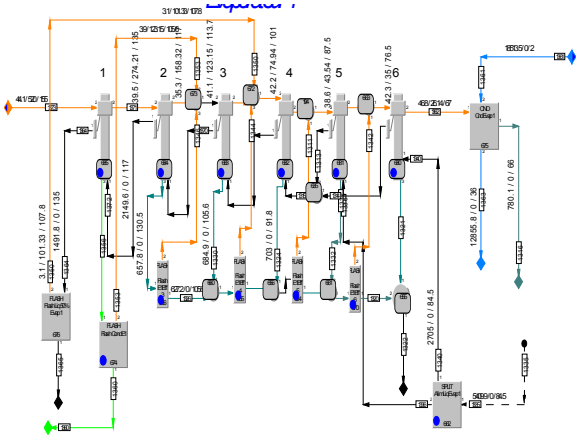


Fig. 2 Detailed view of the black liquor concentration system

3. SOFT SENSORS FOR BLSC ESTIMATION

3.1. Soft-sensor development

To accomplish this work, the authors used a PLS method to build a model using all available variables from a pulp and paper Process Information (PI) historical database to infer BLSC at different locations in the black liquor concentration sub-system. For practical reasons concerning online implementation of soft sensors, the number of variables needs to be reduced to the most important variables. A new PLS model is then developed using the reduced data set. The obtained PLS model is then compared to a neural network model. Developing a neural network using a reduced data set requires much less computing time for training. The prediction capacity of the two models was compared for online estimation of BLSC.

3.2. *PLS* model building

3.2.1 Data selection and preparation

The modelling data base contained 106 variables with 2520 observations – hourly averages of process measurements for a period of three months. The dataset is then cleaned by removing erroneous and non representative data such as drift and malfunction of measuring instruments and shutdowns. A fixed and moving average filter was then applied to remove the noise from the data. Data is then scaled between -1 and +1. A simple missing data replacement consisting of mean replacement when less that five consecutive value are missing is used.

3.2.2 PLS model development

Partial least squares (PLS) models are based on principal components of both the independent data **X** and the dependent data **Y**. The central idea is to calculate the principal component scores of the **X** and the **Y** data matrix and to set up a regression model between the scores (and not the original data).

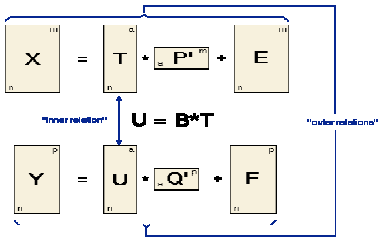


Fig. 3 PLS modelling principle

Thus the matrix **X** is decomposed into a matrix **T** (the score matrix) and a matrix **P'** (the loadings matrix) plus an error matrix **E.** The matrix **Y** is decomposed into **U** and **Q** and the error term **F**. These two equations are called outer relations. The goal of the PLS algorithm is to minimize the norm of **F** while keeping the correlation between **X** and **Y** by the inner relation **U** = **BT** .

The important point when setting up a PLS model is to make a decision for the optimum number *a* of principal components involved in the PLS model. While this can be done from variation criteria for other models, for PLS the optimum number of components has to be determined empirically by cross validation of the PLS model using an increasing number of components. Before computing the PLS models, the data was used to study the variables correlation and autocorrelation structures. This allows determining the degree of the process linearity and dynamics. Lags were evaluated and applied to specific variables to account for dynamics. New variables were also created by expanding each variable in the dataset to its respective squared and cubed version. The results indicate that there is no significant effect of nonlinearity or lagging between the input and output variables.

A PLS model for estimating the BLSC at the concentrator feed was developed using all 106 variables and 80% of the data selected randomly. The cross-validation error Q^2 of the obtained model equals 93.8% which is an indication of a good model. The predictied values versus the actual measurements of the BLSC at the concentrator feed are shown on figure 4.

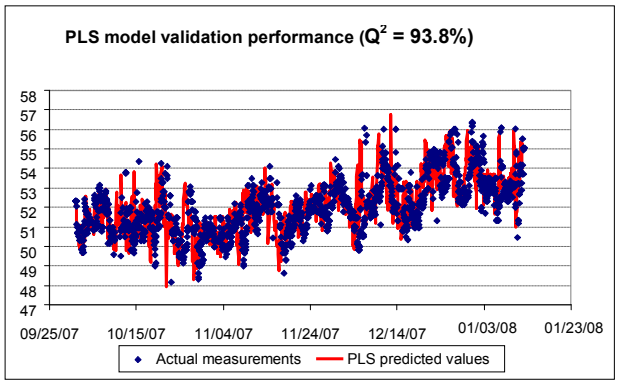


Fig. 4. Prediction accuracy of the PLS model with initial dataset

3.2.3 PLS model validation

As mentioned previously, 20% of the complete data set was left for validation. The cross-validation performance of the PLS model is equal to 87.9 (see figure 5).

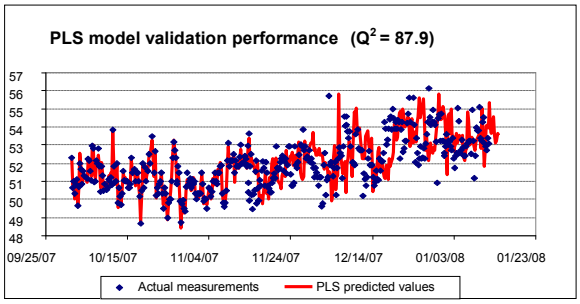


Fig. 5. Validation performance of the PLS model

3.2.4 Variables pruning

Since the soft sensor model uses on-line measurements of process variables as inputs in order to infer the desired process parameter, its performance can be seriously affected by instrument-related erroneous measurements. Using reduced number of input variables minimises the risk of soft sensor performance degradation due to instrument malfunction, but it can also result in poor predictive performance. For a soft sensor industrial application, it is important that a trade-off be achieved in terms of the number of inputs and the model's predictive performance.

A study was then performed to determine the significance of all input variables with respect to two criteria: their ability to explain the variability in the input dataset, and their importance in predicting the output variable. Finally, it was decided to select the first 20 most important variables to form a new, smaller modelling database. Using this new dataset, a reduced PLS model was computed. The modelling and validation predictive Q^2 are 85.3% and 73.9%, respectively (figures 6 and 7).

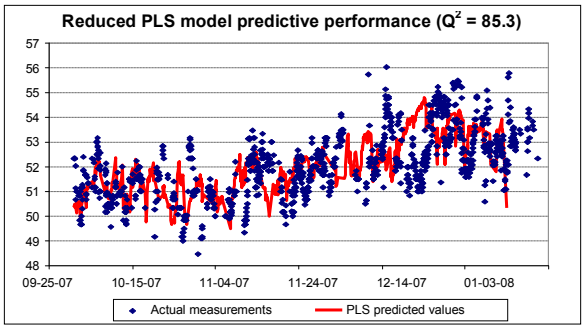


Fig. 6. Prediction accuracy of the PLS model with reduced dataset

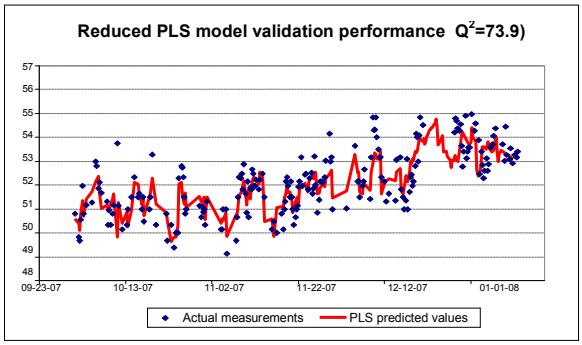


Fig. 7. Validation accuracy of the PLS model with reduced dataset

3.3.Neural network modelling

There are many types of neural network (e.g. perceptron, Kohonen nets and Hopfield nets) but the most suitable for modeling chemical engineering processes is the feedforward configuration. The same reduced dataset that was used for computing the second PLS was used for computing the neural network model. The first 20 most significant variables, as determined by the PLS analysis were used as inputs to the neural network in order to infer the BLSC. One-hidden-layer feed-forward neural network using hyperbolic tangent as activation functions was selected as the neural paradigm. NNs were configured for training using the Levenberg–Marquardt optimization algorithm together with the cross validation based early stopping mechanism to prevent over-fitting [6]. Several feed-forward NNs were investigated considering different number of hidden neurons, ranging from 5 to 15. Each network was trained at least three times using different sets, randomly generated, of initial weights. The number of neurons in the hidden layer was chosen considering the performance of the network giving the least error on the test data. The neural model selected in this work was an one-hidden-layer feed-forward neural network (Figure 8) with twenty neurons in the input layer, fourteen hyperbolic tangent sigmoid hidden neurons, and one linear neuron in the output layer, representing the estimated BLSC.

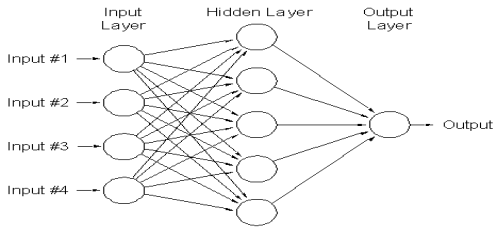


Fig. 8. A single hidden layer feed forward network

The prediction capability of the neural network model is shown in figure 9. The Q2 is equal to 89.6.

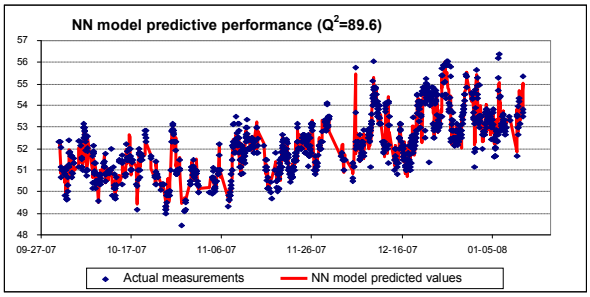


Fig. 9. Measured (blue) and NN predicted values (red) for the reduced dataset

4.4 Neural network model validation

The predictive performance of the neural network model was tested on the validation data set using 20% of the initial data set. It can be seen that the network performs well as shown in figure 10 and a Q^2 value of 75.3%.

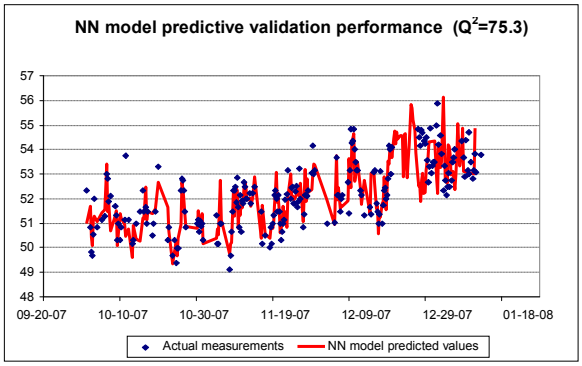


Fig. 10. Measured (blue) and neural network predicted values (red)

4. RESULTS ANALYSIS

Predictive models for the BLSC at the concentrator feed were developed using PLS and neural network techniques.

The PLS model using the complete database – 106 input variables – has the best predictive performance compared to the reduced models. When the reduced database – 20 inputs – is used, the neural network model slightly outperforms the PLS model in terms of modeling and validation (see table 2).

Using a small number of inputs reduces the risk of soft sensor performance degradation due to possible sensor faults. It is also very important to reduce the number of

variables for online application. It is much faster and easier to acquire 20 variables and compute the output for online application at a high frequency data acquisition.

Table 2. Comparison of modelling techniques performance

Model type	Number of inputs	Q^2 validation
PLS (original)	106	87.9
PLS (reduced)	20	73.9
Neural Network	20	75.3

5. SOFT SENSOR ONLINE IMPLEMENTATION AND MODEL UPDATING

The neural network is implemented online with success in a pulp&paper mill. The inputs data to the models are retrieved from a process information system and imported into an Excel sheet and the output value is computed and displayed on an Excel plot (figure 10). The most important variables (up to twenty) are also display in order to check if no variable is outside its initial range. In case this happens, an alarm will be displayed on the screen for action. When the BLSC value is out it's of normal range (defined by operator or 3σ), a warning message is displayed and the variables responsible for the fault are identified. Action can be then taken to correct the situation manually or via the control system.

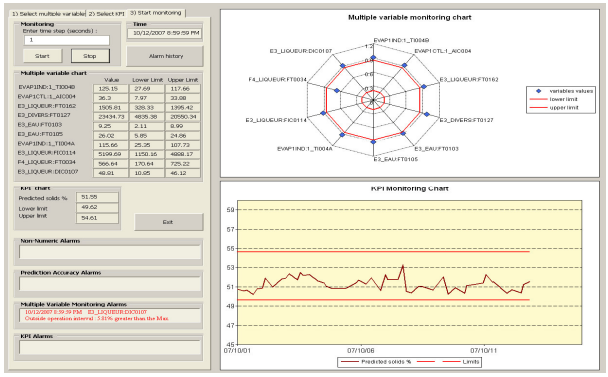


Fig.10 Screenshot of the online soft sensor implementation for BLSC monitoring

As it is stated before, the developed model is good as far as there is no major change to the process operation and equipments. When for example, equipment is changed or when there is a change to the operation procedures, the initial model should be updated in order to keep the goodness of prediction. In this work, a procedure has been added to automatically allow the process engineers and operators to update the model with new historical data corresponding to the period of change in the process.

6. CONCLUSION

PLS and neural network techniques were used to develop models for predicting the BLSC at the concentrator feed. First, a PLS model using all 174 variables present in the database were used as inputs. The PLS analysis was used to identify the variables most significant for predicting the solid content, and the first twenty variables were used as inputs to a new PLS and a neural network models for computing simplification and practical deployment of the models. Both models displayed satisfactory predictive performances, but the neural network clearly outperformed the PLS model, both for the modeling and validation data sets.

The developed neural network soft sensor performed with satisfaction since it was implemented online at the plant. Being able to inferring in real-time the value of BLSC by using most important variables can lead to fouling prevention, reduction of shutdowns due to cleaning, and an overall process stability improvement. It is expected that the online monitoring of the BLSC and its incorporation in the control system will reduce its variability by 25 to 50%.

REFERENCES

[1] K. Rajesh and A.K. Ray (2006). Artificial neural network for solving paper industry problems: A review, *Journal of Scientific & Industrial Research*, 65, pp. 565-573

[2] Zamprogna E., Barolo M. and Seboprg D. (2005). Optimal selection of soft sensor inputs for batch distillation columns using principal component analysis, *Journal of Process Control*, 15, pp. 39-52

[3] Wold S. et al. (2003). Improving pulp and paper process diagnostics and knowledge by means of multivariate analytical techniques, *Pulp and Paper Canada*, 104(5), pp. 48-50

[4] S. Marcikic. (1994). Application of Feedforward Neural Networks and Partial Least Squares Regression for Modelling Kappa Number in a Continuous Kamyr Digester. *Pulp and Paper Canada*, 95 (1), pp. 26-32.

[5] Dayal et al. Application of feed forward neural networks and partial least squares regression for modelling Kappa number in a continuous digester. Pulp and paper Canada, 95, 26-32.

[6] S. Haykin, Neural Networks: A comprehensive Foundation (2nd ed.), Prentice Hall (1999).

Acknowledgment

The authors would like to thank the Program for Energy Research and Development Program (PERD) of Canada for the financial support.

Ant Colony Optimization with Ants' Individual Memories

Yasuhiko Kato[1] **and Hiroki Inoue**[2]
[1]Department of Economics, Kumamoto Gakuen University, Kumamoto City, Kumamoto Pref., Japan
[2]Graduate School of Economics, Kumamoto Gakuen University, Kumamoto City, Kumamoto Pref., Japan

Abstract - *Since the ant colony optimization (ACO) algorithm was introduced by Dorigo in 1992, several researchers have enhanced it. An ant in the basic ACO algorithm has no information (long-term memory); it searches using only pheromone information. In this paper, we propose a variant of the ACO algorithm that uses an ant's individual memory to seek an optimum solution. Moreover, it incorporates not only the case in which an ant's individual memory is permanent but also the case in which the memory is lost with a certain probability. The proposed algorithm's effectiveness is demonstrated by testing with benchmark test problems from the TSP library (TSPLIB).*

Keywords: Ant colony optimization, Traveling Salesman Problem

1 Introduction

Recently, ant colony optimization (ACO) algorithms incorporating metaheuristics have been actively researched. Since its introduction by Dorigo in 1992, ACO has been applied to permutation-type optimization problems such as the travelling salesman problem (TSP) and scheduling problems. It is difficult to reach the optimum solution for a combinatorial optimization problem within a realistic timeframe. Therefore, various methods to arrive at an approximate solution have been researched and proposed. Results have demonstrated that the ACO algorithm has superior performance in solving the TSP. The ACO algorithm was developed after studying ant behavior. Ants share information about their foraging behavior through pheromones. Herein, we propose a new ACO algorithm incorporating pheromone matrix information but also ants' individual memories. An ant's individual memory is the excellent solution obtained during the search.

In section 2, we describe the TSP. Section 3 introduces several variants of the basic ACO algorithm. In section 4, we describe extension of the ACO using external memory. Section 5 presents the experimentally obtained results.

2 Traveling Salesman Problem

The travelling salesman problem (TSP) is a combinatorial optimization problem. Given a list of cities and their pairwise distances, the goal is identification of the shortest tour that visits each city exactly once (Fig. 1). The optimum solution can be found by calculating the lengths of all possible tours if the problem involves only a few cities. However, because of a phenomenon known as combinatorial explosion, as the number of cities increases, TSP becomes insoluble in polynomial time. The TSP is known as an NP-hard problem because the optimum solution cannot be found efficiently. Various optimization algorithms designed to seek the approximate solution of the TSP have been investigated. Genetic algorithms and simulated annealing are well-known methods for solving the TSP. The TSP is often used as a benchmark against which an algorithm for solving a combinatorial optimization problem is checked. Benchmark test problems are available to the public on the internet at TSPLIB.

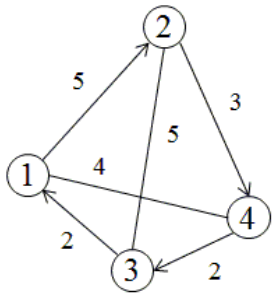


Fig. 1. Simple TSP.

Now, the solution space of the TSP is examined. It is assumed that a higher mountain is a better solution (Fig. 2). Metaheuristics concentrates on the place in which the optimum solution seems to exist in the solution space. When the search converges to local optima, local optima are misjudged to be the optimum solution if the optimum solution does not exist in the range of the search (similar to mistaking a trail to the top of a mountain). Therefore, it is important that a method not converge at local optima. Maintaining diversity is

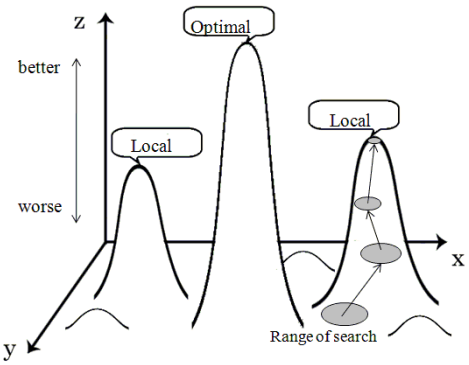


Fig. 2. Local solutions and optimal solution.

a point of improving the algorithm.

3 Ant Algorithms

3.1 Ant System (AS)

The ant algorithm is a metaheuristic that emulates the cooperative foraging behavior of ants to solve combinatorial optimization problems. In an ant algorithm, ants seek a solution based on the distance between cities and the pheromone strength. Therefore, only the distance between cities is used to select an edge in the first search because the pheromone strength at each edge is constant at that moment. When an ant passes over an edge, the pheromone strength on that edge increases. Thereby, more ants are attracted to that edge. Moreover, the pheromone on each edge decays over several iterations. The balance between 'centralization of the search area' because of pheromone deposition by the ants and 'decentralization of the search area' because evaporation of the pheromone is of utmost importance in the ant algorithm. The first ant algorithm was the ant system (AS) [1] proposed by Dorigo in 1992. It constitutes the basis of the ant algorithm for TSP. The basic AS algorithm is described as follows.

First, ant agents are placed on a randomly chosen city that serves as the starting point. An ant agent selects the next city based on the "pheromone strength τ" and "distance between cities d." When applying the ACO algorithm to the TSP, a pheromone strength τ_{ij} is associated with each edge (i,j), where τ_{ij} is numerical information that is modified during the run of the algorithm. The city in which an ant agent k is presently located is assumed to be i. The probability that city j is selected as the next city is given by the following equation (Fig. 3).

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in N^{k}}[\tau_{il}]^{\alpha}[\eta_{il}]^{\beta}} \qquad (1)$$

n that equation, N^k is the set of unvisited cities for an ant agent k; α and β are parameters that respectively determine the importance of 'pheromone strength τ' and 'neighborhood of the distance η'. The balance between the values of α and β is important. Also, η_{ij} is $1/d_{ij}$. The initial value of the pheromone strength is denoted as τ_0.

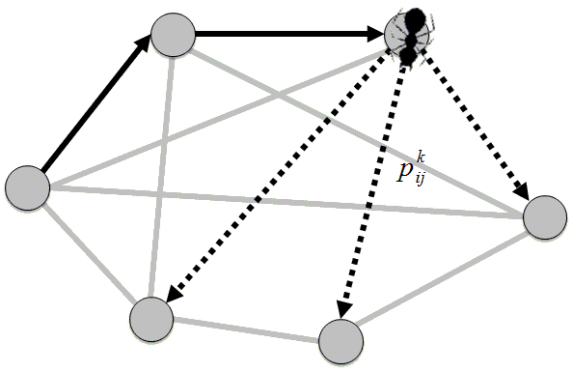


Fig. 3. Next city selection of ant system for TSP.

After all ants complete the tour, τ_{ij} is updated according to the following rules. The number of ants is m; the pheromone decay parameter is ρ in [0,1]

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \sum_{k=1}^{m}\Delta\tau_{ij}^{k} \text{ and} \qquad (2)$$

$$\Delta\tau_{ij}^{k} = \begin{cases} 1/L^{k} & \text{if } (i,j) \in T^{k} \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

where L^k is the total length of the tour covered by ant k in a tour and T^k is the edge set for the previously described tour.

The operation of the AS algorithm to solve TSP is shown by the pseudo-code in Fig. 4.

```
Initialize
For t =1 to number of iterations do
  For k =1 to m do
    Repeat until ant k has completed a tour
      Select the city j to be visited next with probability p_ij^k
      given by equation (1)
    End
    Calculate the length L^k of the tour generated by ant k
  End
  Update the pheromone τ_ij on all edges according to equation
(2)
End
```

Fig. 4. Ant system algorithm in pseudo code.

3.2 Ant Colony System (ACS)

The ant colony system (ACS) [2] proposed by Dorigo in 1996 for city selection and pheromone update are improved. The city is selected by the pseudo-random-proportional rule defined in ACS. According to this rule, the city where the selection probability is the maximum is sometimes selected as the next city. An ant positioned on city i selects the next city j by application of the rule given by the following equation.

$$j = \begin{cases} \arg\max_{l \in N^{k}}\left\{[\tau_{il}]^{\alpha}[\eta_{il}]^{\beta}\right\} & \text{if } q \le q_0 \\ J & \text{otherwise} \end{cases} \qquad (4)$$

Therein, q is a random number uniformly distributed in [0,1], q_0 is a parameter ($0 \le q_0 \le 1$), and J is a random variable selected according to the probability distribution given in (1).

The pheromone update in ACS is conducted in two stages, using the 'global update rule' and the 'local update rule'.

1) Local pheromone update rule: The local update rule is applied whenever the ant selects the next city and moves and

the pheromone is updated. Therefore, the number of times the equation is applied at iteration is given by the product of the number of ants and the number of cities. The search area is centralized by the local update rule at iteration. Moreover, the amount of increase in a pheromone in the local update rule is given by a pheromone decay parameter multiplied by the initial amount of the pheromone. The purpose of the local update rule is to bring the pheromone strength close to the initial value. Consequently, the pheromone concentration in the search area is prevented from increasing considerably.

$$\tau_{ij} \leftarrow (1-\psi)\tau_{ij} + \psi\tau_0 \qquad (5)$$

In that expression, ψ signifies the pheromone evaporation rate in the local pheromone update.

2) Global pheromone update rule: After all ants complete the tour as well as AS, then the global update rule is applied. Here, a difference from AS is that only the ant that travels the shortest tour at the iteration is allowed to deposit a pheromone. (All ants update the pheromone in AS.) The pheromone update by the global update rule differs from the local pheromone update rule. It has the effect of centralizing the search area.

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij} \qquad (6)$$

$$\Delta\tau_{ij} \begin{cases} 1/L^+ & if\ (i,j) \in T^+ \\ 0 & otherwise \end{cases} \qquad (7)$$

In those expressions, T^+ and L^+ respectively signify the tour and the tour length of the best solution at the iteration (*iteration best*).

Operation of the ACS algorithm to solve TSP is shown by the pseudo-code in Fig. 5.

```
Initialize
For t =1 to number of iterations do
  For k =1 to m do
    Repeat until ant k has completed a tour
      Select the city j to be visited next with probability p_ij^k
          given by equation (4)
      Local pheromone τ_ij update on selected edge (i,j)
          according to equation (5)
    End
    Calculate the length L^k of the tour generated by ant k
  End
  Global pheromone update τ_ij on all edges according to
equation (6)
End
```

Fig. 5. Ant colony system algorithm in pseudo code.

3.3 Ant-Q

Another algorithm proposed by Dorigo in 1995, ANT-Q [3], applies a reinforcement learning technique to the pheromone update: Q-learning. The influence of selecting a city as the next city is considered. Then the pheromone that is secreted by the ant based on the local pheromone update rule is estimated as

$$\tau_{ij} \leftarrow (1-\Phi)\tau_{ij} + \varphi\Phi \cdot \max_{l \in N^k} \tau_{il}, \qquad (8)$$

where Φ is the pheromone evaporation rate and φ is a coefficient of the pheromone update. In that expression, max τ_{il} is the maximum pheromone strength in the next unvisited city.

3.4 Max–min Ant System

The max–min Ant System (MMAS) [4] is an algorithm proposed by Stutzle in 2000. In MMAS, the upper and lower bound values are set for the pheromone strength to prevent premature search convergence. Instead of τ_0, an initial pheromone value is used as the upper bound value τ_{max}. In fact, MMAS differs from AS in the manner in which the pheromone update takes place. Only the ant that travels the shortest tour at the iteration is permitted to update the trail. The pheromone is updated according to equation (6).

The value of the pheromone strength is checked after an update, and it is adjusted using the following conditional equation.

$$\tau_{ij} \leftarrow \begin{cases} \tau_{\min} & if\ \tau_{ij} < \tau_{\min} \\ \tau_{ij} & if\ \tau_{\min} \le \tau_{ij} \le \tau_{msx} \\ \tau_{\max} & otherwise \end{cases} \qquad (9)$$

In addition to prevention of premature convergence, a smoothing mechanism was introduced into MMAS. A λ-branching factor is used to judge the convergence of the search process. Λ denote the value of the λ-branching factor and search convergence is judged using the following equations.

$$\Lambda = \frac{1}{n} \sum_{1 \le i \le n} \sum_{1 \le j \le n | j \ne i} \varepsilon_{ij} \qquad (10)$$

$$\varepsilon_{ij} = \begin{cases} 1 & if\ \tau_{ij} > \lambda\{\tau_{\max} - \tau_{\min}\} + \tau_{\min} \\ 0 & otherwise \end{cases} \qquad (11)$$

In those equations, n is the number of cities. If the value of Λ is less than a certain threshold, then the value of the pheromone strength is initialized. Premature convergence of the search is prevented by initializing the pheromone value.

3.5 Ant System with Elitist Strategy and Ranking

For AS with elitist strategy and ranking (ASrank) [5], the pheromone update rule is improved by improving AS as well as MMAS. Only the ant that travels the shortest tour and has the maximum ranking σ at the iteration is permitted the update in MMAS.

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \sigma\Delta\tau_{ij}^{best} + \sum_{\mu=1}^{\sigma-1}\Delta\tau_{ij}^{\mu} \qquad (12)$$

$$\Delta\tau_{ij}^{best} = \begin{cases} 1/L^{best} & if\ (i,j) \in T^{best} \\ 0 & otherwise \end{cases} \qquad (13)$$

$$\Delta\tau_{ij}^{\mu} = \begin{cases} (\sigma-\mu)/L^{\mu} & if\ (i,j) \in T^{\mu} \\ 0 & otherwise \end{cases} \qquad (14)$$

Therein, T^{best} and L^{best} respectively denote the tour and the tour length of the best solution through the search. The order is denoted as μ, and T^{μ} and L^{μ} respectively signify the tour and the tour length.

4 ACO with Ants' Individual Memories

4.1 An Overview of Related Research

This study is inspired by [6]. An earlier investigation maintains k excellent solutions of the search process as external memories shared across the entire colony. If a solution better than that of the k excellent solutions is found, then an existing excellent solution is replaced with the newly found solution, and the pheromone of the discarded solution is removed from the pheromone matrix. Additionally, we referred to [7], which concludes that for diversity maintenance, direct control of the pheromone matrix using the excellent solutions is better than the control by information exchange among pheromone matrices. In addition, the method of using segments of excellent past solutions for the pheromone update is described in [8], which is an implementation of the use of external memory for ACO.

4.2 ACO with Ants' Individual Memories

ACO using ants' individual memories is extended based on the AS and the MMAS algorithms. Therefore, our description is focused only on the aspects pertaining to this extension. The main aspects are (i) the ability of ants to memorize and (ii) the method of pheromone update. The problem with (i) is to determine the information that individual ants must memorize. In the proposed algorithm, an individual ant is made to memorize the optimum solution of a past search iteration. With respect to (ii), an ant updates the common pheromone matrix using its own optimum solution before edge selection. In other words, after the common pheromone is reinforced by each ant's memory, it selects an edge.

$$p_{ij}^{k} = \frac{\left[\tau_{ij} + \Delta\tau_{ij}^{k}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{l\in N^{k}}\left[\tau_{il} + \Delta\tau_{il}^{k}\right]^{\alpha}\left[\eta_{il}\right]^{\beta}} \qquad (15)$$

$$\Delta\tau_{ij}^{k} = \begin{cases} 1/L^{best^{k}} & if\ (i,j) \in T^{best^{k}} \\ 0 & otherwise \end{cases} \qquad (16)$$

Therein, $best^k$ represents the best tour traveled by ant k in the search. The method for common pheromone update is the same as that in MMAS. Moreover, we set the upper and the lower bound values for the pheromone strength.

The operation of this algorithm is shown by the pseudo-code portrayed in Fig. 6.

```
Initialize
For t =1 to number of iterations do
  For k =1 to m do
    Repeat until ant k has completed a tour
      Select the city j to be visited next with probability p_ij^k
      given by equation (15)
    End
    Calculate the length L^k of the tour generated by ant k
  End
  Update the pheromone τ_ij on all edges according to equation (6)
  Control the pheromone τ_ij on all edges according to equation (9)
End
```

Fig. 6. ACO with external memory algorithm in pseudo code.

4.3 Extended ACO with Ants' Individual Memories

In [9], the effectiveness of the centralization–decentralization balance adjustment of the solution search was verified by incorporating a random selection function in ASrank in addition to the stable fraction. A function to delete each ant's individual memory with a constant probability is added to the algorithm in this study. Therefore, an ant with different characteristics is formed. One characteristic is to select the edge only by the common pheromone matrix. Another characteristic is to add an original memory before an edge selection. Usually, ants select the next city according to equation (15). However, when the memory is deleted, the next city is selected according to equation (1). The search can be decentralized using ants of two kinds with different characteristics.

The algorithm operation is shown by the pseudo-code in Fig. 7.

```
Initialize
For t =1 to number of iterations do
  For k =1 to m do
    If ant k forgets his memory  Then  Initialize best^k
    Repeat until ant k has completed a tour
      If ant k has the memory about best^k
          Then  Select the city j to be visited next with
          probability p_ij^k given by equation (15)
      Else  Select the city j to be visited next with
                    probability p_ij^k given by equation (1)
    End
    Calculate the length L^k of the tour generated by ant k
  End
  Update the pheromone τ_ij on all edges according to equation (6)
  Control the pheromone τ_ij on all edges according to equation (9)
End
```

Fig. 7. Extended ACO with external memory algorithm in pseudo code.

5 Experimental Result and Conclusion

The benchmark test problems from TSPLIB are used for the performance comparison experiment. The parameters used in this experiment are α = 1.0, β = 2.0, ρ = 0.02, λ = 0.05, and a threshold of Λ is 1.00001. The number of ants is equal to the number of cities, $m = n$. Here, τ_{max} and τ_{min} are given, respectively, by the following equations.

$$\tau_{\max} = \frac{1}{\rho \cdot \min_{i \neq j} d_{ij}} \tag{17}$$

$$\tau_{\min} = \frac{\tau_{\max}}{2n} \tag{18}$$

In all, 25 experiments spanning 10,000 iterations are performed: *Best* represents the best solution obtained in 25 searches; *Avg*. is the average of experiments; *Std* is the standard deviation; *Error* (%) is the error rate.

$$Error = \frac{100(Avg. - opt.)}{opt.} \tag{19}$$

The refresh rate is the probability of deleting an ant's individual memory. It should be varied inversely with the difficulty of a given problem: a simple problem would warrant a high refresh rate. The results of the performed comparison experiment show that the proposed algorithm performs better than the equivalent MMAS. It generally has superior performance among ant algorithms. However, more computer memory is needed more for each ant's storage information. Future work will be undertaken to verify the effect of maintaining diversity by deleting individual ants' memories by examining the course of the search process in greater detail.

TABLE I
EIL51

eil51 (opt.=426)	Extending ACO with External Memory: Refresh rate 0.00	Refresh rate 0.01	Refresh rate 0.20	AS	MMAS
Best	**426**	**426**	**426**	430	**426**
Avg.	426.48	426.48	**426.24**	435.72	426.32
Std.	0.82	0.77	**0.52**	3.12	0.56
Error (%)	0.113	0.113	**0.056**	2.282	0.075

TABLE II
EIL76

eil76 (opt.=538)	Extending ACO with External Memory: Refresh rate 0.00	Refresh rate 0.01	Refresh rate 0.20	AS	MMAS
Best	**538**	**538**	**538**	544	**538**
Avg.	**538.12**	538.16	538.28	551.32	**538.12**
Std.	**0.33**	0.37	0.46	3.41	**0.33**
Error (%)	**0.022**	0.03	0.052	2.476	**0.022**

TABLE III
EIL101

eil101 (opt.=629)	Extending ACO with External Memory: Refresh rate 0.00	Refresh rate 0.01	Refresh rate 0.20	AS	MMAS
Best	**629**	630	**629**	668	630
Avg.	**632.88**	633.2	633.24	678.6	633.28
Std.	2.26	**1.38**	2.17	5.18	1.88
Error (%)	**0.617**	0.668	0.674	7.886	0.68

TABLE IV
KROA100

kroA100 (opt.=21282)	Extending ACO with External Memory: Refresh rate 0.00	Refresh rate 0.01	Refresh rate 0.20	AS	MMAS
Best	**21282**	**21282**	**21282**	21844	**21282**
Avg.	**21282**	21283	**21282**	22267	21283
Std.	2	2.8	**0**	127.7	3.37
Error (%)	0.002	0.003	**0**	4.629	0.005

TABLE V
D198

d198 (opt.=15780)	Extending ACO with External Memory: Refresh rate 0.00	Refresh rate 0.01	Refresh rate 0.20	AS	MMAS
Best	**15848**	15855	15945	16642	15876
Avg.	**15948**	15953	15957	16857	15943
Std.	34.01	25.53	**9.12**	91.71	27.27
Error (%)	1.062	1.097	1.119	6.826	**1.035**

6 References

[1] M. Dorigo and L. M. Gambardella, "Ant Colonies for The Traveling Salesman Problem," *BioSystems*, vol. 43, 1997, pp. 73–81.

[2] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *Evolutionary Computation*, 1997, pp. 53–63.

[3] L. M. Gambardella and M. Dorigo, "Ant-Q: A Reinforcement Learning Approach to The Traveling Salesman Problem," in *Proc. Twelfth International Conference on Machine Learning* (*ML-95*), 1995, pp. 252–260.

[4] T. Stutzle, "Max–min Ant System," *Future Generation Computer Systems*, vol. 16, no. 8, 2000, pp. 889–914.

[5] B. Bullnheimer, R. F. Hartl, and C. Strauss, "A New Rank Based Version of the Ant System: A Computational Study," *Central European Journal of Operations Research and Economics*, vol. 7, no. 1, 1999, pp. 25–38.

[6] M. Guntsch and M. Middendorf, "A Population Based Approach for ACO," *Applications of Evolutionary Computing*, vol. 2279, 2002, pp. 72–81.

[7] M. Middendorf, F. Reischle, and H. Schmeck, "Information Exchange in Multi Colony Ant Algorithms," *Parallel and Distributed Computing: Proc. 15th IPDPS Workshops*, 2000, pp. 645–652.

[8] A. Acan, "An External Memory Implementation in Ant Colony Optimization," in *Proc. Fourth International Workshop of ANTS*, 2004, pp. 73–82.

[9] Y. Nakamichi and T. Arita, "The Effects of Diversity Control Based on Random Selection in Ant Colony Optimization," *IPSJ Journal*, vol. 43, no. 9, 2002, pp. 2939–2946. (in Japanese)

[10] M. Bellmore and G.L. Nemhauser, "The Traveling Salesman Problem: A Survey," *Operations Research*, vol. 16, no. 3, 1968, pp.538 –558.

[11] G. Leguizamon and Z. Michalewicz,"A New Version of Ant System for Subset Problems," in *Proc. 1999 IEEE Congress on Evolutionary Computation*, 1999, pp. 1459 – 1464.

[12] M. Dorigo and T. Stutzle, "The Ant Colony Optimization Metaheuristics: Algorithms, Applications, and Advances," *Handbook of Metaheuristics*, 2002, pp. 251–285.

[13] M. Dorigo, G. D. Caro and L. M. Gambardella, "Ant Algorithms for Discrete Optimization," *Artificial Life*, vol. 5, no. 2, 1999, pp. 137–172.

[14] S. Gilmour and M. Dras, "Understanding the Pheromone System within Ant Colony Optimization," *Lecture Notes in Computer Science*, vol. 3809, 2005, pp. 786–789.

[15] O. Cordon, I. F. de Viana, F. Herrera and L. Moreno, "A New ACO Model Integrating Evolutionary Computation Concepts," in *Proc. Ants 2000*, 2000, pp. 22–29.

A Prediction Model for Recognition of Bad Credit Customers in Saman Bank Using Neural Networks

M. Yaghini[1], T. Zhiyan[2] , and M. Fallahi[3]
[1]School of Railway Engineering, Iran University of Science & Technology, Tehran, Iran
[2] School of Industrial Engineering, Iran University of Science & Technology, Tehran, Iran
[3] School of Railway Engineering, Iran University of Science & Technology, Tehran, Iran

Abstract - *The aim of this paper is to present a model based on feed forward neural networks to recognize bad credit customers in Saman Bank. To find an appropriate structure for the proposed neural network model, three different strategies called quick, dynamic and multiple strategies are investigated. The registered data of credit customer in Saman Bank from 2000 to 2008 year is used. To prevent models from over fitting with training data specifications, according to cross validation, we divide existing data set into three subsets called training, testing, and validation set, respectively. To evaluate the proposed model, we compare the result of three different strategies in neural networks with each other and with some common prediction methods such as decision tree and logistic regression. The results revealed that the three-layer neural network based on the back propagation learning algorithm with quick strategy has higher accuracy.*

Keywords: Banking, Saman Bank, feed forward neural networks, prediction, bad credit customers

1 Introduction

Since banking industry survival necessitates riskiness, it cannot be prevented and it only can be managed. Risk management is a professional process which its main goal is improving decision quality in all levels of economic institutions including banks, in order to increase wealth of stakeholders. Risk in a financial institution means uncertainty about expected return for assets. One of the main functions of banks is credits granting to real and legal customers. Thus, banks have to minimize probability of any inappropriate decision before granting credit to decrease risk and attract low risk customers.

Credit risk is a result of default probability or probability of loan non-repayment by borrower; this risk is the same as expected loss. Credit risk evaluation is an important topic in financial risk management and has been the major focus of financial and banking industry. Data-mining methods, especially pattern classification, using real-world historical data, are of paramount importance in building such predictive models [1], [2].

Prediction models are classified into two groups. The first group includes models for classifying new credit customers based on their credit risk. The data used for modeling generally consist of financial information and demographic information about the loan applicant. In contrast, the second type of models deals with existing customers and along with other information, payment history information is also used here [3], [4], [5].

In this study, prediction model for new credit customers and prediction of their repayment situation based on neural networks is used. The different learning strategies are used in order to gain high accuracy in neural networks.

The paper is organized as follows. In section II, related literature is reviewed. The collected data from Saman bank customers' are described in section III. In section IV, an overview of the used neural network and its weighting updating way is provided and then different strategies for finding appropriate structure for the proposed model are discussed. Section V modeling results are described, and conclusions as well as recommendations for future works are presented in section VI.

2 Literature Review

In search for credit risk prediction model with minimum limiting hypothesis, authors have suggested conditional probability models such as linear probability distribution, Logit model and Probit model [3]. Logit and Probit models are more difficult than discriminant analysis models in terms of calculation. The main problem when using these models is using a long and logical time of time series. They are under influence of econometric limitations such as shorter access period to time series of dishonored data. Then expert systems and artificial intelligence were introduced in this field. Neural networks, genetic algorithm and decision trees are among currently available methods in the field [3], [6].

The use of neural networks in business application has been increased recently. Studies indicate that neural networks are an accurate tool for credit risk assessment among others [7], [8]. Lim and Sohn [9] proposed a neural network-based behavioral scoring model, which dynamically accommodates the changes of borrowers' characteristics after the loans are made. This work suggested that the proposed model could

replace the currently used static model to minimize the loss due to bad creditors. In 2007, an overview of rule extraction techniques for support vector machine was performed to credit risk assessment [10]. This work proposed also two rule extraction techniques taken from the artificial neural networks domain.

In [11] an application of neural networks to credit risk assessment related to Italian small businesses was described. This work presented two neural network systems, one with a standard feed-forward network, whiles the other with special purpose architecture; and suggested that both neural networks can be very successful in learning and estimating the default tendency of a borrower.

In [12] a hybrid SVM-based credit scoring models were proposed to evaluate an applicant's credit score from the applicant's input features. This work used the Australian and German data sets in its implementation. The work in Abdou et al. investigated the ability of neural networks, such as probabilistic neural nets and multi-layer feed-forward nets, and conventional techniques such as, discriminant analysis, probit analysis and logistic regression, in evaluating credit risk in Egyptian banks applying credit scoring models [13]. This work concluded that neural network models gave better average correct classification rates than the other techniques.

Tsai and Wu [14] compared performance of single classifier as the baseline classifier with multiple classifiers by using neural networks based on three data sets. This work presented the ensemble classifier outperforms single classifier in a set of data. Setiono et al. used recursive algorithm for extracting classification rules from feed-forward neural networks that have been trained on credit scoring data sets [15].

In [16], a multistage neural network ensemble learning model was proposed to evaluate credit risk at the measurement level. The proposed model consisted of six stages: (1) generating different training data subsets especially for data shortage, (2) creating different neural network models with different training subsets obtained from the previous stage, (3) training the generated neural network models with different training data sets and obtaining the classification score, (4) selecting the appropriate ensemble members, (5) selecting the reliability values of the selected neural network models, and (6) fusing the selected neural network ensemble members to obtain final classification result by means of reliability measurement.

Lin [17] used a three two-stage hybrid models of logistic regression-artificial neural network to construct a financial distress warning system suitable for Taiwan's banking industry, and to provide an optimal model of credit risk for supervising authorities, analysts and practitioners in conducting risk assessment and decision making. Wang and Huang [18] used back propagation algorithm based on support vector machines for decision making about credit granting to applicants. In [19], a reassigning credit scoring model (RCSM) involving two-stages was proposed. The classification stage is constructing an ANN-based credit-scoring model, which classifies applicants with accepted (good) or rejected (bad) credits. The reassign stage is trying to reduce the Type I error by reassigning the rejected good credit applicants to the conditional accepted class by using the CBR-based classification technique.

Crook et al. compared different methods of credit customers' classification. They found that neural networks had higher accuracy in prediction [20].

In this study, neural networks of two-, three- and four-layer perceptron under error back propagation learning algorithm are used in order to predict customers' status. Three strategies, quick, dynamic and multiple, are used for finding appropriate structure for networks. Then accuracy of this network is assessed by decision tree and logistic regression methods. The best model as the proposed model is selected following comparison between results from different methods.

3 Data Understanding

Saman Eqtesad Credit Corporation was established on September 23, 1999 with a share capital of US$ 1.4mln. It opened its first branch on November 22, 1999 and managed to achieve already in its first year of activity a 5% return on equity. In 2002 Saman was the third private financial institution in post-revolutionary Iran to receive a banking license. In this context, the share capital increased to US$ 26mln.

Data set used in this study is related to real customers of Saman bank who have received loan during 2000-2008. This data set includes 20 features including Customer No., Birth date, Entrance date, Gender, Occupation, Education, Marital status, Loan No., Loan status, Repayment way, Beginning date, Final date, Interest, Wage, ISICCODE, ISICEDSC, Type of assurance, Time of contract D., Time of contract M., and Amount of loan.

It should be noted that assurance types received by bank for granting credit to customers includes common documents, declaration, movable properties in bank mortgage, participation bonds, property insurance in bank mortgage, disclaimer, check for premium, certified check, time deposit investment, certified promissory note, leasing bonds, mortgage property documents, property document of civil partnership, shares, bank guarantee, legal agreement, mandate, and credit insurance.

This data set includes 82,093 loan received by real customers. Since all features cannot be used in modeling and due to need for some change in some features, first in this step data are cleaned and new features are constructed so that data are prepared before being used in modeling.

In preliminary examination of features, it was found that since interest and wage features have similar nature and each has many missing values, these two features are integrated. Customer No. feature is omitted because of uniqueness. Results of change and integration of features can be seen in table 1.

TABLE 1
CONSTRUCTING NEW FEATURES FROM AVAILABLE FEATURES

Row	Old Feature	New Feature
1	Birth date	Age
2	Entrance date	Background
3	Entrance date	Entrance year
4	Loan No.	City
5	Loan No.	Type of loan
6	Beginning date	Beginning year
7	Final date	Final year
8	Beginning & Final date	Time if contract

Generally following changes were performed:

- Customer Age feature was extracted from customer Birth date.
- Background and Entrance year features were extracted from customer bank Entrance date feature.
- Regarding explanation mentioned for Loan No., City and Type of loan features were extracted from Loan No. feature.
- Loan contract Beginning year feature was extracted from Beginning date feature and Loan contract Final year feature was extracted from Final date feature.

In the next step, data were examined in terms of quality and validity, and various methods were used for cleaning dirty data. Manual value replacement methods were used for Age, Background and Time of contract features because of missing and noisy values. Occupation feature was omitted from data set due to high amount of missing values and lack of access to these values. Missing values of other features were replaced using C&RT decision tree algorithm.

Since the goal for this study is predicting customers' loan status, Fig 1 shows way of value distribution for this variable. As it can be seen from diagram, most loans are in settled and active status, and limited numbers of loans are in risky status of doubtful debts, outstanding, and past due.

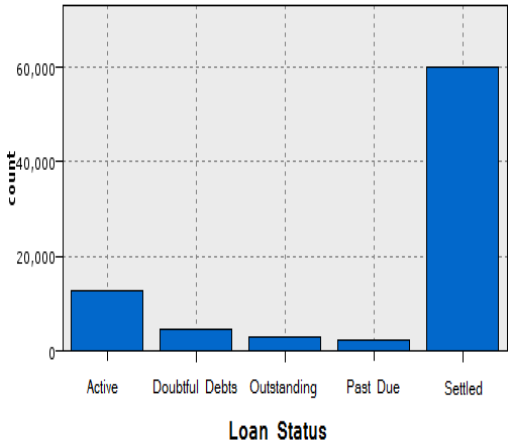


Fig. 1. Histogram diagram of loans status

Fig 2 indicates relationship between loans status feature with repayment way. The figure imply that most settled loans are related to loans with all at once and installment repayment ways. Fig 3 indicates that most settled, active, outstanding, past due and doubtful debts are related to customers entering to bank in 2004.

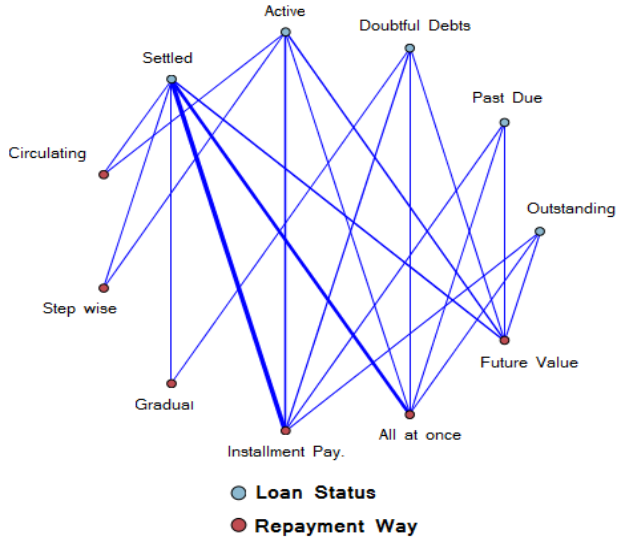


Fig. 2. Relationship between loan status feature and repayment way

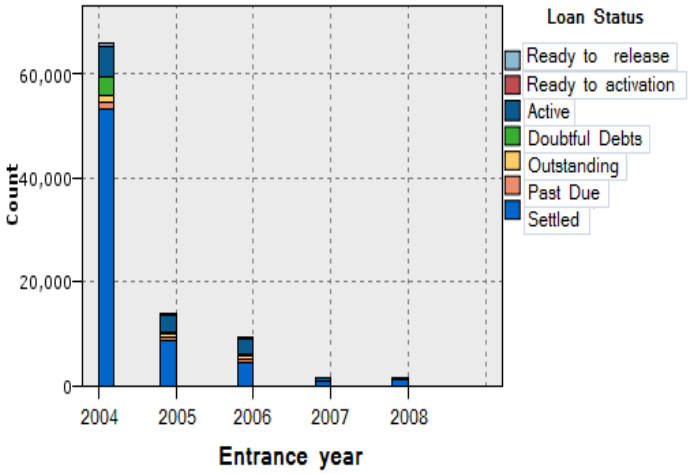


Fig. 3. Histogram diagram of entrance year variable

4 The neural network model

4.1 Neural Network Topology

Artificial neural network is regarded as a main technique in supervised learning and unsupervised learning in data mining. Neural network used in this work is feedforward neural network, which is also called multilayer perceptron. For network training, error back propagation algorithm and momentum were used. Activation function used for this network, was Sigmoid function, which is as follows:

$$\sigma(x) = 1/1 + e^{-x}, \quad (1)$$

All weights are randomly initialized in [-0.5, 0.5]. At the all weights were selected randomly in ,beginning of training

$-0.5 \le w_{ij} \le 0.5$ .distance in networkThe weights are updated as follows:

$$\Delta w_{ij}(t+1) = \eta \delta_{pj} o_{pi} + \alpha \Delta w_{ij}(t), \quad (2)$$

where, parameter η is learning rate, δ_{pj} is propagated error, o_{pi} is *i*th output neuron for sample *p*, parameter α is momentum, and $\Delta w_{ij}(t)$ is change extent in previous iteration. ∞ is assumed as constant in learning process, but η value changes during iterations of learning process. First, it is given as an initial value, and then it reduces by η_{low} logarithmically. When η is less than η_{low}, then its value is set as η_{high}. That is $\eta(t-1) < \eta_{low}$, then $\eta(t) = \eta_{high}$. Logarithmic deduction function for learning rate parameter is as follows:

$$\eta(t) = \eta(t-1).\exp(\log(\eta_{low}/\eta_{high})/d), \quad (3)$$

where, *d* is the value set by user and is called η decline. This process continues until the end of learning period. δ_{pj} error value for outer layers is calculated as follows:

$$\delta_{pj} = (t_{pj} - o_{pj}) o_{pj} (1 - o_{pj}), \quad (4)$$

for other layers, it is as follows:

$$\delta_{pj} = o_{pj} (1 - o_{pj}) \sum_{k} \delta_{pk} w_{kj}, \quad (5)$$

where, t_{pj} is expected output for prediction. Network weights are updated after each output prediction [21].

4.2 Neural Network Structure

One challenging problem in neural networks is finding optimal structure for network based on available data. To overcome this problem, various strategies were used in this work, and finally the most accurate method was selected.

1) Quick Strategy

In fast strategy, only one neural network of three-layer perceptron type is trained separately. This network has a hidden layer with $\max((\Theta_i + \Theta_o)/20, 3)$ neurons by default, where Θ_i is neuron numbers in input layer and Θ_o is the neuron numbers in output layer [21].

2) Dynamic Strategy

In dynamic strategy, network structure changes during learning process and neurons are added to network so that network efficiency increases and reaches to optimal accuracy. This stage includes two sub stages for finding appropriate structure, and then final network training. In order to find appropriate structure, first a network is constructed with two hidden layers each having two neurons. Initial learning rate is set as α=0.05 and α=0.9, and network is trained for one iteration. A copy of network is developed, one is called right side network and the other one is called left side network. Then a neuron is added to the second hidden layer of right network, and again both networks are trained for one iteration and total error is measured on both networks. If left side network has less error, it is maintained and one neuron is added to first hidden layer of right side network. If right side network has less error, left side network is replaced by a version of right side network, and again one neuron is added to second hidden layer of right side network. Both networks are trained for one iteration, and this process is repeated so that a termination condition is reached. For learning rate setting in each iteration, two vectors are calculated; first is motion vector, $M(t)$, based on weights change in one iteration, and the second one is change vector, $C(t)$, based on current iteration momentum. Vectors $C(t)$ and $M(t)$ are calculated as follows;

$$M(t) = 2[W(t) - W(t-1)], \quad (6)$$

$$C(t) = 0.8.C(t-1) - M(t), \quad (7)$$

Where, *W(t)* is current iteration weight vector and *W(t-1)* is previous iteration weight vector. magnitude ratio vector for these two vectors is defined as follows:

$$m(t) = \|M(t)\| / \|C(t)\|, \quad (8)$$

this islearning acceleration indicator. If it is less than, learning is slowing down and learning rate is multiplied by 1.2, and if it is higher than 5, learning is accelerating and learning rate is multiplied by $4/m(t)$. Once appropriate structure is found for neural network by dynamic strategy, final network should be trained based on error back propagation method. The network is trained with an initial learning rate of α=0.02 and α=0.09 [21].

3) Multiple Strategy

In this strategy, multiple networks are trained simultaneously in parallel until it is reached to termination constraint, and the network with highest accuracy is selected. It is accomplished in this way: first, multiple networks are constructed with hidden layer, with neurons in hidden layer varies between 3 to maximum number of neurons in input layer. Then multiple networks with two hidden layers are constructed per every network with one hidden layer, where the number of neurons in first hidden layer is exactly equal to number of single-layer network's. but the number of neurons in second hidden layer varies and can be 2, 5, 10, 17 and so on, we can have maximum number of neurons in first hidden layer in the second hidden layer, and it is possible to train the networks by constructing them [21].

5 Predicting Customer Repayment Status

One of main problems in training prediction models is model overfitting with current training data characteristics. This causes wrong and less accurate prediction for new values. To prevent models from overfitting with training data specifications, according to cross validation, we divide existing data set into three subsets called training, testing and validation set, respectively, regarding high amount of data. Then training was stopped periodically and network was evaluated using validating data following every training period. Thus, we are able to determine beginning of overfitting.

In present database, customer loans status include 5 statuses: settled, active, past due, outstanding, and doubtful debts. The output layer of neural network consists of 5 neurons, and each of these statuses is assigned as a neuron in output layer. The proposed model can predict these statuses.

Input layer variables in the proposed neural network include: Age, Entrance year, Gender, Educational level, Marital status, Loan type, Repayment way, Interest, ISICCODE, Time of contract, Amount, Assurance type 1-19, and Assurances sum. Concerning the number of statuses for each of these variables, input layer of neural network includes 61 neurons.

Results for comparing predictions by two-, three- and four-layer neural networks with three strategies using input and output neurons can be seen in tables 2 to 4. All model results were obtained by SPSS Clementine 12.0 software.

TABLE 2
RESULTS FOR NEURAL NETWORK MODEL

Strategy for network structure	The number of layers	Prediction accuracy in		
		validating data (%)	testing data (%)	training data (%)
Quick	Two	85.77	86.15	85.64
Quick	Three	86.33	86.71	86.36
Quick	Four	84.70	85.78	85.55
Dynamic	Four	85.47	85.98	85.97
Multiple	Four	86.20	86.46	86.16

Comparison of above tables suggests that the highest prediction accuracy is for three-layer neural network with quick strategy. Quick and multiple strategies require more memory as well as longer time for algorithm execution. The least prediction accuracy is for four-layer neural network with quick strategy. As it can be seen from results, one cannot select a specific method definitely as the best method; rather there is need for analysis on accuracy, time and required memory for each method.

The number of correct predictions in each status (output neurons) by selected neural network is given in table 3 In addition, table 4 indicates results for status prediction using four-layer neural network with fast strategy and first learning schema having least accuracy among networks.

TABLE 3
RESULTS FOR STATUS PREDICTION USING THE PROPOSED NEURAL NETWORK

Status	Status prediction				
	Active	Doubtful debts	Outstanding	Past due	settled
Active	7302	0	0	0	5291
Doubtful debts	0	3518	331	513	1
Outstanding	1	770	732	1227	1
Past due	1	495	351	1435	0
settled	2131	0	0	0	57975

TABLE 4
RESULTS FOR STATUS PREDICTION USING NEURAL NETWORK WITH LEAST ACCURACY

Status	Status prediction				
	Active	Doubtful debts	Outstanding	Past due	settled
Active	7903	0	0	0	4690
Doubtful debts	0	3099	140	1124	1
Outstanding	0	428	191	2112	1
Past due	0	30	88	22182	0
Settled	3265	0	0	0	56841

Decision tree and multinomial logistic regression techniques were used for evaluating obtained results. Chi-squared Automatic Interaction Detector (CHAID) algorithm is used for constructing decision tree. CHAID algorithm is an effective classification technique, which utilizes capability of statistic test as a measure for evaluating a possible predicting characteristic value. It determines similar values regarding target variable, and integrates them and keeps non-identical values. Then it selects the best predictor so that the first branch of tree is formed; each node is composed of a group of similar values for selected characteristic. This process continues until tree matures. Used statistic test depends on measuring level of key characteristic. If target characteristic is continuous, F test is used, if it is discrete, Chi-Square test is used [25]. table 5 shows results for prediction by desicion tree and logistic regression.

TABLE 5
RESULTS FOR PREDICTION BY DESICION TREE AND LOGISTIC REGRESSION

Algorithm	Prediction accuracy in		
	training data (%)	testing data (%)	validating data (%)
Decision tree	83.71	83.70	83.23
Decision tree	83.70	83.72	83.23
Decision tree	83.71	83.73	83.23
Logistic regression	84.95	85.33	84.67
Logistic regression	84.90	85.25	84.62
Logistic regression	85.02	85.42	84.64

Comparison of results for decision tree and logistic regression with neural networks shows that neural networks have higher accuracy in predicting customers' repayment status. It should be mentioned that there is not much difference in accuracy in methods used in this work, but they differ in processing time and model construction considerably. Finally, three-layer neural network model with quick strategy was selected as the model for this work.

6 Conclusion

In this work, a model based on neural networks with high accuracy is suggested for predicting credit customers' repayment status in Saman bank. Three networks with different layer and neuron numbers as well as three strategies were used so as to determine model optimal structure. Credit customers' data during 2000-2008 was used for prediction model construction. For evaluating the proposed model, obtained results were compared to results for decision tree and logistic regression. Results indicate that three-layer neural network with quick strategy has the highest accuracy. Using the proposed model it makes possible to have an accurate prediction of customer repayment status, and acting for appropriate planning for reducing bank credit risk. Thus, bank can minimize probability of any inappropriate decision before granting credit to decrease risk and attract low risk customers.

There are various fields for future study. In order to improve prediction accuracy, creative methods such as genetic algorithm or simulated annealing for finding optimal neural network structure can be used. Creative algorithms can be used for increasing network's learning speed. In addition, integrated models can be applied for accurate prediction of customer repayment status.

7 References

[1] L. Yu, S. Y. Wang, and K. K. Lai , " Credit risk assessment with a multistage neural network ensemble learning approach," Expert Systems with Applications 34, 2008, pp. 1434–1444.

[2] N. C. Hsieh, "An integrated data mining and behavioral scoring model for analyzing bank customer," Expert Systems with Applications 27, 2004, pp. 623-633.

[3] A. Saunders and L. Allen, *Credit Risk Measurement: New Approaches to Value at Risk and Other Paradigms*, 2nd Edition, Wiley Finance, 1999.

[4] A. Laha, "Building contextual classifiers by integrating fuzzy rule based classification technique and k-nn method for credit scoring", Advanced Engineering Informatics 21, 2007, pp. 281–291.

[5] S. T. Li, W. Shiue and M. H. Huang, "The evaluation of consumer loans using support vector machines," Expert Systems with Applications 30, 2006, pp. 772–782.

[6] R. Malhotra, D.K. Malhotra, "Differentiating between good credits and bad credits using neuro-fuzzy systems," European Journal of Operational Research 136, 2002, pp. 190-211.

[7] Y. M. Huang, C.M. Huang, and H.C. Jiau, "Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem," Nonlinear Analysis: Real world Application 7, 2006, pp. 720-747.

[8] J. H. Min and Y. C. Lee, "A practical approach to credit scoring," Expert Systems with Applications 35, 2008, pp. 1762–1770.

[9] M. K. Lim and S. Y. Sohn, (2007). Cluster-based dynamic scoring model. Expert Systems with Applications, 32(2), 427–431.

[10] D. Martens, B. Baesens, T. Van Gestel and J. Vanthienen, "Comprehensible credit scoring models using rule extraction from support vector machines," European Journal of Operational Research 183, 2007, pp. 1466–1476.

[11] E. Angelini, G. D. Tollo, and A. Roli, "A neural network approach for credit risk evaluation," the Quarterly Review of Economics and Finance 48, 2008, pp. 733-755.

[12] C. L. Huang, M. C. Chen, and C. J. Wang, "Credit scoring with a data mining approach based on support vector machines," Expert Systems with Applications 33, 2007, pp. 847–856.

[13] H. Abdou, J. Pointon, and A. EI-Masry, " Neural nets versus conventional techniques in credit scoring in Egyptian banking," Expert Systems with Applications 35, 2008, pp. 1275-1292.

[14] C. F. Tsai and J.W. Wu, "Using Neural Network ensembles for bankruptcy prediction and credit scoring," Expert Systems with Applications 34, 2008, pp. 2639-2649.

[15] R. Setiono, B. Baesens, and C. Mues, "Recursive neural network rule extraction for data with mixed attributes," IEEE Transactions on Neural Networks 19, 2008, pp. 299–307.

[16] L. Yu, Sh. Wang, and K. Lai, " Credit risk assessment with a multistage neural network ensemble learning approach," Expert Systems with Application 34, 2008, pp. 1434-1444.

[17] S. L. Lin, "A new two-stage hybrid approach of credit risk in banking industry," Expert Systems with Applications 36, 2009, pp. 8333–8341.

[18] C. M. Wang and Y. F. Huang, "Evolutionary-based feature selection approaches with new criteria for data mining: A case study of credit approval data," Expert Systems with Applications 36, 2009, pp. 5900–5908.

[19] C. L. Chuang and R. H. Lin, "Constructing a reassigning credit scoring model," Expert Systems with Applications 36, 2009, pp. 1685-1694.

[20] J. N. Crook, D. B. Edelman, and L. C. Thomas, "Recent developments in consumer credit risk assessment," European Journal of Operational Research 183, 2007, pp. 1447–1465.

[21] SPSS Inc., "Clementine® 12.0," Integral Solutions Limited, 2007.

A new Approach for Handling Numeric Ranges for Graph-Based Knowledge Discovery

Oscar E. Romero A.[1], Lawrence B. Holder[2], and Jesus A. Gonzalez B.[1]
[1]Computer Science Department, INAOE, San Andres Cholula, Puebla, Mexico
[2]The School of Electrical Engineering and Computer Science, WSU, Pullman, Washington, USA

Abstract— *Discovering interesting patterns from structural domains is an important task in many real world domains. In recent years, graph-based approaches have demonstrated to be a straight forward tool to mine structural data. However, not all graph-based knowledge discovery algorithms deal with numerical attributes in the same way. Some of the algorithms discard the numeric attributes during the preprocessing step. Some others treat them as alphanumeric values with an exact matching criterion, with the limitation to work with domains that do not have this type of attribute or discovering patterns without interesting numerical generalizations. Other algorithms work with numerical attributes with some limitations. In this work, we propose a new approach for the numerical attributes handling for graph-based learning algorithms. Our experimental results show how graph-based learning benefits from numerical values handling by increasing accuracy for the classification task and descriptive power of the patterns found (being able to process both nominal and numerical attributes). This new approach was tested with the Subdue system in the Mutagenesis and PTC (The Predictive Toxicology Challenge) domains showing an accuracy increase around 22% compared to Subdue when it does not use our numerical attributes handling method. Our results are also superior to those reported by other authors, around 7% for the Mutagenesis domain and around 17% for the PTC domain.*

Keywords: Data Mining, Graph-Based, Numerical Ranges, Knowledge Discovery.

1. Introduction

In data mining and machine learning the domain data representation determines in a great measure the quality of the results of the discovery process. Depending on the domain, the Data Mining process analyzes a data collection (such as flat files, log files, relational databases, etc.) to discover patterns, relationships, rules, associations, or useful exceptions to be used for decision making processes and for the prediction of events and/or concept discovery. Graph-based algorithms have been used for years to describe (in a natural way) flat, sequential, and structural domains with acceptable results [1], [2]. Some of these domains contain numeric attributes (attributes with continuous values). Domains containing this type of attributes are not correctly manipulated by graph-based knowledge discovery systems, although they can be appropriately represented. To the best of our knowledge there does not exist a graph based knowledge discovery algorithm that deals with continuous valued attributes in the same way that our new approach does. A solution proposed in the literature to solve this problem is the use of discretization techniques as a pre-processing or post-processing step but not at the knowledge discovery phase. However, we think that these techniques do not use all the available knowledge that can be taken advantage of during the processing phase. When graph-based knowledge discovery systems first appeared, they were not able to identify that number 2.1 was similar to 2.2, taking them as totally different values. This was the reason why those algorithms did not obtain so rich results as other algorithms that dealt with numeric attributes in a special way (such as the C4.5 classification algorithm that although it does not work with structured domains, it works with numerical data for flat domains). After some years, the number of real world structural domains containing numerical attributes increased as well as the need to have knowledge discovery systems able to deal with that type of attributes. Then, some of the available algorithms were extended in different ways in order to deal with numerical data as we describe in the Related Work Section. There are two main contributions of this work. The first one consists of the creation of a graph-based representation for mixed data types (continuous and nominal). The second one corresponds to the creation of an algorithm for the manipulation of these graphs with numerical ranges for the data mining task (both, classification and description). In this way, we can work with structural domains represented with graphs containing numeric attributes in a more effective way as we describe in the experimental Results' Section of the paper.

2. Related Work

In this Section, we describe two methods that work with structural domains (and in some way, with numerical

attributes) that were used in our experiments. The first is an Inductive Logic Programming (ILP) system: "CProgol" and the second a Graph-based system: "Subdue", which in this work, was extended with our novel method to deal with numerical attributes.

2.1 Inductive Logic Programming

ILP is a discipline that investigates the inductive construction of logic programs (first order causal theories) from examples and previous knowledge. Training examples (usually represented by atoms) can be positive (those that belong to the concept to learn) or negative (those that do not belong to the concept). The goal of an ILP system is the generation of a hypothesis that appropriately models the observations through an induction process [3]. ILP systems can be classified in different ways depending on the way in which they generate the hypothesis (i.e. top-down or bottom-up). Some examples of top-down ILP systems are MIS [4] and Foil [5]. Examples of bottom-up ILP systems are Cigol [6], Golem [7], and "CProgol" [8]. The ILP system used in this research to compare our graph-based results is "CProgol". The "CProgol" learning process is incremental. It learns a rule at a time, and it follows the one rule learning strategy. "CProgol" computes the most specific clause covering a seed example that belongs to the hypothesis language. That is, it selects an example to be generalized and finds a consistent clause covering the example. All clauses made redundant by the found clause, including all the examples covered by the new clause, are removed from the theory. The example selection and generalization cycle is repeated until all examples are covered. When constructing hypothesis clauses consistent with the examples, "CProgol" conducts a general-to-specific search in the theta-subsumption lattice of a single clause hypothesis. Inverse entailment is a procedure that generates a single, most specific clause that, together with the background knowledge, entails the observed data. The search strategy is an A*-like algorithm guided by an approximate compression measure. Each invocation of the search returns a clause, which is guaranteed to maximally compress the data. However, the set of all found hypotheses is not necessarily the most compressive set of clauses for the given examples set. "CProgol" can learn ranges of numbers and functions with numeric data (integer and floating point) by making use of the built-in predicates "is", $<$, $=<$, etc. The hypothesis language of "CProgol" is restricted by mode declarations provided by the user. The mode declarations specify the atoms to be used as head literals or body literals in hypothesis clauses. For each atom, the mode declaration indicates the argument types, and whether an argument is to be instantiated with an input variable, an output variable, or a constant. Furthermore, a mode declaration bounds the number of alternative solutions for instantiating an atom. Types are defined in the background knowledge through unary predicates, or by CProlog built-in functions. Arbitrary Prolog programs are allowed as background knowledge. Besides the background theory provided by the user, standard primitive predicates are built into "CProgol" and are available as background knowledge. "CProgol" provides a range of parameters to control the generalization process. One of these parameters specifies the maximum cardinality of hypothesis clauses. Another parameter specifies an upper bound on the nodes to be explored when searching for a consistent clause. CProgol's Search is guided to maximize the understanding of the theory with a refinement operator, which avoids redundancy. In this way, the search produces a set of high precision rules although it might not cover all the positive examples and might cover some negative examples.

3. Previous work with subdue

Given a continuous variable, there are different ways to generate numerical ranges from it. The selection of the best set of numerical ranges (obtained with a specific discretization method) is based on its capability to classify the data set with high precision. The evaluation algorithm must also have the property of finding the lower possible number of cut points that generalizes the domain without generating a cut point for each value. The problem now consists of finding the borders of the intervals and the number of intervals (or groups) for each numerical attribute. The number of possible subsets (or ranges) of values for a given attribute is exponential. "Subdue" had previously dealt with numerical attributes in three different ways (in a limited way). One of them is using a threshold parameter. A second one is applying a-match cost function. The third one is the conceptual clustering version of "Subdue". In "Subdue" we define the match type to be used for the numerical labels of the input graph. There are three match type conditions. Given two labels "l_i and "l_j, a threshold t (defined as a general parameter): **a) Exact match:** "l_i" $=$ "l_j". **b) Tolerance match:** "l_i matches "l_j iff $|l_i - l_j| < t$. **c) Difference match:** where a function is matchcost(l_i,l_j) is defined as the probability that "l_j is drawn from a probability distribution with the same mean as l_i and standard deviation defined in the input file [9]. There is another way in which "Subdue" matches instances of a substructure that differ from each other. This is done using the threshold parameter. Note that this threshold differs from the "t" threshold described in [9]. This parameter defines the fraction of the instances of a substructure (or subgraphs) to match (in terms of their number of vertices + edges) that can be different but can still be considered as a match. Here we use a cost function that only considers the difference between the size of the substructure and its instances. The function cost is

defined as matchcost(substructure, instance) size(instance) threshold. The default value of the threshold parameter is set to 0.0, which implies that graphs must match exactly. The conceptual clustering version of "Subdue" (SubdueGL) uses a post-processing phase that joins the substructures that belong to the same cluster. It is based on local values of the variable using the concept of variable discovery [10]. With this approach, SubdueGL finds numerical labels of some of the existent values for each numerical variable. These substructures are then given to "Subdue" as predefined substructures in order to find their instances. The instances of these substructures can appear in different forms throughout the database. Then, an inexact graph match can be used to identify the substructures instances. In this inexact match approach, each distortion of a graph is assigned a cost. A distortion is described in terms of basic transformations such as deletion, insertion, and substitution of vertices and edges (graph edit distance). The distortion costs can be determined by the user to bias the match to particular types of distortions [11]. This implementation allows finding substructures with small variations in their numerical labels. It groups them in a cluster and allows them to grow in a post-processing phase. This post-processing is local to each variable, and therefore, it involves a new search of the substructures. This process consumes extra time and resources (SubdueGL [10]). This happens because the substructures to be explored to find more instances with the new Variables' Representation had already been found. "Subdue" will not discover new knowledge, it will only refine those predefined substructures (clusters). The previous Subdue's Approaches deal with numerical attributes. However, it is important to choose adequate values for the $matchcost$, $threshold$, and conceptual clustering parameters. These parameters control the amount of inexactness allowed during the search of instances of a Substructure. Consequently, the quality of the patterns found is affected. The problem with discovering patterns in this way is that the user has to make a guess about the amount of threshold that will produce the best results. This is a very complex task. Our new graph-based approach to deal with numerical attributes differs from others because it considers numerical ranges during the search of the substructures (in the processing phase). We do not use initial information about the domain neither matching parameters. The numerical ranges used at each iteration are dynamic (regenerated at each iteration). This creates better substructures, more useful for the classification task as we show in the experimental Results' Section.

4. Dealing with Numerical Ranges

Discrete values have important roles in the knowledge discovery process. They present data intervals with more precise and specific representations, easier to use and understand. They are a data representation of higher level than that using continuous values. The discretization process makes the learning task faster and precise. The main point of a discretization process is to find partitions of the values of an attribute which discriminate among the different classes or groups. The groups are intervals and the evaluation of the partition is based on a commitment, few intervals and strong classes discriminate better. Clustering consists of a division of the data samples in groups based on the similarity of the objects. It is often used to group information without a label. Given a continuous variable, there are different ways to generate numerical ranges from it (different ways to discretize it). The selection of the best set of numerical ranges (obtained with a specific discretization method) is based on its capability to classify the dataset with high precision. The idea of this work is to add to the graph-based knowledge discovery system, "Subdue", the capacity to handle ranges of numbers [12]. In the following subsection, we describe the new numerical attributes handling algorithm that we created. In order to add "Subdue" this capacity, we propose a graph-based data representation, for temperature: "Temp" with a value of "4.5" and humidity: "Hum" with a value of "3.2". For this example of a flat domain, we use a star-like graph, but we will also work with structural domains. We transform each example into a star-like graph with a center vertex named example: "Exa". We then create a vertex for each of the attribute values of the example and link them to the "Exa" vertex. Those edges are labeled with the name of the attribute. We need to extend this representation to allow the use of numerical ranges, where attribute "Temp" has now a value between "3.8" and "9.5" representing the range $[3.8, 9.5]$ and attribute "Hum" has a value between "3.0" and "4.7" for the range $[3.0, 4.7]$. This new data representation is working inside of "Subdue" and is transparent to the user.

5. Handling of Numerical Ranges

In this Section, we describe the Numerical Ranges' Generation algorithm (based on frequency histograms). Our algorithm calculates distances using any of seven measures. The first distance that we use is a modification to the Tanimoto distance [13]. The second distance is a modification of the Euclidean distance [15]. The third distance is a modification to the Manhattan distance [16]. The fourth distance is a modification to the Correlation distance [17]. The fifth is a modification to the Canberra distance [14]. The sixth and seventh are two new distance measures that we propose. Figure 1 shows the pseudo-code of our algorithm [12]. The algorithm shown in figure 1, works as follows. The General function (GenerateRange) receives as parameters the

```
GenerateRange (data, N)
    Sort (data)
    for i = 1 to 7
        new histo
        SetInitial (data, N, histo)
        GenerateHistogram (histo)
        distance = TypeofDistance(i, Average(histo), Center(histo), histo)
        threshold = distance + Minimal (histo)
        TypeofGroup(i, threshold, histo)
        rangetable[i] = histo
    return rangetable
```

Fig. 1: Numerical Ranges Generation Algorithm

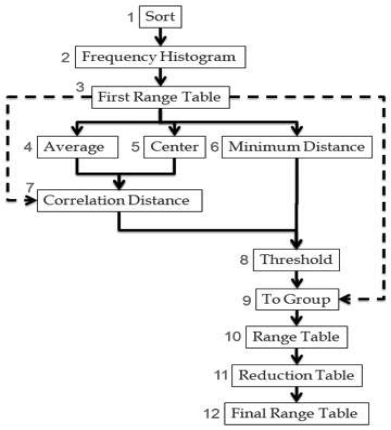


Fig. 2: Block Diagram of the Numerical Ranges Generation Algorithm

dataset and the number of examples. In the first step, and for each numerical attribute, we sort the numerical attribute in ascending way (Sort function). Next, we create a frequency histogram of the ordered data. Then, we create an initial ranges table with four fields corresponding to the center of the range, its frequency, and its low and high limits. In this initial ranges table, the center, the low and high limits contain the same value taken from the Frequencies' Histogram (GenerateHistogram Function). After that, we calculate the minimum distance between any two consecutive rows, using their center fields (Minimal function), and we calculate an average of all the center fields of the frequency histogram (Average function). After that, we calculate the centroid for the center fields of the Ranges' Table, which corresponds to the element of the frequency histogram closest to the value of the average (Center function). We now calculate a type of distance that is used to calculate the grouping threshold (to decide which values are grouped to create a numerical range). In the next step, we calculate a grouping threshold, which is the sum of the minimum distance plus the distance. After that, we iteratively group the elements of the Ranges' Table until the Ranges' Table does not suffer any modification (TypeofGroup function). At this step, we have obtained a final Ranges' Table. Figure 2 describes our algorithm through a blocks diagram [12]. The blocks diagram shown in figure 2 works as follows. **Block 1.** In this block we sort the values of the numerical attribute using the Quick Sort Algorithm. **Block 2.** In this block we calculate the frequency histogram values, grouping elements with the same value and generating a list composed of two fields, one for the element value and other for its frequency. **Block 3.** In this step we generate the ranges table. This table is composed of four fields, the center of the range, its frequency, and its low and high limits. When we initialize this table, the center and the low and high limits have the same value. The value of the frequency field is taken from the frequencies histogram. Blocks 4, 5, and 6 take as input the center field of the ranges table. **Block 4.** Calculates the average or arithmetic mean of the center field of the ranges table. **Block 5.** Calculates the centroid of all the elements of the center field of the ranges table (numerical attribute), which corresponds to the smaller value closest to the average. (i.e. if the average has the value 6.2 and is the closest value to the average 5.8, then the center field is 5.8). **Block 6.** Calculates the minimum distance between any pair of elements of the center field of the ranges table. (i.e. if element one has a value of 7.1 and element two has a value of 8.4 then the minimum distance is 1.3). **Block 7.** In this step we calculate the distance among the elements of the center field of the ranges table. This calculation depends on the distance type used, in the blocks diagram we use the correlation distance with the formula $R = \frac{\mathrm{cov}(x,y)}{Sx*Sy}$. The distance measures used in this work are modifications of the original equations that we applied to the dataset in order to generate different ranges tables with respect to the number of ranges generated and the number of elements grouped in each range. In order to calculate this distance we use the center (block 5), the average (block 4), and the values of the center field from the ranges table. **Block 8.** Calculates the grouping threshold, which is the sum of the distance type (block 7) plus the minimum distance (block 6). **Block 9.** In this step we perform an iterative grouping process that is repeated until there are no more changes in the ranges table. **Block 10.** The positive ranges table has been created (the positive ranges table), obtained from the positive examples of the attribute. **Block 11.** The final ranges table, this final ranges table (positive ranges table) is obtained from all the positive examples for this attribute, and in the same way we obtain a final negative ranges table for the negative examples

for this attribute (negative ranges table). **Block 12**. This step corresponds to the reduction process of the ranges table considering the SetCovering approach, this grouping process continues until no more changes between the ranges table occur, it is necessary to mention that we only modify the positive ranges table. This reduction is based on the five points of intersection between the ranges.

6. Structural Databases

In this work, we experimented with different data representations of numerical ranges for two structural databases. The names of the databases are Mutagenesis and PTC (The Predictive Toxicology Challenge for 2000-2001). The National Institute of Environmental Health Sciences (NIEHS) created both databases.

6.1 Carcinogenesis Domain

The PTC (Predictive Toxicology Challenge) carcinogenesis databases contain data about chemical compounds and the results of laboratory tests made on rodents in order to determine whether chemical compounds induce cancer to them or not. This database was built for a challenge to predict the result of the test using machine learning techniques. The PTC reports the carcinogenicity of several hundred chemical compounds for Male Mice (MM), Female Mice (FM), Male Rats (MR) and Female Rats (FR). According to their carcinogenicity, each of the compounds are labeled with one of the following labels: EE, IS, E, CE, SE, P, NE, N where CE, SR, and P mean that the compound is "relatively active"; NE and N mean that the compound is "relatively inactive", and EE, IS and E indicate that its carcinogenesis "cannot be decided". In order to simplify our problem, we label CE, SE, and P as positive while NE and N are taken to be negative. EE, IS, and E instances were not considered for the classification task.

6.2 Mutagenesis Domain

The mutagenesis database originally consists of 230 chemical compounds assayed for mutagenesis in Salmonella Typhimurium. From the 230 available compounds, 188 (125 positive, 63 negative) are considered to be learnable (known as regression friendly) and thus are used in the simulations. The other 42 compounds are not usually used for simulations (known as non regression friendly). These databases have been used with different classification algorithms as described in the Results' Section.

7. Results

In this Section, we present our experimental results with the Mutagenesis and PTC datasets. In the first experiment, we did not give any special treatment to any numerical attribute. In this way, we can show how "Subdue" can be enhanced when adding it the capability to deal with numeric attributes. With our second test, we show that "Subdue" can find interesting patterns containing numerical values using our approach (also improving its classification accuracy). We included our Numerical Ranges' Information to our graph-based representations and executed a 10-fold cross-validation with "Subdue".

7.1 Graph-Based Data Representation

We generated two general graph-based data representations to be used for both the mutagenesis and PTC databases. The first one considers each compound as a different example. **Compounds** are represented by ***seven attributes*** (**compound element**, **compound name**, **ind1 act** (this value is set to 1 for all compounds containing three or more fused rings), **inda** (this value is set to 1 for the five examples of acenthrylenes as they had lower than expected activity), **logmutag** (log mutagenicity), **logP** (log of the Compound's octanol/water partition coefficient hydrophobicity), and **energy of ϵ LUMO** (energy of the compounds lowest unoccupied molecular obtained from a quantum mechanical molecular model). We also consider that each compound has atoms and links between the atoms and the compound. Each **atom** is represented by ***five attributes*** (**atom element**, **atom name**, **type of atom**, **type of quanta**, and **a partial charge**). Atoms are connected by **bonds** represented by labeled edges. There exist ***eight types of bonds*** (**1=Single**, **2=Double**, **3=Triple**, **4=Aromatic**, **5=Single or Double**, **6=Single or Aromatic**, **7=Double or Aromatic**, and **8=Any**) depending on the type of connection between the atoms. The edge labels for bonds are given by a number, which is not considered to be a numeric attribute since it represents a category. The graph-based data representation for this dataset is shown in figure 3. The word "Compound" is used to indicate an example in the database. We use this as a central vertex for every example. From this compound vertex, we generate an edge to each of the atoms of the compound, which are labeled as "Element". This compound vertex has also edges and vertices that define the particular characteristics of every compound (attributes of the compound). The edge label gives the name to the attribute, and the vertex label gives it its value. Finally, every atom represented with a vertex with the word "Atom" has edges and vertices to define the particular characteristics of the atom (attributes of the atom). Atoms are connected to other atoms through bond edges, which were described above (seven different types of bonds). All the graphs shown in this report were generated with the Graphviz [18] application. This system requires every vertex to have a different name, for example, atom1, atom2, ..., atomN, but this difference does not exist in our graph-based data representation. It is used only for printing

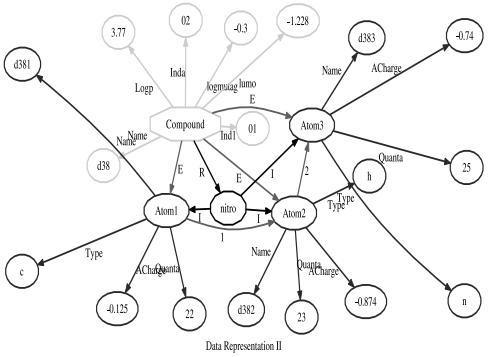


Fig. 3: Partial Graph-Based Data Representation "I".

purposes. All these vertices have the name "Atom" in our graph-based data representations. Our second graph-based data representation for the mutagenesis domain is based on the first representation. We added it information about the **aromatic rings** formed by the atoms of the compound in every example. This type of information regards to the **type of ring**. There are ***forty two*** different types of them, and to the atoms that form the ring. In order to represent this data, we add a vertex with the name of the type of ring. We connect one edge to every atom contained in the ring to this vertex.

7.2 Numerical Ranges Table Generation

The process of Numerical Ranges' Generation creates seven tables of numerical ranges for each numeric attribute. Each table corresponds to a different set of numerical ranges based on the specific type of distance to their neighbors. For this process, we used the reduction of the numerical ranges function. This function creates the numerical ranges of the positive examples in such a way that they do not cover negative examples. For some cases of numerical ranges, all the ranges after the reduction process covered negative examples, being completely eliminated and producing an empty table. In this case, we did not use this type of ranges and only considered the other types of numerical ranges (the positive ranges). We included our Numerical Ranges' Information to our graph-based representations and executed a 10-fold cross-validation with "Subdue". We can see the results for the mutagenesis domain in table 1 and for the PTC domain in table 2. Table 1 shows the results obtained for the mutagenesis domain with and without the use of numerical ranges for both data representations (with and without rings). The behavior of the results for both representations is stable. We think that by adding rings to representation "I" to create representation "II", we obtained better accuracy results and more descriptive models (with structural information about rings). The input graph of representation "II" is larger than the one created for representation "I". This means that the search space for representation "II" is larger than the one for representation "I". Then, we need to increase Subdue's Parameters in order to find a better model in terms of classification accuracy. However, we obtained a 22% increment when using numerical attributes with both data representations. This means that providing "Subdue" the capability to handle numerical ranges makes it able to find better models. Table 2 shows the results obtained with

Table 1: Accuracy Achieved for the Mutagenesis database. 10-fold-cross-validation.

Type of Graph-Based Data Representation		Regression	
		Unfriendly	Friendly
Without Rings	Without Ranges	47.23%	58.54%
	With Ranges	86.85%	85.80%
With Rings	Without Ranges	56.12%	61.54%
	With Ranges	81.36%	87.71%

Table 2: Accuracy achieved for the PTC database using a 10-fold-cross-validation.

Type of Graph-Based Data Representation		PTC			
		MM	FM	MR	FR
Without Rings	Without Ranges	66%	62%	54%	58%
	With Ranges	73%	70%	64%	70%
With Rings	Without Ranges	69%	65%	57%	61%
	With Ranges	78%	74%	72%	83%

"Subdue" (with and without ranges) for the PTC domain. In this table, we can see that on average, the classification accuracy increased 17% over the algorithms reported in the literature when we use our proposed method to handle numerical ranges for both data representations (with and without rings). This accuracy increment is not as high as we expected it to be, but as in the previous table, it is due to the execution of "Subdue" with limited parameters. We can also see in the table that the classification accuracy for all the subsets of both domains is stable. This differs from the results reported in related works. Table 3 shows the results

Table 3: Accuracy achieved for the mutagenesis and PTC databases using "CProgol" and a 10-fold-cross-validation.

PTC				Regression	
MM	FM	MR	FR	Unfriendly	Friendly
58.93%	55.73%	58.00%	59.10%	67.23%	83.50%

obtained when we executed "CProgol" with the second data representation (with rings). The data representations used in "CProgol" are equivalent to those used in our proposed method to handle numerical ranges for graph-based systems ("Subdue"). In these results, we can see that our approach obtains an increase of almost 19.5% for both the PTC and Mutagenesis databases. The results that we obtained with "CProgol" are slightly inferior (around 3% to 4%) than those reported in the citations. This might happen because the background knowledge (or the parameters setting) used in those other works could be different to those used by us. We

should also consider that "Subdue" does not use background knowledge, and that we executed "Subdue" with limited parameters. The background knowledge used by "CProgol" consists of a set of rules to describe the Rings' Structures, but we cannot define data types in "Subdue" as it is done in "CProgol". We compared the results of the previous table (the Mutagenesis and PTC databases) with the results of other authors (as shown in the Related Work Section). As we can see, the numerical ranges handling (using numerical and structural data at the same time) that we used with the graph-based data mining system "Subdue", increased Subdue's Accuracy with respect to other algorithms. Analyzing our results we can see that when we add structural data to representation "I" (which uses numerical data and some basic relations between its attributes) to obtain representation "II" (we add relations based on the Rings' Components), accuracy increases by 13% (on average) with respect to the accuracies obtained without using rings.

8. Conclusion and Future Work

There are two main contributions of this work. The first one consists of the creation of a graph-based representation for mixed data types (continuous and nominal). The second, the creation of an algorithm for the manipulation of these graphs with numerical ranges for the data mining task (classification and discovery). For our future work we will test different domains to enrich the results of our approach. We will also include temporal information with numerical values. After we have collected this data, we will be able to perform a spatial and temporal data mining process. Finally we will compare our results with "Subdue" against other algorithms that can deal with structural representations with other inductive logic programming systems and we will continue the comparison with "CProgol". This new approach was tested with the "Subdue" system in the Mutagenesis and PTC domains showing an accuracy increase around 22% compared to "Subdue" when it does not use our numerical attributes handling. Our results are also superior to those reported by other authors, around 7% for the Mutagenesis domain and around 17% for the PTC domain. The subdue model helped to distinguish models of Mutagenesis and PTC from others in a better way than "CProgol" (accuracy results are shown in the Results Section) because it is more descriptive. Finally, the substructures found with subdue are richer in structure without need to specify background knowledge to understand them.

References

[1] Jesus A. Gonzalez, Lawrence B. Holder and Diane J. Cook, Experimental comparison of graph-based relational concept learning with inductive logic programming systems, *In Lecture Notes in ArtificialIntelligence*, volume 2583, 2002, 84-99, (Springer Verlag).

[2] N. S. Ketkar, Lawrence B. Holder and Diane J. Cook, Comparison of graph-based and logic-based multi-relational data mining, *SIGKDD Explor Newsl*, 7(2), 2005, 64-71.

[3] A. Srinivasan, S. H. Muggleton, M. J. E. Sternberg, and R. D. King, Theories for mutagenicity: a study in first-order and feature-based induction *Artificial Intelligence*, volumen 85, 1996, 277-299.

[4] E. Shapiro, Inductive inference of theories from facts, *Computational Logic: Essays in Honor of Alan Robinson*, 1991, 199-255, (Publisher MIT).

[5] J. R. Quinlan, Determinate literals in inductive logic programming, *In IJCAI'91: Proceedings of the 12th international joint conference on Artificial intelligence*, 1991, 746-750, (San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.).

[6] S. Muggleton and W. Buntine, Machine invention of first-order predicates by inverting resolution, *In Proceedings of the 5th International Conference on Machine Learning*, 1988, 339-352, (Ann Arbor, Michgan, USA: CA: Morgan Kaufmann).

[7] S. Muggleton, W. Building and P. Road, Inverse entailment and progol, *New generation Computing*, volume 13, number 3, 245-286, 1995.

[8] S. Muggleton and J. Firth, Cprogol4.4: a tutorial introduction, *In Inductive Logic Programming and Knowledge Discovery in Databases*, 2001, 160-188, (Editorial Springer-Verlag).

[9] A. Baritchi, Diane J. Cook and Lawrence B. Holder, Discovering structural patterns in telecommunications data, *In Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference*, 2000, 82-85, (AAAI Press).

[10] I. Jonyer, Lawrence B. Holder and Diane J. Cook, Concept formation using graph grammars, *In Proceedings of the KDD Workshop on Multi-Relational Data Mining*, volume 2, 2002, 19-43, (Cambridge, MA, USA: MIT Press).

[11] I. Jonyer, Lawrence B. Holder and Diane J. Cook, Graphbased hierarchical conceptual clustering, *International Journal on Artificial Intelligence Tools*, 2001, 10(1-2), 107-135.

[12] Oscar E. Romero A., Jesus A. Gonzalez and Lawrence B. Holder Handling of numeric ranges for graph-based knowledge discovery, *In FLAIRS Conference*, 2010.

[13] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, (Morgan Kaufmann Publishers, 2nd ed edition, Series in Data Management Systems, 2006, pages 533).

[14] I. H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes and S. Cunningham, Weka: Practical machine learning tools and techniques with java implementations, *In International Workshop: Emerging Knowledge Engineering and Connectionist-Based Info*, 1999, 192-196, (Morgan Kaufmann Publisher).

[15] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Series in Data Management Systems*, (Second Edition, Morgan Kaufmann Series in Data Management Systems, Paperback, 2005, pages 385).

[16] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, From data mining to knowledge discovery: An overview, *In Advances in Knowledge Discovery and Data Mining*, 1996, 1-34, (AAAI Press / The MIT Press).

[17] S. Shekhar, P. Zhang, Y. Huang and R. R. Vatsavai, Chapter 3 Trends in Spatial Data Mining, Data Mining, in Editor AAAI/MIT Press (Ed.), *Next Generation Challenges and Future Directions*, (Department of Computer Sciencie and Engineering, University of Minnesota: AAAI/MIT Press, 2003), pages 24.

[18] E. R. Gansner, E. Koutsofios, S. C. North and K. phong Vo, A technique for drawing directed graphs, *IEEE Transactions on Software Engineering*, volumen 19, 1993, 214-230.

A Framework for Detecting Vulnerable, Cascaded Fuzzy Cycles in the Carbon Chain

James P. Buckley and Jennifer M. Seitzer
Computer Science Department
University of Dayton
300 College Park
Dayton, Ohio 45469-2160

Abstract - Traditional data mining algorithms identify associations in data that are not explicit. Cycle mining algorithms identify *meta-patterns* of these associations depicting inferences forming chains of positive and negative rule dependencies in our knowledge bases. The carbon chain is the process through which carbon is cycled through the air, ground, plants, animals, and fossil fuels. We identify cycles in collected data that contribute to one or more of the steps of the carbon chain. We name these contributing cycles *cascading cycles*. In this work, we show that if we can enable or disable any of these cascading cycles, it will have an effect on the overall carbon chain. Moreover, in some cases there are contributing cycles that have not yet manifested themselves. For instance, there may be a cascaded cycle that has one vertex with too low a value to make the cycle complete. We describe a methodology to detect this nearly complete, cascaded cycle.

Key words: data mining, cycle mining, climate change, fuzzy logic, knowledge-based systems

1. Introduction

This paper initially defines what traditional data mining is and the concepts of cycles, cycle mining, cascaded cycles, and fuzzy α-*cycles* and β-*cycles.* Then we discuss the carbon cycle, why it is important, and a formal description of what a cascaded cycle is. The paper then describes the fuzzy cycle paradigm, the concept of Ω-nodes, and defines a fuzzy mining algorithm. We include steps to reinforces or diminish fuzzy cycles

1.1 Traditional Data Mining

With the present state of technology, we have the capability to store extremely large amounts of data in organized and automated systems. The preponderance of data warehouses and datamarts [3], [5] are concrete evidence that this is not only possible, but of great interest to researchers, government agencies, and large corporations. But what is the meaning and usefulness of these large repositories of data?

The classical definition tells us that "Data mining is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data [4]". We are no longer looking for tabular answers or aggregations of the data; rather we are looking for *patterns* within the data that reveal knowledge previously unknown. One of the most common applications of data mining is to generate all significant association rules between items in a data set. We can employ an efficient algorithm to mine a large collection of basket data type transactions for association rules between sets of items with some minimum specified confidence [1], thus providing both meaning and usefulness to the data.

1.2 Why Meta-Patterns are Important

The patterns we discover in our data sets through data mining may not be necessarily isolated. There may be chains of rules forming patterns of patterns, or *meta-patterns*, where the head of one rule is the body of another rule. In particular, the chain of rules may form a *cycle*. This form of meta-pattern is the focus of our work in this paper.

Traditional data mining algorithms identify associations in data that are not explicit. Cycle mining algorithms identify *meta-patterns* of these associations depicting inferences forming chains of positive and negative rule dependencies. This paper examines those events that are cyclic in nature and contribute to or detract from the carbon chain. The carbon chain is the process through which carbon is cycled through the air, ground, plants, animals, and fossil fuels. It is generally believed that the carbon chain creates an environment whereby climate change will occur over time.

We can identify cycles present in our data that contribute to one or more of the events of the carbon chain. We

define these cycles as *cascading cycles* because they are not independent, and directly affect the behavior of the overarching cycle. By modifying one participating rule of a cascading cycle, we can enable or disable it. This in turn, for this particular application, will have an effect on the overall carbon chain cycle. In other words, the effect is *cascaded* to the primary (carbon) chain. In particular, we identify cascaded cycles as those having one or more vertices that are shared by the overarching cycle. Thus, changes made to the ancillary cycle will "cascade" down to the main cycle. Our work here centers on applying this formalism to the carbon chain.

In some cases there are cycles that have not yet manifested themselves. For instance, there may be a cascaded cycle that has one vertex with too low a value to make the cycle whole. This paper describes an algorithm to detect this nearly complete, cascaded cycle.

If the cycle is incomplete, a formal paradigm for cycle +mining these partial cycles using fuzzy techniques is defined. In order to differentiate between those cycles that we want to perpetuate and those that we want to break, this research will use the α-*cycle* and β-*cycle* as the underlying formalism of the paradigm. An α-*cycle* is a cycle that is good and should be perpetuated. A β-*cycle* is a cycle that has negative consequences and should be broken. Specifically, α-*cycles*, desirable cycles, should be reinforced such that complete positive cycles are created, and β-*cycles* can be weakened to keep a negative incomplete cycle from forming.

2. The Carbon Chain

Carbon is an element that is part of the ocean, air, rocks, soil and all living things. Carbon is always on the move. Organic chemicals are made from carbon more than any other atom, so the Carbon Chain is a very important one.

In the atmosphere, carbon is attached to oxygen in a gas called carbon dioxide (CO2). With the help of the Sun, through the process of photosynthesis, carbon dioxide is pulled from the air to make plant food from carbon. Through food chains, the carbon that is in plants moves to the animals that eat them. Animals that eat other animals get the carbon from their food too. When plants and animals die, their bodies, wood and leaves decay bringing the carbon into the ground. Some becomes buried miles underground and will become fossil fuels in millions and millions of years.

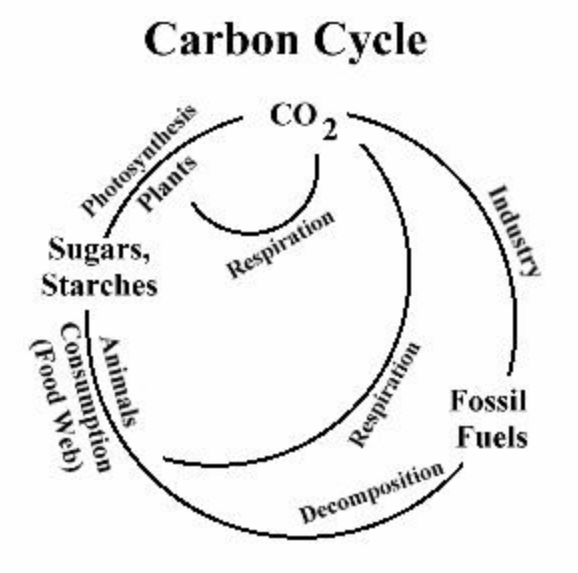


Figure 1 The Carbon Cycle

Each time you exhale, you are releasing carbon dioxide gas (CO2) into the atmosphere. Animals and plants get rid of carbon dioxide gas through a process called respiration. When humans burn fossil fuels to power factories, power plants, cars and trucks, most of the carbon quickly enters the atmosphere as carbon dioxide gas. Of the huge amount of carbon that is released from fuels, 3.3 billion tons enters the atmosphere and most of the rest becomes dissolved in seawater. The oceans, and other bodies of water, soak up some carbon from the atmosphere. [6]

3. Cascading Cycles

Of primary interest in this paper are both complete and partial cascading cycles that are discovered in our data. We are particularly interested in cascading cycles related to the carbon chain. But first, we define our term.

Intuitively, two cycles are *cascading* if they share at least one vertex.

Definition 3.1 (cascading cycles)
Let cycle X= <x_1, ..., x_k, x_1> and let cycle Y =<y_1, ..., y_l, y_1>. Cycles X and Y are cascading *if $x_i = y_j$ for some vertex x_i in X and some vertex y_j in Y.*

Example 3.1 (cascading cycles)
Let X = <a1, b1, c1,d1, a1> and Y = < h1, i1, c1, h1>. Y is a *cascading* cycle of X because it has a node in common with X (c1). (Likewise, X is a cascading cycle of Y.) Figure 2 illustrates this situation.

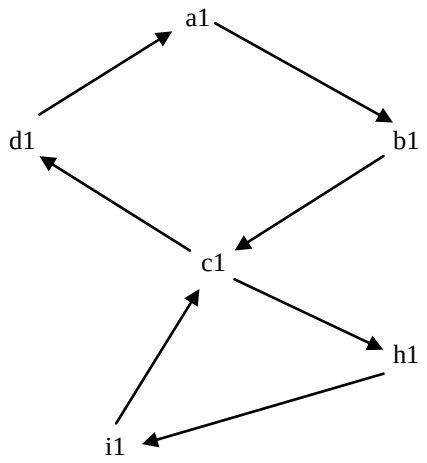


Figure 2 Cascading Cycle

Imagine for a moment that we have another cycle W that is a cascading cycle of Y. If we were to break cycle W, then cycle Y would be broken or at least weakened thereby also breaking or weakening cycle X. If we were to strengthen cycle W, this would have a similar effect on cycles Y and X. Therefore, changes to cascading cycles will have a direct and linear impact on the associated cycles "downstream".

4. Fuzzy Cycle Paradigm

In previous work [2], the authors presented a methodology that uses the individual rule supports and confidences to detect and categorize different types of cycles, as well as presenting an enhanced cycle detection algorithm that uses a metric τ computed from constituent rule support and confidence factors. This metric is used to characterize the strength of the encompassing cycle. The definition of support and confidence was used and presented in [7]..

We define τ, the average of the above two metrics, to be our threshold measurement for any specific rule. No rule with τ less than a user-specified threshold *U* will be considered meaningful enough to be placed in the system knowledge base. Hence, it will not be detected as part of any cycle.

Definition 4.1 (τ)

τ = (support + confidence) / 2

Cycles are composed of *n* individual rules, so we define the strength metric τ applied to cycles as

$T = \min(\tau_1, \ldots, \tau_n)$, where τ_i is the strength measurement of rule *i.*

The revised methodology and cycle detection algorithm do not consider rules with τ less than user-specified threshold *U*. Thus, any detected cycle has *T* at least U.

4.1 α-cycles and β-cycles

Our Cycle Mining algorithm is able to detect all cycles in a particular data set that meet a specific confidence and support threshold. However, cycles differ to the extent that they meet external system goals or semantic domain criteria. In previous work, the authors defined the concepts of a partial cycle, α-cycles, β-cycles, and partial α-cycles and β-cycles.

There is no automated method to classify cycles as desirable (α-cycles) or undesirable (β-cycles). These assessments are made externally by humans of the enterprise owning the computer system. The assessments are useful to the enterprise because they indicate whether the cycle should be perpetuated as in the case of α-cycles, or broken as in the case of β-cycles.

An example of an α-cycle is as follows:

lower electric car price→ more affordable
more affordable → consumer interest increases
consumer interest increases → sell more cars
sell more cars → lower electric car price

An example of a β-cycle is as follows:

higher electric car price→ less affordable
less affordable → consumer interest decreases
consumer interest decreases → sell fewer cars
sell fewer cars → higher electric car price

4.2 Partial Cascading Cycles

Of primary interest in this paper are partial cascading cycles of the carbon chain, because, in the case of partial cascading α-cycles, they can be useful in completing a partial or near cycle that will have a linear effect on the overall carbon chain. We always wish to reinforce or complete any cascading α-cycle or partial cascading α-cycle. Partial cascading β-cycles can be a warning of a potentially negative effect on the carbon chain if completed. To handle these detections, we wish to diminish or break any cascading β-cycle or partial cascading β-cycle. In previous work, the authors present an enhanced cycle detection algorithm. However, this algorithm gives no indication that these chains of dependencies that are almost cyclical exist. We now present a formalism that facilitates identification and handling of partial α and β-cycles.

4.3 Identifying Ω Nodes

Once we have identified partial cascading α-cycles and partial cascading β-cycles, we want to strengthen or weaken them. In order to accomplish this, we need to strengthen or weaken one or more of the constituent nodes. This involves the identification of a node or nodes that will effect a change in a cycle.

We define two types of nodes: static nodes and Ω -nodes. *Static nodes* are those nodes that can not normally be changed. For instance, a person's age or their Social Security Number is an example of a static node. *Ω-nodes* are nodes that can be readily changed to some degree. CO_2 emissions and respiration level are examples of Ω -nodes.

Associated with each Ω-node is an alterability factor, Ψ, that expresses how "changeable" a particular node is. Trivially, for static nodes, Ψ would have a value of 0. The remaining nodes would have a Ψ value such that $0 < \Psi \leq 1$.

For example, we may have the following nodes in our cycle:

Node	Ψ
Age	0
Decomposition Level	.7
SSN	0
CO_2 Emissions	.9
Received_Recycle_Mailing	1.0

Age and SSN are static nodes, whereas Decomposition Level, CO_2 Emissions, and Received_Recycle_Mailing are alterable to some degree Ψ.

4.4 Fuzzy Mining Algorithm

Fuzzy logic allows us to discover information in datasets that is not crisp [8]. The following algorithm produces a stratification of all cycles existent in a given knowledge base. By repeatedly calling the cycle detection algorithm, we find all types of cycles in the program dependency hypergraph. This algorithm uses a sequence of thresholds to identify differing strength cycles such as those defined above: complete cycles as well as near, mid, and weak cycles. Because the stratification of cycles desired by the user could contain an arbitrary number of strata, we pass in a sequence *S* of threshold values as a parameter. For each threshold U_i in the sequence, all cycles with threshold T such that $U_{I-1} \leq T \leq U_i$ will be identified. Thus, Algorithm 4.1 returns *m* classes of fuzzy cycles where *m* is the number of strata, or equivalently, the number of thresholds provided in sequence *S*.

Algorithm 4.1 (Fuzzy Mining of Cycles)
Input: -Hypergraph representation ***P,***
- threshold sequence $S = <U_1, \ldots, U_m>$
Output: -Sequence of *m* classes $<P_1, \ldots, P_m>$ of cycles where for each cycle $C \in P_i$, $U_{i-1} \leq T_c \leq U_i$

BEGIN ALGORITHM 4.1
For each threshold $U_i \in S$ {form cycle class P_i)
  For each node *h* in program hypergraph **P**
    For each incoming neighbor *b* of node *h*
      $P' :=$ *Cycle Mining(***P**, $h \leftarrow b, U_i$)
      Let T be *T* returned by *Cycle Mining*
        If T $> U_i$ then
          discard cycle *P'*
        else
          $P_i = P_i \cup P'$
Return $<P_1, \ldots, P_m>$
END ALGORITHM 4.1

To produce the above mentioned fuzzy cycle sets including complete, near, mid, and weak cycles, the following call to Algorithm 3.1 could be made.

*sets_of_cycles = Fuzzy Mining(***P**, <.85, .8, .75, .7>).

4.5 Reinforcing Cascading α-cycles and Diminishing Cascading β-cycles

It is important to note that any cycles we discover using our algorithm for a given data set are static. Only a change to the data itself can cause a cycle to be modified. The enterprise associated with the data set must incorporate change. This will produce different data which in turn may alter the cycles already discovered. In the case of cascaded cycles, a change to one edge of the cycle may cause the entire cycle to be strengthened or weakened. This makes it all the more important to identify cascading cycles as they allow for a linear effect in inter-related cycles.

α-cycles and β-cycles are cycles we want to perpetuate or remove respectively. In order to accomplish this we do the following:

For (each new data set)

execute algorithm 4.1 (to enumerate all complete cycles)
for(each cycle)
*examine cycle (to determine the set of α-cycles and the set of β-cycles) **
for (each α-cycle)

examine the individual nodes that comprise the cycle and determine all Ω nodes.
rank the nodes in terms of Ψ (measure of what to modify first)
*enterprise makes changes (if deemed necessary)**

for (each β-cycle)
examine the individual nodes that comprise the cycle and determine all Ω nodes.
rank the nodes in terms of Ψ (measure of what to modify first)
*enterprise make changes (if deemed necessary) **
a new data set is generated

*** indicates non-automated step*

Figure 3 illustrates the fuzzy cycle paradigm. It should be noted that the last step in this paradigm feeds back to the first step. This is due to the fact that knowledge produces decisions which may change the data set.

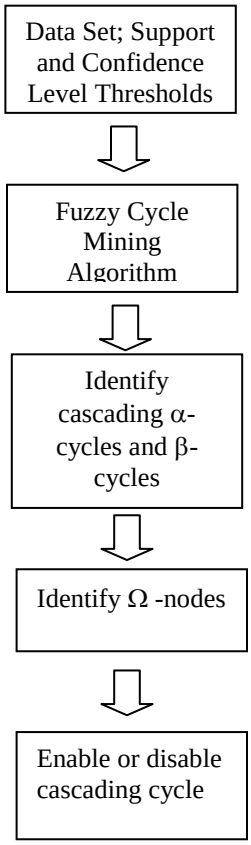


Figure 3 Fuzzy Cycle Paradigm

5. Conclusion

In this paper we defined and described cycle mining algorithms depicting inferences forming chains of positive and negative rule dependencies. The carbon chain is discussed and we defined the concept of *cascading cycles*. We show that if we can enable or disable one of these cascading cycles, it will have an effect on the overall carbon chain. In some cases there are cycles that have not yet manifested themselves.

We described a methodology to detect this nearly complete, cascaded cycle. Pertinent to this discussion, we define what cascaded α-cycles and cascaded β-cycles are and what their impact is on the entire process. In order to make any change to the cycle, we must change one or more constituent nodes in the cycle. Ω-nodes are those that can be changed by an alterability factor, Ψ.

6. Future Work

In future work, we are considering new algorithms to mine cycles using an exhaustive traversal tree. We also will examine certain properties that hold on a particular cycle and show that it also holds for all sub-cycles.

Additionally, a complete software system that models the carbon chain environment will be developed and implemented. We would also like to design an automated way to detect cascaded cycles.

7. References

[1] Agrawal, R., Imielinski, T., and Swami, A., "Mining association rules between sets of items in large databases," *SIGMOD Bulletin*, May 1993, pp. 207-216.

[2] Buckley, J. P., Seitzer, J., M., ,A Paradigm for Detecting Cycles in Large Data Sets via Fuzzy Mining, *Proceedings of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop*, Chicago, Illinois, November 8, 1999. Pages 49-55.

[3] Chaudhuri, S. and Dayal, U., "An overview of data warehousing and OLAP technology," *SIGMOD Record*, Vol.26, Num. 1, 1997, pp. 65-74.

[4] Frawley and Piatetsky-Shapiro, editors, *Knowledge Discovery in Databases*, chapter Knowledge Discovery in Databases: An Overview, AAAI Press/The MIT Press. 1991.

[5] Kimball, R., *The Data Warehouse Toolkit*, John Wiley and Sons, 1996.

[6] Lawrence, E. Henderson's Dictionary of Biological Terms, 14th Ed., Benjamin Cummings, 2008.

[7] Seitzer, J., Buckley, J. P., Monge, A., "Meta-Pattern Extraction: Mining Cycles", *Proceedings of the Florida Artificial Intelligence Research Society International Conference (FLAIRS-99)*, Orlando, FL; pp. 466-470.

[8] Zadeh, I. A. "Fuzzy Sets," *Information And Control*, Vol 8, 1965, pp. 338-353.

Reliability Analysis of Markov Blanket Learning Algorithms (1996-2010)

Shunkai Fu, Michel Desmarais, Weibin Chen, *Member*, *IEEE*

***Abstract*— In this paper, we focus on the reliability, or data efficiency, problem of the existing Markov blanket learning algorithms. We first define as well as demonstrate the seriousness of this problem. Secondly, we review eleven published algorithms ranging from 1996 to 2010, including their design and the reliability problem. Then, we discuss the challenge to improve this performance for different strategy as taken in existing works. Finally, we point out IPC-MB [1, 2] is the one with best data efficiency among published works, but it is still only sub-optimal. This work can be a reference for future effort aiming at a more reliable solution, though we conclude here that the best solution may not exist if an optimal output is expected.**

I. Introduction

THE problem of identifying essential variables is critical to the success of decision support systems and knowledge discovery tools due to the impact of the number of variables on the speed of computation, the quality of decisions, operational costs, and understandability and user acceptance of the decision model [3]. For example, in medical diagnosis, the elimination of redundant tests may reduce the risks to patients and lower the health care cost.

A principle solution to the feature reduction problem is to determine a subset of features that can render of the rest of whole features independent of the variable of interest [4]. In 1996, Koller and Sahami first showed that the Markov blanket (MB) of a given target variable T, denoted as MB_T, is the theoretically optimal set of features to predict T's value [4], although Markov blanket itself is not a new concept and can be traced back to the work of Pearl [5] on Bayesian network in 1988. Based on the findings that the full knowledge of MB_T is enough to determine the probability distribution of T and that the values of all other variables become superfluous, the induction of MB_T actually is a procedure of feature selection.

Koller and Sahami proposed the first known, but approximate, algorithm for the induction of MB_T in [4], called KS for shorthand. Since 1996, there are many following effort to improve the efficiency of search, including GSMB [6] and its many variants (IAMB and more [7-10]), MMPC/MB , HITON-PC/MB [11], PCMB[12], IPC-MB [2, 13](or BFMB in its earlier publication [14]) and MBOR [15]. As categorized by Fu in [16], these works actually belong to two groups. One group, denoted as **GROUP I**, relies on no underlying graph topology information, and it contains KS, GS, IAMB and its variants. The remaining ones belong to a different group, called **GROUP II**, because, given faithfulness assumption (its definition is ignored due to the wide availability, e.g. in [5]), they make use of the topology information to realize a more "smart" search, dividing the learning into the induction of parents and children of T first, and then the spouses of T.

General review and discussion of Markov blanket algorithms can be found in [16, 17], which are published in 2008 and 2010 respectively. In this paper, we focus on the discussion of reliability, or data efficiency, problem of algorithms for Markov blanket learning. Besides, we include two additional algorithms not found in [16, 17], MBOR[15] and λ-IAMB[10], and they belong to **GROUP II** and **I** respectively. Though reliability was noticed and studied by the authors of existing works, this is the first comprehensive in-depth study of this topic, over so many algorithms. It is expected to a useful reference for researchers to determine whether to revise existing works to get improvement, or to explore for brand new but more promising direction.

The remaining text is organized as below. In Section 2, the definition and seriousness of this problem are presented. In Section 3, we review algorithms of **GROUP I**, analysis their reliability performance, and discuss possible enhancement chance. Then, in Section 4, similar discussion is conducted on **GROUP II** algorithms. We conclude in Section 5.

II. Definition and Status of Reliability Problem

Insufficient data presents a lot of problems when working with statistical inference techniques like the independence test employed in all existing algorithms for the induction of Markov blanket. However, problems with high dimensionality and data insufficiency are so common in applications, especially in medical and biological research where collecting samples is costly. Therefore, reliability is an important standard we have to consider in the evaluation of one algorithm.

There is no direct measure about reliability, and normally, we claim that algorithm *A* is more reliable than algorithm *B* if *A* performances better, e.g. higher accuracy or short distance, when it is given the same amount of observations as *B*. This can be viewed as an informal definition. Since when

This work was supported by the Quanzhou Science and Technology Bureau of China under Grant 2009G5.

S.-K. Fu is with the Computer Science and Technology School, Huaqiao University, Xiamen, China (e-mail: shunkai.fu@gmail.com).

M.C.Desmarais is with Computer Engineering Department, Ecole Polytechnique de Montreal, Montreal, Canada (e-mail: Michel.desmarais@polymtl.ca).

W.-B. Chen is with the Computer Science and Technology School, Huaqiao University, Xiamen, China (e-mail:chwb@hqu.edu.cn).

algorithm A is more reliable than B, it means that A makes better usage of the data, reliability is also refereed as data efficiency by some [1, 16], which corresponds another widely referred measure - time efficiency. Both reliability and data efficiency concept will be used in this text, and they refer to the same concept.

Of the eleven algorithms to review in next sections, their reliability may vary greatly. For example, in **Error! Reference source not found.**, given Alarm problem, the precision, recall and distance measure by IAMB, PCMB and IPC-MB, three typical progresses in this field, are listed, given different sample size. *Precision* is the number of true positives in the output divided by the number of nodes in the output. *Recall* is the number of true positives in the output divided by the number of true positives in the Markov blanket. *Distance* is defined as the distance to perfect performance, and it is a combination of precision and recall, i.e. $\sqrt{(1-\text{precision})^2+(1-\text{recall})^2}$. From **Error! Reference source not found.**, it is observed that with the same sample size, IPC-MB and PCMB performs much better than IAMB. Besides, with the amount of observations increases, the accuracy of IPC-MB and PCMB increases quickly, much faster than IAMB.

TABLE I
ACCURACY COMPARISON AMONG IAMB, PCMB AND IPC-MB GIVEN ALARM PROBLEM. REGARDING PRECISION, RECALL AND DISTANCE, WE LIST THE MEAN AND THE STANDARD DEVIATION (BY 10-FOLDER EXPERIMENTS).

Instances	Algorithm	Precision	Recall	Distance
250	IAMB	.50±10	.43±06	.80±10
	PCMB	.66±10	.68±06	.53±08
	IPC-MB	.67±10	.67±06	.53±08
500	IAMB	.57±03	.55±02	.67±04
	PCMB	.86±03	.78±04	.31±05
	IPC-MB	.85±02	.77±04	.32±04
1000	IAMB	.57±02	.60±02	.64±02
	PCMB	.93±02	.84±02	.20±03
	IPC-MB	.94±02	.84±02	.19±03
2000	IAMB	.52±03	.58±01	.67±02
	PCMB	.97±03	.89±03	.13±04
	IPC-MB	.98±02	.90±03	.11±04
3000	IAMB	.52±03	.58±02	.68±03
	PCMB	.97±01	.92±03	.10±04
	IPC-MB	.99±01	.93±02	.07±03
4000	IAMB	.51±03	.59±02	.68±03
	PCMB	.97±02	.94±03	.07±04
	IPC-MB	.99±01	.95±01	.06±03
5000	IAMB	.49±02	.58±02	.70±03
	PCMB	.98±01	.96±03	.06±03
	IPC-MB	.99±01	.95±01	.05±02

III. REVIEW AND ANALYSIS OF GROUP I ALGORITHMS

In this section, we review and analysis the reliability of algorithms of GROUP I; we also discuss the challenge to further improve the reliability of this group of algorithms.

A. KS

Koller and Sahami's work towards optimal feature selection is the original one to recognize that the Markov blanket of T is theoretically the optimal set of features to predict its value [4]. Along this finding, they also proposed a theoretically justified framework, but computationally intractable, for optimal feature selection based on using cross-entropy to minimize the amount of predictive information lost during feature (backward) elimination. They implement a heuristic approach (called KS algorithm) to do an efficient search, however, it doesn't guarantee to produce correct outcome. KS algorithm requires two parameters: (1) the number of variables to retain, and (2) the maximum number of variables the algorithm is allowed to condition on. These two limits are helpful to reduce the search complexity greatly, but neither of them can be known in advance.

KS's reliability is influenced by the second parameter as mentioned in the last paragraph. By restricting the maximum size of conditioning set to a pre-defined value, KS may maintain the accuracy of probability computation, so as the cross-entropy estimates they depend on. In practice, we can increase/decrease this parameter based on the amount of instances available. However, this gain is exchanged with possible loss of precision theoretically since some not belonging to the MB_T may fail to be eliminated.

B. GSMB and Its Derivatives

GSMB (Grow-Shrink Markov Blanket) algorithm was proposed by Margaritis and Thrun in 2000[6], and it is the first proved published algorithm for the induction of MB_T. It depends on the fact that for any $X \in MB_T$, X is dependent of T conditioned on $MB_T - \{X\}$, denoted as $dep(X, T|MB_T - \{X\})$. GSMB (see Fig. 1) conducts the search in two phases, growing first and then shrinking, which explains its name. GSMB is so simple that its time complexity is very impressive, requiring only $O(n)$ conditional tests theoretically, where n is the number of features. However, its simplicity is paid with poor reliability in many problems. In its growing phase, $X \notin MB_T$ may be added, which will result with cascading error. At the end of the growing phase, the candidate Markov blanket set may contain partial true MB_T and some $X \notin MB_T$. Then, in the shrinking phase, those belonging to MB_T but not selected in the growing phase have no chance to be recognized and added. In addition, we may fail to remove incorrect members due to (1) the non-reliable probability computation caused by large conditioning set, and/or (2) the existence of false positives. Consequently, GSMB may have low precision as well as low recall. Even though, GSMB is still widely cited since its proved local search for MB_T guides and influences many following works.

IAMB inherits the mechanism of GSMB by dividing the search into forward selection (corresponds to growing in GSMB) and backward elimination (corresponds to shrinking in GSMB) [8] (see Fig. 2). One difference is that it selects the one most dependent on T in the growing step, instead of adding the first such variable in GSMB. This update is useful for IAMB to add the most likely candidate, making more efficient use of the observations to improve the potential

accuracy. However, IAMB's actual performance is still very poor, especially when the target $|MB_T|$ is not tiny. For example, when $|MB_T| = 8$, then the minimum conditioning set size in the growing phase will be at least 8. Whenever an incorrect variable is added, IAMB may face the similar status as in GSMB by the end of its growing step. Considering that they have exactly the same second step, IAMB's reliability is still very low.

This problem was noticed by the authors of IAMB, and three upgraded versions were proposed, i.e. InterIAMB, IAMBnPC and InterIAMBnPC [8]. Compared with IAMB, they try to interleave the growing-shrinking to remove incorrect members at an early time, or they apply PC algorithm [18] during the shrinking phase, or both revising effort being applied together. All these three algorithms perform more reliably than IAMB [8], but their complexity increases greatly as relative to GSMB and IAMB.

GS(T: Target, D: Dataset, ε: Significance Value)
{
1. $\mathbf{MB}_T^C \leftarrow \{\}$;
   //Growing phase
2. **repeat**
3. $stillGrow \leftarrow$ **false**;
4. **for**$(\forall X \in \mathbf{U} \backslash \mathbf{MB}_T^C \backslash \{T\})$ **do**
5. **If** $(I_D(T, X|\mathbf{MB}_T^C) \leq \varepsilon)$ **then**
6. $\mathbf{MB}_T^C \leftarrow \mathbf{MB}_T^C \cup \{X\}$;
7. $stillGrow \leftarrow$ **true**;
8. **end if**
9. **end for**
10. **until** $stillGrow =$ **false**
    //Shrinking phase
11. **repeat**
12. $stillShrink \leftarrow$ **false**;
13. **for**$(\forall X \in \mathbf{MB}_T^C)$ **do**
14. **If** $(I_D(T, X|\mathbf{MB}_T^C - \{X\}) > \varepsilon)$ **then**
15. $\mathbf{MB}_T^C \leftarrow \mathbf{MB}_T^C \backslash \{X\}$;
16. $stillShrink \leftarrow$ **true**;
17. **end if**
18. **end for**
19. **until** $stillShrink =$ **fasle**
20. **return** $\mathbf{MB}_T^C$;
}

Fig. 1.GSMB algorithm.

Fast-IAMB [9] was proposed more recently in 2005, and it is actually a revising version of InterIAMB. Two heuristics are proposed on the basis of InterIAMB to reduce the time complexity. One is that the more statistically appropriate significance of a G^2 conditional statistical test is used, rather than the raw conditional information value. The second is that one or more than one candidate feature will be selected in the growing phase, and the authors explained that it will save the number of statistical testing. Fast-IAMB may be faster than IAMB, but it still does not solve the most problem of IAMB, i.e. reliability. Actually, poorer reliability is expected on Fast-IAMB over IAMB because, by adding more candidates in the growing phase, the statistical tests conducted in the shrinking phase may be larger than that in the shrinking phase in IAMB.

IAMB(T: Target, D: Dataset, ε: Significance Value)
{
1. $\mathbf{MB}_T^C = \{\}$;
   //Growing phase
2. **repeat**
3. $stillGrow \leftarrow$ **false**;
4. $Y \leftarrow argmax_{X \in (\mathbf{U} \backslash \mathbf{MB}_T^C)} I_D(T, X|\mathbf{MB}_T^C)$;
5. **If** $(I_D(T, Y|\mathbf{MB}_T^C) \leq \varepsilon)$ **then**
6. $\mathbf{MB}_T^C \leftarrow \mathbf{MB}_T^C \cup \{X\}$;
7. $stillGrow \leftarrow$ **true**;
8. **end if**
9. **until** $stillGrow =$ **false**
   //Shrinking phase
10. **repeat**
11. $stillShrink \leftarrow$ **false**;
12. **for**$(\forall X \in \mathbf{MB}_T^C)$ **do**
13. **If** $(I_D(T, X|\mathbf{MB}_T^C - \{X\}) > \varepsilon)$ **then**
14. $\mathbf{MB}_T^C \leftarrow \mathbf{MB}_T^C \backslash \{X\}$;
15. $stillShrink \leftarrow$ **true**;
16. **end if**
17. **end for**
18. **until** $stillShrink =$ **fasle**
19. **return** $\mathbf{MB}_T^C$;
}

Fig. 2.IAMB algorithm.

In our review, λ-IAMB [10] is the last one derived directly from IAMB. Its authors also recognized the possible cascading error upon introducing a false positive in the growing phase, and they proposed a two-aspect improving strategy to partially resolve the problem existed in IAMB. One is using entropy instead of conditional mutual information to measure the conditional independence between two variables to reduce the actual computational complexity, and the other is to refine the growing phase. No obvious gain on reliability is expected from their proposal over IAMB.

C. Discussion and Proposal

Algorithms of **GROUP I** are simple and easy to prove, understand and implement; they are faster than those of

GROUP II. However, none of them are expected to be data efficient. Authors of this branch recognized this problem, and the most choice made is to interleave the growing-shrinking to eliminate false positive(s) in time possibly. It indeed allows us to have some gain on data efficiency, but it doesn't change the fundamental problem of this group, i.e. both growing and shrinking of this search strategy may suffer from the non-reliable decision:

- With the candidate Markov blanket set, denoted as MB_T^C (Fig. 1), continues to grow, the conditioning set of the statistical test becomes larger and larger, and the test result becomes less reliable. Besides, all $X \in \mathbf{U}\backslash MB_T^C\backslash\{T\}$ will be tested with the same (possibly large) conditioning set, MB_T^C, the risk of selecting a wrong candidate is not low; and
- In the shrinking phase, with continuously increasing MB_T^C, the test of each $X \in MB_T^C$ and T conditioned on $MB_T^C\backslash\{X\}$ may face the same problem as in the growing phase. Not only false positive may failed to be eliminated, but true one may be removed incorrectly.

Therefore, the risk of making incorrect decision exists in both steps of the algorithms of **GROUP I**, and it easily brings cascading error, i.e. introducing more false one(s) in the growing step but failing to remove them. Both precision and recall performance of them then can be quite low, as shown in **Error! Reference source not found.**.

There is no simple way to ensure that we can always select the true Markov blanket member in the growing phase, then effort should be spent on the shrinking phase, trying to remove those which should be removed. InterIAMBnPC is one such trial by restricting the conditioning set of test as small as possible. Another possible enhancement is to extend InterIAMBnPC by adding another shrinking phase after the current shrinking step. This additional shrinking phase is applied to remove some $X \notin MB_T$ from the remaining $\mathbf{U}\backslash MB_T^C$ by the updated MB_T^C, which helps to diminish the "noisy" variables for the next growing processing.

IV. Review and Analysis of GROUP II Algorithms

In this section, we review and analysis the reliability of algorithms of **GROUP II**; we also discuss the challenge to further improve the reliability of this group of algorithms. The earliest work of **GROUP II** appeared in 2003, and the most difference from **GROUP I** is that topology information is considered by algorithms of **GROUP II**.

A. MMPC/MB and HITON-PC/MB

MMPC/MB, in fact, are two algorithms proposed in 2003 [7], with MMPC (Max-Min Parents and Children) for the induction of parents and children of T, and MMMB(Max-Min Markov Blanket) for the induction of Markov blanket of T. For the first time, given faithfulness assumption, MMPC/MB divides the induction of MB_T into the search of the target's parents and children, denoted as PC_T (by MMPC) and its spouse, denoted as Sp_T, separately, which enables it to avoid conditioning on the whole MB_T as in previous works. HITON-PC/MB [11] conducts a similar search as MMPC/MB, hence it achieves improvement on reliability as well. This usage of the underlying topology information is novel, and it is viewed as a promising direction to solve the known data efficiency problem met on algorithms in **GROUP I**. However, neither MMPC/MB nor HITON-PC/MB is sound as shown in [12], so they are just mentioned briefly.

B. PCMB

PCMB [12] also proceeds in a divide-and-conquer manner as in MMPC/MB and HITON-PC/MB. It is the first proved algorithm in **GROUP II**. In the induction of candidate parents and children, PCMB learns lesson from IAMB and InterIAMBnPC by not only interleaving the growing and shrinking, but adding one additional shrinking step at the beginning of each search round, which just corresponds to our enhancement proposal for **GROUP I** algorithms by the end of Section 3. This additional shrinking phase can help to remove false positives from the remaining non-tested future set, reducing the search complexity as well as decreasing the risk of adding false one by conditioning small set. This novel shrinking-growing-shrinking strategy is effective to improve the reliability of this algorithm, and actually, the increase is very impressive [1, 12, 13].

C. IPC-MB

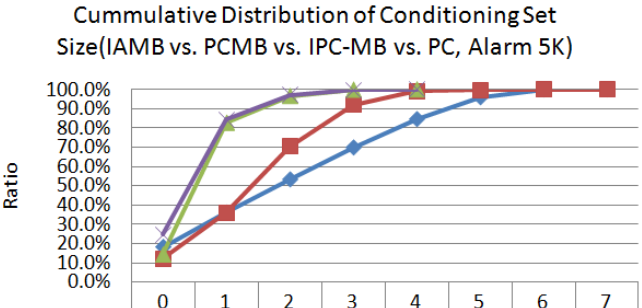
IPC-MB was first proposed in [14], with revised discussion and proof in [1, 2]. It takes the similar framework as proposed in MMPC/MB by inducing PC_T and Sp_T separately. However, its search of PC_T (by *RecognizePC* in [1]) is completely different from MMPC/MB, HITON-PC/MB and PCMB. All the previous three ones actually conduct an interleaving growing-shrinking search, and it is not time and data efficient since they have to compute all the conditional dependency of candidate variables with T, sort and select the most possible one. While in IPC-MB, it proceeds by pure shrinking, removing those independent with T (see Fig. 3). Besides, due that this shrinking starts with empty conditioning set, increasing with one every time when there is nothing more can be eliminated with smaller conditioning set, it demonstrates quite impressive reliability performance as compared with all previous work covered. This simple backward elimination, surprisingly, has similar effect to the shrinking-growing-shrinking as done in PCMB, and as proved in [1, 13], IPC-MB guarantees to produce correct results. However, it saves great time cost relative to PCMB, requiring up to 95% fewer of conditional independence tests [1].

```
RecognizePC(T: target, ADJ_T : Adjacency set to search, D: dataset, ε : threshold)
{
1.  NonPC = φ;
2.  cutSetSize = 0;
3.  do
4.    for( each X ∈ ADJ_T)  do
5.      for(each S ⊆ ADJ_T \{X}  with  |S|= cutSetSize) do
6.        if( I_D(T, X | S) ≤ ε ) then
7.          NonPC = NonPC ∪{X};
8.          Sepset_{T,X} = S;        //Cache for later reference
9.          break;
10.       end if
11.     end for
12.   end for
13.   if(|NonPC| > 0 ) then
14.     ADJ_T = ADJ \ NonPC ;
15.     cutSetSize = cutSetSize + 1;
16.     NonPC = φ;
17.   else
18.     break;      // No more to do, but exit;
19.   end if
20. while(| ADJ_T |> cutSetSize)
21. return ADJ_T ;
}
```

Fig. 3.*RecognizePC*, the procedure used by IPC-MB to induce the candidate parents and children candidate set.

D. MBOR

Reducing the search space by removing obvious non-Markov blanket variables is followed by the author of MBOR [15] as well. Compared with PCMB, MBOR takes an independent pre-processing step to shrink the problem domain **U**, getting a so-called Markov blanket superset which actually is composed by parents-and-children superset and spouse superset.



	0	1	2	3	4	5	6	7
IAMB	18.3%	36.1%	53.3%	69.9%	84.6%	95.9%	99.9%	100.0
PCMB	12.3%	36.0%	70.6%	92.0%	99.1%	99.8%	100.0	100.0
IPC-MB	14.6%	82.7%	96.5%	99.9%	100.0			
PC	24.6%	84.4%	97.3%	99.8%	100.0			

Fig. 4. This figure is just the Figure 4-20 in [1], and it is copied here for quick reference. It is about the distribution of different conditioning set size happening in IAMB, PCMB and IPC-MB in the induction of Markov blanket, given Alarm network.

In the induction of parents-and-children superset, only conditioning sets of size zero and one are used, which ensures reliability. In the preparation of spouse superset, the maximum conditioning set may be of size two. By doing so, the author expects to reduce the search space with minimum cost. Actually, given faithfulness assumption, this is effective since most real network is not dense, and many false positives may be eliminated given conditioning set of size zero, one and two (see Fig. 4).

E. Discussion and Proposal

Recognizing that the Markov blanket is composed of parents, children and spouses of the target, and taking a divide-and-conquer mechanism in the search is the most advantage of **GROUP II** algorithms over **GROUP I**. From lessons learned from previous work, extra attention is paid during each step to maintain the conditioning set as small as possible. Among PCMB, IPC-MB and MBOR, forward selection can be found, in the search of PCMB and MBOR, as well as backward elimination; however, in IPC-MB, only backward elimination is found by conducting pure shrinking. Considering that most real problems may have a sparse network, the strategy taken by IPC-MB seems be reasonable. As shown in the example of Fig. 4, more than 95% of statistical tests done in IPC-MB requiring conditioning set of size two or smaller, which indicates that most false negatives may be eliminated with small conditioning set. Regarding PCMB, the corresponding number is 70%, so, IPC-MB is expected to be more reliable than PCMB. Besides, in the search of smallest conditioning set to remove a false positive, the way by IPC-MB is the simplest.

Is IPC-MB the most data efficient one among existing works? To our best knowledge, the answer is positive, and , and it is trivial to prove that the conditioning set used in IPC-MB to eliminate a false positive is the smallest by contradiction [1]. However, it is hard to prove that IPC-MB is the most reliable solution to do Markov blanket induction because that

- The actual reliability performance of IPC-MB will be influenced by the underlying topology, including the size of Markov blanket and the connectivity. Generally, the thinner is the connectivity of target network, the more obvious gain is achieved by IPC-MB over the other algorithms;
- In the RecognizePC (Fig. 3), the search has to continue on until $|ADJ_T| \leq cutSetSize$, i.e. there is no more conditional independence test left undone so as nothing more to remove. Then given a polytree like the one in Fig. 5(A polytree is a direct graph with at most one undirected path between any two vertices) and assuming that we want to induce the Markov blanket of D, even all false positives can be removed successfully, the search may not terminate due that $|ADJ_T| = 4$, $cutSetSize$=1 and $|ADJ_T| > cutSetSize$. Additional three rounds of search have to be conducted here, and possibly some true Markov blanket variable may be eliminated due to non-reliable test. Extra time cost is consumed as well, though nothing valuable is found.

There is no way to predict the target topology, so some stopping rule is expected in future research to make IPC-MB more time and data efficient. A naïve one, as proposed in [1], is to stop the search whenever the *cutSetSize* exceeds a pre-defined value, which just corresponds to the second parameter required by KS algorithm as mentioned in Section 3.1. This is trivial, but no correct output is guaranteed.

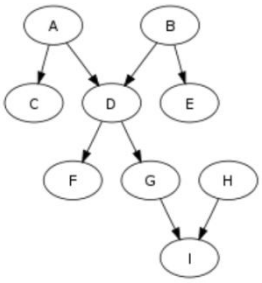


Fig. 5. One polytree example.

V. CONCLUSION

In this paper, we review the reliability of published work about the learning of Markov blanket. In total, eleven algorithms are covered in our discussion. In addition to the analysis of reliability given each algorithm's design individually, we also compare them by cross reference. Finally, we propose possible way for enhancement for existing works. Based on our experience, IPC-MB achieves a best tradeoff among existing works, including time and data efficiency. It can be a valuable reference for researchers of this topic, while deciding where to work further. Also, it can be helpful for practitioners to choose a suitable one for applications.

REFERENCES

[1] S.-K. Fu, "Efficient Learning of Markov Blanket and Markov Blanket Classifier," Ph.D Dissertation, Computer Engineering Department, Ecole Polytechnique de Montreal, Montreal, 2010.

[2] S.-K. Fu and M. C. Desmarais, "Fast Markov Blanket Discovery Algorithm Via Local Learning within Single Pass," in Canadian Conference on AI, Windsor, Canada, 2008, pp. 96-107.

[3] X. Bai, "Tabu Search Enhanced Markov Blanket Classifier for High Dimensional Data Sets," School of Computer Science, Carnegie Mellon University2005.

[4] D. Koller and M. Sahami, "Toward Optimal Feature Selection," in ICML, 1996, pp. 284-292.

[5] J. Pearl, Probabilistic reasoning in expert systems. San Matego: Morgan Kaufmann, 1988.

[6] D. Margaritis and S. Thrun, "Bayesian Network Induction via Local Neighborhoods " in Neural Information Processing System (NIPS), 2000, pp. 505-511.

[7] I. Tsamardinos, et al., "Time and sample efficient discovery of Markov blankets and direct causal relations," in the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 673-678.

[8] I. Tsamardinos, et al., "Algorithms for Large Scale Markov Blanket Discovery," in the Sixteenth International Florida Artificial Intelligence Research Society Conference, St. Augustine, Florida, USA, 2003, pp. 376-381.

[9] S. Yaramakala and D. Margaritis, "Speculative Markov Blanket Discovery for Optimal Feature Selection," in ICDM, 2005, pp. 809-812.

[10] Y. Zhang, et al., "An Improved IAMB Algorithm for Markov Blanket Discovery," Journal of Computers, vol. 5, pp. 1755-1761, 2010.

[11] C. F. Aliferis, et al., "HITON: A novel Markov blanket algorithm for optimal variable selection," in American Medical Informatics Association Annual Symposium, 2003, pp. 21-25.

[12] J. M. Peña, et al., "Towards scalable and data efficient learning of Markov boundaries," International Journal of Approximate Reasoning vol. 45, 2007.

[13] S.-K. Fu and M. Desmarais, "Feature Selection by Efficient Learning of Markov Blanket," in World Congress on Engineering, London, 2010, pp. 302-308.

[14] S.-K. Fu and M. C. Desmarais, "Local Learning Algorithm for Markov Blanket Discovery," in Australian Conference on Artificial Intelligence, Gold Coast, Australia, 2007, pp. 68-79.

[15] S. R. d. Morais and A. Aussem, "A novel scalable and data efficient feature subset selection algorithm," in European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD), Antwerp, Belgium, 2008.

[16] S.-K. Fu and M. Desmarais, "Markov Blanket Based Feature Selection: A Review of Past Decade," in World Congress on Engieering, London, 2010, pp. 321-328.

[17] S.-K. Fu and M. C. Desmarais, "Tradeoff Analysis of Different Markov Blanket Local Learning Approaches," in Advances in Knowledge Discovery and Data Mining, 12th Pacific-Asia Conference (PAKDD), Osaka, Japan, 2008, pp. 562-571.

[18] P. Spirtes, et al., Causation, Prediction and Search (2nd Edition): The MIT Press, 2001.

Sobek: a Text Mining Tool for Educational Applications

[1]E. Reategui, [1]M. Klemann, [2]D. Epstein, [3]A. Lorenzatti
[1]PPGEDU/PPGIE - UFRGS, Av. Paulo Gama, 110, 90040-060, Porto Alegre, RS - Brazil,
eliseoreategui@gmail.com, miriamklemann@gmail.com
[2]Informatics Institute - UFRGS, Postal box 15064, 91501-970 Porto Alegre, RS, Brazil
daepstein@gmail.com
[3]Endeeper, Av. Bento Gonçalves, 9500 - UFRGS, 91509-900 - Porto Alegre - RS – Brazil
alorenza@gmail.com

Abstract — *This paper presents a mining tool to extract relevant terms and relationships from texts, and proposes its use in educational applications. A particular text mining technique is employed to analyze texts and build graphs from them, in which nodes represent concepts and edges represent the relationships between them. Some adjustments are proposed here in the original mining and representation methods, in order to provide results which are more suitable for our educational applications. Two experiments exemplifying the extraction of graphs from students' essays are presented in the paper. Results showed that the mining tool was able to identify a considerable number of relevant terms from the texts analyzed, providing concise representations of documents which can support students' and teachers' tasks.*

Keywords: text mining, graphs, education

1. Introduction

In recent years, data mining and text mining have become more popular in the field of Education mostly because of the growing number of systems which store large databases about students, their accesses to material available, their assignments and corresponding grades. Such expansion in the field yielded the establishment of a community committed to Educational Data Mining applications. This community is concerned mostly with the development of methods for exploring data coming from educational settings, and employing those methods to better understand students and learning processes [2]. In this work, our main goal has been to design and develop a text mining tool to be used in educational applications. Below, we list a few examples of the uses of the tool to support either students' or teachers' work:

- Helping teachers to evaluate students writings from a qualitative point of view;
- Assisting teachers in identifying the significance of students' contributions in discussion forums;
- Supporting reading strategies;
- Supporting text writing;

A particular text mining technique based on statistical analysis has been used to extract graphs from texts, representing relevant terms and their relationships [1]. This technique has been customized here in order to provide results which were more suitable for the targeted applications. Typically, for long documents, the original mining algorithm extracted graphs that were too large to be comprehensible in the proposed educational applications. The changes implemented worked on the reduction of the number of nodes and relationships of the graphs, including on the extraction of compound terms. The next section presents different methods for representing data extracted from texts, including graph-based approaches. Section 3 describes the text mining method on which we have based our research, detailing what has been changed in the original algorithms. Section 4 presents the text mining tool Sobek, and section 5 describes some experiments carried out in order to validate this research. The last section of the paper presents conclusions are directions for future work.

2. Representing information extracted from texts

Representing the information extracted from texts requires specific data-structures. Graphs are an interesting approach which can be used to organize words extracted from texts and keep the relationships between them, including their location inside the text. Since graphs are an abstraction created to represent relationships between objects or concepts, they are easily understood and are widely applied [3]. Adam Schenker proposed a text mining technique to extract information from Internet pages, and defined six different graph models to represent the information extracted from the texts [1]. One of these models, the n-simple distance model, is based on the idea that each statistically relevant word of the text should be connected to the N subsequent relevant words. An interesting feature of this representation approach is that it enables the storage of the relationships between relevant terms found in a text.

Another common representation scheme which has been frequently used in Information Retrieval systems is the vector space model, typically used in text retrieval and document ranking [4][5]. Different adaptations in the model have been proposed along the years as to adjust it to very distinct applications, such as content-based image retrieval combining textual and visual data [6], user modeling [7] or web information retrieval [8]. One of the main features of the

model is to represent each possible term that can appear in a document as a feature dimension. The value assigned to each dimension indicates the number of times the corresponding term appears on it, or it may be a weight that takes into account other frequency information, such as the number of documents in which the terms appear. The model is simple and allows the use of traditional machine learning methods that deal with numerical feature vectors in a Euclidean feature space. However, it discards information such as the order in which the terms appear, where in the document the terms are, how close the terms are to each other, and so forth. As in our educational application it was important to keep a more precise representation regarding the relationships between terms, the n-simple distance model seemed to be a more suitable alternative. The next section explains and proposes some small changes in the model.

3. The Text Mining Method

The text mining method used in this work has been based on the n-simple distance graph model, in which nodes represent the main terms found in the text, and the edges used to link nodes represent adjacency information [1]. Therefore, nodes and edges represent how the terms appear together in the text. Figure 1 shows a graph extracted from a short text about the atomic bomb.

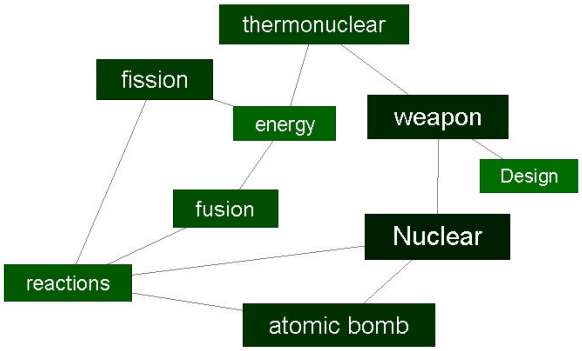


Fig. 1. Graph representing relevant terms extracted from a text about the atomic bomb.

In our graphical representation of the graph, nodes which are more relevant are presented in a larger rectangle and in darker color (e.g. the terms "Nuclear", "weapon", "atomic bomb"). While other text mining approaches rely on the analysis of relevant morph syntactic patterns (such as *"Noun Noun"*, *"Noun Preposition Noun"*, *"Adjective Noun"*, etc.) in order to generate compound terms for the mining process [9], here we used a simpler method which was based on the frequency with which these compound terms appeared in the text.

The method used here relies on a parameter *n* to extract the compound concepts with more than one word. According to this parameter we create a combination of the current word with the *n* subsequent words. What we try to do is to create a wide combination of words to find the most frequent group of words that appear in the text. For instance, considering *n =3*, the analysis of the sequence of terms "AA BB CC DD EE FF GG HH" would lead us to the following combinations "AA", "AA BB", "AA BB CC", "BB", "BB CC", "BB CC DD", and so on. In order to avoid sequences starting with prepositions or articles, specific filters are used. After identifying the most frequent combinations of words, which we will call *concepts*, the mining process selects the primary set of relevant ones based on their frequency in the text.

The next step is to compute the similarity between *concepts*. Consider two *concepts* x = "AA DD BB" and z = "BB CC DD EE FF AA". The similarity coefficient between them is computed with the dot product also used in the Vector Space Model.

The similarity coefficient, represented by S, computes the quantity of words present in both concepts represented by P, and the number of words of the larger concept represented by B. Therefore we have:

$$S = P / B$$

In the example above $S=0.5$ as the concepts have three words in common, words "AA", "BB" and "DD". Concept z, being the biggest, has six terms. After computing the value of S, the relevancy coefficient R is computed for each concept. The number of words of the concept (C) and the absolute frequency (F) are introduced in the computation process. To calculate the R value for each concept, the following formula is employed:

$$R = S * C + F$$

The concept with the largest value for R is kept on the base, and at the end of the process, it is included in the graph. In the example above, let us consider that concept x has $C=3$ and $F=3$, and concept y has $C=6$ and $F=2$. We can conclude that concept z is going to remain in the base to be part of the graph, even if its F value is smaller then that of concept x. The idea behind the relevancy coefficient R is to compare two concepts and keep the one that "says mores", even if it appears a fewer number of times.

4. The Text Mining Tool Sobek

The text mining tool Sobek was developed using the method described in the previous section. The name Sobek comes from the Egyptian mythology where it represents a god related with discernment and patience, features needed to find out relevant and useful information from large amounts of data. Sobek was developed using the Java programming language. The Interactive Graph Drawing API was used to render the graphs on the screen [10]. Sobek is able to analyze documents in "TXT" format, as well as in "PDF" and "DOC" formats. This functionality has been obtained with the use of two different APIs: JPedal [11] and POI [12].

Although Sobek can be employed for the analysis of any type of text, its development has been originally inspired by the need of university teachers who work with distant learning and who have to read dozens of texts, messages and posts written by students [13]. By providing these teachers

with concise graphical representations of the students' texts, Sobek enables them to speed up their work, giving these educators more time to concentrate on specific problems which have to be tackled.

Sobek can be used in different ways. The analysis of plain text is Sobek's simplest operation. The text to be analyzed can be copied and pasted in the tool or it can be loaded from a file. If the text is in a "PDF" or "DOC" format, it is automatically converted to the text format. The main goal of the text analysis is to extract relevant terms and concepts from the text and to visualize their graphical representation in the form of a graph. A teacher could use this procedure, for instance, to visualize and get the main ideas students addressed in their essays. The interface of the mining tool is presented in figure 2, where a text about the atomic bomb has been loaded. The resulting graph obtained from the mining process has been shown in a previous example (figure 1).

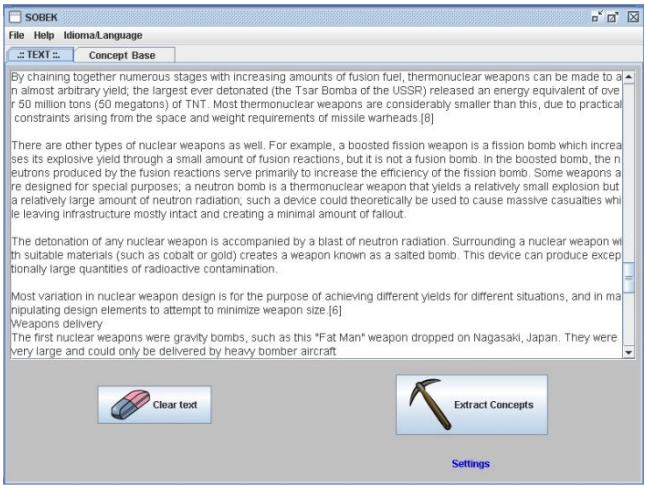


Fig. 2. Sobek's main graphical user interface.

Sobek can also analyze a collection of texts, even if they are in different formats. By comparing the list of terms extracted from the collection of texts with the terms extracted from a student's essay, the system may help teachers to check whether the student's essay addressed topics which were important to be covered. For instance, a teacher could ask his/her students to write a paper based on other articles and texts. Using Sobek, the bibliography given to the students could be employed to create a base of concepts. This database may then be used to verify if the students discussed in their articles the main issues contained in the literature suggested. The base of concepts created automatically can also be edited. Concepts can be added or removed and new relationships between them can be created or extinguished. The base of concepts does not necessarily have to be created from a collection of articles; it can also be created manually by adding concepts and establishing their relationships.

Sobek's first step is to break one or more texts into a set P of words w. This set P of words is analyzed statistically so we may know how many times each word appears in the texts. During the extraction of the words a list of stopwords is used to remove articles, prepositions and terms with no meaning from the base of concepts.

In order to narrow down the number of concepts on the graph and keep only the most important words, we propose the following method:

- Firstly, we set a minimum frequency Θ that will indicate the lowest number of occurrences that a word w must have in order to appear in the graph. As our goal is to provide students and teachers with a concise representation about a text, or a group of texts, it is important to present them only with the most relevant information. Thus, we discard those terms that have a number of occurrences lower than Θ, producing a smaller graph without too many irrelevant terms.
- Secondly, we use a stemmer to remove inflectional and derivational endings of words in order to reduce word forms to a common stem. For example, there is no need for both words *lamp* and *lamps* to appear in the graph. As they both express the same meaning, the one with the highest number of occurrences is displayed.
- After Sobek removes from set P those words that will not be part of the graph, it must verify the relationship between the ones remaining. To do so, for each word $w_i \in P$, we must know which words $w_{j,k} \in P$ come after and before it in the original text. The terms w_j and w_k, called "neighbors of w_j", are added to a list N_i with a counter. If a word w_j appears more than once after or before the word w_i in the original text, its counter will be increased in list N_i. After this process is completed, we have a list of every concept and its neighbors and we can sort it to use only those neighbors with the highest counters. This process enables the tool to display only important relationships between terms, being based on the idea that if word w_j is the neighbor with the highest counter for w_i, it is likely that those two words have some meaning together.

To determine how many relationships have to be shown in the graph for each concept, we use the number of occurrences for those concepts. As a fully connected graph provides no information about how each word is related to the next, we set a maximum number of possible connections Ω. The number of connections for a word $w_i \in P$, will be called Con_i here. The number of occurrence of the word w_i in the original text will be called $NumOc_i$ and the highest number of occurrence will be called $MaxOc$. The word with the highest number of occurrence will have, at most, Ω connections in the graph. Each word will have a number of connections proportional to its $NumOc_i$ and to $MaxOc$. Hence, the number of connections for each word $w_i \in P$ is:

$$Con_i = \frac{NumOc_i * \Omega}{MaxOc}$$

This will assure that those concepts that have a higher $NumOc_i$ will also have a higher Con_i, as they seem to be more important concepts in the text.

5. Evaluation and Results

A first experiment was carried out in order to verify how accurate were the graphs extracted by Sobek. Initially, a two-pages text[1] (816 words) about the topic "Realism" was presented to 20 high school students. The students extracted the graphs from the text, and then each of them wrote about the topic, being able to use the original text as well as the graph as a reference. Figure 3 shows the graph obtained from the text used in the experiment.

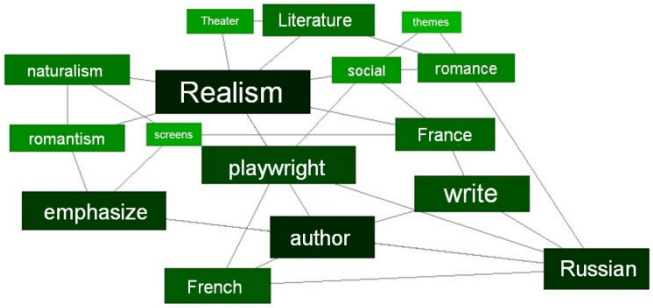


Fig. 3. Graph extracted from text about Realism

The essays produced by the students were then analysed to verify whether the terms of the graph were also present in the students' writings. The results showed that each student used, on average, between nine and ten terms of the graph in his/her essay (9.85 terms, to be precise). Considering that the graph contained a total of 16 terms, an average of 61.6% of the terms identified by the tool as relevant appeared in each text produced. Such results demonstrate that Sobek was able emphasize a considerable number of relevant terms from the text. Table 1 shows the total occurrence of each of the graph's terms in all of the students' texts.

The most cited term of the graph in the students' texts was the word "Realism", which is also highlighted in the graph as the most important term. The least cited terms were the words "write" and "France". Although the order of importance given in the graph for all of the terms does not follow exactly the number of occurrence of these terms in the students' writings, it is interesting to observe that all of the terms extracted from the original text were used by the students in their essays. Such results reinforce the fact that the mining algorithm was able highlight a large number of relevant terms from the text.

A complementary experiment was carried out with Sobek in order to evaluate its capacity to provide summary representations of students' writings. Seven undergraduate students in Computer Science related courses participated, being asked to discuss in a forum about how to design websites, the tools and programming languages available, and the artistic abilities involved.

[1] Complete text, in Portuguese, available at: http://www.artesbr.hpg.ig.com.br/Educacao/11/interna_hpg10.html

Table 1: Total occurrence of terms in students' texts

Graph's terms	Number of occurrence
Realism	100
author	34
Russian	9
playwright	15
emphasize	12
write	5
Literature	42
France	5
romantism	24
naturalism	24
romance	18
Theater	34
social	10
theme	23
screen	23

The teacher who proposed the activity confirmed that the graphs extracted from the students' essays provided a good way to grasp the main topics discussed. Furthermore, the graphs elicited automatically were said to be of great "help not to understand thoroughly what the students had to say, but mostly to skim through good and bad contributions". Figure 4 shows one of the graphs extracted from a student's essay[2].

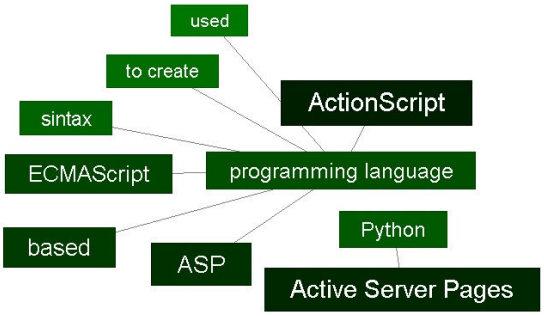


Fig. 4. Graph extracted from a student's essay.

In another example, we may observe how a concise representation of a student's post can reveal a non significant contribution (figure 5).

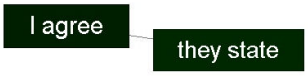


Fig. 5. Smaller graph from another student's writings.

In the example, the only idea represented in the graph was that the student agreed with what was said by other

[2] The graphs presented in this section were originally created in Portuguese, but they have been translated into English here to improve the readability of the paper.

classmates. These results reiterate the applicability of Sobek's mining procedures to elicit relevant information from texts.

6. Discussion

This paper focused on detailing Sobek's main algorithm and showing its effectiveness in providing correct representations of texts' contents, without dealing with domain-specific knowledge. Gao et al. [14] also proposed a method for extracting terms from texts automatically, focusing mainly in business applications. Our approach to differs considerably from this method mainly for its representation mechanism based on graphs, and the consequent specificity it its algorithms.

Other approaches rely heavily on domain knowledge to enrich the mining process, as in Dayanik [15] where a method for text classification is proposed. Information extraction mechanisms also employ domain-specific knowledge to tag and classify data from unstructured or semi-structured documents [16]. The benefits of using or not using domain-specific knowledge in mining applications has also been compared [17], showing advantages and disadvantages of each approach. In the case of our educational applications, it has been an important feature that the mining method runs automatically, with none or little interference from students or teachers. Because of that, the use of domain-specific knowledge has not been considered yet. However, we are currently working on the integration of Sobek with domain ontologies and natural language processing, in order to tag terms and obtain more refined results in the mining process.

As for the presentation of the mining results, other tools present relevant terms extracted from texts by highlighting these terms in the actual document [18], or by simply ranking terms through a frequency count [19][20]. Our solution is based on a more visual representation. From an educational perspective, presenting the mining results in the form of a graph is interesting as it takes learners to focus of concepts and their relationships, and to reflect about them. The associations suggested by the graphs lead learners to reasoning processes that are in many respects similar to those which are triggered during the analysis/development of conceptual maps [21]. However, while conceptual maps represent *noun verb noun* propositions, our representation do not follow such norm. In our mining method, terms may be connected without following any syntatic rule. Besides, conceptual maps are normally built manually, do not relying on any automatic mechanism.

Information Retrieval (IR) is another approach which can be used to find relevant information on textual data, providing an easy way to search textual documents by indexing them with a collection of words. While IR's main objective is to find the right information to satisfy a given query [22], our goal has been to look for hidden patterns inside a body of textual data, with the focus of providing concise representations of documents.

7. Conclusion

The main contribution of this work has been to design a text mining mechanism for educational applications where we proposed a modification in a known text mining process as to produce more knowledgeable outcomes. While the original method generated graphs with one single term represented in each node and with a very large number of connections, in our approach several terms can be placed in a single graph node, and the number of connections have been considerably reduced, according to the size of the text being mined. It could be argued that by connecting nodes with words that appear together frequently in the text, one could represent concepts just the same way we do by placing them together in a single node. However, for the user who has to interpret the graph, it is more difficult to grasp the meaning of a compound term that is dispersed in different nodes.

Other known text mining methods group together terms in order to make more accurate concept extraction from texts, as in [9] where relevant morph syntactic patterns are searched for in order to create meaningful tokens. While such procedure relies on the some level of linguistic processing, our approach is much simpler in that it is based mainly on a statistical analysis of the frequency with which the complete tokens appear in the texts.

Previous research has already shown promising results regarding the use of Sobek in educational applications. For instance, in Macedo et al. [13] it has been demonstrated how Sobek and its graph representation mechanism could give teachers a concise view of the students' assignments by emphasizing important concepts that appeared in the texts. The results of the experiments carried out demonstrated the potential of Sobek's text mining for the analysis of students' work. Furthermore, the tool has also been evaluated by Azevedo et al. [23], who proposed a method for identifying the quality of contributions in discussion forums through a computational method employing the graphs extracted from the students' posts. The authors showed how it is possible to order students' contributions through their concise representations extracted by Sobek. Here, we have focused on detailing Sobek's mining process, specially the add-ons made in the original method. Our validation procedures emphasized not so much the applicability of the mining tool in educational settings, but the accuracy of the mining results.

As for the current use of Sobek, the possibility to create a database of concepts before mining students' contributions showed to be useful approach when dealing with small texts. As discussed in [24], it has been observed that the simple application of statistical analysis on small texts can produce undesirable results, which is inevitable. Sobek is currently being integrated to a virtual learning environment and it will be used by a large number of teachers in several courses. The tool's mining feature is also being improved by connecting it to different domain ontologies.

Acknowledgment

This work has been partially supported by the National Council for Scientific and Technological Development (CNPq - Brazil) under grant 476398/2010-0, FAPERGS Research Support Foundation, under grant 1018248, and Fiocruz-Fiotec project no. ENSP 060 LIV 09.

References

[1] A. Schenker. "Graph-Theoretic Techniques for Web Content Mining". PhD thesis, University of South Florida, 2003.

[2] R. S. J. D. Baker, K. Yacef. "The State of Educational Data Mining in 2009: A Review and Future Visions", Journal of Educational Data Mining, vol. 1, no. 1, p. 3-17, Oct. 2009.

[3] M. Chein, M-L. Mugnier. "Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs", Berlin: Springer Verlag, 2009.

[4] V. V. Raghavan, S. K. M. Wong. "A critical analysis of vector space model for information retrieval". Journal of the American Society for Information Science, vol. 37, no. 5, John Wiley & Sons, p. 279-287, 1986.

[5] D. L. Lee, H. Chuang, K. Seamons, "Document Ranking and the Vector-Space Model," IEEE Software, vol. 14, no. 2, p. 67-75, Mar./Apr. 1997.

[6] T. Berber, A. Alpkocak. "An extended vector space model for content-based image retrieval". In Proceedings of the 10th international conference on Cross-language evaluation forum: multimedia experiments (CLEF'09), C. Peters, B. Caputo, J. Gonzalo, G. J. F. Jones, and J. Kalpathy-Cramer (Eds.). Berlin: Springer-Verlag, p. 219-222, 2009.

[7] E. Mangina, J. Kilbride. "Utilizing vector space models for user modeling within e-learning environments". Computers in Education, vol. *51,* no. 2, p. 493-505, September 2008.

[8] W. Luo, C. Liu, Z. Liu, C. Wang. "On N-layer Vector Space Model-Based Web Information Retrieval". In Proceedings of 6th the International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), Chengdu, China, 23-25 September, p. 1 – 3, 2010.

[9] R. Feldman, M. Fresko, Y. Kinar, Y. Lindell, O. Liphstat, M. Rajman, Y. Schler, and O. Zamir. "Text mining at the term level". In Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD '98), J. M. Zytkow and M. Quafafou, Eds., London, UK: Springer-Verlag, p. 65-73, 1998.

[10] U. Erlingsson and M. Krishnamoorthy. Interactive Graph Drawing. Available at: http://www.cs.rpi.edu/research/groups/pb/graphdraw/. Accessed in March, 2011.

[11] JPedal. Available at: http://www.jpedal.org/ Accessed in March, 2011.

[12] Apache POI Java API. Available at: http://poi.apache.org/ Accessed in March, 2011.

[13] A. Macedo, E. Reategui, A. Lorenzatti, and P. A. Behar. "Using Text-Mining to Support the Evaluation of Texts Produced Collaboratively", in Education and Technology for a Better World: Selected papers of the 9th World Conference on Computers in Education, A. and A. Jones, Eds, Berlin: Springer, 2009, p. 368-377.

[14] X. Gao, S. Murugesan, B. Lo, "Extraction of Keyterms by Simple Text Mining for Business Information Retrieval", In IEEE International Conference on e-Business Engineering, 2005, p.332-339.

[15] A. Dayanik, Using domain knowledge for text mining, PhD dissertation, Rutgers State University, New Brunswick, NJ, 2006.

[16] R. Feldman and J. Sanger. Text Mining Handbook. Cambridge, UK: Cambridge University Press, 2006.

[17] M. Banko, and O. Etzioni. "The Tradeoffs Between Traditional and Open Relation Extraction", In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, Stroudsburg, PA: Association for Computational Linguistics, p. 600-607, 2006.

[18] K. Frantzi, S. Ananiadou, and H, Mima. "Automatic recognition of multi-word terms". International Journal of Digital Libraries, vol. 3, no. 2, p.117-132, 2000.

[19] Textanalyser. Available at: http://textalyser.net/ Accessed in March, 2011.

[20] Wordcounter. Available at: http://www.wordcounter.com/ Accessed in March, 2011.

[21] J. D. Novak and D. B. Gowin. Learning how to learn. New York, NY: Cambridge University Press, 1984.

[22] C. D. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval, Cambridge, UK: Cambridge University Press, 2008.

[23] B. Azevedo, E. Reategui, P. Behar. "Qualitative Analysis of Discussion Forums". In Proceedings of IADIS International Conference on e-Learning, Freiburg, Germany, 2010.

[24] D. S. Leite, L. H. Rino, T. A. Pardo, M. .G. Nunes, "Extractive Automatic Summarization: Does more linguistic knowledge make a difference?". In Proceedings of the Workshop of Graph-based Algorithms for Natural Language Processing (TextGraphs-2), Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics, Rochester-NY, p. 17-24, 2007.

A Novel Soft Computing Hybrid for Data Imputation

Narravula Ankaiah and Vadlamani Ravi* [1]

Institute for Development and Research in Banking Technology (IDRBT),
Castle Hills Road #1, Masab Tank, Hyderabad-500 057 (AP), India
ankireddy.cse@gmail.com ; rav_padma@yahoo.com[1]

Abstract - *We propose a novel 2-stage soft computing approach for data imputation, involving local learning and global approximation in tandem, whereas in the literature only one of them is used. In stage 1, K-means algorithm is used to replace the missing values with cluster centers. Stage 2 refines the resultant approximate values using multilayer perceptron (MLP). MLP is trained on the complete data with the attribute having missing values as the target variable one at a time. The hybrid is tested on 2 benchmark problems each in classification and regression using 10-fold cross validation. In all datasets, some values, which are randomly removed, are treated as missing values. The actual and the predicted values obtained are compared by using Mean Absolute Percentage Error (MAPE). We observe that, the MAPE value is reduced from stage 1 to stage 2, indicating the hybrid approach resulted in better imputation compared to stage 1 alone.*

Key words - Missing data, Data imputation, Multilayer perceptron, K-means clustering, Soft computing, MAPE.

1 Introduction

Missing data in real life data sets is an unavoidable problem in many disciplines. For analyzing the available data completeness and quality of the data plays a major role, because the inferences made from a complete data are more accurate than those made from an incomplete data [1]. Data in the databases may be missed because of data entry errors, system failures at the time of data retrieval or several other reasons like sensor failures, noisy channels cultural issues in updating the databases etc. According to Little and Rubin, 1987 [2], missing data is categorized into 3 categories: (i) missing completely at random (MCAR), (ii) missing at random (MAR), (iii) missing not at random (MNAR). MCAR occurs if the probability of missing variable X does not depends on the values of any other variable in the dataset. This means that the value of missing variable is unrelated to any other variable. For example, if the age of the husband is missed in customers database then it does not depend on the any other variable of database which is meant for wife.MAR occurs if the probability of a missing variable X depends on the other remaining variables in that dataset but not on the variable X. For example, income of a person is missed because of missing in profession and age. MNAR occurs when the probability of a missing variable X depends on the variable X itself. For example, if citizens did not participate in a survey, then MNAR occurs. MCAR and MAR data are recoverable, whereas MNAR is irrecoverable.

Missing data creates various problems in many research fields like data mining, mathematics, statistics and various other fields [1]. The process of replacing or estimating missing data is called data imputation. Data imputation is very useful for data mining applications for getting completeness in the data. For analyzing the data through any technique completeness and quality of data are very important things. For example researchers rarely find the survey data set that contains complete entries [3]. The respondents may not give complete information because of negligence, privacy reasons or ambiguity of the survey questions. But the missing parts of variables may be important things for analyzing the data. So in this situation data imputation plays a major role. Data imputation is also very useful in the control based applications like traffic monitoring, industrial process, telecommunications and computer networks, automatic speech recognition, financial and business applications, and medical diagnosis etc.

To impute with incomplete or missing data, several techniques are reported based on statistical analysis [4]. These methods include like mean substitution methods, hot deck imputation, regression methods, expectation maximization, multiple imputation methods. Some other techniques proposed based on machine learning methods include SOM, K-Nearest Neighbor, multi layer perceptron, recurrent neural network, auto-associative neural network imputation with genetic algorithms, and multi-task learning approaches.

In this paper, we propose a novel soft computing hybrid for data imputation by using K-means clustering and multilayer perceptron. It is a 2 stage approach. In stage 1, the imputation is based on K-means clustering and in stage 2, imputation is based on multi layer perceptron. The remainder of this paper is organized as follows: a brief review of literature imputation of missing data is presented in section 2. Details of design of novel hybrid are presented in section 3. Experimental design described in Section 4. Description about the datasets is given in section 5. Results and discussions are presented in section 6, followed by conclusions in section 7.

* Corresponding author: FAX: +91-40-23535157; Phone: +91-40-23534981; Ext: 2042

2 Literature Review

Missing data handling methods can be broadly classified into two categories: deletion and imputation [5]. The missing data ignoring techniques or deletion techniques simply delete the cases that contain missing data. Because of their simplicity, they are widely used [6] and tend to be the default for most statistics packages, but this solution is not an effective solution. This approach has two forms: (i) Listwise deletion omits the entire cases or records containing missing values. The main drawback of this method is that the application may lead to large loss of observations, which may result in high inaccuracy in particular if the original dataset is itself too small [7]. (ii) Pairwise deletion method considers each feature separately. For each feature, all recorded values in each observation are considered and missing data ignored (Strike et al., 2001). Unlike list wise deletion which removes cases (subjects) that have missing values on any of the variables under analysis, pair wise deletion only removes the specific missing values from the analysis (not the entire case). It is good when the overall sample size is small or missing data cases are large [7].

On the other hand, imputation method uses the available data to estimate the missing values. The earliest method of imputation is mean imputation, in which the missing values of a variable are filled with the average value of all remaining cases of that particular variable [2]. The disadvantage of this method is that it ignores the correlations between various components [8]. When the variables are correlated data imputation can be done with regression imputation. In the regression imputation regression equations are fitted each time by making the variable with incomplete values as the target variable. This method preserves the variance and covariance of missing data with other variable. Hot and cold deck imputation replaces the missing values with the closest complete components. Closest is in terms of components that are present in both vectors for each case with a missing value [8]. The drawback with hot deck imputation is that the estimation of missing data is based on single complete vector. It ignores the global properties of the dataset. The drawback of cold deck imputation is that missing values are replaced with the different dataset values [2]. In multiple imputation procedure, each missing value is replaced with a set of reasonable and valid values, so that we will get M complete datasets by filling each value M times and by analyzing all datasets we can make combined inferences. According to [2], multiple imputation is better than case wise and mean substitution. Regression methods are not as effective as multiple imputation. Expectation maximization is an iterative process that continues until there is convergence in the parameter estimates.

In K-nearest neighbor (K-NN) approach the missing values are replaced with nearest neighbors. The nearest neighbors are the complete components which minimize the distance. In this Method, K nearest neighbors are selected from the complete cases or components. Jerez, Molina, Subirates, and Franco [9] used K-NN for breast cancer prognosis. Batista and Monard [10][11] also used K-NN for missing data imputation. Samad and Harp (1992) implemented SOM approach for handling the missing data. In this imputation once the training of SOM is over with complete records, then incomplete pattern is presented to SOM, its image node is chosen ignoring the distances in the missing variables. An activation group composed of image nodes neighbor is selected. Based on the weights of activation group of the nodes in the missing dimensions the missing values are imputed.

In Multi layer perceptron approach, by using only the complete cases MLP should be trained as nonlinear regression model by making each time one variable as target. By using appropriate MLP model, each incomplete pattern values are predicted. Several researchers [12],[13],[14],[15],[16] used MLP scheme for missing data imputation. Imputation using auto-associative neural network (AANN) is another machine learning technique. In AANN the network is trained for predicting the some inputs by taking same input variable as target variable. Researchers [17,[18] developed imputation models based on AANN.

In this paper we proposed a novel soft computing hybrid which uses K-means clustering and MLP. It is a 2 stage approach; wherein we are approximate the missing values with nearest cluster center in stage 1. The identification of the nearest cluster center is based on the distance. The distance is measured with all cluster centers obtained in K-means clustering and available components of incomplete record. In stage 2, MLP is trained with complete set of records which we have before stage1 as each time incomplete variable as target and remaining variables as inputs. The complete details of proposed hybrid are explained in the following section.

3 Proposed Soft Computing Architecture

The proposed missing data imputation approach is a 2 stage approach. The block diagram (Fig 1) depicts the schema of the proposed imputation method. In this novel hybrid we using K-means [19] clustering for stage 1. K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure for stage 1 imputation as follows:

1. Identify K cluster centers by using K-means clustering algorithm with complete records.
2. Fill the incomplete records with the corresponding features of the nearest cluster center by measuring the Euclidean distance of complete components of an incomplete record and cluster centers.

The distance is measured by using the following formula:

$$d_j = \sum_{i=1}^{m} \left| x^{(j)}{}_i - c_j \right|^2$$

Where j is the number of cluster centers. m is the number of complete components in each record (The value of m may change from one incomplete record to other).

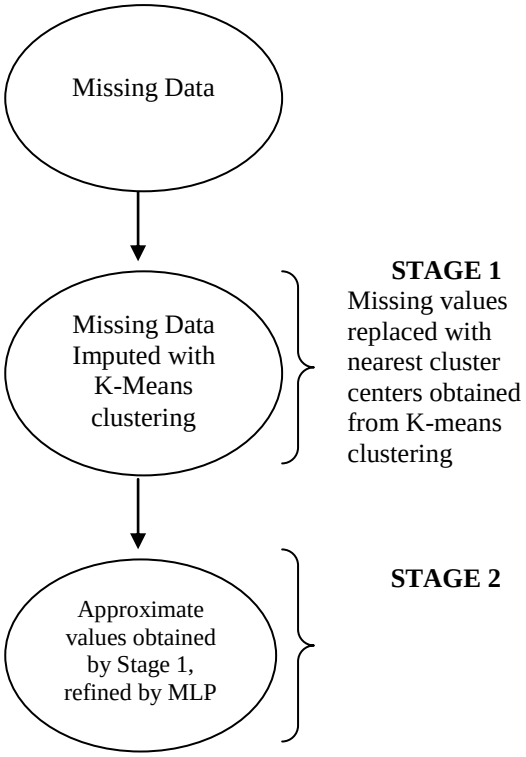


Fig 1: Data Flow diagram for proposed 2-stage imputation

In the second stage, we used multilayer perceptron (MLP) for imputation. MLP is trained by using only complete cases. We have to train as a regression model by taking one incomplete variable as target and remaining variables as inputs. So that we have to form different regression models that are equal to the number of incomplete variables in a given dataset. The steps for MLP imputation (Stage 2) scheme as follows:

1. For a given incomplete dataset X, separate the records that contain missing values from the set of those without missing values (or with complete values). Let us take the set of complete records as known values X_k and incomplete records as unknown records X_u
2. For each incomplete variable, construct an MLP by considering the remaining variables in X_k as inputs for training.
3. Predict the missing values in the variable, which is the target variable in MLP. While predicting we use the initial approximate which are given by K-means clustering from stage 1 as part of test set for predicting the target variable if we have more than one missing value in a record.
4. Repeat step 2 and step 3 for all incomplete variables.

4 Experimental Design

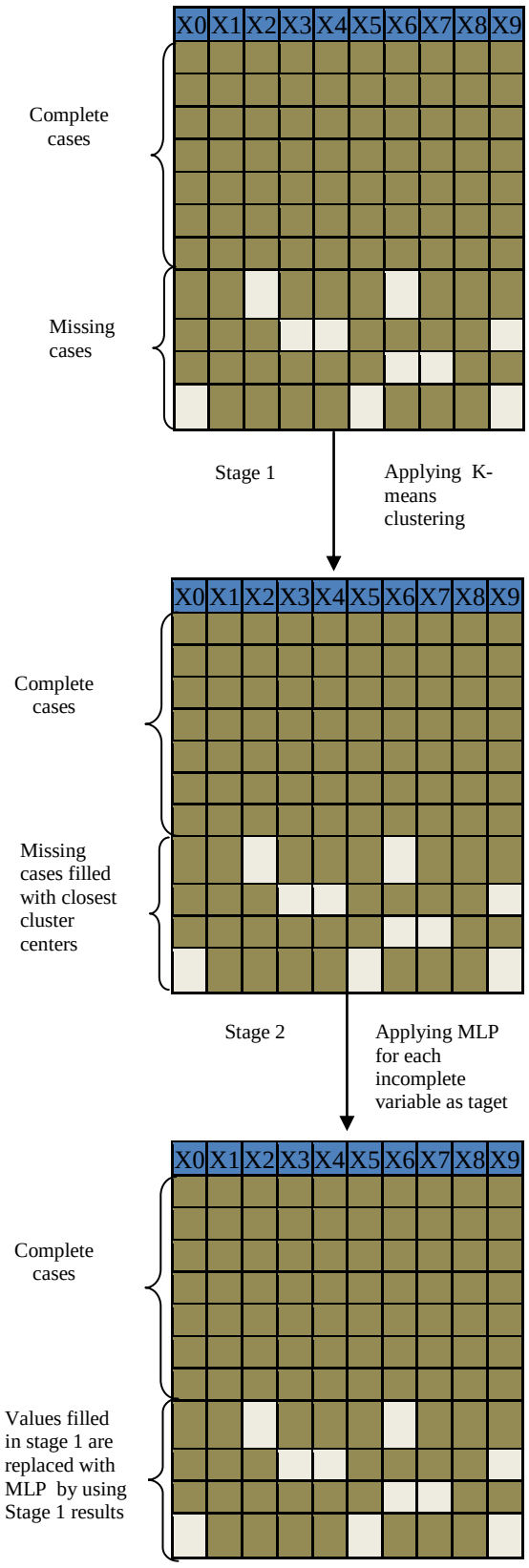
The effectiveness of the proposed method is tested on 2 classification and 2 regression datasets. Since none of these datasets has missing values, we conducted the experiments by deleting some values from the original datasets randomly. Every dataset is divided into 10 folds and 9 folds are used for training and the tenth one is left out for testing. From th test fold, every time, we deleted nearly 10% of the values (cells) randomly. We ensured that at least one cell from every record is deleted. In the stage 1 of data imputation, K-means clustering is performed by using only complete set of records (training data comprising 9 folds). The value of K in K-means is set equal to the number of classes in case of classification datasets. In the case of Wine data the number of classes is 3, so we have chosen K-value as 3. Similarly, in the case of UK banks dataset the number of clusters are chosen as 2. However, in the case of regression datasets, the number of clusters, K, is chosen by visualizing the data using principle component analysis (PCA). By visualizing the plot of PC1 vs PC2, we can set the approximate number of clusters. Thus, the number of clusters is taken as 2 for Boston housing dataset and 3 for forest fires dataset. We can see the plots of PCA visualization for Boston housing and forest fires dataset in Figures 3 and 4 respectively.

In stage 1, the missing values of incomplete records are replaced by closest cluster center components by measuring the distance as explained in section 3. So in stage 1 missing values are replaced by local approximate values. In stage 2, we use MLP for approximating the values closest to actual values by using stage 1 values. We predict the missing values in one attribute, which is the target variable in MLP. While predicting we use the initial approximate which are given by K-means clustering from stage 1 as part of test set for predicting the target variable if we have more than one missing value in a record. The estimation is using 10 fold cross validation of all datasets.

5 Datasets Description

In this paper we analyzed 4 datasets. Those include two regression datasets viz., Forest fires, Boston housing and two classification datasets viz., Wine and UK banks. The benchmark datasets, Wine, Boston housing, and Forest fires are taken from UCI machine learning repository. Forest fires dataset contains 11 predictor variables and 517 records, whereas Boston housing dataset contains 13 predictor variables. Another two datasets we used are Wine and UK bank bankruptcy datasets. Both these datasets are classification datasets. Wine dataset contains 13 predictor variables and 248 records. UK banks dataset contains 10 predictor variables and 60 records. The predictor variables of UK banks dataset are *(i)* Sales *(ii)* Profit Before Tax / Capital

Fig 2: *Block diagram for proposed 2 stage data imputation technique (Stage 1 uses K-means clustering and Stage 2 uses Multi linear perceptron)*



Employed *(%) (iii)* Funds Flow/Total Liabilities *(iv)* (Current Liabilities + Long Term Debits)/Total Assets *(v)* Current Liabilities/Total Assets, (*vi)* Current Assets/Current Liabilities *(vii)* Current Assets-Stock / Current Liabilities *(viii)* Current Assets-Current Liabilities/Total Assets *(ix)* LAG (Number of days between account year end and the date of annual report *and (x)* Age.

6 Results and Discussion

We measured the performance of the proposed approach by using Mean Absolute Percentage Error (MAPE) as the measure of accuracy. MAPE is defined as

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \frac{|x_i - \hat{x}_i|}{x_i}$$

Where n is the number of missing values in a given dataset. $\hat{x}_i$ is the predicted value by the hybrid model for the missing value and x_i is the actual value.

All the estimations of MAPE value using 10 fold cross validations on all datasets. The MAPE values of stage 1 and stage 2 for four datasets are shown in table I. In the case of wine dataset the MAPE value after K-means clustering i.e stage 1 is 28.84% and the MAPE value is reduced to 21.58% after performing stage 2 on the results of stage 1. Similarly in case of another classification dataset the MAPE value is reduced from 46.45% in stage 1 to 32.17% in stage 2.

TABLE I
MAPE VALUES FOR DATASETS (IN %)

Dataset Name	After Stage1	After Stage 2
Wine	28.84	21.58
UK Banks	46.45	32.17
Boston Housing	26.55	15.64
Forest Fires	37.88	26.61

In case of Boston housing dataset also the value of MAPE is reduced from 26.55% in stage 1 to 15.64% in stage 2. In forest fires dataset also the value of MAPE in stage1 is 37.58% and it is reduced to 26.61% after performing stage 2. T- test is applied on 10 folds of all datasets to know that the reduction is chance happening or not. The T-test values are shown in table II. The values in all datasets proves that the reduction is not a chance happening.

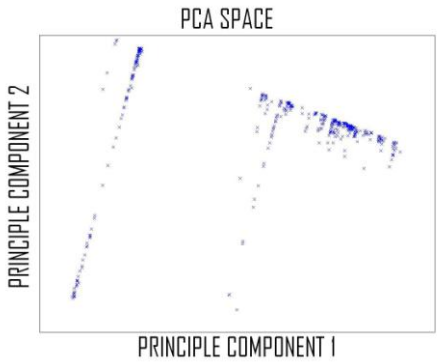


Fig 3: Data visualization by using PCA for Boston housing dataset

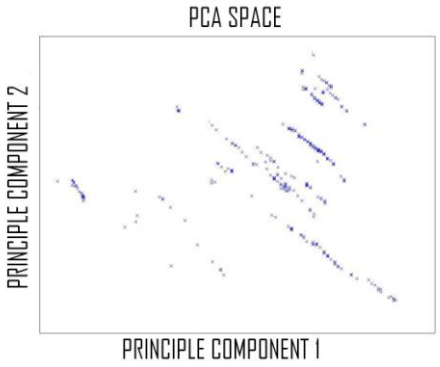


Fig 4: Data visualization by using PCA for Forest fires dataset

We also experimented by changing the K-value in stage 1 to investigate the impact of K value on the results. The resultant variation in MAPE values are presented in Table III. For Wine dataset the MAPE value increasing from 21.58 to 28.32 by changing the K value from 2 to 1. In case of UK banks dataset also MAPE value increasing from 32.17 to 40.25 with the change of K-value from 2 to 1. Thus, in case of wine and UK banks dataset, if the K-value is not equal to the number of classes, then the MAPE value after stage 2 is more compared to the case where K-value equals the number of classes. So for the classification datasets MAPE value is less if K value equals to the number of classes. We also conducted similar experiment by changing the K-value in case of regression datasets. We have taken the K-value as 1 for Boston housing dataset and as 2 for forest fires dataset. In Boston housing dataset, MAPE is more when K=1 than when K=2. In case of forest fires dataset the MAPE value is 32.15 for K=2, and for K=3 MAPE value is 26.61.

TABLE II
VALUES OF T-TEST

Dataset Name	T- test value
Wine	5.56
UK Banks	1.87
Boston Housing	2.54
Forest Fires	2.69

TABLE III
CHANGE IN MAPE VALUES WITH K-VALUE

Dataset Name	MAPE	
Wine	21.58 (K=2)	28.32 (K=1)
UK Banks	32.17 (K=2)	40.25 (K=1)
Boston Housing	15.64 (K=2)	25.84 (K=1)
Forest Fires	26.61 (K=3)	32.15 (K=2)

7 Conclusion

We presented a 2-stage novel soft computing hybrid based on K-means clustering and MLP. The techniques proposed for missing data imputation in the literature used either local learning or global approximation only. In this paper, we replaced the missing values by using both local learning and global approximation. The proposed hybrid is tested on four datasets in the framework of 10 fold cross validation. In all the data sets some values are randomly removed and we treated those values as missing values. In stage 1, by using K-means clustering we replaced missing values by local approximate values. In stage 2 by using the local approximate values which are resulting from stage 1 and trained MLP from complete records, we further approximate the missing value to the actual value. The missing values are replaced by using proposed novel hybrid approach, and then we compared predicted values with actual values by using MAPE. We observed that MAPE value decreased from stage 1 to stage 2. t- test is performed on four datasets, and from the values of t-test we can say that the reduction in MAPE from stage 1 to stage -2 is statistically significant. We conclude that, we can use the proposed approach as a viable alternative to the extant methods for data imputation. In particular, this method is useful for a dataset with a records having more than one missing values.

REFERENCES

[1] M. Abdella and T. Marwala, "The use of genetic algorithms and neural networks to approximate missing data in database," *Computational Cybernetics, ICCC 2005. IEEE 3rd International Conference*, pp. 207-212, 2005.

[2] R.J.A. Little and D.B. Rubin, "Statistical analysis with missing data", *Wiley*, 2nd ed., New Jersey, 2002.

[3] W. Hai , W. Shouhong, "The Use of Ontology for Data Mining with Incomplete Data", *Principle Advancements in Database Management Technologies*, pp. 375-388, 2010.

[4] P.J. Garcı´a-Laencina , J.L. Sancho-Go´mez, and A.R. Figueiras-Vidal, "Pattern classification with missing data: a review", *Neural Comput & Applic*, Vol. 19, pp. 263–282, 2010.

[5] I.A. Gheyas, L.S. Smith, "A neural network-based framework for the reconstruction of incomplete data sets", Neurocomputing, Vol. 73, no. 16, pp. : 3039-3065, 2010.

[6] S. Punnee,"Estimating missing data of wind speeds using neural networks", IEEE proceedings SoutheastCon, 2002.

[7] Q.Song, M. Shepperd, "A new imputation method for small software project datasets", Journal of Systems and Software Vol. 80, , pp. 51-62, 2007.

[8] J.L. Schafer, "Analysis of incomplete multivariate data", *Chapman & Hall*, Florida, 1997.

[9] J. Jerez, I. Molina, J. Subirates, and L. Franco, "Missing data imputation in breast cancer prognosis", *BioMed'06 Proceedings of the 24th IASTED international conference on Biomedical engineering*, 2006.

[10] G. Batista and M.C. Monard, "Experimental comparison of K-nearest neighbour and mean or mode imputation methods with the internal strategies used by C4.5 and CN2 to treat missing data", *Tech. Rep.*, University of Sao Paulo, 2003.

[11] G. Batista, and M.C. Monard, " A study of K-nearest neighbour as an imputation method", *Abraham A et al (eds) Hybrid Intell Syst*, Ser Front Artif Intell Appl 87, IOS Press, pp 251–260, 2002.

[12] P.K. Sharpe, and R.J. Solly, "Dealing with missing values in neural network-based diagnostic systems", *Neural Comput Appl*, Vol. 3, no. 2, pp. 73–77, 1995

[13] S. Nordbotten, "Neural network imputation applied to the Norwegian 1990 population census data", *J Off Stat*, Vol. 12, pp. 385–401, 1996

[14] A. Gupta, and M.S. Lam, "Estimating missing values using neural networks", *J Oper Res Soc*, Vol. 47, no.2, pp. 229–238, 1996

[15] S.Y. Yoon and S.Y. Lee, "Training algorithm with incomplete data for feed-forward neural networks", *Neural Process Lett*, Vol. 10, pp. 171–179, 1999

[16] L. Kallin, "Missing data and the preprocessing perceptron", *Tech. Rep.*, Umeaa University, 2002

[17] M. Marseguerra, and A. Zoia, "The autoassociative neural network in signal analysis. II. Application to on-line monitoring of a simulated BWR component", *Ann Nuclear Energy*, Vol. 32, no.11, pp. 1207– 1223, 2002

[18] T. Marwala, S. Chakraverty, "Fault classification in structures with incomplete measured data using autoassociative neural networks and genetic algorithm", *Curr Sci India*, Vol. 90, no. 4, pp. 542–548, 2006.

[19] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability"*, Berkeley, University of California Press, Vol. 1, pp. 281-297, 1967.

Bankruptcy Prediction in Banks by Principal Component Analysis Threshold Accepting trained Wavelet Neural Network Hybrid

Madireddi Vasu and Vadlamani Ravi[1]
Institute for Development and Research in Banking Technology
Castle Hills Road #1, Masab Tank,
Hyderabad-500057 (AP) India

Abstract - *This paper proposes new principal component analysis-wavelet neural network hybrid (PCA-TAWNN) architecture trained by Threshold Accepting (TA) algorithm to predict bankruptcy in banks. This architecture consists of an input layer, the principal component layer consisting of a few selected principal components, a hidden layer with wavelet activation function and finally an output layer with a sigmoid activation function. The effectiveness of PCA-TAWNN is tested on Turkish, Spanish and UK banks bankruptcy datasets and two benchmark datasets Wine and WBC. We observed that PCA-TAWNN convincingly outperformed other techniques in terms of Area under ROC curve (AUC) in 10-fold cross-validation.*

Key words - Bankruptcy Prediction, Principal Component Analysis, Wavelet Neural Networks, Threshold Accepting

1 Introduction

Over the past three decades several researchers made interesting contributions to predict business failure in banks and firms. It is observed that timely and correctly predicting the failure of a firm is of paramount importance specifically for financial institutions. Bankruptcy prediction for financial firms has been the extensively researched area since 1960's [1]. Creditors, auditors, stockholders and senior management are all interested in bankruptcy prediction because it affects all of them in the same way [2]. In the online methods on site examinations are conducted on banks' premises by regulatory authorities every 12-18 months as mandated by the Federal Deposit Insurance Corporation Improvement (FDIC) act of 1991. Regulators use a six part rating system referred to as CAMELS, which evaluates bank's financial health according to their basic functional areas viz., ***C****apital adequacy*, ***A****sset quality*, ***M****anagement expertise*, ***E****arnings strength*, ***L****iquidity*, and ***S****ensitivity to market risk*. This information is obtained from the bank's balance sheets. While CAMELS ratings clearly provide regulators with important information, Cole and Gunther [3] reported that these CAMELS ratings decay rapidly**.** Further, financial experts are a scarce and expensive resource. Hence, banks thought that it was better to apply off-line, computer based algorithms to determine bank's financial health. They turned out to be not only cheaper but also accurate.

Standard statistical techniques such as regression analysis, logistic regression have been used to analyze a company's financial data in order to predict the financial state of the company. However, this problem can also be solved using various intelligent techniques. The present work introduces a new hybrid PCA-TAWNN, which employs PCA and modified TAWNN in tandem. The rest of the paper is organized as follows. Section II reviews the related work in bankruptcy prediction of banks and firms. Section III presents the proposed PCA-TAWNN architecture. Section IV describes briefly the three bankruptcy datasets that are analyzed by the hybrid model. Results and discussions are presented in section V. Finally, section VI concludes the paper.

2 Literature review

Altman [1] pioneered the research in failure predictions of businesses. He employed financial ratios and Multilinear Discriminant Analysis (MDA) to predict financially distressed firms. While Korobow et al. [4] first applied a probabilistic approach to solve this problem, Karels and Prakash [5] rigorously investigated the normality conditions of financial ratios in the context of DA. Later, Pantalone and Platt [6] employed logistic regression. However, as is well-known, statistical techniques such as MDA have too restrictive assumptions like normality, which are rarely satisfied in practice. Hence researchers turned to non-parametric techniques, which cover the entire gamut of intelligent techniques. Later, statistical techniques like Logit Model [7],[8] and machine learning approaches such as BPNN [2],[9],[10]. Decision Tree [11], K-nearest neighbor and ID3 [12] are applied for failure prediction. Further, hybrid approaches like MDA assisted neural network, ID3 assisted neural network and a self-organizing map assisted neural network [13], MDA, case based reasoning and neural networks [14] are also proposed for predicting the bankruptcy in firms. Many of these studies reported superior performance of BPNN. Therefore, BPNN became an

[1]Corresponding author: FAX: +91-40-23535157; Phone: +91-40-23534981; Ext: 2042

attractive alternative to statistical techniques for bankruptcy prediction.

The trend of hybridizing machine learning and statistical techniques continued. Olmeda and Fernandez [15] predicted bankruptcy in Spanish banks by considering the financial ratios presented in Table 1. They employed BPNN, logistic regression, multivariate adaptive splines (MARS), C4.5 and MDA in devising an ensemble system. They found that BPNN outperformed all other models in the stand-alone mode and the combination of neural network, logistic regression, C4.5 and MDA performed the best among all the combinations. Then, Alam et al. [16] showed that both the fuzzy clustering and self-organizing neural networks are promising tools to predict potentially failing banks. McKee [17] reported that rough set theory significantly outperformed a recursive-partitioning model in predicting bankruptcy. Ahn et al. [18] observed that their hybrid models combining rough sets and BPNN with feature selection and sample size reduction yielded better solutions compared to BPNN and DA for bankruptcy prediction in Korean firms.

Atiya [19] surveyed all the prediction techniques in this context and proposed more financial indicators, to design of a new neural network model**.** Further, Swicegood and Clark [20] compared DA, BPNN and human judgment in predicting bank failures. Shin and Lee [21] generated rules using genetic algorithm (GA) for bankruptcy prediction. Park and Han [22] concluded that K-NN weighted with analytical hierarchy process (AHP) outperformed other models for predicting bankruptcy in Korean firms. Cielen et al. [23] reported that data envelopment analysis (DEA) outperformed C5.0 and a combination of linear programming and discriminant analysis in predicting bankruptcy in Belgian banks. Tung et al. [24] proposed a new neuro-fuzzy system to predict bankruptcy in banks and concluded that the BPNN outperformed this neuro-fuzzy system. Andres et al. [25] reported that a variant of additive fuzzy systems outperformed discriminant analysis and logistic regression in predicting failure of Spanish commercial and industrial firms. Ryu and Yue [26] reported that isotonic separation outperformed BPNN, logistic regression and probit method. Shin et al. [27] concluded that SVM outperformed the BPNN in predicting corporate bankruptcy. Canbas et al. [28] proposed an integrated early warning system (IEWS) for detecting banks experiencing serious problems**.** Becerra et al. [29] reported that wavelet neural networks have advantages over the BPNN in corporate financial distress prediction.

Then, auto associative neural network [30], fuzzy rule based classifier (FRBC) [31], semi-online training algorithm for the radial basis function neural networks (SORBF1 and SORBF2) [32] hybrid of RBF Network with Logit Analysis [33], multiple ensembles of ANFIS, SVM, RBF, SORBF1, SORBF2, Orthogonal RBF and BPNN [34] reported better results compared to BPNN and others. Pramodh and Ravi [35] proposed modified great deluge algorithm trained auto associative neural network for bankruptcy prediction Further, Ravi et al. [36] developed a novel soft computing system based on BPNN, RBF, classification and regression techniques (CART), probabilistic neural network (PNN), and FRBC and PCA based hybrid techniques**.**

Then a comprehensive review of the works using statistical and intelligent techniques to predict bankruptcy in banks and firms during 1968-2005 [37] appeared. Later, , Ravi and Pramodh [38] reported that their threshold accepting trained principal component neural network (PCNN), without a formal hidden layer outperformed BPNN, threshold accepting trained neural network (TANN), PCA-BPNN and PCA-TANN. Then Sun and Li [39] applied weighted majority voting based ensemble of classifiers, Sun and Li [40] employed serial combination of classifiers and Li and Sun [41] developed an ensemble of case based reasoning classifiers to bankruptcy prediction. Most recently, Ramu and Ravi [42] proposed a new privacy preserving data mining technique to predict bankruptcy in banks. Then, Chandra and Ravi [43] applied FRBC preceded by a WNN based feature selection method to predict bankruptcy. Further, Chandra et al. [44] developed Support Vector machine-WNN hybrid (SVWNN) to predict bankruptcy in banks.

3 Proposed PCA-TAWNN

In the recent past, Ravi and Pramod [38], Ravisankar and Ravi [45] and Ravi and Pramod [46] respectively reported hybrid neural networks involving PCA, a kernel variation of PCA and nonlinear PCA for bankruptcy prediction. Also, it is found that WNN [47] yielded good results in predicting bankruptcy. Further it is noticed that TANN performed better compared to BPNN [38]. So the idea of exploring the generalization power of the TAWNN classifier together with the first few principal components (denoted by *npc*) extracted from PCA is worth investigating in bankruptcy prediction. Hence, in this paper, we propose novel hybrid architecture for bankruptcy prediction in banks involving PCA and a modified TAWNN (PCA-TAWNN) as depicted in Fig 1. In the figure, *nin* and *nhn* represent number of input nodes and number of hidden nodes respectively. The proposed hybrid consists of two major phases: (i) The first few principal components are chosen after performing PCA on the data matrix (ii) Then, they are fed as inputs to the modified TAWNN. Here, the original TAWNN (Vinaykumar et al.[48]) is modified by replacing the usual linear activation function by a sigmoidal activation function. At this juncture, it is worth mentioning the difference between PCNN and the proposed PCA-TAWNN. While PCNN follows the design and architecture of Multilayer perceptron, the proposed PCA-TAWNN follows that of wavelet neural network. Secondly, PCNN consists of 3

layers including input layer with a bias node in the PC layer, whereas PCA-TAWNN has 4 layers with no bias node in any layer. Thus, PCA-TAWNN has an extra hidden layer, which performs wavelet related computations by making use of Gaussian wavelet activation function. For more details on PCA and TAWNN, the reader is referred to Rawlings [49] and Vinaykumar et al. [48] respectively.

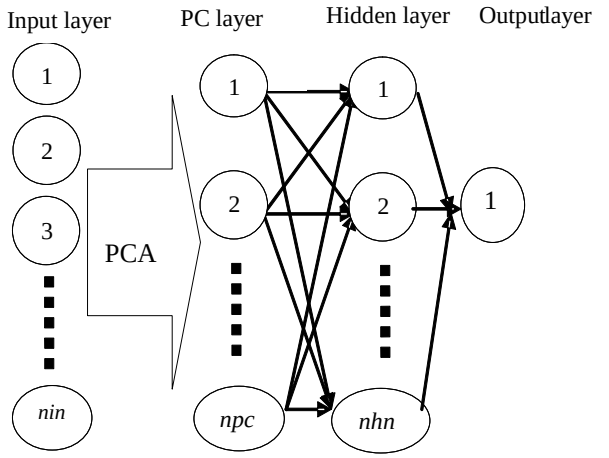


Fig.1.Proposed PCA-TAWNN Architecture

4 Dataset description

In this study, the effectiveness of our proposed hybrid is demonstrated on three bankruptcy datasets *viz.*, Turkish, Spanish and UK banks and two benchmark datasets *viz.*, Wine and WBC. Turkish banks' dataset is obtained from Canbas et al. [28], which is available at (http://www.tbb.org.tr/english/bulten/yillik/2000/ratios.xls). It has 40 banks where 22 banks went bankrupt and 18 banks are healthy. The Spanish banks' data is taken from Olmeda and Fernandez [15]. This dataset contains 66 banks where 37 went bankrupt and 29 healthy banks. The UK banks' data is taken from Beynon and Peel [50].This dataset consists of 60 samples out of which 30 are healthy and 30 are bankrupt. The financial ratios for all the banks are presented in Table 1**.** Wine and WBC datasets are obtained from UCI machine learning repository (http://www.ics.uci.edu/~mlearn/). Wine dataset consists of 178 samples with 13 attributes representing three classes. WBC dataset consists of 683 samples with 9 attributes, where 444 samples belong to benign class and 239 samples belong to malignant class.

5 Results and discussions

We performed 10-fold cross validation throughout and the average results obtained are presented in Table 2-4. The results of the hybrid are compared with those of the previous studies conducted by the research group led by the second author. This comparison was possible because the same experimental design and the same folds were used in the 10-fold cross validation for all the techniques. The quantities employed to measure the quality of the classifiers are sensitivity, specificity and accuracy, which are defined as follows [51].

Table 1: Financial ratios of the datasets

	Turkish banks' data
1	*Interest expenses/Average profitable assets*
2	*Interest expenses/Average non-profitable assets*
3	*(Share holders' Equity + Total income)/(Deposits + Non-deposit funds)*
4	*Interest income/Interest expenses*
5	*(Share holders' Equity + Total income)/Total assets*
6	*(Share holders' Equity + Total income)/(Total assets + Contingencies & Commitments)*
7	*Networking Capital/Total assets*
8	*(Salary and Employees' benefits + Reserve for retirement)/No. of personnel*
9	*Liquid Assets/(Deposits + non-deposit funds)*
10	Interest Expenses/Total Expenses
11	liquid assets/total assets
12	*Standard Capital ratio*
	Spanish banks' data
1	*Current assets/total assets*
2	*Current assets-cash/total assets*
3	*Current assets/loans*
4	*Reserves/loans*
5	*Net income/total assets*
6	*Net income/total equity capital*
7	*Net income/loans*
8	*Cost of sales/sales*
9	*Cash flow/loans*
	UK banks' data
1	*Sales*
2	*Profit before tax/capital employed (%)*
3	*Funds flow/total liabilities*
4	*(Current liabilities + long-term debit)/total assets*
5	*Current liabilities/ Total assets*
6	*Current assets/ Current liabilities*
7	*Current assets – stock / Current liabilities*
8	*Current assets – current liabilities/total assets*
9	*LAG (Number of days between account year end and the date of annual report*
10	*Age*

Sensitivity measures the proportion of the true positives (TP) correctly identified by a classifier.

$$\text{Sensitivity} = TP/(TP+FN)$$

Specificity measures the proportion of the true negatives (TN) correctly identified by a classifier.

$$\text{Specificity} = TN/(TN+FP)$$

Accuracy measures the proportion of true positives and true negatives correctly identified.

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN)$$

where TP, TN, FP and FN respectively stand for true positive, true negative, false positive and false negative.

In the case of Turkish dataset, it is observed that PCA-TAWNN yielded 100% accuracy, while PCNN-WFS-LTF [38] and TAWNN [47] also yielded 100% accuracy as presented in Table 2. Further, in the case of Spanish data (see Table 3), PCA-TAWNN yielded 100% accuracy, whereas PC-TANN [38] yielded an accuracy of 97.5% with an AUC of 8475. Thus, PCA-TAWNN achieved the highest accuracy possible, which could not be obtained by any other classifier so far in literature. In the case of UK data (see Table 4) it is observed that the PCA-TAWNN yielded an accuracy of 89.98% with 95.5% sensitivity and AUC of 8691, which is the best obtained so far. But, SVWNN [38] reported an accuracy of 81.67% with AUC of 7900.

Table 2: Average results of Turkish Banks dataset

Classifier	Acc*	Sens*	Spec*	AUC
MLP [34]	81.67	66.61	90.2	7840.5
TANN [38]	92.5	96.8	93.5	9515
PCA-TANN [38]	97.5	96.8	100	9840
PCNN-WFS-LTF [38]	100	100	100	10000
FRBC [43]	97.5	96.67	100	9833.5
SVM [44]	92.5	86.67	95	9041.5
WNN [47]	95	100	95	9750
DEWNN [47]	95	100	95	9750
TAWNN [47]	100	100	100	10000
SVWNN [44]	95	100	95	9750
KPCNN [45]	92.5	93.75	92.17	9317
PCA-TAWNN (Proposed)	**100**	**100**	**100**	**10000**

Acc*=Accuracy, Sens*=Sensitivity, Spec*=Specificity

Table 3: Average results of Spanish Banks

Classifier	Acc*	Sens*	Spec*	AUC
MLP[34]	81.67	66.61	90.2	7840.5
TANN[38]	91.6	98.5	81.5	9000
PCA-TANN[38]	97.5	97.5	72	8475
PCA-BPNN[38]	84.1	75.4	91.5	8345
Linear RBF[34]	75	51.71	90.17	7094
Orthogonal RBF[34]	40.83	97.5	7.3	5240
RSES[34]	92.5	87.5	97.5	9250
TreeNet[34]	77.96	86.55	93	8977.5
ANFIS [34]	63.34	44.45	79	6172.5
FRBC [43]	96.67	96	100	9800
SVM [44]	88.33	85.83	95	9041.5
WNN [42]	86.67	89.16	81	8508
DEWNN [42]	89.99	91.66	93	9233
TAWNN [42]	88.33	79.66	90.5	8508
SVWNN [44]	90	95	85.17	9008.5
KPCNN [45]	91.67	94.17	92.17	9341
PCA-TAWNN (Proposed)	**100**	**100**	**100**	**10000**

Acc*=Accuracy, Sens*=Sensitivity, Spec*=Specificity

Table 4: Average results of UK Banks dataset

Classifier	Acc*	Sens*	Spec*	AUC
MLP [34]	70.00	73.33	67.17	7025
RBF [34]	71.66	71.66	67.66	6966
ANFIS [34]	75	75.167	78.501	7683.4
TANN [38]	78	87.5	68.33	7791.5
SVM [44]	61.67	90	26.67	5833.5
WNN [47]	78.33	76.33	74.17	7525
SVWNN [44]	81.67	78.83	79.17	7900
KPCNN [45]	80	80.25	75.83	7916.5
PCA-TAWNN (Proposed)	**89.98**	**95.5**	**78.32**	**8691**

Acc*=Accuracy, Sens*=Sensitivity, Spec*=Specificity

Moreover, in the case of benchmark datasets, on WBC dataset, PCA-TAWNN yielded an accuracy of 79.86% with AUC of 7391 and on the Wine dataset it yielded an accuracy of 92.23%. These are however, not the best. The spectacular performance of the PCA-TAWNN on bankruptcy datasets is attributed to the dimensionality reduction capability of the PCA as well as the power and accuracy of the modified TAWNN.

6 Conclusions

In this paper we present a novel hybrid PCA-TAWNN for predicting bankruptcy in banks using PCA and modified TAWNN in tandem. Modified TAWNN comprises logistic or sigmoidal activation function at the output layer instead of the linear one unlike the original WNN and TAWNN. We analyzed three bank datasets *viz.*, Turkish, Spanish and UK and benchmark datasets Wine and WBC. First few principal components obtained from PCA are fed to modified TAWNN. It is observed from the empirical analysis that PCA-TAWNN performs best compared to other recent techniques. In the case of Spanish and Turkish banks datasets the proposed model yielded an astounding 100% percent accuracy, whereas in the case of UK banks dataset it yielded 89.98% accuracy with 95.5% sensitivity. The proposed model performed well on Wine and WBC as well. We conclude that the PCA-TAWNN hybrid can be used as a sound alternative to extant classification algorithms for bankruptcy prediction.

References

[1]. E. Altman, "Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy", *Journal of Finance*, Vol. 23, no. 4, pp. 589-609, 1968.

[2]. R.L. Wilson, and R. Sharda, "Bankruptcy prediction using neural networks", *Decision Support Systems*, Vol. 11, no. 5, pp. 545-557, 1994.

[3]. R. Cole and J. Gunther, "A CAMEL Rating's Shelf Life", *Federal Reserve Bank of Dallas Review*, pp. 13-20, 1995.

[4]. L. Korobow, D. Stuhr and D. Martin, "A Probabilistic Approach to Early Warning Changes in Bank Financial Condition", *Federal Reserve Bank of New York, Monthly Review*, pp.187-194, 1976.

[5]. G.V Karels and A.J. Prakash, "Multivariate Normality and Forecasting of Business Bankruptcy", *Journal of Business Finance and Accounting*, Vol. 14, no. 4, pp. 573-595, 1987.

[6]. C.C. Pantalone and M.B. Platt, "Predicting bank failure since deregulation" *New England Economic Review*, Federal Reserve Bank of Boston, pp. 37-47, 1987.

[7]. J.A. Ohlson,"Financial Rations and the Probabilistic Prediction of Bankruptcy", *Journal of Accounting Research*, Vol. 18, no. 1, pp. 109-131, 1980.

[8]. T. Bell, "Neural Nets or the Logit Model? A Comparison of Each Model's Ability to Predict Commercial Bank Failures" *International Journal of Intelligent Systems in Accounting, Finance and Management, Vol.* 6, no. 3, pp. 249-264, 1997.

[9]. M.D. Odom and R. Sharda, "A Neural Network Model for Bankruptcy Prediction", *IJCNN International Joint Conference on Neural Networks*, Vol. 2, pp. 163-168, San Diego, CA, 1990.

[10]. L. Salchenberger, C. Mine, and N. Lash, "Neural Networks: A Tool for Predicting Thrift Failures", *Decision Sciences*, Vol. 23, no. 4, pp. 899-916, 1992.

[11]. K.Y. Tam and M. Kiang, "Predicting Bank Failures: A Neural Network Approach", *Decision Sciences*, Vol. 23, pp. 926-947, 1992.

[12]. K.Y. Tam, "Neural Network Models and the Prediction of Bank Bankruptcy", *OMEGA*, Vol.19, no.5, pp. 429-445, 1991.

[13]. K.C. Lee, I. Hanand Y. Kwon, "Hybrid neural networks for bankruptcy predictions" *Decision Support Systems*, Vol. 18, no.1, pp. 63-72, 1996.

[14]. H. Jo, I. Han and H. Lee, "Bankruptcy prediction using case-based reasoning, neural networks and discriminant analysis", *Expert Systems with Applications*, Vol.13, no.2, pp. 97-108, 1997.

[15]. I. Olmeda, and E. Fernandez, "Hybrid Classifiers for Financial MultiCriteria Decision Making: The Case of Bankruptcy Prediction", *Computational Economics*, Vol.10, no.4, pp.317-335, 1997.

[16]. P. Alam, D. Booth, K. Lee, and T. Thordarson, "The use of fuzzy clustering algorithm and self-organization neural networks for identifying potentially failing banks: an experimental study", *Expert Systems with Applications*, Vol.18, no.3, pp.185-199, 2000.

[17]. T.E. McKee, "Developing a bankruptcy prediction model via rough set theory", *International Journal of Intelligent Systems in Accounting, Finance and Management*, Vol.9, no.3, pp. 159-173, 2000.

[18]. A.B. Ahn, S.S. Cho, and Y.C. Kim, "The integrated methodology of rough set theory and artificial neural network for business failure prediction", *Expert Systems with Applications*, Vol. 18, no.2, pp. 65-74, 2000.

[19]. F. Atiya, "Bankruptcy prediction for credit risk using neural networks: A survey and new results", *IEEE Transactions on Neural Networks*, Vol.12, no.4, pp.929-935, 2001.

[20]. P. Swicegood, and J.A. Clark, "Off-site monitoring for predicting bank under performance: A comparison of neural networks, discriminant analysis and professional human judgment", *International Journal of Intelligent Systems in Accounting, Finance and Management*, Vol.10, no.3, pp.169-186, 2001.

[21]. K-S. Shin, and Y-J. Lee, " A genetic algorithm application in bankruptcy prediction modeling", *Expert Systems with Applications*, Vol.23, no.3, pp.321-328, 2002.

[22]. C-S. Park, and I. Han, "A case-based reasoning with the feature weights derived by analytic hierarchy process for bankruptcy prediction*", Expert Systems with Applications*, Vol.23, no.3, pp.255-264, 2002.

[23]. A. Cielen, L. Peeters, and K. Vanhoof, "Bankruptcy prediction using a data envelopment analysis", *European Journal of Operational Research*, Vol.154, no.2, pp.526-532, 2004.

[24]. W.L. Tung, C. Quek, and P. Cheng, " GenSo-EWS: a novel neural-fuzzy based early warning system for predicting bank failures", *Neural Networks*, Vol.17, no.4, pp.567-587, 2004.

[25]. J.D. Andres, M. Landajo, and P. Lorca, "Forecasting business profitability by using classification techniques: A comparative analysis based on a Spanish case", *European Journal of Operational Research*, Vol.167, no.2, pp.518-542, 2005.

[26]. Y.U. Ryu, and W.T. Yue, "Firm Bankruptcy Prediction: Experimental Comparison of Isotonic Separation and Other Classification Approaches", *IEEE Transactions on Systems, Management and Cybernetics-Part A: Systems and Humans*, Vol.35, no.5, pp.727-737, 2005.

[27]. K.S. Shin, T.S. Lee, and H.J. Kim, "An application of support vector machines in bankruptcy prediction model", *Expert Systems with Applications*, Vol.28, no.1, pp.127-135, 2005.

[28]. S. Canbas, A. Caubak, and S.B. Kilic, "Prediction of commercial bank failure via multivariate statistical analysis of financial structures: The Turkish case", *European Journal of Operational Research*, Vol.166, no.2, pp.528-546, 2005.

[29]. V.M. Becerra, R.K.H. Galvao, and M. Abou-Seads, " Neural and Wavelet Network Models for Financial Distress Classification", *Data Mining and Knowledge Discovery*, Vol.11, no.1, pp.35-55, 2005.

[30]. J. Baek, and S. Cho, "Bankruptcy prediction for credit risk using an auto-associative neural network in korean firms", *In the proceedings of the CIFEr*, pp.25-29, Hong Kong, 2003.

[31]. P. Ravikumar, and V. Ravi, "Bankruptcy prediction in banks by Fuzzy Rule based classifier", *In the proceedings of 1st IEEE International Conference on Digital and Information Management*, pp.222-227, Bangalore, India, 2006.

[32]. V. Ravi, P. Ravikumar, E. Srinivas, and N.K. Kasabov, "A semi-online training algorithm for the radial basis function neural networks: Applications to bankruptcy prediction in banks", In V. Ravi (Ed.), *Advances in Banking Technology and Management: Impact of ICT and CRM, IGI Global Inc*., USA, 2007.

[33]. B. Cheng , C. L. Chen and C. J. Fu, "Financial Distress Prediction by a Radial Basis Function Network with Logit Analysis Learning". *Computers and Mathematics with Applications*, Vol.51,no.3-4, pp.579-588, 2006.

[34]. P. Ravi Kumar, and V. Ravi, "Bankruptcy prediction in Banks by an Ensemble classifier", *In the proceedings of IEEE International Conference on Industrial Technology*, pp.2032-2036, Mumbai, India, 2006.

[35]. C. Pramodh, and V. Ravi, "Modified Great Deluge Algorithm based Auto Associative Neural Network for Bankruptcy Prediction in Banks", *International Journal of Computational Intelligence Research*, Vol.3, no.4, pp.363-370, 2007.

[36]. V. Ravi, H. Kurniawan, T.P. NweeKok, and P. Ravi Kumar, " Soft Computing System for Bank Performance prediction", *Applied Soft Computing Journal*, Vol.8, no.1, pp.305-315, 2008.

[37]. P. Ravi Kumar, and V. Ravi, "Bankruptcy prediction in banks and firms via statistical and intelligent techniques - A Review", *European Journal of Operational Research*, Vol.180, no.1, pp.1-28, 2007.

[38]. V. Ravi, and C. Pramodh, "Threshold Accepting trained principal component neural network and feature subset selection: Application to bankruptcy prediction in banks", *Applied Soft computing Journal*, Vol.8, no.4, pp.1539-1548, 2008.

[39]. J. Sun, and H. Li, "Listed companies' financial distress prediction based on weighted majority voting combination of multiple classifiers", *Expert Systems with Applications*, Vol.35, no.3, pp.818-827, 2008.

[40]. J. Sun, and H. Li, "Financial distress prediction based on serial combination of multiple classifiers", *Expert Systems with Applications*, Vol.36, no.4, pp.8659-8666, 2009.

[41]. H. Li, and J. Sun, "Majority voting combination of multiple case-based reasoning for financial distress prediction", *Expert Systems with Applications*, Vol.36, no.3, pp.4363-4373, 2009.

[42]. K. Ramu, and V. Ravi, "Privacy preservation in data mining using hybrid perturbation methods: an application to bankruptcy prediction in banks", *International Journal of Data Analysis Techniques and Strategies*, Vol.1, no.4, pp.313–331, 2009.

[43]. D.K. Chandra and V. Ravi, "Feature Selection and Fuzzy Rule Based Classifier applied to Bankruptcy Prediction in Banks", *International Journal of Information and Decision Sciences*, Vol.1, no.4, pp.343-365, 2009.

[44]. D.K. Chandra, V. Ravi, and P. Ravisankar, "Support vector machine and wavelet neural network hybrid: application to bankruptcy prediction in banks", *International Journal of Data Mining, Modelling and Management*, Vol.2, no.1, pp.1 – 21, 2010.

[45]. P. Ravisankar, and V. Ravi, "Failure Prediction of Banks Using Threshold Accepting Trained Kernel Principal Component Neural Network", *World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 7-12, Coimbatore, India, 2009.

[46]. V. Ravi, and C. Pramod**,** "Non-linear Principal Component Analysis based Hybrid Classifiers: An application to bankruptcy prediction in

banks", *International Journal of Information and Decision Sciences*, Vol.2, no.1, pp.50-67, 2009.

[47]. N. Chauhan, V. Ravi, and D.K. Chandra, "Differential evolution trained wavelet neural network: application to bankruptcy prediction in banks", *Expert Systems with Applications*, Vol.36, no.4, pp.7659-7665, 2008.

[48]. K. Vinay Kumar, V. Ravi, M. Carr, and N. Raj Kiran, "Software cost estimation using wavelet neural networks", *Journal of Systems and Software*, Vol.8, no.11, pp.1853–1867, 2008.

[49]. J.O. Rawlings, "Applied Regression Analysis: A Research Tool", *Wadsworth and Brooks/Cole Statistics and Probability Series*. Wadsworth Inc., California, USA, 1988.

[50]. M.J. Beynon, and M.J. Peel, "Variable precision rough set theory and data discretization: an application to corporate failure prediction", *Omega*, Vol.29, pp.561–576, 2001.

[51]. T. Fawcett, "An introduction to ROC analysis", Pattern Recognition Letters, Vol.27, pp.861–874, 2006.

A real application on non-technical losses detection: the MIDAS Project

J.I. Guerrero[1], C. León[1], Senior Member, IEEE, F. Biscarri[1], Í. Monedero[1], J. Biscarri[2] and R. Millán[2]
[1]Electronic Technology Department, University of Seville, Seville, Spain
[2]Automated Metering Management and Field Works Department, Endesa, Seville, Spain

Abstract - *The MIDAS project began at 2006 as collaboration between Endesa, Sadiel and the University of Seville. The objective of the MIDAS project is the detection of Non-Technical Losses (NTLs) on power utilities. The NTLs represent the non-billed energy due to faults or illegal manipulations in clients' facilities. Initially, research lines study the application of techniques of data mining and neural networks. After several researches, the studies are expanded to other research fields: expert systems, text mining, statistical techniques, pattern recognition, etc. These techniques have provided an automated system for detection of NTLs on company databases. This system is in test phase and it is applied in real cases in company databases.*

Keywords: data mining, expert system, text mining, power, utility

1 Introduction

The main objective of data mining techniques is the evaluation of data sets to discover relationships in information. These relationships may identify anomalous patterns or patterns of frauds. Fraud detection is a very important problem in telecommunication, financial and utility companies. Currently, data mining is one of the most important techniques which are applied to solve these types of problems, joined with: rough sets, neural networks, time series, support vector machines, etc. there are a lot of references about the detection of abnormalities or frauds in a set of data.

The increasing of storage capacity and the process capacity allow one to manage large databases. Data mining provides a set of techniques which make information treatment easier. Additionally, there are several techniques of artificial intelligence which can be used to increase the efficiency of data mining methods.

The utility companies have large databases which support the management of customers. In these databases several maintenance and management processes are performed. In addition, the utility companies have to invest their effort in maintenance of infrastructure and anomaly detection. These anomalies are frauds in telecommunication and financial sectors; breakdown or fraud in power, water or gas sectors; etc. The utility companies invest great quantities of efforts to correct it.

The non-technical losses (NTLs) in power utilities are defined as any consumed energy or service which is not billed because of measurement equipment failure or ill-intentioned and fraudulent manipulation of said equipment. This paper describes new advances developed for the MIDAS project. The paper proposes a framework to analyze all information available about customers. This framework uses: data mining, text mining, expert systems, statistical techniques, etc. The proposed framework is used in a test stage in Endesa Company.

In this paper, a description of the framework is made, following these steps:

- Review of current state about the anomaly detection and NTLs detection. Additionally, the Endesa utility company is described.
- The MIDAS Project is explained.
- Architecture of the framework is described.
- Each module is described.
- Finally the conclusions and references are described.

2 Review of current state

2.1 Review bibliographic

Generally, there are several similarities between detection of NTL and the detection of anomalies or anomalous patterns. The detection of anomalies or frauds is treated in several references. Specifically, [1] describes several data mining techniques for fraud detection in credit card, telecommunication networks and intrusion detection; [2] applies several algorithms of data mining and artificial intelligence for fraud detection in the financial field; [3] compared neural networks to statistical techniques for fraud in transactional systems; [4] uses classification methods (principal component analysis and bivariate statistics) for fraud detection in mobile communication network. There are

more references in fraud detection ([5],[6],[7]) but there are less references to NTLs detection, specifically: [8] uses the rough sets for fraud detection in electrical energy consumers; [9] proposed neural network with radial basis function (RBF); [10] and [11] proposed the application of support vector machines (SVMs) with genetic algorithms. Other references use predictive techniques for fraud detection, for example: [12] proposes the integration of artificial neural networks, a genetic algorithm to predict electrical energy consumption and [13] proposed the integration of neural network, time series and ANOVA for forecasting electrical consumption. Sometimes, these references include the demand forecasting in short ([14]), medium ([15]) or long ([16]) term.

2.2 Utility company

The system proposed in this paper is used in test phase in Endesa company. Endesa is the most important energy distribution company of Spain with more than 12 million of clients, and it is found in European and South American markets.

Traditionally, there are two types of losses: non-technical losses (NTLs) and technical losses. The technical losses are caused by faults in distribution lines. These faults are predictable with a low rate of error. The NTLs are caused by breakdown or illegal manipulation in customer facilities. These types of losses are very difficult to predict. Normally, utility companies use massive inspection to reduce NTLs. These inspections are performed on the customer that carry out a series of conditions, as example: customers who have measure equipment without transformers and it is located in a limited geographic zone. These conditions reduce the volume of number of customers to inspect.

When the inspector finds an NTL, the company has to be notified. The inspector stores all information about the problem when it is detected until it is solved. This information is named proceeding.

3 The MIDAS Project

The objective of the MIDAS Project is the detection of Non-Technical Losses (NTLs) using computational intelligence over Endesa databases. This project is the collaboration between Endesa, Sadiel, FIDETIA and Electronic Technology Department of University of Seville. This project began at 2006 with the study of a little set of customers, and getting good results.

Utility companies are very interested in the detection of NTLs. The Technical Losses represents the rest of the losses which is produced by distribution problems (Joole effect). The Technical Losses can be forecasted because they are approximate constants, but the NTLs are very irregular and very difficult to forecast.

The MIDAS project follows the prototype life cycle, and gets a new version at each iteration. In this project a lot of lines are researched: data mining, statistical techniques, neural networks, expert systems, text mining, pattern recognition, etc.

Traditionally, the utility companies used to make massive inspections to avoid the NTLs, but this method is very expensive both in time and in money. Currently the utility companies use more advanced systems that allow the selection of clients who carry out some simple conditions. This type of system allows one to reduce the economic and time cost, increasing the efficiency. But these simple conditions aren't automatic and, normally, they only detect some type of NTLs.

Currently, the prototype developed is in test stage and is tested with Endesa databases. This system has provided better results than the traditional system of inspection.

4 System architecture

The proposed system has an architecture with several modules. Each module is implemented with different techniques. The modules are increased each iteration of life cycle with each prototype. Each prototype is tested with real data of Endesa databases and it is validated with inspections made by Endesa staff.

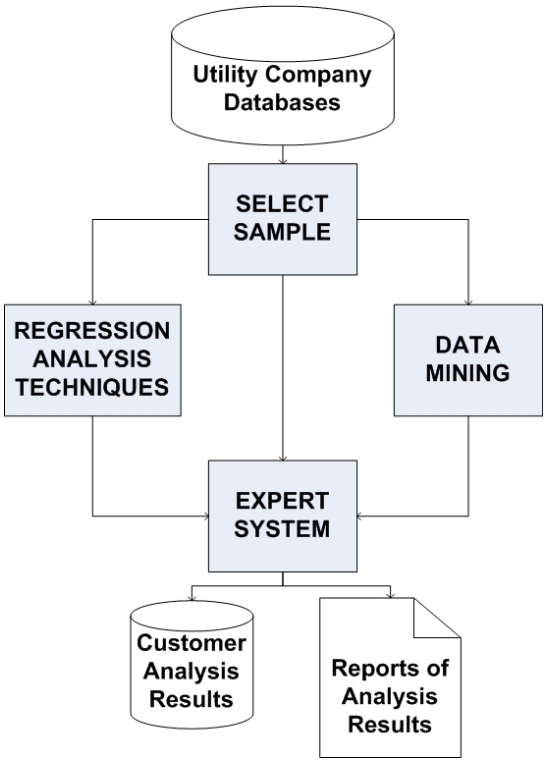


Figure 1. System Architecture

The system architecture is shown in figure 1. In this architecture the different steps of the process are applied in an ordered way. In the first place, a sample of customers is selected using the data stored in utility company databases. In the second place, several artificial intelligence and statistical techniques are applied. This module provides a set of customers who have some anomalies regarding customer self consumption or the other customers consumption. Mainly, this modules works with some parameters: consumption, contracted power, economic activity. In the last place, the knowledge based expert system (KBES) analyzes the rest of information about the customer. The KBES provides the results on databases and reports can be both used by inspectors as an additional source of information. In the following each of these modules is described.

5 Select sample

The sample selection uses several data sources. Each data source provides information about different aspects about the customers. The selection process uses several parameters:

- Period of time of recorded invoices. We use monthly and bimonthly invoices belonging to the sample of customers. Hourly or daily data are not available.
- Geographic localization.
- Contractual power.
- Economic Activity. Some economic sectors historically present a high rate of NTLs. The research of data mining is centered in these sectors.
- Consumption range. Sometimes, the consumption range can be used to restrict the quantity of customers.
- Electricity charges.

These parameters allow one to restrict the quantity of customers to analyze. The information of each customer compounds information about: contract, installed equipment, results of inspections realized over the facilities, etc.

The information about consumption is analyzed by data mining, statistical techniques and neural networks, and then, the rest of information is analyzed by KBES.

6 Data mining

Initially, techniques used in data mining are the outliers analysis and inherent data variability. These techniques are described in [17]. For this process a sample of homogeneous data which have utility customers with similar characteristics are selected. The temporary and the local components of the individual consumption of customer are eliminated by means of normalization. After this process, the probability distribution of the transformed sample, for the normal operating condition, as Gaussian is considered. The threshold of the sample variance is calculated and adjusted. Finally, outliers are used to guide the inspections.

After the use of these methods, the use of other techniques of data mining was considered. Currently, there exists a framework MIDAS. This framework compounds several techniques related with data mining: descriptive data mining, predictive data mining, etc. All of them allow one to increase the efficiency of detection process using the same information. These methods have two processes in common:

- Data Selection. Although, there is a previous step of Selection Sample this step filters the customer who cannot be treated by the proposed data mining techniques.
- Data Preprocessing.

Normally, these processes are used for normalization and discretization purposes.

6.1 Descriptive data mining

This process is described in [18]. Three descriptive techniques are used in this module: one based on the variability of customer consumption, another based on the consumption trend and a third one that summarizes other feature contributions of NTL detection.

The variability analysis provides an algorithm that emphasizes customers with a high variability of monthly consumption in comparison to other customers of similar characteristics. The classic approach to the study of variability classifies data in ‘normal data’ and outliers. The proposed variability analysis uses the standard deviation estimation (STD) to associate to each customer a new feature that will be used as an input for a supervised detection method, showed in the Predictive data mining section.

The consumption trend uses a streak-based algorithm. Streaks of past outcomes (or measurements) are one source of information for a decision maker trying to predict the next outcome (or measurement) in the series. This model is strongly dependent of the cluster of customers considered and highly changeable amongst different clusters. But the study of the individual trend consumption and also the comparative among trends of customer with similar characteristics is very interesting.

There are some feature levels or some feature relationships quite serious in reference to NTLs detection. Some of them are:

- The hours of consumption at maximum contracted power.
- Minimum and maximum values of consumption in different time zones of the day or time discrimination

bands.
- The number of valid consumption lectures. Usually, when there is not a valid lecture value and the company is sure that consumption existed, the consumption is estimated and billed.

6.2 Predictive data mining

This process is described in [18], too. This module uses an inference of a rule set to characterize each of two following classes: 'normal' customer or 'anomalous' customer. Each customer is characterized by means of several attributes. The predictive (or classification) model uses supervised learning with the attributes generated by descriptive data mining.

The classification algorithm uses the Generalize Rule Induction (GRI) model. This model discovers association rules in the data. The advantage is that the association rule algorithm over the more standard decision tree algorithms is that associations can exist between any of the attributes. A decision tree algorithm GRI extracts rules with the highest information content based on an index that takes both the generality (support) and accuracy (confidence) of rules into account. GRI can handle numeric and categorical inputs, but the target must be categorical.

The test of the set of rules generates four values, according to the following classifications [19]:

- True positives (TP). Quantity of test registers correctly classified as fraudulent.
- False positives (FP). Quantity of test registers falsely classified as fraudulent.
- True negatives (TN). Quantity of test registers correctly classified as non-fraudulent.
- False negatives (FN). Quantity of test registers falsely classified as correct.

6.3 Clustering and decision trees

This method is described in [20]. The company in its inspections has developed this method based on the recognition of customers with the same consumption pattern than those NTLs previously detected. This method was held on a process of generation of clusters. Thus, firstly we designed a feature vector that could identify the consumption pattern of each one of the customers. This vector included the following patterns:

- Number of hours of maximum power consumption.
- Standard deviation of the monthly or bimonthly consumption.
- Maximum and minimum value of the monthly or bimonthly consumptions.
- Reactive/Active energy coefficient.

In addition, two parameters are added. These parameters were based on the concept of streak, described in previous section.

7 Regression analysis

This method is described in [21]. This method identifies the patterns of drastic drop of consumption. According to the Endesa inspectors and the studies of consumption, the main symptom of a NTL is a drop in billed energy of the customers.

This method compounds several algorithms: based on regression analysis, based on the Pearson correlation coefficient and based on a windowed linear regression analysis. These algorithms are based on a regression analysis on the evolution of the consumption of the customer. The aim is to search for a strong correlation between the time (in monthly periods) and the consumption of the customer. The regression analysis makes it possible to adjust the consumption pattern of the customer by means of a line with a slope. This slope must be indicative of the speed of the drop of the consumption and, therefore, the degree of correlation. These algorithms identify with a high grade of accuracy two types of suspicious (and typically corresponding to NTL) drops.

8 Expert system

This Rule Based Expert System (RBES) is described in [22]. This system uses the information extracted from Endesa staff. The RBES has several additional modules which provide dynamic knowledge using rules. The expert system has additional modules which uses different techniques: data warehousing (it is used as a preprocessing step), text mining, statistical techniques and data warehousing.

The RBES can be used as additional methods to analyze the rest of information about the customer. The company databases store a lot of information, including: contract, customers' facilities, inspectors' commentaries, customers, etc. All of them are analyzed by RBES using the rules extracted from Endesa staff and rules from the statistical techniques and text mining modules.

The system can be used alone or with other modules to provide an additional method to analyze the information. These modules are described in the following sections.

8.1 Statistical techniques

The statistical techniques are based in basic indicators such as: maximum, minimum, average and standard deviation. These indicators are used as patterns to detect

correct consumption. Additionally, the slope of regression line is used to detect the regular consumption trend. Each of these calculi is made for different sets of characteristics. These characteristics are: time, contracted power, measure frequency, geographical location, postal code, economic activity and time discrimination band. Using these characteristics it is possible to determine the patterns of correct consumption of a customer with a certain contracted power, geographic location and economic activity.

The creation of these patterns needs to study a lot of customers. In this study all customers are not used because the anomalous consumption of the customers with an NTL is filtered. This idea allows the elimination the anomalous consumption getting better results.

Several tables of data are generated as a result of this study. These data are used to create rules which implement the detected patterns. If a customer carries out the pattern, this means that the customer is correct. But if a customer does not carry out the pattern, this does not mean that the customer isn't correct.

8.2 Text mining

Text mining is described in [23] and uses Natural Language Processing (NLP) and neural networks. This method is used to provide a method to analyze the inspectors' commentaries. When an inspection in customer's location is made, the inspector has to register their observations and commentaries. This data is stored in company databases.

This information is not commonly analyzed, because the traditional models are based in consumption study. The text-mining module uses the rest of important information, because the inspectors' commentaries provide real information about the client facilities, which may be different from the stored in database.

This technique uses NLP and fuzzy algorithms to extract concepts from inspectors' commentaries. This concept is classified initially according to their frequency of appearance. The more frequent concepts are classified manually according to their meaning. Additionally, consumption indicators, date of commentary, number of measures (estimated and real), number of proceedings, source of commentary, frequency of appearance, time discrimination band and some others are associated to each concept. This data is used in a neural network, which is trained with data of the more frequent concepts and is tested with the less frequent concepts. This neural network can be used to classify the new concepts which could appear.

9 Highlight cases

The proposed framework has been more efficient in analysis. There are some cases which traditionally were very difficult to detect. Concretely, two cases are treated in this section.

The first case is a client with an irrigation activity. The consumption of this type of client is strongly influenced by climate. The consumption of this client is very irregular, and difficult to analyze. These clients decrease their consumption when rainfalls increase. In this system, data about climate are not available, and only use the information about client. Sometimes, variations of climate conditions make that the data mining or regression analysis techniques select this type of clients. This client is analyzed by expert system, and normally it is dismissed according to the elapsed time since the last inspection.

The second case is the client with seasonal consumption. This type of clients is very difficult to detect with traditionally methods. The consumption of these clients shows one or two great peaks, which can be classified as a fraud. This type of clients can be hotels in coast line, which only has consumption in month with a good climate or in vocational periods. The using of descriptive data mining and expert system allows detect these cases.

10 Conclusions

This paper proposes a framework to detect non-technical losses in power utility. Several techniques are used to detect and classify the customers according to the problem found in them. The main contribution of this framework is the possibility to analyze all the information related to the customer. Traditionally, the analysis is restricted to the consumption and some additional parameters, but this framework compounds:

- Analyzing the coherence of information.
- Analyzing of customers' consumption and trend of consumption:
  - Regarding customer self and their characteristics.
  - Regarding other customers and other customers with same or equivalent characteristics.
- Analyzing the characteristics of customers according to the knowledge extracted from Endesa staff.
- Analyzing the commentaries specified by Endesa staff about the customer and their facilities.
- Reporting all results of analysis.

The analysis of all information provides a more efficient response to the staff company. Additionally, this framework provides advanced knowledge and experience to other users of the company.

This framework uses supervised and unsupervised learning methods. These methods allow one to get better results than traditional methods of massive inspections. The data mining techniques and regression analysis techniques allow one to select consumers with a suspicious consumption and rule based expert system is used to analyze, in depth, the rest of information about each consumer.

11 References

[1] Yufeng Kou, Chang-Tien Lu, Siriat Sinvongwattana, and Yo-Ping Huang, "Survey of fraud detection techniques". Proceedings of the 2004 IEEE International Conference on Networking, Sensing & Control, pp.21-23. Taipei, Taiwan, 2004.

[2] R. Wheeler, and S. Aitken, "Multiple Algorithms for fraud detection" Rev. Knowledge-Based Systems, 13, pp.93-99, 2000.

[3] R. Richardson, "Neural networks compared to statistical techniques". Computational Intelligence for Financial Engineering (CIFEr). Proceeding IEEE/IAFE (1997).

[4] Wang Dong et al., "A feature extraction method for fraud detection in mobile communication networks". Proceeding of the 5 World Congress on Intelligent Control and Automation, Hangzhou, P. R. China, June 15-19, 2004.

[5] David J. Hand, "Prospecting for gems in credit card data". IMA Journal of Management Mathematics 12, pp. 172-200. 2001.

[6] R. Bolton, and D. Hand, "Statistical fraud detection: a review". Statistical Science. Volume 17, Issue 3, pp. 235-255. 2002.

[7] G. K. Palshikar, "The hidden truth – frauds and their control: a critical application for business intelligence". Intelligent Enterprise, Volume 5, Issue 9, pp. 46-51. 2002, 28 May.

[8] J. Cabral, J. O. P. Pinto, E. Gontijo, and J. Reis Filho, "Fraud detection in electrical energy consumers using rough sets". IEEE International Conference on Systems, Man and Cybernetics, Volume 4, pp. 3625-3629, The Hauge, Netherlands, 2004.

[9] J. R. Galván, A. Elices, A. Muñoz, T. Czernichow, and M. A. Sanz-Bobi. "System for detection of abnormalities and fraud in customer consumption". 12th Conference on the Electric Power Supply Industry. Pattaya, Thailand. November 2-6, 1998.

[10] J. Nagi, A. M. Mohammad, K. S. Yap, J. K. Tiong, and S. K. Ahmed. "Non-Technical loss analysis for detection of electricity theft using support vector machines". 2nd IEEE International Conference on Power and Energy (PECon 08), Johor Bahru, Malaysia. December 1-3, 2008.

[11] J. Nagi, S. K. Yap, S. K. Tiong, S. K.Ahmed, and A. M. Mohammad. "Detection of abnormalities and electricity theft using genetic support vector machines". TENCON 2008, IEEE Region 10 Conference. Pp. 1-6. 2008, 19-21 Nov.

[12] A. Azadeh, S. F. Ghaderi, S. Tarverdian, and M. Saberi, "Integration of artificial neural networks and genetic algorithm to predict electrical energy consumption". Applied Mathematics and Computation 186, pp. 1731-1741. 2007.

[13] A. Azadeh, S. F. Ghaderi, and S. Sohrabkhani, "Forecasting electrical consumption by integration of neural network, time series and ANOVA". Applied Mathematics and Computation 186, pp. 1753-1761. 2007.

[14] B. F. Hobbs, U. Helman, S. Jitprapaikulsarn, S. Konda, and D. Maratukulam. Artificial neural networks for short-term energy forecasting: accuracy and economic value. Neurocomputing 23, pp. 71-84, 1998.

[15] M. Gavrilas, I. Ciutea, and C. Tanasa, "Medium-term load forecasting with artificial neural network models". CIRED2001, Conference Publication No. 482. 2001 June.

[16] K. Padmakumari, K. P. Mohandas, and S. Thiruvengadam, "Long term distribution demand forecasting using neuro fuzzy computations". Electrical Power and Energy systems, 21, pp. 315-322, 1999.

[17] F. Biscarri, I. Monedero, C. León, J. I. Guerrero, J. Biscarri, and R. Millán, "A data mining method based on the variability of the consumer consumption". ICEIS 2008: 10th International Conference on Enterprise Information Systems. Barcelona, Spain. Proceedings, Volume 2, pp. 370-374. June 2008.

[18] F. Biscarri, I. Monedero, C. León, J. I. Guerrero, J. Biscarri, and R. Millán, "A mining framework to detect non-technical losses in power utilities". ICEIS 2009: 11th International Conference on Enterprise Information Systems. Barcelona, Spain. Pp- 96-101. May 2009.

[19] J. Cabral, J. Pinto, E. Martins, and A. Pinto, "Fraud detection in high voltage electricity consumers using data mining". In IEEE Transmission and Distribution

Conference and Exposition T&D. IEEE/PES. April 21-24, 2008.

[20] I. Monedero, F. Biscarri, C. León, J. I. Guerrero, J. Biscarri, and R. Millán, "New methods to detect non-technical losses in power utilities". ASC 2009: 13th IASTED International Conference on Artificial Intelligence and Soft Computing. Palma de Mallorca (Spain). Proceedings, pp. 7-13. September 2009.

[21] Iñigo Monedero, Félix Biscarri, Carlos León, Juan I. Guerrero, Jesús Biscarri, and Rocío Millán, "Using regression analysis to identify patterns of non-technical losses on power utilities". KES 2010. Knowledge-Based and Intelligent Information and Engineering Systems, 14th International Conference, Cardiff, UK. September 8-10, 2010.

[22] Carlos León, Félix Biscarri, Iñigo Monedero, Juan I. Guerrero, Jesus Biscarri, and Rocío Millán, "Integrated expert system applied to the analysis of non-technical losses in power utilities". Expert System with Applications, in press [doi: 10.1016/j.eswa.2011.02.062].

[23] J. I. Guerrero, Carlos León, Félix Biscarri, Iñigo Monedero, Jesús Biscarri, and Rocío Millán, "Increasing the efficiency in non-technical losses detection in utility companies". MELECON 2010, 15th IEEE Mediterranian Electromechanical Conference. Pp. 136-141. Valleta, Malta. 25-28 April, 2010.

Capstone Project: Event Monitoring and Alerting System

Wook-Sung Yoo, Geetha Rajgopalan
Software Engineering Department, Fairfield University, Fairfield, CT, USA

Abstract - *Software Engineering program at Fairfield University collaborated with General Electric Corporation through capstone project course to resolve the issue of GE's internal monitoring system in Corporate Information Services (CIS). GE CIS faced the challenge of internal system monitoring strategy to streamline function of its large scaled systems and provide stable services to the businesses. Current monitoring system with static thresholds at GE CIS produced a flood of false alarms and valuable time of IT professionals was lost in responding to these false alarms. The objective of this project was to provide GE a solution by developing a proactive Event Monitoring and Alerting System (EMAS) to predict any abnormal behavior in the monitored systems and send accurate alert on time. Integrating OpenForecast Module and using Java programming and MySQL, we successfully generated multiple performance models to provide early warnings of degrading performance before the services are completely impacted.*

Keywords: Database Mining, Monitoring and Alert system, OpenForecast, Java, MySQL

1 Introduction

General Electric Corporate Information Services (CIS) lacks an enterprise wide system monitoring and management strategy in its large scaled computer systems. With inconsistent local monitoring tools and processes, CIS was faced with challenges in understanding relationships between configuration items and their capabilities. In addition, inconsistent monitoring tools left CIS with no authoritative data source for effective root cause analysis (RCA), automated fault detection or event correlation and analysis (ECA). The conventional alerting system was based on a static threshold, which results in a flood of false alarms and valuable time of IT professional had been lost in responding to these false alarms.

To resolve the issue, GE started an initiative which divided into three phases as shown in Figure 1. Phase I (Standardized Monitoring/Asset), Phase II (a Centralized Business Model), and Phase III (Custom Views). Phase I has previously been completed by GE and became a foundation of Phase II, which houses the scope of this project. The goal of this project is to add intelligence and algorithms to understand the normal behavior of the system and be able to compute and predict abnormal system performance. Algorithms and tools available in the market were researched and analyzed to determine what would align with the needs of GE [1]-[2]. Based on our research we analyzed each tool's compatibility, time investment, ease of use, accuracy in tasks and performance, data set size, projected outcomes, interface capabilities, and accuracy of predictions.

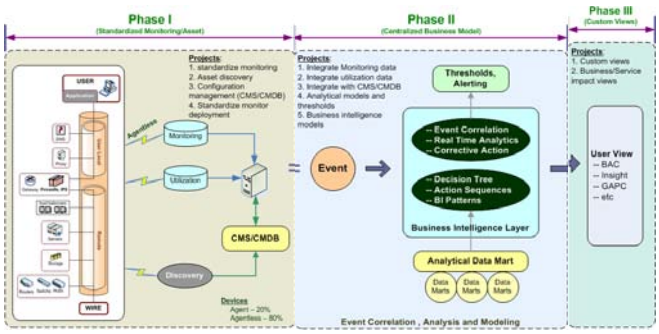


Fig. 1: Three Phases of GE CIS System Monitoring Initiatives

Classical time-series prediction algorithms use history of past observations to predict successive observations [3]. They use historical data to create models which can predict the system's behavior at any given time. Studies indicate that a model can be created to reflect the behavior of the system and this can be used to extrapolate data any number of steps ahead [4]-[6]. Several commercial tools based on time-series prediction algorithms such as *Netuitive* and *Integrien* are available to provide automated, proactive system management [7]-[8]. However, since GE was looking at developing an in-house tool based on open-source tools supporting time series algorithms, we mostly focused on open source products for this project. GE provided data of various system variables captured periodically through their monitoring agents and it was used to create a model using time-series algorithm. *R*, an open-source data analysis tool which supports various predictive algorithms such as Holtz-Winter or Double Exponential Smoothing, was reviewed as a potential candidate [9]. However, the graphical interfaces available for *R* were not easy to customize and converting the data provided by GE to a data format acceptable by *R* was not non-trivial. Other open-source tools such as *Zaitun*, *Cronos*, *RRD*, etc. were also reviewed [10]-[12]. *OpenForecast*, a package of general purpose, forecasting models written in Java, provides support for a wide variety of forecasting models such as linear regression, exponential smoothing, etc [13]. *OpenForecast* also provides support for graphical display using the open source package *JFreeChart* [14]. *OpenForecast* met our functional requirements and was easy to integrate with Java application. Comparison of the some tools under consideration was summarized in table 1.

TABLE 1 Comparison of Data Mining Tools

Tools	Algorithms	Platform	Max Size
WEKA	Time series/clustering/etc	Multi	350 MB
R	Clustering/regression/etc	Multi	300 MB
R Zoo series	Time- Series	Multi	Limited
R Forecast	Time-Series	Multi	Limited
Oracle	Decision Trees/Apriori/etc	Multi	Undetermined
SQL Server	time series/clustering/etc	Windows	Undetermined
Sipina	clustering/regression/etc	Windows	1 GB
Hotz Winter	Time series	Multi	Limited
ZAITUN	Time series	Multi	Limited
Open Forecast	Time series/Clustering/etc	Multi	Unlimited

Concluding our research and analysis, *OpenForecast* Module was selected as the most efficient and supportive tool to fit our project needs.

2 Methodology

2.1 Data Overview

GE provided data of various system variables captured periodically through their monitoring agents and it was used to create a model. The data included 6 servers in GE CIS compiled over a month. Data attributes considered for this project included Server Name, Time Stamp, CPU Utilization, and Ping. These data points were collected at an interval of 5 minutes by the *Sitescope* monitoring agents.

2.2 *OpenForecast* Module

OpenForecast supports various Regression and Exponential models. The forecasting model we selected for the alerting system is the Double Exponential Smoothing model. The Double exponential smoothing model also known as Holt-Winters Exponential smoothing takes into account any trends in the data such as value increasing or decreasing over time. In addition weights are given to observations; meaning that a more recent observation will receive more weight in forecasting than an older observation.

There are two equations associated with Double Exponential Smoothing.

$$st = a.Yt+(1-a)(st-1+bt-1) \quad with\ 0 < a < 1 \qquad (1)$$

$$bt = g.(st-st-1)+(1-g).bt-1 \quad with\ 0 < g < 1 \qquad (2)$$

,where

Yt is the observed value at time t.

st is the smoothed value at time t.

bt is the estimated slope at time t.

a representing alpha - the first smoothing constant, used to smooth the observations.

g representing gamma - the second smoothing constant, used to smooth the trend.

To initialize the model, the first forecasted value is set to the first observed value. The initial slope is set to the difference of the first two observations. The values of the smoothing constants alpha and gamma are calculated by modeling the data series.

2.3 EMAS with Graphical User Interface

To better integrate with Java-based *OpenForecase* Module, the user interface for the Event Monitoring and Alerting System (EMAS) was developed using Java Swing API, a part of the Java Foundation Classes (JFC). JFC encompasses a group of features for building graphical user interfaces and adding rich graphics functionality and interactivity to Java applications. It emulates the native look and feel of the platform it's running on. Java is a platform-independent and its Model-View-Controller framework is highly extensible and customizable. *JFreeChart*, an open-source package, was also used to provide the charting capabilities of the system. For backend development, *MySQL* database was used to save historic data provided by GE and data generated by algorithms. The EMAS has three main functional components: training the model, forecasting with the model, and monitoring and alerting system.

2.3.1 Training the Model

The first stage of the EMAS is training the model. The model takes data from past scenarios and learns from past patterns to predict future scenarios. To train the model the user must select the metric to predict, server to train, and start and end date for data. Only one metric can be selected at a time, hence each metric for the selected server will have its own prediction model and will have to be trained individually. After running the training, the training model generates two constants called alpha and gamma which are stored in the model table for future use in forecasting. The mean absolute deviation for the model is also generated and stored with each server and metric.

Training the model allows for the model to be as up to date and accurate as possible Therefore, if the system is trained every day using previous day's data, the model will be accurate and effective in predicting the data for the next day or week.

2.3.2 Forecasting

The second stage of EMAS is running the predictive model. To run the predictive model, the user selects the server and the metric to predict. Using the model coefficients generated during the training process, the model will generate a forecast for the current day in time interval of 5 minutes.

To forecast for *one*-period ahead the equation (3) is used:

$$F_{t+1} = f_t + b_t \qquad (3)$$

To forecast for *m*-periods ahead the equation (4) is used:

$$F_{t+m} = f_t + mb_t \quad (4)$$

2.3.3 Monitoring and Alerting

The third part of the system is monitoring and alerting. The model forecasts every 5 minutes; it generates a predictive data value and compares it to the value stored in the database. The forecast error, which is the difference between the actual value and the forecasted value, is determined. The control limit for the forecast error is two to five times the mean absolute deviation. If the forecast error is five times the mean absolute deviation, the behavior is marked as abnormal. Two consecutive abnormal values will trigger an email alert.

3 Experimental Results

Tthe sample dataset provided by GE was used to generate a model for a server named *cihcispapp222* for the metric *LoadAverage15minAvg*, which is CPU Utilization rate. The model constants found by the algorithm were: alpha = 1 and gamma = 0.286 as shown in Figure 2.

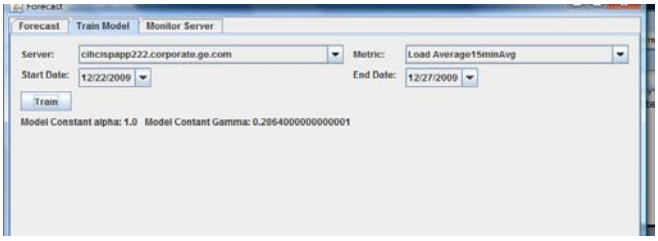


Fig. 2: Training the Model

Figure 3 shows the observed values in red and forecasted value in blue on the value of *LOADAVERAGE15MINAVG* on server *CIHCISPAPP222*. The graphs in figure 3 show that the forecasted values closely follow the pattern exhibited by the observed values.

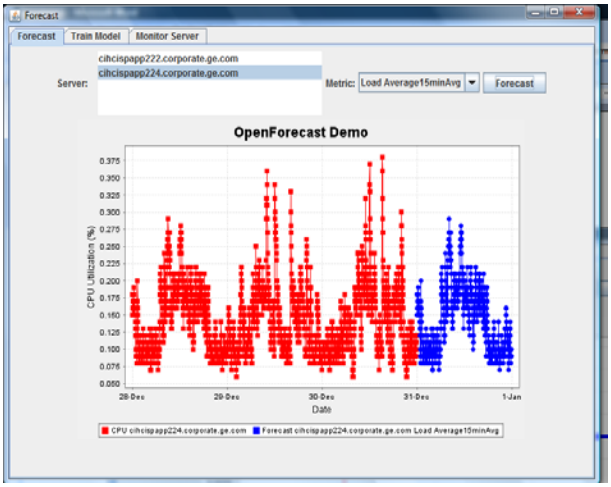


Fig. 3: Forecasting *LoadAverage15MinAvg*

Figure 4 shows the graphs for the measured metric of *Round Trip time*, the time taken by the server to respond back to a request. The observed value is displayed in red (left) and the values forecasted by the model displayed in blue (right).

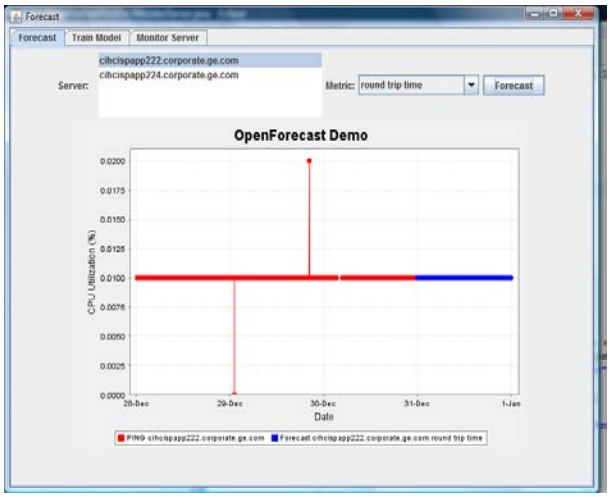


Fig. 4: Forecasting *Round Trip Time*

It is hard to predict a time-series perfectly due to prediction errors. The mean absolute percentage error (MAPE) is commonly used criterion to determine forecast errors. MAPE below 10% is usually considered as good prediction and EMAS generated average of 6% MAPE for CPU Utilization metric. Similarly, the MAPE for the round trip time metric in our experiment showed 0%, which indicates that the forecasted value follows the observed value.

To further validate the results generated by the EMAS, we used the Paired t-test. A paired t test is usually used to compare two groups of data which have matching data points. In this case, it will help us determine if there is any significant difference between the observed values for a server metric against the values forecasted by the model. The two tail p-value helps determine if the two groups are statistically different or similar. If the p-value is less than 0.05, there is a significant difference between the forecasted and observed values. As shown in Table 2, the p-value for the LoadAverage15minAvg is 1.96. This signifies that forecasted and observed values follow similar trend.

TABLE 2. Paired t-Test (Mean of *LoadAverage15minAvg*)

	Forecasted	**Observed**
Mean	0.143754	0.143798799
Variance	0.002443	0.002451863
Observations	666	666
Pearson Correlation	0.999849	
t Critical one-tail	1.647148	
P(T<=t) two-tail	0.179909	
t Critical two-tail	**1.963538**	

Similarly, Table 2 shows the two tail p-value of Round Trip Time to be 2.048 which further confirms that our model is able to forecast pretty accurately.

TABLE 3. Paired t-Test (Means of *Round Trip Time*)

	Forecasted	**Observed**
Mean	0.01	0.010344828
Variance	1.25E-35	3.45E-06
Observations	29	29
Pearson Correlation	2.66E-15	
P(T<=t) two-tail	0.325874707	
t Critical two-tail	2.048407115	

Based on the experimental results shown above, we conclude that the model developed with Double Exponential Smoothing Algorithm can predict server metrics with reasonable accuracy.

4 Conclusions

To resolve the issue of monitoring huge computer systems in GE CIS, EMAS was developed using *OpenForecast* Module with Java and MySQL database and successfully tested. EMAS was delivered to GE CIS in 2010 for further development and is being integrated with current systems for phase III development. However, there were several restrictions in current EMAS related to the data and the model. EMAS assumed input data in consistent time intervals. But, the data provided to the system might have inconsistent time intervals between successive measures of metrics, meaning that data point A will be taken at time interval 5 minutes while data point B will be taken at time interval 7 minutes. Therefore, the data should be properly massaged before using in current EMAS. Further develop0ment of an automatic conversion process that takes a consistent time interval would make the system easier to use. Current EMAS also requires human intervention to train the model for each metric. Additional automatic daily scheduled training module is another recommended future development to capture any changes in the server behavior and hence make the model more adaptive in the future.

ACKNOWLEDGMENT

Our thanks to a team of capstone project at Fairfield University involved in the project: Jennifer Golia, Yao Goka, Geetha, and Preety Sthapit. Special thanks to the GE team: Parag Goradia, Subha Sundaram, and Chris Kalish, the director of GE Edgelab.

5 References

[1] IEEE, ICDM. "Top Ten Algorithms" IEEE, International Conference of Data Mining (ICDM) Springer-Verlag London Limited 2007.

[2] Pal, Sankar. "Scaling Algorithms to Large Data Sets" Pattern Recognition Algorithms for Data Mining CRC Press 2004

[3] G. O. Brockwell, P. J., and Davis, R.. *Introduction to Time-Series and Forrecasting*, 1996.

[4] Vilalta, R., Apte, C.V., Hellerstein,L., MA, L., Weiss, S.M., "Predictive algorithms in the management of computer systems," IBM Systems Journal, VOL 41 NO 3, 2002.

[5] Waclawski, Anthony C., "Autoregressive Integrated Moving Average Models for Comparing Forecasted to Actual Value of CPU Workloads for Open Systems." Paper 286, MCI WorldCom.

[6] Kalekar, Prajakta, "Time Series Forecasting using Holt-Winters Exponential Smoothing", Kanwal Rekhi School of Information Technology, December, 2004. \

[7] Netuitive, Available: http://www.netuitive.com.

[8] Integrien, Available: http://www.vmware.com/products/vcenter-operations/overview.html.

[9] Zucchini, Walter. & Nenadic, Oleg, "Time Series Analysis with R – Part I," 2009.

[10] Zaitun Time Series, Available: http://www.zaitunsoftware.com/.

[11] Cronos Time Series Analysis Package, Available: http://www.stat.cmu.edu/~abrock/oldcronos/.

[12] RRD Tool, Available: http://www.mrtg.org/rrdtool/.

[13] Gould, Steven R, "Getting Started with Open Forecast", Open Forecast User Guide, SAS OnlineDoc, Version 8.

[14] JFreeChart, Available: http://www.jfree.org/jfreechart/api/javadoc/index.html .

Constrained Nonnegative Matrix Factorization for Data Privacy

Nirmal Thapa, Lian Liu, Pengpeng Lin, Jie Wang, and Jun Zhang

Abstract— **The amount of data that is being produced has increased rapidly so has the various data mining methods with the aim of discovering hidden patterns and knowledge in the data. With this has raised the problem of confidential data being disclosed. This paper is an effort to not let those confidential data be disclosed. We apply constrained nonnegative matrix factorization (NMF) in order to achieve what is also known as dual privacy protection that accounts for both the data and pattern hiding, though in this paper, we mainly focus on pattern hiding. To add the constraint we change the update rule as well as the objective function in NMF computation. As the procedure reaches the convergence, it yields a new dataset, which suppresses the patterns that are considered confidential. The effectiveness of this novel hiding technique is examined on two benchmark datasets (IRIS and YEAST). We show that, an optimal solution can be computed in which the user specified confidential memberships or relationships are hidden without undesirable alterations on non-confidential patterns, also referred to as side effects in this paper. This paper presents our idea of how the different parameters will vary to achieve convergence.**

Keywords: Nonnegative Matrix Factorization, Privacy Protection, Data Hiding, Constraint, K-means.

I. Introduction

Privacy concern in data mining has grown to a great level in recent years. Multiple techniques like Value-Class Membership, Value Distortion [1], Matrix Factorization [2], Heuristic-Based Techniques, and Cryptography-Based Techniques [3] have been areas of key interest for the researchers, with each algorithm having its own purpose and limitations. NMF, a matrix factorization technique has been used in various scenarios like text mining [8], part based learning, handwritten digit recognition [10] and many more. Singular value decomposition and nonnegative matrix factorization for the purpose of privacy-preserving has been studied by Wang et al. [4] and Xu et al. [5]. Wang in particular, studied how the pattern-hiding in terms of clustering can be achieved using NMF.

Clustering is very widely studied topic that has been used in different areas including machine learning, data mining, pattern recognition, image analysis, information retrieval, etc. There are many algorithms available for clustering. Among them, k-means is one of the most popular and widely used techniques. Work utilizing NMF for clustering is not a new idea but [11] goes one step further and presents the idea of similarity between the k-means and NMF. In this paper, we present our idea of combining clustering and NMF for the purpose of membership hiding by imposing additional constraint on NMF.

NMF with additional constraints like orthogonality constraint [6] and sparseness constraint [7] have been applied to various fields. Our study uses constrained nonnegative matrix factorization for the purpose of hiding particular membership in a data analysis task. Some initial work in this field i.e., applying NMF for privacy protection was done by Wang et al. [4], [2]. The work by Wang et al. [4] applies NMF in the first phase and then tries to suppress the data pattern using different ad-hoc algorithms. This paper proposes explicit incorporation of the additional constraint in order to suppress the data patterns in the process of performing the matrix factorization, which is a single stage operation.

Nirmal Thapa is with Department of Computer Science, University of Kentucky, Lexington, KY, 40506, Tel: (859) 227-6786, Fax: (859) 323-1971, Email: nirmalthapa@uky.edu

Lian Liu is with Department of Computer Science, University of Kentucky, Lexington, KY, 40506, Tel: (859) 218-6558, Fax: (859) 323-1971, Email: lliuc@uky.edu

Pengpeng Lin is with Department of Computer Science, University of Kentucky, Lexington, KY, 40506, Tel: (859) 218-6558, Fax: (859) 323-1971, Email: pli222@uky.edu

Jie Wang is Assistant Professor in Department of Computer Information Systems, Indiana University Northwest, Gary, IN 46408, Tel:(219) 980-6623, E-mail: wangjie@iun.edu

Jun Zhang is Professor in Department of Computer Science, University of Kentucky, Lexington, KY, 40506, Tel: (859) 257-3892, Fax: (859) 323-1971, Email: jzhang@cs.uky.edu

II. Background

A. K-means clustering

There are many clustering algorithms, like k-means and its variant, Hierarchical clustering, Density based clustering. As mentioned earlier, k-means is the most popular clustering algorithm.

The basic objective of k-means is to cluster the n data items that can be given by $(x_1, x_2, ..., x_n)$, into k sets ($k \leq n$) such that $S = (S_1, S_2, ..., S_k)$ so as to minimize the within cluster sum of squares. Euclidean distance is used as a metric and variance is used as a measure of cluster scatter. Common applications of k-means algorithm are image segmentation and principal component analysis [12], [13].

This paper uses k-means to compare the result. Experiment first runs the k-means on the original data and based on the result, ground truth is established. It must be noticed that the ground truth is the result that k-means returns, that may not be the exact classification. We perform the NMF and then compare the result to the one from the run on original data to see if there is any side effect (discussed later) or if any element that we wanted to change has not been changed.

B. Nonnegative Matrix Factorization

There are many kinds of matrix factorization like principal component analysis (PCA), singular value decomposition (SVD), and NMF. NMF is different in the sense that it imposes additional constraint that none of the elements of

the factor matrix H and basis matrix W can be negative. Another notable thing about NMF is that, results are non-unique which provides even better ground for it to be used for data protection.

Nonnegative matrix factorization is a way in linear algebra where a matrix A is decomposed into the product of two matrices H and W. R is the residual since $H \times W$ will not always be equal to A.

$$NMF(A) \Rightarrow H \times W$$
$$A = H \times W + R, A \approx H \times W = \widetilde{A}$$

Formally it can be defined as *Given a nonnegative data model A(n×m), find two nonnegative matrices $H^{n\times k}$ and $W^{k\times m}$ with k being the number of clusters in A, that minimize Q, where Q is an objective function defining the nearness between the matrices A and HW. The modified version of A is denoted as $\widetilde{A}$= H×W .*

Generally, $(n + m)k < nm$, which reduces the rank of the original matrix. In other words, the original matrix will be compressed. There are two main aspects, one is the $objective\ function$ and the other is the $update\ rule$. Objective function quantifies the quality of factorization usually in terms of distance between the two matrices A and HW. The Euclidean distance or the Frobenius norm is the common function to consider. Objective for NMF would be to minimize the distance between A and HW.

$$\min_{H\geq 0, W\geq 0} f(A, H, W) = \|A - HW\|_F^2$$

Since, NMF is an iterative technique; there is the need to update matrices H and W in each iteration. Rule to do so is termed as update rule. We will discuss more on that one in the following sections.

C. Data Pattern Hiding

Data Hiding can be defined as the process of changing the data with the aim of hiding the confidential data and at the same time minimizing the alteration to the non-confidential data. This paper mainly focuses on the problem of confidentiality in terms of clustering. We want the information about the cluster membership of some particular data not to be disclosed.

As said earlier, NMF generates two matrices H and W for a nonnegative data matrix A, which are nonnegative factor matrices generated by minimizing the objective functions. Matrix W represents coefficients for clusters and has size of $k \times m$ defining basis vectors. While H has size of $n \times k$, contains cluster membership indicators representing additive combination for each subject. To apply this idea to data pattern hiding, we can find out cluster membership of data by finding the largest element in the factor vector from H, provided factor vectors are related to the cluster property of the subjects [2]. The shift of a subject from one cluster to another cluster occurs whenever the factors are modified. This is the essence on which data pattern hiding is based on. Let us say, we have n items in total with k clusters, we want to change the cluster membership of an item X which was originally in cluster C_i. In such a case, there are two ways in which we change the membership. It can be either of the following two:

- Change the membership of item X to a particular cluster C_j, such that $i \neq j$.
- Change the membership of item X to any cluster other than cluster C_i.

We discuss about how to explicitly specify that information into the NMF in a later section. One important aspect Wang et al. [2] mentions in their work is the issue of side effect, which is discussed in the following section.

D. Side Effect

Side effect can be defined as the unwanted changes that are introduced after applying the constrained nonnegative matrix factorization. In our case, it is the cluster membership of the data. As it is directly related to the utility of the data, it is necessary to keep the changes in cluster membership of non-confidential data to a minimum. Any technique must have the property to keep the side-effect to the minimum level in order for it to be useful. Ideally, all the confidential data are changed and nothing else is altered. In our method, we strive to achieve this goal.

There must be some measure of side effect and for that we make the comparison against the k-means that we run for the original data. Hence, the number of subjects that get changed by the application of the method can be taken as the measure of side effect.

III. Constraint on Nonnegative Matrix Factorization

Researchers have come up with different constraints to be incorporated in NMF for solving different tasks. Some works are based on orthogonality [6] and some are based on sparseness [7]. Wang et al. [2] used the magnitude of the elements of the factor vector H_x in the H matrix to determine the cluster category of the subject X. Seeing at the work previously done on constraints, we would like to add a constraint which we called the $clustering\ constraint$ that results in the matrix H that will either have one of the element significantly large compared to others which represents the new cluster for the item or one of the element insignificant in terms of magnitude so as to make sure that the element does not fall in that cluster.

The objective function can be modified to accommodate penalty terms as;

$$f(A, H, W) = \alpha\|A - HW\|_F^2 + \beta\|H - C\|_F^2 \qquad (1)$$

Here, C is a matrix of size $n \times k$, and the elements of C are such that

- If the item is not to be changed then, its contents will be 1 on the index representing its cluster and the rest of them are 0.
- If the item is to be changed to another particular cluster, then contents will be 1 on the index representing destination cluster and the rest of them are 0, which we refer as *in a cluster change*.

- If the item is to be changed to any other cluster, then contents will be 0 on the index representing source cluster and the rest of them are some random number in the range [0-1], referred to as *not in a cluster change*.

Typical example of it is

C1	C2	C3	
0	0	1	Element in Cluster 3
1	0	0	Element in Cluster 1
0.45	0.55	0	Element in any cluster other than Cluster 3

A. Update Formula

Mathematical derivation for update formula

Let,

$$\begin{aligned} Q &= \|A - HW\|_F^2 \\ &= tr((A-HW)^T(A-HW)) \\ &= tr(A^TA - A^THW - W^TH^TA + W^TH^THW) \\ &= tr(A^TA) - 2tr(A^THW) + tr(W^TH^THW) \quad (2) \end{aligned}$$

also let,

$$\begin{aligned} L &= \|H - C\|_F^2 \\ &= tr((H-C)^T(H-C)) \\ &= tr(H^TH - H^TC - C^TH + C^TC) \\ &= tr(H^TH - 2H^TC + C^TC) \quad (3) \end{aligned}$$

- H fixed and W changing,

$$\begin{aligned} &\frac{\delta f(A,H,W)}{\delta W} \\ &= \frac{\delta(\alpha\|A - HW\|_F^2 - \beta\|H - C\|_F^2)}{\delta W} \\ &= \alpha\frac{\delta(Q)}{\delta W} - \beta\frac{\delta(\|H-C\|_F^2)}{\delta W} \\ &= -2\alpha\frac{\delta(tr((A^THW)))}{\delta W} + \alpha\frac{\delta(tr((W^TH^THW)))}{\delta W} \\ &= -2\alpha H^TA + 2\alpha H^THW \quad (4) \end{aligned}$$

- W fixed and H changing

$$\begin{aligned} &\frac{\delta f(A,H,W)}{\delta H} \\ &= \frac{\delta(\alpha\|A - HW\|_F^2 - \beta\|H - C\|_F^2)}{\delta H} \\ &= \alpha\frac{\delta(Q)}{\delta H} - \beta\frac{\delta(\|H-C\|_F^2)}{\delta H} \quad (5) \end{aligned}$$

We know, first term gives,

$$\alpha\frac{\delta Q}{\delta H} = -2\alpha AW^T + 2\alpha HWW^T \quad (6)$$

Second term gives,

$$\begin{aligned} \beta\frac{\delta\|H-C\|_F^2}{\delta H} &= \beta 2H - 2C + 0 \\ &= 2\beta H - 2\beta C \quad (7) \end{aligned}$$

Combining (6) and (7) in (5),

$$\begin{aligned} &= -2\alpha AW^T + 2\alpha HWW^T + 2\beta H - 2\beta C \\ &= 2\alpha HWW^T + 2\beta H - 2\alpha AW^T - 2\beta C \quad (8) \end{aligned}$$

For optimal solution $\frac{\delta q}{\delta W}$=0 and $\frac{\delta q}{\delta H}$=0. Hence,

$$H^TA \oslash H^THW = I$$

$$H(\alpha WW^T + \beta) \oslash (\alpha AW^T + \beta C) = I$$

where, $\oslash$ represents element-wise division, I denotes identity matrix. This gives rise to the update formula for W and H as,

$$W_{i,j} = W_{i,j}\frac{[H^TA]_{i,j}}{[H^THW]_{i,j}} \quad (9)$$

$$H_{i,j} = H_{i,j}\frac{[\alpha AW^T + \beta C]_{ij}}{[H(\alpha WW^T + \beta)]_{ij}} \quad (10)$$

B. Objective Function

As mentioned earlier, the objective function needs to be changed to incorporate the constraint. Let us start with our initial formula:

$$f(A,H,W) = \alpha\|A - HW\|_F^2 + \beta\|H - C\|_F^2 \quad (11)$$

The Objective here is to not only make $\|A-HW\|_F^2$ smaller but to make the sum of both the terms in the above equation small, which gives rise to,

$$\min_{H\geq 0, W\geq 0}(\alpha\|A - HW\|_F^2 + \beta\|H - C\|_F^2) \quad (12)$$

This is the objective function that will be used to check the convergence. If the value is below certain limit the NMF process is considered to have converged.

IV. Algorithm

In this section, we present the algorithm devised for the data pattern hiding. The algorithm for the constrained NMF is as follows:

Original data matrix A, k, C, tol, $maxIter$, $mainIter$, α,

Algorithm 1: Constrained NMF

input : $A \in \mathbb{R}_+^{n\times m}, 0 < k \ll min(n,m), C \in \mathbb{R}_+^{n\times k}, mainItr, tol, maxItr, \alpha, \beta$
output: $H \in \mathbb{R}_+^{n\times k}, W \in \mathbb{R}_+^{k\times m}$
Initialize H and W with the random initial estimates
$H_{i,j}^{(0)} \Leftarrow nonnegativevalue, 1 \leq i \leq n, 1 \leq j \leq k$
$W_{i,j}^{(0)} \Leftarrow nonnegativevalue, 1 \leq i \leq k, 1 \leq j \leq m$
for $i \leftarrow 1$ **to** $mainItr$ **do**
  for $j \leftarrow 1$ **to** $maxItr$ **do**
    $H_{i,j} \leftarrow H_{i,j}\frac{[\alpha AW^T+\beta C]_{ij}}{[H(\alpha WW^T+\beta)]_{ij}}$
    $W_{i,j} \leftarrow W_{i,j}\frac{[H^TA]_{i,j}}{[H^THW]_{i,j}}$
    Calculate new Â
    if *value(Objective Function)*$\leq$ *tol* **then**
      break
  if *sideeffect=0* **then**
    break
  Change value of α
  Change value of β

β are the input to the algorithm. The tol provides the stopping criterion, in other words measurement of convergence, $maxIter$ presents the number of update to perform in H and W before stopping an NMF if convergence is not achieved. The output from the algorithm is the two matrices H and W, such that $\tilde{A} = H \times W \approx A$, where all the confidential data are hidden and non-confidential data are intact. The constrained NMF algorithm is run for a certain number of iterations and checked each time if pattern hiding has been achieved, if there is any side effect the algorithm continues to perform NMF other wise it stops. The paper does not show how the side-effect is calculated in the algorithm above. It mainly is comparing the k-means result on the modified data with the k-means result on the original data for the non-confidential data and comparing against what we wanted in the beginning for the confidential data.

V. Experimental Results

For this experiment, positive real data are needed as it deals with NMF. It was necessary to avoid categorical data, since this research does not deal with categorical data. IRIS dataset is fairly standard in data mining community but it is desirable to base the experiments on multiple datasets. We wanted to make sure that the two datasets have different number of attributes that will give more variation in what the experiments were tested against. Experiments were performed with IRIS and YEAST datasets, both of which are fairly known datasets.

- **IRIS Data Set**: IRIS is a simple data set with 150 instances in a 4-dimensional attribute space. The four attributes are sepal length, sepal width, petal length and petal width. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant: Iris Setosa, Iris Versicolour, and Iris Virginica.

- **YEAST Data Set**: YEAST is a real-valued data set having 1484 instances and 8 attributes. It is used to predict the localization site of protein, which has 9 classes. The experiments used three class of data from YEAST dataset. Variation in size of the dataset was another focus for the experiment so, we took the classes having the highest number of tuples.

All the experiments took 3 classes of data whether it is IRIS data or YEAST data. All the comparisons were based on the ground truth which was the result obtained by using k-means algorithm on the original data.
In all of the experiments it was made sure that there were not any side effects, it was encouraging that algorithm was able to change the membership of all the confidential subjects while keep the membership of non-confidential data intact.

A. Experiment 1

Two types of changes were made, first was *in a cluster change* with graphs as shown in Fig. 2, 4 and second was *not in a cluster change* shown in Fig. 1, 3 . Experiment was

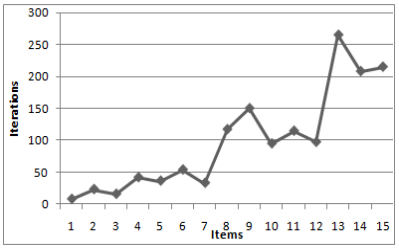


Fig. 1. Not in a cluster (IRIS)

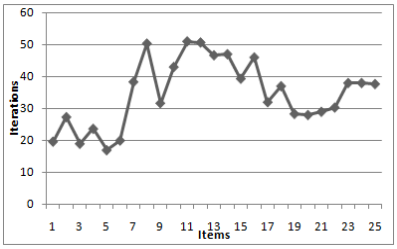


Fig. 2. In a cluster (IRIS)

done to observe the number of iterations it took to make all of the changes. The algorithm was unable to get convergence for more than 15 items for *not in a cluster change* while the number goes beyond 25 for *in a cluster change* in the case of IRIS data. Similar observation was made for the YEAST data. It can be concluded from the experiment that it takes lot lesser iterations to make the *in a cluster change* compared to not *in a cluster change*. It can be attributed to the fact that one element in a row of the matrix H needs to be significantly large compared to the others for it to work efficiently.

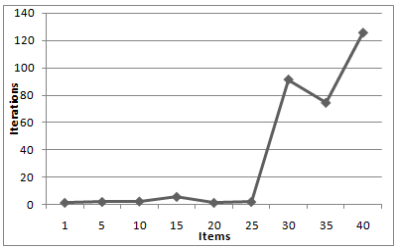


Fig. 3. Not in a cluster (YEAST)

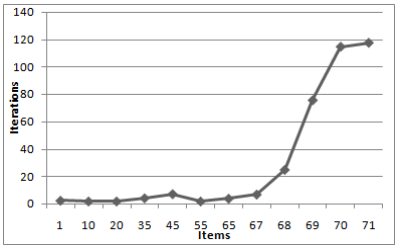


Fig. 4. In a cluster (YEAST)

B. Experiment 2

Next experiment was to study the relation between the value of α and β with the number of confidential subjects in order to achieve convergence. We performed this experiment both with the IRIS and the YEAST data. For each dataset, we ran experiment for small n (total number of changes) and larger n. One thing to remember is that the number of *in a cluster changes* were equal to the number of *not in a cluster changes*. Initially, $\alpha = 0.9$ and $\beta = 0.1$ and then the value of α was decreased by 0.1 and value of β was increased by 0.1, the aim is to keep $(\alpha + \beta) = 1$, so that our estimated solution does not diverge from the actual solution. Experiment was repeated again, but this time with the initial value of $\alpha = 0.1$ and $\beta = 0.9$ and increase the value of α by 0.1 and decrease β by 0.1. Each of the experiment was performed 100 times and to see what region in terms of value of α and β gives the most convergence.

Class	α	β
1	0.1	0.9
2	0.2	0.8
3	0.3	0.7
4	0.4	0.6
5	0.5	0.5
6	0.6	0.4
7	0.7	0.3
8	0.8	0.2
9	0.9	0.1

Fig. 5. Classes for α and β

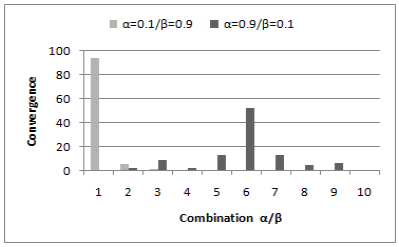


Fig. 6. IRIS data with n=10

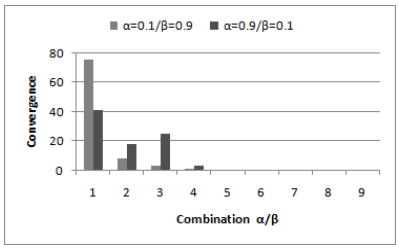


Fig. 7. IRIS data with n=26

We can see from Fig. 6 that, when n is small, we get most convergence in that class of α and β combination where we start the iteration from, but as we increased n to 26 as in Fig. 7, we can see, that most convergence occurs in the region where $\beta > \alpha$.

Similar observation was made for the YEAST data as well from Fig. 8,9, when n is small values of α and β do not play as important part in the convergence. When n grows large then the distribution shifts towards the region with smaller β and greater α, indicating that to change larger number of

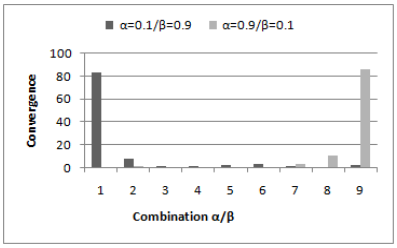


Fig. 8. YEAST data with n=10

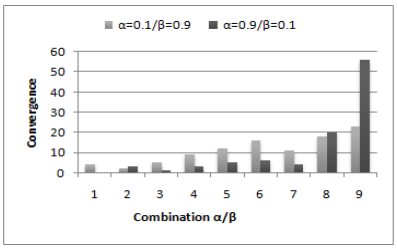


Fig. 9. YEAST data with n=38

data the value of α and β should be in this particular range. The value of α and β basically depends upon the data. As in the case of IRIS data it was $\beta > \alpha$ but for YEAST data it was more $\beta < \alpha$ region.

C. Experiment 3

In the third experiment, we tried to study the relation between the total amount of data and the changes that can be made. The following tables show that total number of changes that were made successfully with different amount of data.

Data Size	Changes
60	10
90	14
120	20
150	24

TABLE I
IRIS CHANGES AND DATA SIZE

Data Size	Changes
150	50
180	70
240	90
300	110
360	140
420	140
450	150
480	150

TABLE II
YEAST CHANGES AND DATA SIZE

It can be seen from the table that convergence can be achieved even for the ones we did not get convergence for after increment in the number of items each cluster is made.

VI. CONCLUSION

We proposed a technique to change the membership of confidential data while making no change to non-confidential data by integrating the constraint on the NMF algorithm. We were able to change the membership of more than one item. The lesser number of iterations for *in a cluster change* against *not in a cluster change* is a good indicator that one of the elements in H matrix needs to be significantly higher

compared to other elements. From the experimental result we were able to see how the values of α and β should be changed as the number of items that we are changing grows or shrinks. The relation between the number and size of data was also studied.

VII. Future Works

There is still much to be done in this field. One prospective area might be to study the relation between the dimension of the data and the number of iterations that it takes for the convergence; it can be a very important and interesting thing to do in the future. For the immediate future, study of the utility of data with other application or even some sort of metrics on the distortion level with the change in the number of subjects can be a good step. We applied the clustering constraint in this paper; there is the possibility of combining the constraint with other constraint like orthogonality constraint or sparseness constraint to better the result.

References

[1] R. Agrawal and R. Srikant,"Privacy Preserving Data Mining,"*Proc. ACM SIGMOD Conf. Management of Data*, pp. 439-450, May 2000.

[2] Jie Wang, Jun Zhang, Lian Liu, and Dianwei Han, "Simultaneous data and pattern hiding in unsupervised learning," *The 7th IEEE International Conference on Data Mining - Workshops(ICDMW07)*, pages 729-734, Omaha, NE, USA, October 2007. IEEE Computer Society.

[3] V.S. Verykios, E. Bertino, I.N. Fovino, L.P. Provenza, Y. Saygin, and Y. Theodoridis, " State-of-the-Art in Privacy Preserving Data Mining," *ACM SIGMOD Record*,vol. 3, no. 1, pp. 50-57, Mar. 2004

[4] Jie Wang, Weijun Zhong, and Jun Zhang, " NNMF-based factorization techniques for high-accuracy privacy protection on non-negative-valued datasets," *2006 IEEE Conference of Data Mining, International Workshop on Privacy Aspects of Data Mining*, pp. 513-517. IEEE Computer Society, 2006.

[5] Shuting Xu, Jun Zhang, Dianwei Han, and Jie Wang, " Singular value decomposition based data distortion strategy for privacy protection," *Knowledge and Information Systems*, 10(3):383-397, 2006.

[6] Li, H., Adal, C., Wang, W., Emge, D., and Cichocki, A.,"Non-negative matrix factorization with orthogonality constraints and its application to raman spectroscopy," *The Journal of VLSI Signal Processing*, 48, pp. 83-97 (2007).

[7] Patrik O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, 5:1457-1469, 2004.

[8] Wei Xu, Xin Liu , Yihong Gong, "Document clustering based on non-negative matrix factorization," *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. New York: Association for Computing Machinery*, pp. 267-273, 2003.

[9] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational Statistics and Data Analysis*, 52(1):155-173, 2007.

[10] M. Mazack, "Non-negative Matrix Factorization with Applications to Handwritten Digit Recognition," Department of Scientific Computation, University of Minnesota, 2009.

[11] C. Ding, X. He, and H. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," *In Proceedings of SIAM Data Mining Conference*, 2005.

[12] H. Zha, C. Ding, M. Gu, X. He and H.D. Simon, "Spectral Relaxation for K-means Clustering," *Neural Information Processing Systems vol.14 (NIPS 2001)*, pp. 1057-1064, Vancouver, Canada. Dec. 2001.

[13] C. Ding and X. He, " K-means Clustering via Principal Component Analysis," *Proc. of Int'l Conf. Machine Learning (ICML 2004)*, pp 225-232. July 2004.

An Optimization Framework for Process Discovery Algorithms

A.J.M.M. (Ton) Weijters
Department of Industrial Engineering and Innovation Science
Eindhoven University of Technology
Email: a.j.m.m.weijters@tue.nl

***Abstract*—Today there are many process mining techniques that, based on an event log, allow for the automatic induction of a process model. The process mining algorithms that are able to deal with incomplete event logs, exceptions, and noise typically have many parameters to tune the algorithm. Therefore, the user needs to select the right parameter setting using a trail-and-error approach. So far, there is no general method available to search for an optimal parameter setting. One of the problems is the lack of negative examples and the omission of a standard measure for the quality of mined process models. Therefore, the so-called *k-fold-cv* experimental set up as used in the machine learning community cannot be applied directly. This paper describes an adapted version of the k-fold-cv set-up so that it can be used in the context of process mining. Illustrative experimental results of applying this method in combination with the HeuristicsMiner process mining algorithm and three different performance measurements are presented. Using the k-fold-cv experimental set-up and an event log with low frequent behavior and noise, it appears possible to find the optimal parameters setting. Another important result is that the simple combination of yes/no parsing of a trace in combination with negative examples based on noise is sufficient for parameter optimization. This makes the framework universal applicable for benchmarking of different process mining algorithms with different process model representation languages.**

Keywords: Process Mining, Parameter Optimization, 10-fold Cross Validation

I. Introduction

Process mining has proven to be a valuable approach that provides new and objective insights into the way business processes are actually handled within organizations. Taking a set of real executions (the so-called *event log*) as the starting point, these techniques attempt to extract non-trivial and useful process information.

The driving element in the process mining domain is some *operational process*, for example a business process such as an insurance claim handling procedure in an insurance company, or the booking process of a travel agency. Nowadays, many business processes are supported by *information systems* that help coordinating the steps that need to be performed in the course of the process. Work-flow systems, for example, assign work items to employees according to their roles and the status of the process. Typically, these systems record events related to the activities that are performed, e.g., in *event logs*, audit trails, or transaction logs [2].

These event logs are the input for process mining algorithms, which include techniques for the discovery of different process perspectives (e.g., control-flow, time, resources, and data) and related techniques such as conformance checking, verification, etc. In the case of *discovery*, a model is automatically constructed based on an event log. In the remainder of this paper we focus on algorithms that discover the control-flow perspective of a process. In particular, we focus on validation techniques for these process discovery algorithms. While process mining has reached a certain level of maturity and has been used in a number of real-life case studies (see [1] for an example), *a common framework to evaluate process mining results is still lacking*. Many publications contain some illustrative experimental examples (see [3], [5], [10], [6] for example) but many different event logs and quality measurements are used. We believe that there is the need for a framework that enables (a) users of process mining tools to search for an optimal process model, and (b) process mining researchers to compare the performance of their algorithms. A first attempt is presented in [15], [16]. This paper focuses in more detail how to use an adapted version of the k-fold-cv experimental set up [12] to search for an optimal process model.

From a theoretical point of view, the rediscovery problem discussed in this paper is related to the work discussed in [4], [8], [9], [14]. In these papers the limits of inductive inference are explored. For example, in [9] it is shown that the computational problem of finding a minimum finite-state acceptor compatible with given data is NP-hard. Several of the more generic concepts discussed in these papers could be translated to the domain of process mining. It is possible to interpret the problem described in this paper as an inductive inference problem specified in terms of rules, a hypothesis space, examples, and criteria for successful inference. The comparison with literature in this domain raises interesting questions for process mining, e.g., how to deal with negative examples (i.e., suppose that besides log W there is a log V of traces that are not possible, e.g., added by a domain expert). However, despite the many relations with the work described in [4], [8], [9], [14] there are also many differences, e.g., we are mining at the net level rather than sequential or lower level representations (e.g., Markov chains, finite state machines, or regular expressions).

There is a long tradition of theoretical work dealing with

the problem of inferring grammars out of examples: given a number of sentences (traces) out of a language, find the simplest model that can generate these sentences. There is a strong analogy with the process-mining problem: given a number of process traces, can we find the simplest process model that can generate these traces. A good overview of prominent computational approaches for learning different classes of formal languages is given in [13]. Many issues important in the language-learning domain are also relevant for process mining (i.e. learning from only positive examples, how to deal with noise, measuring the quality of a model, etc.). However, an important difference between the grammar inference domain and the process-mining domain is the problem of concurrency in the traces: concurrency seems not relevant in the grammar inference domain.

There are many process mining techniques that, based on an event log, allow for the automatic induction of a process model. However, all existing mining techniques have limitations when mining common constructs (i.e. loops, long distance dependencies, noise, duplicate tasks, etc.) in process models. More formal process mining algorithms have the tendency to construct perfect models that only allow for the exact behavior as available in the event log (i.e. the learning material). In practical situations with incomplete event logs and noise, these kinds of algorithms are less useful. Heuristics driven process mining algorithms are more robust for noise and incomplete event logs. Most of the time different parameters are available with which the amount of detail in the mined model can be influenced. However, so far there is no general method available to search for an optimal parameter setting. One of the problems is the lack of negative examples and the absence of a standard measure for the quality of mined process models. Therefore, the so-called *k-fold-cv* experimental set up as used in the machine learning (ML) community cannot be applied directly. This paper describes an adapted version of the k-fold-cv set-up so that it can be used in the context of process mining. Illustrative experimental results of applying this method in combination with the HeuristicsMiner (HM) process mining algorithm are presented. We have chosen the HM because this algorithm requires many parameter settings and it is able to deal with noise, exceptions, and incompleteness.

The paper is organized as follows. First, in Section 2 a characterization of the HM is presented. In Section 3 we describe the adapted version of the k-fold-cv experimental set-up in the context of process mining. In Section 4 we present the experimental results of applying the adapted experimental set-up in combination with the HM algorithm and two event logs. Finally, in Section 5 we draw our conclusions and discuss future work.

A. The HM Mining Algorithm

This section shortly describes "heuristics driven" process mining algorithm; the so-called "HeuristicsMiner" (HM). The algorithm is implemented as a plug-in in the ProM framework [7]. The *Heuristics Miner* (HM) [17] is a practical applicable mining algorithm that can deal with noise and can be used to express the main behavior (i.e., it can abstract from details and exceptions) registered in an event log. It supports the mining of all common constructs in process models (i.e., sequence, choice, parallelism, loops, invisible tasks and some kinds of non-free-choice), except for duplicate tasks. The HM algorithm has three steps. In the first step, a *dependency graph* is built. In the second step, the *semantics of the split/join points* in the dependency graph are set. In the third step, long distance dependencies are added. Below the basic ideas of the three steps are presented; for the details of the algorithm we refer to [17].

Step 1 - Dependency Graph The basic building blocks of the dependency graph are the *dependency relations* that are derived for any two tasks in the log. The basic idea is that the more frequently a task t *directly* precedes another task t' in the log, and the less frequently the opposite situation occurs, the higher the dependency relation between t and t'. Since short loops are exceptions to this basic idea, special dependency relations have been defined for *length-one* and *length-two* loops. Additionally, as the HM algorithm targets the mining of noisy logs, thresholds have been defined to set which dependency relations are valid. The construction of the dependency graph works as follows. First, all dependency values are calculated. Second, it accepts the short-loop dependency relations that are above the thresholds specified by the parameters "Length-one-loops threshold" (σ_{L1L}) and "Length-two-loops threshold" (σ_{L2L}). Third, for every different task t in the log, it selects its "best" precedence/succession dependency relations. After this third step completes, the "trunk" of the dependency graph has been built. Finally, other relevant dependency relations are added to the dependency graph. These relations are identified based on the values for the parameters "Dependency threshold" ($sigma_D$), the "Positive observations" (σ_P), and "Relative-to-best threshold" (σ_R). In short, the HM algorithm also accepts dependency relations between tasks that have (i) a dependency measure above the value of the respective dependency threshold parameter, *and* (ii) a frequency in the log that is higher than the value of the positive observations, *and* (iii) a dependency measure whose value is higher than the value of the "best" dependency measure *minus* the value of the "Relative-to-best threshold". Once the dependency graph has been built, the semantics of the split/join points are determined.

Step 2 - Semantics Split/Join Points For every split point, the HM algorithm calculates the "AND-strength" of this split point - say t - and two of its outgoing elements - say t_1 and t_2. The more often the log contains traces in which t is followed by *both* t_1 and t_2, and the less often it has traces in which t is followed by *only one* of the elements t_1 or t_2, the higher the probability that t has an AND-split semantics with its outgoing tasks t_1 and t_2. A similar reasoning is used for the AND-join points. Thus, only the triples (t, t_1, t_2) with an "AND-strength" higher than the parameter "AND threshold" (σ_{AND}) will have a respective AND-split/join semantics. If

this condition does not hold, the OR-split/join semantics is used.

Step 3 - Long distance dependencies In some process models the choice between two activities isn't determined inside some node in the process model but may depend on choices made in other parts of the process model (so called long distance dependency). Clearly, such non-local behavior is difficult to mine for mining approaches primarily based on direct following information (i.e. in the event log activity A is directly followed by activity B). In the HM there is a extra mining step available based on a measure for the direct or indirectly following relation. The basic idea is as follows. If this measure for two activities A and C is above the "Long distance dependency threshold" (σ_{LD}) the mined process model is checked for this dependency relation. For instance, if in the model, A causes B and B causes C there is already an (indirect) dependency relation between A and C. If this is not the case, an extra dependency relation is added to the model.

It is clear that a lot of parameters are used to influence the amount of mined details. This makes the HM useful as mining algorithm to illustrate the proposed parameters optimization framework for process discovery algorithms.

B. Running Example

The process model in Fig. 1 is used as the running example. This process model is used for generating an event log (called `Log1`). One noise free event log with 1000 random traces is generated. However, the loop from activity J back to C is low frequent (only in 17 out of 1000 cases) and the situation in which activity is D followed by K without F and H as intermediate is also low frequent (only 26 times). To incorporate noise in our event logs we define two different types of noise generating operations: (ii) remove one event, and (ii) interchange two randomly chosen events. To incorporate e.g. 5% noise, 5% of the traces from the noise free event log are randomly selected and then one of the two above described noise generating operations is applied (each noise generation operation with an equal probability). The resulting event log with 5% noise (called `Log1n5`) and the original log (i.e. `Log1`) are used to illustrate the mining behavior of the HeuristicsMiner and the optimization framework. The combination of parallelism (after activity A two parallel processes are started), loops (length-one, length-two and longer loops), hidden task, low frequent activities, and noise make this event log difficult to mine (i.e., it is a real challenge to discover the process shown in Fig. 1).

Process models in the HeuristicsMiner are so-called "Causal Matrices". Below we define the concept of a Causal Matrix. As an example, we show how the process model in Fig. 1 can be described by the causal matrix as shown in Table I. The process model in Fig. 1 has 11 activities ($A...K$) with each an input and output set. The empty input set of activity A indicates that A is a start activity. The output set $\{\{B\},\{C\}\}$ indicates that after activity A there is an AND-split to B and C. The input set $\{\{J\},\{D,H\}\}$ of activity K is a combination of an OR-join and an AND-join (J AND (D OR H)). For more details about the semantics of Causal Matrices we refer to [11].

Definition [Causal Matrix] A Causal Matrix is a tuple $CM = (A, C, I, O)$, where

- A is a finite set of activities,
- $C \subseteq A \times A$ is the causality relation,
- $I \in A \rightarrow \mathcal{P}(\mathcal{P}(A))$ is the input condition function,[1]
- $O \in A \rightarrow \mathcal{P}(\mathcal{P}(A))$ is the output condition function,

such that

- $C = \{(a_1, a_2) \in A \times A \mid a_1 \in \bigcup I(a_2)\}$,[2]
- $C = \{(a_1, a_2) \in A \times A \mid a_2 \in \bigcup O(a_1)\}$,
- $C \cup \{(a_o, a_i) \in A \times A \mid a_o \overset{C}{\bullet} = \emptyset \wedge \overset{C}{\bullet} a_i = \emptyset\}$ is a strongly connected graph. [3]

TABLE I
THE CAUSAL MATRIX FOR THE PROCESS MODEL OF FIG. 1.

ACTIVITY	*INPUT*	*OUTPUT*
A	$\emptyset$	$\{\{B\},\{C\}\}$
B	$\{\{A\}\}$	$\{\{E,D\}\}$
C	$\{\{A,J\}\}$	$\{\{I\}\}$
D	$\{\{B,G,F\}\}$	$\{\{F,K\}\}$
E	$\{\{B,G,F\}\}$	$\{\{G\}\}$
F	$\{\{D\}\}$	$\{\{E,D,H\}\}$
G	$\{\{E\}\}$	$\{\{E,D,H\}\}$
H	$\{\{G,F\}\}$	$\{\{K\}\}$
I	$\{\{C,I\}\}$	$\{\{I,J\}\}$
J	$\{\{I\}\}$	$\{\{K,C\}\}$
K	$\{\{J\},\{D,H\}\}$	$\emptyset$

Up to here, a short characterization of the HM process mining algorithm. For the details of the algorithm we refer to [17]. In the same publication we report about the mining performance of the HM over an extensive set of different event logs with different numbers of tasks, different amounts of imbalance and different amounts of noise. After performing 12.000 experiments it appears that the HM is very robust with respect to imbalance and noise. However, all these experiments are performed with the default parameter settings of the HM and without any effort to optimize them. In the next section a framework is presented that enables the search for an *optimal* process model.

II. AN ADAPTED K-FOLD-CV EXPERIMENTAL SET-UP FOR PROCESS MINING

Within the ML community there is a relative simple experimental framework called *k-fold cross validation* [12]. Starting with a ML-technique and a data set the framework is used (i) to build, for instance, an optimal classification model (i.e. with the optimal parameter settings), (ii) to report about the performance of the ML-technique on this data set, (iii) to

[1] $\mathcal{P}(A)$ denotes the power set of some set A.
[2] $\bigcup I(a_2)$ is the union of the sets in set $I(a_2)$.
[3] $a_o \overset{C}{\bullet}$ denotes the output places of a_o and $\overset{C}{\bullet} a_i$ the input places of a_i.

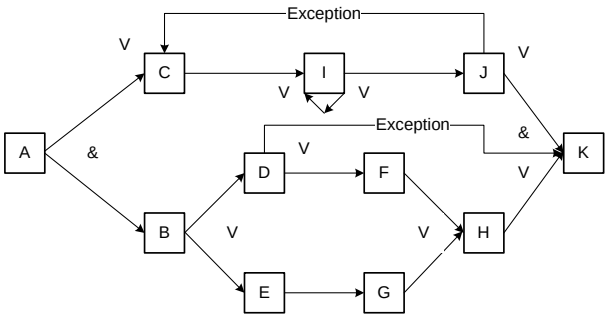


Fig. 1. The process model used for generating event logs.

estimate the performance of the definitive learned model, and (iv) to compare the performance of the ML-technique with other learning techniques. The following steps can be distinguished:

1) In the first step a series of experiments is performed to determine an optimal parameter setting for the current learning problem. The available data is divided into k subsets of roughly equal size. The ML-algorithm is trained k times. In training n, subset n is used as test material, the rest of the material is used as learning material. The performance of the ML-algorithm with a specific parameter setting is the average classification error over the k *test* sets.
2) Based on the best average performance in Step 1, the optimal parameter setting is selected. The goal of a second series of experiments is to estimate the expected classification performance of the ML-technique. The available data [4] is again divided in k subsets and again the ML-algorithm is trained k times (in combination with the parameter setting as selected in Step 1). The average classification performance on the k test sets is used to estimate the expected classification performance of the ML-technique on the current data set and the T-test is used to calculate a confidence interval.
3) If useful, a definitive model is build. All the available material is used in combination with the parameter setting as selected in Step 1. The performance results of Step 2 are used to predict the performance of the definitive model.

To compare the performance of two learning techniques for a specific data set the steps as enumerated above are performed for both techniques and the results of Step 2 in combination with a paired T-test are used to check if one of the techniques has a significant better performance. Moreover, there are well know data sets (e.g. the UCI Machine Learning Repository, CMU NNbenchmark, Proben1, StatLog, ELENA-data, etc.) that can be used for testing and comparing different techniques. In the description above, the classification error is used to measure the quality of a model. However, other measurements (i.e. the mean square error in the situation of an estimation learning task) can also be used. The attractiveness of this framework is its simplicity and the focus on the performance on *new* material. Over-fitting the learning material indicates a relatively low performance on the new material and, on the other hand, if there is no generalization, the performance on new material is also weak. In practical situations the balance between overgeneralization and over-fitting is a major challenge. However, the starting point for process discovery is an event log with, in principle, only positive examples; negative examples are not available or it is not clear which traces contain noise (impossible process behavior). This starting-point is an extra handicap during process mining; especially the combination of low frequent behavior and noise is problematic.

[4]If enough learning material is available it make sense to start step 2 with a totally new data set.

A. The adapted k-fold-cv set-up

In this subsection we explain the adapted version of k-fold-cv process mining experimental set-up. We have to deal with the following questions: (i) How to deal with the omission of negative examples? and (ii) How to measure the quality of a mined process model? Because this is a first attempt different possible options are tested.

Concerning the first question, two simple options are chosen. Starting with an event log (called `EventLog`) two event logs with negative examples are generated. In the first option, a set (`EventLogNoise`) is generated by adding some noise to the original traces of the event log. In *each trace* one event is removed or two events are interchanged. In the second option, a set (`EventLogRandom`) is generated by building m random traces of length n (where m is the number of traces in `EventLog` and n is the number of different activities in event log `EventLog`).

Concerning the second question the following measurements are used during our experiments: (i) the Parsing Measure (PM), (ii) the Continuous Parsing Measure (CPM), and (iii) the Missing/Left parsing measure (MLPM). All these metrics are based on replaying logs in process models and they mainly differ in what they consider to be the "unit of behavior". For the first metric (PM), the *trace* is the unit of behavior (i.e. the percentage of the traces that can be correct parsed by a model). For the second metric (CPM), tasks are

units of behavior. Even when a trace cannot be completely replayed by a model, the metric assesses which percentage of this trace can be correctly replayed. To do so, a non-blocking replay is used (i.e., if a parsing error occurs, it is registered but the replay continues). The last one (MLPM), is more fine-grained than the metric CPM because it also considers problems (missing and remaining activities) that happened during the log replay. Remark that only the first measure is independent of the semantic of the model representation language, the details of the other measures depend on the semantic of the representation language. This can be a disadvantage during performance comparing of mining techniques with a different representation language. In our experiments 6 different measurement combinations are tested; the test material is the *k-fold* subset in combination with one of the two negative examples sets (i.e. `EventLogNoise` or `EventLogRandom`) and one of the three performance measurements (PM, CPM, and MLPM). A possible combination is for instance the PM+noise-combination. Let us use this combination to explain some details of the adapted k-fold-cv set-up. Starting point is an event log `EventLog` with, for instance, 1000 traces. If $k = 10$ (a common used value), than in experiment n of the k experiments, 900 traces are used to induce a process model. Two parsing test are performed. The first parsing test is performed with 100 test cases of experiment n. Traces that can be parsed are interpreted as *positive*. If 91 from 100 positive test traces are correctly parsed, the positive parsing error is 100-91/100=9%. The second parsing test is performed with all 1000 traces in `EventLogNoise`. Parsing traces of this test log is interpreted as *negative*. If 142 of the 1000 traces in `EventLogNoise` are parsed, the negative parsing error is 142/1000=14.2%. The overall *PM + noise* performance error is calculated by taking the average of the normalized values (divided with the average value over all experiments in Step 1) of these two errors.

III. Experimental results

In this section we perform two adapted k-fold-cv process mining experiments. The basic material is the event log `Log1` and `Log1n5` of the running example in Fig. 1. As stated before, the combination of parallelism (after activity A two parallel processes are started), loops (length-one, length-two and longer loops), hidden task, low frequent activities, and noise make the second event log (i.e. `Log1n5`) difficult to mine. Note that of the 17 dependency connections in the original model, 15 connections are frequently used and 2 connections, not. Table II gives an overview of the 16 different parameter settings used during the experiments. Experimenting with the other parameters seems irrelevant for the current mining problem and were invariant during all experiments (i.e. the positive observations threshold $\sigma_P = 3$, all other parameters are default). Remark that the parameter setting for the first experiment (i.e. 0.99/0.99/0.99/0.005) is very intolerant for accepting extra dependency connections. The parameter setting for experiment 16 (i.e. 0.5/0.5/0.5.0.5) is very tolerant for accepting extra connections.

A. k-fold-cv experiment without noise

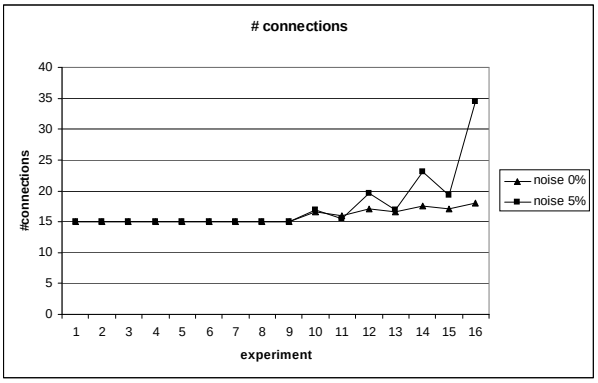


Fig. 2. The average number of connections in the 16 different series of experiments with 0% noise and 5% noise.

The results of applying Step 1 of a *10-fold-cv* experiment on `Log1` are depicted in Fig. 2 and Fig. 3. Fig. 2 shows the average number of connections during the experiments without noise (and with 5% noise). Remark that in Fig. 3 only 5 measurements are displayed in the graph. It appears that the *PM+random* combination is not very informative, because none of the random traces of `Log1random` can be parsed with any of the mined models. Moreover, it is surprising that the other 5 graphics are very similar, resulting in roughly one line. It seems that the first 90 mined models (experiments 1 to 9) are all exactly equal. The explanation is that a basic process model is build on the basis of the all-activities-connected heuristic. Because the parameters are so intolerant that no other connections are accepted. A closer inspection of the experimental results shows that all these models contain exactly the 15 frequently used connections. The models in experiments 10 (i.e. 0.80/0.80/0.80/0.20) contain 16 or 17 connections. The error measure of experiment 12, 14, 15 and 16 is minimal. All models in experiment 12 contain exactly 17 connections and appear equal to the model used for generating `Log1` (Fig. 1). The models of experiments 14, 15 and 16 contain 17 or 18 connections but hardly suggesting some form of over-fitting. A simple explanation is that in the event log `Log1` there is only low frequent behavior but is noise-free. The parameter setting as used in experiment 12 results in simple models (i.e. the minimal number of connections) with minimal performance errors. Therefore, parameter setting 12 (i.e. 0.7/0.7/0.7/0.3) is selected in Step 1 of this k-fold-cv experiment.

This concludes Step 1 of the noise free k-fold-cv experiment. Using the optimal parameters of Step 2 we can now estimate the expected quality measurements of the HM on the event log `Log1`. The available event log `Log1` is again divided in k subsets and again the HM-algorithm is trained k times (in combination with the parameter setting as selected in Step 1). The average quality values of the different measurements in combination with the different test material can now be estimated (i.e. Table III). The T-test is used to calculate 95%

TABLE II
THE 16 DIFFERENT PARAMETER COMBINATIONS DURING THE *10-fold-cv* PROCESS MINING EXPERIMENTS. $\sigma_P = 3$ IN ALL EXPERIMENTS THE OTHER PARAMETERS ARE DEFAULT.

Exp #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
σ_D	.99	.99	.98	.98	.95	.95	.90	.90	.80	.80	.70	.70	.60	.60	.50	.50
σ_{L1L}	.99	.99	.98	.98	.95	.95	.90	.90	.80	.80	.70	.70	.60	.60	.50	.50
σ_{L2L}	.99	.99	.98	.98	.95	.95	.90	.90	.80	.80	.70	.70	.60	.60	.50	.50
σ_R	.005	.01	.01	.02	.025	.05	.05	.10	.10	.20	.15	.30	.20	.40	.25	.50

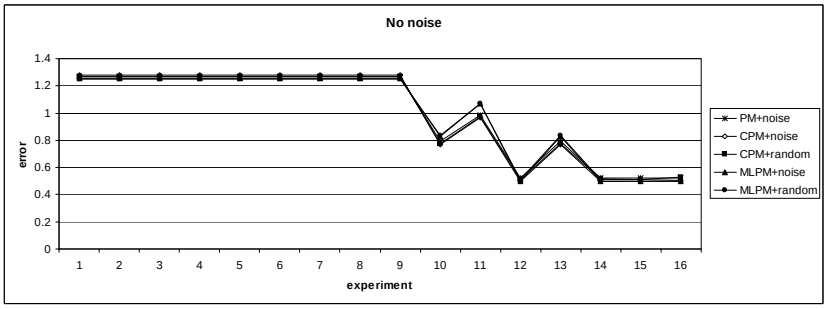


Fig. 3. The average performance errors for 16 different parameter settings, event log `Log1`, and five different measurement combinations. The results strongly indicate that the parameter settings in experiments 12, 14, 15 and 16 are optimal.

confidence intervals. To make the results easier to understand the errors for the different test sets (i.e. positive test material, EventLogNoise and EventLogRandom) are given. For instance, the value 0 and 6.50 under k-folds + MP indicates that in the situation without noise the expected trace error for new material is 0% and in the situation with with 5% noise 6.5% ($\pm$1.63). Finally Step 3 is performed, using all the available material in `Log1n5`, the parameter setting as selected in Step 1, a definitive model is mined. The results of Step 2 are used to predict the quality of this definitive process model.

B. k-fold-cv experiment with 5% noise

The results of applying Step 1 of a *10-fold-cv* experiment on `Log1n5` are depicted in Fig. 2 and Fig. 4. Again, the PM+random combination is not informative and is left out. The 5 remaining graphics in Fig. 4 are still very similar (but not as similar as the graphs in experiments without noise). Again, the first 90 mined models (experiments 1 to 9) are all exactly equal and have the 15 frequently used connections. The same explanation given for the `Log1` experiments holds for these experiments (the role of the all-activities-connected heuristic). In contrast with the results of experiments with the noise free log `Log1` in the experiments with noise (see Fig. 2) both the number of connections and the number of errors strongly increases for the experiments 13 and higher (this indicates over-fitting). It appears that both parameters settings in experiment 10 (0.8/0.8/0.8/0.2) and 13(0.6/0.6/0.6/0.2) are good candidates. All measurements performed over these two experiments are equal. That is a strong indication that all the models for both parameter settings (exp. 10 and exp. 13) are equal. However, the 10 mined models within a fold are not equal and have different number of connections (17,16,18,18,17,16,17,17,17,16). Because the average number of connections in experiment 10 is lower than in experiment 13, the parameter setting of experiment 10 is selected for use in Step 2.

This concludes Step 1 of this k-fold-cv experiment with noise. Using the optimal parameters of Step 2 we can now estimate the expected quality measurements of the HM on the event log `Log1n5` and mine a definitive model is mined (i.e. Step 3). For the results of Step 2 see Table III. It is not surprising that the 95% confidence intervals for the event log with noise (i.e. `log1n5`) are large than the intervals for the noise free log (i.e. `log1`)

IV. CONCLUSIONS AND FUTURE WORK

This paper presented an adapted k-fold-cv experimental setup for parameter optimization in the context of process mining. The method has been explained and tested in combination with the HeuristicsMiner process mining algorithm. In this first attempt different quality measurements (on the trace level, the event level, and the missing/left activities level) in combination with different methods to generate (possible) negative trace examples (based on noise or random material) are tested. It appears that the combination of the trace level and random material is not useful; no traces of the random material can be parsed. The results with the other combinations are remarkable consistent. The experimental results for the noise free log `Log1` shows that the danger of over-fitting is minimal. In the case of noise, there is the danger of over-fitting, but using the adapted k-fold-cv method it appears possible to select an optimal parameter setting.

During our experiments three parsing measures are tested: (i) the Parsing Measure (PM), (ii) the Continuous Parsing Measure (CPM), and (iii) the Missing/Left parsing measure (MLPM). Only the first measure is independent of the semantic of the model representation language, the details of the

TABLE III
THE RESULTS (I.E. ERRORS AND 95% CONFIDENCE INTERVALS) OF STEP 2 OF THE *10-fold-cv* EXPERIMENTS FOR `Log1` AND `Log1N5`. TO MAKE THE RESULTS EASIER TO UNDERSTAND THE ERRORS FOR THE DIFFERENT TEST SETS ARE GIVEN. FOR INSTANCE, THE VALUE 0 AND 6.50 UNDER k-FOLDS + PM INDICATES THAT IN THE SITUATION WITHOUT NOISE THE EXPECTED TRACE ERROR FOR NEW MATERIAL IS 0% AND IN THE SITUATION WITH 5% NOISE 6.5% ($\pm$1.63).

	k-folds			`EventLogNoise`			`EventLogRandom`		
noise 0%	PM	CPM	MLPM	PM	CPM	MLPM	PM	CPM	MLPM
Step 2	0	0	8.33	10.70	11.61	88.54	0	58.69	50.02
95% conf.int.	$\pm$0.00	$\pm$0.00	$\pm$0.08	$\pm$0.00	$\pm$0.04	$\pm$0.01	$\pm$0.00	$\pm$0.36	$\pm$0.18
noise 5%									
Step 2	6.50	0.70	09.12	10.52	11.57	88.52	0	58.17	50.95
95% conf.int.	$\pm$1.63	$\pm$0.23	$\pm$0.32	$\pm$0.05	$\pm$0.16	$\pm$0.33	$\pm$0.00	$\pm$0.66	$\pm$0.93

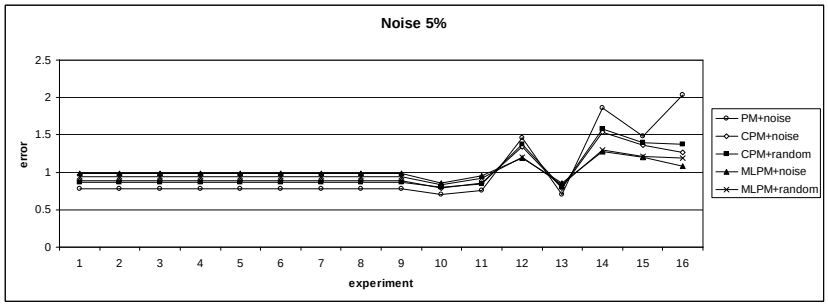


Fig. 4. The average performance errors for 16 different parameter settings, event log `Log1n5`, and five different measurement combinations. The results strongly indicate that the parameter settings in experiments 10 and 12 are optimal.

other measures depend on the semantic of the representation language. It appears that the simple combination of yes/no parsing of a trace (i.e. measure (i)) in combination with negative examples based on noise is sufficient for parameter optimization and performance comparison. This makes it possible to compare control-flow mining techniques with different underlying process representation languages.

In the near future we plan to test the method on more data sets and in combination with different process mining algorithms. If the method turns out to be robust, we will add a corresponding plug-in to the next release of ProM. In combination with appropriate process mining benchmark material this will enable researchers to assess the quality of their process mining techniques and compare their performance with existing techniques.

REFERENCES

[1] W.M.P. van der Aalst, H.A. Reijers, A.J.M.M. Weijters, B.F. van Dongen, A.K. Alves de Medeiros, M. Song, and H.M.W. Verbeek. Business Process Mining: An Industrial Application. *Information Systems*, 32(5):713–732, 2007.

[2] W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. Workflow Mining: A Survey of Issues and Approaches. *Data and Knowledge Engineering*, 47(2):237–267, 2003.

[3] R. Agrawal, D. Gunopulos, and F. Leymann. Mining Process Models from Workflow Logs. In *Sixth International Conference on Extending Database Technology*, pages 469–483, 1998.

[4] D. Angluin and C.H. Smith. Inductive Inference: Theory and Methods. *Computing Surveys*, 15(3):237–269, 1983.

[5] J.E. Cook and A.L. Wolf. Discovering Models of Software Processes from Event-Based Data. *ACM Transactions on Software Engineering and Methodology*, 7(3):215–249, 1998.

[6] A.K. Alves de Medeiros. *Genetic Process Mining*. PhD thesis, Eindhoven, University of Technology, Eindhoven, The Netherlands, 2006.

[7] B. van Dongen, A.K. Alves de Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters, and W.M.P. van der Aalst. The ProM framework: A New Era in Process Mining Tool Support. In G. Ciardo and P. Darondeau, editors, *Application and Theory of Petri Nets 2005*, volume 3536 of *Lecture Notes in Computer Science*, pages 444–454. Springer-Verlag, Berlin, 2005.

[8] E.M. Gold. Language Identfication in the Limit. *Information and Control*, 10(5):447–474, 1967.

[9] E.M. Gold. Complexity of Automaton Identification from Given Data. *Information and Control*, 37(3):302–320, 1978.

[10] J. Herbst. *Ein induktiver Ansatz zur Akquisition und Adaption von Workflow-Modellen*. PhD thesis, Universitat Ulm, 2001.

[11] A.K.A. de Medeiros, A.J.M.M. Weijters, and W.M.P. van der Aalst. Using Genetic Algorithms to Mine Process Models: Representation, Operators and Results. BETA Working Paper Series, WP 124, Eindhoven University of Technology, Eindhoven, 2004.

[12] T.M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[13] R. Parekh and V. Honavar. Automata Induction, Grammar Inference, and Language Acquisition. In Dale, Moisl, and Somers, editors, *Handbook of Natural Language Processing*. New York: Marcel Dekker, 2000.

[14] L. Pitt. Inductive Inference, DFAs, and Computational Complexity. In K.P. Jantke, editor, *Proceedings of International Workshop on Analogical and Inductive Inference (AII)*, volume 397 of *Lecture Notes in Computer Science*, pages 18–44. Springer-Verlag, Berlin, 1889.

[15] A. Rozinat. *Process Mining: Conformance and Extension*. PhD thesis, Eindhoven University of Technology, Eindhoven, 2010.

[16] A. Rozinat, A.K. Alves de Medeiros, C.W. Gnther, A.J.M.M Weijters, and W.M.P. van der Aalst. The need for a process mining evaluation framework in research and practice. In H.Y. Paik A.H.M. ter Hofstede, B. Benatallah, editor, *Business Process Management Workshops (BPM 2007)*, volume 4928 of *Lecture Notes in Computer Science*, pages 84–89. Springer-Verlag, Berlin, 2008.

[17] A.J.M.M. Weijters, W.M.P. van der Aalst, and A.K. Alves de Medeiros. Process Mining with the HeuristicsMiner-algorithm. BETA Working Paper Series, WP 166, Eindhoven University of Technology, Eindhoven, 2006.

FACIAL NERVE STREAM TRAJECTORY DATA MODELLING AND VISUALIZATION

Jalel Akaichi, Hanen Bouali, Zeineb Dhouioui
Department of Computer Science
ISG-University of Tunis
41, Rue de la Liberté, Cité Bouchoucha
Le Bardo 2000
Tunisia
{Jalel.Akaichi, Hanen.Bouali, Zeineb.Dhouioui}@isg.rnu.tn

Abstract—Bell's palsy is the paralysis of facial muscles caused by perturbations affecting the facial nerve. It is the origin of a physical suffering, and has an emotional and psychological impact on patients. Treatments of Bell's palsy are still not well-defined, nevertheless physical therapies techniques, such as facial exercise, biofeedback, laser, electrotherapy, massage and thermotherapy, are used to speed up the recovery. The main disadvantage of used techniques is the absence of a clear and concise modelling of manipulated data resulting from treatments performed on various patients by a range of physicians. This makes the exchange and the large-scale exploitation of these data difficult and complex. The objective of this work is to supervise the states evolution of patients affected by facial paralysis leading to recovery. Modelling facial nerve stream trajectory data seems to be an essential step leading to perform our purpose. In fact, it permits representation unification and facilitates data querying in order to ensure recovery surveillance and the disease understanding. Moreover a visualization algorithm is proposed to track facial nerve stream in order to observe recovery progress and to identify the occurrence of conduction problem preventing it.

Keywords: Facial nerve stream, moving objects, trajectory data, visualization.

I. INTRODUCTION

Peripheral facial paralysis (PFP), also called Bell's palsy or idiopathic paralysis is the result of the motor neuron lesion. More than sixty percent of PFP results from virally induced inflammation of the peripheral facial nerve (cranial nerve VII) [1]. Some clinical symptoms Bell's palsy disease are the droop of the mouth, the inability to frown, raise the eyebrow, close the eye and show teeth or whistle, the pain in or behind the ear, the increase of sensitivity to sound, etc. As further symptom, patient can loose the taste in the anterior two-thirds of the tongue due to the lesion of the sensory ganglion of the facial nerve and can be numbed in the affected side of his face. It can be also accompanied with involuntary movements of the facial muscles like muscles twitching or with voluntary movements like twitching of the eyelid with voluntary movement of the lips. These last symptoms can be considerate as a recovery progress [2].

Various techniques were proposed to assist physicians in treating and understanding the disease. Some of them are subjective such as the guidelines and the grading systems. The gold standard for grading facial nerve function is the House-Brackmann grading system [3] witch was proposed in 1985. Moreover new tests exist; the electrical facial nerve ones can be performed to diagnose the paralysis' gravity. Most of used techniques are subjective and others qualified to be more objective are proposed to undergo observed drawbacks (subjectivity of judgment and insensitivity to regional difference of function in the different part of the face). Several new scales [4] of various degrees of objectivity and ease of use, including systems based on computer analysis [5], [6], and moiré photography [7], have been introduced to undergo the above drawbacks. Recently, He and colleagues [8] proposed an automated, objective and reliable facial grading system in order to assess the degree of movement in the different regions of the face. Moreover, the most common techniques today are electroneurography, electromyography, the nerve excitability test, and the maximum stimulation test [14].

Whatever objective or subjective the main disadvantage of used techniques is the absence of a clear and concise modeling of manipulated data resulting from treatments performed on various patients by a range of physicians working in a variety of institutions. This makes the exchange, the querying, and the large-scale exploitation of these data difficult and complex.

The objective of this work is to supervise the states evolution of patients affected by facial paralysis leading to recovery. Modeling facial nerve stream trajectory data seems to be an essential step leading to perform our purpose. Moreover, visualizing the facial stream nerve trajectories may help physicians to understand deeply the disease through comparisons performed on patient state in time, or between different patient states.

This paper is structured as follows. In section 2, we describe related work. In section 3, we propose the facial nerve modeling. In section 4, we put forward a visualization algorithm. Finally, we present conclusion and future works.

II. STATE OF THE ART

The diagnostic approach of PFP consists on seeking, by clinical examination and complementary explorations, arguments in favor of different etiologies, the nature of the disease affecting a patient. In [9], authors introduced the main steps for the PFP's clinical identification. It consists on an interrogation and a clinical examination. The interrogation attempts to clarify the patient's medical history (i.e. neurological) by identifying facts that could interfere with treatment (diabetes), tracing the occurrence of some suggestive cases (i.e. otitis), tacking the profile of the disease development (i. e. brutal, progressive, etc.) and the prodromal signs (i. e. ear ache, watery eyes, etc.), and finally seek the existence of signs of neurological localization.

To diagnose the gravity of the PFP and evaluate the patient's state, there exist facial nerve paralysis evaluation systems [3], [10-11]. The gold standard for grading facial nerve functioning is the House-Brackmann grading system [3] which was proposed in 1985. It consists on a classification used mainly for monitoring the medium and long term patients. They are graded from I (normal) to VI (total paralysis). Those grades are obtained by asking the patient to perform some movements and then the physician assigns a grade according to his clinical observation and subjective judgment.

Later on, in 1994, Murty and colleagues [11] proposed the Nottingham system for measuring facial nerve functioning. It's based on measuring the movements of four points of the face and comparing the abnormal side with the normal side by computing a percentage.

Due to the drawbacks of these classifications, mostly related to the subjectivity of the physicians and caregivers judgment, objective and quantitative evaluation systems were developed. Murata and colleagues [12] developed an evaluation system using computer analysis and rating scores. This computer system included a video and infrared cameras with markers placed on the face to record movements. Somia and colleagues [13] demonstrated how computerized eyelid motion analysis can quantify the human reflex blink by using a video camera incorporated into a helmet. The quantification of the kinematic parameters of eyelid closure has been carried out using reflective markers on normal subjects and following facial nerve paralysis. Sargent and colleagues [5] adopted a desktop computer software to objectively grade facial movement based on the Nottingham system. They used the computer software to subtract digitized images and derive a facial movement score. The facial grading system proposed by He and colleagues [8] is based on videotaping the patient performing some facial movements like raising eyebrows and smiling. Following that, an optical flow is calculated to identify the direction and amount of movement between image sequences. The optical flow computation results are used to measure the symmetry of the facial movements between each side of the face.

III. FACIAL NERVE MODELING

Understanding the facial nerve anatomy (figure 1) is essential to reach the main objective of our work. Indeed, facial nerve can be subdivided into two main components: motor components and sensory components. Essential motor components are the branchial motor (or special visceral) that efferent supplies the muscles of facial expression, and visceral motor (or the general visceral) that vehicles the parasympathetic innervations to all glands of head (i. e., lacrimal, sublingual and submandibular glands) except the parotid. Sensory components are located in the ear, in the tympanic membrane, and in the anterior two-thirds of tongue.

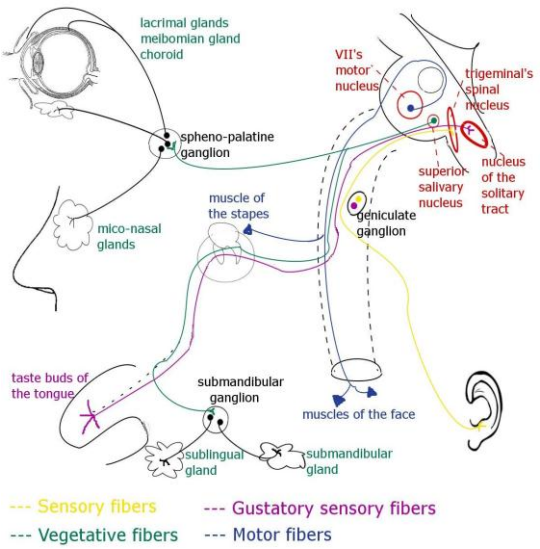


Figure 1: Facial nerve anatomy

We modelled, using UML class diagram (figure 2), the above two main components of the facial nerve: the motor components (branchial and visceral) and the sensory components (general and special). In figure 3, we describe in details the branchial motor components which are composed by the facial nerve portion and the facial muscles. We subdivide the facial muscles into three classes: the posterior muscles, the superior half muscles and the lower half muscles. The superior half

muscles are the frontal muscle (FM), the eyebrow muscles (EyM) and the eyelid muscles (EM).

The lower half muscles are the ear muscles, the nose muscle (NsM), the nostril muscle (NoM), the tongue muscle (TM), the mouth muscles, the depressor muscles, the risorius muscle[1] (RM), the neck muscle (NeM), the cheek muscles and finally the chin muscle (ChM). The ear muscles are composed by the auricular muscle (PrAM), the higher auricular muscle (HAM) and the posterior auricular muscle (PoAM). The mouth muscles are the lower lip muscle (LLM) and the upper lip muscle (ULM). The depressor muscles are the mouth's angle depressor muscle (MAD) and the lower lip depressor muscle (LLD). The cheek muscles are the zygomatic muscle[2] and the whelk muscle (WM). And finally the Zygomatic muscles are the major zygomatic muscle [3] (MZM) and minor[4] zygomatic muscle[5] (SZM).

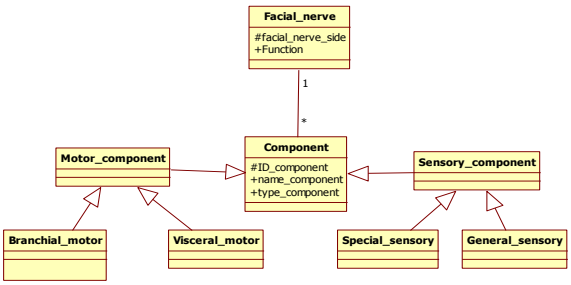


Figure 2: Facial nerve class diagram

Recall that the facial nerve stream is considered as a moving object which circulates through a defined "network", the facial nerve in our case of studies. In figure 4, we simplified the ramications of the facial nerve, where their end-points innerve the facial muscles. For each ramification, we place an intersection point to acquire a new portion.

We modelled the facial nerve ramifications using an oriented graph (figure 5). For the branchial motor components, nodes represent intersection points or muscles and arcs represent the connections between them.

The start node of the graph describes the beginning of the facial nerve of one side (the left side or the right side) of the face, the end-nodes represent facial muscles, and arcs describes connections between nodes.

The choice of the oriented graph is explained by the fact that the facial nerve stream is unidirectional. Then, we identify each portion with its start-point and its end-point. We can generate the measures of amplitude and frequency of the stream nerve using electrical tests (e. g. electrodes) applied each portion. We can measure also the intensity of the facial nerve stream crossing one specific portion. An intensity superior to zero means that the stream nerve is crossing a portion, otherwise is not.

Figure 3: Branchial motor component class diagram

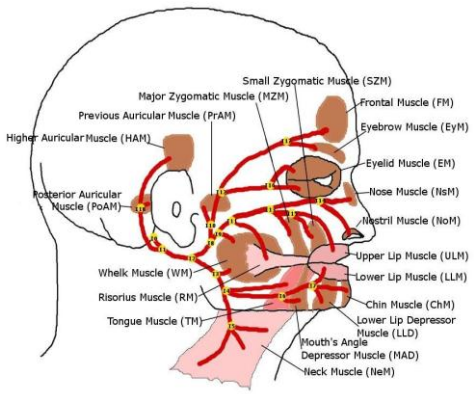


Figure 4: Facial nerve map

[1] The Risorius muscle is the muscle of the commissure of the lips used to express laugh.
[2] Cheek's transverse muscle which contracts when smiling
[3] To elevate corners of mouth
[4] We replace minor by small in order to distinguish between major zygomatic muscle and minor zygomatic muscle.
[5] To elevate upper lip

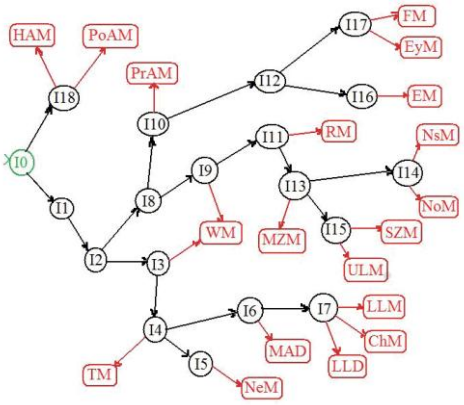


Figure 5: Facial nerve graph

IV. VISUALIZATION ALGORITHM

To follow the evolution of the Bell's palsy patient's state, the electromyography (EMG) is used to detect the electrical muscle activity, to amplify it, to make it audible and displayable on a monitor. There are two main techniques of electromyography [15]. The first one record the potential for muscle individually analyzed using a needle electrode inserted into the muscle. The second one, used in the BFEMG (biofeedback-EMG) studies and kinesiology, records all the electrical activity of muscle using electrodes attached to the external skin. Whatever used technique, collected muscles responses (table 1) are stored into a database to be exploited to be used by our visualization algorithm called FAcial Nerve Stream Visualization or FAN (figure 6).

The FAN algorithm takes as input the graph G representing the facial nerve graph (figure 5) and the muscles responses recoded in (table 1), and produces a colored graph ColoredG indicating the trajectories of the facial nerve stream.

We define a function called "Stream" which performs the main following steps:

- Looks, into the table 1, for muscles which do not respond normally to the excitation provoked by electrodes. Those are characterized by a score, computed according to amplitude and intensity, inferior to a defined threshold. Later on, the above function calls for a another one called Color_trajectory(End_node, I0, red) which colors in red the trajectory starting from the End_node linked to a not responding muscle until the graph starting node I0 indicating that the facial nerve stream do not cross this trajectory.
- Looks, into the table 1, for muscles which respond normally to the excitation provoked by electrodes. Those are characterized by a score, computed also according to amplitude and intensity, superior to a defined threshold. Later on, the above function calls also for Color_trajectory(End_node, I0, green) which colors in green the trajectory starting from the End_node linked to the considered muscle until the graph starting node I0 demonstrating that the facial nerve stream cross this trajectory.

Muscle Name	Score	Threshold
1. I18_HAM	2	2
2. I18_PoAM	3	2
3. I3_WM	4	1
4. I4_TM	3	3
5. I5_NEM	2	1
6. I6_MAD	1	2
7. I7_LLD	2	2
8. …	…	…

Table 1. Facial nerve response recording sheet

Note that, some portions of the graph may be colored using both colors. This indicates to analysts that the facial nerve stream is cut somewhere between an end node and a node in the graph. This pushes them to perform further deep analysis. Steps 1 and 2 can performed periodically to assess the patients states evolution through a methodic comparisons.

```
Algorithm FAN;
    Input: G, table 1;
    Output: ColoredG
Begin
  Stream(table1, table11, table12);
  For i:=0 to size(table11) do
Color_trajectory(i, I0, red);
  For i:=0 to size(table12)
Color_trajectory(End_node, I0, green);
End.
```

Figure 6. Visualization algorithm

Examples

Table 2 describes a full recovery of a patient. All facial nerve stream trajectories are colored in green. This means that stream amplitudes and intensity are acceptable, and consequently all stream scores reached the required thresholds.

Table 3 describes a partial recovery of a patient. Indeed, some facial nerve stream trajectories are colored in red. This means that stream amplitudes and intensity are not acceptable or null, and consequently all stream scores did not reach the required thresholds. The stream does not affect positively the muscle MAD through the partial trajectory (I4, I6), the muscles LLM, ChM, and LLD through the partial trajectory (I4, I6, I7), the muscles EM through the partial trajectory (I12, I16), the muscle RM through the partial trajectory (I9, I11), and the muscles ULM and SZM through the partial trajectory (I13, I15).

Stream Trajectory	Color
I0 I1 I2 I3 I4 I5	Green
I0 I1 I2 I3 I4 I6	Green
I0 I1 I2 I3 I4 I6 I7	Green
I0 I1 I2 I8 I10	Green
I0 I1 I2 I8 I10 I12 I16	Green
I0 I1 I2 I8 I10 I12 I17	Green
I0 I1 I2 I8 I9 I11	Green
I0 I1 I2 I8 I9 I11 I13 I14	Green
I0 I1 I2 I8 I9 I11 I13 I15	Green
I0 I18	Green

Table 2. A full recovery

Stream Trajectory	Color
I0 I1 I2 I3 I4 I5	Green
I4 I6	Red
I4 I6 I7	Red
I0 I1 I2 I8 I10	Green
I12 I16	Red
I0 I1 I2 I8 I10 I12 17	Green
I9 I11	Red
I0 I1 I2 I8 I9 I11 I13 I14	Green
I13 I15	Red
I0 I18	Green

Table 3. A partial recovery

CONCLUSION AND FUTURE WORK

Bell's palsy disease is the origin of a physical suffering, and has an emotional and psychological impact on patients. To contribute in the improvements of treatments and analysis automation, we proposed clear and concise modeling, based on UML class diagrams, of facial nerve and some of its essential components. We also represent it using a graph leading to track facial nerve dream and to determine by the way patients' recovery progress and eventual conduction problems to be solved thanks to observation. This latter is ensured thanks to a simple but convenient visualization algorithm. Future works will focus on integrating manipulated data resulting from treatments performed on various patients by a range of physicians in a various health care institutions. This, obviously, will enhance analysis and large-scale exploitation of these data which is difficult and complex.

REFERENCES

[1] Medical policy: Electrical stimulation for the treatment of facial palsy. HealthLink.Availableonline: http://www.healthlink.com/provider/medpolicy/policies/MED/elect_stim_facial_palsy.html. Accessed March 23rd, 2009.

[2] H. J. Diels, "Facial paralysis: is there a role for a therapist?", Facial Plastic Surgery, vol. 16, no. 4, 2000, pp. 361-374.

[3] J. W. House, D. E. Brackmann, "Facial nerve grading system," Otolaryngol. Head Neck Surg., 1985, pp. 146–147.

[4] T. S. Kang, J. T. Vrabec, N. Giddings, Terris DJ, "Facial nerve grading systems (1985–2002): beyond the House–Brackmann scale," Otol Neurol, vol. 23, 2002, pp. 767–771.

[5] E. W. Sargent, O. Fadhi, R. S. Cohen, "Measurement of facial movement with computer software," Arch Otolargyngol Head Neck Surg, vol. 124, 1998, pp. 313–318.

[6] A. Bajaj-Luthra, T. Mueller, P. C. Johnson, "Quantitative analysis of facial motion components: anatomic and non-anatomic motion in normal persons and in patients with complete facial paralysis". Plastic Reconstr Surg, vol. 99, 1997, pp. 1894–1902.

[7] K. Yuen, I. Inokuchi, M. Maeta, S. Kawakai, "Evaluation of facial palsy by moiré topography index," Otolargyngol Head Neck Surg, 1997, pp. 567–572.

[8] H. Shu, J. Soraghan, and B. F. O'Reilly, "Biomedical Image Sequence Analysis with Application to Automatic Quantitative Assessment of Facial Paralysis", 2007.

[9] F. Tankéré, I. Bernat, "Paralysie faciale à frigoré : de l'étiologie virale à la réalité diagnostique," Rev Med Interne, 2009.

[10] B. G. Ross, G. Fradet, J. M. Nedzelski, "Development of a sensitive clinical facial grading system," Otolaryngol Head Neck Surg, vol. 114, no. 3, 1996, pp. 380-386.

[11] G. E. Murty, J. P. Diver, P. J. Kelly, G. M. O'Donoghue, P. J. Bradley, "The Nottingham System: objective assessment of facial nerve function in the clinic," Otolaryngol Head Neck Surg, vol. 110, no. 2, 1994, pp. 156–161.

[12] K. Murata, M. Isono, K. Saito, H. Miyashita, "Quantitative analysis of synkinesis following facial nerve palsy", 2003.

[13] N. N. Somia, G. S. Rash, E. E. Epstein, M. Wachowiak, M. J. Sundine, R. W. Stremel, J. H. Barker, "A computer analysis of reflex eyelid motion in normal subjects and in facial neuropathy," Clin Biomech, vol. 15, no.10, 2000, pp. 766-771.

[14] http://www.utsouthwestern.edu/utsw/cda/dept28151/files/289976.html. Accessed March 23rd, 2009.

[15] http://www.sante.cc/electro/dossiers/biofeedback/emgg/bfb02.htm#Diagnostic%20EMGBF. Accessed March 23rd, 2009.

Domain Specific Services for Continuous Diagnoses in the Context of Ambient Assisted Living – AAL

Bjoern-Helge Busch[1] **and Ralph Welge**[1]
[1]Institute VauST, Leuphana University Lueneburg, Lueneburg, Lower Saxony, Germany

Abstract - *This article broaches the issue of a human centered assistance system which implements a continuous health monitoring system based on situation recognition. Concerning the initiating background of Ambient Assisted Living, the assistance system, addressing the domain of geriatric home care, evaluates distributed sensor-, actuator- and communicator networks to afford situation dependent services for people with special demands. After an introduction to the field of research under consideration of essential exigencies, the architectural concept of the middleware and their coherent functional modules are described. Thereby, the focus lies on the preprocessing techniques and the probabilistic modeling through Hidden Markov Model, which are a sufficient solution for the delineation of discrete event systems. Mapping the medical problem of heart insufficiency (HI), we gain an ostensive instance for the demonstration of our approach and obtain and present first results based on real and simulated data for the demonstration and validation of the system.*

Keywords: probabilistic, telemedicine, assistance systems

1 Introduction

The momentary observable demographic trend in Germany leads straightforward to crucial personal, organizational and financial burdens for NGOs and governmental institutions in the sector of geriatric care. Against this background, assistance systems are an appropriate remedy to master the coherent aftermath. In respect to the intended user group of senior citizens, the proposed assistance system has to implement several, outstanding features as health monitoring and an adaptive user integration. Hence, this contribution introduces actual deployments in the context of AAL in section 1.1 and elucidates – concerning the claimed preventive diagnosis functionality and medical monitoring – current issues in the field of telemedicine in section 1.2. In chapter 2, our own approach is described with the focus on the architectural concept of the software, the classification of the addressed domain and the employed techniques for the situation recognition including activity analysis and the coherent vital sign interpretation. In chapter 3, the general mode of operation is unveiled through a simulation which regards real data from our living lab, an involved medical data base and gathered multi-parametric information from polysomnography studies. Finally, after a discussion of the experimental results, the conclusion and the next steps in the research project close this article.

1.1 Ambient intelligence and AAL

The emerging area of research indicated by the acronym *AAL – Ambient Assisted Living* – comprehends concepts, products and services for the unobtrusive support of elderly people with special requirements comprising questions of comfort, safety and health maintenance. Many research projects were launched, which address the domain of ambient assistance with miscellaneous concepts for the recognition of life patterns or respectively *ADLs – Activities of Daily Life*. For example, the multinational research project *AMIGO* picked out the development of a scalable, open and interoperable middleware for the fusion of multiple data sources as home automation components, consumer electronics, mobile devices and personal computing in form of a home networking infrastructure to gain information and obtain context data for user's benefit [1]. An alternative approach combines information/ communication infrastructures of smart buildings and telemedicine devices in order to provide social and medical services within the familiar environment of the user [2]. Closed meshed maintenance servicing through ambient technologies, linked communication services for external institutions as hospitals and a tight supervision by medical or custodial attendants is the purpose. The project *SmartSenior* is also focused on the demands of elderly people and the facilitation of their daily life. It is not restricted to home care scenarios but deals also with mobile, ubiquitous services just such as emergency detection within a car including vital sign transmission. Expanded automatic location services are used to increase the impartial and subjective safety of the user [3]. The project Ambience [4] concentrates on the development of system architectures for context aware environments relating to home- and professional indoor domains under inclusion of natural interaction, adaptivity to user behavioral, identification and tracking. Our proposal, the research project *AAL@Home* is engaged in the design and implementation of a user centered assistance system, which originates from the mergence of signal processing and time series analysis of environmental process data, gathered by distributed sensor networks, as the precondition for probabilistic modeling of user situations [5].

1.2 Current issues in the area of telemedicine

The prevalence and incidence of cardiovascular diseases like heart arrhythmia or cerebral insults increased significantly in the recent years due to improved health care standards and the decreased lethality after critical heart events [6]. In this context, heart insufficiency (HI) takes an exposed position as a cost driver in the public health sector regarding the rate of oc-

currence and the high number of consecutive hospitals stays after decompensation [7]. Therefore, we decided to design a reliable prediagnosis system to initiate preventative interventions as emergency calls or the application of appropriate preparations. This requires a robust continuous vital sign monitoring which is also essential for the cooperative compliance control (CCC), a functionality which is indispensable for applications of geriatric care. Studies with modern telemedical devices are proving an increased rate of detection of cardiovascular diseases, but they fail in the area of domestic care by external institutions [8]. Unfortunately, applying body attached sensor systems reveal numerous disadvantages, because they restrict the user in his mobility and autonomy. Therefore, telemedical ECG recordings are restricted to the checkup of arrhythmia and applied to technologies as cardiac pacemakers, ICDs and event-recorder [9]. Alternative concepts as camera based approaches lack also confidence and acceptance because the user may reject or have reservations towards the system; this solution implies an impression of a residential observation. Thus, it is mandatory to use contactless measurement methods for the data acquisition, if there is a claim to achieve an auspicious alternative to established systems or a complementing technology to them.

2 Methodological approach

The assistance system as a pervasive construct adheres to a generic topology, a layer concept which is independent from the addressed domain. For more detail refer to previous publications [10-11].

2.1 Architectural concept

The different layers of the assistance system, classified by the degree of gathered information and functionality (refer to fig. 1), are implemented as domain specific services within an *OSGi*-platform. The port to a unified middleware for AAL-solutions as the instantly establishing *universAAL* [12] is discussed. Like a residential gateway for UPnP-devices and handhelds with standardized communication protocols, services and data formats for supreme interoperability, the assistance system, running on a set-top box (STB), subsumes the heterogeneous sensor-actuator-communicator networks as an organic system, integrating seamlessly within the users familiar environment. Bundles of OSGi-services, addressing for example the layer of data acquisition, information fusion or semantic integration (for the representation of the explicit and implicit knowledge about the domain and ascertained context data we use *OWL-DL*-ontologies), are exchangeable and tailored to the selected domain resp. the specified problem. Context aware agents, attaching the probabilistic situation recognition, provide resource sharing for medical data processing, prediagnosis and knowledge transfer, integrate a preventive emergency detection concerning an escalation chain, instantiate communication services for kin people, custodians, doctors by viewphone and connect with acquaintances in the context of social networking in a local residential community for elderly persons. In addition, perception of environmental data and the offer of natural cooperative, transparent interfaces considering aspects of usability for the multimodal and simultaneous communication are also key features to concern.

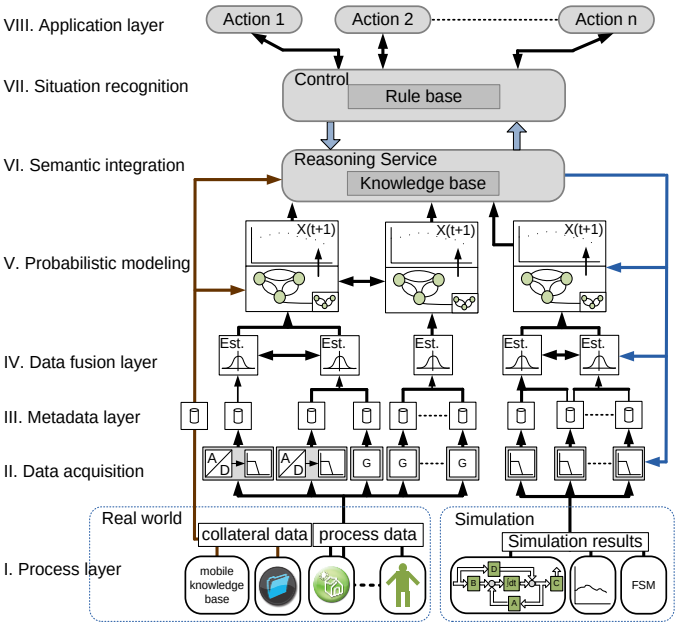


Figure 1: Architectural approach of the assistance system

Due to the limited space we concentrate only on the classification of the process environment/mock habitat and the utilized techniques for the estimation of the state of health, depicted with the detection of heart insufficiency and recommend referenced publications for further information.

2.2 Process environment – living lab

2.2.1 Utilized ambient sensor networks

The assistance system is based on the evaluation of embedded distributed sensor networks, home automation components, telemedical solutions and mobile devices called smart nodes (MSN) capturing individual related information as a mobile knowledge base.

UWB-Radar	Home automation	Telemedicine	Collateral data sources
Respiratory rate (RR)	HCI events	Blood pressure	Mobile knowledge base (User preferences)
Heart rate (HR)	System states	Heart Rate	Parameter files for system calibration
Tidal volume (TV)	Embedded sensors (temperature etc.)	Blood glucose	Probabilistic health models
Position	Automation events	Tidal volume	Trend models
Posture	Control output	Body weight	Digital health record (HL7)
Agility		Body temperature	PhysioNet-database
Material coefficients		ECG (heart events etc.)	Polysomnography study results

Table 1 : Data sources of the living lab and their features

Table 1 recapitulates the actual topology of our mock habitat resp. living lab addressing the typical dwelling of a senior including the desired process values for situation recognition. In respect to interoperability and considering facets as indivi-

dual system reconfiguration, adaption to user preferences and interconnectivity, these components are self-describing, utilizing an electronic type plate and are therefore interchangeable with an intelligent management for device discovery. Actually, we are using *BACnet* and *XML-RPC* to access two different installed home automation systems to derive HCI-events, automation events, system states and sensor data. Other interfaces such as *LON* or *EIB* will be integrated soon by descriptive meta-files for compatibility with third party smart home devices. Gathered raw data is stored in a data set in form of an n-tuple.

To avoid *Body Sensor Networks – BSNs* – we use UWB-radar (ultra wideband) for the unobtrusive acquisition of user related values. Contactless measurement procedures for the acquisition of health parameters are essential for user acceptance. Hence, we prefer the usage of a m-sequence radar system for the measurement of respiratory rate, heart rate and the detection of position and posture as proposed by [13-15]. The sensor devices comes with a bandwidth of 3.9 GHz, needs one transmitting antenna and two receiving aerials for heart rate detection. One sensor network is integrated within an armchair and another within a test bed. For the measurement of respiratory signals over a distance of nine meters, the mounting of antennas within the habitat at walls is sufficient. The measurement of oscillatory vital signs relates to the mapping of bioelectrical values to the superposition of mechanical quantities, caused by the expansion of the heart muscle and the upheaval of the thorax through the expanding lung. In addition to the frequency, we extract the tidal volume of the patient to indicate adumbrating edema – tidal volume sinks in this case.

Finally, a lot of data is gained through telemedical devices which are used autonomously at deterministic points of time due to the anamnesis of the patient. These devices use the Bluetooth interface and trigger the assistance system automatically if they are used; the data is collected and stored in a SQL-database by OSGi-services and evaluated through techniques related to signal-/data processing for feature extraction or to be precise, emission detection.

2.2.2 Feature extraction

The data acquisition deals with many totally different values regarding the resolution and dynamics of the respective signal. Some of the process values rely to equispaced intervals of measurement; others are bound to irregular sampling which we have to countervail by the aid of reconstruction methods as spline interpolation, polynomials or Gaussian relations.

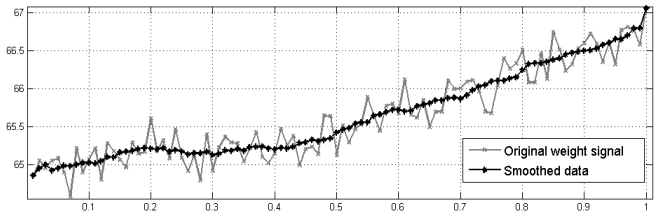


Figure 2 : Raw data - Weight

This section specifies the process of gathering raw data, their smoothing and feature extraction. Human body weight for example, is an important parameter to examine for the detection of edema, a most significant indicator for an exacerbating stadium of heart insufficiency – the impending decompensation. Body weight is also a widely varying parameter which depends on anthropometric traits, individual for each user.

Khosla and W.Z. Billewicz analyzed a lot of measuring campaigns with different persons with different gender and age, and determined characteristic behaviors due to activities and habits [16]. Pregnancy, for instance, has obviously a significant curve and this is also applied for other situations or health states. To derive expressive features from the original unstable signal, we prefer to use a Savitzky-Golay filter with a varying span of interpolation points regarding following expression

$$y_k^* = \frac{1}{f_{Norm}} \sum_{j=m}^{m} C_j \cdot y_{k+j^*} \qquad (1)$$

for smoothing the raw data (refer to fig. 2). This modus operandi preserves us features of the distribution such as relative maxima, minima and width, which are usually flattened by other adjacent averaging techniques as the ordinary moving average. This is the main reason, why we decided to use this technique for smoothing others, much more dynamic process values as vital signs detected through UWB-sensor devices. Table 2 lists a subset of features which are observed.

UWB-Radar	Telemedicine	HCI-interaction and behavioral monitoring
RR-variability	Δ(Ø weight)	Categorized symptoms as
HR-variability	HR-variability	a) dizziness
Δ(ØTV)	RR-variability	b) qualmishness
Examined agility	Δ(ØTV)	c) headache…
Predicted HR	Δ(Ø mm/Hg)syst.	d) Shortness of breath
Predicted RR	Δ(Ø θBody)	Change in behavior as
Predicted TV	ECG-arrhythmia	Altered sleeping habits…

Table 2: Set of evaluated features for emission detection

For the feature extraction procedure we had to define threshold values adaptive to every person we want to supervise. After a calibration process in order to gain standard values like sigma and mean for each parameter, in this case we can classify the normalcy of the weight signal and distinguish between normal, noticeable, significant and exposed progress of weight due to medical advisory service by a physician. These boundaries are relative to the body mass and checked up to the differentiated filter response due to the simple relationship

$$\frac{\Delta y_n^*}{(n-m)} = \frac{y_n^* - y_{n-m}^*}{(n-m)}; n \in N_+; y_n^* \in R \cdot \qquad (2)$$

In addition to the extraction of the actual emission for the probabilistic modeling, we execute a time series analysis with a set of basic functions for trend detection. The trend analysis enables us to predict initiating values of weight for the early detection of emerging edema or significant mass increase in general. At the moment, this mechanism is only used for safeguarding if probabilistic modeling of health states fails. If the designed stochastic models are much more mature, this proce-

dure will hopefully be unnecessary and will be used for another owing approach which is not to be discussed here at the moment.

2.3 Probabilistic modeling – HMMs

Since Rabiner consolidated the theory of Hidden Markov Models and their adaption to selected fields of application [17], this technique for modeling stochastic processes gained more importance and got enhanced by the work of many scientists. HMMs are completely described by the quintuple

$$\lambda = \{A, B, X, Y, \pi\}, A = \{a_{ij}\}, B = \{b_j(k)\} \quad (3)$$

with the state space and the alphabet

$$X = \{x_1, x_2, x_3, ...x_{N1}\} Y = \{y_1, y_2, y_3, ...y_{N2}\}, N1, N2 \in N_+ . \quad (4)$$

Considering the state transition probability distribution $\{a_{ij}\} = P(q_{t+1} = x_j \mid q_t = x_i)$ and the relationship of emitted symbols $\{b_j(k)\} = P(o_t = y_j \mid q_t = x_i, \lambda)$, the probability of state sequences $Q = \{q_1, ...\}$ relating to a sequence of observables $O = \{o_1, ...\}$ can be computed through

$$P(O \mid \lambda) = \sum_{n=1}^{|X|^{|Q|}} \prod_{t=1}^{|Q|} P(o_t = yj \mid q_t = x_j, \lambda) \cdot P(Q_n \mid \lambda). \quad (5)$$

Their robustness enables HMMs as an appropriate solution to a wide area of technical problems. For example, they are used for Attack Intention Prediction – AIP in the context of computer security by [18] with standard techniques and by [19] with advanced Fuzzy HMMs (FHMM). The identification of resident people within a test habitat through HMMs is part of the work of [20]. Forecasting the grid load is solved by a proposal of [21] and another nice approach deals with the prediction of space weather focused on appearing sunspot areas [22]. Methodical approaches deal with deterministic initialization of HMMs [23] in the context of human activity recognition while [24] use/parameterize coupled HMMs (CHMM) for the approximation of posture and activity. The training of HMMs of second order through multiple observation sequences is part of the work of [25]. We use Hidden Markov Models to estimate user /environmental situations based on the extracted features alluded in the previous sections and assess vital signs under consideration of the most likely situation to approximate possible states of health. For the sake of clearness, we will discuss this in detail in the following sections through a vivid example.

3 Experiments – Emergency detection

For the interpretation of vital signs, it is most important to estimate the most likely activity of the user; spans of heart rate and respiratory rate are obviously varying due to the respective situation.

3.1 Basic activity recognition

The process environment respectively the mock habitat is equipped with a lot of embedded sensor nodes as mentioned before and consists of seven rooms, representing the typical home of a senior citizen. The activity recognition model is implemented as a hierarchical stochastic automaton. The lowermost subsystem (refer to fig. 3) attends to the approximation of the patient's location within his dwelling. Therefore we designed a first basic HMM

$$\lambda_{\text{Pos}} = \{A_{Pos}, B_{Pos}, X_{Pos}, Y_{Pos}, \pi_{Pos}\}, \lambda_{Pos} \in \lambda_{Sit} \text{ with} \quad (6)$$

$$X_{\text{Pos}} = \{\text{bedroom, study, floor, storage room, bathroom, living room, kitchen}\} \quad (7)$$

For the estimation of the most likely position, we use emissions based on the extracted features from the home automation which produce binary output and UWB-sensor systems. Statistical blur follows from the sensor characteristics and insufficient coverage by UWB-devices for localization. Figure 3 depicts the HMM λ_{Pos} mapping the footprint of our living lab. State *x* marks the transition to an additional model for tracking; leaving the habitat for example.

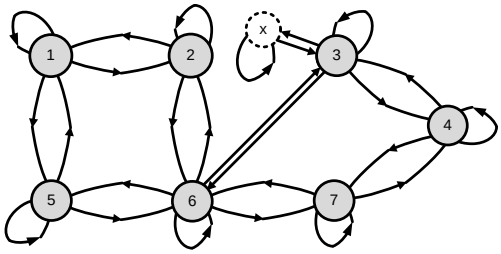


Figure 3: HMM-Model for position tracking

For convenience, we denote door contacts with d_x, window contacts with w_x, and motion sensors with m_x and obtain an alphabet

$$Y_{Pos} = \{w_{11}, ...w_{72}, m_{31}, d_{31}...d_{76}, \varepsilon\} \quad (8)$$

with indices corresponding to their placement. For example, a door contact d_{23} is the third sensor element in room two and m_{12} corresponds with the second motion sensor in the bedroom. The surrogating symbol ε is used if no concrete emission caused by a sensor is detected. As an emission itself each room/location corresponds to a partial HMM-graph consisting of a set of certain activities, typically for the particular location (refer to fig. 4).

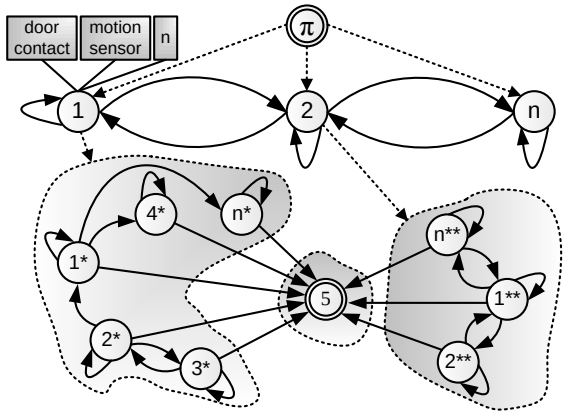


Figure 4: HMM-Model activity recognition

The situation recognition is restricted to a single person – the prospective senior citizen – and is predicated on the successive analysis of emission and state sequences. Every single sequence corresponds to a certain model/activity, approximated

continuously through the evaluation of extracted features by usage of the Viterbi-algorithm as explained by [17] in much more detail.

$$\delta_{t(j)} = \max_{q1,q2,...q_{|Q|}} P(X_t, X_{t+1},...X_{To}, O_t...O_{To} \mid \lambda) \quad (9)$$

$$\delta_1(i) = \pi_i b_i(k = o_i), 1 \le i \le |X|, \Psi_1(i) = 0 \quad (10)$$

$$\delta_{t(j)} = \max_{1 \le i \le |X|} [\delta_{t-1}(i) a_{ij}] b_j(o_t), \quad (11)$$

$$\Psi_t(j) = \arg\max_{1 \le i \le |X|} [\delta_{t-1}(i) a_{ij}] \quad (12)$$

Computing the most likely path through the trellis structure, we are able to select the best fitting model for the situation recognition. The different models relate to real measurement data, collected over an evaluation interval about two weeks. We are using the Baum-Welch algorithm [17] to train the HMMs even under consideration of their temporal validity. This means, every data track produces one model, valid only for a certain time interval T. The length of T is limited to half an hour. Therefore, we gain

$$\lambda_{Sit} = \{\lambda_{(t=0:00)} ... \lambda_{(t=24:00)} \text{ with} \quad (13)$$

$$\{\{a_{ij}\}_{nT}\}, \{\{b_j(k)\}_{nT}\} \quad (14)$$

One part of the training data was used for the estimation of the model parameters stored in A and B, the remainder was used for validation of the models for testing. Online situation recognition is still in progress.

3.2 Heart insufficiency approximation

In respect to the detected user situation, vital sign threshold values are selected (refer to figure 5). The boundary values relate to an anthropometric survey of the user resp. the subject who evaluates the mock environment for testing. As mentioned in section 2.2.2 for the example of mass increase, the boundaries for vital signs are relative to the users' average characteristics. The mean and sigma of the different vital signs shift synchronous with the physical abilities and exposures. Concerning our introductory background of cardiovascular diseases, we have to model the risk of an acute heart insufficiency by a stochastic automaton also.

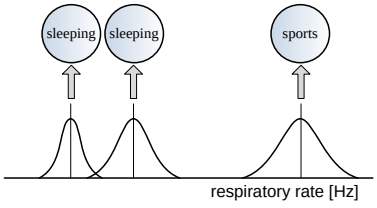


Figure 5: Activity sensitive parameter selection

In addition, we use the results of the time series analysis to control our probabilistic prediction mechanism. Due to the anamnesis, heart insufficiency can be classified into four typical stages if we follow the New York Heart Association. These stages (NYHA I – NYHA IV) care about the physical impairments due to the disease. In our model, we use these stages for the adjustment of the transition and emission matrices in accordance to real vital sign recordings from HI- sufferers from clinical observations in order to understand the dependencies between them. Polysomnography studies and the PhysioNet-database provide a lot of data to train and adjust the model. Cardinal symptoms of heart insufficiency are the emergence of edema, Cheynes-Stokes respiratory (nearly 30-40% of the patients), a noise of bubbling while gasping, shortness of breath and others. Many of these symptoms can be measured indirectly through other observations. A significant mass increase, for example, suggests the emergence of an edema in the lung. Decreasing amplitude of breath including a gain of breath's frequency suggests a sinking volume of inhaled air which corresponds with a fluid retention within the lung also. The detection mechanism of HI-risk is one partial HMM in the continuous prediagnosis module; preventive interventions are initiated if any dangerous situation or change of the health state is recognized or respectively presumed.

4 Discussion of results

Due to the high density of implemented sensor nodes, we achieved a high accuracy for the localization of the user within the mock environment (refer table 3).

Sensor density	Average localization rate
Basic (+Actuators & Sensors of the home automation)	71.63 %
Enhanced (+motion sensors)	85.74 %
Premium (+UWB-radar)	99.34 %

Table 3: Localization accuracy

Utilizing UWB-radar components placed at convenient locations covering transit areas within the home provides us a reliable tracking system. But sensor density is also a financial aspect to concern and therefore important for the assessment of the approach. The activity recognition works also fine, but scales obviously with the number of utilized data sources, too. In addition to the sensors and actuators, we evaluate electronic consumers also through the home automation.

Detection window	Average activity recognition rate
0:00 – 2:00	99.89 %
2:00 – 4:00	99.82 %
4:00 – 6:00	99.78 %
6:00 – 8:00	87.57%
8:00 – 10:00	85.21%
10:00 – 12:00	84.17%
12:00 – 14:00	82.29%
14:00 – 16:00	82.76%
16:00 – 18:00	84.93%
18:00 – 20:00	84.79%
20:00 – 22:00	97.34%
22:00 – 24:00	98.45%

Table 4: Activity detection accuracy

Controlled electrical sockets enable the determination of power consumption and hence, the identification of the used de-

vices. The recognition of activities varies during the day (refer to table 4). Activities like sleeping or reading in the evening or in the night are regularly always recognized. The estimation of the user situation depends on the highest degree of sensor density – the position tracking reaches in this case an accuracy of 99.34%. During night and in the evening, the situation recognition reaches a high precision. This is a direct result of the state reduced models for this detection window and furthermore a consequence following from the embedded sensor networks – UWB components are integrated within the bed and monitor the sleeping or reading person continuously at this time. In spite of the observable error rate regarding the activity recognition in the daytime, the selection of the valid boundary values for the vital sign evaluation fits the requirements, because the different states or activities which are detected by mistake are very similar concerning their physical properties. Boundary selection suits well in 98% of the detected situations. The identification of critical health states achieved a high accuracy. Considering the simulation results, every harmful situation was recognized, 8% of the initiated alerts were false alarms which leaves enough room for improvement. Focusing on the medical problem of HI, the detection of a significant mass increase achieved nearly 98%, but the interpretation of the mass increase lacks confidence because only a small subset of diagnoses was evaluated. This leaves obviously too much room for interpretation in real world scenarios and field trials and needs to be adjusted by the expansion of the state space for continuous diagnosis estimation module.

5 Conclusion and outlook

The identification of a deteriorating state of health is one essential task an assistance system in the context of AAL has to regard. Therefore, we introduced a novel approach based on probabilistic modeling through Hidden Markov Models for the approximation of user situations in order to interpret vital signs correctly; only the knowledge about the actual user activity provides a convenient selection of threshold values for emission detection for a trustworthy pre-diagnosis. In spite of the prototypical state of the embedded sensor nodes, signal processing and feature extraction from UWB-data work fine and promise success for the development of successors for the following field trials within geriatric care institutions. This approach accomplishes the issue for contactless, unobtrusive vital sign acquisition in a satisfactory way. With this framework for gathering environmental information – telemedical devices included - , the first results of the probabilistic modeling show a high degree of reliability when detecting activities – in analogy to other similar approaches – and reveal much potential for the early detection of crucial stadiums of health; in this concrete case adulterating heart insufficiency which was detected credibly in the simulation. This conclusion leads directly to the next steps in research: the improvement of the actual concept and the validation through a field trial, executed within a medical randomized study with, of course, patients with different stages of heart insufficiency. Additional, we want to improve our sensor network with capacitive carpets for the detection of influencing force components. This allows us to the determination of a person's movement without expensive radar components and the recognition of the posture of a laying person for data fusion; correcting measured vital sign amplitudes

6 References

[1] AMIGO. (2011, March). [Online]. Available: http://www.hitech-projects.com /euprojects/amigo/amigo.htm

[2] WOHNSELBST. (2011, March). [Online]. Available: http://www.wohnselbst.de/

[3] SMARTSENIOR. (2011, March). [Online]. Available: http://www1.smart-senior.de

[4] AMBIENCE. (2011, March). [Online]. Available: http://www.hitech-projects.com/euprojects/ambience/

[5] AAL@HOME. (2011, March). [Online]. Available: http://www.leuphana. de /institute/vaust/forschung-projekte /aal-ambient-assisted-living.html

[6] L. Gerhard. "Suppression von paroxysmalem Vorhofflimmern durch bifokale, rechtsatriale Schrittmacherstimulation". Berlin, Charité, Univ.-Med., Dissertation. 2005

[7] J. Klauber et al. "Krankenhaus -Report 2010– Krankenhausversorgung in der Krise ?". Schattauer, F.K., ISBN 3794527267, 2010

[8] M. Hördt et al. "Differentialdiagnose und Dokumentation tachykarder Rhythmusstörungen". Herzmedizin. Vol. 20. No. 3, pp. 146-152, 2003

[9] E. Kouidi et al. "Transtelephonic electrocardiagraphic monitoring for an outpatient cardiac rehabilitation programme". Clinical Rehabilitation, 2006

[10] B.H. Busch et al. "Architecture of an adaptive, human-centered assistance system". Proceedings of the 2010 International Conference on Artificial Intelligence (ICAI'10). Las Vegas, USA, pp. 691-696, ISBN 1-60132-146-5, 2010

[11] B.H. Busch et al."Preventive emergency detection based on the probabilistic evaluation of distributed, embedded sensor networks". Ambient Assisted Living: 4. AAL-Kongress 2011 Berlin, Germany, January 25-26, 2011, Springer, Berlin; 1.st Edition.(Januar 2011), ISBN-13 978-3642181665, 2011.

[12] UNIVERSAAL (2011, March). [Online]. Available: http://www.universaal.org/

[13] J. Sachs et al. "M-Sequence Hardware for UWB-Imaging: Current state and future goals". UKoLoS-Colloquium on Localisation and Imaging with UWB-sensor-technology, Georgenthal, Germany, 2007

[14] J. Sachs et al. "Through-Wall Radar". Proceedings of the IRS 2005. Berlin, Germany, 2005

[15] F. Thiel et al. "Fusion of magnetic resonance imaging and ultra-widebandradar for biomedical applications". Proceedings of the 2008 IEEE International Conference on Ultra-Wideband (ICUWB2008), 2008

[16] T. Khosla et al. "Measurement of change in body-weight". British Journal of Nutrition Vol. 18, pp. 227-239. 1964

[17] R. Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". Proceedings of the IEEE, Vol. 77, No. 2, pp. 257–286, 1989

[18] X. Zan et al. "A Hidden Markov Model based frame-work for tracking and predicting of attack intention". Pro-ceedings of the IEEE International Conference on Multimedia Information Networking and Security, 2009

[19] Y. Li et al. "Intrusion detection method based on Fuzzy Hidden Markov Model". Proceedings of the IEEE – Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009

[20] Crandall et al. "Using a Hidden Markov Model for Resident Identification". Proceedings of the IEEE – Sixth International Conference on Intelligent Environments, 2010

[21] D. Niu et al. "Mid-long Term Load Forecasting Using Hidden Markov Model". Proceedings of the IEEE - Third International Symposium on Intelligent Information Technology Application, 2009

[22] M. Al Omari et al. "Next-Day prediction of sunspots area and McIntosh Classification using Hidden Markov Models". Proceedings of the IEEE – International Conference on CyberWorlds, 2009

[23] Moghaddam et al. "Deterministic Initialization of Hidden Markov Models for Human Action Recognition", Proceedings of the IEEE – Digital Image Computing: Techniques and Applications, 2009

[24] E. Guenterberg et al. "A Distributed Hidden Markov Model for Fine-grained Annotation in Body Sensor Networks". Proceedings of the IEEE - Sixth International Workshop on Wearable and Implantable Body Sensor Networks, 2009

[25] D. Shiping et al. "Training Second-Order Hidden Markov Models with Multiple Observation Sequences", Proceedings of the IEEE - International Forum on Computer Science-Technology and Applications, 2009

Position of Gateway Drugs in the Spectrum of Adolescent Drug-Use Initiation in Indiana: Relevance of Market Basket Analysis of Data Mining in Detection of Common Substance-Use Initiation Sequences

Ahmed H. YoussefAgha*(PhD), Wasantha P. Jayawardene*†(MD)

* Department of Applied Health Science, School of Health, Physical Education, and Recreation, Indiana University Bloomington, Indiana, United States; † Corresponding Author

Abstract

Background: *Empirical evidence confirms that the majority of individuals' legal or illicit drug use begins after use of cigarettes, alcohol, or cannabis (marijuana) hence they are known as the gateway drugs.*

Objectives: *To study the sequences of initiating use of gateway drugs and other drugs among adolescents, especially among 12th-grade students, in Indiana.*

Methods: *Data from the 1993-2009 Annual Surveys of Alcohol, Tobacco, and Other Drug Use by Indiana Children and Adolescents was used for the current study. Market Basket Analysis was used to identify sequences and association among drug- use.*

Results: *The majority of adolescents initiated drug use with gateway drugs followed by combining them with prescription drugs.*

Conclusion: *Market Basket Analysis reveals that the gateway theory is applicable to Indiana adolescents because many of them initiate drug use with alcohol, tobacco, or marijuana or with a combination of them followed by use of prescription drugs.*

Keywords: Gateway drugs, Prescription drugs, Market basket analysis; Drug-use initiation

1 Introduction

Empirical evidence confirms that the majority of individuals' illicit drug use starts only after they use cigarettes, alcohol, or cannabis (marijuana). Because using these less deleterious drugs may lead to more dangerous hard drugs, these three substances are known as the gateway drugs [1]. The gateway pattern of drug initiation describes a normative sequence that begins with alcohol and tobacco use followed by cannabis use and then other illicit drugs [2]. In the United States also, this progression proceeds from the use of tobacco or alcohol to the use of marijuana and other illicit drugs. Kandel presented the first systematic discussion of the gateway hypothesis [3]. He explored the hypothesis from various perspectives ranging from developmental social psychology to prevention and intervention science, animal models, neurobiology and analytical methodology. Although this pattern is recognized, some evidence suggests that prevention strategies focusing on breaking this chain to chronic and severe drug abuse may be of limited value [4]. Considering the prevalence of these three substances among the U.S. adolescent population, the gateway drugs should still be given maximum attention when compared to all other substances that can cause abuse and/or dependence [5].

Some studies revealed that weekly alcohol consumption came after marijuana use and preceded all other illicit drug use for Hispanic, White, and Black youth, but followed use of hard drugs for Asians [6]. There was evidence supporting that both Hispanic ethnicity and gender was significant in determining adolescent polydrug use. In the first year of middle school, gender moderated the effect of Hispanic ethnicity on lifetime polydrug use, and more serious levels of polydrug use were observed in the second year of middle school [7].

According to data from a recent national survey, nonsmokers who were also nondrinkers were most likely to indicate they would never use drugs in the future; users of both substances were most likely to indicate they would use drugs. And while smokers and drinkers were most likely to indicate expected illegal substance use in the future, those who only smoke were more likely than those who only drink to indicate probable use of such substances [8]. Adolescents were not likely to use marijuana without first using alcohol and tobacco, and they will not use more serious drugs, such as cocaine and heroin, without first using marijuana [9, 10]. Some research revealed that Whites, both male and female, had a significantly stronger positive association between predictors and drug use when compared to Blacks, while White females were most likely to use gateway substances, followed by White males [11].

Positive alcohol and cannabis outcome expectancies were meaningfully related to expectancies of future substance use, and positive cannabis outcome expectancies increased, while negative cannabis outcome expectancies decreased, with increasing frequency of alcohol use [12]. In another research, it was revealed that alcohol intake has a positive correlation with drug abuse among smokers [13]. A South African study showed that the pattern of onset of drug-use was similar across genders. Black South African adolescents first used either alcohol or tobacco, followed by both, and then dagga (cannabis) [14]. A prospective study among male Japanese students found that those who had smoking experiences by 16 years of age were more likely to abuse alcohol by 22 years of age than those without smoking experiences, but females did not show such a correlation [15]. In another study in Japan, cigarette smoking in high school students showed a strong relationship with solvent inhalation, while drinking alcohol depended on the presence of adults [16].

Particularly strong associations were found between smoking and having many friends who smoke. Using marijuana was strongly correlated with having friends or older siblings who use marijuana. The self-esteem scale score was not correlated with substance use. Anomie and antisocial behavior were more strongly associated with substance use than depression was. These associations were stronger for females than for males. Therefore, when designing and implementing preventive interventions in schools, it is essential to simultaneously address the use of gateway drugs and deviant behavior with environmental risk factors [17]. Current prevention programs are often ineffective when substance use is normalized by schools, community, and family. In such situations, positive adult role models can deter use, and focus group discussions can strengthen the development of tailored prevention programs [18].

Between U.S. and European students, there were the similar trends in drug use of gateway drugs already in use by early adolescence and higher rates of drug use among males than among females. However, cross-national differences in cannabis use revealed that U.S. pre-adolescents were 25 times more likely to use it than those in Europe. In the same study, Arizona Latinos reported higher cannabis use than Arizona non-Latinos, which was probably influenced by their acculturation level [19]. Low physical activity is associated with cigarette smoking and marijuana use. For alcohol consumption, significant links were found to race/ethnicity or sex, suggesting that socio-cultural factors may affect the relationships between physical activity and substance use [20]. Data from the Youth Risk Behavior Survey revealed that those using cigarettes before age 13 were 3.3 times more likely to have used marijuana than those who had never smoked. And users of alcohol were 4.5 times more likely to have used marijuana than those who had never used alcohol. The same survey showed that youth using marijuana before the age of 14 were 7.4 times more likely to have used other drugs even after controlling for selected other risk and protective behaviors [21].

The use of both tobacco and illicit drugs appears to be strongly associated with alcohol use, which is more prevalent, and the risk of smoking and illicit drug use is particularly high in adolescents who report high levels of drunkenness [22]. Factors which tend to increase the probability of alcohol use by adolescents include the fact that their friends drink, their awareness of the risks associated with alcohol use, and ease in obtaining alcohol. Hence, the typical adolescent who uses alcohol seems to be a risk-taker who may enjoy the dangers surrounding alcohol use and have alcohol-using friends. Religion, gender, race, academic performance, and extracurricular school activities are not directly related to the use of alcohol [23].

In 2006, marijuana (cannabis) was the most commonly used illicit drug (14.8 million past-month users) among persons in the United States aged 12 or older Marijuana users were twice as likely to use illicit drugs as young adults than non-users. Shared environmental factors mediated much of the relationship between adolescent marijuana use and young adult drug use [24]. Individuals who used cannabis by age 17 had higher rates of other drug use, alcohol dependence, and drug abuse or dependence. The association may arise from the effects of the peer and social context within which cannabis is obtained and used. In particular, early access and use of cannabis may reduce perceived barriers against the use of other illegal drugs and provide more access to these drugs [25].

Although the gateway effects of marijuana exist, they are not required to explain the common-factor model of initiating illicit drug use [26]. Extensive research does suggest that marijuana use tends to precede the use of other illicit substances among adolescents; however, there is a simultaneous argument that the correlation between marijuana use and other drug use may be spurious, which reflects the influence of one or more confounding variables that can concurrently cause both behaviors. Even after adjusting for the influence of variables, National Youth Survey results confirm the hypothesis that marijuana use exerts a causal influence on one's probability of using other illicit substances [27].

2 Purpose of the Study

This study aims to study the sequences of initiating use of gateway and other drugs among adolescents, especially among 12th-grade students, in Indiana. The study also explores the applicability of method "Market Basket Analysis" of data mining in detection of common drug-use initiation patterns in the community.

3 Methods

3.1 Study Design

Data from the 1993 to 2009 Annual Surveys of Alcohol, Tobacco, and Other Drug Use by Indiana Children and Adolescents—a survey conducted by the Indiana

Prevention Resource Center—was used for the current study [1]. The original survey's purpose was to provide data for state and local planning in respect to alcohol, tobacco, and other drug use, gambling behaviors, and risk and protective factors.

3.2 Study Population

Surveys were administered through local school officials to students in grades 6 through 12 in 557 schools throughout Indiana. Participation in the survey was voluntary. In 2009, a total of 202,091 youth (182,496 usable surveys) from both public and private schools completed it [1]. The current research sample was composed of 12th-graders. They are of particular interest because they are on the verge of entering college or the workforce. Their substance abuse habits can negatively impact their academic and career functioning, which has detrimental effects on society.

3.3 Instrument

A closed-ended, self-administered written questionnaire that was anonymous and voluntary asked adolescents about their use of twenty different types of drugs or drug categories, including prescription medications. When necessary, examples and descriptions were provided next to the drug's name or its classification.

3.4 Analysis

In 1980s, Yamaguchi and Kandel [28] utilized the log linear model to study associations and sequences of drug use. Their model was applied to study sequences up to five drugs; alcohol, cigarette, marijuana, cocaine, and heroin. This model is selected for our research to study the preferences of using drugs.

Annual prevalence, the percentage who report using a particular drug at least once in the past year, was used for the current study. Annual prevalence in abuse of prescription drugs, alcohol, tobacco, marijuana, cocaine, and heroin, including the self-reported age at first-time use of each drug, was obtained from the survey [1]. Prescription drug use among 12th-graders was identified using a sample of 260,183 students during the 3-year period of 2007-2009. For alcohol, cigarettes, and marijuana use among 12th-graders during the same time period, a sample of 246,793 students was utilized.

Adolescents were asked, in which grade he or she started using each drug. Based on this information, the chronological order of drugs use initiation was established. This study focuses on data derived from a single item of the survey. Both items were used to collect information about 11 different drugs or drug categories: alcohol, cigarettes, marijuana, OTC substances, tranquilizers, narcotics, Ritalin, cocaine, crack, steroids, and heroin. The question asks "At what age did you first use drug A" with the response options: 10 or younger, 11, 12, 13, 14, 15, 16, 17, and 18 or older.

Data obtained from the second question was analyzed using a derivative of Market Basket Analysis (MBA), a model utilized in studying patterns of goods sales (here they are drugs) focusing associations and sequences. In general, Market Basket Analysis is utilized in business management and marketing to understand how particular items were bought in conjunction with other items. MBA is one of the data mining models under SAS Enterprise Minor. The term "market basket" means a basket of items purchased in a transaction. The purpose of such analyses typically is to find subsets of items that are purchased together in many transactions [29].

We used a modified version of MBA to determine sequences of drug-use initiation by suggesting that individuals initiate use (equivalent to purchase) of specific drugs from among a larger set of available drugs (equivalent to market). Specifically, we drew from the overall sample those adolescents who report ever having used a gateway substance. We used SAS 9.3 to identify the most frequent sequences of drug use initiation among those individuals by ordering the respondents' age of first use for each substance.

Analyses focused on the "confidence %" statistic, which represents the probability, that event "Y" occurred after event "X." If 500 people abuse drug "X," and then, in a subsequent year, 125 of those people abuse drug "Y," the confidence for X→Y is 25%. For example, the sequence "Alcohol → Cocaine" is assessed using a rule that identifies all alcohol users, and then determines, how many initiated cocaine use in a year subsequent to their initiation of alcohol use.

4 Results

Of the 492,957 students who provided data on drug use, 45.3% used at least one category of drugs in the past year, and 44.8% of the total used non-prescription drugs in the past year. But, only 34.4% of the total used non-prescription drugs alone and 10.4% used them simultaneously with prescription drugs. Of the 48,227 twelfth-graders who provided data on drug use, 33,961 (70.4%) used non-prescription drugs and 25,652 (75.5%) of non-prescription drug-users used only gateway drugs, while 22.1% used gateway drugs with prescription medications and 2% used gateway drugs with hard drugs (heroin, cocaine, and crack). Of the 12th-graders, 19.3% used alcohol only; 8.4% used cigarettes only and 4.8% used marijuana only (Table 1).

More males compared to females used prescription and nonprescription drugs simultaneously. In addition, more males compared to females used at least one category of drugs, either prescription or non-prescription. Both of these findings are unexpected differences ($p<0.0001$). Similarly, more Whites compared to other races used prescription and nonprescription drugs simultaneously. In addition, more Whites compared to other races used at least one category of drugs, either prescription or non-prescription. Both of these findings are also unexpected differences ($p<0.0001$).

Drug User Category (In Grade 12)	Use Non-Prescription Drugs	Don't Use Non-Prescr Drugs	Total
Use Prescr. Drugs	9,174 (19.0%)	104 (0.2%)	9,278 (19.2%)
Don't Use Prescr. Drugs	26,070 (54.1%)	12,879 (26.7%)	38,949 (80.8%)
Total	35,244 (73.1%)	12,983 (26.9%)	48,227 (100.0%)

Table 1: Self-Reported Annual Use of Prescription and Non-Prescription Drugs Among Indiana Adolescents in Grade 12 During the Period 2007-2009.

Gateway drug use and the use of prescription and over-the-counter (OTC) medications have strong relationships (Figure 1). Also evident is that there are always two groups of drugs representing two chains. The first group contains alcohol, cigarettes, marijuana, cocaine, and crack. The second group consists of psychostimulants, CNS depressants, narcotics, and OTC drugs (Figure 2). After adjusting for age, gender, and race, similar sequences of using gateway drugs followed by hard drugs and prescription drugs were observed (Figure 2). But analysis of normative beliefs revealed that a majority of peers and parents of adolescents do not pay attention to prevention of tobacco, alcohol, prescription drug, and cocaine use, although considerable attention is paid to marijuana use [1].

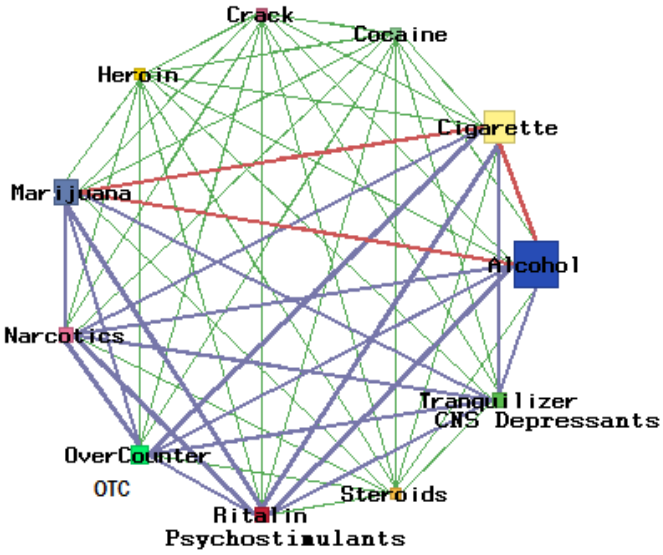


Figure 1: Relationships and Strength Between Use of Alcohol, Cigarettes, and Marijuana as Gateway Drugs and the Use of Psychostimulants, Narcotics, CNS Depressants, or OTC Drugs Among Indiana Adolescents (2007-2009) According to Market Basket Analysis. Thickness of the Lines is Proportionate to Strength of Association.

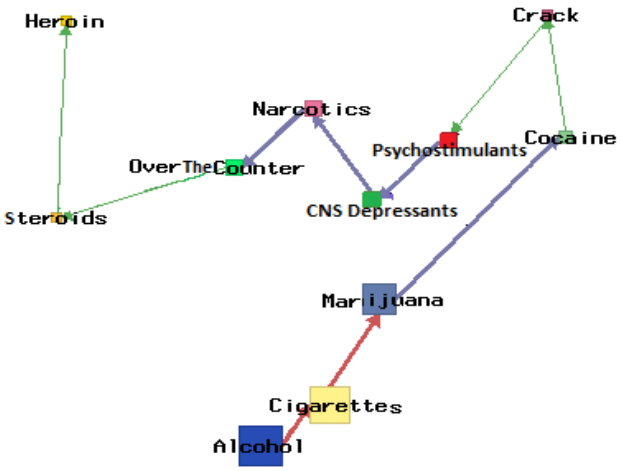


Figure 2: Sequence and Strength Between Use of Non-Prescription Drugs (Alcohol, Cigarettes, Marijuana, Cocaine, & Crack) and the Nonmedical Use of Medications (Psychostimulants, Narcotics, CNS Depressants, & OTC Drugs) Among Indiana Adolescents (2007-2009) According to Market Basket Analysis. Thickness of the Lines is Proportionate to Strength of Association.

5 Discussion

Slightly less than half of Indiana adolescents use at least one category of prescription or non-prescription drugs, and almost 99% of these drug users report annual use of non-prescription drugs, including alcohol, cigarettes, and marijuana. This indicates that almost all prescription drug users simultaneously use non-prescription drugs as well. This pattern intensifies when 12th-graders are considered because almost three quarters of these adolescents use some kind of drug.

In general, males are more prone to drug use and non-prescription drug use than females as males tend to report combined use of gateway drugs with other drugs, such as prescription drugs and hard drugs. The tendency of males to experiment may be the reason for this trend.

Market Basket Analysis reveals that the gateway theory is applicable to Indiana adolescents because many of them initiate drug use with alcohol, tobacco, or marijuana or with a combination of them followed by use of prescription drugs. The majority of adolescents who use gateway drugs start using psychostimulants non-medically, follow them with CNS depressants, and then use prescription narcotics and OTC drugs. This model proves that the gateway theory is useful in explaining adolescent drug abuse. But many parents and peers of adolescents do not pay attention to the prevention of tobacco, alcohol, prescription drug, and cocaine use, although considerable attention is paid to marijuana use.

In 2002, Yamaguchi and Kandel [28] applied two statistical models for analysis of drug use progression:

quasi-independence model for the analysis of events sequence data [30] and the log linear model for analysis of parametric events sequences. Our future studies will focus on how the outputs of Market Basket Analysis can be integrated with the models of Yamaguchi and Kandel.

6 Limitations

This is a self-administered anonymous survey on drug use. The responses of younger adolescents may not be very accurate. The percentage of missing data in some sections of the questionnaire is somewhat high.

7 Conclusion

Of the 48,227 twelfth-graders who provided data on their drug use, 70.4% use non-prescription drugs and 75.5% of those non-prescription drug-users abuse gateway drugs only while 22.1% use gateway drugs with prescription medications and 2% use gateway drugs with hard drugs. The majority of adolescents start drug use with gateway drugs followed by combining them with prescription drugs. Males, Whites, and older adolescents are more likely to report single-use of gateway drugs and combined use of gateway drugs with other drugs. Market basket analysis in data mining is found to be a useful technique in identifying common substance-use initiation patterns.

8 Conflict of Interests Statement

This research is not fully or partially supported by any funding source and authors have no conflict of interests.

9 Acknowledgement

The authors would like to thank Prof. Mohammad Torabi and Dr. Ruth Gasman of Dept. of Applied Health Science, Indiana University Bloomington for their help.

10 References

[1] "Survey of Alcohol, Tobacco, and Other Drugs use by Indiana Children and Adolescents. Indiana Prevention Resource Center," 2010.

[2] L. Degenhardt, *et al.*, "Does the 'gateway' matter? Associations between the order of drug use initiation and the development of drug dependence in the National Comorbidity Study Replication," *Psychological Medicine,* vol. 39, pp. 157-167, 2009.

[3] D. B. Kandel, *Stages and Pathways of Drug Involvement: Examining the Gateway Hypothesis* Cambridge: Cambridge University Press, 2002.

[4] M. E. MackesyAmiti, *et al.*, "Sequence of drug use among serious drug users: Typical vs atypical progression," *Drug and Alcohol Dependence,* vol. 45, pp. 185-196, 1997.

[5] (2010). *Monitoring the Future Survey: A Continuing Study of American Youth*. Available: http://monitoringthefuture.org/

[6] P. L. Ellickson, *et al.*, "Stepping Through the Drug-Use Sequence - Longitudinal Scalogram Analysis of Initiation and Regular Use," *Journal of Abnormal Psychology,* vol. 101, pp. 441-451, 1992.

[7] J. A. Epstein, *et al.*, "Gateway polydrug use among Puerto Rican and Dominican adolescents residing in New York city: The moderating role of gender," *Journal of Child & Adolescent Substance Abuse,* vol. 12, pp. 33-46, 2002.

[8] P. B. Johnson, *et al.*, "The relationship between adolescent smoking and drinking and likelihood estimates of illicit drug use," *Journal of Addictive Diseases,* vol. 19, pp. 75-81, 2000.

[9] P. Willner, "Alcohol and cannabis expectancies in adolescents: Relationship to alcohol and cannabis use, and a potential mechanism for an alcohol-drug gateway," *Journal of Psychopharmacology,* vol. 14, p. A10, 2000.

[10] R. J. Kane and G. S. Yacoubian, "Patterns of drug escalation among Philadelphia arrestees: An assessment of the gateway theory," *Journal of Drug Issues,* vol. 29, pp. 107-120, 1999.

[11] N. Graham, "A test of magnitude: Does the strength of predictors explain differences in drug use among adolescents?," *Journal of Drug Education,* vol. 27, pp. 83-104, 1997.

[12] P. Willner, "A view through the gateway: expectancies as a possible pathway from alcohol to cannabis," *Addiction,* vol. 96, pp. 691-703, 2001.

[13] L. M. Sanchez-Zamorano, *et al.*, "Prevalence of illicit use in function of tobacco smoking in Mexican students sample," *Salud Publica De Mexico,* vol. 49, pp. S182-S193, 2007.

[14] M. E. Patrick, *et al.*, "A Prospective Longitudinal Model of Substance Use Onset Among South African Adolescents," *Substance Use & Misuse,* vol. 44, pp. 647-662, 2009.

[15] K. Suzuki, *et al.*, "Is adolescent tobacco use a gateway drug to adult alcohol abuse? A Japanese longitudinal prospective study on adolescent drinking," *Japanese Journal of Alcohol Studies & Drug Dependence,* vol. 43, pp. 44-53, 2008.

[16] K. Wada, "The concept of "Gateway Drug"," *Japanese Journal of Alcohol Studies and Drug Dependence,* vol. 34, pp. 95-106, 1999.

[17] A. Kokkevi, *et al.*, "Psychosocial correlates of substance use in adolescence: A cross-national

study in six European countries," *Drug and Alcohol Dependence,* vol. 86, pp. 67-74, 2007.

[18] J. Peterson, "A Qualitative Comparison of Parent and Adolescent Views Regarding Substance Use," *Journal of School Nursing,* vol. 26, pp. 53-64, 2010.

[19] M. A. Luengo, *et al.*, "A cross-national study of preadolescent substance use: exploring differences between youth in Spain and Arizona," *Subst Use Misuse,* vol. 43, pp. 1571-93, 2008.

[20] R. R. Pate, *et al.*, "Associations between physical activity and other health behaviors in a representative sample of US adolescents," *American Journal of Public Health,* vol. 86, pp. 1577-1581, 1996.

[21] J. C. Merrill, *et al.*, "Cigarettes, alcohol, marijuana, other risk behaviors, and American youth," *Drug and Alcohol Dependence,* vol. 56, pp. 205-212, 1999.

[22] I. Sutherland and P. Willner, "Patterns of alcohol, cigarette and illicit drug use in English adolescents," *Addiction,* vol. 93, pp. 1199-1208, 1998.

[23] B. M. Yarnold, "The use of alcohol by Miami's adolescent public school students 1992: Peers, risk-taking, and availability as central forces," *Journal of Drug Education,* vol. 28, pp. 211-233, 1998.

[24] J. M. Lessem, *et al.*, "Relationship between adolescent marijuana use and young adult illicit drug use," *Behavior Genetics,* vol. 36, pp. 498-506, 2006.

[25] M. T. Lynskey, *et al.*, "Escalation of drug use in early-onset cannabis users vs co-twin controls," *Jama-Journal of the American Medical Association,* vol. 289, pp. 427-433, 2003.

[26] A. R. Morral, *et al.*, "Reassessing the marijuana gateway effect," *Addiction,* vol. 97, pp. 1493-1504, 2002.

[27] C. J. Rebellon and K. Van Gundy, "Can social psychological delinquency theory explain the link between marijuana and other illicit drug use? Al ongitudinal analysis of the gateway hypothesis," *Journal of Drug Issues,* vol. 36, pp. 515-539, 2006.

[28] K. Yamaguchi and D. B. Kandel, *Loglinear Sequence Analyses: Gender and Racial/Ethnic Differences in Drug Use Progression - Stages and Pathways of Drug Involvement: Examining the Gateway Hypothesis*. Cambridge: Cambridge University Press, 2002.

[29] V. Ganti, *et al.*, "Mining very large databases," *Computer,* vol. 32, pp. 38-45, 1999.

[30] L. A. Goodman, "A new model for scaling response patterns: An application of quasi-independent concept," *journal of American Statistical Association,* vol. 70, pp. 755-768, 1975.

SESSION

SEGMENTATION, CLUSTERING, ASSOCIATION

Chair(s)

Nikolaos Kourentzes
Philippe Lenca

Clustering Approach Based On Von Neumann Topology Artificial Bee Colony Algorithm

Wenping Zou[1,2], Yunlong Zhu[1], Hanning Chen[1], Tao Ku[1]
[1]Key Laboratory of Industrial Informatics, Shenyang Institute of Automation,
Chinese Academy of Sciences,
110016, Shenyang, China
[2]Graduate School of the Chinese Academy of Sciences,
100039, Beijing, China
{zouwp, ylzhu, chenhanning, kutao}@sia.cn

Abstract - *Article Bee Colony (ABC) is one of the most recently introduced algorithms based on the intelligent foraging behavior of a honey bee swarm. This paper proposes a new variant of the ABC algorithm based on Von Neumann topology structure, namely Von Neumann Neighborhood Article Bee Colony (VABC). VABC significantly improves the original ABC in solving complex optimization problems. Clustering is a popular data analysis and data mining technique. The most popular technique for clustering is k-means algorithm. However, the k-means algorithm highly depends on the initial state and converges to local optimum solution. In this work, VABC algorithm is tested on a set of widely-used benchmark functions and is used for solve data clustering on several benchmark data sets. The performance of VABC algorithm is compared with ABC and Particle Swarm Optimization (PSO) algorithms. The simulation results show that the proposed VABC outperforms the other two algorithms in terms of accuracy, robustness, and convergence speed.*

Keywords: Article Bee Colony; Von Neumann topology; Swarm Intelligence; Data cluster; k-means;

1 Introduction

Artificial Bee Colony (ABC) algorithm is a new swarm intelligent algorithm, which was first introduced by Karabog in Erciyes University of Turkey in 2005 [1], and the performance of ABC is analyzed in 2007 [2]. The ABC algorithm imitates the behaviors of the real bees on searching food source and sharing the information of food sources to the other bees. Since the ABC algorithm is simple in concept, easy to implement, and has fewer control parameters, it has been widely used in many fields, such as constrained optimization problems [3], neural networks [4] and clustering [5].

In [6][7], James Kennedy et al. had introduced the effects of various population topologies on the PSO algorithm in detail and they considered that the PSO with Von Neumann configuration performed more better than the other topologies structure. Hence, this paper applies Von Neumann topology structure to the ABC. In order to evaluate the performance of the VABC, we compared the performance of the VABC algorithm with that of ABC and PSO on a set of well-known benchmark functions. From the simulation results, the VABC algorithm shows remarked performance improvement over the ABC and PSO algorithms in all benchmark functions.

Data clustering is the process of grouping data into a number of clusters. The goal of data clustering is to make the data in the same cluster share a high degree of similarity while being very dissimilar to data from other clusters. Clustering algorithms can be simply classified as hierarchical clustering and partitional clustering [8]. This paper mainly focuses on partitional clustering. The most popular partitional clustering algorithm is k-means. In the past three decades, k-means clustering algorithm has been used in various domains. However, k-means algorithm is sensitive to the initial states and always converges to the local optimum solution. In order to overcome this problem, many methods have been proposed. Over the last decade, more and more stochastic, population-based optimization algorithms have been applied to clustering problems. For instance, Shelokar et al. have introduced an evolutionary algorithm based on ACO algorithm for clustering problem [9][10], Merwe et al. have presented PSO to solve the clustering problem [11][12] and Karaboga et al. have used the ABC algorithm [13]. In this paper, a VABC algorithm is applied to solve the clustering problem, which has been tested on a variety of data sets. The performance of the VABC on clustering is compared with results of the ABC and PSO algorithms on the same data sets. The above data sets are provided from the UCI database [14].

The rest of the paper is organized as follows. In Section 2，we will introduce the original ABC algorithm. Section 3 will discuss the Von Neumann topology structure, and our Von Neumann topology implementations of the ABC algorithm will be presented. Section 4 tests the algorithms on the benchmarks, and the results obtained are presented and discussed. The application of VABC algorithm on clustering is shown in Section 5, and the performance of VABC algorithm is compared with ABC and PSO algorithms on clustering problem in this section. Finally, conclusions are given in Section 6.

2 The article bee colony algorithm

The artificial bee colony algorithm is a new population-based metaheuristic approach, initially proposed by Karaboga [1][2] and further developed by Karaboga and Basturk [3][4]. It has been used in various complex problems. The algorithm simulates the intelligent foraging behavior of honey bee swarms. It is a very simple and robust optimization algorithm. In the ABC algorithm, the colony of artificial bees is classified into three categories: employed bees, onlookers and scouts. Employed bees are associated with a particular food source which they are currently exploiting or are "employed" at. They carry with them information about this particular source and share the information to onlookers. Onlooker bees are those bees that are waiting on the dance area in the hive for the information to be shared by the employed bees about their food sources, and then make decision to choose a food source. A bee carrying out random search is called a scout. In the ABC algorithm, first half of the colony consists of the employed artificial bees and the second half includes the onlookers. For every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources around the hive. The employed bee whose food source has been exhausted by the bee becomes a scout. The position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution represented by that food source. Onlookers are placed on the food sources by using a probability based selection process. As the nectar amount of a food source increases, the probability value with which the food source is preferred by onlookers increases [1][2]. The main steps of the algorithm are given in Table I:

In the initialization phase, the ABC algorithm generates a randomly distributed initial food source positions of *SN* solutions, where *SN* denotes the size of employed bees or onlooker bees. Each solution x_i (i =1, 2,..., *SN*) is a *D*-dimensional vector. Here, *D* is the number of optimization parameters. And then evaluate each nectar amount fit_i. In the ABC algorithm, nectar amount is the value of benchmark function.

In the employed bees' phase, each employed bee finds a new food source v_i in the neighbourhood of its current source x_i. The new food source is calculated using the following equation (1):

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (1)$$

where $k \in$ (1, 2,..., *SN*) and $j \in$ (1, 2,..., *D*) are randomly chosen indexes, and *k* has to be different from *i*. ϕ_{ij} is a random number between [-1,1]. And then employed bee compares the new one against the current solution and memorizes the better one by means of a greedy selection mechanism.

In the onlooker bees' phase, each onlooker chooses a food source with a probability which related to the nectar amount (fitness) of a food source shared by employed bees. The probability is calculated using the following equation (2):

TABLE I
MAIN STEPS OF THE ABC ALGORITHM

1:cycle=1
2:Initialize the food source positions x_i , i =1 ...*SN*
3:Evaluate the nectar amount (fitness fit_i) of food sources
4:repeat
5: Employed Bees' Phase
  For each employed bee
    Produce new food source positions v_i
    Calculate the value fit_i
    Apply greedy selection mechanism
  EndFor.
6: Calulate the probability values p_i for the solution.
7: Onlooker Bees' Phase
  For each onlooker bee
    Chooses a food source depending on p_i
    Produce new food source positions v_i
    Calculate the value fit_i
    Apply greedy selection mechanism
  EndFor
8: Scout Bee Phase
  If there is an employed bee becomes scout
  Then replace it with a new random source positions
9: Memorize the best solution achieved so far
10 cycle=cycle+1.
11:until cycle=Maximum Cycle Number

$$p_i = fit_i / \sum_{n=1}^{SN} fit_i$$

(2)

In the scout bee phase, if a food source can not be improved through a predetermined cycles, called "limit", it is removed from the population and the employed bee of that food source becomes scout. The scout bee finds a new random food source position using the equation (3) below:

$$x_i^j = x_{\min}^j + rand[0,1](x_{\max}^j - x_{\min}^j) \qquad (3)$$

where $x_{\min}^j$ and $x_{\max}^j$ are lower and upper bounds of parameter j, respectively.

These steps are repeated through a predetermined number of cycles, called Maximum Cycle Number (MCN), or until a termination criterion is satisfied [1][2][15].

3 Article bee bolony algorithm based on von neumann topology structure

The essence of driving swarm algorithm activity is social communication. The individual of the swarm will communicate their knowledge with their neighborhoods. Hence, the different concepts for neighborhood lead to different neighborhood topologies. Different neighborhood topologies primarily affect the communication abilities. Some kinds of population structures work well on some functions, while other kinds work well better on other functions. In [7], Kennedy theorized that populations with fewer connections might perform better on highly multimodal problems, while highly interconnected populations would be better for unimodal problems. After studying the various population topologies on the PSO performance, Kennedy considered that Von Neumann topology structure worked well on a wide range of problems [7]. Actually, the original ABC algorithm is a star topolgy structure, which is a fully connected neighhor relation. From equation (1), we noticed that k could be every particle except i. That means every particle is neighbor of ith particle. Because of Kennedy's studying, in this paper, we will we apply Von Neumann topology structure to the ABC, namely VABC.

Von Neumann topology was proposed by Kennedy and Mendes [7]. In the Von Neumann topolgy structure, an individual can communicate with four of its neighbors using a rectangular lattice topology. A graphical representation of the Von Neumann model is shown in Figure 1. In order to form the Von Neumann topology structure for M particles, we adopt below approch:

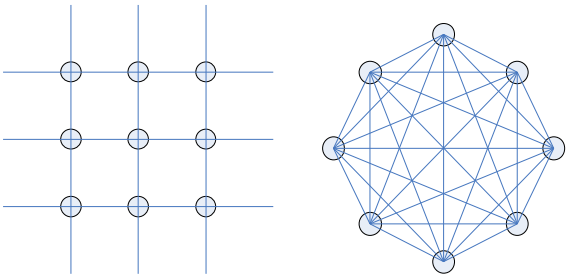


Fig. 1. The Von Neumann neighborhood is shown on the left and the star neighborhood is shown on the right.

(A) Arrange the M particles in rows rows and cols columns, that is M=*rows***cols*

(B) For the ith particle, $i \in \{1,2,\ldots,M\}$:

a) Up neighbor: $N_i(1)$ = (i-*cols*) mod M, if $N_i(1)$ = 0, $N_i(1)$=M

b) Left neighbor: $N_i(2)$ = i-1, if (i-1) mod *cols* = 0, $N_i(2)$ = i–1+*cols*.

c) Right neighbor: $N_i(3)$ = i+1, if i mod *cols* = 0, $N_i(3)$ = i+1-*cols*.

d) Down neighbor: $N_i(4)$ = (i+*cols*) mod M, if $N_i(4)$ = 0, $N_i(4)$=M

Notice the negative number in mod calculation: (-5) mod 20 = 15 [16]

In the VABC algorithm, in the employed bees' phase, we use Von Neumann topology. In the equation (1), k could only be the up, left, down right neighbor of particle i. However，in the onlooker bees' phase, we don't change the original ABC algorithm. This will improve convergence speed.

4 Experiments

4.1 Benchmark functions

In order to compare the performance of the proposed VABC algorithm with ABC and PSO, we used a set of well-known benchmark functions. The formulas and the properties of these functions are listed as follows:

Sphere function:

$$f_1(x) = \sum_{i=1}^{D} x_i^2$$

$$x \in [-5.12, 5.12]$$

Rosenbrock function:

$$f_2(x) = \sum_{i=1}^{D} 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2$$

$$x \in [-15, 15]$$

Griewank function:

$$f_3(x)=\frac{1}{4000}(\sum_{i=1}^{D}x_i^2)-(\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}))+1$$

$$x\in[-600,600]$$

Rastrigin function:

$$f_4(x)=\sum_{i=1}^{D}(x_i^2-10\cos(2\pi x_i)+10)$$

$$x\in[-15,15]$$

Ackley function:

$$f_5(x)=20+e-20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2})-\exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i))$$

$$x\in[-32.768,32.768]$$

Schwefel function:

$$f_6(x)=D*418.9829+\sum_{i=1}^{D}-x_i\sin(\sqrt{|x_i|})$$

$$x\in[-500,500]$$

4.2 Simulation results

In the experiment, all functions are tested on 30 dimensions; and the population sizes of VABC, ABC and PSO algorithms were 100. The PSO algorithm we used is the standard PSO. In PSO algorithm, inertia weight ω varies from 0.9 to 0.7 linearly with the iterations and the acceleration factors c_1 and c_2 were both 2.0 [17]. The experimental results, including the mean and standard deviation of the function values found in 30 runs are proposed in Table II and the two algorithms were terminated after 1,000 generation.

From Table II, the VABC algorithm is better than the ABC and PSO algorithms on all functions. On f_2 and f_6 functions, both ABC and VABC algorithms are almost the same. On f_4 functions, in the experiment, we found that the best value of the fitness functions for ABC algorithm is as good as VABC algorithm. However, ABC algorithm is easy trapped at local optimum, so the average value is worse than VABC. This means that the ability of VABC algorithm to get rid of local minima is very strong. On the other functions, the VABC algorithm is much better than the ABC algorithm. The VABC algorithm can increase the mean and the standard deviation of the functions by almost two orders of magnitude than ABC algorithm. The PSO converges very slowly and its performance is very bad on f_3, f_4, f_5 and f_6.

In order to show the performance of the VABC algorithm more clearly, the graphical representations of the results in Table II are reproduced in Figures 2-7. From the figures, we are concluded that the speed of convergence of VABC is much faster than ABC and PSO algorithms on all functions. From Figure 4, we can observe that the ABC algorithm is easy trapped at local optimum and the VABC algorithm is able to continue improving its solution on these two functions. For PSO algorithm, we can see that PSO algorithm is easy trapped in local optimum on f_3, f_4, f_5 and f_6. The performance PSO algorithms deteriorate in optimizing these functions

TABLE II
Units for Magnetic Properties Results comparison of different optimal algorithms for 30 runs

30D		VABC	ABC	PSO
f_1	Mean	6.7374e-016	1.1396e-014	2.8575e-008
	Std	9.4347e-017	8.0826e-015	3.8261e-008
f_2	Mean	1.7060e-001	3.3325e-001	2.6555e+001
	Std	1.7242e-001	2.3784e-001	1.7080e+001
f_3	Mean	1.0191e-008	6.0208e-007	4.1956e+002
	Std	5.3151e-008	3.2419e-006	2.9238e+001
f_4	Mean	9.1437e-007	1.8603e-001	4.6671e+001
	Std	4.8177e-006	3.6710e-001	1.2656e+001
f_5	Mean	4.1871e-008	6.0643e-006	4.2520e+000
	Std	1.6534e-008	3.5254e-006	8.3370e-001
f_6	Mean	9.4874e+001	1.9897e+002	9.5252e+003
	Std	8.4636e+001	1.1697e+002	3.7111e+002

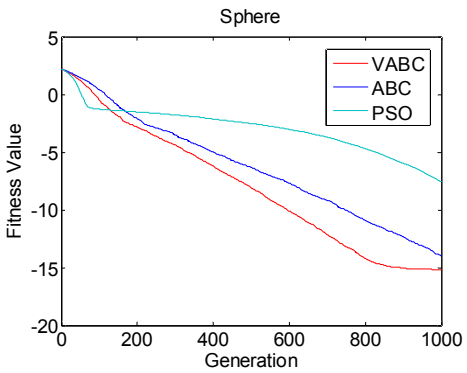


Fig. 2. The median convergence characteristics of Sphere function.

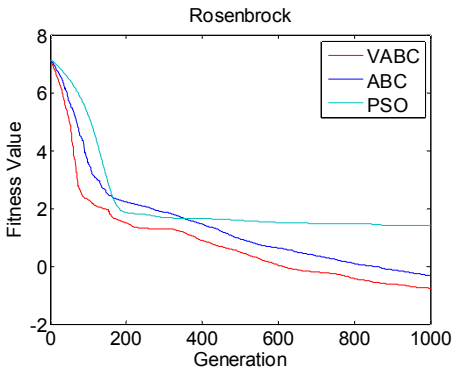


Fig. 3. The median convergence characteristics of Rosenbrock function.

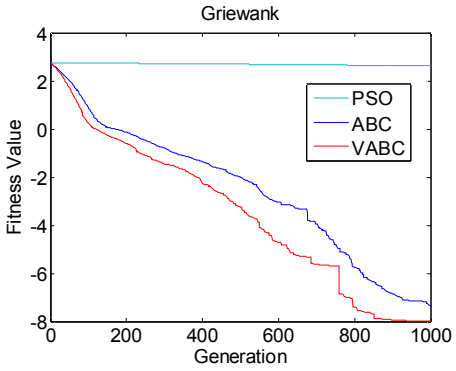


Fig. 4.The median convergence characteristics of Griewank function.

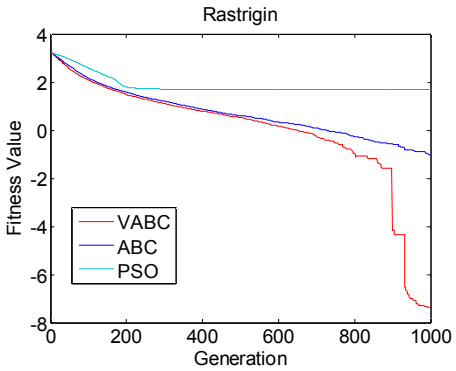


Fig. 5.The median convergence characteristics of Rastrigin function.

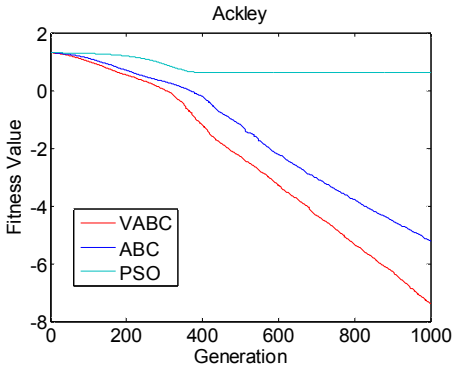


Fig. 6.The median convergence characteristics of Ackley function.

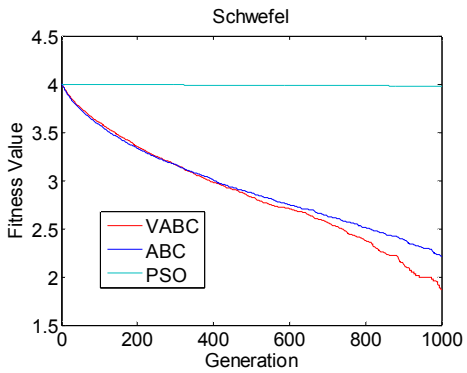


Fig. 7.The median convergence characteristics of Schwefel function.

5 Data clustering

5.1 K-means algorithms

As mentioned above, the goal of data clustering is grouping data into a number of clusters and k-means algorithm is the most popular clustering algorithm. In this section, we briefly describe the k-means algorithm. Let $X=(x_1, x_2,\cdots, x_n)$ be a set of n data and each data vector is a p-dimensional vector. Let $C=\{c_1, c_2,\cdots, c_k\}$ be a set of K clusters and K denotes the number of cluster centroids which is provided by the user. In k-means algorithm, firstly, randomly initialize the K cluster centroid vectors and then assign each data vector to the class with the closest centroid vector. In this study, we will use Euclidian metric as a distance metric. The expression is given as follows:

$$d(x_i, c_j) = \sqrt{\sum_{k=1}^{P} (x_{ik} - c_{jk})^2} \tag{4}$$

After all data being grouped, recalculate the cluster centroid vectors by using:

$$c_j = \frac{1}{n_j} \sum_{\forall x_i \in c_j} x_i \tag{5}$$

where n_j is the number of data vectors which belongs to cluster j. After the above process, reassign the data to the new cluster centroids and repeat the process until a criterion is satisfied. In this study, the criterion is when the maximum number of iterations has been exceeded. To know whether the partition is good or not, a measure for partition must be defined. A popular performance function for measuring goodness of the partition is the total within-cluster variance or the total mean-square quantization error (MSE) [18][19], which is defined as follows:

$$Perf(X, C) = \sum_{i=1}^{N} Min\{\| X_i - C_l \|^2 | l = 1,..., K\} \tag{6}$$

Because k-means algorithm is sensitive to the initial states and always converges to the local optimum solution, more population-based stochastic search algorithms are presented. In this paper, we will use VABC algorithm to solve clustering problem.

5.2 VABC algorithms on clustering

In the VABC algorithm, each individual represents a solution in K dimensional space. The number of dimension is equal to the number of clusters. Each component of an individual represents a cluster centroid and each cluster centroid is a p-dimensional vector. In the

initialization phase, we use maximum and minimum value of each component of the data set (which is to be grouped) as VABC algorithm individuals' initialization range. And initial solution is randomly generated in this range. We use the expression (6) to calculate the fitness function of individuals. Here the main steps of the fitness function are given below:

Main steps of the fitness function
For data vector x_i
Calculate the Euclidean distance by using (4)
Assign x_i to the closest centroid cluster c_j.
Calculate the measure function using equation (6)
EndFor.
Return value of the fitness function.

5.3 Clustering experimental results

To evaluate performance of the proposed VABC approach for clustering, we compare the results of the ABC and PSO clustering algorithms using five different data sets. They are Iris, Wine, Contraceptive Method Choice, Wisconsin breast cancer and Ripley's glass, which are selected from the UCI machine learning repository [14]. The detailed description of the test datasets can be seen [20].

For every data set, each algorithm is applied 30 times individually with random initial solution. The parameters of all algorithms are set like the section 4. Table Ⅲ summarizes the intra-cluster distances, as defined in Eq (6), obtained from all algorithms for the data sets. The average and standard deviation are presented.

TABLE III
COMPARISON OF INTRA-CLUSTER DISTANCES FOR THE THREE CLUSTERING ALGORITHMS

30D		VABC	ABC	PSO
Iris	Mean	**9.4607e+001**	9.4607e+001	9.7526e+001
	Std	1.0551e-003	7.7734e-003	4.4576e+000
Wine	Mean	1.6302e+004	**1.6298e+004**	1.6372e+004
	Std	1.4904e+000	6.2411e+000	1.0718e+002
CMC	Mean	**5.6952e+003**	5.6954e+003	5.7293e+003
	Std	1.6994e+000	1.3824e+000	4.0245e+001
Cancer	Mean	**2.9644e+003**	2.9644e+003	2.9656e+003
	Std	2.2817e-003	1.0731e-002	2.2730e+000
Glass	Mean	**2.2403e+002**	2.2539e+002	2.5881e+002
	Std	7.4652e+000	1.2685e+001	1.4607e+001

From the values in Table Ⅲ, we can conclude that the results obtained by VABC are as good as ABC algorithm and it is clearly much better than the PSO algorithm for all data sets. For all test data set, except for Wine, the VABC found solutions smaller. Especially in Cancer and Glass data set, we can see from Table Ⅲ the mean value of VABC is one order of magnitude better than that of ABC. From the mean value of VABC, we can conclude that the VABC algorithm is able to converge to the global optimum in all of runs.

6 Conclusion

This paper, based on the Von Neumann topology structure, a novel Article Bee Colony (ABC) algorithm is presented, namely Von Neumann Neighborhood Article Bee Colony (VABC). This resulted in a significant improvement in the performance in terms of solution quality, convergence speed and robustness.

In order to demonstrate the performance of the VABC algorithm, we compared the performance of the VABC with those of ABC and PSO algorithms on a set of benchmark functions. From the simulation results, it is concluded that the proposed algorithm has the ability to attain the global optimum and get rid of local minima, moreover it definitely outperforms the original ABC.

Because the VABC algorithm can be efficiently used for multivariable, multimodal function optimization, we apply it to solve clustering problems. The algorithm has been tested on several well-known real data sets. To evaluate the performance of the VABC algorithm on clustering problems, we compare it with the original ABC and PSO. From the experimental results, we can see that the proposed optimization algorithm is better than the other algorithms in terms of average value and standard deviations of fitness function. Therefore VABC can be considered a viable alternative to solve multivariable, multimodal optimization problems.

7 Acknowledgment

This work is supported by the 863 Hi-Tech research and development program of China under contract No.2008AA04A105.

8 References

[1] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[2] D.Karaboga and B.Basturk, On the performance of artificial bee colony(ABC) algorithm, Applied Soft Computing 8(2008), pp.687-697, 2008

[3] D. Karaboga, B. Basturk, Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems, LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft

Computing, Vol: 4529/2007, pp: 789-798, Springer-Verlag, 2007, IFSA 2007.

[4] D. Karaboga, B. Basturk Akay, Artificial Bee Colony Algorithm on Training Artificial Neural Networks, Signal Processing and Communications Applications, 2007. SIU 2007, IEEE 15th. 11-13 June 2007, Page(s):1 - 4, doi: 10.1109/SIU.2007.4298679

[5] C. Ozturk, D. Karaboga, Classification by Neural Networks and Clustering with Artificial Bee Colony (ABC) Algorithm,6th INTERNATIONAL SYMPOSIUM on INTELLIGENT and MANUFACTURING SYSTEMS "FEATURES, STRATEGIES AND INNOVATION", October 14-17, Sakarya, Turkey.

[6] R. Mendes, J. Kennedy, and J. Neves, " The fully informed particle swarm: Simpler, maybe better", IEEE Trans. Evol. Comput, vol. 8, no. 3, pp. 204-210, 2004.

[7] J. Kennedy and R. Mendes. "Population structure and particle swarw performance." in Proceedings of the IEEE Congress on Evolutionary Computation. (Honolulu. Hawaii USA). May 2002.

[8] J.Han, M.Kamber, Data Mining: Concepts and Techniques, Academic Press, 2001.

[9] P.S.Shelokar, V.K.Jayaraman, B.D.Kulkarni, An ant colony approach for clustering, Analytica Chimica Acta 509 (2) (2004) 187–195.

[10] Y.Kao, K.Cheng, An ACO-based clustering algorithm, in:M.Dorigo, etal. (Eds.), ANTS, LNCS4150, Springer, Berlin, 2006, pp.340–347.

[11] M.Omran, A.Engelbrecht, A.Salman, Particle swarm optimization method for image clustering, Int.J.PatternRecogn. Artif. Intell. 19 (3) (2005) 297–322.

[12] V.D. Merwe, A.P. Engelbrecht, Data clustering using particle swarm optimization, in: Proceedings of IEEE Congress on Evolutionary Computation 2003 (CEC2003), Canbella, Australia, 2003, pp. 215–220

[13] D.Karaboga and C.Ozturk, A novel clustering approach: Artificial Bee Colony(ABC) algorithm, Applied Soft Computing 2010

[14] C.L. Blake and C.J. Merz. UCI Repository of Machine Learning Databases, Available from: http://archive.ics.uci.edu/ml/datasets.html.

[15] Akay, B., Karaboga, D.: Parameter tuning for the artificial bee colony algorithm. In Proceeding of the First International Conference, ICCCI 2009, Wroclaw, Poland (2009)

[16] CHEN Zi yu, HE Zhong shi, ZHANG Cheng,: "Image multi-level thresholds based on PSO with Von Neumann neighborhood," Application Research of Computers, vol 26, no 5, pp.1977-1979, 2009

[17] Y.Shi and R.C.Eberhart, "Empirical study of particle swarm optimization," in Proceedings of the IEEE Congress on Evolutionary Computation(CEC'99), vol.3, pp.1945–1950, Piscataway, NJ, USA, 1999.

[18] K.R.Zalik, An efficient k-means clustering algorithm, Pattern Recognition Letters 29(2008) 1385–1391.

[19] Zulal Gungor and Alper Unler, K-harmonic means data clustering with simulated annealing heuristic, Appl. Math. Comput. (2006).

[20] T.Niknam, B.Bahmani Firouzi, M.Nayeripour, An efficient hybrid evolutionary algorithm for cluster analysis, World Applied Sciences Journal4(2) (2008)300–307.

Hierarchical Random Graphs for Networks with Weighted Edges and Multiple Edge Attributes

David Allen[1], Tsai-Ching Lu[1], Dave Huber[1], and Hankyu Moon[1]
[1]HRL Laboratories, LLC, Malibu, CA, USA

Abstract - *Network analysis has proven a useful tool for analyzing properties of complex systems. Many real-world systems exhibit hierarchical structure and analysis tools should leverage that knowledge in order to better extract essential features of the systems. Many datasets do not simply have binary network relationships, but have strengths of relationships or have multiple attribute relations. In this work we leverage and extend the recent work in hierarchical clustering, using hierarchical random graphs (HRGs), to enable analysis of networks with weighted edges and networks which contain multiple edge attributes. The technique also enables prediction of missing links in the dataset and identification of possible spurious links. We demonstrate the advantages of leveraging the edge weights and multiple edge attributes on example networks with future application to two real-world datasets.*

Keywords: HRGs, hierarchical clustering, weighted edges

1 Introduction

Network analysis has proven to be a useful tool for determining and quantifying the relationships between entities. Many complex systems exhibit hierarchical organization in these relationships. Their analysis has been successfully used in areas such as social network analysis, Internet data processing, authorship networks, ecological food webs, bioinformatics and medical data processing, among others. The proliferation of the World Wide Web has dramatically increased the ability to collect and analyze such network datasets.

The Hierarchical Random Graph (HRG) model is useful for clustering nodes in network graphs according to their connectivity with one another. The original algorithm was developed by Aaron Clauset [1], and employs Monte Carlo sampling methods to perform hierarchical clustering. It also provides the ability to infer possible missing links and identify spurious links. The HRG algorithm is applicable to networks with binary edges (i.e., two nodes are either connected by an edge or are not connected by an edge).

In [2] the authors use HRG modeling to analyze genetic diversity; in Fig. S1 of [2] they explicitly mention that part of the limitations they had were do "in part because this method does not take the edge weights into account. Unfortunately, no hierarchy test that takes edge weights into account exists to date" [2].

The main contributions of this work are to extend the HRG algorithm to networks consisting of weighted edges and networks where edges may have multiple weighted attributes. This greatly enhances the applicability of this model and makes it useful for clustering more realistic datasets, by explicitly incorporating the weights.

We demonstrate our approach on small examples, showing the advantages it provides, and then show preliminary results on two real-world datasets: 1) West Bank - a network dataset extracted from published literature about organizations in the West Bank, and 2) service/repair network linking parts and labor performed.

2 Background

2.1 Introduction

HRG is a flexible model for analyzing the hierarchical structure in network data from complex systems [1]. It provides an explicit approach for modeling the relationships between entities and provides means for identifying missing links and spurious links in data.

One of the core elements is the flexibility of the model. By focusing on the entities and the probability of links between hierarchical elements the model is able to fit a wide variety of data.

Significant prior work has been done on identifying clusters and structure within networks. [8] proposes a dynamic clustering algorithm which focuses on both interconnectivity and closeness. [3] focus on optimizing modularity and [4] evaluate communities as sets of adjacent motifs and extends the k-clique algorithm to find overlapping communities of interest. [5] identifies modules by finding optimal compression of the network topology. HRG's uniqueness stems from its reliance on explicitly modeling the hierarchical structure rather than simply on clustering.

2.2 HRG Model

Typically hierarchies are modeled as binary trees, or dendrograms, where closely-related entities share common ancestors whose position in the tree is lower than those which are more distantly related. The HRG model uses dendrograms where leaf nodes represent entities and each internal node of the tree is annotated with a probability value, θ_i, which is the

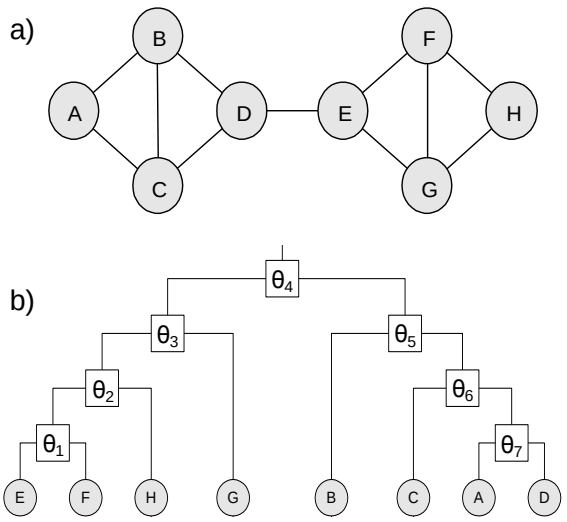


Fig. 1. a) Input binary network for clustering. b) Example "most likely" dendrogram produced by HRG algorithm. Each internal node is annotated with the probability of an edge between leaf nodes which have this as their common ancestor. For example, θ_3 is the probability that node G is connected to E, which is the same as connections for G-F and G-H.

probability of an edge between any two entities which contain this internal node as their common ancestor (see Fig. 1).

Most clustering algorithms tend to focus on groups of vertices which have high edge connectivity between nodes in the group and sparse connectivity externally, referred to as assortative behavior. HRGs can model this behavior when the θ_i values have high probability values lower in the tree and decrease at the higher nodes. HRGs are also able to model disassortative behavior, where groups have high external connectivity and low internal connectivity; an example of this type of network would be in food web networks where the predators and prey link to one another, but not to their own group. Not only can HRGs model assortative and disassortative behaviors, but combinations of these behaviors can be captured by the dendrogram and probability values.

The input to the HRG algorithm is a binary network, such as shown in Fig. 1a. The algorithm then performs hierarchical clustering using a maximum-likelihood approach and Monte Carlo sampling. This produces dendrograms, such as shown in Fig. 1b. Using techniques from phylogeny reconstruction many dendrograms can be "averaged," resulting in a single consensus dendrogram (see [1] for details).

The hierarchical clustering can then be used to identify missing links or spurious links in the dataset. Missing links are those which the hierarchical structure predicts but which do not appear in the provided dataset; one possible explanation for these would be unreliable sensor data producing the input data. Spurious links are defined as those which appear in the dataset, but which don't match with the hierarchical clustering.

HRGs are not only useful for extracting the hierarchical structure, but can also be used as a generative model to produce data with similar characteristics as the original dataset.

3 Weighted Links (wHRG)

Network datasets in many domains inherently have strengths associated with edges. For example, in a social network the strengths may be the number of communications between two users or the length of time two people have been friends with one another. Many analysis techniques begin by converting these strengths to binary network relations, for example by setting a threshold value such that if the value is less than the threshold the edge is not included and otherwise it is included. This however removes and ignores significant amounts of information; edges are treated the same whether they are very weak or very strong.

3.1 Algorithm

The construction of a dendrogram using the original HRG algorithm and weighted hierarchical random graph (wHRG) algorithm follow a similar process. Since the wHRG algorithm expresses the connection between nodes in the graph as a probability measure the weights must fall within the continuous interval from 0 to 1. Modeling the edge weights in this way preserves their compatibility with probabilistic modeling. If the weights fall outside that range they can be normalized. Both algorithms are initialized with a random dendrogram.

Given a dendrogram structure, the wHRG algorithm computes the θ_i values as follows:

$$\theta_i = \frac{\sum_{j=1}^{E_i} w_j}{L_i R_i}$$

where E_i is the number of edges in the dataset that have dendrogram node i as the lowest common ancestor, L_i and R_i are the number of leaves in the left and right subtree rooted at i, and w_j is the edge weight between a given edge that spans the separation between the two subtrees. This differs from the original HRG algorithm, which uses

$$\theta_i = \frac{E_i}{L_i R_i}$$

where E, L, and R are defined as above. This is the maximum likelihood approach and corresponds to the fraction of potential edges between two subtrees which appears in the dataset relative to the total possible number of edges. Rather than simply looking at the fraction of edges, the weighted version is adjusted to take the edge weights into account by computing the weighted sum of the edges relative to the maximal number. The weighted version does maintain consistency with the original, since in any network consisting of only unity weights (binary edges with weight 1.0) the θ_i values will be the same as in the HRG algorithm.

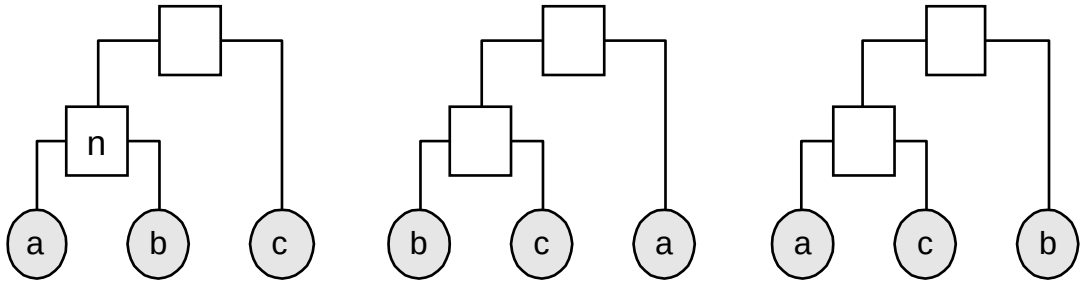


Fig. 2. Each internal network node, n, has three associated subtrees (a, b, c). Since the ordering of left/right children is not relevant there are two possible permutations, shown in the middle and the right of the figure.

Together the structure and θ_i values enable the algorithms to compute the likelihood of the dendrogram:

$$\mathcal{L} = \prod_{i=1}^{n-1} (\theta_i)^{E_i} (1-\theta_i)^{L_i R_i - E_i}$$

The algorithms continue by random perturbations of a single split in the dendrogram. First, one split location is randomly selected. As shown in Fig. 2, there are two alternate configurations of the associated subtrees (since there is no difference between swapping left and right subtrees). One of those is chosen as the new potential split node configuration. It computes the likelihood of the new dendrogram and then uses the standard Metropolis choice of deciding to accept the new configuration: always accept a change which increases the likelihood or does not change it and otherwise accept with probability specified by the likelihood values (see [6] for additional details).

3.2 Examples

To more fully analyze the differences between the algorithms we will use the example in Fig. 3. The left side of the figure depicts the weighted network input and the right is a dendrogram structure. If we examine the split at the root node (marked in blue) we see the subtree to the left contains leafs {a,b,c,d} and the right contains {e,f}. We have drawn boxes around each of these sets on the left of the figure (dashed blue line around the left branch and solid blue line around the right branch).

In the weighted HRG algorithm the θ_i value for this root node would be (1.0+0.2) / (4*2) = 0.15. However the original HRG algorithm would use: 2 / (4*2) = 0.25; these differences can lead to substantially different dendrograms of maximum likelihood. The split marked in red separates the sets {a,b,c} and {d}. The θ_i value for the red node are:

wHRG: 0.8 / (3*1) = 0.27
HRG: 1 / (3*1) = 0.33

To demonstrate how the addition of weights to the network graph can affect the resulting dendrogram, we examine the input network shown in Fig. 4. The network contains the same structure as that shown in Fig. 1, however this one contains weighted edges rather than binary edges.

When the Fig. 1a network is analyzed by the original HRG algorithm it produces a population of dendrograms similar to that shown in Fig. 1b (note this dendrogram is not unique, however those with similar likelihood mainly swap the positions of G&H or B&C). Applying the wHRG algorithm to a weighted version of the same network graph (Fig. 4) produces a dendrogram with significant differences, shown at the bottom of the figure. In this case, the clustering of the left and right portions of the graph are less defined, sharing a lower common branch in the dendrogram.

These examples demonstrate that leveraging edge strengths can lead to significant differences in the hierarchical

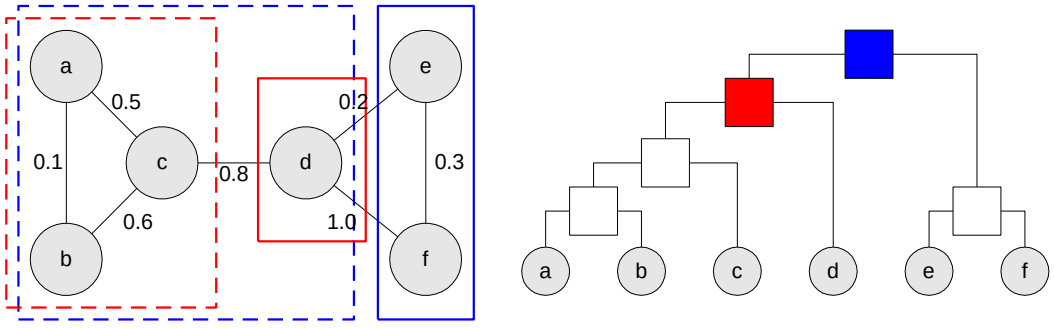


Fig. 3. Computing the dendrogram probabilities in a weighted network. Each split in the dendrogram (right) has a corresponding probability, θ_i, that is the key to computing the likelihood of the dendrogram. This is done by summing the weights of the connections that span the gap between the left and right subtrees produced by the split (in the figure the right subtree elements are denoted by a solid line and the left subtree elements use a dashed line) and dividing by the maximal number of connections between the two subtrees. In this example, the blue split results in (1.0+0.2)/(4*2)=0.15 and the red split results in 0.8/(3.0*1)=0.27.

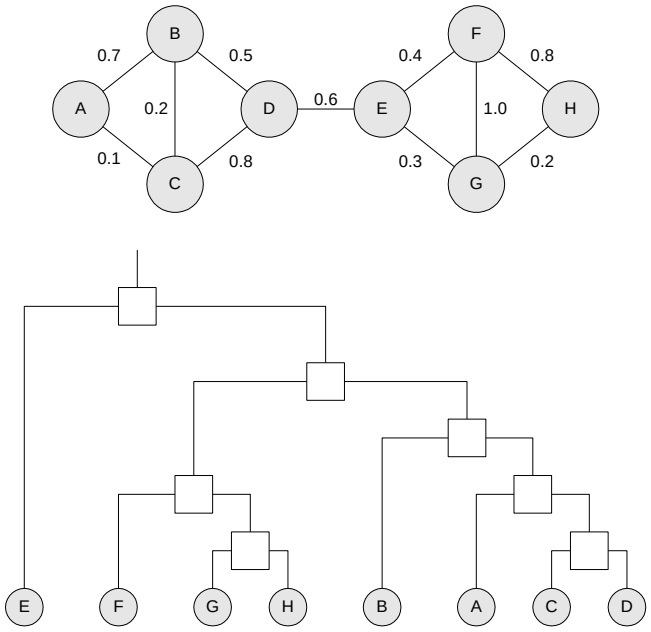


Fig. 4. (top) Network similar to that in Fig. 1, but with weighted edges. (bottom) Dendrogram produced by by wHRG algorithm, which differs from that in Fig. 1.

clustering; therefore employing this additional information may yield better analysis of the hierarchical structure. The differences between the weighted and unweighted versions are amplified further as the network datasets become larger.

4 Multi-Attribute Links (vHRG)

In many domains relationships not only have strengths associated with them, but usually there are multiple attributes connecting entities together. For example, in social networks (e.g. collected from mobile communications data) we may have friendship connections with an attribute based on the number of physical meetings which have taken place between them and also one based on how many times they have communicated electronically (phone calls, emails, etc.). In data from corporate environments the hierarchy may reflect the hierarchical nature of the organization, such as groups, departments and labs. Similar to the weighted case, many analysis algorithms begin by collapsing the data down to binary or singly weighted edges, or analyze each attribute independently. For example the friendship connections may be reduced to "friend or not friend" or the types of meetings and communications could be reduced to a single real-valued weight of "friendness" or communication frequency. Our proposed vectorized hierarchical random graph (vHRG) algorithm seeks to leverage this information rather than ignoring it, to produce more informative hierarchical structures.

4.1 Algorithm

The vHRG algorithm models multiple weighted attributes by specifying a vector of weights for each edge. Hence, instead of each edge having a single w_j value, it instead has a vector of values w_{nj} values where n indexes the attribute and j indexes the edge. Similar to the wHRG algorithm, the vHRG algorithm begins by normalizing the weights. In this case each attribute's weights can be normalized independently of one another, hence more emphasis could be given to one over the other, or the normalization process can be used to make weights which were in different scales comparable to one another.

The wHRG approach computes the θ_i values as the weighted sum of the edges between the subtrees relative to the maximal number of edges. The vHRG algorithm will use a similar technique over each of the attributes. When there are N attributes with weights

$$W_n = \{w_{nj} | j = 1 \dots K\},$$

the computation of the likelihood of the nth attribute is

$$\theta_{ni} = \frac{\sum_{j=1}^{E_i} w_{nj}}{L_i R_i}$$

where L, R, and E are defined the same as in the wHRG algorithm.

The likelihood, L_n, of the dendrogram with respect to the nth attribute is:

$$\mathcal{L}_n = \prod\nolimits_{i \in D} (\theta_{ni})^{E_{ni}} (1 - \theta_{ni})^{L_i R_i - E_{ni}}$$

Each likelihood score L_n represents the probability that the dendrogram, D, correctly reflects the hierarchical structure of the network constrained by the n^{th} attribute:

$$\mathcal{L}_n = \Pr(D | n^{th}\ attribute) = \Pr(D | W_n)$$

The final likelihood of the dendrogram that takes into account the entire N attributes is the probability:

$$\mathcal{L} = \Pr(D | all\ attributes) = \Pr(D | W_1, \dots, W_n) = \prod_{n=1}^{N} \Pr(D | W_n) = \prod_{n=1}^{N} \mathcal{L}_n$$

assuming that the conditional probabilities for the attributes are independent. If they are not dependent, those dependencies can be incorporated into the computation to extract more informative hierarchical clusters.

4.2 Examples

To illustrate the added utility of the vHRG algorithm we present two examples and illustrate how the features of the vHRG algorithm significantly enhance the hierarchical network analysis problem.

The first example illustrates extracting hierarchical structure from two attributes. The input network has fifty nodes along

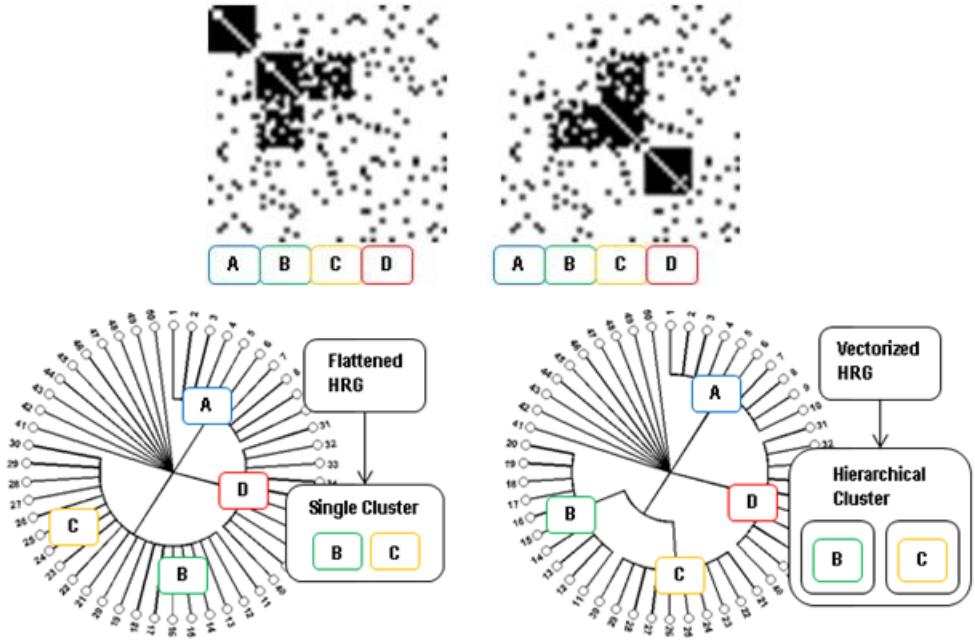


Fig. 5. Example demonstrating vHRG algorithm on a network consisting of two attributes and 50 nodes (40 of which can be clustered into four groups A, B, C, and D). The top of the figure depicts the strength of each attribute, where location (i,j) is the strength of the n^{th} attribute between node i and j. The consensus trees on the bottom demonstrate that flattening the network and then clustering was not able to extract the groups (e.g. B and C were combined), while the vHRG algorithm (bottom right) was able to identify all four groups and even identify the group {B,C} at the higher level.

with two sets of weights (one per attribute). The top two plots in Fig. 5 show these two attributes. The x and y axis each represent nodes in the network and the pixel intensity at the image location (i,j) is the strength of the edge between i and j. We have built into the network some clusters of nodes, which are labeled A, B, C and D in the figure. From the plots one can observe that the first attribute (top left) defines cluster A and B, while the second (top right) defines clusters C and D. Both of the attributes constrain the relation between the clusters B and C in a complementary manner.

We applied both the wHRG and vHRG algorithms to the network. Since the wHRG algorithm cannot handle multiple link attributes, the attributes were averaged and merged into a 'flattened' network. The consensus tree generated from the wHRG algorithm is shown on the bottom left of the figure. The clusters B and C have been merged into a single cluster. On the other hand, the vHRG consensus tree (bottom right) shows that B and C were each individually identified as well as the merged cluster {B,C} at the higher level of the tree. Both algorithms were able to correctly extract clusters A and D and also the nodes which were not contained in a group.

The second example also illustrates extracting hierarchical structure from two attributes which constrain the groups synergistically. Similar to the last example, the input network has 50 nodes with two weighted edge attributes (shown on left and middle of Fig. 6). The first attribute defines a large cluster while the second defines clusters A, B and C individually.

We again applied both the wHRG and vHRG algorithm to the network, where for the wHRG the edge attributes were again 'flattened.' The results of the wHRG algorithm are shown on the left of Fig. 7. Neither the large cluster nor the individual clusters have been correctly identified; clusters B and C were partially detected, but without any recognizable hierarchical structure. We also processed each of the attributes individually using the wHRG algorithm and the results are shown in the middle and right of Fig. 7. In these cases the algorithm failed to reveal the correct clusters.

On the other hand, the vHRG algorithm successfully discovered the clusters at the 'finer' level and the 'coarser' level as shown on the right of Fig. 6. A, B and C were all identified at the lower level and the larger cluster containing all of them were identified at the higher level. From the fact that the individual attributes failed to identify the corresponding clusters independently, the vHRG algorithm seems to utilize the information from the attributes in a synergistic manner.

The two examples clearly demonstrate the capability of the vHRG algorithm for revealing hierarchical structures within a network which would not have been discovered using previous approaches. The strength comes from preserving information from multiple relational attributes.

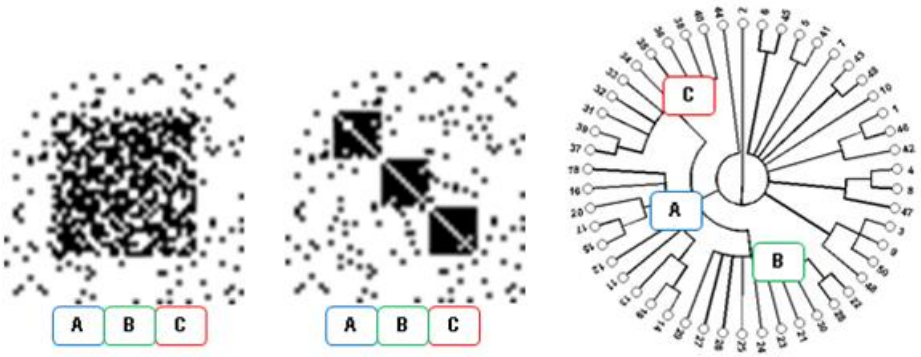


Fig. 6. Example demonstrating vHRG algorithm on a network consisting of two attributes and 50 nodes. The left and middle of the figure depict the strength of each attribute. The right contains the consensus tree from the vHRG algorithm showing that clusters A, B and C were successfully identified both at the finer grain and the coarser grain.

5 Missing and Spurious Links and Attributes

Similar to the HRG algorithm, our two extensions can enable the detection of missing and spurious links. We will also make a distinction between links and attributes in order to deal with networks with multiple edge attributes. For binary networks with one attribute, missing links are defined as those which do not appear in the dataset, but are predicted by the hierarchical structure. Spurious links are those which exist in the dataset, but have very low support from the extracted hierarchical structure.

In networks which have N attributes (e.g. **N** = {meeting, phone calls, emails…}) a missing or noisy attribute **K** will be identified as **K**⊆**N**. To detect missing links and attributes we use the following pseudocode:

1. Sample a set of dendrograms **D** from those generated by the Markov chain
2. For each pair of vertices (i,j) for which attributes in **K** are not observed in the network data, calculate the mean probability of θ_{ki} values for each k∈**K** for the pair.
3. Sort these pairs (i,j) and attribute k in descending order of the mean probability value and declare k a missing attribute if it is above a user specified threshold.
4. For pairs in which none of the attributes are observed in the network data, compute the mean probability over all attributes and declare a missing link between the pair if it is above a user specified threshold.

The above algorithm can easily be modified to compute spurious links and attributes by simply searching for pairs of vertices which are observed in the dataset and then sorting them in ascending order.

6 Real-World Datasets

In addition to the example networks shown in previous sections we have applied the wHRG and vHRG algorithms to two real-world datasets.

The first dataset is a network extracted from published literature about organizations in the West Bank by the CASOS group [7]. The network focuses on relationships between agents, knowledge, locations, organizations, resources, and tasks. It contains 240 nodes and 543 edges. The top of Fig. 8 depicts the original network and the bottom depicts the hierarchical structure extracted. Table 1 presents some examples of predicting missing links in the dataset.

The second network dataset was extracted from a service database and edges model relationships between parts and labor performed during product repair. The network contains 382 nodes and 1487 edges. Fig. 9 depicts the extracted hierarchical structure from this network. Clustering of this type of data can be used to detect irregularities in the dataset, such as people improperly assigning labor to jobs.

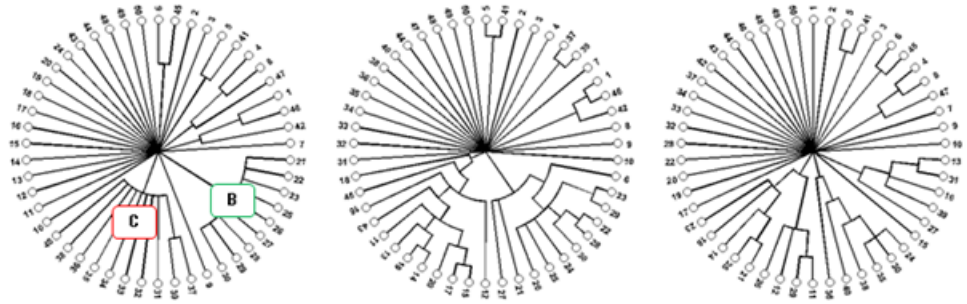


Fig. 7. Results from the network of Fig. 6 showing that the wHRG run on flattened data (left) or run independently on each of the two attributes (center and right) was unable to successfully extract the clusters A, B, and C.

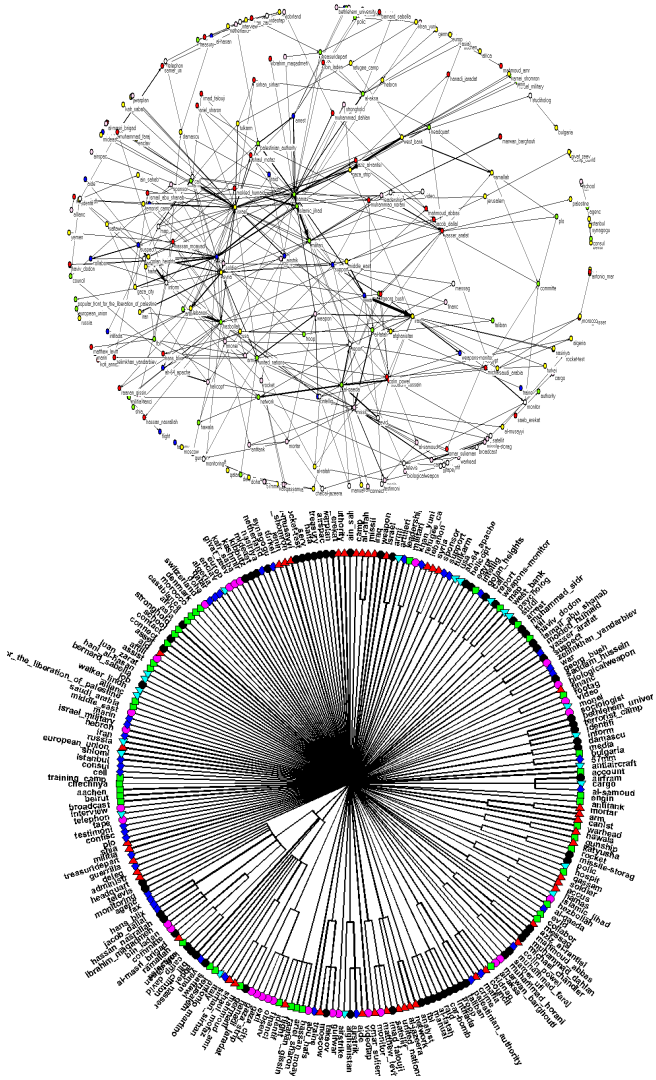


Fig. 8. (top) Graphical depiction of West Bank Dataset collected by CASOS group. (bottom) Hierarchical structure extracted using wHRG algorithm.

7 Conclusion

In this work we propose extensions to the hierarchical random graph model to improve the ability to identify hierarchical structure from network datasets. We do this by explicitly incorporating edge weights and multiple edge attributes. Additionally the tools enable prediction of missing links and spurious links in the dataset to enable further analysis and understanding of the data. We demonstrate the advantages on multiple example networks and two real-world networks.

8 References

[1] A. Clauset, C. Moore, and M. Newmann (2008), "Hierarchical Structure and the Prediction of Missing Links in Networks," in Nature 453:98-101.

[2] S. Halary, J. Leigh, B. Cheaib, P. Lopez, and E. Bapteste (2010), "Network analyses structure genetic diversity in independent genetic worlds," in PNAS, 107:127-132.

[3] M. Newman and M. Girvan (2004), "Finding and Evaluating Community Structure in Networks," in Phys. Rev. E, 69:026113.

[4] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek (2005), "Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society," in Nature, 435:814.

[5] M. Rosvall and C. Bergstrom (2007), "An Information-theoretic Framework for Resolving Community Structure in Complex Networks," in PNAS, 104:7327–7331.

[6] A. Clauset, C. Moore, and M. Newman (2007), "Structural Inference of Hierarchies in Networks in Airoldi," in IMCL 2007, 4503:1-13.

[7] West Bank Dataset http://www.casos.cs.cmu.edu/computational_tools/datasets/internal/west_bank_18/index2.html

[8] G. Karypis, E-H. Han, and V. Kumar (1999), "Chameleon: hierarchical clustering using dynamic modeling," in IEEE Computer, 32:8.

Table 1. Predicting missing links in West Bank Dataset

Source	Target	Probability
war	administr	0.4508
missle	weapon	0.435305
missle	bans_blix	0.408646
missle	satellite	0.404397
missle	wpns-monit	0.357264
missle	michael_ch	0.342774
west bank	bulgaria	0.306516
west_bank	plo	0.301491
west_bank	committee	0.30149

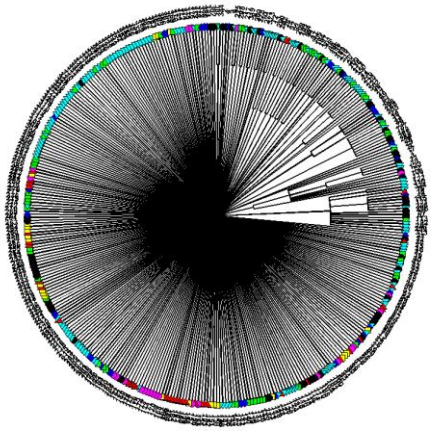


Fig. 9. Hierarchical structure detected in dataset of service repair database connecting parts and labor performed.

A Two-Stage Algorithm for Data Clustering

Abdolreza Hatamlou[1] and Salwani Abdullah[2]

[1]Islamic Azad University, Khoy Branch, Iran

[2]Data Mining and Optimisation Research Group, Center for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Abstract - *Cluster analysis is an important and popular data analysis technique that is used in a large variety of fields. K-means is a well-known and widely used clustering technique due to its simplicity in implementation and high speed in most situations. However, it suffers from two major shortcomings: it is very sensitive to the initial state of centroids and may converge to local optimum solution. In order to overcome the shortcomings of K-means, we present a two-stage approach, called KM-HS, which is based on K-means and a heuristic search algorithm. At the first stage of the proposed approach, K-means algorithm is applied to find an initial solution to the clustering problem and then at the second stage, a heuristic search algorithm is used to improve the quality of the initial solution by searching around it. The performance of the proposed algorithm is evaluated using four benchmark datasets from UCI repository. The experimental results indicate that the KM-HS algorithm not only finds high quality clusters but also converges more quickly than other evolutionary algorithms.*

Keywords: Cluster Analysis; K-means; Heuristic Search Algorithm

1 Introduction

Clustering analysis partitions a set of objects into subsets, as clusters, so that objects in the same cluster are similar or related to each other, while objects in different clusters are dissimilar or unrelated [1, 2]. Clustering analysis has been used in a large variety of fields and applications ranging from pattern recognition, web mining, image segmentation, genetics, microbiology, geography, remote sensing, psychology, education, marketing and business [3-8].

Clustering algorithms are broadly based on two main approaches: hierarchical and partitional [2]. Hierarchical algorithms can be divided as agglomerative (bottom-up) or divisive (top-down) methods. Agglomerative methods consider each object as a distinct cluster and combine them sequentially in larger clusters. Divisive methods start with the all objects as a singleton cluster and continue to divide it into sequentially smaller clusters.

Partitional methods, on the other hand, attempt to divide objects directly into a set of disjoint clusters, without making a tree structure. They try to optimize an objective function. Typically, the objective function involves in minimizing the dissimilarity in the objects within each cluster, while maximizing the dissimilarity between objects in different clusters.

Among traditional clustering algorithms, K-means is the most popular and widely used method due to its simplicity and computational efficiency, with linear time complexity [9]. However, it is very sensitive to the initial choice of centroids and may converge to local minima [10].

Recently, to overcome the drawbacks of K-means algorithm, many clustering techniques based on evolutionary algorithms such as genetic algorithm (GA), tabu search (TS), simulated annealing (SA), honey bee mating optimization (HBMO), ant colony optimization (ACO) and particle swarm optimization (PSO) have been proposed [11-16]. However, most of evolutionary methods are typically very slow to find optimal solution.

In order to overcome the above mentioned shortcomings, we present a novel clustering algorithm which combines K-means and a heuristic search algorithm, called KM-HS. In particular, we use K-means algorithm to find an initial solution for the clustering problem. Afterward, we use a heuristic search algorithm to thoroughly explore around of the

initial solution so that KM-HS might find global optimum.

The rest of this paper is organized as follows. Section 2 explains the basic principles of cluster analysis and K-means algorithm, while the proposed KM-HS algorithm is explained in section 3. Section 4 presents experimental results using four standard benchmark datasets. Finally, conclusion of this work is drawn in section 5.

2 Cluster analysis

Clustering is the process of partitioning a set of objects into a finite number of *k* clusters so that the objects within each cluster are similar, while objects in different clusters are dissimilar. In most of clustering algorithms, the criterion that is used to measure the quality of resulting clusters is defined as in Eq.(1), which is known as minimizing sum of squared error [17].

$$f(O,C) = \sum_{l=1}^{k} \sum_{O_i \in C_l} d(O_i,\ Z_l)^2 \quad (1)$$

where $d(O_i,\ Z_l)$ specifies the distance between object O_i and the cluster centroid Z_l.

Usually, similarity and dissimilarity between objects are expressed through some distance functions. The most common distance function is the Euclidean distance that is defined as follows:

$$d(O_i,\ O_j) = \sqrt{\sum_{p=1}^{d} (o_i^p - o_j^p)^2} \quad (2)$$

where $d(O_i, O_j)$ specifies the distance between two objects O_i and O_j. P is the dimensionality of the dataset.

As mentioned earlier, K-means is the most popular and widely used clustering algorithm. K-means starts with *k* initial centroids (these centroids are created randomly or derived from some heuristic approaches). Each object in the dataset is then assigned to the closest centroid. Centroids are updated by using the mean of the objects within each cluster. This process is repeated until a termination criterion is met.

3 Proposed approach

The proposed approach is built based on two main stages. At the first stage, the K-means algorithm is applied to find an initial solution to the clustering problem. K-means algorithm can do this efficiently, while it is very fast. However the output of K-means possibly is far from optimal solution and it can be improved by some other techniques. To ensure to get a good solution by the K-means algorithm in the first stage of the proposed algorithm, K-means is conducted 3 times and the best solution among them will be passed to the next stage for further improvement. At the second stage, we have applied a heuristic search method to search around the initial solution found by the k-means algorithm at the first stage. The proposed approach will try to improve the quality of the initial solution (output of the first stage) by searching around it in all dimensions. The structure of the proposed heuristic search is as follows:

At the first, an initial value will be considered as the initial step of movement for the algorithm. This value will be added to all features in the initial solution one by one. In other words, the threshold will be added to the first feature in the first centroid and then the fitness value of the new produced centroid will be calculated and compared with the fitness value of the current centroid. If there is an improvement in terms of fitness value, then the current centroid will be replaced by the new centroid. Otherwise, the current centroid will be reloaded and the search direction changes to the other side for the respective feature. Which means that, at the next iteration, the threshold value will be subtracted from the current value of the respective feature, and the above procedure will be done again. If there is no improvement in both sides of the considered feature in the considered centroid using the current threshold, the threshold value of the respective feature will be divided by two for the next iteration. This causes the heuristic search to act in a binary way and the time complexity of this stage to be logarithmic. The above mentioned procedure will be repeated for all features of the considered centroid, and then for the other centroids sequentially until the termination criteria are satisfied.

Based on the above description, the pseudo code of the KM-HS is stated as follows:

Stage 1: K-means method

1.1. Select k points as the initial centroids in a random way.
1.2. (Re)Assign all objects to the closest centroid.
1.3. Recalculate the centroid of each cluster.
1.4. Repeat steps 1.2 and 1.3 until a termination criterion is met.
1.5. Pass the solution to the next stage.

Stage 2: Heuristic search

```
For all centroids i=1...k do
  For all features j=1...d do
      If SDi (j)==1
          Ci(j)= Ci(j)+SSi(j);
          Calculate fitness value for the new centroid.
          If the fitness value has been improved
              Make the new centroid permanent
```

```
        Else
            Reload the old centroid
            SDi (j)=-1
        End if
    Else if SDi (j)==-1
        Ci(j)= Ci(j)-SSi(j)
        Calculate fitness value for the new centroid.
        If the fitness value has been improved
            Make the new centroid permanent
        Else
            Reload the old centroid
            SDi (j)=0
        End if
    Else if SDi (j)==0
        SSi(j)= SSi(j)/2;
        SDi (j)=1;
    End if
  End for
End for
```

In the above pseudo code, $SD = [SD_1, SD_2, ..., SD_k]$ is the search direction, and $SS = [SS_1, SS_2, ..., SS_k]$ is the search step, where SD_i is the search direction of the *i-th* centroid and the length of this array is *d*, which is the dimensionality of the test dataset. All fields in this array are initialized to 1 at the beginning, and change to -1, 0 and 1 during the search process. SS_i is the search step for the *i-th* centroid, and it will be set to *Max(dataset)* at the beginning. *Max(dataset)* is an 1-dimension array with length of *d*, where each member of it contains the maximum value of the corresponding field in the test dataset. $C=[C_1, C_2, \ldots, C_k]$ contains the centroids of *k* clusters and $C_i(j)$ specifies the *j-th* feature in the *i-th* cluster. To explain how the proposed heuristic search algorithm works, imagine that the snapshot of the system is like the Fig 1(a). For simplicity, we have considered only one centroid in this example with four features. As seen from the Fig 1(a) at the end of the current iteration a new centroid is produced using the current centroid, current search direction and current search step. Improvement has happened on the first and second features, whereas the third and forth features have not been changed. So, at the next iteration, search process will continue in the current direction for the features 1 and 2. For the 3rd feature, the search direction is 0, meaning that, no improvement happened for this feature in both directions in the previous iterations. So, for the next iteration, the search step will be divided by 2 and search direction will be set to 1. For the last feature, the search direction will be set to -1 i.e. to search for a better solution in the opposite direction as there is no room for improvement in the current direction so, the system situation for the next iteration will be like as Fig 1(b).

Current Centroid	4.63	1.28	7.41	0.37
SD	1	-1	0	1
SS	0.2	0.1	0.3	0.08
New Centroid	4.83	1.18	7.41	0.45
Final Centroid	4.83	1.18	7.41	0.37

(a)

Current Centroid	4.83	1.18	7.41	0.37
SD	1	-1	1	-1
SS	0.2	0.1	0.15	0.08
New Centroid	5.03	1.08	7.56	0.29
Final Centroid	?	?	?	?

(b)

Figure 1. An example for explaining the proposed heuristic search method

4 Experimental results

Four benchmark datasets are used to assess the performance of the proposed approach, KM-HS, in comparison with K-means [9], tabu search (TS) [11], particle swarm optimization (PSO) [12], genetic algorithm (GA) [13], honey bee mating optimization (HBMO) [14], simulated annealing (SA) [15] and ant colony optimization (ACO) [16]. The used datasets are Iris, Wine, Contraceptive Method Choice (CMC) and Wisconsin Breast Cancer that are provided by UCI repository of machine learning databases [18]. Datasets have the following characteristics:

Iris dataset (n=150, d=4, k=3): This dataset was collected by Anderson (1935). It contains three classes of 50 objects each, where each class refers to a type of iris flower. There are 150 random samples of iris flowers with four numeric attributes in this dataset. These attributes are sepal length and width in cm, petal length and width in cm. There is no missing value for attributes.

Wine dataset (n=178, d=13, k=3): This dataset contains the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. This dataset contains 178 instances with 13 continuous numeric attributes. There is no missing attribute value.

Contraceptive Method Choice also denoted as CMC (n = 1473, d = 9, k = 3): This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who either

Table 1
The results for different clustering algorithms

Dataset	Criteria	K-means	GA	SA	TS	ACO	HBMO	PSO	KM-HS
Iris	Best	97.33	113.98	97.45	97.36	97.10	96.75	96.89	96.65
	Average	106.05	125.19	99.95	97.86	97.17	96.95	97.23	96.65
	Worst	120.45	139.77	102.01	98.56	97.80	97.75	97.89	96.65
	Std	14.631	14.563	2.018	0.53	0.367	0.531	0.347	0
	NFE	120	38 128	5 314	20 201	10 998	11 214	4 953	660
Wine	Best	16 555.68	16 530.53	16 473.48	16 666.22	16 530.53	16 357.28	16 345.96	16 292.66
	Average	18 061.00	16 530.53	17 521.09	16 785.45	16 530.53	16 357.28	16 417.47	16 292.66
	Worst	18 563.12	16 530.53	18 083.25	16 837.53	16 530.53	16 357.28	16 562.31	16 292.66
	Std	793.21	0	753.084	52.073	0	0	85.49	0
	NFE	390	33 551	17 264	22 716	15 473	7 238	16 532	1 470
CMC	Best	5 842.20	5 705.63	5 849.03	5 885.06	5 701.92	5 699.26	5 700.98	5 693.72
	Average	5 893.60	5 756.59	5 893.48	5 993.59	5 819.13	5 713.98	5 820.96	5 693.72
	Worst	5 934.43	5 812.64	5 966.94	5 999.80	5 912.43	5 725.35	5 923.24	5 693.72
	Std	47.16	50.369	50.867	40.845	45.634	12.690	46.95	0
	NFE	270	29 483	26 829	28 945	20 436	19 496	21 456	1 200
Cancer	Best	2 999.19	2 999.32	2 993.45	2 982.84	2,970.49	2 989.94	2 973.50	2 964.38
	Average	3 251.21	3 249.46	3 239.17	3 251.37	3 046.06	3 112.42	3 050.04	2 964.38
	Worst	3 521.59	3 427.43	3 421.95	3 434.16	3 242.01	3 210.78	3 318.88	2 964.38
	Std	251.14	229.734	230.192	232.217	90.500	103.471	110.80	0
	NFE	180	20 221	17 387	18 981	15 983	19 982	16 290	840

were not pregnant or did not know if they were at the time of interview. The problem is to predict the choice of current contraceptive method (no use has 629 objects, long-term methods have 334 objects, and short-term methods have 510 objects) of a woman based on her demographic and socioeconomic characteristics.

Wisconsin breast cancer (n = 683, d = 9, k = 2): This dataset contains 683 objects, which characterized by nine features: clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. There are two clusters in the data: malignant (444 objects) and benign (239 objects).

For evaluating the efficiency of clustering algorithms, we have used two famous criteria: The first one is the mean square error (MSE), or within cluster variance (Eq. 1). Obviously, a small value for the MSE indicates high quality results and vice versa. The second criterion is the number of evaluations of the objective function (MSE in this study) to indicate that how fast the respective algorithm can find the solution of the given dataset. This is shown using NFE abbreviation. Clearly, the smaller value of the NFE shows the high convergence to optimal solution.

In this study, each experiment is done twenty times and the average and the standard deviation of solutions are calculated and reported as well as the best and worst solutions. Table 1 lists the results of the experiments.

The simulation results given in Table 1 confirm that the proposed approach, KM-HS, is robust and faster in comparison with other algorithms. In terms of MSE, which shows the quality of resulting clusters, KM-HS provides the optimum value and small standard deviation in all the test datasets. The KM-HS converges to the global optimum of 96.65, 16292.66, 5693.72 and 2964.38 on the iris, wine, CMC and cancer datasets, respectively in all of the runs, which these are better than other approaches. The standard deviation of the solutions found by the KM-HS is 0 for all of the test datasets, meaning that, it might converge to optimal value in all of the runs, whereas other approaches converge to local optima in some of the runs. In terms of the number of function evaluations (NFE), K-means algorithm is better than other methods. However, the quality of the output of the K-means algorithm is not satisfactory. In compare to other methods, the KM-HS needs the least number of function evaluations.

In brief, the results confirm that KM-HS has three significant merits in comparison to other methods. Firstly, it is a robust approach and able to find high quality clusters in all the test datasets. Secondly, it is a viable approach that can converge to global optimum in all runs. Finally, it is a fast algorithm and converges to optimal solution more quickly than other methods.

5 Conclusion

A hybrid clustering approach has been developed in this work, which is based on K-means and a heuristic search algorithm. In the proposed algorithm, the K-means is used to produce an initial solution to

the clustering problem and after that a heuristic search algorithm has been applied to improve the quality of this solution by searching around it. The performance of the proposed algorithm is evaluated using a number of standard benchmark datasets. The simulation results confirm that the proposed KM-HS algorithm able to obtain high quality clusters. Moreover, the convergence speed of the proposed algorithm is more quickly than other methods in comparison.

6 References

[1] J. Han, M.K.: 'Data Mining: Concepts and Techniques' (Academic Press, 2001.)

[2] Rui, X., and Wunsch, D., II: 'Survey of clustering algorithms', Neural Networks, IEEE Transactions on, 16, (3), pp. 645-678, 2005.

[3] Barni, M., and Gualtieri, R.: 'A new possibilistic clustering algorithm for line detection in real world imagery', Pattern Recognition, 32, (11), pp. 1897-1909, 1999.

[4] Cai, W., Chen, S., and Zhang, D.: 'Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation', Pattern Recognition, 40, (3), pp. 825-838, 2007.

[5] Cinque, L., Foresti, G., and Lombardi, L.: 'A clustering fuzzy approach for image segmentation', Pattern Recognition, 37, (9), pp. 1797-1807, 2004.

[6] Fan, J., Han, M., and Wang, J.: 'Single point iterative weighted fuzzy C-means clustering algorithm for remote sensing image segmentation', Pattern Recognition, 42, (11), pp. 2527-2540, 2009.

[7] Scheunders, P.: 'A genetic c-Means clustering algorithm applied to color image quantization', Pattern Recognition, 30, (6), pp. 859-866, 1997.

[8] Tjhi, W.-C., and Chen, L.: 'Possibilistic fuzzy co-clustering of large document collections', Pattern Recognition, 40, (12), pp. 3452-3466, 2007.

[9] Forgy, E.W.: 'Cluster analysis of multivariate data: efficiency versus interpretability of classifications', Biometrics, 21, pp. 2, 1965.

[10] Selim, S.Z., and Ismail, M.A.: 'K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality', Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-6, (1), pp. 81-87, 1984.

[11] Al-Sultan, K.S.: 'A Tabu search approach to the clustering problem', Pattern Recognition, 28, (9), pp. 1443-1451, 1995.

[12] Ching-Yi, C., and Fun, Y.: 'Particle swarm optimization algorithm and its application to clustering analysis', in Editor (Ed.)^(Eds.): 'Book Particle swarm optimization algorithm and its application to clustering analysis', pp. 789-794, Vol.782, 2004.

[13] Cowgill, M.C., Harvey, R.J., and Watson, L.T.: 'A genetic algorithm approach to cluster analysis', Computers & Mathematics with Applications, 37, (7), pp. 99-108, 1999.

[14] Fathian, M., Amiri, B., and Maroosi, A.: 'Application of honey-bee mating optimization algorithm on clustering', Applied Mathematics and Computation, 190, (2), pp. 1502-1513, 2007.

[15] Selim, S.Z., and Alsultan, K.: 'A simulated annealing algorithm for the clustering problem', Pattern Recognition, 24, (10), pp. 1003-1008, 1991.

[16] Shelokar, P.S., Jayaraman, V.K., and Kulkarni, B.D.: 'An ant colony approach for clustering', Analytica Chimica Acta, 509, (2), pp. 187-195, 2004.

[17] Jain, A.K.: 'Data clustering: 50 years beyond K-means', Pattern Recognition Letters, 31, (8), pp. 651-666, 2010.

[18] C.L. Blake, C.J.M.: 'UCI repository of machine learning databases. Available from: <http://www.ics.uci.edu/-mlearn/MLRepository.html>.

A binary based approach for generating association rules

Med El Hadi Benelhadj[1], Khedija Arour[2], Mahmoud Boufaida[1] and Yahya Slimani[3]
[1]LIRE Laboratory, Computer Science Department, Mentouri University, Constantine, Algeria
[2]National Institute of Applied Science and Technology, Tunis, Tunisia
[3]Computer Science Department, Faculty of Sciences, Tunis, Tunisia

Abstract - *Advanced database application areas, such as computer aided design, office automation, digital libraries, data-mining, hypertext and multimedia systems, need to handle complex data structures with set-valued attributes. These information systems contain implicit data that will be necessary to extract and exploit, by using data mining techniques. To exploit the data from these systems, the choice of appropriate storage structures becomes essential. In this paper, we propose a new compact structure to represent a transactions database, called a signatures tree, to speed up the signature file scanning. The construction of this tree requires only one single access to the transactions database. This tree will be used later to compute maximum support, extract frequent itemsets and generate association rules.*

Keywords: Data Mining, Frequent itemset, Signature file, Signature tree.

1 Introduction

Extracting Knowledge from Databases involves the extraction of implicit information, unknown and potentially useful, stored in large databases. The amounts of data collected are becoming increasingly important and their analysis more tedious. Data mining is an essential step in a KDD process.

The efficient search of information in large databases to extract knowledge from the contributing to a decision is vital for any expert. Several methods and techniques are used in KDD process to extract knowledge from large databases. Mining association rules which trends to find interesting association or correlation relationships among large amounts of data is one of these techniques. An association rule R is defined as an implication of the form R: S $\rightarrow$ T such that S $\subset$ I and T $\subset$ I and S $\cap$ T = Ø, I being a set of items. This generation of association rules involves two steps:

1. The extraction of frequent itemsets (with *support* $\geq$ *Minsup*),
2. The generation of association rules (with *confidence* $\geq$ *Minconf*).

Using this technique, we can generate, from a set of transactions, the frequent itemsets (itemsets with support above a minimum fixed by the user) and then the association rules from these itemsets. The first step is the most expensive with high demands for computation and data access [1] [3]. Because of that, we focus our attention in this paper on the frequent counting.

Our proposition consists to adopt a binary approach for generating frequent itemsets. Transaction database are represented by using a new compact data structure based on tree signatures. The use of signatures provides a low cost of storage and a speed of binary operations.

Several algorithms that generate association rules are based on two sub-steps "Generate" and "Verify" such as Apriori [1]. However, this phase is the most expensive, because of multiple access to transactions database. Other algorithms have tried to improve the algorithm Apriori. The partition algorithm [10] divides the transaction database into partitions, which increases the number of locally frequent itemsets that are globally rare, thus generating a loss of time doing redundant computation [13]. However, the algorithm *Dynamic Itemset Counting* [4] is a generalization of the *Apriori* algorithm. *FP-Growth* [9] extracts the frequent itemsets without generating candidate itemsets. It is based on a *FP-Tree* structure, which requires a complete reconstruction of the *FP-Tree*, for each updating. *DFPMT-A* [12] mining frequent itemsets are based on Apriori algorithm and uses dynamic approach like Longest Common Subsequence.

In order to compute the support of a collection of itemsets, it is necessary to access to the transaction database. As the transaction database is generally large, a solution for avoiding repetitive and costly access is to represent it using compact structures. As an example of these compact structures, we can mention: *BitMap* [8], *FP-Tree* [9], *Patricia tree* [11], *Transposed Form* [12] and so on.

Standard data structures cannot provide scalability, in terms of the data size and the performance for large databases, we must rely to adopt a binary and compact structure to improve performance and search space.

In this paper, we propose an approach using a binary tree structure to represent the transaction database. Each transaction is represented by a binary signature. The set of signatures is a signature file which is represented as a signature tree. In the process of generating frequent itemsets, a signature S_I is associated with each itemset I and is constructed with the same way that the selected transaction signatures. Each S_I is associated with the identifier of transaction (*Tid)*, which generating S_I. This process constructs the signatures transaction tree, a compact structure, finds all the frequent itemsets, based on a maximum support, and une only one access to transactions database.

The reminder of the paper is organized as follows: Section 2 gives an overview of the concept of tree signature. Section 3 presents our proposed structure called *STT* (Signature

Transaction Tree) and the tree construction process. Section 4 gives some basic concepts. In Section 5, we discuss the search process of a signature in *STT*. Section 6 is devoted to the process of generating frequent itemsets based on *STT*. In Section 7, we analyze theoretical complexity of our proposition. In Section 8, we report the experiment results. Finally, Section 9 concludes the paper.

2 Specification of the Signature tree

A signature file can be considered as a set of bit strings, which are called signatures. Several approaches have been proposed to represent the signature file: sequential signature file, bit_slice file and several other variants. The signature file method involves a high processing. This problem is resolved by partitionning the signatures file or by introducting an auxiliary data structure. Recently, Chen proposed an approach to represent the signature file as a signature tree [5].

Definition 1 [6]**:** A signature is a binary vector of length *m* obtained by applying one (or several) hash function (s).

Table I shows an example of signature extraction of a block *("full text scan"*). The signatures of the words of the block are combined by superposition (by or-ing) the word signature.

TABLE I. Signature Generation

Full Text Scan	*0000* *0000* *0000*	*0000* *0001* *1000*	*0000* *0000* *0000*	*0010* *0000* *0000*	*0000* *0000* *0000*
Bloc Signature	*0000*	*1001*	*0000*	*0010*	*0000*

Definition 2 [5]**:** A tree of signatures *Ts* represents a set of signatures $S = \{S_1, ..., S_n\}$ where: $S_i \neq S_j$ *for all* $i \neq j$ and $|S_k| = m$ *for* $1 \leq k \leq n$. *Ts* is a binary tree such that:

- For each internal node of *Ts*, the left edge leaving it is always labeled with "**0**" and the right edge is always labeled with "**1**".
- *Ts* have *n* leaves labeled *1, 2, …, n*, used as pointers to *n* different signatures $S_1, ..., S_n$ in *S*. Let *nf* be a leaf node. Denote the pointer *p(nf)* to the corresponding signature
- Each internal node *v* is associated to a positive number, noted by *position(v)*, to tell which bit will be checked.

3 Structure of STT

Improvement of algorithm performance for discovering association rules requires an optimization of the extraction phase of frequent itemsets. To reach this objective, we propose, the *STT* structure representing the transaction signatures. Each transaction is represented by a signature of size m. *STT* has the advantage of being both a compact structure (binary representation) and dynamic (care of updates). A signature tree contains two types of nodes: internal nodes and leaf nodes. For each internal node of *STT*, the left child corresponds to the value "**0**" and the right one to the value "**1**". Each leaf node contains three informations: a signature, the number of transactions generating this signature and the transactions identifier (Tid). The number of leaf node in STT is equal to the number of signatures.

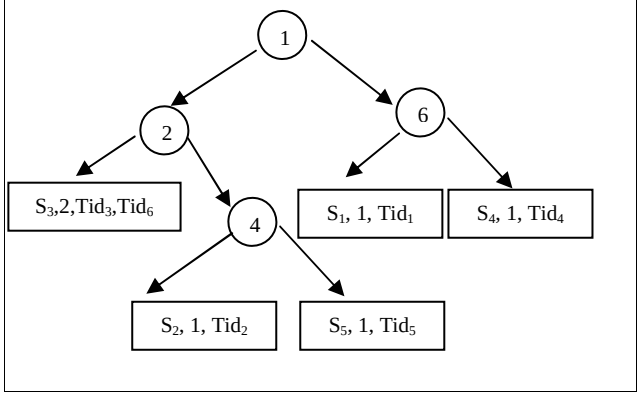
3.1. Example

The construction of a signature tree requires two phases:

- The application of the hash function *H(item) = Integer* (for example, we use modulo function) to obtain the signature for each item in a transaction; the composition of these signatures will give the transaction signature. All transactions and their signatures are represented in *Table II*. We note that the transactions T_3 and T_6 generate the same signature (the phenomenon of collision). It will be represented only once in *STT* (S_3 in our example).
- Each transaction signature is inserted in the *STT* tree. Each leaf of this tree contains the signature, the Tid and the number of transactions generating this signature.

TABLE II. Transactions signatures and corresponding *STT*

Tid	Transactions	Signatures
1	*1, 2, 4, 6, 8, 10*	*11101010*
2	*1, 2, 6, 10*	*01100010*
3	*3, 4, 10*	*00111000*
4	*1, 2, 4, 5, 8*	*11101100*
5	*1, 3, 4, 6, 10*	*01111010*
6	*2, 3, 4*	*00111000*



3.2. Construction process of STT

At the beginning, the tree contains an initial node: a node containing the first signature transaction, his Tid and the number "**1**". Then, we take a new signature transaction, a composition of signatures items, and we insert it into the *STT*. Let *s* be the signature we wish to enter. We cross the tree from the root. Let *v* be the node encountered and assume that *v* is an internal node with *position (v) = i*. Then, *s[i]* will be checked. If *s[i] = 0*, we go left, otherwise, we go right. If *v* is a leaf node, we compare *s* with the signature *s'* into *v*. If *s = s'*, we add only the *Tid* in the leaf *v* and increment the number of transactions *nt*. Otherwise, *s* is the new signature. We

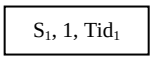
assume that the first *k* bits of *s* agree with *s'*; but *s* differs from *s'* in the $(k+1)^{th}$ position. We construct a new node *u* with *position (u) = k+1* and replace *v* with *u*. We mean that the position of *v* in the tree is occupied by *u* and *v* becomes one of u's children. If *position (u) = 1*, we make *v* be the left and *s* be the right child of *u*, respectively. If *position (u) = 0*, we make *v* the right child of *u* and *s* the left child of *u*.

1) Steps to generate STT

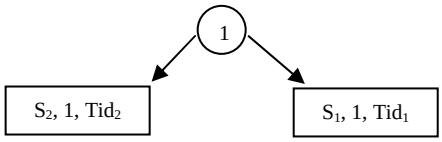
The following steps (a) to (f) show how to create the *STT*:

The step (a) build a root node *r* such that *r* is a leaf node and contains the signature S_1, the number "**1**" and the identifier of the first transaction T_1.

(a) Insert T_1 (S_1)

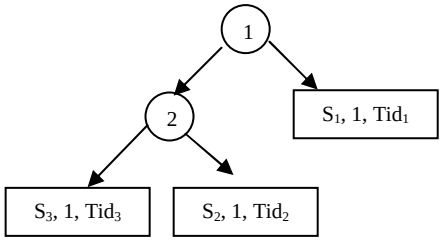


(b) Insert T_2 (S_2): $S_1[1] \neq S_2[1] \Rightarrow$ create internal node *v* with position(v) = 1 and leaf node {S2, 1, Tid_2}.

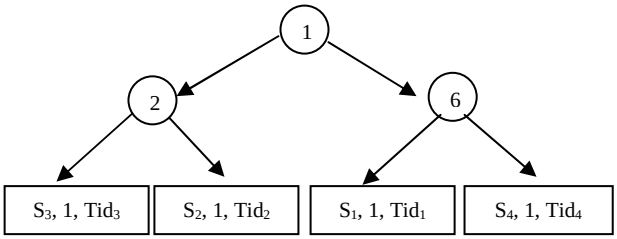


The steps (b) to (e) insert a new signature S_i to the corresponding leaf node in STT, using the value of signature bit position in each internal node.

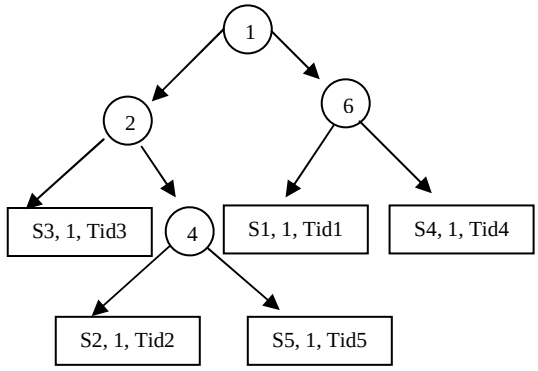
(c) Insert T_3 (S_3): $S_3[1] = 0$, $S_2[2] \neq S_3[2] \Rightarrow$ create internal node *v* with position(v) = 2 and leaf node {S2, 1, Tid_2}.



(d) Insert T_4 (S_4): $S_4[1] = 1$, the first different bit between S_4 and S_1 is the 6^{th} bit, $S_4[6] = 1$ and $S_1[6] = 0 \Rightarrow$ create internal node *v* with position(v) = 1 and leaf node {S2, 1, Tid_2}.



(e) Insert T_5 (S_5): $S_5[1] = 0$, $S_5[2] = 1$. The first different bit between S_2 and S_5 is the 4^{th} bit, S2[4] = 0 and S5[4] = 1 $\Rightarrow$ create internal node *v* with position(v) = 4 and leaf node {S5, 1, Tid_5}.



The step (f) inserts an existing signature. We use the same way of the steps (b) to (e). When arrives in a corresponding leaf node, we remarques that $S_6 = S_3$. We increment the number and we add the identifier of the transaction T_6.

(f) Insert T_6 (S_6): S6[1] = 0, S6[2] = 0. $S_6 = S_3 \Rightarrow$ Increment the number and add Tid_6 in the leaf node.

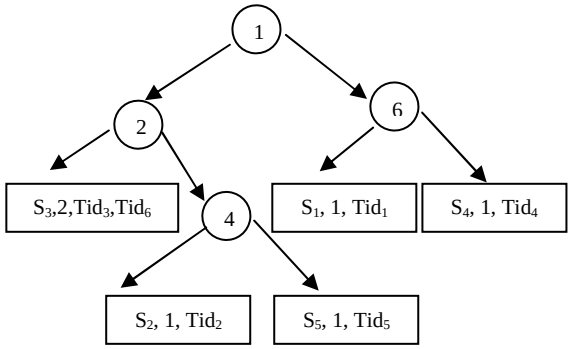


Figure 1. Steps of STT Construction.

2) Algorithm to construct STT

The algorithm *Cons_STT* to construct the tree is presented below. It is composed of two parts:

- Signature Generation: A hash function *"Gen_Sig(Ti)"* provides the signature of the transaction *Ti*.
- Insertion: Call *Insert(S_I)* procedure that inserts the signature S_I in *STT*.

TABLE III. STT Construction Algorithm

```
Algorithm Cons_STT
/*  STT construction*/
Begin
   /* Input: Set of Transactions */
   /* Output: STT */
   /* v is the current internal node, it contains 3
      fields: bit position to check, left child
     pointer and right child pointer */
   /* f is the current leaf node, containing 3
      fields: the signature S, Tids and number  */
   S1 = Gen_Sig (T1)
   Build a root node r such that r is a leaf node.
   /* It will contain the first transaction
     signature S1 and the corresponding Tid */
```

```
    For i = 2 à n Do
      Si = Gen_Sig (Ti)
      Call Insert (Si)
    EndDo
End
```

The following procedure inserts a given transaction signature to *STT*:

TABLE IV. Insert algorithm in STT

```
Procedure Insert (s,STT)
Begin
  Stack ← root
  While Stack not empty Do
      v ← pop (Stack)
      If v is internal node Then
        j ← position (v)
        If s[j] = 1 Then
          push (Stack, right_child)
        Else
          push (Stack, left_child)
        Endif
      Else  /* v: leaf node = s' and nt */
        If s = s' Then /* Old signature */
          nt ← nt + 1
        Else /* New signature */
          Assume that the first k bits of s agree with s'.
          s differs from s' in the (k+1)th position.
          Generate a new internal node with position(u)
          = k+1.
          Generate a new leaf node v' ←{s, Tid, 1}
          If s [k+1] = 1 Then
            v' will be the right child of u and v the left
            child
          Else
            v' will be the left child of u and v the right
            child
          Endif
        Endif
      Endif
  EndDo
End
```

The insertion procedure presents two possible cases:

- $s = s'$. In this case, we add Tid in the leaf node and we increment the transactions number *nt*.
- $s \neq s'$. A new internal node *u* is created, containing the corresponding position to first different bit between *s* and *s'*. We create also a new leaf node *v'* containing {*s, Tid, 1*}. If *position (s) = 1*, we make *v'* be the left and *v* be the right child of *u*. If *position (s) = 0*, we make *v'* the right child of *u* and *v* the left child of ***u***.

4 Basic Concepts

Definition 3: An item I_i is any object, attribute, literal, into a finite set of distinct elements D = $\{I_1, I_2, ..., I_n\}$.

Definition 4: An itemset *I* is a subset of *D*. A *k-itemset* is an itemset of cardinality *k*.

Definition 5: A transaction T_i is an itemset wich is associated to an identifier: the Transaction Identifier (Tid).

Definition 6: The support of an itemset *I* denoted *Support (I)* is the number of transactions containing *I*.

Definition 7: A minimum support *Minsup* is a threshold fixed by the user.

Definition 8: An itemset *I* is denoted fréquent if *Support(I)* $\geq$ *Minsup*.

Definition 9: The maximum support *Maxsup* of an itemset *I* is equal to the sum of the selected *Tids* sets. If S_I is the signature of the itemset *I* and $L_1, L_2, \ldots, L_p$ the set of selected *Tids* during the search process of S_I, the maximum support is:

$$Maxsup\ (I) = |L_1| + |L_2| + \ldots + |L_p|$$

Where $|L_i|$ = number of Tids in the leaf *i*.

Définition 10: An itemset *I* is said *Mfréquent* if *Maxsup(I)* $\geq$ *Minsup*.

5 Search Process in STT

Now, we discuss how to search a signature S_I of an itemset *I* in *STT*. During the traversal of *STT*, the inexact matching is done as follows:

1. Let *v* be the node encountered and *position(v)* be the position to be checked.
2. If *position(v) = 1*, we move to the right child of *v*.
3. If *position(v) = 0*, both the right and left child of *v* will be explored.

In fact, this process corresponds to the signature matching criterion, i.e., for a bit position *i* in S_I, if it is set to "**1**", the corresponding bit position in *s* must be set to "**1**"; if it is set to "**0**", the corresponding bit position in *s* can be "**1**" or "**0**".

This reflects that only the signatures *s*, such that $s \wedge S_I = s$, are selected.

Example 1. The itemset I is composed by the items 1, 2 and 6. If we apply the hash function, we obtain the signature S_I = *01100100*. The procedure *Search*(S_I) will select all the signatures that contain S_I (in our example, S_1, S_2 *and* S_5).

Figure 2 represents the bold path in the *STT* tree when searching S_I.

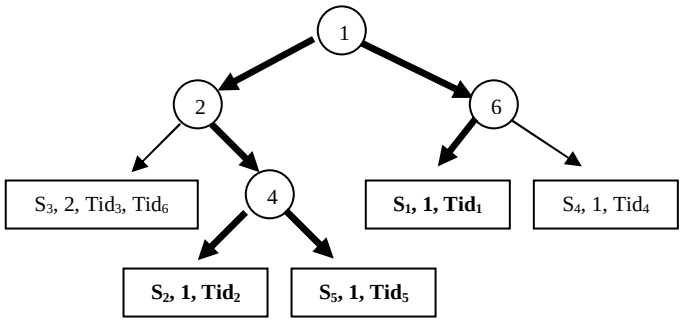


Figure 2. Result of search SI = 01100100

5.1. Algorithm to search a signature in STT

This algorithm search a signature and computes the maximum support of itemset using *STT* structure.

TABLE V. Search Algorithm in STT

```
Algorithm Search
    /* Input: an itemset I */
    /* Output:Maxsup (I) */
Début
    S_I = Gen_Sig (I)
    ST ← Ø
    push (Stack, root);
    While Stack not empty Do
      v ← pop (Stack);
      If v is an internal node Then
        i ← position (v) /*bit position to check */
        If S_I [i] = 1 Then
          push (Stack, right_child(v))
        Else
          push (Stack, left_child(v))
          push (Stack, right_child(v))
        Endif
      Else /* v is a leaf node */
        Compare S_I with the signature S
        If S contains SI Then
          ST ← ST ∪ {Tids}
        Endif
      Endif
    EndDo
    Use ST to compute Maxsup(I)
    Return Maxsup(I)
End
```

6 Frequent Itemsets Generation

The generation of frequent itemsets computes, for each candidate itemsets *I*, the maximum support of *I* M*axsup(I)* and compares it to a minimum *Minsup*, defined by the user. An itemset *I* is said frequent if *Maxsup(I)* is greather than M*insup*.

TABLE VI. Generation Frequent Itemset Algorithm

```
Algorithm Generation_FI
    /* Input: Set of itemsets I = {I1, I2,...,In}*/
    /* Output: Set of frequent itemsetsFI */
Begin
    /* Initially, FI is empty */For  i = 1 to n Do
        Search (I_i,Maxsup(I_i))
        If Maxsup(I_i) > Minsup Then
          FI = FI ∪{I_i} /* Union */
        Endif
    EndDo
    Return FI
End
```

Example 2. If we consider database transaction of *Table II*, the itemset *I = {1, 2, 6}* and *Minsup = 2*, we can select 3 signatures S_1, S_2 and S_5. Then, *Support(I) = 3* and is greather than Minsup. We conclud that the itemset *I* is a frequent one.

7 Complexity Study

The algorithm *Gen_STT* to build the signatures tree has a complexity of ***O(n*m)*** where:

- ***n***: number of transaction signatures
- ***m***: size of a signature

For against, the procedure for insertion of each signature in STT requires one tree parsing for the first signature, 2 for the second, and so on. The number of path traversed is:

$$1 + 2 + ... + n = n\,(n+1) / 2 = (n^2 + n) / 2$$

The complexity of the insertion algorithm is about $O\,(n^2)$.

The search procedure of a signature in the tree *STT* has been studied by *Chen* [5] and is of order $\boldsymbol{O\,(n/2^l)}$, where:

- ***n*** is the number of signatures
- ***l*** the number of bits to "**1**" in S_I.

The *Generation_FI* algorithm for generating frequent itemsets contains a loop that is run *n* times (*n* being the number of candidate itemsets).

Complexity to handle a candidate itemset is equal to n times that of the search procedure, thus the order of:

$$\boldsymbol{O(n\,(n/2^l)) = O\,(n^2/2^l)}.$$

Finally, the complexity of generating frequent itemsets is:

$$\boldsymbol{O\,(nm) + O\,(n^2) + O\,(n^2/2^l) \sim O\,(n^2).}$$

8 Experimental Study

We have implemented our proposal in *C++*, on an *Intel Core 2 Duo 1,80 GHZ and 2 GB RAM.*

At a first experimentation, we performed the same test on two transaction databases, dense and sparse, varying the minimum support to measure its influence on the total number of frequent itemsets extracted and comparing our results with those obtained by *Apriori* [2]. *Figure 3* and *Figure 4* show the results obtained respectively with Mushroom and *T10I4D100K* transactions bases. The number of extracted frequent itemset by our approch, based on maxsup, is about equal to the frequent itemset obtained by *Apriori*.

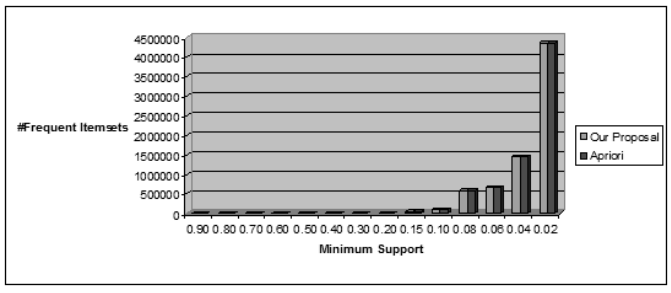


Figure 3. Experimentation with a dense database (Mushroom)

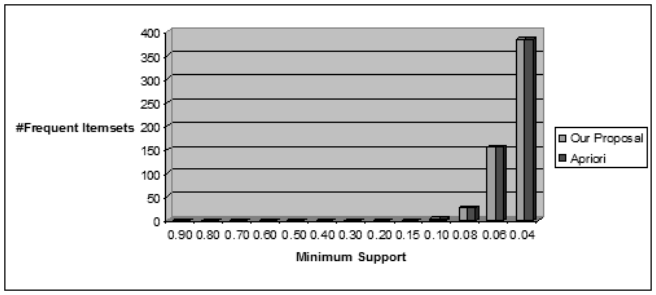


Figure 4. Experimentation with a sparse database (T10I4D100K)

In the second experience, we consider several transaction databases and we compute the time required for different ***Minsup.***

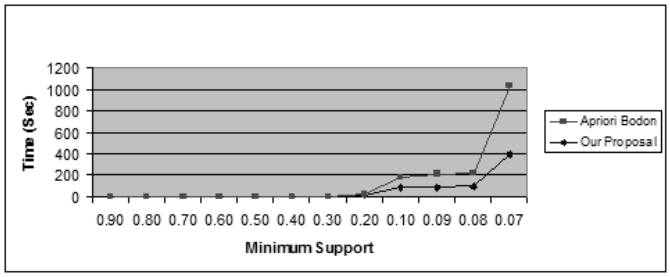


Figure 5. Time vs Minsup for Mushroom

Figure 5 shows the Time vs Minsup for *Mushroom* obtained by our approach and respectivelly *Apriori* [4]. Our approach gives a better result for a support less than 20% and the same result for support greather than 20%.

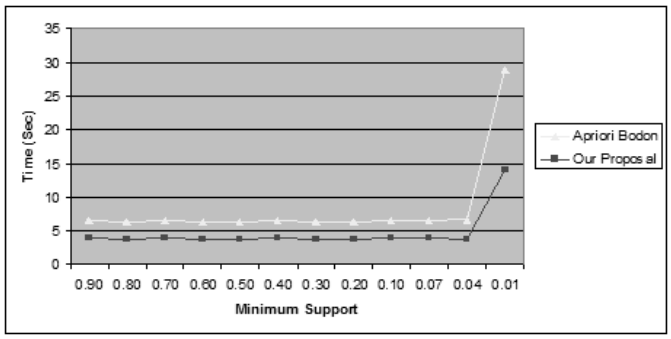


Figure 6. Time vs Minsup for Retail

Figure 6 gives the *Time* vs *Minsup* for *Retail* obtained by the both approch. Our result is also better for all supports.

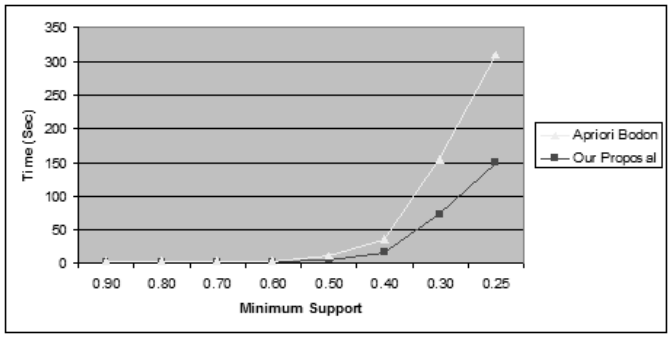


Figure 7. Time vs Minsup for Accident

In *figure 7*, we have the same time for support between 50% and 90% and a better time for support less than 50%.

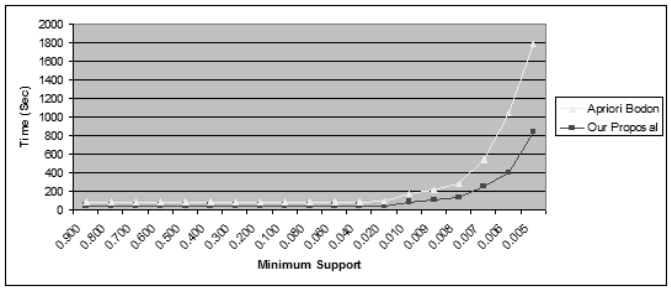


Figure 8. Time vs Minsup for Kosarak

Figure 8 shows that our result is better only for a little support (less than 1,5%).

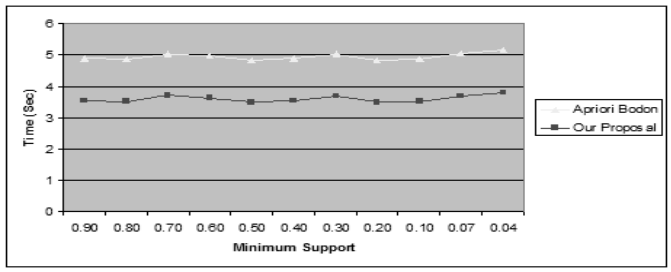


Figure 9. Time vs Minsup for T10I4D100K

Figure 9 gives a linear time for the both approch. Our approach provides better result than Apriori.

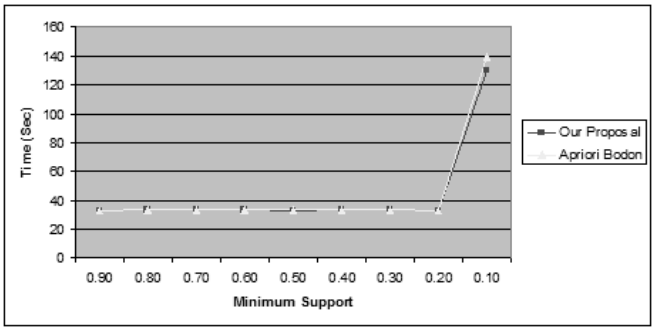


Figure 10. Time vs Minsup for T40I10D100K

In *figure 10*, for *T40I10D100K*, the result is approximatly the same for our approach and Apriori.

9 Conclusion

In this paper, we proposed a new data structure to represent database transactions. The main caracteristics of this structure is the use of a binary signature, which can be associated to each transaction. These binary signatures are then organised in a tree, in which each edge is labeled with "**0**" or "**1**", and each internal node is associated with a number, indicating which bit in a signature to check. Thus, the searching of a signature uses only a signature binary tree and nead only one acces to transactions database .

The complexity of our proposal is linear. In order to show the efficiency of our approch, we have conducted a series of experimentation to compare our proposal with the *Apriori* algorithm, using different database transactions and différent supports. The results of these experimentation show that the signature tree algorithm outperforms significantly *Apriori*.

10 References

[1] Agrawal R., Srikant Ramakrishnan, "Fast Algorithm for Mining Association Rules", in Proceeding of the 20th VLDB Conference Santiago, September 12-15, pp. 487- 499, Chile, 1994.

[2] Bodon F. "A Trie-based APRIORI Implementation for Mining Frequent Item sequences", OSDM05 Proc. of the 1st Int. Workshop on Open Source Data mining, pp. 56-65, August 21, Chicago, Illinois, USA, 2005.

[3] Bodon F., "A Fast APRIORI Implementation", Proceeding of FIMI'03, 19th Workshop on Frequent Itemset Mining Implementations. In conjonction with the 3rd IEEE International Conference on Data Mining, pp. 16-25, November 19, Melbourne, Florida, USA, 2003.

[4] Brin S., Motawni R., Ulman J.D., "Dynamic itemset counting and implication rules for market basket data", In Proceedings of the ACM SIGMOD, , pp. 255-264, May 11-15, Tucson, Arizona, USA, 1997.

[5] Chen Yangjun, Chen Yibin.,"On the Signature Tree Construction and Analysis", IEEE Transactions on Knowledge and Data Engineering, vol. 18, Issue 9, pp. 1207-1224, September 2006.

[6] Faloutsos C. "Signature Files: Design and Performance Comparaison of Some Signature Extraction Methods", ACM Sigmod Record, Volume 14, Issue 4, pp. 63 – 82, May 1985.

[7] FIMI repository. http://fimi.cs.helsinki.fi/data.

[8] Gardarin G., Ph. Pucheral, and F. Wu., "Bitmap based algorithms for mining association rules", In Proceedings of 14th Int. Conf. Bases de Données Avancées, pp. 157-175, Octobre 26-30, Hammamet, Tunisie, 1998.

[9] Han J., Pei J. and Yin Y., "Mining frequent patterns without candidate generation". In Proceedings of the 2000 ACM SIGMOD Int. Conf. on Management of Data, Dallas, pp. 1-12, May 14-19, 2000.

[10] Savesere A., Omiecinski E., Navathe S., "An efficient algorithm for mining association rules in large databases", In Proceedings of the 21th VLDB Conference, pp 432-444, September 11-15, Zurich, Switzerland,1995.

[11] Zandolin D. and Pietracaprina A., "Mining Frequent Itemsets using Patricia Tries", In Proceedings of the Workshop on Frequent Itemset Mining Implementations, FIMI03, vol. 90 of CEUR Workshop Proceedings, Melbourne, Florida, USA, 2003.

[12] Joshi S. and Jain R.C., "A Dynamic Approach for Frequent Pattern Mining Using Transposition of Database", In Proceeding of the International Conference on Communication Software and Networks (ICCSN'10), Feb 26-28, pp 498-501, Singapore, 2010.

[13] Zaki M., J. Parthasarathy, S. Ogihara and M., Li W., "New Algorithms for Fast Discovery of Association Rules", In proceeding of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD-97), pp 283-286, August 14–17, Newport Beach, California, USA, 1997.

Casino Fraud Data Mining

R. Woodley[1], **W. Noll**[1], **and K. Shallenberger** [1]
[1]21st Century Systems, Inc., 6825 Pine Street, Suite 141, Omaha, Nebraska, USA

Abstract - *Average revenue per casino hotel resort per year is $87,887,253 [1]. This much revenue attracts fraud and criminals leading to millions in lost revenue [2]. Recently, casinos have begun to track patrons. Their vital statistics and spending habits are all recorded in massive databases. This, however, has led to the challenge of extracting the pertinent information from these data sets and how to connect actions to fraud in causal chains. As data becomes more prevalent, the need to link causal data into actionable information becomes paramount. Analysts are faced with mountains of data, and finding that piece of relevant information is the proverbial needle in a haystack, only with dozens of haystacks. Analysis tools that facilitate identifying causal relationships across multiple data sets are sorely needed. 21st Century Systems, Inc. (21CSI) has initiated research called Causal-View, a causal data-mining visualization tool, to address this challenge. Causal-View provides causal analysis tools to fill the gaps in the causal chain. We present here the Causal-View concept, the initial research into data mining tools that assist in forming the causal relationships, and our initial findings.*

Keywords: Causal data mining, Casino fraud, Causal data relationships, Mahalanobis Taguchi System, Evidence Reasoning

1 Introduction

21st Century Systems, Inc. (21CSI) has been working with the Borgata Casino, Hotel and Spa in Atlantic City to develop a software package called Kaimi. We are developing Kaimi as a decision support tool to assist Casino surveillance teams with tracking the spending of patrons and "connect the dots" between disparate data sets in a uniform and integrated fashion to reduce revenue loss from fraud. The potential for combining data sets and discovering hidden patterns decreases the investigative time, allowing casino surveillance teams to further combat fraud at the casino. Kaimi was developed utilizing both new enabling technologies and previously researched algorithms. The system prototype was first deployed 90 days after the initial requirements meetings. The Kaimi capability assists casino personnel in answering the question, "Who is sitting at my gaming table?" Kaimi looks for players and employees who live near or with each other, players with the same or similar attributes, players who normally play together, betting patterns, and other possible relationships. The amount of data exceeds two million records updated multiple times throughout the day. To take Kaimi to the next level, we are adding the causal data reasoning capability of Causal-View.

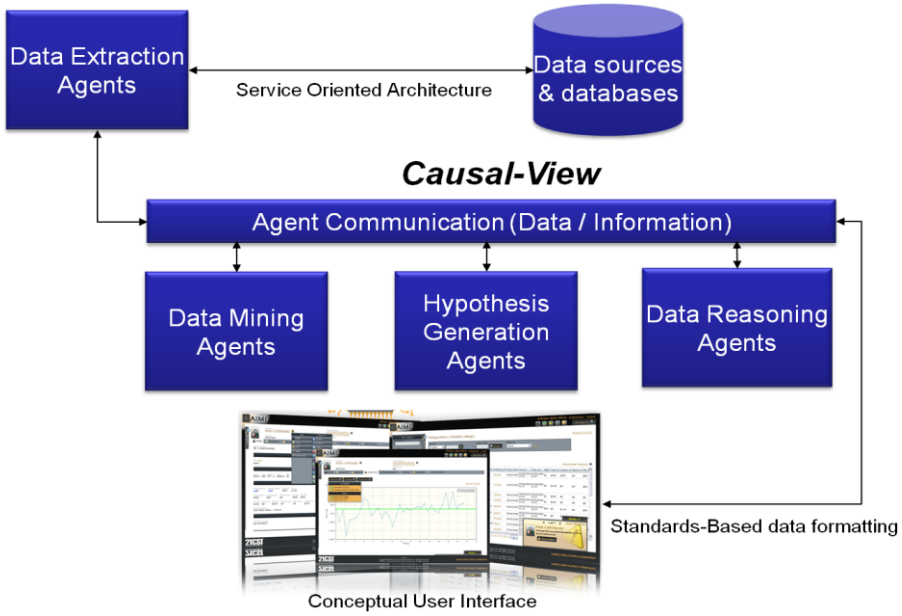


Figure 1: Causal View data flow diagram.

Causal-View is a causal data-mining visualization tool being developed for the U.S. Army. In Figure 1, we see a conceptual illustration of Causal-View. Causal-View is built on an agent-enabled framework. The purpose behind using an agent structure is that much of the processing that Causal-view will do is in the background. When a user makes a request for information, e.g., the betting history of a patron for a particular table, Data Extraction Agents launch to gather information. This initial search is a raw, Monte Carlo type search designed to gather everything available that may have relevance to the patron, the table, the dealers, and more. This data is then processed by Data-Mining Agents. The Data-Mining Agents are driven by user supplied feature parameters. For example, if the analyst is looking to see if the patron bets more or less for particular dealers the extraction agent can make a direct link. On the other hand, if the analyst is trying to see if there is a pattern in the patron's bet, the mining agent can be instructed with the type and relevance of the information fields to look at. The same data is extracted from the database, but the Data Mining Agents customize the feature set in order to determine causal relationships that the user is interested in. At this point, the Hypothesis Generation and Data Reasoning Agents take over to form conditional hypotheses about the data and pare the data, respectively. The newly formed information is then published to the agent communication backbone of Causal-View to be displayed in the Kaimi user interface.

2 Data Extraction, Hypothesis Generation, and Evidence Reasoning

As illustrated by Thearling [3], data mining as a science, growing out of the data collection and warehousing of the 1960s – 1980s, extends the ability of simple querying data into guided search and information discovery. Causal Data Mining (CDM) extends typical data mining even further as shown by Silverstein, et.al. [4]. Silverstein shows that mining association rules is quite complex, particularly in unstructured data. The association rule of "X implies Y" can often be misinterpreted from the raw data and that additional data and analysis is needed to justify the rule. The Bayesian techniques in [4] do a reasonable job when enough information is known about the data to form the *a priori* conditional probabilities that drive the Bayesian network. In our concept, we will try to extend this work by adding technology (evidential reasoning) that intrinsically handles the uncertainty created by the data-driven association rule without the need for calculating the conditional probabilities. We also move past the Bayesian approach utilizing the Mahalanobis-Taguchi System (MTS) for data clustering.

2.1 Data Extraction

From **Error! Reference source not found.**, the first agent the data encounters is the extraction agent. The extraction agent is the difference between a database query and autonomous data extraction. We need to give the agent the capability of discovering important pieces of data in both structured and unstructured data. The data the agent finds will trigger the reasoning engine to create a hypothesis. The hypothesis, in turn, causes further data mining and trending agents to find corroborating evidence until a consensus is met.

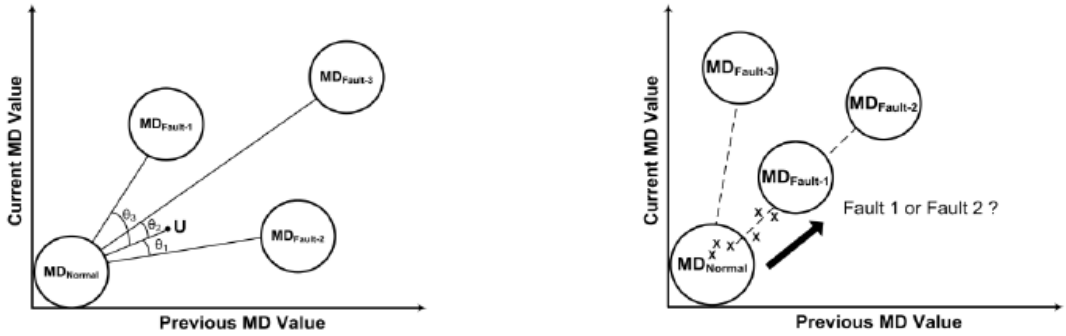
Information extraction techniques vary by domain and source of the data. Statistical mechanisms and manual annotations are commonly used on unstructured information, while less linguistically intensive approaches have been developed for the Internet using rule-based approaches that are aware of a particular page's content format. Statistical techniques include Maximum Entropy [5], Support Vector Machines [6], Hidden Markov Models [7] (HMMs), while information extraction techniques on the web generally deal with the structured HTML/XHTML content that can be reused on a site-to-site basis, or Resource Description Framework (RDF) feeds.

We implemented a metasearch engine capable of connecting to existing data sources and providing relevant results from search queries. In addition to the metasearch engine, a distributed search engine capable of crawling and searching file systems, Intranet sites, etc. may be utilized.

2.2 Hypothesis Generation – Data Clustering

The primary research for the Causal-View project has been in the Hypothesis Generation. Unlike most data clustering, causal data mining rarely has ground truth by which you may train a clustering algorithm. However, the nature of the data allows us to make some assumptions that we can use to create clusters. Primarily, given the enormous volume of gambling action that occurs at the casino, it is unlikely that, for a given period of time, any fraud is occurring. This allows us to take a small subset as a normal example and then compare against the larger data set where an anomaly may or may not exist.

What we needed was a method that could be easily reconfigured for different parameters of the data, use only a small subset to "train," and provide results that are readily discernable. A candidate algorithm for the data clustering challenge was Mahalanobis-Taguchi System (MTS) [8]. MTS is a fault detection, isolation, and prognostics scheme. Currently, MTS fuses data from multiple sensors into a single system-level performance metric using Mahalanobis Distance (MD) and generates clusters based on MD values. MD thresholds derived from the clustering analysis are used for detection and isolation. We are investigating the extension of the MTS scheme into causal mining. At present, the cluster identification is performed manually off-line. We are researching self-learning to generate the cluster heads for MTS. A conceptual example of MTS for fault detection is shown in Figure 2(a) whereby the MD (magnitude and angle)



(a) (b)

Figure 2: MTS where the Mahalanobis distance around a fault cluster determines the variance from normal (lower left corner) for simple fault conditions (a) and compound fault conditions (b).

can help detect that a fault is occurring and which type of fault (root cause). Figure 2(b) shows the same concept with a compound fault. In this case, either Fault 1 or Fault 2 may be indicated by the MD. In particular, a change in parameters would be needed to properly identify the fault. By creating a self-learning scheme, the proper faults can be identified, and, more importantly, which parameters to use to separate the faults. This type of information can then be used to alert the user that more information is needed.

Figure 3 shows a physical example of a compound fault that is indistinguishable using only outlet pressure on a pump [9].

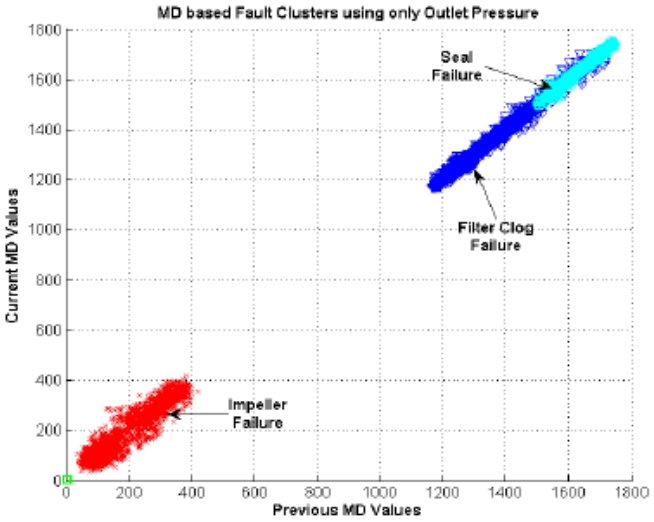


Figure 3: Example MD based fault clusters using only the outlet pressure for a pump.

2.3 Data Reasoning

The final component of the Causal-View system is reasoning about the information clusters. We employ technology called the Evidential Reasoning Network (ERN®) to assist in the data reasoning. The goal of the ERN component is to indicate the uncertainty about a particular hypothesis when compared against other hypotheses for the same data. By minimizing the uncertainty, we can provide a clear indication of the believability (or, conversely, the disbelief) about a hypothesis. This should then lead to a causal chain of evidence, if such a chain exists. Typical decision-support approaches will use either a simplistic uncertainty tracking method or something along the lines of a Bayesian probability approach. Simple uncertainty tracking does not fully account for the propagation and combination of uncertainty. It does not propagate the error whereby it may allow potentially erroneous data to bias the results. Bayesian approaches are better and account for the error propagation, but have the basic need of *a priori* probability measures on the uncertain elements. What is sometimes needed is a way to incorporate various degrees of uncertainty ranging from simple percent unknown up to probabilistic measures, where available. The ERN technology is designed for this purpose.

The ERN technology uses a belief algebra structure for providing a mathematically rigorous representation and manipulation of uncertainty within the evidential reasoning network. Since the introduction of the Dempster-Shafer Theory of Evidence [10], new evidential reasoning methods have been, and continue to be, developed, including fuzzy logic [11] and Subjective Logic [12], [13]. An evidential reasoning framework was needed to ensure that evidential reasoning expressions are coherent, consistent, and computationally tractable. ERN is a novel structure that addresses these needs. The two prime belief algebra operators required are *consensus* and *discount*. These operators allow the propagation of belief values through the network amongst various opinion generating authorities, i.e., the clustering generated in MTS, which perform some sort of data analysis, processing, and reasoning. The belief algebra structure is capable of using probabilistic belief mass assignments through the use of belief frames. The ERN Toolkit includes a Subjective Logic and Dempster-Shafer belief algebra implementation.

The belief algebra equations direct ERN how to combine information. A consensus operator is an additive function that increases assenting opinion, where the discount operator is multiplicative and will act to attenuate dissenting opinion due to the normalized opinion values found within

the opinion-space used by ERN. The implementation of the belief algebra is currently under development; results from this research are expected shortly.

Our plan for the assigning of opinion and the subsequent belief algebra equations would be similar to a nearest-neighbor approach, only with much more information and sophistication in how the opinions are grouped. Using the calculated MD value as the measure, we will assign a belief value that corresponds to known results (e.g., that a player is winning/losing within expected ranges). An uncertainty value can then be calculated based on changes in behavior. These values form the initial opinion concerning the likelihood of fraud. As other events occur (e.g., additional play by the patron, or wins/losses by neighboring players), they likewise generate opinion. By then combining these events within the opinion space we can determine if the player is indeed acting normally, or if potential fraud may be occurring. As more events occur in the same location in either the MD space or the opinion space, a causal relationship between action and result can be determined.

3 Results

As mentioned previously, this is an active research project in its preliminary stages. The results, thus far, show the use of the MTS algorithm on the casino data. The data set has over 2 million entries where a typical data subset is shown in Table 1.

Table 1: Data subset of the casino data.

avgbet	theo	shift	hours	totalin	estwl
1	1	1	0.0167	5000	0
1	1	1	0.0333	5000	0
1	1	1	0.0167	1000	-1000
1	1	2	0.0167	2000	0
50	6	2	0.2167	2000	-1200
100	1	1	0.0167	2800	0
100	1	2	0.0167	0	500
100	1	1	0.0167	1100	100
100	1	1	0.0167	2100	-100
100	1	1	0.0167	0	700
100	1	1	0.0167	2500	-300
100	2	1	0.0333	600	-300
100	2	1	0.0333	300	-300
100	2	2	0.0333	0	800
100	2	1	0.0333	500	-200
100	2	1	0.0333	500	-100
100	2	2	0.0333	700	-300
100	3	1	0.05	1500	-200
100	3	2	0.05	200	-200
100	3	2	0.05	0	100

Table 1 contains the first twenty entries (of over 500 for this player on this particular table) for a player's average bet (*avgbet*), the theoretical win/loss of the casino (*theo* – the amount of money the casino should've won based on house advantage of the game, pace of play, *avgbet*, and hours), which work shift the player was at the game (shift), the number of hours played at the game (hours), the total amount bet (*totalin*), and the estimated win/loss (*estwl* – an estimate of money going to the casino during the session). There are many more columns in the full data set that indicate the specific times played, the dealer during the time, and any other piece of information that can be captured at the time the player swipes his ID card at the game. The data shown is currently sorted by average bet, but still shows some of the volatility in the data. Part of our challenge with this data is that we do not have the ground truth concerning if any fraud has occurred in the data or not.

As a first pass on the data, we were interested to see how our sample player compared against other players at the same table. We began by selecting only the player for that table and calculate the MD values. The amount of variation in even this small example is extreme. Depending how the data is sorted, what variables are used in the MD calculations, and the weighting factor can influence how the data is clustered. For this initial pass, we double sorted the data by *avgbet* and *theo*. You can see from Table 1 that there is still the possibility of some large variance in the data even though it is sorted. Figure 4 shows that the player has a fairly consistent betting history with only a few points that fall outside the main cluster at the origin (from the scatter plot). The line graphs indicate that the user has a general trend in that he prefers mid-size wagers slightly more than small wagers (as indicated by the MD distance near the center of the green line graph) and a large preference over large wagers (very large MD values at the upper end).

We next ran the same configuration (i.e., same game table, for the same overall time period as our example player) to see how his betting history compared against all other patrons. Figure 5 shows the results against all other players. We see from this data that almost all other players are very close to our example player in their betting preferences as indicated by the tight clustering occurring at the origin. However, there are many outliers that may indicate that significant differences are present. At this time, we have no ground-truth information to indicate that fraud is occurring, but this gives the analyst a much more narrowed field of entries to investigate. Future work will allow the analyst to simply click on the outliers to get more information concerning these events. Furthermore, we can now begin to apply the ERN technology to compare the results. For example, if a particular grouping of outliers has a set of similar characteristics, we can form a belief space opinion cluster. If further analysis shows that the cluster indicates possible suspicious behavior, we can trigger an alert in the monitoring software the next time a patron exhibits the same behavior. The causal chain is now formed that says if Patron A exhibits Activity B, then the possible fraud has probability C. This information can give security personnel the information they need to catch the perpetrator.

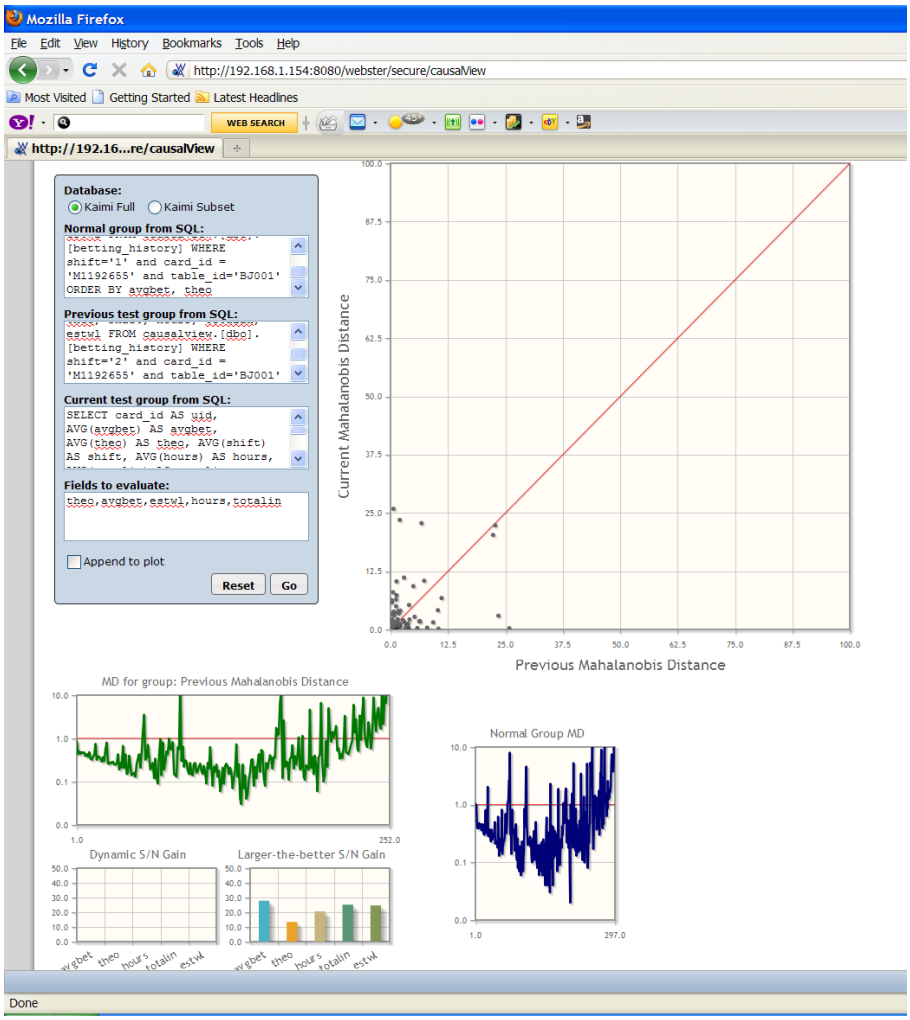


Figure 4: MD clustering for example user.

4 Conclusions

In this paper, we have presented the conceptual idea of Causal-View. Causal-View is a causal data mining engine that finds relationships hidden within extremely large data sets. Furthermore, we present here the underlying technology with a discussion of how the components work together to form the causal chains. We use a Monte Carlo data gathering scheme to pull data in from whatever sources are available. We are experimenting with MTS to find potential hypotheses. The hypotheses are then evaluated leading to the final causal chain by 21CSI's ERN technology. While the concept shows promise, the work is only in its preliminary state. We present an initial example problem in which we were able to find some relationships (and differences) between a particular patron's activity and all other patrons on a particular game. We are confident that this technology will continue to develop in a positive manner.

Our future work will include improvements in the algorithms and the user interface. We will be giving the analyst the ability to query the results of the clustering action. We are researching methods to configure the MTS output to provide multiple "views" of the data to help the reasoning engine to discover patterns. Finally, we are developing the belief algebra equations that pull the hypotheses into a causal chain.

5 Acknowledgment

21st Century Systems, Inc. would like to thank the U.S. Army for sponsoring this research. (contract number: W15P7T-11-C-H217). We also thank the Borgata Casino, Hotel and Spa in Atlantic City for allowing us to analyze their data.

6 References

[1] G. Haussman, "Nevada Reaps $2.1 Billion in Casino Profit Casino resorts throughout the state celebrate the New Year with record profits and strong indications of a robust 2007., 'http://www.hotelinteractive.com/article.aspx?articleID=6854'," *Hotel Interactive*, 08-Jan-2007.

[2] The Executive Office of the Governor, "Casinos in Florida: An analysis of the Economic and Social Impacts, 'http://casinowatch.org/loss_limit/casinos_florida.html'."Office of Planning and Budgeting, The Capitol, Tallahassee FL., 2007.

[3] K. Thearling, *An Introduction to Data Mining*. http://www.thearling.com/text/dmwhite/dmwhite.htm: .

[4] C. Silverstein, S. Brin, R. Motwani, and J. Ullman, "Scalable techniques for mining causal structures," *Data Mining and Knowledge Discovery*, vol. 4, no. 2, p. 163–192, 2000.

[5] A. E. Borthwick, "A maximum entropy approach to named entity recognition," New York University, 1999.

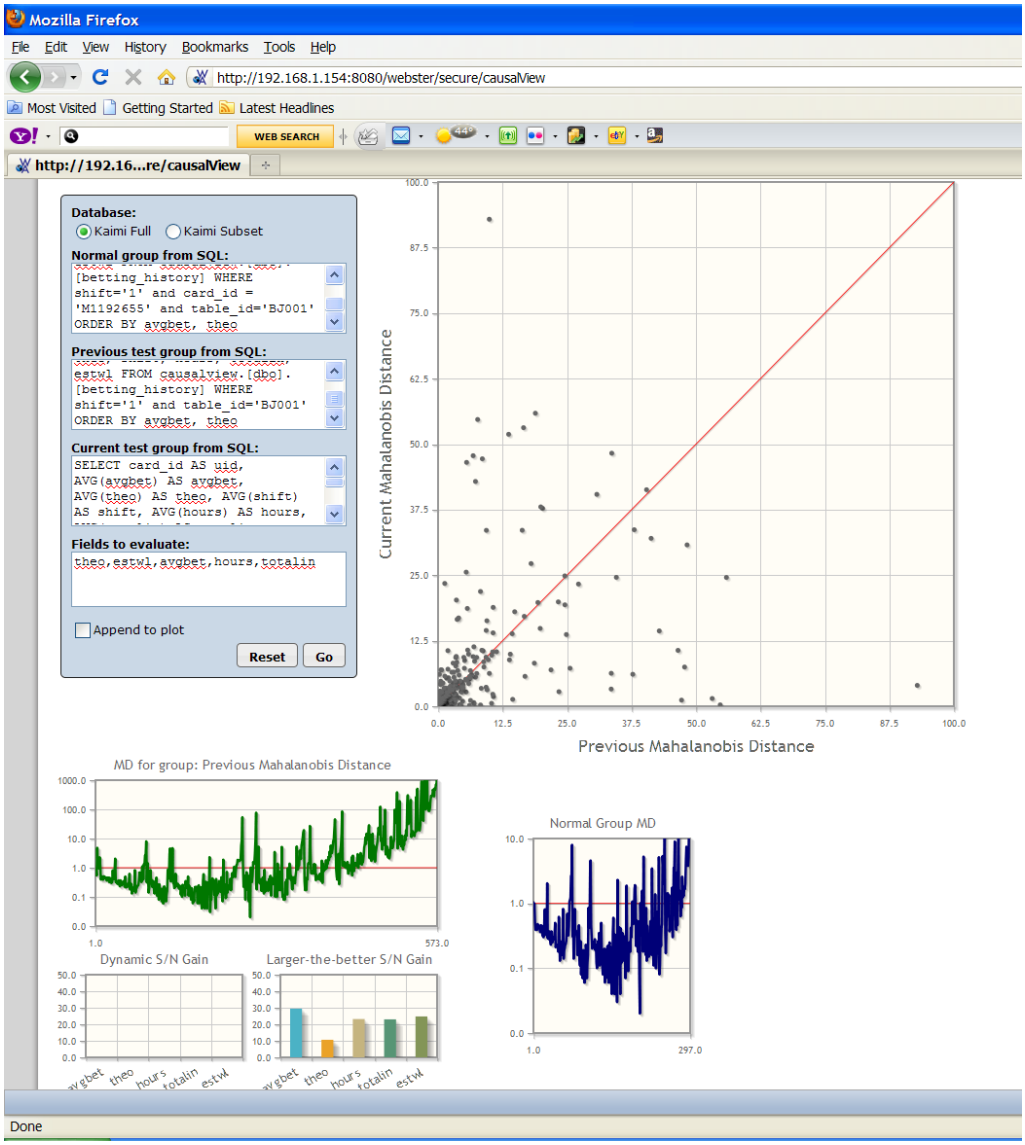


Figure 5: Comparison of player 1 versus all other patrons.

[6] T. Joachims, F. Informatik, F. Informatik, F. Informatik, F. Informatik, and L. Viii, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," 1997.

[7] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel, "Nymble: a High-Performance Learning Name-finder," *IN PROCEEDINGS OF THE FIFTH CONFERENCE ON APPLIED NATURAL LANGUAGE PROCESSING*, p. 194--201, 1997.

[8] G. Taguchi, S. Chowdhury, and Y. Wu, *The Mahalanobis-Taguchi System*. McGraw-Hill Professional, 2001.

[9] Soylemezoglu, Ahmet, "Sensor Data-Based Decision Making," Missouri University of Science and Technology, dissertation, 2010.

[10] G. Shafer, *A mathematical theory of evidence*. Princeton NJ: Princeton University Press, 1976.

[11] P. Palacharla and P. Nelson, "Understanding relations between fuzzy logic and evidential reasoning methods," in *IEEE Proceedings of the Third IEEE Conference on World Congress on Computational Intelligence*, 1994, pp. 1933-1938.

[12] A. Jøsang, "A Logic for Uncertain Probabilities," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. 3, pp. 279-311, Jun. 2001.

[13] A. Jøsang, "Subjective Evidential Reasoning," *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2002)*, p. 1671--1678, Jul. 2002.

A Case Study on Clustering and Mining Business Processes from a University

Pedro Miguel Esposito[1], **Marco Aniceto Vaz**[1], **Jano Moreira de Souza**[1], **and Luciano Terres**[2]
[1]COPPE, Federal University of Rio de Janeiro, Rio de Janeiro, RJ, Brazil
[2]Petrobras, Rio de Janeiro, RJ, Brazil

Abstract - *Business process modeling is an important activity for enterprises that wish to remain competitive and to build strategic advantage over their competitors. Nonprofit organizations, such as public universities, also have much to gain from the analysis of their internal processes, enhancing their effectiveness and allowing the visualization of hidden strategic goals. A process modeling and analysis project, however, is extremely complex, and demands a cost that may be perceived as above its benefits. This paper proposes the use of process mining and clustering techniques to map a university's processes, lowering the project's costs and enhancing its return. A case study is shown using the diploma registration process, which receives complaints due to its low efficiency. A two-level clustering is proposed, the first using each task's executor unities as relevant features, and the second using each resulting cluster's task flow to extract clustering attributes.*

Keywords: Cluster Analysis, Process Mining, University Processes

1 Introduction

Business processes are a vital component of any institution, being an important strategic asset that needs to be managed. They need to be correctly defined, understood and documented, to be executed in a uniform and consistent manner. Tacit processes are only in the minds of their executors. When made explicit, they become intellectual property of the entire organization [1].

Business process modeling is the task of eliciting, documenting and analyzing an enterprise's internal procedures. For complex processes, their design involves the establishment of a costly project, requiring a large amount of time, money and other resources. Several organizations consider this spending to be economically unfeasible [2]. This issue is especially relevant on non-profit organizations, such as public universities, which are focus of this paper. The field of process mining appears as an elegant solution to this, with several techniques being proposed for the extraction of *as-is* models from transactional logs. This approach is low-cost and brings faster results than traditional modeling techniques.

The mining of business processes assumes that it is possible to obtain a task flow from a transactional log. A log contains real data from the execution of process, and can be extracted from information systems that support these processes. It contains events, each being related to a process instance, representing the execution of a task, at a moment in time, by an executor [3]. This information is enough to reconstruct a model for the flow of the process.

Although traditional process mining techniques have good results for structured processes, they fail for unstructured processes, or when there are not strong relationships between their activities [4]. This issue results in spaghetti models [5], which contain a huge amount of edges. An example of a spaghetti model is shown in figure 1. It contains only a small portion of a real mined graph. These models are extremely complex to understand and have little meaning to business analysts.

One of the main causes of spaghetti models is the junction of several types of flow into a single model. Several clustering techniques have been proposed in the literature to deal with this case. They split the log into smaller clusters of instances that have high internal similarity.

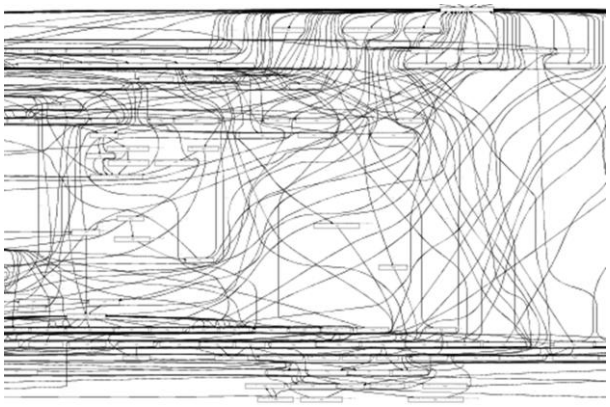


Fig. 1. Spaghetti model excerpt.

Academic and administrative processes at universities are typically unstructured and would greatly benefit from the use of clustering techniques. Examples of these processes include diploma registration, scholarship allocation, creation of new courses and student admission.

Their tasks are executed through several organizational units, where the knowledge about the process is tacit, being strongly associated to their employees. The substitution someone central to the task, for example, will bring changes to the task flow of the processes. These changes may include real workflow distinctions or just different ways of registering information in the supporting system, which will impact the resulting event log. The great decentralization between several centers of the university also contributes to a chaotic process model.

Process modeling and analysis projects are rare at universities, due to their high complexity and low perception of return. This is aggravated at public universities, which are non-profit organizations and where the concept of return is hard to define [6]. The increased operational efficiency caused by the analysis of its processes, however, can have a large impact on the academic community and on the society as a whole. Explicit processes should be regarded as a high valued strategic asset, in the same way it happens at other organizations.

The rest of this paper is organized as follows. Section 2 covers the main aspects of process mining. Section 3 discusses the application of process mining at universities. Section 4 introduces the approach used to extract, cluster and mine processes. Section 5 shows the results of this work. Section 6 concludes this paper and covers future directions.

2 Process Mining

Process can be defined as the manner enterprises organize their resources. This includes their workflow, which is split into tasks, and who is responsible for each task. When there is no formalization of a process, its execution is based on tacit knowledge from its executors. More and more organizations are going through process modeling and analysis projects, expecting to eliminate this issue. However, even if a process is well-documented, the actual task flow many not correspond to what was expected [7].

Even if a process is not documented or its execution doesn't reflect reality, it may be possible to extract its task flow from information systems. These systems record, usually on transactional logs, the occurrence of events that are related to the tasks of a process.

The goal of process mining is to reverse engineer these execution logs, using them to support business process modeling and analysis activities [8]. There are three basic types of process mining: discovery, when a model is built from an execution log; conformity, when tests are made to check if a predefined model corresponds to the reality represented on the log; and extension, with the objective to enrich a model with existing data on the log [4]. This paper focuses on the first type, the discovery.

An example of the main information that can ve extracted from an execution log can be seen in table 1. Each event of the log (table rows) contains information about the process instance related to it; the task that it represents; who was responsible for its execution; and the moment when the event was registered. Extra information can also be extracted from a log. One example would be marking an event as the beginning or the completion of a task, allowing refined analyses.

2.1 ProM

ProM [3] is a framework with the goal of unifying the main existing process mining algorithms. It is developed in Java, implementing a plug-in system that allows its extension with new techniques. Today it has more than 230 plug-ins available [9], allowing the selection of the best solutions for each case.

The ProM framework represents the state of the art in the field, containing its most recent and advanced techniques and algorithms. Considering all this, it was used as the basis for the mining of the processes studied in this paper.

The framework contains two main data input formats, both XML-based: MXML [10] and XES [11]. MXML is a well-established model used by most of the publications in the field. Its structure defines a log, which contains a process. A process is composed of several instances. Each instance has several audit trail entries, each representing an event registered on the log. An audit trail entry contains the task that was executed, its originator, its event type (start of completion of the task, for example) and a timestamp. Additional information may be inserted on each node of the XML tree [10].

XES is another log interchange model, more generic than MXML. It was developing with the goals of simplicity, flexibility, extensibility and expressivity. Its structure defines a log, composed of traces process instances, which are composed of events. Each basic component can have several attributes, which are defined through model extensions. [11]. The model aims for the ability to express any use case that may need the recording and exchange of information extracted from an execution log.

TABLE 1
WORKFLOW LOG EXAMPLE

Instance	Task	Executor	Timestamp
1	Task 1	John	11/01/2010 12:01
1	Task 2	Paul	11/02/2010 15:25
2	Task 1	John	11/10/2010 17:33
2	Task 2	Peter	11/12/2010 22:12
1	Task 3	Joseph	12/03/2010 10:32

As ProM imports data from both formats, we chose to use the MXML format. It supports all data needed for the execution of this work, in addition to having a simpler

structure and greater adoption in the literature.

The main discovery algorithms implemented in the ProM tool include the α algorithm [12] and the heuristics miner [7]. For this work the heuristics miner was used, due to its robustness relative to noise on real-life logs. It is based on the construction of a dependency graph, having a metric that indicates the chance that a relationship between two tasks exists. Detailed comparisons between several process mining algorithms can be found in [12], [7], [8] and [13].

2.2 Process Clustering

For unstructured processes, traditional process mining techniques build over-generalized models [5], allowing too many flows between tasks. These models are hard to understand, and have little utility to business analysts. Several clustering techniques have been proposed in an attempt to solve this issue. They presuppose that it is possible to split an execution log into smaller instance clusters having high inner similarity. This way, a complex problem is divided into smaller, simpler ones.

Most process clustering algorithms search for a set of features meaningful to the domain. They are used as input for traditional clustering techniques (e.g. k-means). The results are groups of instances, which are then mined using other process mining algorithms. In this section, some of the main process clustering algorithms will be introduced.

The DWS (Disjunctive Workflow Schema) [14] searches for task sequences whose frequencies are lower than that of their subsequences. They detect overgeneralizations in the model, and are recursively used with the k-means algorithm, until no more overgeneralization is found. The models for each resulting cluster are generated using the heuristics miner algorithm.

The trace clustering algorithm [15] defines several profiles for the extraction of features, representing different perspectives of the log. The clustering is then executed using traditional techniques. The paper suggests a profile that uses each task's originator (e.g. organizational unit or person) as dimensions. This approach is similar to the first clustering used in this paper. However, the case study presented on [15] uses only the activity profile, which creates a clustering attribute for each task in the process.

The sequence clustering algorithm [5] introduces an approach common in the bioinformatics field, based on Markov chains. Instead of extracting meaningful features from the sequences, as other algorithms do, the clustering occurs over the sequences themselves. The fuzzy miner [4], instead of clustering process instances, clusters tasks contained in the same model. These clusters can then be zoomed-in to show their inner tasks. It uses several metrics which represent significance or correlation values. These metrics are based on several process perspectives. Eventually, [16] presents a clustering algorithm based on the edit distance between two sequences.

3 Process Mining Applied to Universities

This paper presents an approach for the mining of administrative processes from a public university, through the use of cluster analysis. The data used was gathered from one of the largest universities in Brazil, which is currently implementing an IT governance model. The modeling of business processes falls within the scope of this project.

An IT governance model must be grounded in organizational transparency and be aligned with the institution's strategy. It must support all administrative and academic activities of the university. The process mapping and optimization is critical so that this vision can be accomplished. It must be done through a reflection and debate exercise, so that internal information flows, the responsibility for each task and the deliverables from each process can be fully mapped. Every distortion in the processes must be found and dealt with [17].

The goal of this work is to maximize the efficiency and effectiveness of the university's processes, with the drawing of its tasks, the flows between them (actual and desired) and the discovery of the organizational units that are involved on each process [17].

The modeling of processes is important so that needed changes on the university's information systems can be identified. The cause of an inefficient process may be a badly designed support system, resulting on delays, high costs and user frustration [17]. Work methodologies outside support systems can also be optimized.

3.1 Process Monitoring System

This work used data from the Process Monitoring System of the studied university. It supports the input and monitoring of all academic processes and their tasks. At the time of data extraction, the database contained roughly 1.700.000 processes.

Each process instance contains a subject attribute, selected between 242 predefined values. Each instance also contains a summary, which covers information not captured by the subject field, a registration date and the stakeholder of the process. Each instance also contains several transitions, indicating when it was sent from one unit to another. Each transition contains a dispatching order. Although the database includes 76 types of dispatching orders, they have poor semantic, such as "to register" or "to attend". The relationship between all entities is shown in figure 2.

3.2 Diploma Registration Process

The diploma registration process was selected as the object of this study. It is the subject in the database with the

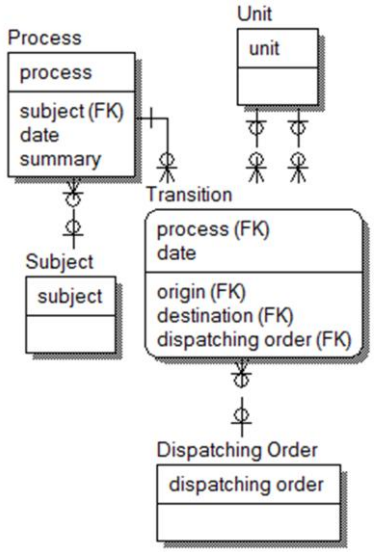


Fig. 2. Simplified data model for the Process Monitoring System

largest number of instances. We considered only the period between 1996 and 2010, totalizing about 65.000 instances. This process is inefficient, suffering frequent complaints from students due to the long wait time until their diplomas are ready after graduating.

The diploma registration process is an interesting study case due to the highly complex analysis it demands. It includes the participation of several units throughout the university. Some units, however, only participate on the emission of diplomas from specific schools, such as the Engineering School or the Medical School. Other units are central to the process, and execute tasks on all of the university's diplomas. Altogether, about 400 units execute tasks for the diploma generation process. Their separation and hierarchy is not well defined in the database, and they range from entire schools to specific offices.

Moreover, this process is unstructured, containing several exceptions and variations on the task flow, even for instances with similar features. There are several causes for this, including the lack of a formal process model and a loose information system supporting the process.

4 Approach Description

The goal of this paper is to define an approach for the modeling of a university's the administrative processes. This task was accomplished through some key decisions and the usage of a group of techniques that will be discussed in this section. Although only the diploma registration process was selected, this approach can be easily used for the analysis of other processes.

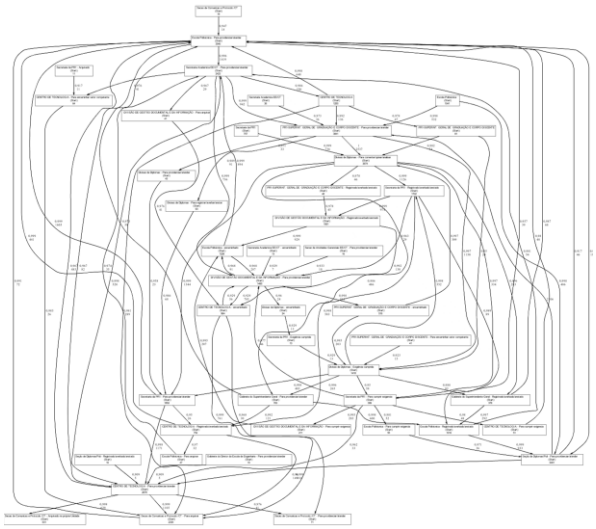
The data extraction and filtering was done using the Mana tool, which is currently under development. It imports process instances from several sources into a canonical database. These instances can be later filtered into views based on filters. Several filters are implemented, such as process type, date and task relevance. Process views can then be clustered, mined into process models and visualized. The clustering of data uses the library from the Weka [18] tool. The mining of processes used the ProM framework, which was detailed on section 2.

The initial problem met was the naming of the tasks. A task's name should represent a meaningful piece of work made into the process. The first candidate for that was the dispatching order of each process transition. However, as said before, the dispatching order naming is too generic, such as "to register" or "to attend". It doesn't reflect with precision the tasks executed. For example, several processes would execute the "to attend" task four times in a row. The resulting process model from this would be nearly useless. In the studied domain, the organizational unit that executed a task is as important as its dispatching order. Having this in mind, a task's name was defined as the concatenation between the executor unit and the dispatching order, e.g. "Engineering School – To Attend".

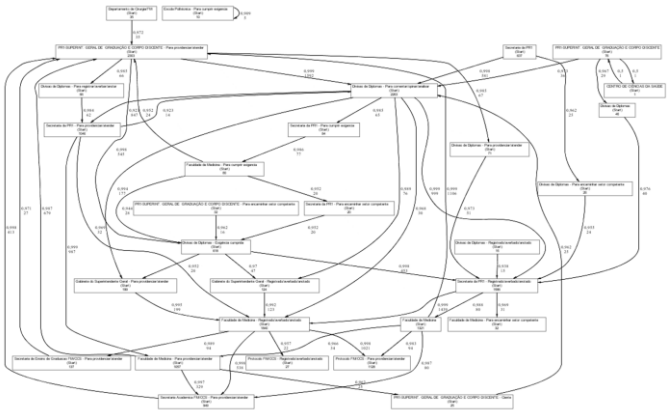
The second issue found was the large amount of tasks and connections of this process, resulting in a spaghetti model with no practical value. Figure 1 shows a small part of the model resulting from the mining of all diploma registration processes, using the heuristics miner. To solve this issue, we adopted four preprocessing solutions: filtering through the "summary" field, outlier removal, unit clustering and DWS clustering.

The summary field contains extra information entered upon the registration of process' instance, complementing the subject field. Some common examples are "diploma kit" and "electrical engineering graduation". In an attempt to simplify the process analysis problem, instances marked as doctorate, master's degree, copy request or change request were removed from the database. This way, only the undergraduate diplomas, which represent most instances, were considered.

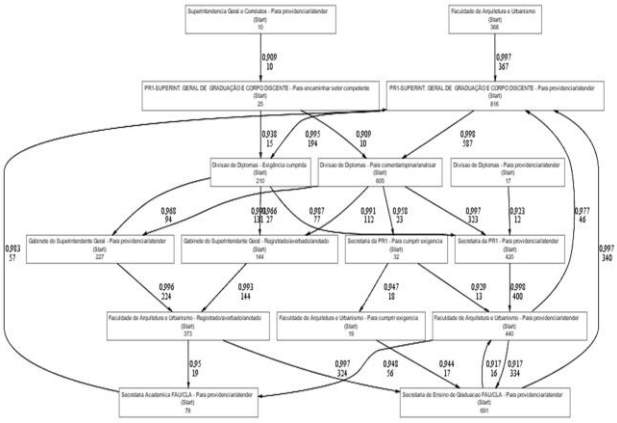
Another filtering used was the removal of outliers. They represent low frequency flows that may appear from rare exceptions or input error. Outlier instances aren't useful for this study, since its goal is to model the main task flows contained in the process. The detection of outliers, however, is not trivial, and a detailed analysis is out of the scope of this paper. So we opted to eliminate only extreme cases, instances containing tasks present on less than 0.015% of all instances, or 10 instances. This number was considered safe, as the database contains 65.000 instances. This filter was very effective, removing half of the tasks contained in the process.



(a) Engineering School



(b) Medical School



(c) Architecture School

Fig. 3. Sample models from first clustering (by organizational unit)

The main preprocessing solution, whose results are detailed in the next section, was the clustering of process instances using their executor units as relevant features. All units participating on the diploma registration process were regarded as clustering attributes. An excerpt of the resulting table is shown in table 2. Each line represents a process instance, and the value for each field indicates the presence of absence of the corresponding unit as the executor of a task for that instance. This table was then used as input for the k-means algorithm, configured to find 20 clusters. This value was obtained through experimentation, achieving good results for the goal of this work. We should highlight that an attempt was made to cluster the process instances using only their creator unities. However, 207 units start this process, many of them related to the same schools. A better clustering precision was reached when every unit participating in the process was considered. The goal of this analysis was to create clusters that correspond each to one school of the university.

The final analysis was executed to detect task flow variations between processes of the same school. The resulting cluster for the Engineering School diplomas was clustered a second time, using ProM's implementation of the DWS algorithm, since it detects task flow variations. The results were then plotted by year, to support the hypothesis that task flow distinctions may be caused by process evolutions through time.

TABLE 2
CLUSTERING DATA BY UNIT

Instance	Eng. School	Technology Center	Medical School	Diploma Division
1	1	1	0	1
2	0	0	0	1
3	0	0	1	1
4	0	1	0	1

5 Results

5.1 Clustering Diplomas by Organizational Unit

The first clustering was performed using the organizational units participating in the process as instance

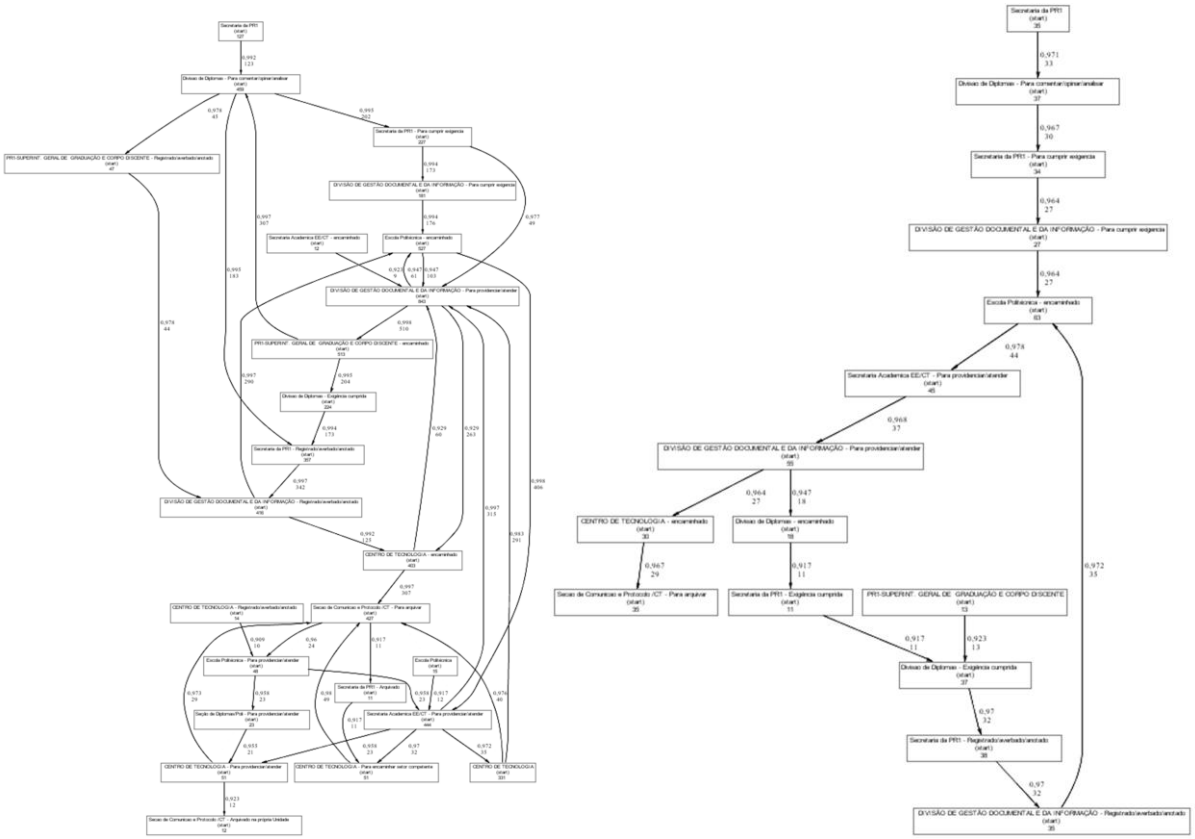


Fig. 4. Sample models from second clustering (DWS algorithm over Engineering School results)

attributes, as detailed in the last section. The k-means algorithm was used, being configured for 20 clusters. This value was sufficient to isolate several of the university's schools, reaching the desired goal. The resulting clusters were than mined using the heuristics miner algorithm. Figure 3 shows the resulting process models for the Engineering School, the Medical School and the Architecture School.

We can see that these models have different complexity degrees. While the Architecture School's process is fairly structured and useful to business analysts, the Engineering School's process is still unstructured, with too many flows and with little utility for the business.

5.2 Second Level Clustering – DWS Algorithm

To solve the Engineering School's spaghetti model, its instances went through a second clustering. This step can be done for all unsatisfactory models resulting from the first clustering. Since there are no explicit domain features to select, as occurred in the first clustering, the DWS algorithm was used. It is one of the main clustering algorithms implemented in ProM, and was described in further detail in section 2. It was configured for 5 clusters, which was enough to greatly simplify the process. On a deeper study, however, the process may be split further. Figure 4 shows two of the resulting clusters. They show how effective the clustering was, when compared to the model in figure 3(a).

This analysis is important to show how a process can have several distinct flows, even inside the same school. The lack of a defined process, without proper documentation and/or an information system that makes the task flow explicit is the main cause for this. Some of its side effects are input errors and different work methodologies for each employee. Non-documented evolutions in the process may also be a cause, either due to personnel substitution or due to process optimizations. This tendency can be seen in figure 5. It shows the number of each year's diploma registrations that fell into each cluster, for the Engineering School. Two groups were made: one for clusters 1, 2 and 3, and another for clusters 0 and 4. This graph shows a predominance of the first group until 2007 and of the second group after that. However, a deeper study is needed to find further variation causes, and how each one affects the process.

6 Conclusion and Future Work

This paper presented an approach for the extraction of *as-is* models from a university's administrative processes, using as a target the diploma registration process. It was shown that as the whole process is impossible to be

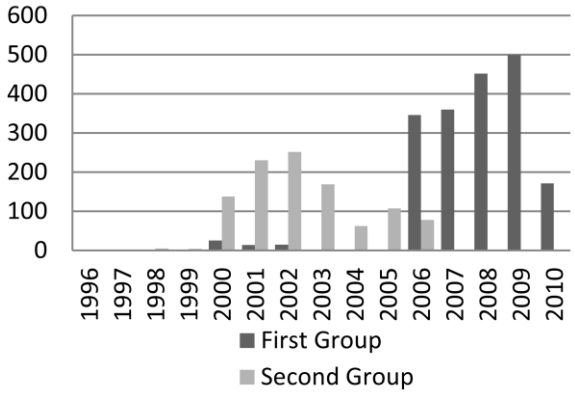


Fig. 5. Number of instances for each cluster group, by year.

analyzed; it is necessary to split the problem into smaller parts, using a divide and conquer technique. This division was done in two levels. First instances related to the several schools that compose the university were split into clusters, using their task's originators as attributes. The second level clustering used the DWS algorithm, splitting the process instances analyzing their task flows to extract features.

Further study is needed to explicit further causes for the task flow variability, which may be originated from process evolutions, as demonstrated, or from side effects of the lack of proper documentation and of the high freedom allowed to the process.

This work motivates further study on several aspects of the problem. First, an exhaustive work is needed to map not only the diploma registration process, but every other process in the university. These processes may be stored not only in the monitoring system being studied, but also in other information systems. A process may also have logs stored in more than one system, with tasks being executed on several of them. A consolidation project would be needed to gather all this information and to connect related process instances.

Another future direction would be the analysis of the "summary" field available for each process instance, which stores additional information about it. Text mining techniques may be used to obtain more precise filters. The traceability between the summary of an instance and its resulting task flow could also be studied.

This paper presented a clustering of the diploma registration process based on the executor units of its tasks. A more detailed analysis could be achieved through the construction of an ontology containing the relationship between every unit in the university. In [19] is presented a study about the usage of semantic data for the mining of business processes. This ontology could then be used to build a refined clustering of the process.

Finally, the refactoring of an institution's business process must define of to-be processes, which represents the ideal structure for each process. To this goal, interviews must be conducted with the stakeholders of the system, using the mined results as a baseline. Performance analysis techniques must also be used to detect inefficient tasks or bottlenecks in the processes. Every improvement must be prioritized according to their impact to achieve the university's strategic goals.

7 References

[1] M. Schedlbauer, *The Art of Business Process Modeling: The Business Analyst's Guide to Process Modeling with UML & BPMN*. CreateSpace, 2010.

[2] G. Greco, A. Guzzo, L. Pontieri, e D. Sacca, "Discovering Expressive Process Models by Clustering Log Traces", *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, p. 1010–1027, ago. 2006.

[3] B. F. Van Dongen, A. K. A. De Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, e W. M. P. van der Aalst, "The ProM framework: A new era in process mining tool support", in *Lecture Notes in Computer Science*, Miami, USA, 2005, vol. 3536, pp. 444-454.

[4] W. M. P. van der Aalst e C. W. Gunther, "Finding Structure in Unstructured Processes: The Case for Process Mining", in *Proceedings of the Seventh International Conference on Application of Concurrency to System Design*, Washington, DC, USA, 2007, p. 3–12.

[5] G. M. Veiga e D. R. Ferreira, "Understanding Spaghetti Models with Sequence Clustering for ProM", in *Business Process Management Workshops*, Ulm, Germany, 2010, vol. 43, p. 92–103.

[6] J. D. Mott e G. Granata, "The value of teaching and learning technology: beyond ROI", *Educause Quarterly*, vol. 29, n. 2, 2006.

[7] A. J. M. M. Weijters, W. M. P. van der Aalst, e A. K. A. De Medeiros, "Process Mining with the HeuristicsMiner Algorithm", *BETA working paper, Eindhoven University of Technology*, 2006.

[8] W. M. P. van der Aalst, B. F. van Dongen, J. Herbst, L. Maruster, G. Schimm, e A. J. M. M. Weijters, "Workflow mining: a survey of issues and approaches", *Data & Knowledge Engineering*, vol. 47, p. 237–267, nov. 2003.

[9] J. I. B. da Cruz e D. Ruiz, "Uma experiência em mineração de processos de manutenção de software", in *Companion Proceedings of the XIV Brazilian Symposium on Multimedia and the Web*, New York, NY, USA, 2008, p. 247–253.

[10] B. F. Van Dongen, "A Meta Model for Process Mining Data", *IN PROCEEDINGS OF THE CAISE WORKSHOPS*, vol. 2, p. 309--320, 2005.

[11] C. W. Gunther, *XES Standard Definition*. Fluxicon Process Laboratories, 2009.

[12] W. M. P. van der Aalst, A. J. M. M. Weijters, e L. Maruster, "Workflow Mining: Discovering process models from event logs", *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, n. 9, pp. 1128-1142, 2004.

[13] W. M. P. van der Aalst e A. J. M. M. Weijters, "Process mining: a research agenda", *Computers in Industry*, vol. 53, n. 3, pp. 231-244, abr. 2004.

[14] A. K. A. De Medeiros et al., "Process mining based on clustering: a quest for precision", in *Proceedings of the 2007 international conference on Business process management*, Berlin, Heidelberg, 2007, p. 17–29.

[15] M. S. Song, C. W. Günther, e W. M. P. van der Aalst, "Trace clustering in process mining", in *Proceedings of the 4th workshop on business process intelligence.*, 2008, pp. 51-62.

[16] R. P. J. C. Bose e W. M. P. van der Aalst, "Context Aware Trace Clustering: Towards Improving Process Mining Results", in *Proceedings of SDM'2009*, 2009, pp. 401-412.

[17] UFRJ, *Plano Diretor de Tecnologia da Informação da Universidade Federal do Rio de Janeiro*. UFRJ, 2011.

[18] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, e I. H. Witten, "The WEKA data mining software: an update", *ACM SIGKDD Explorations Newsletter*, vol. 11, p. 10–18, nov. 2009.

[19] A. De Medeiros, A. Karla, e V. D. Aalst, "Semantic process mining tools: core building blocks", in 16th European Conference on Information Systems, Galway, Ireland, June 2008, p. 9-11.

Centrality Preservation in Anonymized Social Networks

Traian Marius Truta, Alina Campan, Ashley Gasmi, Nicholas Cooper, Andrew Elstun

***Abstract*—Social network sites continue to grow in number and size and accumulate information about their members. Among the data provided by members on the social sites they use, there are pieces of sensitive information about themselves. The identity and confidential information about social networks' individual nodes should be protected in all situations, including when the data is made public or released to third parties for analytical tasks. A possible solution to preserve the privacy of individuals is to anonymize the social network data and / or structure, i.e. to modify social network data and structure such that to make several individuals in the network alike, data and neighborhood-wise. Several anonymity definitions and methods to achieve them were introduced in the last few years. Of course, all anonymization approaches aim to preserve as much as possible the data and structural content of the initial social network; the less the inherent informational content is disturbed in the anonymization process, the more accurate are the results obtained by exploring the anonymized social network. Our work aims to study an existing anonymization approach with respect to how it preserves the structural content of the initial social network; specifically, we study how various graph metrics (centrality measures, radius, diameter etc.) change between the initial and the anonymized social network. This study is carried out for a number of synthetic social network datasets.**

I. Introduction and Motivation

The advent of social networks in the last few years created an enormous amount of social network data that could be potentially used for many purposes: for marketing, research, etc. This huge amount of data has created a revitalized interest in social network analysis and mining [1], [18], [20].

Some of the social networks gather individuals' confidential information and/or confidential relationships between individuals. For instance, PatientsLikeMe [24], Rareshare [26], and Daily Strength [12] are social networks in the healthcare field that create communities of patients for various diseases. As a result, privacy in social networks has become a serious concern and the research in this area has flourished in the past few years. Not only the privacy in social networks has become a topic discussed by scientists, but also the large public has shown a vivid interest for this matter. Concerns about privacy with respect to various social networks sites such as Facebook are reported in various media outlets and raise general public awareness about this problem [9]. Yet, the research in the social network privacy area is still very recent, and many problems remain to be solved.

Here are a few research directions in social network privacy.

Attacks in social networks are discussed in several research papers. In one of the early works in this field, Backstrom et al. described two types of attacks: active and passive [2]. An interesting de-anonymization experiment was performed by Narayanan and Shmatikov [23]. They showed that a third of the users who have accounts on both Twitter and Flickr can be re-identified in the anonymous Twitter graph with only a 12% error rate. An inference attack for released social networking data to infer undisclosed private information about individuals is presented in [21].

To defend against privacy attacks, several privacy models, which can be classified as graph modification and clustering-based approaches, were introduced. In the *graph modification approach* category, Liu and Terzi add edges to the original social network so that there are at least k nodes with the same degree [22]. Zhou and Pei introduce a stronger requirement: that each vertex must have k others with the same k-neighborhood characteristics [31]. In order to achieve this property, edge deletions and additions are performed. Other works in this direction include [7], [17], [29]. Unfortunately, it is not clear how well the graph structure is preserved during these graph modification processes, and this represents a major limitation of the graph modification techniques. In the *clustering-based approaches*, vertices and edges are grouped together in clusters and super-nodes and super-edges are created. One clustering-based approach is briefly presented in Section 2, and the full presentation can be found in [5]. Other works in this subarea include [3], [17], [30].

The research in social network privacy extends beyond the privacy attacks and defenses. Anonymization in bipartite graphs is studied in [8]. A relaxation of differential privacy [13] in the context of social networks is presented in [25]. A recent survey of this field can be found in [32].

In this paper the focus is how much data utility is preserved in the anonymized social networks. Specifically, we look at how social networks characteristics such as radius, diameter, and centrality measures [14], [15] are preserved through anonymization.

Other recent papers have also explored utility preservation in anonymized graphs. In our previous work, we introduced a measure of structural information loss that quantifies the probability of error when trying to reconstruct

T. M. Truta (phone: +1-859-572-7551; fax: +1-859-572-5398, e-mail: trutat1@nku.edu), A. Campan (e-mail: campana1@nku.edu), N. Cooper (e-mail: coopern1@mymail.nku.edu), and A. Elstun (e-mail: elstuna1 @mymail.nku.edu) are with the Department of Computer Science, Northern Kentucky University, Nunn Drive, Highland Heights, KY 41099, USA.

A. Gasmi is with the Department of Computer Science, ENSICAEN, 14000 Caen, France (e-mail: ashley.gasmi@ ecole.ensicaen.fr).

the structure of the initial social network from its masked version [5]. In the graph modification approaches, utility preservation was discussed in the context of preserving the same average degree distribution and the same shortest paths length [7], [29].

To our knowledge, no previous work has addressed how graph centrality measures are changed between original social networks and anonymized social networks, neither in graph modification approaches, nor in clustering-based approaches. Moreover, the only work that analyses some graph measures (degree, shortest-paths) was performed only for graph modification approaches such as *k*-isomorphism [7] and *k*-symmetry [29].

The remaining of this paper is structured as follows. Section 2 presents a clustering-based social network privacy model, in particular the concepts of edge generalization and *k*-anonymous masked social network. Section 3 briefly describes various graph measures that we comparatively analyze in our experiments, for original and anonymized social networks. Section 4 describes our experiments, and presents our preliminary findings. The paper ends with future work directions and conclusions.

II. Social Networks Anonymization Model

In this paper we use the social network anonymization model introduced in [5]. We briefly summarize it next.

We consider the social network modeled as a simple undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of edges. Each node represents an individual entity. Each edge represents a relationship between two entities. Usually, the set of nodes, $\mathcal{N}$, is described by a set of attributes that are classified into three categories: *identifier* (such as *Name* and *SSN*), *quasi-identifier* (such as *ZipCode* and *Sex*), and *sensitive* (such as *Primary Diagnosis* and *Income*). In this paper, we focus only on social network structure and therefore we will ignore the node attribute values during the anonymization process. For details about how the node attribute values are used during the anonymization process refer to [5].

We allow only binary relationships in our model. Moreover, we consider all relationships as being of the same type and, as a result, we represent them via unlabeled undirected edges. We also consider this type of relationship to be of the same nature as all the other "traditional" quasi-identifier attributes. We will refer to this type of relationship as the *quasi-identifier relationship*. In other words, the graph structure may be known to an intruder and used by matching it with known external structural information, therefore serving in attacks that might lead to identity and/or attribute disclosure [19].

Using the graph structure, an intruder is able to identify individuals due to the uniqueness of the neighborhoods of various individuals. As shown in [17], when the structure of a random graph is known, the probability that there are two nodes with identical 3-radius neighborhoods is less than 2^{-cn}, where n represents the number of nodes in the graph, and c is a constant value, $c > 0$; this means that the vast majority of the nodes can be uniquely identified based only on their 3-radius neighborhood structure.

To achieve anonymity for social networks, we have adapted the *k*-anonymity model [25], [28]. For social network data, the *k*-anonymity model has to impose both the quasi-identifier attribute and the quasi-identifier relationship homogeneity, for groups of at least *k* individuals. We have also reused the generalization technique for the generalization of node attributes' values [25] and we extended it for edges. To our knowledge, the only equivalent methods for the generalization of a quasi-identifier relationship that exist in the research literature appear in [17], [30] and consist of collapsing clusters of nodes together with their component nodes' structure. Edge additions or deletions are currently used, in all the other approaches, to ensure nodes' indistinguishability in terms of their surrounding neighborhood; additions and deletions perturb to a large extent the graph structure and therefore they are not faithful to the original data. We have employed a generalization method for the quasi-identifier relationship similar to the one exposed in [17], [30], but enriched with extra information, that will cause less damage to the graph structure, i.e. a smaller structural information loss.

Let n be the number of nodes from the set $\mathcal{N}$. Using a grouping strategy, one can partition the nodes from this set into v pairwise disjoint clusters: cl_1, cl_2, …, cl_v. For simplicity we assume that the nodes are not labeled (i.e., do not have attributes), and they can be distinguished only based on their relationships. Our goal is that any two nodes from any cluster to be indistinguishable based on their relationships. To achieve this goal, we introduced an edge generalization process, with two components: *edge intra-cluster* and *edge inter-cluster generalization*.

Edge intra-cluster generalization. Given a cluster cl, let $\mathcal{G}_{cl} = (cl, \mathcal{E}_{cl})$ be the subgraph of $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ induced by cl. In the masked data, the cluster cl will be generalized to (collapsed into) a node, and the structural information we attach to it is the pair of values $(|cl|, |\mathcal{E}_{cl}|)$, where $|cl|$ represents the cardinality of the set cl. This information permits assessing some structural features about this region of the network that will be helpful in some applications. From the privacy standpoint, an original node within such a cluster is indistinguishable from the other nodes in the cluster. At the same time, if more internal information was offered, such as the full nodes' connectivity inside a cluster, the possibility of disclosure would be too high, as discussed in [5].

Edge inter-cluster generalization. Given two clusters cl_1 and cl_2, let $\mathcal{E}_{cl1,cl2}$ be the set of edges having one end in each of the two clusters ($e \in \mathcal{E}_{cl1,cl2}$ if and only if $e \in \mathcal{E}$ and $e \in cl_1 \times cl_2$). In the masked data, this set of inter-cluster edges will be generalized to (collapsed into) a single edge and the structural information released for it is the value $|\mathcal{E}_{cl1,cl2}|$.

This information permits assessing some structural features about this region of the network that might be helpful in some applications and it reduces the disclosure risk.

Given a partition of nodes for a social network $\mathcal{G}$, we are able to create an anonymized graph by using edge intra-cluster generalization within each cluster and edge inter-cluster generalization between any two clusters.

Definition 1. (***anonymized social network***): Given an initial social network, modeled as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, and a partition $\mathcal{S} = \{cl_1, cl_2, \dots, cl_v\}$ of the nodes set $\mathcal{N}$, $\bigcup_{j=1}^{v} cl_j = \mathcal{N}$, $cl_i \cap cl_j = \emptyset$; $i, j = 1..v$, $i \neq j$; the corresponding ***anonymized social network*** $\mathcal{AG}$ is defined as $\mathcal{AG} = (\mathcal{AN}, \mathcal{AE})$, where:

- $\mathcal{AN} = \{Cl_1, Cl_2, \dots, Cl_v\}$, Cl_i is a node corresponding to the cluster $cl_j \in S$ and is described by the intra-cluster generalization pair $(|cl_j|, |E_{clj}|)$;
- $\mathcal{AE} \subseteq \mathcal{AN} \times \mathcal{AN}$; $(Cl_i, Cl_j) \in \mathcal{AE}$ iif $Cl_i, Cl_j \in \mathcal{AN}$ and $\exists X \in cl_j$, $Y \in cl_j$, such that $(X, Y) \in \mathcal{E}$. Each generalized edge $(Cl_i, Cl_j) \in \mathcal{AE}$ is labeled with the inter-cluster generalization value $|\mathcal{E}_{cli,clj}|$.

By construction, all nodes from a cluster *cl* collapsed into the generalized (masked) node *Cl* are indistinguishable from each other.

To have the *k*-anonymity property for a masked social network, we need to add one extra condition to Definition 1, namely that each cluster from the initial partition is of size at least *k*. The formal definition of a masked social network that is *k*-anonymous is presented below.

Definition 2. (***k-anonymous anonymized social network***): An anonymized social network $\mathcal{AG} = (\mathcal{AN}, \mathcal{AE})$, where $\mathcal{AN} = \{Cl_1, Cl_2, \dots, Cl_v\}$, and $Cl_j = [(|cl_j|, |E_{clj}|)]$, $j = 1, \dots, v$ is *k*-anonymous iff $|cl_j| \geq k$ for all $j = 1, \dots, v$.

Example 1: Suppose the social network $\mathcal{G}_{ex}$ depicted in Figure 1 is given. Two possible 3-anonymous social networks $\mathcal{AG}_{e1}$ and $\mathcal{AG}_{e2}$ are depicted in Figure 2.

The algorithm used in the anonymization process, called the *SaNGreeA* (Social Network Greedy Anonymization) algorithm, performs a greedy clustering processing to generate a *k*-anonymous masked social network, given an initial social network modeled as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$.

Specifically, *SaNGreeA* puts together in clusters nodes that are as similar as possible in terms of their neighborhood structure. To do so, it uses a measure that quantifies the extent to which the neighborhoods of two nodes are similar with each other, i.e. the nodes manifest the same connectivity properties, or are connected / disconnected among them and with others in the same way.

To assess the proximity of two nodes' neighborhoods, we proceed as follows. Given $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, assume that nodes in $\mathcal{N}$ have a particular order, $\mathcal{N} = \{X^1, X^2, \dots, X^r\}$. The neighborhood of each node X^i can be represented as an *n*-dimensional boolean vector $B_i = (b_1^i, b_2^i, , \dots, b_r^i)$, where the j^{th} component of this vector, b_j^i, is 1 if there is an edge $(X^i, X^j) \in \mathcal{E}$, and 0 otherwise, $\forall j = 1, r$; $j \neq i$. We consider the value b_i^i to be *undefined*, and therefore not equal to 0 or 1. We use a classical distance measure to assess the similarity of vectors of this type: the *symmetric binary distance* [15].

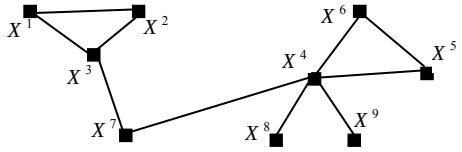
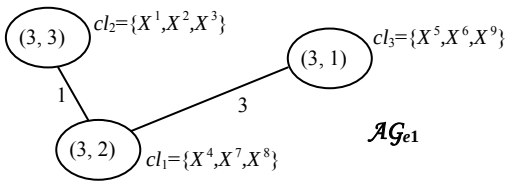


Fig. 1 The Social Network $\mathcal{G}_{ex}$



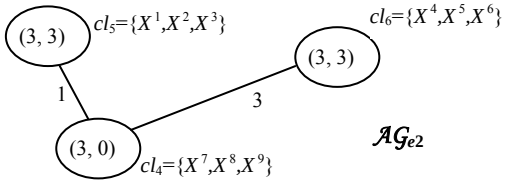


Fig. 2 The 3-anonymous social networks $\mathcal{AG}_{e1}$ and $\mathcal{AG}_{e2}$

Definition 3. (***distance between two nodes***): The ***distance between two nodes*** (X^i and X^j) described by their associated *n*-dimensional boolean vectors B_i and B_j is:

$$dist(X^i, X^j) = \frac{|\{\ell | \ell = 1..r \wedge \ell \neq i, j; b_\ell^i \neq b_\ell^j\}|}{r-2}.$$

We exclude from the two vectors' comparison their elements *i* and *j*, which are undefined for X^i and respectively for X^j. As a result, the total number of elements compared is reduced by 2.

In the cluster formation process, our greedy approach will select the closest remaining node to be added to the cluster currently being formed. To assess the structural distance between a node and a cluster we use the following measure.

Definition 4. (***distance between a node and a cluster***): The ***distance between a node*** *X* ***and a cluster*** *cl* is defined as the average distance between *X* and every node from *cl*:

$$dist(X, cl) = \frac{\sum_{Xj \in cl} dist(X, X^j)}{|cl|}.$$

Using the above introduced measures, we explain next how clustering is performed for a given initial social network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. The clusters are created one at a time. To form a new cluster, a node in $\mathcal{N}$ with the maximum degree and not yet allocated to any cluster is selected as a seed for the new cluster. Then the algorithm gathers nodes to this currently processed cluster until it reaches the desired cardinality *k*. At each step, the current cluster grows with one node. The selected node has to be unallocated yet to any cluster and it will minimize the *dist* measure (see Definition 4).

It is possible, when *n* is not a multiple of *k*, that the last constructed cluster will contain less than *k* nodes. In that case, this cluster needs to be dispersed between the previously constructed groups. Each of its nodes will be added to the cluster that is closest to that node w.r.t. our previously defined distance measure.

A version of the pseudocode of the *SaNGreeA* algorithm that includes node attributes and an additional optimization criterion can be found in [5].

III. Social Network Measures

A variety of social network analyses concentrate on determining how relationships are distributed in a social network between the entities participating in the network. These studies focus on assessing the individual nodes' influence or power in the network. Several graph connectivity and centrality metrics exist that quantify this notion of nodes' influence. Freeman suggested three measures for a node's centrality, as described next [14]. There also are other measures of graph connectivity (radius, diameter) and measures that describe the influence of a node on its network [11]. These social network measures try to capture complex relations between nodes in a network.

In our work, we plan to explore the effect that social network anonymization has on various measures. We investigate if a relationship between such connectivity and centrality measures exists– for the initial social network and for a corresponding anonymized social network. If such measures describing the influence of a node on its network transferred from an original node to its cluster / supernode, then network analysis in various fields (such as viral marketing, communication networks) could be successfully conducted on anonymized networks, while preserving the privacy of individual network nodes. Next, we briefly describe the social network measures analyzed in our experiments.

Let $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ be an undirected graph (that represents a social network), where $\mathcal{N}$ (the cardinality of $\mathcal{N}$, $|\mathcal{N}| = n$) is the set of nodes and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of edges (the cardinality of $\mathcal{E}$, $|\mathcal{E}| = m$).

The **eccentricity of the node *v*** is the maximum distance from *v* to any node. That is, $\varepsilon(v) = \max\{d(v, w) \mid w \in \mathcal{N}\}$.

The **radius of G** is the minimum eccentricity among the nodes of $\mathcal{G}$. Therefore, $radius(G) = \min\{\varepsilon(v) \mid v \in \mathcal{N}\}$.

The **diameter of G** is the maximum eccentricity among the nodes of $\mathcal{G}$. In other words, $diameter(G) = \max\{\varepsilon(v) \mid v \in \mathcal{N}\}$.

The **degree centrality of a node *v*** is the number of edges adjacent to the node (degree) normalized to the interval [0, 1]. Thus, $C_D(v) = \frac{\deg(v)}{n-1}$. The larger the degree centrality of a node *v*, the stronger its communication potential; the lower the degree centrality, the more peripheral the node is perceived.

The **degree centrality of $\mathcal{G}$** is defined as follows: $C_D(G) = \frac{\sum_{i=1}^{n}[C_D(v*) - C_D(v_i)]}{n-2} = \frac{\sum_{i=1}^{n}[deg(v*) - deg(v_i)]}{(n-1)\cdot(n-2)}$, where *v** is the node that has the maximum degree centrality from all nodes from $\mathcal{G}$.

The **betweenness centrality of a node *v*** is the sum of the number of shortest paths between any pair of vertices (except the considered node) going through the node, divided by the number of shortest paths between any pair of vertices. This sum is normalized to [0, 1]. In other words, $C_B(v) = \frac{2\cdot\sum_{s\neq v\neq t\in N}\frac{\sigma_{st}(v)}{\sigma_{st}}}{(n-1)\cdot(n-2)}$, where σ_{st} is the number of shortest paths from *s* to *t*, and $\sigma_{st}(v)$ is the number of shortest paths from *s* to *t* that pass through a vertex *v*. This measure expresses a node's potential for control of communication.

The **betweenness centrality of $\mathcal{G}$** is defined as follows: $C_B(G) = \frac{\sum_{i=1}^{n}[C_B(v*) - C_B(v_i)]}{n-1}$, where *v** is the node that has the maximum betweenness centrality from all nodes from $\mathcal{G}$.

The **closeness centrality of a node *v*** is defined as the inverse of the average of shortest paths length between the node *v* and all other nodes from $\mathcal{G}$. This sum is normalized to [0, 1]. In other words, $C_C(v) = \frac{n-1}{\sum_{i=1}^{n} d(v_i,v)}$, where $d(v,w)$ is the length of the shortest path from *v* to *w*. This measure gives the potential for independent communication of a node, or in other words, how much the node can avoid the potential control of others.

The **closeness centrality of $\mathcal{G}$** is defined as follows: $C_C(G) = \frac{\sum_{i=1}^{n}[C_C(v*) - C_C(v_i)]}{(n-1)\cdot(n-2)/(2n-3)}$, where *v** is the node that has the maximum betweenness centrality from all nodes from $\mathcal{G}$.

For all three centrality measures of $\mathcal{G}$, the denominators are computed based on the maximum possible sum of differences in node centrality for a graph of *n* nodes, $max\sum_{i=1}^{n}[C_X(v^*) - C_X(v_i)]$, where *X* represents degree (*D*), betweenness (*B*), and closeness (*C*). More details about these measures can be found in [14].

IV. Experiments Design and Results

We designed a series of experiments that allowed us to explore if the proposed graph anonymization algorithm (*SaNGreeA*) preserves some of the graph properties, in particular centrality properties, of social networks. The general framework of our experiments is presented in Figure 3.

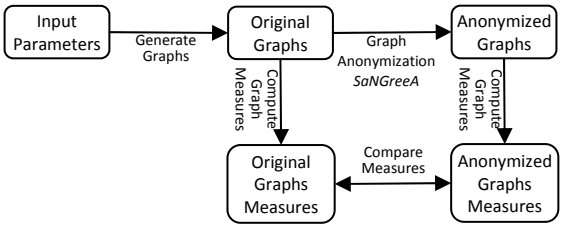


Fig. 3 General framework of the experiments.

We divided our experiment into several phases. In our first phase, labeled Graph Generation in Figure 3, we implemented an R-MAT graph generator [6] and a Random graph generator.

The R-MAT graph generator takes the number of nodes (n), the average node degree (*avg_deg*), and four probabilities as input parameters. The algorithm computes how many edges such a graph has, and for each edge, its location is determined based on the recursive algorithm that divides the adjacency matrix into 4 equal-sized partitions and the location of the edge is probabilistically selected in one of the 4 locations, based on the four probability parameters. Once a partition is found, it is again divided into four sub-partitions until there will be only one location left in the partition. If an edge was already placed on that location, we will repeat this procedure from the beginning (multiple edges between the same pair of nodes is not allowed in our graph model). For all our tests we used the following values for the four probabilities: 0.45, 0.15, 0.15, and 0.25. This choice seems to model better many real-world graphs that follow power-law degree distributions [6]. More details about this algorithm can be found in [6].

The Random graph generator creates a random undirected graph using the Erdos-Renyi model [4]. In this model, each edge is included in the graph with probability p, with the presence or absence of any two distinct edges in the graph being independent. For our generator we use two input parameters: number of nodes (n) and average node degree (*avg_deg*), and we estimate the probability as the *avg_deg* / n. Using this approach, the generated graph will have a slight different average node degree than the input parameter.

We used both graph generator models with various parameter values to create a large number of synthetic graphs on which we performed our experiments. For the number of nodes (n) we used the following values: 10, 25, 50, 75, 100, 250, and 500. For the average node degree (*avg_deg*) we used 2, 3, 4, 5, 8, 10, 25, 50, 75, 100, and 250. Of course, the average node degree is strictly less than the number of nodes (we are not interested in complete graphs in our experiments). Since most of the centrality measures are defined only for connected graphs, for any given combination of input parameters we wanted to generate a connected graph. To achieve this, we generated up to 10,000 graphs and we stopped our graph generator at the first connected graph. In some cases (such as number of nodes = 500, and average node degree = 2) we were not able to generate a connected graph. The list of all generated graphs with the corresponding parameter values is provided in Table I. The total number of generated graphs is 78.

TABLE I
The List of All Generated Graphs

Graph Generator Model	(*n*, *avg_deg*)
R-MAT and RANDOM	(10, 2), (10, 3), (10, 4), (10, 5) (25, 2), (25, 3), (25, 4), (25, 5), (25, 8), (25, 10) (50, 3), (50, 4), (50, 5), (50, 8), (50, 10), (50, 25) (75, 4), (75, 5), (75, 8), (75, 10), (75, 25) (100, 4), (100, 5), (100, 8), (100, 10), (100, 25), (100, 50) (250, 5), (250, 8), (250, 10), (250, 25), (250, 50), (250, 100) (500, 8), (500, 10), (500, 25), (500, 50), (500, 100), (500, 250)

In the second phase of this experiment we generated anonymized graphs using the *SaNGreeA* algorithm presented in Section 2. For each generated graph we used various values for k (k as in k-anonymous social network). For n = 10 we used k as 2 and 5; for n = 25, we used k = 2, 5 and 10, and for all other values of n, we used k = 2, 5, 10, 15, and 20. In total 342 anonymized graphs were generated.

In the third phase, we implemented all graph measures described in Section 3. For all 420 graphs (78 generated graphs and 342 anonymized graphs), we computed these graph measures. For an anonymized graph we did not use the weight of an edge between super-nodes, and we considered these graphs as unweighted graphs.

In the last phase of our experiment we compared the original graph measures with the corresponding anonymized graph measures. We are still in the process of analyzing all these results, some preliminary findings are presented next.

Figure 4 shows a sample of the results we obtained for radius and diameter. As expected, both these measures decrease as k increases.

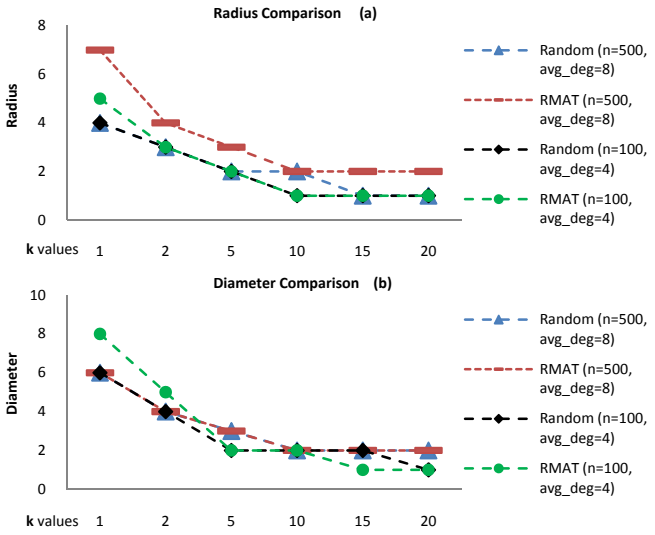


Fig. 4 Radius and diameter values for some of the experiments.

Figure 5 shows partial results with respect to centrality measures. For all measures we report the centrality measure for the anonymized graph divided to the centrality measure for the original graph. The reference value for the original graph is 1 for all three measures. We illustrate these results for four distinct original graphs (2 Random graphs, one with 500 nodes and average node degree 8, and the second one with 100 nodes and average node degree 4, and 2 RMAT graphs with the same number of nodes and average node degrees). For each original graph we created 5 *k*-anonymous graphs, k ∈ {2, 5, 10, 15, 20}.

The degree centrality, illustrated in Figure 5 (a), increases as *k* increases to 5 (for the smaller graphs) or 10 (for the larger graphs) and then decreases. This is due to how *SaNGreeA* algorithm creates clusters. For smaller *k* values, it creates supernodes from nodes highly connected between them and loosely connected to other nodes, which results in lower connectivity between supernodes; this means that the anonymized graph becomes sparser than the original graph. However, when *k* increases, there are not enough similarly connected nodes that could become alone a supernode; as a result, nodes with different connectivity properties are merged into supernodes and the anonymized graph gets closer to the complete graph. We notice the initial increase for degree centrality is steeper for Random graphs than RMAT. This is expected since an original Random graph has a uniform distribution of node degrees.

The betweenness centrality showed in Figure 5 (b) usually decreases for the anonymized graphs. Again, this is because the anonymized graph gets closer to the complete graph as *k* increases, and therefore there are many short paths of length 1. The small increase between $k = 2$ and $k = 5$ is, at the first view, unexpected. This is due to the fact that for small *k* values, the anonymized graph still has variety in supernodes' connectivity, and some of the supernodes gain more control over the shortest paths that exist in the anonymized graph; these nodes have a high betweenness centrality.

The closeness centrality decreases for anonymized graphs when the value of *k* increases as shown in Figure 5 (c). This is again due to the anonymized graph getting closer to the complete graph.

Overall our experiments show a weak correlation between the anonymization level (the *k* value) of a graph and the centrality measures: same changes are observed for graphs of different sizes and with different network properties.

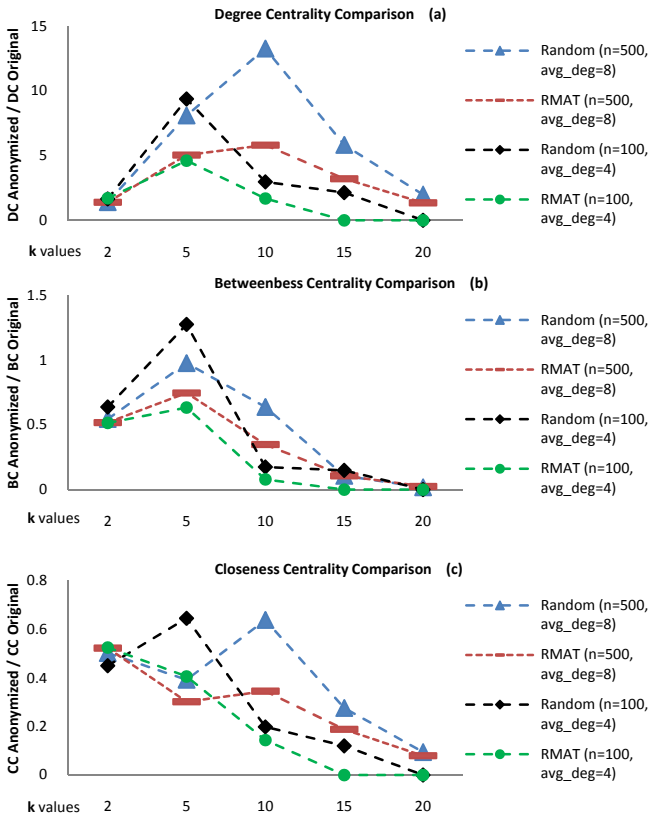


Fig. 5 Centrality measures values for some of the experiments.

V. CONCLUSIONS

In this paper we studied a clustering-based anonymization approach with respect to how it preserves the structural content of the initial social network; specifically, we looked at how various graph metrics (centrality measures, radius, diameter etc.) change between the initial and the anonymized social network. Our results showed that there are similarities in how various centrality measures are modified from an original graph to its anonymized versions even if we change the graph size and network properties. We plan to study how other anonymization models behave with respect to centrality measures.

REFERENCES

[1] E.Adar and C.Re, "Managing Uncertainty in Social Networks," *Data Engineering Bulletin*, vol. 30, no. 2, pp. 23-31, 2007.

[2] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore Art Thou R3579X? Anonymized Social Networks, Hidden Patterns, and Structural Steganography," in *Proc. WWW'07*, pp. 181-190, 2007.

[3] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava, "Class-based Graph Anonymization for Social Network Data," in *Proc. VLDB'09*, pp. 766-777, 2009.

[4] B. Bollobás, *Random Graphs*, 2nd ed., Cambridge University Press, 2001.

[5] A. Campan and T. M. Truta, "Data and Structural K-Anonymity in Social Networks," *Lecture Notes in Computer Science*, Berlin, Germany: Springer, vol. 5456, pp. 33-54, 2009.

[6] D. Chakrabarti, Y. Zhan and C. Faloutsos, "R-MAT: A Recursive Model for Graph Mining," in *Proc. SDM'04*, pp. 442-446, 2004.

[7] J. Cheng, A. W. C. Fu, and J. Liu, "K-Isomorphism: Privacy Preserving Network Publication against Structural Attacks," in *Proc. SIGMOD'10*, pp. 459-470, 2010.

[8] G. Cormode, D. Srivastava, T. Yu, and Q. Zhang, "Anonymizing Bipartite Graph Data using Safe Groupings," in *Proc. VLDB'08*, pp. 833-844, 2008.

[9] D. Costa, "Facebook: Privacy Enemy Number One," *PCMag*, Available: http://www.pcmag.com/article2/0,2817,2362967,00.asp, 2010.

[10] L. Costa, F. Rodrigues, G. Travieso, and P. Boas, "Characterization of Complex Networks: A Survey of Measurements," *Advances in Physics*, vol. 56, no. 1, pp. 167-242, 2007.

[11] P. Domingos, and M. Richardson, "Mining the network value of customers," in *Proc. KDD'01,* pp. 57-66, 2001.

[12] DS, "Daily Strength," Available: http://www.dailystrength.org, 2006.

[13] C. Dwork, "Differential Privacy: A Survey of Results," *Theory and Applications of Models of Computation*, pp. 1-19, 2008.

[14] L. C. Freeman, "Centrality in Social Networks: Conceptual Clarification," *Social Networks*, vol. 1, no. 3, pp. 215-239, 1979.

[15] J. Han and M. Kamber, *Data Mining, Second Edition: Concepts and Techniques*, Morgan Kaufmann, 2006.

[16] F. Harary, *Graph Theory*, Addison-Wesley, 1994.

[17] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weiss, "Resisting Structural Re-identification in Anonymized Social Networks," in *Proc. VLDB'08*, pp. 102-114, 2008.

[18] J. Kleinberg, "Challenges in Mining Social Network," in *Proc. KDD'07*, pp. 4-5, 2007.

[19] D. Lambert, "Measures of Disclosure Risk and Harm" Journal of Official Statistics, vol. 9, pp. 313-331, 1993.

[20] S. Levine and R. Kurzban, "Explaining Clustering in Social Networks: Towards an Evolutionary Theory of Cascading Benefits," *Managerial and Decision Economics*, vol. 27, pp. 173-187, 2007.

[21] J. Lindamood, R. Heatherly, M. Karantacioglu, and B. Thuraisingham, "Inferring Private Information Using Social Network Data," in *Proc. WWW'09*, pp. 1145-1146, 2009.

[22] K. Liu and E. Terzi, "Towards Identity Anonymization on Graphs," in *Proc. SIGMOD'08*, pp. 93-116, 2008.

[23] A. Narayanan and V. Shmatikov, "De-anonymizing Social Networks," Proc. IEEE Security and Privacy, pp. 173-187, 2009.

[24] PLM "Patients Like Me," Available: http://www.patientslikeme.com.

[25] V. Rastogi, M. Hay, G. Miklau, and D. Suciu, "Relationship Privacy: Output Perturbation for Queries with Joins," in *Proc. PODS'09*, pp. 107-116, 2009.

[26] RS, "Rareshare," Available: http://www.rareshare.org, 2008.

[27] P. Samarati, "Protecting Respondents Identities in Microdata Release," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010-1027, 2001.

[28] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy," *International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems*, vol. 10, no. 5, pp. 557 – 570, 2002.

[29] W. Wu, Y. Xiao, W. Wang, Z. He, and Z. Wang, "K-Symmetry Model for Identity Anonymization in Social Networks," in *Proc. EDBT'10*, pp. 111-122, 2010.

[30] E. Zheleva and L. Getoor, "Preserving the Privacy of Sensitive Relationships in Graph Data," in *Proc. Privacy, Security, and Trust in KDD Workshop*, pp. 153-171, 2007.

[31] B. Zhou and J. Pei, "Preserving Privacy in Social Networks against Neighborhood Attacks," in *Proc ICDE'08*, pp. 506-515, 2008.

[32] B. Zhou, J. Pei, and W. S. Luk, "A Brief Survey on Anonymization Techniques for Privacy Preserving Publishing of Social Network Data," *SIGKDD Explorations*, vol. 10, no. 2, pp. 12-22, 2008.

Design of Customer Behavior Analysis Model in Automobile Marketing

Mao Yuanyuan, Huang Lan, Wang Guishen, Zou Shuxue*
College of Computer Science and Technology, Jilin University, Changchun, China, 130012
(E-mail: zousx@jlu.edu.cn)

Abstract—**To adapt to the development of automobile industry in China, a customer behavior analysis system in automobile marketing should be built to enhance the competitiveness of automobile enterprises. In this paper, the technical architecture of customer behavior analysis model for marketing themes, which is based on the data of automobile enterprise's customers and management information systems, is designed by integrating the basic data through Data Mining technology. In this model, four function modules, which are customer value classification, identification of potential customer, customer loss analysis and customer satisfaction analysis, will be realized.**

Keywords-automobile marketing; customer behavior analysis; Data Mining(DM); Customer Relationship Management(CRM)

I. INTRODUCTION

The environment of Chinese automobile market has marked changes since China joined the WTO: the market has been opened up gradually and many multinational automobile groups have entered new market pattern appears--international automobile enterprises become domestic while domestic automobile enterprises become international. With the intense competition of automobile industry, the differences automobiles' brands and performances, as well as different characteristics of their customers such as region, culture, income and social status, have cultivated the huge diversity of needs in the automobile market. In order to gain more profit and keep more customers, the policymakers of automobile enterprises must make accurate forecasting and decisions. Therefore, the managers should make reasonable analysis and accurate prediction, classify their target customers, make active marketing activities and improve service for all customer segments, to achieve the purpose that they can keep the business volume continuously growing and expand the market share [1].

II. TECHNOLOGIES AND THEORIES

A. Applications of Customer Relationship Management in the Automobile Industry

Customer Relationship Management is a kind of business strategy, which aims at making the biggest long-term value of the target customers by selecting valuable customers, expanding new customers and eliminating worthless customers. It is a process of active marketing, market expansion and high-quality products and services provision, which is based on the customer-centric management theories and culture[2]. With right staff, strategy and culture, enterprises could use the theory to form the customer-centric business model which is profitable in a large scale.

The applications of Customer Relationship Management in the automobile industry have four levels: in the first level, customer services are based on the call center, mostly passive services and active concern, whose value is reflected in cost savings and improving the low level customer satisfaction. The second level is the management of customer information and process. In this level, the vehicle manufacturers use the ERP system and DMS system to manage some customers' information and trading process. The third level is to identify the customers' classification, value, satisfaction and loyalty. Based on the perfection and accumulation of the second level, the real and effective customer related data will be analyzed and the model will be built. The fourth level is the synergy of enterprise's value chain. The related enterprises in the value chain of the entire automobile industry would establish the synergy system, and share as well as manage resources effectively. However, no enterprise has achieved this level now because of the high demand. With the customers becoming the core competitiveness of automobile enterprises, CRM will be the key application of the world's automobile industry informationization in next period[3].

B. Applications of Data Mining in the Automobile Customer Management

The applications of Data Mining technology in Customer Relationship Management provide the strategic and decision-making support for the development of enterprises, and also help automobile enterprises find the business trend and predict the unknown results, as well as help the automobile sales enterprises analyze and complete the key factors needed by their tasks. In this way, automobile companies can increase the revenue, reduce the cost and achieve a good competitive position [4].

Since the data of customer information, sales, vehicles and complaints exist respectively in different systems which are independent and can not be utilized by combination, these databases must be integrated into a unified data warehouse to achieve the effective use. According to the data of different data sources, a lot of extraction, conversion and transmission should be done. The integrated data model includes some tables, such as dimension table of customers, dimension table of goods, dimension table of time and fact table of sales. The first three dimension tables are associated with the fact table of sales simultaneously, which includes information such as the numbers of goods, time and customers, sales quantity, sales, costs and sales profits, etc. These information are undoubtedly

the most advantageous data needed in the analysis of customer's requirement, targeting customers and making relevant marketing strategies[5].

III. DESIGNS OF TECHNICAL FRAMEWORK AND SYSTEM STRUCTURE

The UML modeling language will be used to build a model which is focused on customer behavior analysis system in this paper. In each stage of the system UML will be used to establish a complete model, and the modeling features in each stage will be summarized. During the early development of the system, cases which are closely related to the framework will be identified, then by analyzing and designing these cases a framework will be built. At the same time the software system framework will be defined and described from different angle through case models, analysis models and design models which are associated with the framework.

A. Design of Technical Framework

The software system of this system contains two parts: the client software and server software. This system expands the traditional B-S structure and also realizes a system structure model of new intelligent client. The client realizes the independent client application based on the RIA technology, and the server provides remote service call. The client uses the framework of Blaze DS to call the service of server remotely in the way of RPC. It is pure data which is transmitted between the server and the client but not pages like the traditional B - S patterns, thus the system can improve the communication efficiency greatly. Simultaneously, it will realize a lot of business logic in the client, also it can reduce the burden of server and improve the operation efficiency of the system. Technical framework of this system is shown in Fig. 1.

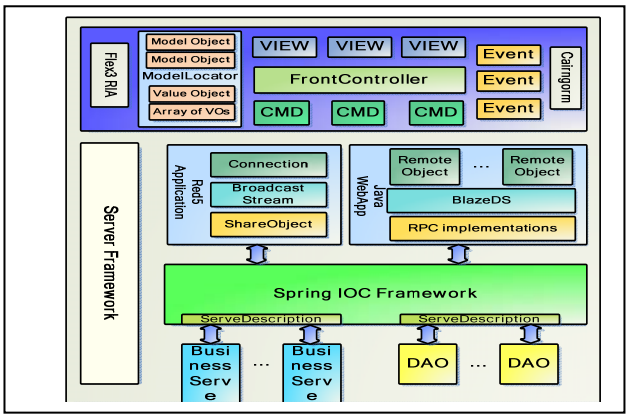


Figure 1. Technical Framework.

Taking the Spring IOC framework as the core, the service system realizes the dynamic equipment of each service object by using implantation technology, thus the service system will have a high degree of flexibility and expansibility. There are some concrete service objects of the service system which can be classified as follows: service object of basic business, service object of business, data access object (DAO) and video processing object. These service objects can call each other through the implantation technology. At the same time, the server framework places these objects which need to provide external services into the system framework through the Spring, and then it provides RMTP streaming service and AMF remote call service through a common external access interface.

The client that takes controller as the core and based on the event-driven way uses the technical development of Flex3 RIA and the design pattern of MVC (Cairngorm micro architecture) to realize loose coupling among view, model and business processing, the high degree of expansibility of front interface.

B. Design of System Structure

The application layer shows the services in graphs, and the function module layer is used to realize the four modules of customer behavior analysis. In the module of customer value classification, the classic K–means algorithm is realized. In the module of identification of potential customer, the Apriori association rules algorithm is realized. In the module of customer loss analysis, the decision tree ID3 algorithm is realized, and finally in the module of customer satisfaction analysis, the algorithm which is fuzzy comprehensive evaluation method is realized. These algorithms provide the reliable basis and guarantee for users to do customer behavior analysis. The data warehouse layer is used to connect underlying database with external data, which will do the corresponding encapsulation for database connection operation and data processing, and then submit them to the function module layer in unified form to realize the algorithms[6]. In the concrete design, three layers of system structure of this system are as shown in Fig. 2.

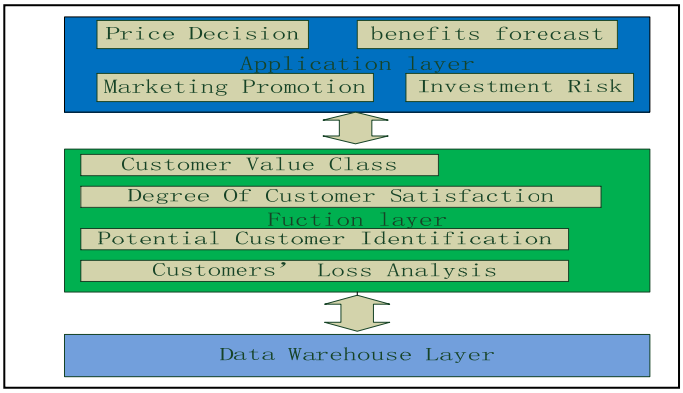


Figure 2. System Layered Architecture.

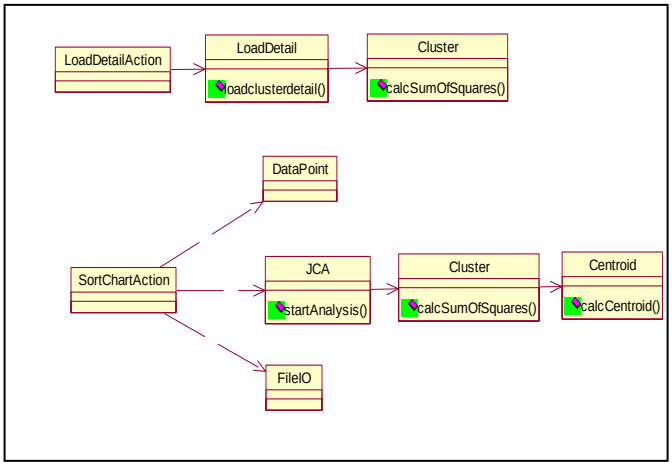
IV. DESIGNS OF FUNCTION MODULES

A. The Module of Customer Value Classification

Customer value classification requires us to assess the customers according to the domain knowledge in every dimension of customer data, and eventually comprehensively get the scores of these customers. Specifically, a specific type of automobile will be used as an example: first, set up the floor and ceiling of customer value, then specify the number of groups, thus we will get the result of customer value classification[7].

When classifying with the Data Mining technology, two conditions should be kept: integrity and exclusiveness. Integrity means each consumer in database must belong to a classification group. Exclusiveness means each consumer in database must not belong to different classification groups. Clustering algorithm is used in this paper.

K-means algorithm is a clustering method, whose core concept is to divide the data objects into different clusters through the iteration in order to minimize the objective function, and then it will keep the generated clusters as compact and separate as possible.



1) Design of Function Module

Figure 3. The Main Implementation Of Customer Value Classification.

The concrete realization of customer value is shown in Fig. 3.This module provides two interfaces: LoadDetailAction interface, which is used to load information of displaying clusters, and SortChartAction interface, which is used to display and process the classification results.

The LoadDetail class belongs to the logic layer, which includes a loadclusterdetail method that is used for loading information, and Cluster is a concrete realization of cluster classes. Thus the LoadDetailAction interface will realize the function for displaying results by calling the loadclusterdetail of the LoadDetail class.

The JCA class is also an algorithm belonging to the logic layer, in which the startAnalysis realizes the realization for the core parts of K-means algorithm. Yet the SortChartAction can realize the K-means algorithm and the functional requirements for customer value classification by calling this method in JCA.

2) Evaluation

As is known to all, the k-means algorithm's clustering results don't work well with noise data. However, we have took data preprocessing. So the results can fit to the reality better.

B. Module of Identification of Potential Customer

Identification of potential customer is a process of looking for potential customers from the specified crowd. After analyzing and summarizing the data correspondingly, we can get the generalities of the data which characterizes the potential customers. If some data element has these generalities when testing data concentration, this data element may characterize the potential customers. Thus, the substance of identificating potential customer is the association analysis on the numeralization information.

1) Apriori Association Rules Algorithm

Association Rules algorithm is one of the most active research methods in Data Mining. It was promoted for the analysis of the shopping basket problem, and its purpose was to find out the association rules between different goods in transaction database. These rules could portray patterns of customers' buying behavior, which can be used to guide the merchants to arrange purchasing, stock and shelves design properly. The process of concrete realization as follows:

a) Initialization: Scan databases in a single trip and calculate the support of each itemset, getting the set which is made of frequent itemsets as L;

b) Connection: Generate a set of potential frequent itemsets as Ck，Ck is obtained by calculating JOIN;

c) Pruning: If some (k - 1) subset of potential k itemsets does not belong to L k -1, remove the potential frequent itemset;

d) Scan databases in a single trip, and calculate the support of each itemset in Ck;

e) Eliminate each itemset which does not satisfy the minimum support in Ck, and form the set L which is made of frequent k itemsets;

f) If it can generate a new set of frequent itemsets, jump to b).

2) Design of Function Module

The concrete realization of identification of potential customer module is shown in Fig. 4. This module provides the PotenialListAction interface, which is used to calculate the frequent itemsets. The class of Apriori_generateRule is the concrete realization of Association Rules, also it is the realization of function apriori_gen().PotenialListAction can realize the Apriori algorithm through the calls of these functionality classes.

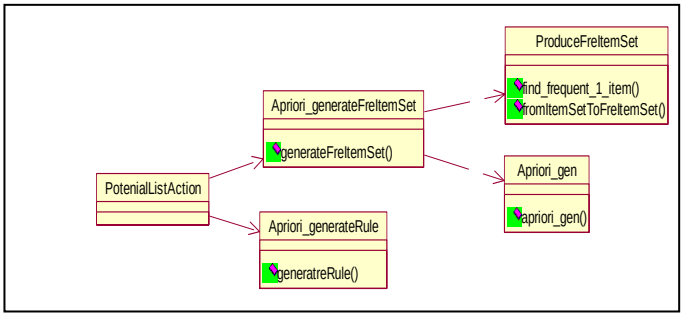


Figure 4. The Main Implementation Of Customer Loss Analysis.

3) Evaluation

We select the field in customer information diagram shown in TABLE I. to carry on the discovery of potential customer. And the results are shown in Fig. 5. In Fig.5 , we define the distance between two points which represents the relationship degree and meets the threshold value. Also in Fig.5, we can conclude that the results can reflect the reality in some extent.

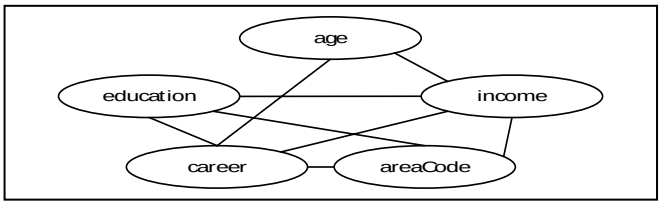


Figure 5. Potential Customer Discovery Results.

TABLE I Selected Customer Field

Customer Information	
Selected Field	***Content***
Id	Customer Serial Number
Age	Customer Age
Career	Customer Career
Income	Customer Income
Education	Customer Education
AreaCode	Area Code

C. Module of Customer Loss Analysis

Customer loss analysis means that we can get a customer loss tree through analyzing the existing data of customer loss, and then we can find out the key factors which cause customer loss from the data. The research of customer loss problem aims at solving the problems by Data Mining algorithm. The Decision Tree algorithm does not need the domain knowledge and it has the characteristics of learning and faster classification, also the expression of classification rules has good interpretability, so that the classification rules are chosen as the scheme of customer loss analysis. The mapping of data sample from the condition attributes to class identification attributes could be realized by building Decision Tree.

1) Decision Tree ID3 Algorithm

Decision Tree is a classification method based on greedy algorithm, which is built top to down and is a tree structure which is similar to flowchart. Each internal node of it represents a test on an attribute, each branch represents a test output, yet each leaf node represents a class or class distribution. ID3 algorithm is the core algorithm of Decision Tree, which is a classification algorithm of Decision Tree based on information entropy, and it judges the category of instances by the values of attribute set. ID3 algorithm as follows:

a) Create node N by using the sample data;

b) If all of the sample data is included in a class, then return the node N and take it as a leaf node, marking this class as C;

c) If the candidate attribute set is empty, then return the node N and take it as a leaf node, marking this class as the most common class C in S. If the candidate attribute set is not empty, calculate the information gain of node N in this attribute for each attribute, find the maximum attribute of information gain marked as attribute, and mark node as text attribute. Then based on the classification of sample data in attribute called attribute, the sample data set is divided into Si, and each branch grows from node N at the same time;

d) Traverse Si. If Si is empty, return to the leaf node and mark Si as the most common class in S. Or else, recursively call this process.

2) Design of Function Module

The concrete realization of customer loss analysis module is shown in Fig. 5.This module provides the LostTreeAction interface. This interface is used to analyze customer loss, and the class DTre is the concrete realization of Decision Tree. The function of creating and saving is realized by calling the related auxiliary class. LostTreeAction completes the process of creating, training and deciding the Decision Tree by calling three methods of DTree, Create and SaveTree.

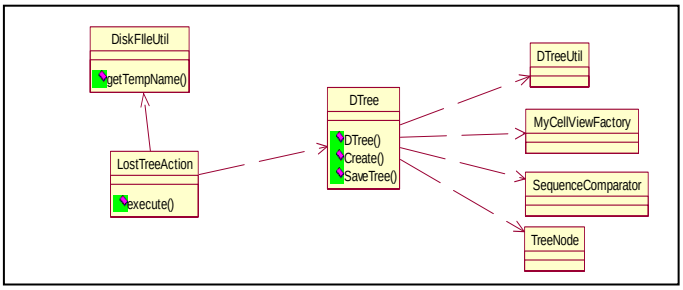


Figure 6. The Main Implementation Of Customer Loss Analysis.

3) Evaluation

Decision Tree ID3 algorithm analyses the lost customers' basic data and the consuming behavior; tries to find out the reason that causes customers losing; and describes the characteristic of the lost customers. After these steps, we can build predict model based on existing customers' dataset and take adaptive measures to avoid more customer loss. Customer loss analysis means that we can get a customer loss tree through

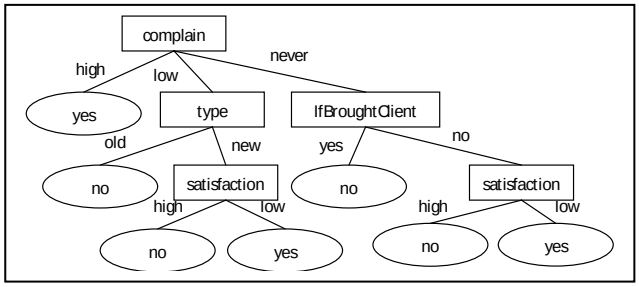


Figure 7. Customer Loss Decision Tree

D. Module of Customer Satisfaction Analysis

Customer satisfaction analysis means that we can get the overall satisfaction and each index satisfaction of customers over a automobile brand by setting the metrics of four indexes: price, quality, service and brand. The views, preferences and attitude of customers over the things have certain fuzziness. When evaluating the evaluation objects, they need to quantify them. We often take the normalized fuzzy comprehensive evaluation set as the corresponding fuzzy set of customer satisfaction.

1) Fuzzy Comprehensive Evaluation Method

Systematically evaluating with the fuzzy evaluation method, the main steps as follows:

a) Determine the system evaluation project set F， $F = (f_1 , f_2 , \dots , f_n)$;

b) Determine the evaluation scale set of each evaluation project E， $E = (e_1, e_2, \ldots, e_m)$;

c) According to expert experience or using the AHP(Analytic Hierarchy Process), determine the weight of each evaluation project W， $W = (w_1, w_2, \ldots w_n)$;

d) Evaluate the evaluation projects with the determined evaluation scale, and this evaluation is a fuzzy mapping. The evaluation result is indicated by the membership matrix R_k, in which the element r^k_{ij} indicates the percentage of the number of customers that make the j-th evaluation scale ej for the i-th evaluation project f i in the A k plan in the total number of customers in evaluation

e) Calculate the comprehensive evaluation vector S_k in the plan A_k, $S_k = W R_k$;

f) According to the membership matrix R_k,

$$R_k = \begin{bmatrix} r^k_{11} & r^k_{12} & \cdots & r^k_{1j} & \cdots & r^k_{1m} \\ r^k_{21} & r^k_{22} & \cdots & r^k_{2j} & \cdots & r^k_{2m} \\ r^k_{i1} & r^k_{i2} & \cdots & r^k_{ij} & \cdots & r^k_{im} \\ r^k_{n1} & r^k_{n2} & \cdots & r^k_{nj} & \cdots & r^k_{nm} \end{bmatrix}$$

g) calculate the percentage of each evaluation project in different satisfaction levels.

2) Design of Function Module

The concrete realization of customer satisfaction analysis module is shown in Fig. 6. This module realizes SatisfyChartAction interface, which is used to analyze customer satisfaction. The class of WeightMatrix saves the membership matrix in algorithm, SatisfyChartAction completes this function by calling the method getcountbycolumeandtype() in WeightMatrix.

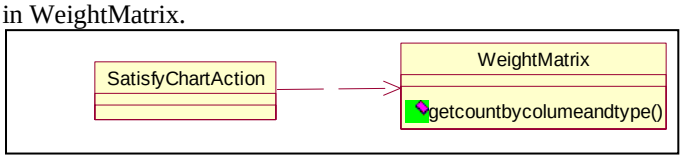


Figure 8. The Main Implementation Of Customer Satisfaction Analysis.

3) Evaluation

Fuzzy Comprehensive Evaluation algorithm avoids direct measurement of expectation before algorithm carrying on and evaluation after algorithm carrying on. In the meanwhile, ID3 algorithm take the influence of main factors into consideration. It has stronger practicability. It can help enterprises know the delivered value of goods or service they supply to customer and the gap between competitor and itself. And after these steps, they can analyze the main factors of customers' satisfaction and dissatisfaction. So they can take effective measures to improve the satisfaction of customer.

V. CONCLUSION

This paper discusses some application fields in customer behavior analysis model of automobile marketing： use Data Mining technology in analyzing and processing amounts of customer consuming information which is collected, processed and stored in customers database, to determine interests, consumption habits, consumption tendency and consumption needs of the specific consumer groups or individuals; provide both qualitative and quantitative analysis of customers' characteristics and explore the operation regularities of automobile enterprises and the relative market; feed back results to the managers and the entire internal enterprise, provide decision support for the work of Customer Relationship Management in enterprise; build and maintain the customer relationships, forecast the customers' tendency accurately and in time, as well as manage and develop the customer relationship effectively, which will provide enterprises unique competitive advantages. These applications are becoming the new development tendency in the informatization of Chinese automobile industry.

In the future, we will go on improving algorithms' capability of practical use. For example, we can combine several algorithms to improve their efficiency.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant No. 60873146, 60973092, 60903097, 11001106.

REFERENCES

[1] HUANG Lan，ZHOU Chun-guang，ZHOU Yu-qin，WANG Zhe. Research on Data Mining Algorithms for Automotive Customers' Behavior Prediction Problem [J]. The Seventh International Conference on Machine Learning and Applications (ICMLA'08), 677~681.

[2] E.W.T. Ngai, Li Xiu, D.C.K. Chau. Application of data mining techniques in customer relationship management: A literature review and classification [J]. Expert Systems with Applications, 2009: 2592~2602.

[3] HUANG Lan, ZHOU Chun-guang, ZHAI Yan-dong. Research on the Constructions of Intelligent Decision Support System in Automobile Marketing-Oriented[J]. China Management Informationization, 2008, 11(12):79~82.

[4] J.T.Hartigan, M.A.Wong. A K-Means Clustering Algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), Vol.28, No. 1(1979), pp. 100-108.

[5] Ma Jun, Yang Fan. The Forecast Model Study for Chinese Customer's Requirement on Telematics.Shanghai Auto,2011,01, pp.41-45.

[6] YanDa. The Design and Implementation of Customer Satisfaction Evaluation System Based on Fuzzy Method of Evaluation. Jinlin University,2010,04.

[7] Wang Ying-shuang, Zhou yu-qin, Huang Lan. Research on Prediction of Automobile Customer Behaviors Based on CRM. Journal of Jilin University(Information Science Edition),2008,06.

An Approach to Selecting Proper Dimensions for Noisy Data

Yong Shi and Jerry Meisner
Department of Computer Science and Information Systems
Kennsaw State University
Building 11, Room 3060
Kennesaw, GA 30144

Abstract— *In this paper, we present our research on selecting proper dimensions for noisy data. We select those dimensions with dramatic variation of data distribution through the data mining processes as good dimension candidates for further data analysis. This dimension selection approach can assist to improve the performance of existing data analysis approaches and affect the results of well known algorithms.*

1. Introduction

Data sets are generated everyday in various fields. A lot of approaches have been designed to analyze these data, trying to find the patterns and other valuable information hidden in the data sets ([7], [16], [4], [11], [13], [8], [21], [20], [19], [10], [6], [15], [9], [3], [2]). A lot of real data sets have irrelevant features, and some of them affect the accuracy of data mining algorithms such as clustering, outlier detection, etc. High dimensional data sets continue to pose a challenge to data mining algorithms at a very fundamental level. To improve the data analysis performance, researchers proposed the method of dimension reduction ([2], [1], [18]) in which a data set is transformed to a lower dimensional space but still preserves the major information it carries, thus further processing is simplified without compromising the quality of the data mining results. Dimension reduction is often used in clustering, classification, and many other machine learning and data mining applications.

Different ways were proposed to reduce the dimensionality of the data space in which certain data analysis process is performed. One category of approaches optimally select a subset from the existing dimensions. A different kind of approaches generate new dimensions as linear or unlinear combination of old dimensions. For example, some approaches use principal component analysis through singular value decomposition [12] for numerical attributes which defines new attributes (principal components) as mutually-orthogonal linear combinations of the original attributes. In information retrieval, latent semantic indexing uses singular value decomposition to project textual documents represented as document vectors. Singular value decomposition is shown to be the optimal solution for a probabilistic model for document/word occurrence. However, this kind of approaches has a major drawback in that the generated low dimensional subspace has no intuitive meaning to users. There are also some other approaches that use random projections to generate subspaces.

Data preprocessing procedures can greatly benefit the utilization and exploration of real data. There are many data preprocessing procedures designed in the data mining field such as data cleaning and data transformation.

2. Related Work

Recently various approaches have been designed to select proper dimensions. Ding etc. [5] introduced a LDA-Km framework that enhances K-Means clustering by combining it with linear-discriminant analysis (LDA) to dynamically select proper sub-spaces to form clusters. They also presented two variants of LDA-Km algorithm and discusses their relationships to earlier approaches. Lee etc. [14] presented a method of clustering that uses a PCA based dimensionality reduction with a Frechet mean. They provided examples in the form of two-dimensional images using only color and intensity as the cluster defining features and demonstrated how their algorithm performs after periods of iterations. Sanguinetti [17] introduced a probabilistic latent variable model inspired by the Extreme Component Analysis (XCA) model to perform linear dimensionality reduction on data sets which contain clusters. The author also used real and artificial data sets to demonstrate the performance of the model.

3. Dimension Selection

In this paper, we propose a dimension selection approach for multi-dimensional data analysis to address the inadequacies of current data mining algorithms in handling multi-dimensional data. It tends to solve the dimension selection problem from a new perspective.

Dimension selection process is performed based on the difference of the data distribution projected on each dimension through data mining processes. Here the data mining processes can be various types of approaches such as the adjustment of data positions based on certain criteria, dynamic data analysis process such as insertion, update and deletion of data points, etc. For those dimensions which make large contribution to the good results of data analysis,

the alterations of data distributions on them through the data mining processes are significant. By evaluating the statistics of the data distribution through data mining processes, good dimension candidates for further data analysis steps can be chosen efficiently, and unqualified ones can be discarded. It can improve the performance of existing data mining algorithms such as clustering and outlier detection. In our approach, we consider optimal selection of a subset of existing dimensions for the purpose of easy interpretation.

The main strategy is that for those dimensions which make large contribution to the good results of data analysis, the alterations of the data distribution on them through the process are very prominent. We select good dimension candidates based on the observation of the difference of the statistics status of each dimension.

The alteration of the histogram distribution through the data mining process on each dimension defines the unique feature of the data distribution on a certain dimension better than the histogram distribution itself. Although some dimensions have much wider histogram distribution than other dimensions before the data mining processes, these dimensions might give poor support for the following data analysis processes. By evaluating the ratio of the histogram distribution (referred to as bin difference in the following sections) on a dimension instead of the histogram distribution itself, those dimensions can be discarded efficiently.

In this section we formalize our dimension selection approach. We first introduce a few notations and definitions before we describe our approach. Let the input *d*-dimensional data set be **X**: $\mathbf{X} = \{\vec{X}_1, \vec{X}_2, ..., \vec{X}_n\}$, which is normalized to be within the hypercube $[0,1]^d \subset R^d$.

The dimensions in this *d*-dimensional data space are D_1, D_2, ..., D_d. Each data point $\vec{X}_i$ is a *d*-dimensional vector: $\vec{X}_i = \{X_{i1}, X_{i2}, ..., X_{id}\}$. Data mining process is performed accordingly.

For each dimension of **X**, a histogram is set up based on the current data distribution information: $\mathbf{H} = \{\mathbf{H_1}, \mathbf{H_2}, ..., \mathbf{H_d}\}$. The number of segments on each dimension is not necessarily the same. Let η_i be the number of bins in the histogram on the *i*th dimension. We denote each histogram as: $\mathbf{H_i} = \{\mathrm{H}_{i1}, \mathrm{H}_{i2}, \ldots, \mathrm{H}_{i\eta_i}\}$, in which H_{ij}, j=1, 2, ..., η_i is a bin for the histogram.

We denote the region of bin H_{ij} as [Min_{ij}, Max_{ij}] for $j = \eta_i$, or [Min_{ij}, Max_{ij}) otherwise. The size of bin H_{ij} is the amount of data points whose *i*th attributes are in the region of H_{ij}: $|H_{ij}| = |\{\mathbf{X_l}|Min_{ij} \leq X_{lj} \leq Max_{ij}\}|$ for $j = \eta_i$, or $|H_{ij}| = |\{\mathbf{X_l}|Min_{ij} \leq X_{lj} < Max_{ij}\}|$ for $j \neq \eta_i$.

For each dimension D_i, i=1, 2, ...,d, let MAX_{H_i} be the maximum size of the bins on D_i: $MAX_{H_i} = Max_{j=1}^{\eta_i}|H_{ij}|$; let MIN_{H_i} be the minimum size of the bins on D_i: $MIN_{H_i} = Min_{j=1}^{\eta_i}|H_{ij}|$. Let γ_{H_i} be the ratio between MAX_{H_i} and the maximum value of MIN_{H_i} and 1:

$$\gamma_{H_i} = \frac{MAX_{H_i}}{Max(MIN_{H_i}, 1)}. \quad (1)$$

The reason we specify $Max(MIN_{H_i}, 1)$ is that there might be the cases that some bins on dimension D_i do not have any data points at all, thus their sizes might be 0. We call γ_{H_i} the bin difference on D_i. Figure 1 shows an example of the bin difference on dimension D_i before the data mining processes.

For each dimension D_i, we first calculate the bin difference γ_{H_i} defined in equation (1).

The data mining process is then performed. In the data mining step, the data distribution can be altered on each dimension D_i, i=1, 2, ..., d.

After the data mining process, let $|H'_{ij}|$ be the size of bin H_{ij} after the data mining process, let MAX'_{H_i} be the maximum size of the bins on D_i: $MAX'_{H_i} = Max_{j=1}^{\eta_i}|H'_{ij}|$; let MIN'_{H_i} be the minimum size of the bins on D_i: $MIN'_{H_i} = Min_{j=1}^{\eta_i}|H'_{ij}|$. Let γ'_{H_i} be the ratio between MAX'_{H_i} and the maximum value of MIN'_{H_i} and 1:

$$\gamma'_{H_i} = \frac{MAX'_{H_i}}{Max(MIN'_{H_i}, 1)}. \quad (2)$$

Figure 2 shows an example of the bin difference on dimension D_i after data mining processes.

We compute the change of the bin difference on D_i before data mining processes and after data mining processes, and calculate their ratio:

$$\tilde{\gamma_{H_i}} = \frac{\gamma'_{H_i}}{\gamma_{H_i}} \quad (3)$$

Dimensions having prominent bin difference alteration through the data mining process are selected as good candidates for the following data analysis processes.

We sort dimensions D_1, D_2, ..., D_d in descending order according to $\tilde{\gamma_{H_i}}$. Suppose the dimension list after sorting is $\hat{D}_1$,$\hat{D}_2$, ..., $\hat{D}_d$, we select the first several dimensions. The cut on the dimension list is performed as follows. To keep most valuable dimensions, the second half of the ordered dimension list is checked, and the cut spot is set on the first sharp descent dimension.

3.1 Time and space analysis

Throughout the dimension selection process, we need to keep track of the histogram information of each dimension. Suppose the maximum bin amount of a dimension is η_{max}, and the dimensionality is d. The dimension selection process occupies $O(d\eta_{max})$ space. We also need to keep track of the positions in the d-dimensional data space for each data points, which occupies $O(nd)$ space.

In order to calculate γ'_{H_i}, γ_{H_i}, and γ_{H_i}, we need to perform linear searches to find out the maximum size (MAX_{H_i}) and minimum size (MIN_{H_i}) of the bins on

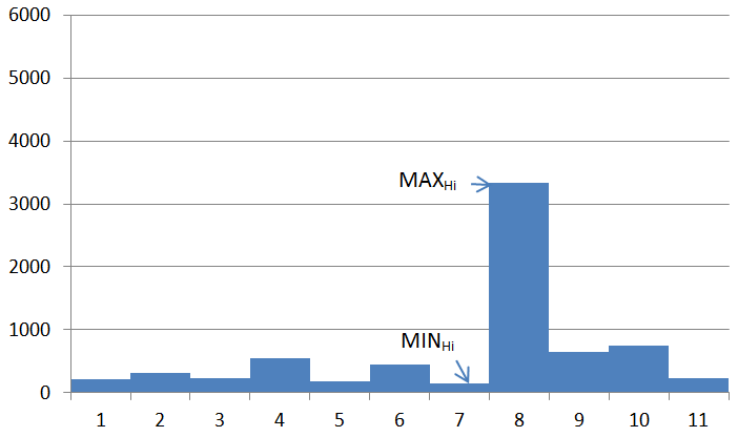


Figure 1: Data distribution on dimension D_i before data mining process

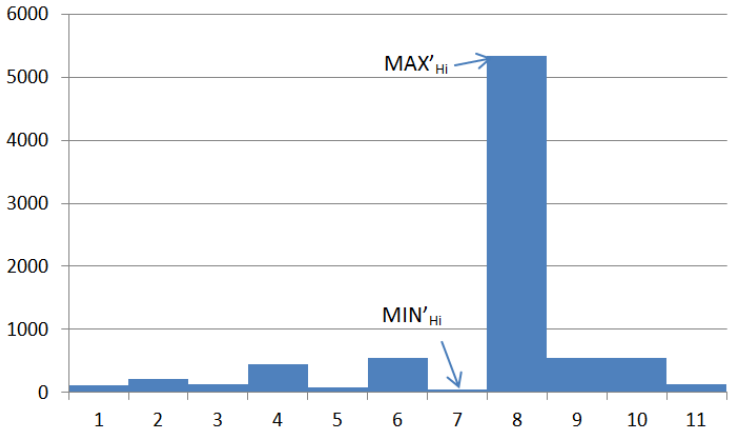


Figure 2: Data distribution on dimension D_i after data mining process

each dimension D_i (i=1, 2, ..., d) before the data mining processes, and the maximum size (MAX'_{H_i}) and minimum size (MIN'_{H_i}) of the bins on each dimension D_i (i=1,2,...,d) after the data mining processes. The time of the dimension sorting based on the ratio $\gamma_{\tilde{H}_i}$ is $O(d \log d)$.

4. Conclusion and Future Work

In this paper, we present a dimension selection approach for multi-dimensional noisy data. We select good dimension candidates for further data analysis based on the observation of the alteration of the bin difference of each dimension through the data mining processes. Data analysis methods still pose many open issues. Many approaches rely on the information of the data sets to a certain degree. Improvement of the information retrieval from data sets will greatly benefit the implementation. This is one of our primary further researches.

We will also conduct experiments on both synthetic and real data sets with noise to test and demonstrate the effectiveness and efficiency of our approach.

References

[1] C. C. Aggarwal, C. Procopiuc, J. Wolf, P. Yu, and J. Park. Fast algorithms for projected clustering. In *Proceedings of the ACM SIGMOD CONFERENCE on Management of Data*, pages 61–72, Philadelphia, PA, 1999.

[2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 94–105, Seattle, WA, 1998.

[3] Ankerst M., Breunig M. M., Kriegel H.-P., Sander J. OPTICS: Ordering Points To Identify the Clustering Structure. *Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'99), Philadelphia, PA*, pages 49–60, 1999.

[4] D. Barbara, W. DuMouchel, C. Faloutsos, P. J. Haas, J. H. Hellerstein, Y. Ioannidis, H. V. Jagadish, T. Johnson, R. Ng, V. Poosala, K. A. Ross, and K. C. Servcik. *The New Jersey data resuction report*. Bulletin of the Technical Committee on Data Engineering, 1997.

[5] C. Ding and T. Li. Adaptive dimension reduction using discriminant analysis and k-means clustering. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 521–528, New York, NY, USA, 2007. ACM.

[6] M. Ester, K. H.-P., J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996.

[7] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAI Press, 1996.

[8] S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. In *Proceedings of the ACM SIGMOD conference on Management of Data*, pages 73–84, Seattle, WA, 1998.

[9] S. Guha, R. Rastogi, and K. Shim. Rock: A robust clustering algorithm for categorical attributes. In *Proceedings of the IEEE Conference on Data Engineering*, 1999.

[10] A. Hinneburg and D. A.Keim. An efficient approach to clustering in large multimedia databases with noise. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 58–65, New York, August 1998.

[11] J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume I,Statistics.*, 1967.

[12] K. R. Kanth, D. Agrawal, and A. Singh. Dimensionality reduction for similarity searching in dynamic databases. In *Proceedings of the ACM SIGMOD CONFERENCE on Management of Data*, pages 166–176, Seattle, WA, 1998.

[13] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley & Sons, 1990.

[14] S. mook Lee and A. L. Abbott. Dimensionality reduction and clustering on statistical manifolds. In *in Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–7, 2007.

[15] R. T. Ng and J. Han. Efficient and Effective Clustering Methods for Spatial Data Mining. In *Proceedings of the 20th VLDB Conference*, pages 144–155, Santiago, Chile, 1994.

[16] T. Redman. *Data Quality: Management and Technology.* Bantam Books, 1992.

[17] G. Sanguinetti. Dimensionality reduction of clustered datasets. *IEEE Transactions on Machine Intelligence and Pattern Analysis*, 30:2008.

[18] T. Seidl and H. Kriegel. Optimal multi-step k-nearest neighbor search. In *Proceedings of the ACM SIGMOD conference on Management of Data*, pages 154–164, Seattle, WA, 1998.

[19] G. Sheikholeslami, S. Chatterjee, and A. Zhang. Wavecluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of the 24th International Conference on Very Large Data Bases*, 1998.

[20] W. Wang, J. Yang, and R. Muntz. STING: A Statistical Information Grid Approach to Spatial Data Mining. In *Proceedings of the 23rd VLDB Conference*, pages 186–195, Athens, Greece, 1997.

[21] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An Efficient Data Clustering Method for Very Large Databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 103–114, Montreal, Canada, 1996.

Adaptive Neuro Fuzzy Networks based on Quantum Subtractive Clustering

Ali Mousavi*, Mehrdad Jalali and Mahdi Yaghoubi

***Abstract*—Data mining techniques can be used to discover useful patterns by exploring and analyzing data and it's feasible to synergistically combine machine learning tools to discover fuzzy classification rules. In this paper, an adaptive neuro fuzzy network with TSK fuzzy type and an improved quantum subtractive clustering has been developed. Quantum clustering (QC) is an intuition from quantum mechanics which uses Schrödinger potential and time-consuming gradient descent method. The principle advantage and shortcoming of QC is analyzed and based on its shortcomings, an improved algorithm through a subtractive clustering method is proposed. Cluster centers represent a general model with essential characteristics of data which can be use as premise part of fuzzy rules. The experimental results revealed that proposed Anfis based on quantum subtractive clustering yielded good approximation and generalization capabilities and impressive decrease in the number of fuzzy rules and network output accuracy in comparison with traditional methods.**

***Index Terms*—quantum clustering, subtractive clustering, fuzzy rules, Anfis.**

I. INTRODUCTION

The last decade has seen an explosive growth in the generation and collection of data, advances in data minig and automation of predictor systems. Hence a lot of new techniques and tools that can intelligently and automatically assist in transforming this data into useful knowledge have been proposed.

An Adaptive Neuro Fuzzy Inference System (ANFIS) is a framework based on the concepts of fuzzy systems which have been improved by artificial neural networks. So Anfis tries to integrate advantages of fuzzy systems and artificial neural networks. This means that the neural networks present learning capabilities to fuzzy systems via a connectionist structure. Moreover fuzzy systems provide a framework to work with human knowledge and the uncertain world [1, 2].

Methods of data clustering are usually based on geometric or probabilistic considerations. The problem of unsupervised learning of clusters based on locations of points in data-space is in general ill-defined [3, 6]. Hence intuition based on other fields of study may be useful in formulating new heuristic procedures. There are a lot of clustering methods which leads to generate adequate numbers of fuzzy rules. The cluster centers represent a general model with essential characteristics of data. Each cluster center can be use as premise part of a fuzzy rule [2, 6].

Here we use a model based on tools that are borrowed from quantum mechanics. The quantum clustering introduced by Horn [3] is a new and unique clustering method that is an extension of idea inherent to scale-space and support-vector clustering. In addition, this is represented by the Schrödinger equation, which is a potential function that can analytically be derived from a probability function. The effectiveness of this clustering has been demonstrated on pattern recognition [1, 3].

Nonetheless the Quantum Clustering (QC) method has some drawbacks. It is difficult to determine the number of valid cluster centers, because this QC severely depends on a one-variable parameter representing the scale of its Gaussian kernel, and it is hard to deal with a high dimension of input space so QC method is modified in a simple manner [1, 3].

Also a subtractive clustering method has been used to determine the adequate number of cluster centers. This is a density based clustering that proposed by Chiu in 1994 [7].

Therefore we propose a new method to construct an adaptive neuro fuzzy network with a TSK fuzyy type. Also we use a modified QC to determine the premise part of fuzzy rules. Moreover subtractive clustering method is applied to determine the optimal number of clusters. We performed a learning method by a hybrid learning scheme using back propagation (BP) and a least-square estimator (LSE). The experiments used the well-known automobile mile-per-gallon (MPG) prediction dataset which consists of 392 records. In this example, six input variables are composed of a car's cylinder number, displacement, horsepower, weight, acceleration, and model year. The output variable to be predicted by the six input variables is the fuel consumption of the automobile.

This paper is organized as follows: In section II, we describe basics of Anfis. The Quantum Clustering method, our modified Quantum clustering and subtractive clustering are presented in section III and followed Section IV gives the experimental results. Finally, we conclude the paper in section V.

*Ali Mousavi is with Department of Artificial Intelligence, Faculty of Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran.
(E-mail: mousavi@mshdiau.ac.ir).
Mehrdad Jalali is with department of Artificial Intelligence, Assistant Professor (PhD), Mashhad Branch, Islamic Azad University, Mashhad, Iran.
(E-mail: mehrdadjalali@ieee.org)
Mahdi Yaghoubi is with department of Electrical Engineering, Assistant Professor (PhD), Mashhad Branch, Islamic Azad University, Mashhad, Iran.
(E-mail: yaghoubi@mshdiau.ac.ir)

II. BASICS OF ANFIS

ANFIS is a neuro-fuzzy system developed by Jang [2]. It has a feed-forward neural network structure where each layer is a neuro-fuzzy system .In this section, we describe the basic architecture, and learning rules of the Anfis. The Anfis structure identification involves two phases: 1) structure identification and 2) parameter identification. The former is related to determining the number of fuzzy if–then rules and a proper partition of the input space. The latter is concerned with the learning of model parameters, such as membership functions and linear coefficients. As shown in Fig. 1, this network is composed of five layers that introduced by Jang et al [2]. Nevertheless the Anfis suffers from curse of dimensionality that the number of extracted fuzzy rules increases exponentially due to grid partitioning of input data. In this paper we use scatter one instead to find adequate number of rules through new proposed clustering method. For a zero and first order Sugeno Fuzzy model common rules are as following respectively. Consider the Anfis model has n inputs [1, 4].

$$R^i: If\ p_1\ is\ A_1^i\ And\ p_2\ is\ A_2^i\ And \ldots And\ p_n\ is\ A_n^i,\quad then\ y^i = a_0^i \tag{1}$$

$$R^i: If\ p_1\ is\ A\,_1^i\ And\ p_2\ is\ A_2^i\ And \ldots And\ p_n\ is\ A_n^i,\quad then\ y^i = a_0^i + a_1^i p_1 + a_2^i p_2 + \ldots + a_n^i p_n \tag{2}$$

Here A_j^i the linguistic value from the jth input variable p_j in the ith fuzzy rule. Also a_j^i is a constant. Now we describe duty of each layer in the following briefly:

Layer 1, the first layer of this network involves n nodes which has a membership function as weight of links. The output determines the firing strength of premise part of associated proposition of a rule. A Gaussian membership function with two parameters (μ, σ) is used where these parameters obtain from clustering phase.

Layer 2, each node in Layer 2 provides the firing strength of the rule by means of multiplication operator. It performs AND operation.

$$w_i = \prod_{j=1}^{n} A_j^i(p_j), \quad i = 1,2,\ldots,n \tag{3}$$

Every node in this layer computes the multiplication of the input values and gives the product as the output according to the above equation.

Layer 3 is the normalization layer which normalizes the strength of all rules according to the following equation.

$$\overline{w_i} = \frac{w_i}{\sum_{i=1}^{r} w_i} \tag{4}$$

Where w_i is the firing strength of the ith rule which is computed in Layer 2. Node i computes the ratio of the ith rule's firing strength to the sum of all rules' firing strength. Layer 4, every node in the fourth layer computes a product operation (And) between the normalized firing strength from layer 3 and consequent part of corresponding rule. It means output of this layer is $\overline{w_i}.y^i$ that shows firing strength of each rule.

Layer 5, the single node of this layer aggregates all of incoming inputs. Here $\hat{y}$ is the final predicted output of the network.

$$\hat{y} = \sum_{i=1}^{r} \overline{w}_i y^i \tag{5}$$

Because of the overall output is obtained through a summation operator, we don't need to complicated defuzzification process.

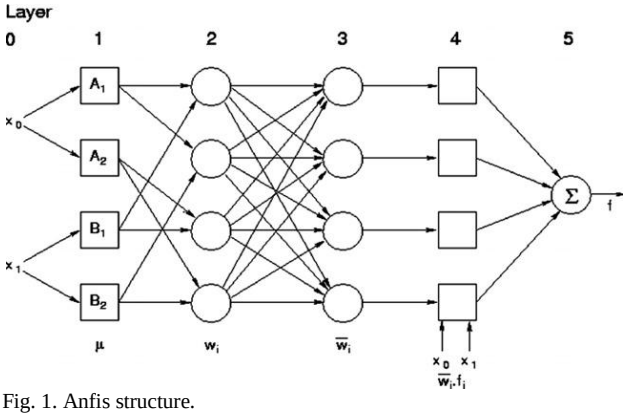


Fig. 1. Anfis structure.

III. RULE EXTRACTION VIA CLUSTERING

A. Quantum Clustering

Clustering is the procedure that classifies the set of physical abstract objects to several clusters composed of similar objects. Methods of data clustering are usually based on geometric or probabilistic considerations [6]. We can consider intuition based on other fields of study to formulating new clustering procedures. Quantum clustering introduced by Horn [3] is a novel method from quantum mechanics that research an operator in Hilbert space expressed by Schrödinger equation, the solution is wave-function. When the wave-function is given, they can work out the potential function by Schrödinger equation, which will determine the distribution of particle, and the cluster centers are the k minimum in the potential. According to the quantum theory, particle with lower potential vibrate less, and relatively stable, so we can treat this kind of particle as the cluster centers, and then distribute data points to the associated clusters [5].

In QC algorithm, wave function in equation (6) is used to describe the sample's distribution and produce the estimator parameter of Parzen Window.

$$\Psi(x) = \sum_{i=1}^{n} e^{-(x-x_i)^2/2\sigma^2} \quad (6)$$

Where x_i is the ith data point.

In the QC algorithm, we use the k minima of Schrödinger potential to determine the location of the cluster centers. This potential is part of the Schrödinger equation (7), for which ψ(x) is a solution.

$$H\psi \equiv \left(-\frac{\sigma^2}{2}\nabla^2 + V(x)\right)\psi(x) = E\psi(x) \quad (7)$$

Where H is the Hamiltonian, V is the potential energy, E is the energy eigenvalue and $E = {}^{d}/_{2}$. When $V(x)$ is positive definite, we obtain $V(x) \geq 0$. Hence, E is defined as follows:

$$E = -min\frac{\frac{\sigma^2}{2}\nabla^2\psi}{\psi} \quad (8)$$

The eigenvalue E of Schrödinger's equation is the lowest eigenfunctions of the operator H representing the ground state. Moreover, when the minima of $V(x)$ are defined as the cluster centers, the assignments of data points to clusters are obtained by a gradient descent algorithm allowing auxiliary point variables $z_k(0) = x_k,\ k = 1, 2, \ldots, N$, to follow dynamics as follows:

$$y_i(t + \nabla t) = y_i(t) - \eta(t)\nabla V(y_i(t)) \quad (9)$$

Here z_k is used as the pre-clusters, and $\eta(t)$ is the learning rate [3].

B. Modified QC

The proposed QC algorithm is a classic quantum clustering algorithm, but it does have some defects. It's difficult to find the learning turns in the gradient descent iterative procedure which seeks the potential minimum. Accordingly that effects on the number of cluster centers and cluster accuracy. Furthermore the gradient descent is a time consuming iterative procedure. The proposed method improves QC to solve above problems by elimination of gradient descent procedure and using a subtractive clustering algorithm. The proposed method is illustrated as following:

First, calculate the vector of potential V for all data samples according to (8) equation. Second, use subtractive clustering explained in section C to obtain optimal number of cluster centers as parameter, k. Third, the cluster centers are calculated through following stages:

1- Sort samples in vector V by ascending.

2- Choose first Nth associated data points as cluster centers.

C. Subtractive Clustering

Subtractive method belongs to density based method was proposed by Chiu in 1994. Chiu suggested an improved version of the mountain method, which is proposed by Yager and Filev [7]. Here we use SC to obtain optimal number of cluster centers.

The SC algorithm assumes each data point is a potential cluster center. Consider a collection of N data point in m-dimensional space. Then calculate a density measure for data points as follows [1]:

$$D_k = \sum_{k'=1}^{N} \exp\left(-\frac{\|z_k - z_{k'}\|^2}{r_\alpha/2)^2}\right) \quad (10)$$

Where r_α is a constant and $z_k\ (k = 1,2,3,\ldots,N)$ is kth data point. A data point with many neighboring data points will have a high density value. Thus, data points with a high value of potential are more suited to be the potential cluster centers. A data point with highest density value is selected as the first cluster center. Then the density measure for all of data points will be update by following equation:

$$D_k = D_k - D_{v1}\exp\left(-\frac{\|z_k - z_{v1}\|^2}{r_b/2)^2}\right) \quad (11)$$

Where $r_b = 1.5r_\alpha$ and v_1 is center of selected cluster. The data points around the first cluster center will have reduced density measures. Thus, these data points never selected as the next cluster center. The process continues until a sufficient number of cluster centers were obtained [5].

IV. Experimental Result

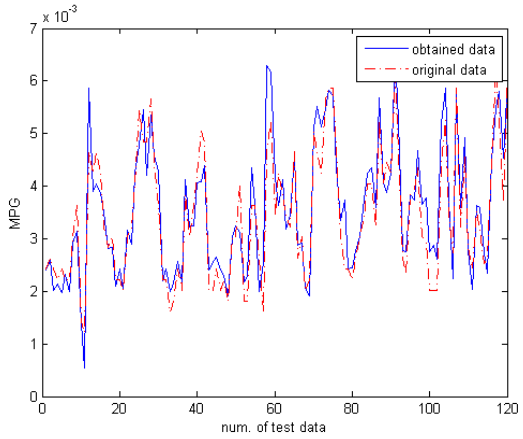
Generally effective partitioning of input space can reduce number of fuzzy rules and increase learning speed of Anfis. In this paper we use a quantum subtractive clustering to determine fuzzy rules, and then a modified Anfis is applied to data set to predict the output. The experiments used the well-known automobile mile-per-gallon (MPG) prediction dataset which consists of 392 records. In this example, six input variables are composed of car's cylinder number, displacement, horsepower, weight, acceleration, and model year. The output variable to be predicted by the six input variables is the fuel consumption of the automobile. At first records with missing values has been eliminated, then we normalize data to [0, 1] interval. Data set divided into two partition; train data and test data, according to even record and odd records from original dataset, respectively. Here we use training data set to construct and learn the model while test data is used to validate the model.

At first number of cluster centers (k) is obtained by subtractive clustering, then we use modified QC to find k cluster centers. Finally, we construct the Anfis model and efficient number of fuzzy rules will be generated. Here we have obtained 20 rules. The network output predicts the fuel consumption of automobile.

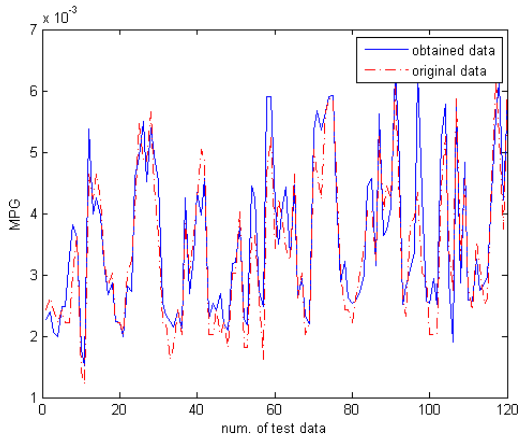
We have compared our new Quantum Subtractive Clustering method with traditional QC applied in Anfis [1].

Fig. 2(a) shows comparison results between the desired and traditional QC model outputs for test data. Here horizontal axis shows number of test data and vertical axis determines desired and model fuel consumption. Also Fig. 2(b) shows comparison diagrams between the desired and our model outputs for test data set.

As shown in Fig. 2, it is obvious that the difference between desired and proposed model output is less than previous method. As the result, proposed method has better approximation and generalization capability.



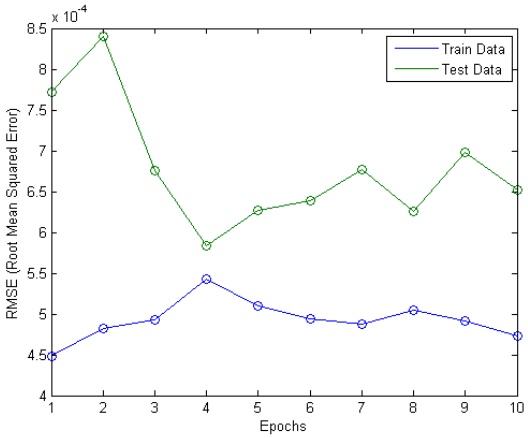
(a)



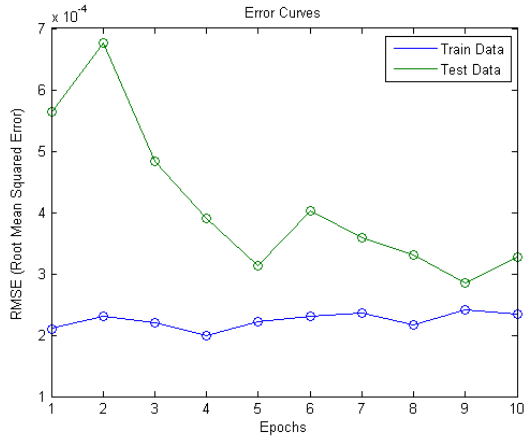
(b)

Fig. 2. Approximation and generalization capability in previous method (a) and proposed method (b).

Fig. 3 shows the root mean square error (RMSE) for both new and previous methods during 10 epochs respectively. As Fig. 3(a) is shown, at the beginning there is a lot of difference between test and train error rate, because of initial epochs of model, but it decreases gradually and reaches to a reasonable value in comparison with error rate in previous method. The calculated average testing error for new method is 0.0029 while it's 0.0056 for previous method. It shows a considerable improvement in comparison with before.



(a)



(b)

Fig. 3. Comparison of RMSE in the training and test data in proposed method (a) and previous method (b).

V. CONCLUSION

An Adaptive Neuro Fuzzy Network with a TSK fuzzy type combined with an improved quantum subtractive clustering method to obtain appropriate number of fuzzy rules is proposed. The subtractive clustering, a density based algorithm, is used to determine number of cluster centers. Moreover a modified quantum clustering, an idea from quantum mechanics is applied to obtain cluster centers. Cluster centers represent a general model with essential characteristics of data which can be use as premise part of fuzzy rules. It caused impressive decrease in number of fuzzy rules and network accuracy. Finally we construct our model to predict fuel consumption in MPG dataset.

The experimental results showed the proposed method has good approximation and generalization capabilities compared with traditional one. Moreover our new quantum based subtractive clustering is more accurate and faster. As the result, fuzzy rules obtained from quantum based clustering improve our model undoubtedly.

Our future work is replacing hard quantum subtractive clustering method with fuzzy quantum subtractive clustering approach to increase the accuracy of method and deal with real problems in our world where fuzzy concept is a solution.

REFERENCES

[1] K. Sung-Suk and K. Keun-Chang, "Development of Quantum-Based Adaptive Neuro-Fuzzy Networks", *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, pp. 91-100, 2010.

[2] Jang, et al., Neuro-Fuzzy and Soft Computing. Massachusetts, USA: Prentice Hall, 1997.

[3] D. Horn and A. Gottlieb, "The Method of Quantum Clustering", Tel Aviv University, Tel Aviv 69978, Israel

[4] M. B. Menhaj, Fundamental of Neural Networks. University of Amir Kabir,Tehran, IRAN, 2003.

[5] Y. Zhang, et al., "Quantum Clustering Algorithm based on Exponent Measuring Distance", in Knowledge Acquisition and Modeling Workshop, IEEE International Symposium on, pp. 436-439, 2008.

[6] R.O. Duda, P.E. Hart and D.G. Stork, Pattern Classification. Wiley-Interscience, 2001.

[7] D.-W. Kim, et al., "A kernel-based subtractive clustering method", Pattern Recognition Letters, vol. 26, pp. 879-891, 2005.

A Clustering Approach to Unsupervised Attack Detection in Collaborative Recommender Systems

Runa Bhaumik[1], **Bamshad Mobasher**[1] **and Robin Burke**[1]
[1] College of Computing and Digital Media, Center for Web Intelligence, DePaul University, Chicago, Illinois, USA

Abstract— *Securing collaborative filtering systems from malicious attack has become an important issue with increasing popularity of recommender Systems. Since recommender systems are entirely based on the input provided by the users or customers, they tend to become highly vulnerable to outside attacks. Prior research has shown that attacks can significantly affect the robustness of the systems. To prevent such attacks, researchers proposed several unsupervised detection mechanisms. While these approaches produce satisfactory results in detecting some well studied attacks, they are not suitable for all types of attacks studied recently.*

In this paper, we show that the unsupervised clustering can be used effectively for attack detection by computing detection attributes modeled on basic descriptive statistics. We performed extensive experiments and discussed different approaches regarding their performances. Our experimental results showed that attribute-based unsupervised clustering algorithm can detect spam users with a high degree of accuracy and fewer misclassified genuine users regardless of attack strategies.

Keywords: Recommender Systems, Collaborative Filtering, Clustering

1. Introduction

Recommender systems are common targets for malicious attackers. The product sellers who are interested in promoting their own product to generate more revenue might be interested in biasing these recommender systems, which have an influence on the customers. Such attackers can use automated tools to create and throw fake profiles in the recommender database, which either may rate the items high or may rate the opponent's items low. These collaborative recommender systems must always be open to users, in order to get their opinions. This is the reason why designing an attack-proof system is a complicated task.

In recent years, research has shown that personalization based on explicit user feedback, typically in the form of ratings, are vulnerable to "profile injection" or "shilling" attacks [1], [2], [3]. These attacks consist of a set of attack profiles, each containing biased rating data associated with a fictitious user identity. Since "shilling" profiles look similar to authentic profiles, it is difficult to identify them.

Early detection algorithms [3] exploited signatures of attack profiles and were moderately accurate. However, these detection algorithms suffered from low accuracy in detecting shilling profiles, since they looked at individual users and mostly ignored the combined effect of such malicious users. Moreover, these algorithms did not perform well when the spam profiles are obfuscated.

Unsupervised anomaly detection approaches address these issues by training on an unlabeled dataset. These methods involve much lesser computational effort as compared to supervised approaches, especially if training data has to be generated. It also facilitates online learning and improves detection accuracy. There has been significant research interest focused on detecting attack profiles in a more unsupervised fashion [4], [5], [6], [7]. Recent approaches are PCA-based clustering, unRAP and Neyman-Pearson statistical detection techniques.

Mehta et al. [5] showed that clustering based on Principal Component Analysis (PCA) performed very well against standard attacks when evaluated on Movielens dataset. The motivation behind this approach is that attacks consist of multiple profiles which are highly correlated with each other, as well as having high similarity with a large number of authentic profiles. However, while other attacks can be detected with high accuracy and fewer misclassified authentic users, performance of AOP attack (a recently studied obfuscated attack [7]) detection is not satisfactory.

Bryan et al. [6] observed that the task of identifying attack profiles in recommender systems is similar to the task of identifying bi-clusters in gene microarray expression data. The mean square residue or H-score was introduced by Cheng and Church [8] to find a subset of genes correlated over a subset of experimental conditions. In the context of attack against recommender systems, Bryan et al. [6] used a variance-adjusted H_v score to find the anomalous profiles which are correlated across a subset of items. The objective is that anomalous profiles will have a higher H_v score. They conducted an extensive evaluation on this metric and showed that this metric performed well in separating attack profiles from genuine profiles for most of the attack

strategies discussed in literature [1], [2], [3]. However, our analysis showed that this metric did not perform well in separating attack profiles for a particular type of “segment" attack [3], which is also very effective against recommender systems.

In a recent study, Hurley et al. [7] proposed to use Neuman-Pearson statistical attack detection to identify attack profiles. They developed a statistical model of standard attacks and introduced a new strategy to obfuscate average attack (Average Over Popular Attack). In previous study, the obfuscation strategy was an ad-hoc modification of standard attacks. The goal of this new obfuscated strategy is to minimize the statistical differences between real and attack profiles. One simple way is to choose the filler items from the top x% most popular movies rather than selecting from entire movie set. It is reported that N-P detection strategy with proper statistical attack models is very effective in detecting anomalous profiles. However, the detection model is dependent on attack models. Moreover, the performance of AOP attack (in terms of misclassified real users) is not satisfactory in their unsupervised strategy.

The main contribution in this paper is describing an attribute-based k-means clustering approach to identify attack profiles regardless of attack types. Our research [9], [10] has shown that the generic attributes, modeled on basic descriptive statistics, attempt to capture some of the characteristics that will tend to make an attacker’s profile look different from a genuine user. This motivates us to cluster user profiles based on these generic attributes.

Our approach involves the clustering of neighborhoods into two clusters, where user profiles in smaller cluster have been given low preferences while generating recommendation, therefore be less likely to influence prediction behavior. This approach assumes that normal and anomalous profiles form different clusters in the feature space. Our conjecture is that the number of normal user profiles largely outnumbers the number of anomalous profiles, hence a small cluster will contain mostly “attack" profiles. Our experimental results show that clustering approach can be a simple and effective tool for detecting and removing anomalous profiles.

Our results confirm that unsupervised clustering using k-means algorithm can be as effective as the other proposed methods discussed in the literature, to detect spam users with a high degree of accuracy and fewer misclassified genuine users regardless of attack strategies.

In addition, we compare our algorithm with UnRAP and PCA clustering techniques. We observe that these two approaches also perform extremely good in terms of recall and precision for most of the well studied attacks. However, the UnRAP algorithm performs poorly for segment attack and PCA clustering performs poorly in terms of misclassified authentic users for AOP attack detection compared to attribute-based clustering approach.

2. Attack Types

Several attack types have been identified in literature. In this paper, we will focus on attacks described in [3], [1], [7] which have been widely studied.

2.1 Standard Attacks

Profile injection attacks can be categorized based on the knowledge required by the attacker to mount the attack, the intent of a particular attack, and the *size* of the attack. We characterize attack types by how they identify the selected items, what proportion of the remaining items they choose as *filler items* , and how they assign specific ratings to each set of items. A profile injection attack [3] consists of a set of attack profiles, each containing biased rating data associated with a fictitious user identity. All attack profiles include a target item which the attacker wants recommended more highly (a *push* attack), or wants prevented from being recommended (a *nuke* attack). The remaining items for attack profiles are selected based on the different attack types as follows.

Random Attack. This attack generates profiles in which the items and their ratings are chosen randomly based on the overall distribution of user ratings in the database, except for the target item. This attack is very simple to implement, but it has limited effectiveness.

Average Attack. In the average attack, each assigned rating for a filler item corresponds to the mean rating for that item, across the users in the database who have rated it. This is a very effective attack; however, it requires knowledge about the system.

Segment Attack. An attacker may be interested primarily in a particular set of users – likely buyers of a product. A segment attack attempts to target a specific group of users who may already be predisposed toward the target item. Increased recommendation of the target item to these users may be just as effective as one that raises the recommendation rate across all users.

A typical segment attack profile consists of a number of selected items that are likely to be favored by the targeted user segment, in addition to the random filler items. Selected items are expected to be highly rated within the targeted user segment and are assigned the maximum rating value along with the target item.

2.2 Obfuscated Attacks

Attacks that closely follow one of the attack types mentioned above can be detected and their impact can be

significantly reduced [3]. As a result, an attacker would need to deviate from these known types to avoid detection. In this paper we focus on five ways proposed in [11] and [6] to illustrate the detection challenges that can occur with even minor changes to existing attack types.

Noise Injection. It involves adding a noise to ratings according to a standard normal distribution multiplied by a constant, α, which governs the degree of noise to be added. This noise can be used to blur the profile signatures that are often associated with known attack types. We set $\alpha = 0.2$ in our evaluations and we add noise to all filler and selected items.

User Shifting. It involves incrementing or decrementing (shifting) all ratings for a subset of items per attack profile by a constant amount in order to reduce the similarity between attack users. Shifts can take the positive or negative form, where the amount of shift for each profile is governed by a standard normally distributed random number. In our experiments, we used a positive shift.

Target Shifting. For a push attack, it is simply shifting the rating given to the target item from the maximum rating to a rating one step lower, or in the case of nuke attacks increasing the target rating to one step above the lowest rating. We have chosen to shift the target item for all attack profiles inserted into the database.

Mixed Attack It involves attacking the same target item and producing from different attack models. The attack profiles contain the same amount of average, random, bandwagon and segment attack profiles. A spam detection technique should perform well against such attacks.

Average Over Popular Items(AOP) A simple and effective strategy to obfuscate the Average attack is to choose filler items with equal probability from the top x% of most popular items rather than from the entire collection of items. This attack strategy (AOP) has been proposed recently in [7]".

3. Unsupervised Attack Detection Via Clustering

There have been some recent research efforts aimed at detecting groups of attack profiles and preventing the effects of profile injection attacks. Several unsupervised algorithms that try to identify groups of attack profiles have been proposed [4], [5], [6], [7]. Generally, these algorithms rely on clustering strategies that attempt to distinguish clusters of attack profiles from clusters of authentic profiles. While these approaches produce satisfactory results in detecting some well studied attacks, they are not suitable for all types of attacks studied recently. In this section, we describe our methodology to attack detection in an unsupervised manner.

3.1 Detection Attributes

Our prior research [10] has shown that the generic attributes, modeled on basic descriptive statistics, attempt to capture some of the characteristics that will tend to make an attackers profile look different from a genuine user. Supervised classification algorithms are then applied and experimental results showed that the accuracy of detection of attack profiles is very high. In this work, we applied an unsupervised clustering algorithm based on several attack detection attributes. For the detection algorithm's data set, we used a number of generic attributes and a residue-based attribute to capture these distribution differences, several of which we extended from attributes originally proposed in [9], [6]. These attributes are:

Rating Deviation from Mean Agreement (RDMA). It is intended to identify attackers by examining the profile's average deviation per item, weighted by the inverse of the number of ratings for that item [9]. The attribute is calculated as follows:

$$RDMA_u = \frac{\sum_{i=0}^{n_u} \frac{|r_{u,i} - \overline{r_i}|}{l_i}}{n_u}$$

where n_u is the number of items user u rated, $r_{u,i}$ is the rating given by user u to item i, l_i is the number of ratings provided for item i by all users, and $\overline{r_i}$ is the average of these ratings.

Weighted Degree of Agreement(WDA) It is introduced to capture the sum of the differences of the profile's ratings from the item's average rating divided by the item's rating frequency. It is not weighted by the number of ratings by the user, and is the numerator of the RDMA equation.

Weighted Deviation from Mean Agreement(WDMA) It is designed to help identify anomalies, places a high weight on rating deviations for sparse items. It differs from RDMA only in that the number of ratings for an item is squared in the denominator inside the sum, thus reducing the weight associated with items rated by many users.

Length Variance(LengthVar) It is introduced to capture how much the length of a given profile varies from the average length in the database. If there is a large number of possible items, it is unlikely that very large profiles come from real users, who would have to enter them all manually, as opposed to a soft-bot implementing a profile injection attack. As a result, this attribute is particularly effective at detecting attacks with large filler sizes. This feature is computed as follows:

$$\mathit{LengthVar}_u = \frac{|n_u - \overline{n}|}{\sum_{k \in U} (n_k - \overline{n})^2}$$

where $\bar{n}$ is the average number of ratings across all users.

Residue Based Metrics Residue Based metrics have their origins within bioinformatics, particularly the gene expression analysis domain. Cheng and Church [8] introduced *The Mean Square residue or H-score* to find the subsets of genes correlated over a subset of experimental conditions, in an attempt to better model the gene functional modules within expression data.

Bryan et al. [6] suggested that the variance adjusted mean square residue or H_v-score may be used to aid detection and retrieval of attack profiles against collaborative recommender systems. They established a partial H_v-score for user u is given by:

$$H_v(u) = \frac{\sum i \in I(r_{ui} - r_{Ui} - r_{uI} + r_{UI})^2}{\sum j \in J(r_{ui} - r_{uI})^2}$$

where r_{ui} is the rating that user u has assigned to item i, r_{Ui} is the average rating that item i has received from all users U, r_{uI} is the average rating that user u has given to all items, and r_{UI} is the average rating the data matrix.

3.2 Identifying Anomalous Clusters

Attack profiles tend to be highly correlated, which is a result of the colluded nature of shilling attacks. We conjecture that the attack profiles are smaller in number and dominate one cluster due to their similarity. This assumption leads to a simple approach for attack detection. To identify an attack, we generate profiles for every user in the database. The representation of a profile consists of features based on the detection attributes described in this paper earlier, such that each feature is built from the user's rating data. The profiles are then partitioned into two groups of similar users. In our work we use the locally optimal k-means clustering algorithm. Assuming that the smaller cluster typically corresponds to attack profiles, we mark the smaller cluster as "anomalous" and give low preference to all the profiles in this cluster when generating recommendation.

4. Experimental Evaluation

We now describe our dataset, detailed description of the metrics we have used to evaluate attributes, followed by experimental results including clustering performances for different values of k.

4.1 Dataset

In our experiments, we use the publicly-available MovieLens 100K dataset[1]. The dataset consists of 100,000 ratings on 1682 movies by 943 users. All ratings are integer values between one and five, where one is the lowest (disliked) and five is the highest (liked). Our data includes all users who have rated at least 20 movies. Later in this work, we also evaluate k-means clustering algorithm using a larger Movielens dataset with one million ratings.

For every profile injection attack, we track *attack size* and *filler size*. Attack size is the number of injected attack profiles, and is measured as a percentage of the pre-attack training set. Filler size is the number of filler ratings given to a specific attack profile, and is measured as a percentage of the total number of movies. The results reported below represent averages over all combinations of test users and attacked movies.

The set of attacked items consists of 50 movies whose ratings distribution matches the overall ratings distribution of all movies. Each movie is attacked as a separate test, and the results are aggregated. In each case, a number of attack profiles are generated and inserted into the original database.

4.2 Detection Performance Metrics

In the context of attack detection, our goal is to provide insight into how accurately the clustering algorithm identifies attack profiles as well as authentic profiles. Knowing what percent of authentic profiles are correctly classified as authentic, is a key factor in the performance of a collaborative system. For measuring detection performance, we use the standard measurements of *specificity* and *sensitivity*. Specificity measures the percent of authentic profiles correctly classified, thus providing insight as to the portion of the original authentic profiles that are used for prediction. Sensitivity measures the proportion of attack profiles correctly identified.

4.3 Evaluation of detection attributes

Our previous research carried out an extensive evaluation of generic attack detection attributes. We found that RDMA, WDMA, WDA and Length Variance were the most distinguishing generic attributes of attack profiles in terms of maximum information gain for the best split between attack profiles and genuine users. In this paper, we added residue-based attribute and compared the information gain result with other attributes.

For our experiments, each attack is inserted for a target movie at 3% attack size and a specific filler size. Each of the test movies is attacked at filler sizes of 1%, 3%, 5%, 20%, 40% and 60%, and the results reported are averaged over the 50 test movies and the 6 filler sizes. As our results show in Figure 1, the LengthVar attribute is very important for distinguishing push attack profiles, since few real users rate more than a small percentage of the items. While residue-based attribute shows the next highest gain for average,random and bandwagon attacks, it is less informative than other attributes for segment attack. These

[1] http://www.cs.umn.edu/research/GroupLens/data/

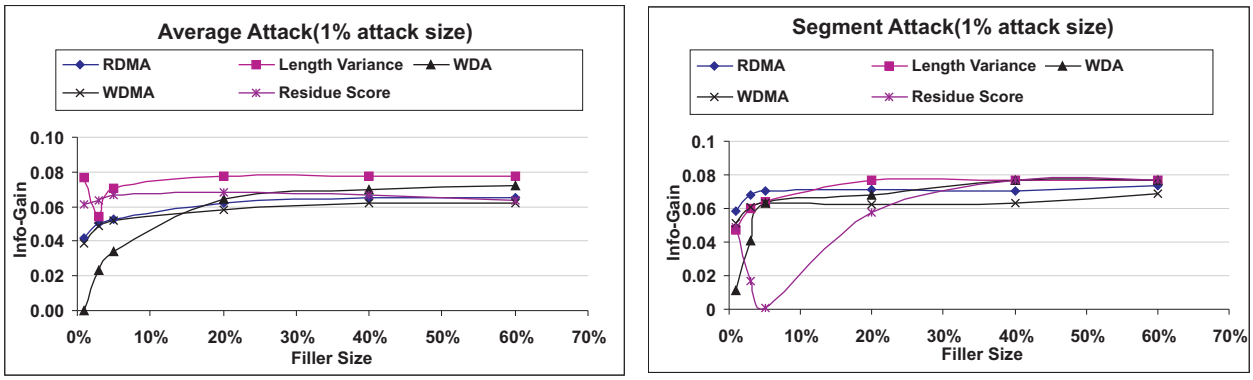


Fig. 1

INFORMATION GAIN RESULT FOR AVERAGE AND SEGMENT PUSH ATTACK ACROSS FILLER SIZES AT 1% ATTACK SIZE.

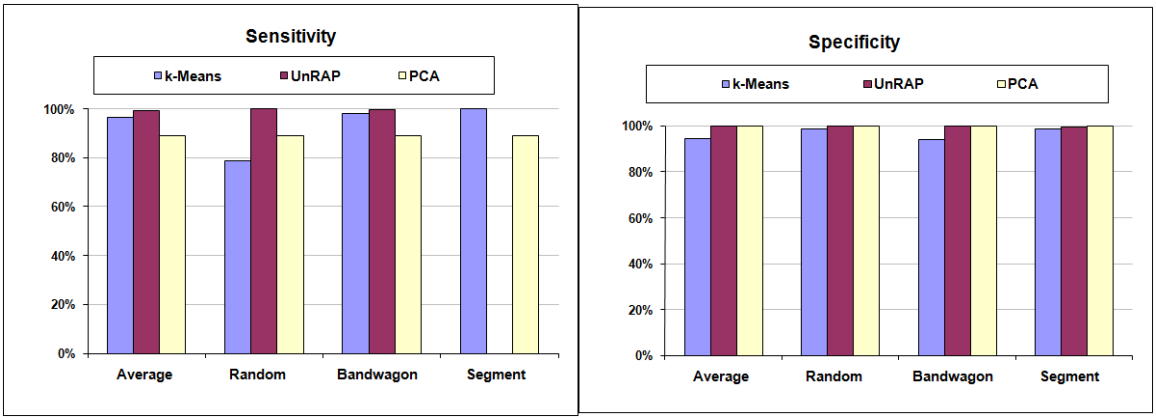


Fig. 2

COMPARING PUSH ATTACK RESULTS WITH UNRAP AND PCA AT 1% ATTACK SIZE AND 3% FILLER SIZE.

results indicate that the clustering approach performed on detection attributes would benefit from an additional residue-based attribute.

4.4 Clustering Performance

In this section we analyze how well our clustering algorithm built on the attributes described above performs at detecting attack profiles. To analyze the performance of clustering, we fix attack size at 1%, profile size at 3% and each of the attacks was inserted individually for various push, nuke and obfuscated attacks. Our goal is to correctly identify the anomalous cluster with fewer genuine user profiles. All results reported here are the average performance over 50 selected items from the Movie-Lens database.

In Figure 2, we present the detection capabilities of k-means algorithm along with other algorithms for push attacks, when $k = 2$. It is observed that the detection performance using k-means is as good as UnRAP algorithm for all other push attacks except segment attack. In fact, UnRAP algorithm couldn't detect any attack profiles for segment attack. The first reason is the approach taken in designing the "segment" attack. A typical segment attack profile consists of a number of selected items that are likely to be favored by the targeted user segment, in addition to the random filler items. Selected items are assigned to the maximum rating value along with the target item and the random filler items are assigned to the minimum rating of 1. So user's mean rating becomes low compared to the other attacks.

The second reason is the approach taken while computing H_v. Due to the large number of null values (rating zero) in the sparse recommender system, the authors in their study [6], computed row, column and matrix means as normal, i.e. over all entries (including null values). So in case of the "segment" attack, each user's rating deviation from their mean rating becomes very low due to the low rating of filler

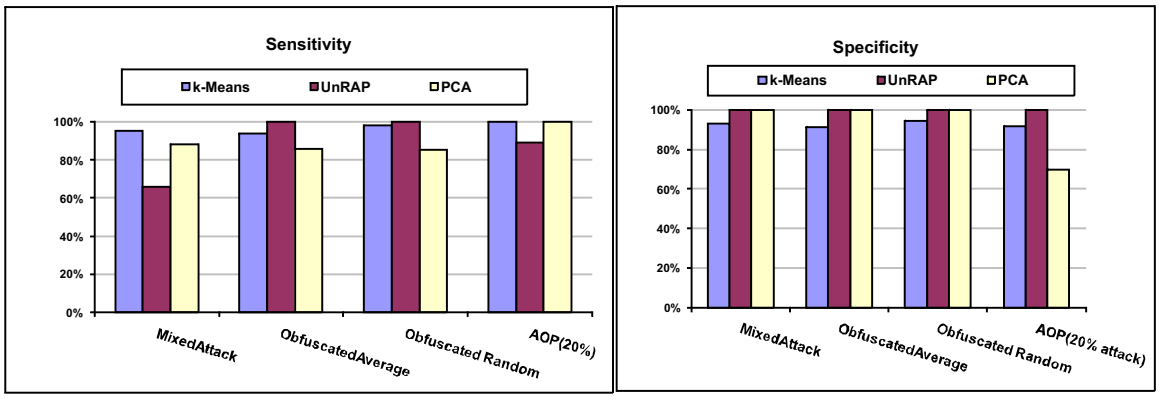


Fig. 3

COMPARING OBFUSCATED ATTACK RESULTS WITH UNRAP AND PCA AT 1% ATTACK SIZE AND 3% FILLER SIZE ($k = 2$).

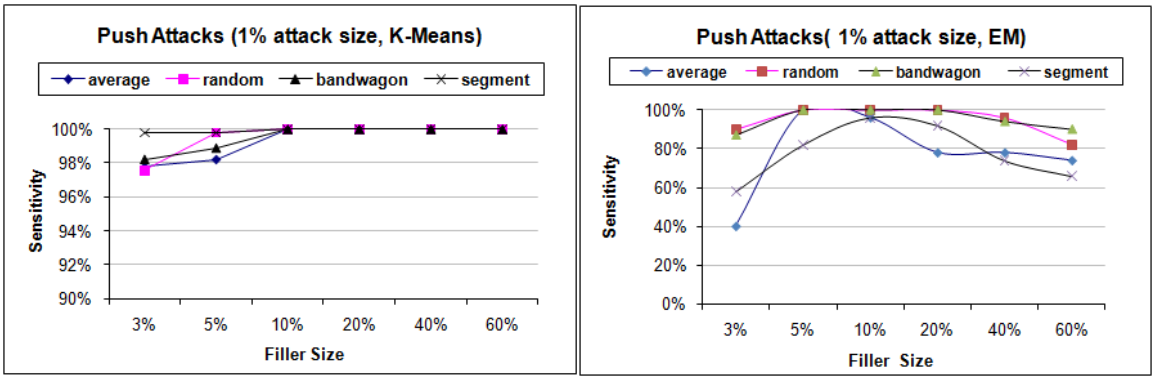


Fig. 4

COMPARING PUSH ATTACK RESULTS AT 1% ATTACK SIZE AND VARYING FILLER SIZES FOR BOTH CLUSTER ALGORITHMS.

items, compared to the other attacks where the rating for the filler items are high. Thus, the H_v scores of the anomalous profiles in “segment attack" cannot be distinguished from the real users and is not successful in detecting segment attack. However, as our results show, the number of real users detected as an attack is very low in UnRAP algorithm, so the specificity is higher than k-means algorithm.

4.5 Clustering Performance

A comparison with UnRAP and Variable selection based on PCA clearly indicates that attribute-based k-means algorithm performs better than the other two in detecting “segment" attack profiles.

To evaluate the obfuscation methods described above, first we apply Noise Injection, User Shifting and Target Shifting approach to average and random attacks and named it “ObsfuscatedRandom" and “ObsfuscatedAverage". We also create attack models based on mixed attack and AOP(20%) strategies. Figure 3 depicts the results of obfuscated attacks. As the results show, the detection performance using k-means and PCA is better than UnRAP algorithm for both mixed and AOP attack. Whereas, in terms of specificity, both UnRAP and PCA algorithm perform significantly better than k-means algorithm for all attacks except “AOP" attack. The performance of PCA-based clustering is significantly lower than both k-means and UnRAP algorithms for AOP attack. Overall our results show that in terms of misclassified real users, k-means does not perform better than the other two algorithms.

Considering two clusters only, everything is forced into these clusters and can potentially result in clusters that are not cohesive. One of the reason is that k-means algorithm generally performs poorly in unlinearly separable case. But other clustering algorithms such as EM clustering can address this problem. Figure 4 shows the comparison of push attacks across filler size for both clustering algorithms. As the results show that for k-means algorithm the filler size didn’t affect the performance of each push attacks whereas

EM clustering algorithm didn't demonstrate the robustness.

We also envisioned that spreading user profiles into more than two clusters may reduce the number of misclassified real user profiles. Our experimental results (not shown here) confirmed that, the trade-off is that specificity decreases for most attack models except AOP 20% attack compared to $k = 2$.

Overall, our experimental results showed that the attribute-based k-means clustering approach can be a good detection technique regardless of attack strategies. The detection performance of "segment" and "AOP(20%)" attack is significantly better than the other approaches exist in the literature. It is also observed that by dividing the user profiles (based on detection attributes) into different clusters, the attack profiles are always in one or two clusters of small size. So we need only one or two smaller cluster to mark as anomalous and disregard while generating predictions. However, this approach suffers from some drawbacks: the detection of the wrong cluster of users can result in filtering out of genuine users, and thus predicting biased estimates. Further, real life attacker might employ strategies which ensure more deviation in their ratings, thus fooling the filtering process.

5. Conclusions

The issue of security and robustness in recommender systems is a major concern. In this paper, we investigated an unsupervised anomaly detection algorithm using k-means clustering for detecting shilling attacks. In particular, our experimental results showed that "segment" attack, which is designed to target a specific group of likely buyers, can be easily detected with high accuracy using k-means clustering. It is also proved that the unsupervised approaches may achieve reasonably good performance against the attack types discussed in the literature. It is not surprising, since the assumption is that these attacks are designed in such a way that the pattern of ratings vary substantially from the real users of the system. Since e-commerce companies are reluctant to disclose vulnerabilities that they have spotted in their own systems, it is hard to see whether the proposed approaches are useful against real-life attacks.

References

[1] M. O'Mahony, N. Hurley, N. Kushmerick, and G. Silvestre, "Collaborative recommendation: A robustness analysis," *ACM Transactions on Internet Technology*, vol. 4, no. 4, pp. 344–377, 2004. [Online]. Available: http://portal.acm.org/citation.cfm?id=1031114.1031116

[2] S. Lam and J. Reidl, "Shilling recommender systems for fun and profit," in *Proceedings of the 13th International WWW Conference*, New York, May 2004. [Online]. Available: http://www2004.org/proceedings/docs/1p393.pdf

[3] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams, "Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness," *ACM Transactions on Internet Technology (TOIT), Volume 7 , Issue 4 (October 2007)*, 2007.

[4] Z. S, C. A, F. J, and M. F, "Attack detection in time series for recommender systems." *In: KDD Š06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 809-814*, 2006.

[5] B. Mehta, "Unsupervised shilling detection for collaborative filtering," *AAAI, 1402-1407*, 2007.

[6] M. O. K. Bryan and P. Cunningham, "Unsupervised retrieval of attack profiles in collaborative recommender systems," in *Technical Report, University College Dublin*, 2008.

[7] N. Hurley, Z. Cheng, and M. Zhang, "Statistical attack detection," *Proceedings of the third ACM conference on Recommender systems*, 2009.

[8] Y.Cheng and G.Church, "Biclustering of expression data." in *Proc Int Conf Intell Syst Mol Bio, 8:93-103, 2000* , 2000.

[9] P.-A. Chirita, W. Nejdl, and C. Zamfir, "Preventing shilling attacks in online recommender systems," in *WIDM '05: Proceedings of the 7th annual ACM international workshop on Web information and data management*. New York, NY, USA: ACM Press, 2005, pp. 67–74.

[10] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik, "Detecting profile injection attacks in collaborative recommender systems," in *To appear in Proceedings of the IEEE Joint Conference on E-Commerce Technology and Enterprise Computing, E-Commerce and E-Services (CEC/EEE 2006)*, Palo Alto, CA, June 2006.

[11] B. Mobasher, R. Burke, C. Williams, and R. Bhaumik, "Analysis and detection of segment-focused attacks against collaborative recommendation," in *To appear in Lecture Notes in Computer Science: Proceedings of the 2005 WebKDD Workshop*. Springer, 2006.

Stable Clustering of Temporal Gene Expression Data

Gaolin Zheng[1], Guowang Mu[1], Chung-Hao Chen[1] and Xinyu Huang[1]
[1]Department of Math & Computer Science, North Carolina Central University, Durham, NC 27707

Abstract - *K-means and its many variants are widely used in bioinformatics. However, one of the drawbacks is that the clustering results are dependent of the initial choice of centroids, hence the experimental results are less reproducible. In addition, poor handling of non-spherical clusters is another weakness of the K-means algorithm. Thus, we investigate spectral clustering on temporal gene expression data in this paper. Experimental results show that when combined with appropriate mother wavelet, spectral clustering is able to improve both clustering accuracy and stability for temporal data, as compared with the K-means approach.*

Keywords: spectral clustering, wavelet, K-means, clustering stability

1 Introduction

Due to the large number of genes that are profiled in each experiment, clustering is a common task applied to temporal gene expression data in order to find genes that are functionally related and help to generate focused hypotheses. Raw features from temporal data can be analyzed directly using autocorrelation and cross-correlation functions [1]. However, time series data are often analyzed in frequency domain because of asymptotic independence of periodogram ordinates [2]. The frequency content of a time series can be obtained by either Fourier transform or wavelet transform. Wavelet transform is widely used because it considers temporal variability in spectral characteristics and assume that the frequency contents can change with time and location [3]. There are many mother wavelets that vary in properties such as orthogonality, symmetry, and existence of phi et al. It will be interesting to explore which mother wavelet is more suitable for temporal gene expression data. In this work, we will examine three commonly used wavelets (complex Morlet, symlet and Haar). We will also compare them with Fourier transform for comparative purposes.

Among the clustering algorithms, K-means and its many variants are widely used in bioinformatics [4-6]. One of the drawbacks is that the clustering results are dependent of the initial choice of centroids, hence the experimental results are less reproducible. Poor handling of non-spherical clusters is another weakness of the K-means algorithm. As a modern clustering technique, spectral clustering has shown promise to overcome some weaknesses of traditional K-means clustering [7]. Spectral clustering has a strong connection with graph theory [8, 9] and Laplacian Eigenmaps [10]. It is simple to implement and can be solved efficiently by standard linear algebra software. Different algorithms are available to perform spectral clustering [7, 11]. Luxburg et al. [9] concluded that normalized rather than unnormalized spectral clustering should be used whenever possible. Moreover, spectral clustering also works well for clustering of non-convex shapes [7]. Thus, the contribution of this paper is to investigate wavelet based spectral clustering on temporal gene expression data. Our experimental results show that when combined with appropriate mother wavelet, wavelet based spectral clustering is able to improve both clustering accuracy and stability, as compared with the K-means approach.

This paper is organized as follows: section 2 introduces the three mother wavelets used in this paper and explains the spectral clustering algorithm in detail, section 3 shows the experiment results and section 4 concludes this paper.

2 Wavelet Based Spectral Clustering Method

In this section, we first introduce continuous wavelet transform and three mother wavelets used here. Then we describe how we adapt spectral clustering for temporal data.

2.1 Continuous wavelet transform

The continuous wavelet transform (CWT) of a dataset $s(t)$ is given by [12]

$$C_{a,b} = \int_{-\infty}^{\infty} s(t) \frac{1}{\sqrt{a}} \Psi\left(\frac{\mathrm{t}-\mathrm{b}}{\mathrm{a}}\right) \mathrm{dt} \quad (1)$$

Where b is the displacement, a is scale, Ψ is the mother wavelet. CWT is essentially the sum over all time of the signal multiplied by scaled and shifted versions of the wavelet function Ψ. The results of the CWT are many wavelet coefficients C, which are the function of scale a and position b.

Continuous wavelet transform can be performed using various mother wavelets. For convenience purposes, we chose Complex Morlet, Symlet and Haar in our study.

2.2 Spectral Clustering

Given an N x N symmetric distance matrix $A = A_{ij}$ where A_{ij} is the distance measure between i^{th} and j^{th} time series. The pseudo code of spectral clustering is listed in Algorithm 1 adapted from [11]. In this work, we use a normalized graph Laplacian because it was found to produce more desirable clustering results [7]. The scaling parameter σ is calibrated to maximize the clustering performance. Although this algorithm resorts to regular K-means at the final step, it is working on the embedded Eigenspace and therefore it has the potential to get around some of the weaknesses K-means has.

3 Cluster Validation

The performance of a clustering algorithm can be evaluated using F-measure [13], adjusted rand index [14], Silhouette width [15], and the Dunn Index [16] etc. A detailed review on the cluster validation methods is given by Handl *et al.* [17]. For processing convenience, we choose F-measure in this study to evaluate our clustering results. The F-measure uses the idea of precision and recall from information retrieval. This measure requires some information of ground truth (e.g. known classes of data). Each class t is regarded as the set of N_t items desired for a query; each cluster C_k (generated by the algorithm) is regarded as the set of N_k item retrieved from a query; N_{tk} gives the number of elements of class t within cluster C_k. For each class t and cluster C_k, precision and recall are then defined as $P(t, C_k) = \frac{N_{tk}}{N_k}$ and $R(t, C_k) = \frac{N_{tk}}{N_t}$, and the corresponding value under the F-measure is given by

$$F(t, C_k) = \frac{(b^2+1)P(t, C_k)R(t, C_k)}{b^2 P(t, C_k) + R(t, C_k)} \quad (2)$$

We set $b = 1$ for equal weighting of $P(t, C_k)$ and $R(t, C_k)$. The overall F-measure value for a partition is computed as

$$F(C) = \sum_{t \in T} \frac{N_t}{N} \max_{C_k \in C} F(t, C_k) \quad (3)$$

Where T is the set of all classes and N is the total number of data points. The overall F-measure is limited to [0, 1] and a higher value indicates a better partition.

4 Results and Discussion

4.1 Synthetic Data

We first design synthetic data to evaluate clustering performance of K-means and spectral clustering algorithms in combination of different transformation schemes. For each clustering algorithm and transformation combination, ten independent

Algorithm 1 Main spectral clustering algorithm

procedure SPECCLUSTERING(A) ▷ A: $N \times N$ distance matrix
- Form affinity matrix W by defining $W_{ij} = \exp^{-A_{ij}^2/2\sigma^2}$ if $i \neq j$ and $W_{ii} = 0$
- $W \leftarrow A$
- set $W_{ii} = 0$ for $i = 1 \ldots N$
- Define D a diagonal matrix whose D_{ii} is the sum of A's row i
- Form the normalized graph Laplacian $L = D^{-1/2} W D^{-1/2}$
- Find eigenvectors $V_1, V_2, \ldots, V_k$ corresponding to the k largest eigenvalues
- Form new matrix Z with $V_1, V_2, \ldots, V_k$ as its columns
- Form matrix Y by normalizing Z so that each row of Y has unit length
- Treat each row of Y as a point in R^k
- Cluster into k clusters via regular K-means

end procedure

experiments are conducted and the corresponding F-measures are obtained. To create a more realistic simulation scenario, the parameters in the synthetic data set are drawn from uniform distribution with small variance instead of leaving the true parameters as fixed constants. We use the notation $X \sim U(a, b)$ to denote that the random variable X follows the uniform distribution in the interval (a, b).

Synthetic data set : Consider three clusters of a total of 18 time series according to the following specification:

For $i = 1, 2, \ldots, 6$, the first half of the time series following $Y_{it} = \cos(\omega_i t) + \sin(\omega_i t) + \varepsilon_{it}$, and the second half of the time series following $Y_{it} = \cos(\beta_i t) + \sin(\beta_i t) + \varepsilon_{it}$, where $\omega_i \sim U(0.97, 1.03)$ and $\beta_i \sim U(0.87, 0.90)$. The error sequences are generated by the normal distribution with standard deviation equal to 1.3. For $i = 7, 8, \ldots, 12$, the first half of the time series following $Y_{it} = \cos(\beta_i t) + \sin(\beta_i t) + \varepsilon_{it}$ and the second half of time series following $Y_{it} = \cos(\omega_i t) + \sin(\omega_i t) + \varepsilon_{it}$. For $i = 13, 14, \ldots, 18$, $Y_{it} = \varepsilon_{it}$, here the error sequence are generated by the normal distribution with standard deviation equal to 0.7. This is a case where the data are divided into two periodic groups and a third group with a flat spectrum. The frequency contents of the two periodic groups are different between the first half and the second half of the time series. Time series of length 50 and 100 are generated for the validation study. We conducted regular K-means and spectral clustering on the synthetic data sets in conjunction with different data transformation methods. The results are shown in Table 1.

Table 1: Simulation results. (Numbers shown are mean ± sd of F-measure of 10 independent experimental runs.)

N	Transformation	K-means	Spectral Clustering
50	FFT	0.81 ± 0.09	0.77 ± 0.00
	Complex Morlet	0.71 ± 0.03	0.72 ± 0.00
	Symlet	0.78 ± 0.10	0.94 ± 0.00
	Haar	0.78 ± 0.10	0.93 ± 0.00
100	FFT	0.72 ± 0.10	0.67 ± 0.00
	Complex Morlet	0.71 ± 0.05	0.75 ± 0.00
	Symlet	0.78 ± 0.13	0.94 ± 0.00
	Haar	0.81 ± 0.09	0.94 ± 0.00

As shown in Table 1, the standard deviation of F-measure in spectral clustering algorithm generated partitions are significantly lower than the corresponding K-means generated partitions. This implies an increase in the clustering stability brought by spectral clustering. When combined with appropriate mother wavelets (symlet and Haar), spectral clustering also improves the clustering accuracy.

4.2 Case Study

We use temporal gene expression data of fission yeast (*Schizosaccharomyces pombe*) from Oliva's lab [18] to evaluate different transformation methods and compare the performance of spectral clustering with regular K-means in terms of accuracy and stability. We compiled a validation data set from Chen et al's study [19]. This validation data set consists of 119 environmental stress responsive genes that fall into three categories: 27 heat shock genes, 62 oxidative stress genes and 30 cadmium stress genes. The F-measure is obtained by comparing the final partition against the validation data set. The results are shown in Table 2.

Table 2: Clustering fission yeast temporal gene expression data. (Numbers shown are mean ± sd of F-measures of 10 independent experimental runs.)

Transform	K-means	Spectral Clustering
FFT	0.52 ± 0.07	0.57 ± 0.01
Complex Morlet	0.63 ± 0.06	0.69 ± 0.02
Symlet	0.60 ± 0.13	0.64 ± 0.03
Haar	0.58 ± 0.09	0.63 ± 0.02

In general, better partitions are generated from wavelet transformed data. Complex Morlet based wavelet transform outperforms either Symlet or Haar (Table 2). More consistent partitioning results are observed in spectral clustering where the standard deviation of F-measure from multiple experimental runs is lower.

5 Conclusions

This work investigated spectral clustering of time series data together with several approaches to transform data from time domain to frequency domain. Our results showed spectral clustering consistently generate stable reproducible clustering results than K-means for simulation data sets. Higher accuracies are achieved when data is transformed with Symlet and Haar wavelets. According to our experimental results on fission temporal gene expression data, the spectral clustering combining with complex Morlet wavelet transform outperforms other methods.

Although Symlet and Haar work better than complex Morlet for simulation data set, complex

Morlet is more suitable for the real gene expression data. One possible explanation is that the gene expression data might not follow the sinusoid curve as used in simulation. So it is important to explore several transformation techniques for a specific biological data set which may vary from case to case.

Working in the embedded Eigenspace might be the reason of the improvements observed in spectral clustering. In addition, it partitions data sets based on the reduced dimension, which can improve clustering efficiency for large biological data sets.

6 Acknowledgements

This work was supported by the National Institutes of Health [5T36GM008789-08 to Alade Tokuta].

7 References

[1] Z. Bohte, *et al.*, "Clustering of time series.," *Proceedings of COMPSTAT80,* pp. 587–593, 1980.

[2] A. Savvides, *et al.*, "Clustering of biological time series by cepstral coefficients based distances," *Pattern Recognition,* vol. 41, pp. 2398-2412, 2008.

[3] G. R. J. Cooper and D. R. Cowan, "Comparing time series using wavelet-based semblance analysis," *Computers & Geosciences,* vol. 34, pp. 95-102, 2008.

[4] D. Huang and W. Pan, "Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data," *Bioinformatics,* vol. 22, pp. 1259-1268, May 15, 2006 2006.

[5] B. J. F. Keijser, *et al.*, "Analysis of Temporal Gene Expression during Bacillus subtilis Spore Germination and Outgrowth," *J. Bacteriol.,* vol. 189, pp. 3624-3634, May 1, 2007 2007.

[6] J. J. Loor, *et al.*, "Temporal gene expression profiling of liver from periparturient dairy cows reveals complex adaptive mechanisms in hepatic function," *Physiol. Genomics,* vol. 23, pp. 217-226, October 17, 2005 2005.

[7] J. Shi and J. Malik, "Normalized cuts and image segmentation.," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 22, pp. 888–905, 2000.

[8] F. R. K. Chung, "Spectral Graph Theory.," in *CBMS Regional Conference Series in Mathematics*, 1997.

[9] W. E. Donath and A. J. Hoffman, "Lower bounds for the partitioning of graphs.," *IBM J. Res. Dev. ,* vol. 17, pp. 420-425, 1973.

[10] M. Belkin and P. Niyogi, " Laplacian eigenmaps for dimensionality reduction and data representation.," *Neural Comput.,* vol. 15, pp. 1373-1396, 2003.

[11] A. Y. Ng, *et al.*, "On spectral clustering: Analysis and an algorithm.," in *Advances in Neural Information Processing Systems (NIPS)*, T. Dietterich, *et al.*, Eds., ed, 2002.

[12] S. Mallat, *A Wavelet Tour of Signal Processing*. New York: Academic Press, 1998.

[13] C. van Rijsbergen, *Information Retrieval (Second Edition)*: Butterworths, 1979.

[14] W. Rand, "Objective criteria for the evaluation of clustering methods.," *Journal of the American Statistical Association,* vol. 66, pp. 846-850, 1971.

[15] P. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis.," *Journal of Computational and Applied Mathematics,* vol. 20, pp. 53-65, 1987.

[16] C. Dunn, "Well separated clusters and fuzzy partitions.," *Journal on Cybernetics,* vol. 4, pp. 95-104, 1974.

[17] J. Handl, *et al.*, "Computational cluster validation in post-genomic data analysis," *Bioinformatics,* vol. 21, pp. 3201-3212, August 1, 2005 2005.

[18] A. Oliva, *et al.*, "The Cell Cycle Regulated Genes of Schizosaccharomyces pombe," *PLoS Biology,* vol. 3, p. e225, July 1, 2005 2005.

[19] D. Chen, *et al.*, "Global Transcriptional Responses of Fission Yeast to Environmental Stress," *Mol. Biol. Cell,* vol. 14, pp. 214-229, January 1, 2003 2003.

Heuristic Approaches for Embedded Processor System Size Reduction

R. Dixit[1], **and H. Singh**[1]
[1]Department of Engineering and Computer Engineering, Wayne State University, Detroit, MI, USA

Abstract *— Today's sensors are capable of generating large amounts of data. Much of the data is often discarded, or simply not processed, because there is no time or processing capability in the embedded or on-board processors. The developers are struggling with algorithms that quickly sort target from non-target, and, with signal extraction techniques to rapidly classify the objects embedded in the scene. Large matrix system models, as is the case when there is data from multiple sensors, also pose significant computational challenges. The problem is relevant and significant when one considers sensor applications where large streams of data are available and an accurate decision is crucial. Conventional data mining techniques, typically attempt to solve full order factorial solutions for Imaging, for Feature Extraction, and for Feature Delineation. This paper presents novel heuristic techniques that can be used for pre-processing, and thus reduces the order of the problem. And, these techniques can be implemented as Neuro-fuzzy decision logic, to automate the incorporation of expert knowledge. It is shown that, with such a-priori knowledge of the computational problem, the information from the various sensors streams can actually help speed-up algorithm convergence.*

Keywords : Heuristic, target classification, Cluster & Factor analysis.

1 Introduction

This is a classical problem akin to 'water-water everywhere, but not a drop to drink'. The on-board processor's refrain is 'data data everywhere, but no time to analyze'. There is simply too much data streaming from sensors, and the compute algorithms embedded in such sensors, either discards the data because the pulse duty-cycle does not afford real-time computation of the parameters, or uses some type of confidence metric to classify the target. This latter approach may result in unacceptable false-positives and false negatives. This is especially worrisome in circumstances where the signal is buried in noise. And, can lead to errors when considering performance improvement investment strategies to determine which Critical Technology Elements may have the most benefit to system performance.

This paper studies an image analysis problem, where the problem is to detect and classify the target of interest. Many classical techniques are available for analysis [1], including Factor Analysis (FA) and Principal Component Analysis (PCA). The analysis and modeling of streams of data, especially incomplete data, also poses special challenges [2]. Conventional algorithms [3] would use full matrix techniques, to analyze and classify the object of interest. What we want to see is, if, exploiting some physical law governing area pattern distribution, we can find data belonging to the same sub-space, and thereby reduce the problem dimensionality. The results we present show that use of such heuristic transformations quickly capture the discriminative capacity of these variables.

Other techniques [4], like combining data gathered from different sources (data fusion), have been used. Multi-sensor data fusion seeks to combine information from multiple sensors and sources to achieve inferences that are not feasible from a single sensor or source. Here, the fusion of information enhances the confidence in the decision. Typically, such algorithms [5] sequentially use Signal-level fusion, then, pixel-level fusion, then, feature-level fusion, and finally, decision-level fusion to give an output. Classification is one of the key tasks in data analysis. The classification accuracy is improved when multiple source image data are introduced to the processing algorithm. This paper deals with a static image, but, techniques presented herein can also be used in conjunction with change detection algorithms which identify differences in the state of an object or phenomenon by observing it at different times. For example, sensor image data with low temporal resolution and high spatial resolution can be fused with high temporal resolution data to enhance the changing information of certain objects.

Each fusion method has its own set of advantages and limitations. The selection and arrangement of which fusion scheme is typically arbitrary, and often depends upon user's a-prior knowledge of what works.

2 Key terminology and Concepts

Data clustering [6] is often used in data intensive applications. Factor analysis with cluster analysis means to approach a data set from two complementary perspectives. The underlying logic of both procedures is classification. Classification in either approach is based on homogeneity. Homogeneity with respect to cluster analysis means that data are classified into clusters with respect to their similarity on variables. Clusters are ideally characterized by in-cluster homogeneity of objects and between heterogeneity of objects. Factor analysis, in contrast,

concentrates on the homogeneity of variables resulting from the similarity of values assigned to variables by respondents. While factor analysis can be used to uncover the dimensional structure within the data, factor analysis, however, cannot give information about groups. A priori knowledge can aid in this resolution because the expert knows that 'what to look for'.

And, clustering algorithms generally follows hierarchical or partition approaches. For the partition approach the k-means [7], and its variants, such as the fuzzy c-means algorithm [8], are the most popular algorithms. Partition clustering algorithms require a large number of computations of distance or similarity measures among data records and clusters centers, which can be very time consuming for very large data bases. Moreover, partition clustering algorithms generally require the number of clusters as an input parameter. However, the number of clusters usually is not known a priori, so that the algorithm must be executed many times, each for a different number of clusters and uses a validation index to define the optimal number of clusters. The determination of the clusters' numbers and centers present on the data is generally referred to as cluster analysis. The main idea, present in most of the validity indexes, is based on the geometric structure of the partition, so that samples within the same cluster should be compact and different clusters should be separate. A visual representation of the distance at which clusters are combined, is displayed using a dendrogram. The dendrogram is read from left to right. Vertical lines show joined clusters. The position of the line on the scale indicates the distance at which clusters are joined. The observed distances are often rescaled, so you don't see the actual distances; however, the ratio of the rescaled distances within the dendrogram is the same as the ratio of the original distances.

In general, we propose that decisions in sensor management can benefit from the involvement of human operators, who are expected to ''discuss'' and ''negotiate'' with the software for working out intelligent solutions. Usually human operators tend to express their opinions in the form of propositions formulated in natural language and they will also want to receive information as linguistically understandable descriptions. Moreover, tolerance of imprecision seems a necessary requirement when making high-level decisions for achieving tractability, robustness, low computational cost, and better rapport with the inexact nature of human thinking.

3 Representative Data

To develop and illustrate the heuristic search algorithms, we'll take the case of an image, and to extract a desired target from this image. Many such reference images are available, along with corresponding data. [9]. The figures and data given in reference [9] are reproduced here as Figure **1** and Table **1** for ready reference. The image is published for context only, so the reader is familiar with the cluttered scene problem, and the issues related to target classification. The data is not specific what we're trying to accomplish…that is, to be able to take such large data sets, and find ways to improve analysis time and target recognition confidence. A number of trials were undertaken, and at the end, target type was identified. The Data tracks how long it took to identify the target of interest. In [9], the track data is also presented, showing 43 files, and the time taken to converge to an identified (classified) target. Typical stream of data from such systems is illustrated from imaging data, where they tracked time to detect a target in various camouflage environments. The terms used are Target number (or Type, for Target classification), Distance to target (in Meters), Aspect Ratio of target (in Sin Radians), Number of vertical pixels, Area covered by these pixels, Luminosity of the Dark area (to determine contrast, and since this was an image processing example, it's the white balance), Surrounding luminosity, Edge points. The approaches we're trying to illustrate should converge on Target Type faster. To do this, many of the data files were used as 'training' data, and a few reserved for Testing. Of course not all data used in Target Type determination is provided, so, this is really an exercise in finding the relationships between data columns that would allow reduction in data size, without loss of confidence.

Figure 1. Representative data [9], that shows an object of interest in a visual scene,

The image [9] used only for illustration purposes, shows a vehicle in the middle of the scene. There are nearby trees

and the vehicle appears to be in an open portion of, what appears like, a valley. **Table 1** below, shows the track data.

TARGET NO	distance	aspect	vert	area	target lum	Dark area lum	Surround lum	Edgepts	SEARCH TIME
type	m	ass(sin)	pixels	(pixels)	scene	dark	grass	Pts	search time(s)
1	4007	0.707	10	141	14	17	29	9571	14.6
1	2998	0.819	11	225	21	10	27	8927	15.2
2	3974	0.707	13	173	20	24	28	9138	12.4
3	5377	0.052	5	49	18	23	30	8970	29.8
2	1013	0.515	50	2708	19	5	34	8706	2.8
4	3052	0	11	100	12	18	30	8755	6.4
5	5188	0.407	9	76	18	23	28	9053	26.7
6	3679	0.122	10	96	12	20	26	8620	10
2	860	0.995	54	3425	9	1.5	40	8961	2.7
4	1951	0.848	16	332	15	11	27	8572	2.8
3	3992	0.788	11	154	20	19	26	9194	11.9
6	1041	0.743	24	1645	11	4	35	9074	2.5
7	2145	0.978	17	553	8	5	18	8280	3.7
3	1998	0.755	19	659	20	10	22	8739	8.1
2	4410	0	11	101	22	18	29	9404	12.4
1	2893	0.423	16	320	12	7	23	8670	2.5
5	1933	0.978	13	368	15	12	23	8606	4.8
1	1850	0.961	28	876	3	4	9	8464	2.8
8	1045	0.087	26	985	19	10	12	8613	12.3
2	1933	0.946	22	867	16	11	27	8376	2.8
7	4206	0	9	79	26	29	38	9506	15.1
1	5722	0.883	7	73	38	40	46	9044	25.6
4	4920	0.423	8	61	20	21	36	8618	12.1
6	4206	0.809	9	142	18	12	21	9152	8
5	2348	0.94	9	198	18	21	30	8504	5.5
1	3992	0.875	11	217	15	14	26	9078	7.8
9	4410	0.956	11	247	16	8	19	9397	9.6
8	2321	0.829	15	458	22	21	47	8365	5.1
5	3661	0.755	9	84	17	25	23	8807	7.5
3	3670	0	13	192	14	15	27	8483	6.1
7	1671	1	19	693	15	13	31	8959	3.5
4	4345	0.809	8	63	15	12	20	9021	12.3
2	3662	0.574	10	203	26	25	44	8702	5.4
5	633	0.707	50	4403	20	5	39	8741	2.5
3	492	0.07	57	3045	20	16	23	8992	2.2
4	1497	0.777	16	560	10	7	20	9014	5.8
5	1041	0.999	33	1613	17	5	32	8486	2.6
1	2891	0.985	19	486	12	12	35	9021	12.1
7	5147	0.934	5	81	18	27	34	9075	34.9
6	1648	0.588	18	648	23	7	37	9070	2.7
8	948	0.731	35	1463	18	5	38	8790	3.7
7	3662	0.407	12	188	19	25	39	8524	5.8
6	2900	0	17	340	20	10	49	8791	4.1
2	5136	0	10	79	25	16	27	8941	10.6

Here, Column 1 is the desired output…ie what is the target classification. And, Column 10 is the penalty we incur as the algorithm converges on the target classification. Columns 2 – 9 are the data that can be used to draw inference about the target classification, and using our heuristic approach, hopefully reduce target search time. The metric we'll use is, if the adjacent column data can be more effectively used for target classification.

4 Data Analysis and Algorithm

. To begin, we start with some check for Output vector(s) correlation Vector orthogonality and looking at Table 1, we find that the target type classification and search time penalty are largely independent variables ie linear correlation value of 0.2. Figure 2 shows the Target type vs. Search time, again showing that there is clustering, but little correlation.

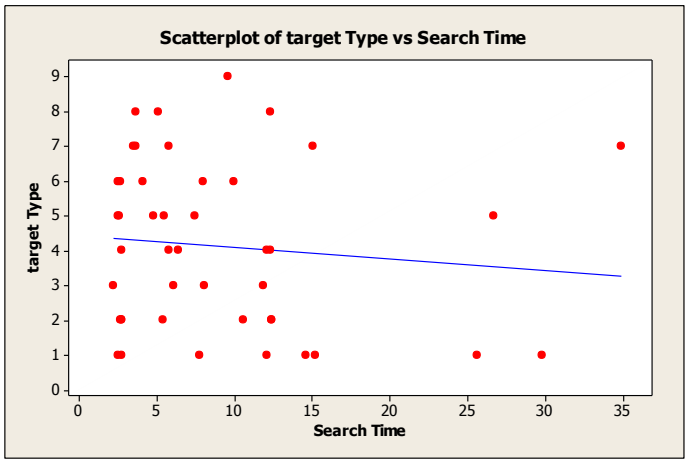


Figure 2. Scatter plot of Target type with respect to Search Time.

Covariance matrices [10] are commonly used for most multivariate analysis—be it regression analysis, principal component analysis, discriminant analysis, or canonical correlation analysis. Thus, we look at the Orthoganility of Input Columns with Output vectors. This is printed in Table 2. And clearly, while there is some orthoganility, the input data is only loosely coupled with the target type. This is intuitive…meaning that any given target type may be in any scene. And, we see the as-expected stronger correlation with search time, depending on area and vert information, Figure 3. And further presence of dark areas and surround luminosity also did not strongly affect search time, but together, high correlation with search time, Figure 4. This also points to system reduction opportunity. We next look at Input vector cross Correlation, and see if there is an opportunity for vector clustering. We had a glimpse of that

in the correlation between area and vertical vectors. This data is printed in Table 2 [9].

TABLE 2 Correlation between the variables of the original search time data

	distance	*aspect*	*vertical*	*area*	*target lumn*	*Dark area*	*Surround*	*Edge points*
Distance	1	-0.24	-0.79	-0.72	0.37	0.72	0.06	0.41
Aspect	-0.24	1	0.08	0.12	-0.24	-0.25	-0.08	-0.10
Vertical	-0.79	0.08	1	0.95	-0.19	-0.58	0.07	-0.19
Area	-0.72	0.12	0.95	1	-0.14	-0.53	0.15	-0.13
target lumn	0.37	-0.24	-0.19	-0.14	1	0.62	0.52	0.25
Dark area	0.72	-0.25	-0.58	-0.53	0.62	1	0.32	0.22
Surround	0.06	-0.08	0.07	0.15	0.52	0.32	1	0.04
Edge points	0.41	-0.10	-0.19	-0.13	0.25	0.22	0.04	1

From Table 2, the highest correlation between variables is 0.95 from the variable, *area*, and the variable, *vertical*. Additionally, Distance is well correlated with area and with dark area. See Dendogram in Figure 5. This too is intuitive, ie when we have large targets, nearby, or against a dark background, the target is easier (faster) to identify. This is especially true , when Target Luminosity is used with Dark Area information. ie a brighter target stands-out against a dark area background. Such heuristic conclusions can be made with a-priori knowledge of the problem and data. And, in a Supervised Training approach, for Neuro-fuzzy techniques, can be used to automatically train the system for faster and higher confidence classifications.

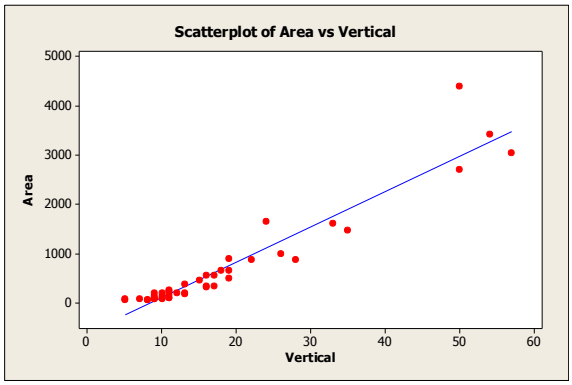


Figure 3. Area versus Vertical correlation.

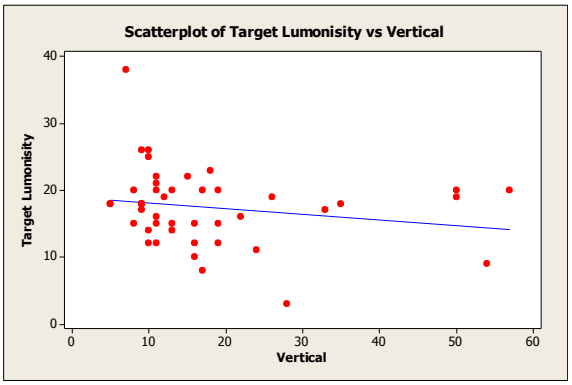


Figure 4. Target Luminosity versus Vertical.

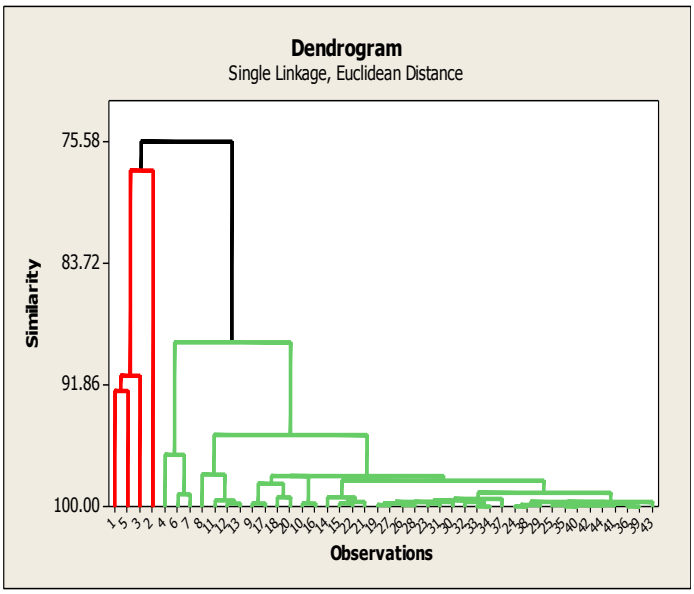


Figure 5. Euclidian distance dendrogram of the variables.

Using the MiniTab™ analysis program we can readily generate additional statistical data about Factor (or Vectors chosen to go forward.). For sake of simplicity, and based on data, we can reduce the input variable space by 1-vector immediately, keeping either area or vert.

This is presented in Table 3. Along with an explanation of the terms as used by MiniTab™. This shows that, if we reduce the matrix size by 1-vector (vert of area), we still retain a 84% confidence in the target type classification data. And, the Best case full matrix data had 87% target type classification confidence. Further reductions degrade the target type confidence further, and the user can select the metric that is applicable for that application. ie. If one has a >70% confidence (CORR term in the matrix below) that the target is (for example) a tank, then, taking precautionary measures have high payback. Whereas for 22% confidence, as indicated by 5 factor analysis, it is likely that too much data has been removed to provide meaningful target type classification in a reasonable time.

TABLE 3 Statistical analysis from implementation [9]

	CORR	*TRMS*	*STD*	*MAD*	*EWI*	*ERR*
5 factor	0.22	21.11	32.05	19.70	73.63	58.87
6 factor	0.71	6.92	10.46	6.46	24.13	18.03
7 factor	0.84	3.31	4.83	3.09	11.39	8.05
Original	0.87	2.84	4.40	2.65	10.02	7.38

CORR: Correlation between the original output and the estimated output.

TRMS: Total Root Mean Square for the distance between the original output and the estimated output using the same testing data through the fuzzy neural system.

$$TRMS = \frac{\sum_{i=1}^{n}\sqrt{(x_i - y_i)^2}}{n-1}$$

where x_i is the estimated value and y_i is the original output value.

STD: Standard Deviation for the distances between the original output and the estimated output using the same testing data through the fuzzy neural system.

MAD: Mean of the absolute distances between the original output and the estimated output using the same testing data through the fuzzy neural system

EWI: The index value from the summation of the values with multiplying the statistical estimation value by its equally weighted potential value for each field .

ERR: The error rate is

$$ERR = \sum_{i=1}^{n}\left(\frac{E(i) - O(i)}{E(i)} \times 100\right)$$

where n is the number of testing data, $E(i)$ is the estimated output, and $O(i)$ is the actual output.

The proposed algorithm then is, to implement the appropriate analysis techniques, and to reduce the problem complexity based on expert use of a-priori knowledge based heuristic approaches.

Assume there are n by p matrix, where n= the number of trials available. And p represents the vectors. The output vectors are included as a separate matrix, and depends on the problem size. In the simple case of Target Type Classification, this is 1-output. The Search Time vector is the penalty vector, but dependent on the 'clutter' on the scene.

1. Calculate the correlation matrix, R, between output and penalty variables. If only one output, this is simply a cross product. Usually this data is orthogonal, meaning that the outputs and other information are not correlated to each other.
2. Next, perform a cross correlation between input vectors and output vectors. This correlation between variables can decide the validation of reducing procedure.
3. Use expert knowledge to judge validity of correlation values, especially if input vectors also indicate cross correlation with each other. This is equivalent to calculating the eigenvalues, λ, from the correlation matrix of the original data.
4. Select the number of reduced factors, m, for m < p, using step 1. If appropriate, and as suggested by [8] the eigenvalue-greater-than-one rule. Sub-matrix analysis is useful in determining further variable dependencies.
5. Calculate the initial factor loadings, F, by multiplying the square root of the eigenvalues from step 3 and corresponding eigenvetors. This step can

be shortened by using the MiniTab™ tool as it has automated much of the MultiVariate Factor analysis process.

6. Compare the factor scores, particularly the CORR value.

5 Conclusion

We have provided a non-technical description of exploratory factor analysis and an example illustrating its usage. This heuristic approach utilizes expert knowledge to reduce system complexity. This can be automated, and the ANFIS tool in MatLab™ has extensive capabilities to automate the capture and use of expert knowledge in their Neuro-fuzzy algorithms. These employ the Sugeno techniques, particularly suited for fast computation and smaller neural nets. The work in this paper showed that we can take a stream of multi-variate data, and using conventional analysis techniques, determine a suitable sub-set of the data that allows good confidence in the output, and where cluster techniques can further aid is sub-matrix data analysis. The data analysis is aided by expert knowledge of the data to be analyzed, and in determining the linkages between the data. In this examination, the input variable stream can be reduced from the 8-vectors, easily to 7, and then with modest additional penalty, to 5 vectors. This represents a 37.5% reduction in system size, and reduces from upto 40k calculations to approx 120 computations. This is a tremendous reduction in computational complexity. The approach can be generalized, but is most suited to applications where there is an expected data and expected correlations between data streams. There is tremendous scope for such work, as more embedded systems get deployed and as the data streams from sensors can be beneficially fused to improve target tying confidence and reducing algorithm convergence time.

References

[1]. H. Shriam, O. Nasri, P. Dague (University of Paris), and O. Heron, M. Cartron, "Smart Distance Keeping: Modeling and Perspectives for Embedded Diagnosis". (CEA LIST, Saclay). Published in 1st International Conference on Intelligent Systems, Modelling and Simulation ISMS'10

[2]. Charles Reinholtz , "DARPA Grand Challenge 2010. Virginia Tech Team". Prof. Charles Reinholtz, Team Leader. Virginia Tech University, Blacksburg, VA 24060

[3]. D. Hall, S. Pesnel, R. Emonet, J. Crowley et al. "Comparison of Target Detection Algorithms using adaptive Background Models". INRIA Rhone-Alpes. Submitted VS-PETS, Oct 2005.

[4]. M. Cunningham, F. Dowla "A Comparison of Digital Signal Extraction Techniques". Jan 2005. Doc. # UCRL-TR-208848 at the Lawrence Livermore National Laboratory.

[5]. J. D. Aplevich. "A Suboptimal Linear System Reduction Algorithm". Page 1375 Proceedings of the IEEE, Proceedings Letters Sep 1973.

[6]. R. King "System Reduction and Solution Algorithms for Singular Linear Differnce Systems under Rational Expectations". Boston University, Dep't of Economics, and M. Watson Princeton University. Published in Computational Economics 20: 57 – 86, 2002. Kluwer Academic Publishers. Netherlands.

[7]. Jiang Dong, Dafang Zhuang, Yaohuan Huang and Jingying Fu, "Advances in Multi-Sensor Data Fusion: Algorithms and Applications". Sensors Open Access Journal Sep. 2009, 9, pp. 7771-7784

[8]] Norman Cliff, "The Eigenvalues-Greater-Than-One Rule and the Reliability of Components,"
Psychological Bulletin, Vol. 103, No. 2, pp. 276-279, 1988.

[9]. D. Nam, H. Singh and T. Meitzler, "Extracting Interstingness Dimensions for Search Time in Visually Cluttered Scenes". 21st. International Conference on Computers and Their Applications, CATA-2006, Seattle WA. Mar 23 – 25, 2006. Published 2006, pp 372 – 377

[10]. Frank Vahid. "Digital Design, with RTL Design, VHDL, and Verilog". 2nd Edition. Wiley 2007.

Probabilistic Vector Machine

Henri Luchian[1] **and Andrei Sucilă**[2]
[1]Faculty of Computer Science, University Alexandru Ioan Cuza, Iasi, Romania
[2]Faculty of Computer Science, University Alexandru Ioan Cuza, Iasi, Romania

Abstract— *Many of the classification algorithms used in practice today are based on extensions of the binary SVM classifier, which has been very successful, especially in the field of bioinformatics. SVMs run on information derived from the convex hulls of the initial training data, ignoring the data points inside the convex hull.*

This paper presents a new way of deriving the separating hyperplane used in linear classification which takes into account the distribution of the data and proves that this directly minimizes the probability of erroneous classification when the data satisfies a certain loose condition.

Keywords: classification, support vector machine, margin distribution, probability

1. Introduction

Ever since the introduction of the Support Vector Machines (SVMs) by [1], [2], the algorithm has been very successful in various fields, not least of which would be the field of bioinformatics. It runs by selecting a hyperplane which is used to derive the classifying rule. The hyperplane separates the space into two semispaces, classifying the data in one semispace as positively labeled and the data in the other as negatively labeled. If one denotes by w the normal vector to the hyperplane and by b its bias, then the classification rule is:

$$label(x) = \begin{cases} 1, < w, x > +b \geq 0 \\ -1, < w, x > +b < 0 \end{cases} \quad (1)$$

where $< \cdot, \cdot >$ denotes the scalar product. This is the general way in which a hyperplane induces a classification rule.

SVM focuses on the way the hyperplane is chosen. It is such that it maximizes the minimum distance of any point to the hyperplane.

Other authors [3], [4] have observed that this method of choosing the hyperplane only takes into account the convex hull of the data points and ignores the distribution of the remaining data. In [4] an algorithm is derived which optimizes the margin distribution, aiming to improve the generalisation capability of the algorithm. They have shown, empirically, that the margin distribution of the data behaves differently than the margin and that it is both better correlated with the accuracy and is more stable in terms of measurements over the training data and expected values.

In this paper we will aim to improve on the ideas from [4], in the sense that the distribution of the entire data set is to be taken into account for choosing the hyperplane. Specifically, we show that the probability of error is directly correlated with the average distances to the separating hyperplane, each distance being measured in units of standard deviation. Furthermore, we show how this can be modelled and properly solved with the help of convex sets.

2. Preliminaries

We are concerned with the binary classification problem of finding an $f : \mathbb{R}^n \rightarrow \{-1, 1\}$ which assigns positive or negative labels to points from an n dimensional space. We shall denote by $S = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\} \subset \mathbb{R}^n \times \{-1, 1\}$ a sample set of m examples. Denote $S_p = \{x \in \mathbb{R}^n | (x, y) \in S, y > 0\}$ and by $S_n = \{x \in \mathbb{R}^n | (x, y) \in S, y < 0\}$ the positively labeled and negatively labeled subsets. Through notation abuse, we shall consider that $S = S_p \cup S_n$.

For a hypothesis $w \in \mathbb{R}^n$ and a bias $b \in \mathbb{R}$, the induced classification rule is $f_{w,b}(x) = sign(< w, x > +b)$. Also, denote by $d(x)$ the distance of a point x to the hyperplane, $d(x) = \frac{<w,x>+b}{||w||}$. Note that $f_{w,b}(x) = sign(d(x))$.

Given a distribution of the points x, this induces a distribution on the distances, $d(x)$. It is this distribution to which we shall refer to as the *margin distribution*. Denote by $D_p = \{d(x) | x \in S_p\}$ and $D_n = \{-d(x) | x \in S_n\}$.

A point is erroneously classified iff $f(x) \neq y$, where y is the label of x and thus, the probability of error is given by $P(f(x) \neq y)$.

3. Probabilistic Vector Machines

Due to the fact that, for a sample set, S, $S_p \cap S_n = \emptyset$ and $S_p \cup S_n = S$, the empirical expected rate of error is:

$$\begin{aligned} &P(f(x) \neq y) = \\ &= P((x \in S_p \cap d(x) < 0) \cup (x \in S_n \cap d(x) > 0)) \\ &= P(x \in S_p \cap d(x) < 0) + P(x \in S_n \cap d(x) > 0) \end{aligned} \quad (2)$$

Note that the two terms on the right hand in the above equation correspond to the probability to obtain a false negative and a false positive respectively. In this paper we shall concern ourselves with the minimization of

$$max\{P(d(x) < 0 | x \in S_p), P(d(x) > 0 | x \in S_n)\} \quad (3)$$

This differs from the absolute probability of error, but do note that a classifier that yields significantly different specificity and sensitivity is of reduced practical importance.

Observe that $P(x \in S_p \cap d(x) < 0)$ decreases as D_p is shifted to the right. Similarly, $P(x \in S_n \cap d(x) > 0)$ decreases as D_n is shifted to the right. Denote by E_p and E_n the expected values for D_p and D_n respectively and by σ_p and σ_n the standard deviations of D_p and D_n.

Observe that, if D_p and D_n are sampled each from a normal distribution, then $P(d(x) < 0 | x \in S_p)$ is directly determined by $\frac{\sigma_p}{E_p}$. For example, if $\frac{\sigma_p}{E_p} = \frac{1}{2}$, then $P(d(x) < 0 | x \in S_p) \cong 0.05$, as the hyperplane is at a distance of $2\sigma_p$ from the average.

Definition Two samples of real numbers, D_0 and D_1, with means E_0, E_1 and standard deviations σ_0, σ_1, are called similarly negatively distributed iff:

$$P(x \leq E_0 - \lambda\sigma_0) = P(x \leq E_1 - \lambda\sigma_1), \forall \lambda \in \mathbb{R}_+ \quad (4)$$

This property holds for sample sets taken from the same type of distribution. For example, in our problem, if D_p is taken from $N(E_p, \sigma_p)$ and D_n is taken from $N(E_n, \sigma_n)$, then this property holds. It may hold for samples from different types of distributions, but this it is no longer trivial.

We shall say that property (4) holds for a tuple $\{S_p, S_n, (w, b)\}$ if the property holds for the induced margin distributions. Note that the bias is irrelevant in this, as a change in bias would only lead to a shift in both of the induced margin distributions.

If this property takes place, then we can replace the proposed cost function, $max\{P(d(x) < 0 | x \in S_p), P(d(x) > 0 | x \in S_n)\}$ with

$$max\{\frac{\sigma_p}{E_p}, \frac{\sigma_n}{E_n}\} \quad (5)$$

3.1 Obtaining the Separating Hyperplane

In order to find the hyperplane that minimizes the cost function (5), we will use a replacement for the standard deviation that can be computed linearly, in order to construct a system of convex inequalities which would be readily solvable. As such, let us consider :

$$\begin{array}{l} \sigma_p = \frac{1}{|S_p|-1} \sum_{x \in S_p} | < w, x > + b - E_p | \\ \sigma_n = \frac{1}{|S_n|-1} \sum_{x \in S_p} | < w, x > + b + E_n | \end{array} \quad (6)$$

Obviously, σ_p and σ_n do not corespond to the standard deviations, but express the same idea. As a consequence of the inequalities between the arithmetic mean and the square mean, σ_p and σ_n as defined above are always smaller than the true standard deviations.

Let's consider the following optimization problems:

$$\left[\begin{array}{l} minmax\{\frac{\sigma_p}{E_p}, \frac{\sigma_n}{E_n}\} \\ \\ \frac{1}{|S_p|} \sum_{x_i \in S_p} d(x_i) = E_p \\ -\frac{1}{|S_n|} \sum_{x_i \in S_n} d(x_i) = E_n \\ E_p > 0, E_n > 0 \\ \\ \sigma_p = \frac{1}{|S_p|-1} \sum_{x_i \in S_p} |d(x) - E_p| \\ \sigma_n = \frac{1}{|S_n|-1} \sum_{x_i \in S_n} |d(x) + E_n| \end{array} \right. \quad (7)$$

$$\left[\begin{array}{l} minmax\{\frac{\overline{\sigma_p}}{E_p}, \frac{\overline{\sigma_n}}{E_n}\} \\ \\ \frac{1}{|S_p|} \sum_{x_i \in S_p} [< w, x_i > + b] = \overline{E_p} \\ -\frac{1}{|S_n|} \sum_{x_i \in S_n} [< w, x_i > + b] = \overline{E_n} \\ \overline{E_p} \geq 1 \\ \overline{E_n} \geq 1 \\ \\ | < w, x_i > + b - \overline{E_p}| \leq \sigma_p^i, \text{ for } x_i \in S_p \\ | < w, x_i > + b + \overline{E_n}| \leq \sigma_n^i, \text{ for } x_i \in S_n \\ \\ \frac{1}{|S_p|-1} \sum \sigma_p^i = \overline{\sigma_p} \\ \frac{1}{|S_n|-1} \sum \sigma_n^i = \overline{\sigma_n} \end{array} \right. \quad (8)$$

where (7) models the problem we are trying to solve.

From the above equations, it is clear that $\overline{E_p} = ||w|| \cdot E_p$, $\overline{E_n} = ||w|| \cdot E_n$ and that $\overline{\sigma_p} \geq ||w|| \cdot \sigma_p$, $\overline{\sigma_n} \geq ||w|| \cdot \sigma_n$. As such,

$$\frac{\overline{\sigma_p}}{\overline{E_p}} \geq \frac{\sigma_p}{E_p}; \frac{\overline{\sigma_n}}{\overline{E_n}} \geq \frac{\sigma_n}{E_n} \quad (9)$$

The following holds:

Lemma System (7) is feasible iff system (8) is feasible. Furthermore, if they're both feasible, then (w, b) is an optimal solution to (7) iff $\exists \lambda \in \mathbb{R}_+$ such that $(\lambda \cdot w, \lambda \cdot b)$ is an optimal solution for (8).

In short, the two systems are equivalent. So, we can solve system (8) and obtain the sought after hyperplane. We shall from now on ommit the overline.

Note that the objective function of (8) is not linear, nor convex. But, due to the fact that $obj_1 = \frac{\sigma_p}{E_p}$ and $obj_2 = \frac{\sigma_n}{E_n}$ are quasi-convex, in the sense defined in [5], which is to say that every sublevel set of obj_1 and obj_2 is convex, and $max\{\cdot, \cdot\}$ preserves quasi-convexity, the objective function is quasi-convex. This means that we may optimaly and efficiently solve this system with the help of a set of feasability linear programs as follows.

Let's define, for $t \in \mathbb{R}_+$, the feasability linear program composed of the same set of equations as (8), to which we add:

$$\begin{array}{l} \sigma_p \leq t \cdot E_p \\ \sigma_n \leq t \cdot E_n \end{array} \quad (10)$$

Note that, for a fixed t, this is a linear feasibility program, which we'll denote by $LP(t)$ and may be solved efficiently.

We may now solve (8) as a binary search over the values of t.

```
Init t_left = 0, t_right = 1;

while (!Feasable(LP(t_right)))
{t_left = t_right; t_right *= 2;};

while (t_right - t_left > EPSILON)
{
t_center = (t_right + t_left) / 2;

if (Feasable(LP(t_center)))
t_right = t_center;
else
t_left = t_center;
};

t = t_right;
```

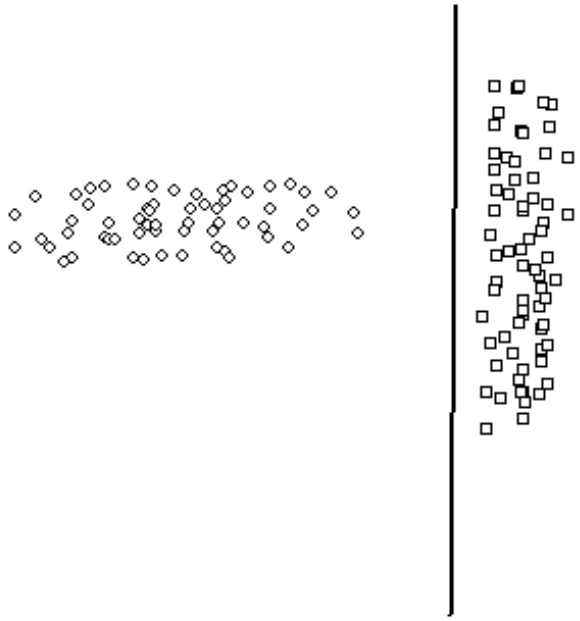
The following holds:

Theorem If $\exists w \in \mathbb{R}^n$ such that $E_p \neq -E_n$ then $\exists t \in \mathbb{R}_+$ such that $LP(t)$ is feasible and thus, (8) is feasible.

The condition in the previous theorem is easily satisfied. Should it not be satisfied, S_p and S_n would induce D_p and D_n with the same means, from every angle (every w), which would almost always require $S_p = S_n$, implying that the problem is ill-posed.

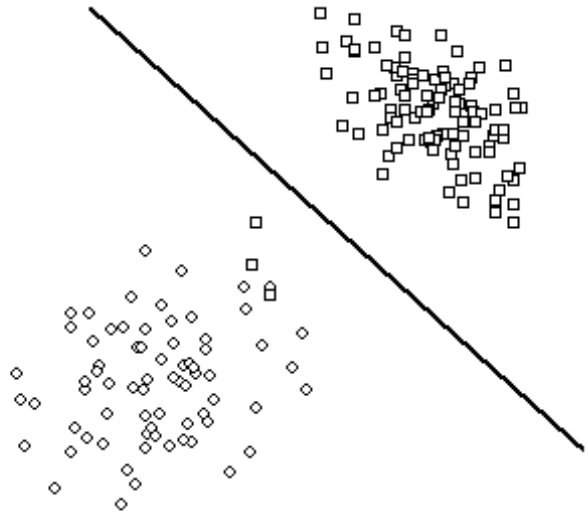
4. Geometric Interpretation and Results

In essence, the described way of choosing the separating hyperplane is based on the idea that the points of one type of label might vary more towards the dividing line than the points associated of the other label. As a consequence, the choice of hyperplane should compensate for this, allowing for equal variability. One such instance is described bellow.

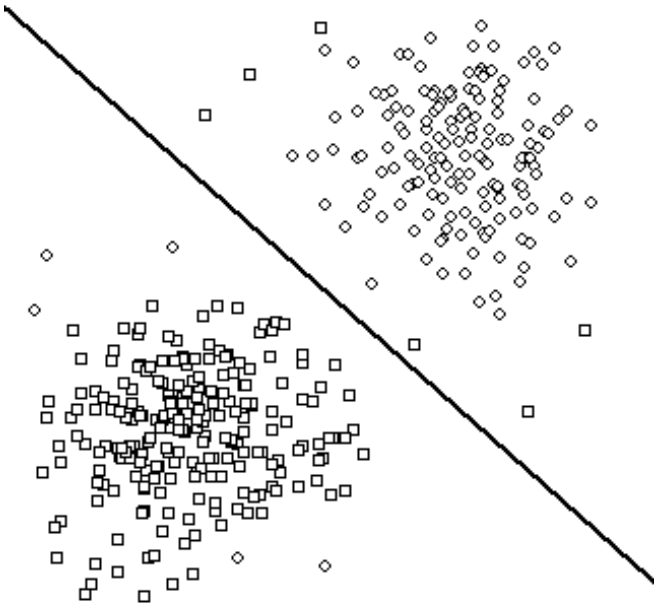


It is clear that in the example above the positively labeled points, represented by circles, have a far greater standard deviation on the x-axis than the negatively labeled points, represented by squares. As such, the separating hyperplane is chosen such that is compensates for the different ways the two sets vary.

As a consequence of the way the problem has been modeled, it naturally allows for training errors.



It easily deals with outliers as well. Such a situation is described in the toy example bellow.



The objective of PVM is to optimize the margin distribution and to that effect, table (1) shows a toy example with the induced margin distribution.

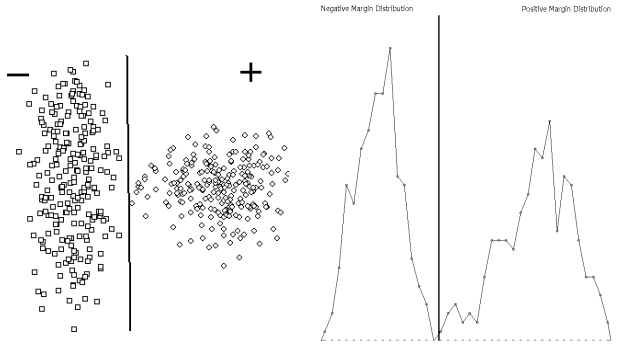


Table 1: Left: Toy Example. Right: Induced margin distribution

We have tested the proposed algorithm against SVM on problems part of the UCI ML repository. Two testing methods have been used. First, we wanted to show that using the same kernel as a typical SVM we would get better results, in order to prove that the hyperplane selection rule leads to better classifiers (on these instances at least). The selection of parameters for the kernel was done using five fold. The results were encouraging. We then allowed the PVM to search for its own kernel parameters using five fold as well. We report the findings in table (2).

As can be seen PVM obtains a better accuracy in most cases, avoiding overfitting in the one test where SVM overfits. It performs worse than SVM in only one of the five data sets.

Problem	SVM	PVM	PVM 5Fold
heart SPECT	75.93	91.97	**91.97**
ionosphere	90.0	90.3	**90.3**
liver	70.04	69.16	**71.36**
heart SPECTF	40.64	43.85	**91.97**
magic telescope	**80.51**	67.88	76.6

Table 2: Testing done with SVM and PVM

As a special note we should mention that for testing, the objective of system (8) was modified to handle badly underdetermined systems. As such, it was transformed to $lexicomin(max\{\frac{\sigma_p}{E_p}, \frac{\sigma_n}{E_n}\}, -min\{E_p, E_n\})$, with the auxiliary constraints

$$\sigma_p < 1e+10, \sigma_n < 1e+10,$$
$$E_p < 1e+20, E_n < 1e+20$$

.

The effect is that when there exists more than one separating hyperplane that yield the same optimal value of the first objective term, then the one which maximizes $minE_p, E_n$ is chosen. Coupled with the bounds on $\sigma_p, \sigma_n, E_p, E_n$ this leads to choosing the hyperplane which induces the largest mean values for distances.

5. Conclusions

We have introduced a novel method for choosing the separating hyperplane in a robust manner and have validaded the method through testing on a benchmark dataset. Unlinke previous methods that use the margin distribution ([4]), the current method guarantees optimality of the separation rule.

As future work we shall focus on improving the computational cost of the method and parallelizing the linear solvers, as currently we were only able to train on modest scale examples. As this is a probabilistic method, its strength lies in exploiting large enough datasets, an aspect hindered by the yet improperly sorted computational costs.

6. Acknowledgements

This research has been supported through financing offered by the POSDRU/88/1.5/S/47646 Project for PHDs.

References

[1] Cortes, C., Vapnik, V. N., "Support-Vector Networks," 1995
[2] Vapnik, V.N., "Statistical learning theory," New York: John Wiley and Sons Inc., 1998
[3] Garg, A., Har-Peled, S., Roth, D., "On generalization bounds, projection profile, and margin distribution," Proc. of the International Conference on Machine Learning, 2002
[4] Garg, A., Roth, D., "Margin Distribution and Learning Algorithms," Proc. of the International Conference on Machine Learning, 2003
[5] Boyd, S., Vandenberghe, L., "Convex Optimization," Cambridge University Press
[6] Chapelle, O., Vapnik, V. N., "Choosing Multiple Parameters for Support Vector Machines," 2001

[7] Rong-En Fan, Pai-Hsuen Chen, Chih-Jen Lin, "Working Set Selection Using Second Order Information for Training Support Vector Machines," Journal of Machine Learning Research, 2005

Mining Association Rules from Responded Questionnaire of Sanitary Education Guidance

Yo-Ping Huang, Zheng-Hong Deng and Shan-Shan Wang
Department of Electrical Engineering
National Taipei University of Technology
Taipei 10608, Taiwan

Abstract - *Since it can provide high-contrast and is good to produce uniform brightness, FTIR multi-touch technology is chosen to design the proposed platform. The device is designed to display propagandas, images, videos, and games to propagate knowledge of sanitary education and disease prevention to users. Furthermore, a physical and mental health questionnaire scale is implemented for users to on-line fill out the questionnaire after logging into the system by their own RFID tags. The responded questionnaires are then used to find the association rules among them based on various combinations of grade intervals, minimum supports, and minimum confidences. Experimental results show that the proposed system can discover interesting association rules from users' responded questionnaire.*

Keywords: data mining, FP-tree, touch display system, sanitary education guidance.

1 Introduction

Sanitary education guidance is normally propagated by pamphlet or by playing propaganda video on screen wall. But the pamphlets are usually glanced and then thrown away. By playing propaganda video on screen wall, the public usually looks at flowers while riding on horseback and does not pay much attention on the story of video contents. In view of these problems, this study designed an easy to operate multi-touch display system that provides better interaction with users to approach the effect of human-computer interaction, and shorten the gap of user and information technology. The proposed multi-touch system shows propagandas, images, videos and games about sanitary education guidance. According to the habits of users, they can enlarge, condense, drag or rotate images/videos by their finger(s). Users' interactions with the multi-touch system can be transformed to transactions-like patterns for underlying association rules mining.

In this paper we focus on discovering the associations that users responded to the physical and mental health questionnaire scale implemented on the proposed multi-touch display platform. Users could realize their mental situations from the statistical results of the questionnaire. To extract interesting association rules from the database each grade level has 500 records and five grade levels in total have 2500 records for performing the simulations. Under different minimum support and minimum confidence constraints, simulation results show that the proposed data mining method can discover interesting association rules among items at different grade levels. The presented results can provide references for psychiatrists in judging the degree of melancholia and in tailoring the contents of sanitary education guidance.

This paper is organized as follows: section 2 discusses the related work. System descriptions are detailed in Section 3. Experimental results and analyses are given in Section 4. Conclusions are made in the final section.

2 Related works

2.1 Data mining

Data mining technology is commonly used to discover the implicit relationships among items recorded in consumers' transaction databases [1,14-19]. The mined association rules can provide valuable information for enterprises engaged in marketing mix and market forecast [2-6]. Association rules are discovered to express relationships among items. Assuming I is a set of all items in the transactional database while T is a subset of I, i.e., $T \subseteq I$. An association rule is described between itemset X and itemset Y. Support and confidence are two constraint parameters defined to determine whether an association rule is interesting or not. Support value of association rule $X \rightarrow Y$ means the probability that both X and Y appear simultaneously in all transactions while the confidence value of $X \rightarrow Y$ expresses the probability of Y's appearance under the premise of X's existence.

(1) FP-tree algorithm

Among various practical data mining algorithms FP-tree algorithm was first proposed to reduce the execution time countered in repeatedly generating

candidate itemsets during mining processes [9]. Only the initial mining step is necessary to find out large one-itemsets. The advantage of FP-tree algorithm is to avoid producing a large number of candidate itemsets that in turn can greatly improve a fatal drawback of traditional Apriori algorithm [7, 8]. The following briefly introduces the procedures in constructing a FP-tree.

Step 1: Scanning the transactional database and omitting the duplicate items to find out large 1-itemset L_1 that has a support value greater than or equal to the minimum support.

Step 2: According to the support values sorting the large 1-itemset L_1 in descending order. Note that each item (e.g., A) is followed by a colon (:) and a figure (e.g., 3) such as A:3 in the tree structure. The figure represents the support value for that item in the database.

Step 3: Following step 2 to reorder each transaction in the original database to create a virtual database with ordered frequent items. A root node is created to construct the FP-tree.

Step 4: The ordered frequent items are sequentially added to the FP-tree. Whenever a transaction is to be added to the current FP-tree, the existing paths are checked. If the prefix of the transaction matched the existing path, then the counter of each node in the path is incremented by 1; otherwise, a new branch is created for the suffix of the transaction. The process continues until all transactions have been added to the FP-tree.

(2) FP-Growth algorithm

FP-Growth algorithm is adopted to extract every frequent itemset from the FP-tree [10]. FP-Growth algorithm searches the frequent itemsets by reversing the order of large 1-itemset L_1. That is, the one with the smallest support value in L_1 is initiated to find frequent itemsets, followed by the second smallest, and so on. The process is continued until the highest-support item is joined to find frequent itemsets. The following summarizes the procedures to find the frequent itemsets.

Step 1: Find all frequent itemsets that include the one with the smallest support value in L_1. Then we can find the conditional FP-tree for the smallest-support item.

Step 2: Find all frequent itemsets that comprise the one with the second smallest support value in L_1. For example, the second smallest item is 7c with a support value of 3, i.e., 7c:3. Suppose there are two paths to traverse from the root node to node 7c, including <5c:4, 9b:4, 2b:3, 3d:2, 7c:2> and <3d:1, 7c:1>. Then, the prefixes of item 7c include <5c:2, 9b:2, 2b:2, 3d:2> and <3d:1>. In the conditional FP-tree for 7c only item 3d has a support value larger than or equal to the minimum support 3. As a result, the frequent itemset under the condition of 7c is {3d, 7c}.

Step 3: The process is continued until all items have been used to find frequent itemsets.

2.2 Multi-touch technology

Frustrated Total Internal Reflection (FTIR) technology was first created by Han [11]. FTIR technology is an application of optical principle. When light is penetrating between two different media part of light source will be refracted to another medium and the rest will be reflected. But, if the incident angle is larger than the critical angle (i.e., light is far away from normal), then the light will not be refracted to the other medium and will be completely reflected to the inside [12]. Critical angle is the minimum incident angle that the FTIR will occur. The critical angle θ_c is calculated as follows:

$$\theta_c = arcsin\left(\frac{n_2}{n_1}\right), \quad (1)$$

where n_2 and n_1 are the refraction rates for the low density medium and high density medium, respectively. When the incident light is exactly equal to the critical angle, the refracted light will follow the tangent of refraction interface to propagate. Taking the visible light to incise to air or vacuum for example, the critical angle is about 41.5 degrees.

FTIR multi-touch technology is chosen to implement the proposed platform. The multi-touch display system that integrates infrared light source, webcam and projector can detect 40 touch points simultaneously and allow users to enlarge, condense, drag or rotate images/videos by their finger(s). Users' interactive events with the multi-touch display platform can be stored in the database for behavior analyses. FP-tree and FP-Growth algorithms are used to discover association rules from multitudes of the responded physical and mental health questionnaire scale.

3 Descriptions of system design

The processes to design the sanitary education guidance system are shown in Fig. 1. There are four principal functions in the proposed system: propaganda, image, video and game about sanitary education guidance. Propaganda is designed to introduce diseases and display the mental health questionnaire scale system. Users can login the system by RFID tags to fill out the questionnaire. The procedures to design the mental health questionnaire scale system are shown in Fig. 2. The back-end database stores every user's record that includes UID code, date, starting time, finish time, total time and total grades in answering the questionnaire. The data can provide references for psychiatrists in judging the degree of melancholia.

Psychiatrists, assistants and system administrators have the authority to manage the questionnaire management system. For security reason, they need to use RFID tags or input RFID UID codes to login and surf the

answered historical records. Fig. 3 shows the flowchart of questionnaire management system.

To realize how users responded to the questionnaire, data mining technology is applied to find the association rules among the answers of each grade level based on combinations of grade intervals, minimum supports, and minimum confidences. The procedures to find the association rules are given in Fig. 4. The purpose to extract the association rules among multitudes of answers is to provide useful information for psychiatrists to understand whether most users have similar physical and mental reactions under melancholia. In this study, we use FP-tree and FP-Growth algorithms to find the association rules from answered questionnaire.

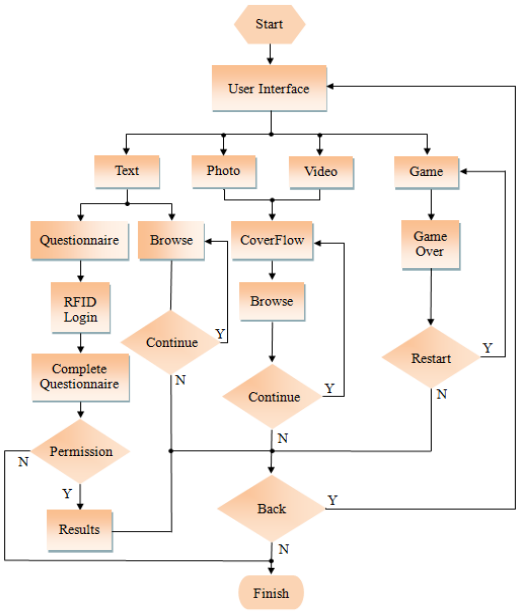


Fig. 1. The flowchart of sanitary education guidance system.

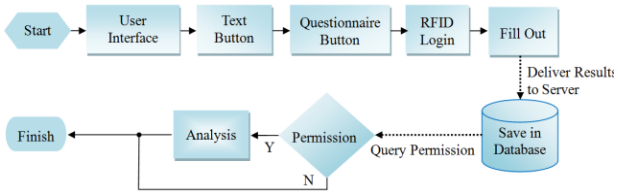


Fig. 2. The procedures in designing the mental health questionnaire scale system.

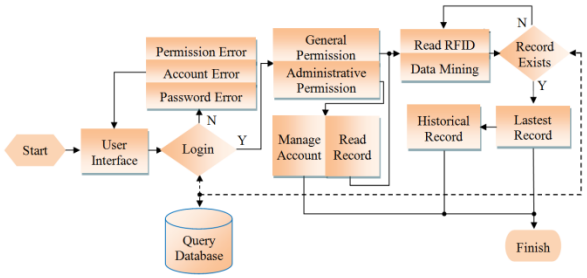


Fig. 3. The flowchart of the questionnaire management system.

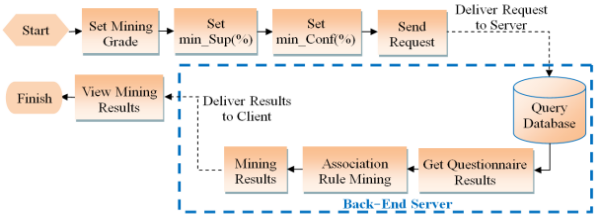


Fig. 4. The procedures of finding association rules from answered questionnaire.

4 Experimental results and analyses

4.1 Multi-touch system

The proposed multi-touch table is implemented by FTIR technology [21]. An infrared camera is beneath the acrylic and infrared LED arrays are mounted around the acrylic edges. A projector is installed at the bottom of the multi-touch table to display the contents. The test of touching the acrylic screen by fingers with CCV (Community Core Vision) version 1.3 is shown in Fig. 5 [13]. We can find out that the contrast effect is pretty good and it improves the problem of highlight if the light is projected bottom-up.

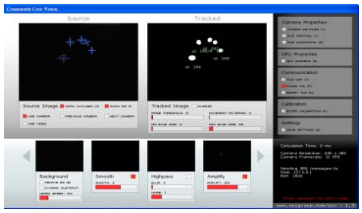


Fig. 5. The touching effect by fingers on CCV version 1.3.

Any objects appearing on the multi-touch table should be identified by the camera whenever they are surfing on the table [20]. The designed acrylic table has a size of 81cm×56cm. A cross mark is labeled when a point cannot be detected by the camera. Then, we can calculate the contact rate from dividing the identified area by the acrylic area as compared in Table 1 when different situations are used. Note that in Table 1, “static” means that fingers contact the surface still; however, “dynamic” means that fingers are allowed to surf on the screen steadily. The higher the contact rate is the better the multi-touch system identifies the finger contacts.

When only two strips of LEDs are used to provide light source of FTIR, the farther contacts to the light source cannot be easily identified. In case two more strips of LEDs are configured against the edges of acrylic sheet, it is obvious that the contact rates are highly improved as given in Table 1. When fingers slide on the surface that has silicon rubber, the camera can recognize the contact positions. When fingers contact still on the acrylic sheet without silicon rubber the contact rates are good. However, when sliding the fingers on the surface without silicon

rubber the contacts cannot be identified because the sliding fingers cannot destroy the IR light source of FTIR. From Table 1, it is also interesting to find that a 1mm thickness of silicon rubber on the surface the contact rates decline due to the lack of sensed pressure from the contacts. If the thickness is reduced to about 0.1mm, it is easier to destroy the characteristic of FTIR.

Table 1. The contact rates of the devised table.

Si rubber \ IR LED		2 strips of LEDs (83 LEDs)	4 strips of LEDs (166 LEDs)
w/o Si rubber	static	70%	99%
	dynamic	0%	<5%
<0.1mm (thin)	static	75%	99%
	dynamic	70%	85%
1mm (thick)	static	60%	90%
	dynamic	40%	60%

4.2 System interface

Adobe Flash CS4 (Creative Suite 4) is used to design the interface of sanitary education guidance and animation is adopted to richly display virtual hospital and background. Sanitary education guidance is programmed with vivid illustrations and animations rather than the traditional boring propaganda. There are five major functions in the interface, including propagandas, images, videos, games and questionnaire about sanitary education guidance. The animations are designed to provide effective human-computer interaction.

(1) Propaganda

There are four functions in propaganda: physical and mental health questionnaire scale, H1N1, enterovirus, and dengue fever. If a user clicked on the dengue fever, the interface is popped up as Fig. 6. The contents of guidance appear in the central interface. Users can either click on the right or left arrow to select different contents or use sliding mode to switch the selections. The browsing disease title as well as the ith film in play and the number of films in this disease will appear at the bottom center. For example, there are 7 films about dengue fever and the 2nd one is being viewed as shown in Fig. 6. Users can also use their mobile phones to decode the QR Code to link to central disease bureau for detailed information. Note that the QR Code image on the screen is allowed to drag, rotate, enlarge or condense under users' preference.

Fig. 6. The propaganda interface.

(2) Photos

There are three categories of disease photos, including H1N1, dengue fever and enterovirus. If users are interested in browsing H1N1 related photos, they can click on "H1N1" icon to enter into CoverFlow photo browsing page as shown in Fig. 7.

When clicking on the active photo, the CoverFlow photo browsing page will disappear and only the active photo appears on the screen. Then it can be dragged, rotated, enlarged or condensed by users' own interests.

Fig. 7. The CoverFlow photo browsing page.

(3) Video

There are also three categories of disease photos, including H1N1, dengue fever and enterovirus. If a user clicked the H1N1 video category, the system will use CoverFlow to capture a frame for playing the video. When the user clicked the captured frame again, the system starts playing the film. In the bottom right hand side corner, there is a counter to indicate how much time remained for playing this film as shown in Fig. 8. After finishing playing the film, there is a button for choosing whether to replay or not. Besides, users can drag, rotate, enlarge or condense the film. The "Back" icon in the bottom left hand side corner is designed for returning back to the CoverFlow photo browsing mode. As to the bottom right hand side corner, the "Home" icon is for returning back to the home page, "Video" icon is for seeing films and "Text" icon is for reading propaganda.

Fig. 8. The video playing interface.

(4) Games

The proposed system also provides memory games for users to match disease names with their corresponding symbols as shown in Fig. 9. At the beginning of the game, all cards are displayed in front faces. When the display time is up, they are flipped to the back faces for users to find the matches. There are 8 pairs, i.e., 16 cards, for the match

game. For example, a plague card should be matched to rat symbol for users to understand that plague is transmitted by fleas carried by rats. To prevent users from losing interests in playing the game, there is a semi-transparent symbol hint on each disease name card so that users can find the matching card easily.

Fig. 9. Card-flipping game.

(5) Physical and mental health questionnaire scale

Currently the physical and mental health questionnaire scale is answered by pen on papers. It is time-consuming to get the statistical results from the responded questionnaire. The proposed sanitary education guidance also provides the physical and mental health questionnaire scale for users to on-line fill out the questionnaire. Users can proceed from the "Text" button and select the questionnaire icon to start the system. Then, users are asked to login the system by their own RFID tags. The responded questionnaire is stored in the back-end database for immediate statistical analysis that allows those who have the privileges to read the results. A responded questionnaire is shown in Fig. 10 where the upper part is the scores and some suggestions for the user from the analytical result.

Fig. 10. The analytical result from a responded questionnaire.

4.3 Mining association rules from responded questionnaire

There are nine questions in the physical and mental health questionnaire scale. Each question uses 4 scales, i.e., 0 to 3 points, to reflect the degree conforming to the question. Currently, the questionnaire is answered by pen on papers that make them tedious for statistical analysis. The proposed multi-touch system is designed to automate the process. Any responded questionnaire is stored in the back-end server for instant analysis.

To find the association rules among answers from different respondents, each answer is relabeled as *ij* where *i* is the question numbering from 1 to 9 and *j* ranges from English alphabets A to D corresponding to scales 0 to 3. For example, a responded questionnaire after transformation may look like the pattern: {1A,2B,3D,4C,5B,6C,7D,8B,9A}. In our experiments, the responded total scores are further divided into 5 grade levels, including good condition (scores 0-4), a little bothersome emotion (scores 5-9), slight melancholia symptom (scores 10-14), medium melancholia symptom (scores 15-19), and heavy melancholia symptom (scores 20 and up). The upper part of Fig. 11 shows an example how the ordered frequent items are found from five responded questionnaires and the lower part is the complete FP-tree from the questionnaires.

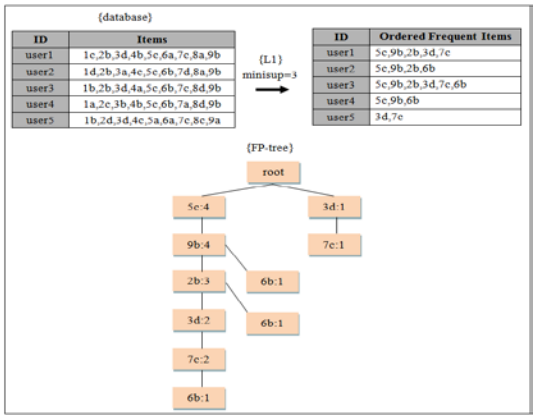


Fig. 11. An example of constructing FP-tree.

Based on the combinations of different grade levels and preset minimum support and confidence values, multitudes of simulations were performed for analyses. To obtain informative analytical results, each score level has 500 randomly generated data and a total of 2,500 data are stored in our database. Simulations under the combinations of minimum support of 2%, 3%, 4%, 5% and minimum confidence of 40%, 45%, 50% were performed for comparisons.

Based on the simulation results, it is hard to find the common association rules from those five melancholia levels. Three other case studies are considered instead.

- The frequent itemsets appeared at least 3 times under the same minimum support and confidence at different levels. In this case, the frequent itemsets only appeared at score levels of (0-4), (5-9), and (10-14) under the minimum support of 2%, 3%, 4% and minimum confidence of 40%. Two of the association rules are listed below for illustration.

Rule 1: {2C,6A}

It means "over half" of the symptom that cannot eat or over eat implies that "not at all" does things without interest or fun.

Rule 2: {4C,6A}

It means "over half" of the symptom that cannot concentrate on things (for example, reading newspaper or watching TV) implies that "not at all" does things without interest or fun.

- The frequent itemsets appeared at least 2 times at the slight to heavy score levels of (10-14), (15-19), and (20 and up) under the minimum support of 2%, 3%, 4%, 5% and minimum confidence of 45%. Three of the association rules are listed below for illustration.

Rule 3: {4B,5D}

It means that "sometimes" the patients cannot concentrate on things (for example, reading newspaper or watching TV) implies that "almost every day" they react things or speak slowly or are apparently fidgety than before.

Rule 4: {6B,8D}

It means "sometimes" the patients do things without interest or fun implies that "almost every day" they feel not good enough (for example, feel themselves failure, sorrow or shame).

Rule 5: {1B,8D}

It means "sometimes" the patients have sleeping problems (for example, cannot fall asleep, easy to wake up, or oversleep) implies that "almost every day" they feel not good enough (for example, feel themselves failure, sorrow or shame).

- The frequent itemsets at the slight to medium score levels of (10-14) and (15-19) only appeared under the conditions of minimum support of 2% and minimum confidence of 40%. Two of the association rules are listed below for illustration.

Rule 6: {3A,6C,7C}

It means "not at all" the patients feel tired or no energy and "over the half" do things without interest or fun implies that "over the half" they have melancholia (for example, feel upset, depressed or despair).

Rule 7: {3B,4C,8C}

It means "sometimes" the patients feel tired or no energy and "over the half" cannot concentrate on things (for example, reading newspaper or watching TV) implies that "over the half" they feel not good enough (for example, feel themselves failure, sorrow or shame).

5 CONCLUSIONS

A multi-touch display system using Frustrated Total Internal Reflection (FTIR) technology based on vision-style was proposed for sanitary education guidance. The physical and mental health questionnaire scales were also implemented by such a design to automate the statistical analysis. The responded questionnaire answers were stored in the back-end database for data mining. FP-tree and FP-Growth algorithms were exploited to extract the association rules among items at different grade levels. The analytical results from physical and mental health questionnaire scales can provide references for psychiatrists in judging the degree of melancholia and in tailoring the contents of sanitary education guidance. Association rules analyses among users' interactive actions from viewing propagandas, images, videos, games and questionnaire about sanitary education guidance on the multi-touch platform are under investigation.

6 ACKNOWLEDGMENTS

This work was supported in part by National Science Council, Taiwan under Grants NSC99-2221-E-027-059- and NSC97-2221-E-027-034-MY3 and in part by Ministry of Education, Taiwan under Grant 99T262.

7 REFERENCES

[1] R. Agrawal, T. Imielinski and A. Swami, "Mining association rules between sets of items in large database," in *Proc. of Int. Conf. on Management of Data*, New York, NY, U.S.A., pp.207-216, May 1993.

[2] C. Berberidis, L. Angelis and I. Vlahavas, "Inter-transaction association rules mining for rare events prediction," in *Proc. of the 3rd Int. Conf. on Artificial Intelligence*, Samos, Greece, pp.308-317, May 2004.

[3] E. Georgii, L. Richter, U. Ruckert and S. Kramer, "Analyzing microarray data using quantitative association rules," *IEEE Trans. on Bioinformatics*, vol. 21, no. 2, pp.123-129, September 2005.

[4] B. Berendt, "The semantics of frequent subgraphs: Mining and navigation pattern analysis," in *Proc. of Int. Conf. on Knowledge Discovery*, Saarbrücken, Germany, pp.91-102, October 2005.

[5] K. Wang, Y. He, D. Cheung and F. Chin, "Mining confident rules without support requirement," in *Proc. of Int. Conf. on Information and Knowledge*, Atlanta, GA, U.S.A., pp.89-96, November 2001.

[6] K. Wang, S. Zhou and Y. He, "Growing decision trees on support-less association rules," in *Proc. of Int. Conf. on Knowledge Discovery and Data Mining*, Boston, MA, U.S.A., pp.265-269, August 2000.

[7] X. Shang, K. Sattler and I. Geist, "SQL based frequent pattern mining without candidate generation," in *Proc. of the ACM Symp. Conf. on Applied Computing*, Nicosia, Cyprus, pp.618-619, March 2004.

[8] C.-M. Cha and Y.-C. Tai, "An SQL-based improvement of the FP-tree construction technique," *Information Management Research*, vol. 6, pp.31-46, July 2006.

[9] J. Li, A.W.C. Fu and P. Fahey, "Efficient discovery of risk patterns in medical data," *Artificial Intelligence in Medicine*, vol. 45, no. 1, pp.77-89, January 2009.

[10] Y.-P. Huang, H.-W. Chiu, W.-P. Chuan and F.E. Sandnes, "Discovering fuzzy association rules from patient's daily text messages to diagnose melancholia,"

in *Proc. of IEEE SMC*, Istanbul, Turkey, pp.3523-3528, October 2010.

[11] J.Y. Han, "Low-cost multi-touch sensing through frustrated total internal reflection," in *Proc. of Int. Conf. on User Interface Software and Technology*, New York, NY, U.S.A., pp.115-118, October 2005.

[12] Wiki for FTIR technology, http://zh.wikipedia.org/zh-tw/%E5%85%A8%E5%8F%8D%E5%B0%84

[13] Getting Started with CCV, http://wiki.nuigroup.com/Getting_Started_with_tbeta

[14] A. Wong and G. Li, "Simultaneous pattern and data clustering for pattern cluster analysis," *IEEE Trans. on Knowledge Analysis and Data Engineering*, vol. 20, no. 7, pp.911-923, July 2008.

[15] H. Xiong, J. Wu and J. Chen "K-Means clustering versus validation measures: a data-distribution perspective," *IEEE Trans. on Systems, Man, and Cybernetics—part B: Cybernetics*, vol. 39, no. 2, pp.318-331, April 2009.

[16] D.-A. Chiang and C.-T. Wang, "The cyclic model analysis on sequential patterns," *IEEE Trans. on Knowledge Analysis and Data Engineering*, vol. 21, no. 11, pp.1617-1628, November 2009.

[17] J.-W. Huang and C.-Y. Tseng, "A general model for sequential pattern mining with a progressive database," *IEEE Trans. on Knowledge Analysis and Data Engineering*, vol. 20, no. 9, pp.1153-1167, September 2008.

[18] Y. Huang and L. Zhang, "A framework for mining sequential patterns from spatio-temporal event data sets," *IEEE Trans. on Knowledge Analysis and Data Engineering*, vol. 20, no. 4, pp.433-448, April 2008.

[19] Q. Ding and W. Perrizo, "PARM- An efficient algorithm to mine association rules from spatial data," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 6, pp.1513-1524, December 2008.

[20] Z. Elizabeth, B. Rachel, G. Peter, D. Kelly and M. Andrew, "Infant imitation from television using novel touch screen technology," *British Journal of Developmental Psychology*, vol. 27, pp.13-26, March 2009.

[21] P.I.S. Lei and A.K.Y. Wong, "The multiple-touch user interface revolution," *IEEE IT Professional*, vol. 11, no. 1, pp.42-49, January/February 2009.

A NEW TERM WEIGHTING SCHEME FOR DOCUMENT CLUSTERING

A. Keerthiram Murugesan[1] **and B. Jun Zhang**[1]
[1]Department of Computer Science, University of Kentucky, Lexington, KY, USA

Abstract— *In this paper, we present a Cluster-Based Term weighting scheme (CBT) for document clustering algorithms based on Term Frequency - Inverse Document Frequency ($TF-IDF$). Our method assigns the term weights using the information obtained from the generated clusters and the collection. It identifies the terms that are specific to each cluster and increases their term weight based on their importance. We used the K-means partitional clustering algorithm to compare our method with three widely used term weighting schemes such as $Norm-TF$, $TF-IDF$, and $TF-IDF-ICF$. Our experimental results show that the new method outweighs the existing term weighting schemes and improves the result of a clustering algorithm.*

Keywords: Document clustering; Term weighting; $TF-IDF$

1. Introduction

Document clustering algorithms cluster documents of similar types together. They are used in information retrieval, information extraction, machine learning, topic detection, document organization and other applications.

Document clustering is an unsupervised technique, i.e., users do not provide any information about the given data, to the clustering algorithm. It needs to find the hidden information in the data by itself. As mentioned in [1], only certain terms extracted from a document can be used for identifying and scoring a document within the collection. Term weighting schemes can be used to identify the importance of each term with respect to a collection and assign weights to them accordingly. Document clustering uses these term weights to compare the similarity of two documents. Several term weight schemes are in use today, but none of them is specific to the clustering algorithms.

In this paper, we present a new term weighting method for clustering algorithms based on the traditional $TF-IDF$ term weighting scheme. Our motivation is based on the idea that the terms of documents in a same cluster have similar importance than the terms of documents in a different cluster. We concentrated on terms that are important within a cluster and consider the other terms as irrelevant and redundant. We implemented this idea by giving more weight to the terms that are common within the cluster but uncommon in other clusters. During our experiment, we found that the new term weighting scheme based on the clusters gives better results than the other well-known term weight schemes traditionally used for the document clustering.

This paper is organized as follows: Section 2 describes the information related to our method. Section 3 gives our proposed method. Section 4 presents the evaluation methodology and the results of our experiment. At the end, we summarize our work and the references used in this paper.

2. Related Works

In this section, we discuss the general information related to term weighting and document clustering. Traditionally, the Boolean retrieval model assigns 1 or 0 based on the presence or absence of the terms in a document. This model performs undesirably in querying for a document. Later, the Vector space model was introduced for ranked retrieval [2]. It is widely used in querying documents, clustering, classification and other information retrieval operations because it is simple and easy to understand. It uses a bag of word approach. Each document d_i in the collection ζ is represented as a vector of terms,

$$d_i = \{term_1, w_{1i}; term_2, w_{2i}; \cdots term_T, w_{Ti}\}. \quad (1)$$

And each term $term_t$ in a document d_i is assigned a weight w_{it} which represents its importance. The term weight w_{it} determines whether the term $term_t$ will be included in the further steps. Several term weighting schemes have been proposed to compute the importance of a term in a document and in a collection. $Norm-TF$, $TF-IDF$, ATC, LTU, and Okapi are some of the widely used term weighting schemes [1], [3]. One of the most commonly used term weighting schemes is $TF-IDF$. It measures the importance of a term using its frequency within a document and the inverse of its document frequency within the collection. This paper extends this idea to the clustering algorithms.

Several papers have suggested modifying an existing term weighting scheme for their methods. [4] shows a modified $TF-IDF$ term weight to avoid single terms from getting higher weight. [5] proposes a new term weighting scheme $TF-ICF$ (Term Frequency Inverse Corpus Frequency) for clustering a dynamic data stream. It uses the existing collections to weight the terms in the data stream. Also, [6] used $DF-ICF$ Inter- and Intra-cluster components for extracting a description from a cluster.

In this paper, we extend the idea of [5] and the cluster components used in [6] to introduce a new term weighting scheme that efficiently uses the cluster information obtained from the clustering algorithm.

Table 1: Term weighting schemes.

	Term weighting schemes
$Norm - TF$	$w_{it} = \frac{f_{it}}{\sqrt{\sum_T f_{it}^2}}$
$TF - IDF$	$w_{it} = f_{it} \log \frac{N}{N_t}$
$TF - IDF - ICF$	$w_{itj} = f_{it} \log \frac{N}{N_t} \log \frac{K}{K_t}$

Table 1 shows representation of some of the term weighting schemes used in this paper. Here, TF is the Term Frequency, IDF is the Inverse Document Frequency, and ICF is the Inverse Cluster Frequency. w_{itj} is the weight of a term t in a document i of the cluster j. $tf_{it} = f_{it}$ is the term frequency of a term t in a document i. $idf_t = \log \frac{N}{N_t}$ is the inverse document frequency for a term t in a collection ζ, where N is the total number of documents in the collection and N_t is the number of documents that contain the term t. $icf_t = \log \frac{K}{K_t}$ is the inverse cluster frequency of a term t in the collection ζ, where K is the total number of clusters in the collection and K_t is the number of clusters that contains the term t.

3. The Proposed Method

In this section, we introduce our new term weighting scheme along with its notation. For a term t, document i, and cluster j, CBT is given as:

$$w_{itj} = tf_{it}.idf_t.df_{tj}.icf_t \quad (2)$$

$$= f_{it}.\log \frac{N}{N_t}.\frac{df_j}{|C_j|}.\log \frac{K}{K_t} \quad (3)$$

Equation (2) is equivalent to Equation (3). Here, $df_{tj} = \frac{df_j}{|C_j|}$ is the document frequency of a term t within the cluster j, where df_j is the number of documents in the cluster j that contain the term t, and $|C_j|$ is the total number of documents in the cluster j.

Our new term weighting method has four components. The first two components are based on the term weighting components discussed in [1]. The last two components are the cluster components as shown in the Table 2.

In other words, CBT assigns a weight to a term which is

- Highest when the term occurs more frequently in the documents of a cluster and uncommon in other clusters.
- Higher when the term occurs less frequently in the documents of a cluster and uncommon in other clusters.
- Lower when the term occurs often in a few clusters.
- Lowest when the term occurs in most of the documents in a collection.

Table 2: List of Components in CBT term weighting scheme.

Component	Description
tf_{it}	Term Frequency Component. High when term t occurs often in a document i.
idf_t	Collection Frequency Component. High when term t occurs less often in the entire collection.
df_{tj}	Intra-cluster Frequency Component. High when term t occurs more often in a cluster j.
icf_t	Inter-cluster Frequency Component. High when term t occurs less often in clusters other than cluster j.

4. Experimental Evaluation

In our experiment, we used one of the popular unsupervised partitional clustering methods, the K-means algorithm [7], [8]. The K-means clustering algorithm tries to minimize the following objective function $O(N, K)$:

$$O(N, K) = \sum_{j=1}^{K} \sum_{d_i \in C_j} \|d_i - c_j\|^2 \quad (4)$$

where $\|d_i - c_j\|^2$ is the distance between the document d_i and the centroid c_j. The centroid of the documents in a cluster C_j can be computed as:

$$c_j = \frac{1}{|C_j|} \sum_{d_i \in C_j} d_i \quad (5)$$

To show that the CBT term weighting scheme improves the quality of the clusters, we ran the K-means algorithm with the four term weighting schemes discussed in the previous sections. The results of the K-means algorithm directly reflect the impact of the term weighting schemes in the clustering algorithm.

Require: An integer $K \geq 1$, Document Collection ζ.

1: **if** $K = 1$ **then**
2: **return** ζ
3: **else**
4: Initialize $l = 0$, $df_{tj} = 1$ and $icf_t = 1$, $t: 1 \ldots T$, $j: 1 \ldots K$
5: $\{C_1^{(0)}, \cdots, C_K^{(0)}\} \leftarrow RANDCLUSTERS(\zeta, K)$
6: **repeat**
7: **for all** $d_i \in \zeta, i: 1 \ldots N$ **do**
8: $m = \arg\min_j |c_j - d_i|$
9: $C_m^{(l+1)} \leftarrow C_m^{(l+1)} \cup d_i$
10: **end for**
11: $l \leftarrow l + 1$
12: $w_{itj} \leftarrow tf_{it}.idf_t.df_{tj}.icf_t$; for each term $term_t$ in a document d_i of a cluster $C_j^{(t)}$, $t: 1 \ldots T$, $i: 1 \ldots N$, $j: 1 \ldots K$
13: **for** $j = 1$ **to** K **do**

14: $c_j \leftarrow \frac{1}{|C_j^{(l)}|} \sum_{d_i \in C_j^{(l)}} d_i$
15: **end for**
16: **until** No change in K centroids
17: **return** $\{C_1^{(l)}, \cdots, C_K^{(l)}\}$
18: **end if**

Initially, the K-means algorithm doesn't have any information about the cluster components, so we start the algorithm by setting df_{tj} and icf_t to 1 as shown in the line 4 and update the inter- and intra-cluster component on each iteration. If a document has a set of terms that doesn't belong to a cluster, then its term weight will be reduced so that it will move to other clusters. It will be repeated until it finds a suitable cluster of its type.

The runtime complexity of the traditional K-means algorithm is $O(LNK)$ where L is the total number of iterations in the outer loop, N is the total number of documents in a collection and K is the total number of clusters. The algorithm shown above updates the term weights for all clusters on each iteration to reflect the changes made in the new clusters, so it takes $O(LNK + LK) = O(LNK)$ since $LK < LNK$.

4.1 Data sets

We used TREC [9], 20 Newsgroup [10], and Reuters-21578 [11] data collections for our experiment. TR11, TR12, TR23, TR31, and TR45 collections are taken from TREC-5, TREC-6 and TREC-7. 20 NG S1 - S5 are the five randomly chosen subsets of 20 Newsgroup documents [12]. RE S1 and RE S2 data sets are from Reuters-21578 collection. For the RE S1 data set, we filtered documents from the original Reuters-21578 data set that belongs only to a single category. We got 4645 documents that have only one category. In addition to that, we used the Reuters transcribed subset (RE S2) [13]. For all the data sets shown in Table 3, we removed the stop words and stemmed using the Porter stemming algorithm [14].

Table 3: Data sets.

Data set	Collection	# of Doc	# of Class
TR11	TREC	414	9
TR12	TREC	313	8
TR23	TREC	204	6
TR31	TREC	927	7
TR45	TREC	690	10
20 NG S1	20 Newsgroup	2000	20
20 NG S2	20 Newsgroup	2000	20
20 NG S3	20 Newsgroup	2000	20
20 NG S4	20 Newsgroup	2000	20
20 NG S5	20 Newsgroup	2000	20
RE S1	Reuters-21578	4645	59
RE S2	Reuters-21578	200	10

4.2 Evaluation Metrics

Document clustering algorithms are evaluated using external quality criterion [7]. External quality criterion evaluates the clustering algorithm by comparing the generated clusters with the known classes. These classes are usually produced by the human judgment or by a classification algorithm. Most commonly used external quality measure for partitional algorithm is the Entropy [15], [16], [17].

If $C_1, C_2, \cdots C_K$ are the K clusters generated by a clustering algorithm and $\Omega_1, \Omega_2 \cdots \Omega_C$ are the C classes, then the Entropy of the cluster C_j can be defined as:

$$H_j = -\sum_{k:\ 1\cdots C} p_{kj} \log(p_{kj}) \quad (6)$$

$$H_j = -\sum_{k:\ 1\cdots C} \frac{|\Omega_k \cap C_j|}{|C_j|} \log \frac{|\Omega_k \cap C_j|}{|C_j|} \quad (7)$$

where j is in $1 \cdots K$ and p_{kj} is the probability (maximum likelihood estimate) of a document being in both the cluster C_j and the class Ω_k. Equations (6) and (7) are equivalent. The total entropy of the clustering algorithm is the weighted sum of entropies of all clusters in a collection ζ.

$$H = \sum_{j:\ 1\cdots K} \frac{H_j * |C_j|}{N} \quad (8)$$

where N is the total number of documents in a collection ζ.

4.3 Results

We used the K-means clustering algorithm with $Norm - TF$, $TF - IDF$, CBT, and $TF - IDF - ICF$ term weighting schemes to identify the importance of using inter- and intra- cluster components in the term weights. Since the K-means algorithm is unstable and sensitive to initial centroids, we ran the algorithm for 10 times with different random seed for the initial centroids on each run. We repeated this experiment for the four term weighting schemes on the data collections listed in the Table 3.

We calculated the entropy for the four term weighting schemes, as given in Equation (8), for each run after the algorithm converged. Then, we computed the average of the entropies obtained in each run. From Table 4, we can see the average entropy calculated for each data sets. The average entropy measured for the K-means algorithm with the CBT term weighting scheme have shown better results compared to the other term weighting schemes on each data set.

According to the Cluster-Based Term weighting scheme, a term is considered important to a cluster if it is unique to that cluster and occurs most frequently within the documents of that cluster. The inter- and intra-cluster components try to identify these important terms by analyzing the term frequency distribution at three levels: document, cluster and collection. And our experimental results have shown that adding these cluster components in the term weighting

Table 4: Average Entropy measured for each data sets.

	Term weighting schemes			
Data sets	$Norm-TF$	$TF-IDF$	CBT	$TF-IDF-ICF$
TR11	0.8413	0.7905	0.8535	**0.7749**
TR12	0.9009	0.6834	**0.6139**	0.6261
TR23	1.0424	0.9246	**0.8390**	0.8501
TR31	**0.9379**	1.2781	0.9657	1.0822
TR45	1.0787	1.3469	**1.0443**	1.1485
20 NG S1	2.4824	0.4037	**0.2995**	0.4475
20 NG S2	2.4954	0.5791	**0.4164**	0.8010
20 NG S3	2.4727	0.9366	**0.5082**	0.7886
20 NG S4	2.4923	0.3138	**0.2845**	0.5210
20 NG S5	2.7464	**0.2983**	0.3274	0.5665
RE S1	**2.0103**	2.0638	**2.0116**	2.0600
RE S2	1.7281	1.7369	1.6810	**1.6711**

scheme significantly improves the average entropy results on each data set. We believe that some of the deviations in the results are due to the K-means clustering algorithm's lack of handling the noise in the data collection. The better average entropy result in each data set is boldfaced.

5. Summary

In this paper, we proposed a new term weight scheme and investigated its use in document clustering algorithms. We introduced two new cluster components for our term weighting scheme. We have demonstrated how these cluster components in addition to the term and collection frequency components in our term weighting scheme improves the average entropy result of the K-means algorithm. Finally, we compared our term weighting scheme (CBT) with the other three term weighting shemes ($TF-IDF$, $Norm-TF$, and $TF-IDF-ICF$) using the average entropy calculated from the K-means clustering algorithm.

References

[1] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. 513–523, 1988.

[2] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[3] K. Spark Jones and P. Willett, *Readings in information retrieval*. Morgan Kaufmann, 1997, ch. 3, pp. 305–312.

[4] H. Ayad and M. S. Kamel, "Topic Discovery from Text Using Aggregation of Different Clustering Methods," in *Proceedings of the 15th Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence*, May 2002, pp. 161–175.

[5] J. Reed, Y. Jiao, T. Potok, B. Klump, M. Elmore, and A. Hurson, "TF-ICF: A New Term Weighting Scheme for Clustering Dynamic Data Streams," in *2006 5th International Conference on Machine Learning and Applications (ICMLA'06)*. IEEE, Dec. 2006, pp. 258–263.

[6] C. Zhang, H. Wang, Y. Liu, and H. Xu, "Document Clustering Description Extraction and Its Application," in *Proceedings of the 22nd International Conference on Computer Processing of Oriental Languages. Language Technology for the Knowledge-based Economy*, ser. ICCPOL '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 370–377.

[7] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge University Press, 2008, ch. 16, p. 496.

[8] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002.

[9] TREC, "Text REtrieval Conference (TREC)," 1999. [Online]. Available: http://trec.nist.gov/

[10] K. Lang, "20 Newsgroups Data set." [Online]. Available: http://people.csail.mit.edu/jrennie/20Newsgroups

[11] D. D. Lewis, "Reuters-21578 text categorization test collection," 1999. [Online]. Available: http://kdd.ics.uci.edu/

[12] X. Zhou, X. Zhang, and X. Hu, "Dragon Toolkit: Incorporating Auto-Learned Semantic Knowledge into Large-Scale Text Retrieval and Mining," in *19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007)*. IEEE, Oct. 2007, pp. 197–201.

[13] S. Hettich and S. D. Bay, "Reuters Transcribed Subset," Irvine, 1999.

[14] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[15] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *KDD workshop on text mining*, vol. 400, no. X, Department of Computer Science and Engineering University of Minnesota. Citeseer, 2000, pp. 525–526.

[16] F. Beil, M. Ester, and X. Xu, "Frequent term-based text clustering," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '02. New York, NY, USA: ACM, 2002, pp. 436–442.

[17] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, vol. 27, no. 4, pp. 379–423, 1948.

A new approach to present prototypes in clustering of time series

Saeed R. Aghabozorgi, Teh Y. Wah, Amineh Amini, and Mahmud R. Saybani

***Abstract*—There are considerable advances in clustering time series data in data mining concept. However, most of which use traditional approaches and try to customize the algorithms to be compatible with time series data. One of the significant problems with traditional clustering is defining prototype specially in partitional clustering where it needs centroids as representative of each cluster. In this paper we present a novel effective approach to define the prototypes based on time series nature. The prototype is constructed based on fuzzy concept efficiently. Moreover, it is demonstrated how the prototypes are moved in iterations. We will present the benefits of the proposed prototype by implementing a real application: Customer transactions clustering.**

I. Introduction

There are different approaches to analyze time series data, which clustering is one of the most frequently used techniques [1], owing to its exploratory nature, and its application as a pre-processing phase in more complex data mining algorithms. There are variety of studies, projects and surveys that have noted different approaches and comparative respects of time series clustering [2-13]. Clustering of time series data has three overall problems which do not exist in traditional clustering algorithm (static objects clustering): representation method, distance measurement and clustering algorithm. Choosing a proper approach to represent time series data as a low dimension data is the problem statement of many papers. Another respect of time series clustering is finding an adequate distance measurement between time series data, whether between raw time series or dimensionality reduced time series. Finally, choosing an accurate and fast clustering algorithm, compatible with time series data is a challenge for some researches.

In order to compare time series with irregular sampling intervals and length, it is of great significance to adequately determine the similarity of time series. There is different distance measurements designed for specifying similarity between time series. The Hausdorff distance and modified Hausdorff (MODH), Euclidean distance, HMM-based distance, dynamic time warping (DTW), Euclidean distance in a PCA subspace, and longest common subsequence (LCSS) are the most popular distance measurement methods used for time series data. Zhang et al. [14] has performed a complete survey on the aforementioned distance measurements comparing them in different applications.

Dynamic time warping (DTW) [15, 16] is one of the most famous algorithms for measuring similarity between two sequences with irregular-lengths without any discretization which can be different in terms of time or speed. In DTW method, the sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. It makes two pairs of the nearest points from each sequence, allowing a one-to-many matching. However, the comparison in DTW does not perform a structural comparison of the time series because the comparison is based on the local dissimilarity. Another roughly similar measurement is LCSS (Longest Common Sub-Sequence) which is useful especially for unequal length data, and it is more robust to noise and outliers than DTW because all points do not need to be matched. In LCSS a point with no good match can be ignored to prevent unfair biasing. For aforementioned reasons LCSS is employed as distance measure for our methodology. However, defined prototypes for clusters are not based upon the measurement, whether DWT or LCSS is used as measurement. That is, if we use DWT or LCSS special measurements, it is not proper to utilize the average value (centroid) or median [17] as prototype of the cluster, because these kind of prototypes are based on Euclidean space. In this situation it is more accurate if a prototype is defined based on each used measurement. For example, in figure 1, the centroid (mean) is constructed based on the mean values of two time series. The results show that surprisingly, the final clusters using this prototype are not accurate enough as we show in experimental results.

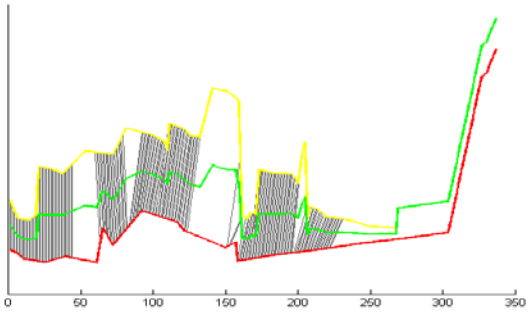


Fig. 1. Centroid (mean) prototype between two time series

Another approach is using mean value of match points when LCSS is used as measurement. Using this method also cannot solve the issue because the length of prototype is

S. R. Aghabozorgi is with University of Malaya, Department of Information Science, Faculty of Computer Science & Information Technology Building, University of Malaya, 50603 Kuala Lumpur, Malaysia (e-mail: saeed@siswa.um.edu.my).

Y. W. Teh The (e-mail: tehyw@um.edu.my) , A. Amini (e-mail: amineh@siswa.um.edu.my) and M. R. Saybani (e-mail: saybani@siswa.um.edu.my) are with University of Malaya, Department of Information Science, Faculty of Computer Science & Information Technology Building, University of Malaya, 50603 Kuala Lumpur, Malaysia

decreased when there are many time series belong to a cluster (Figure 2).

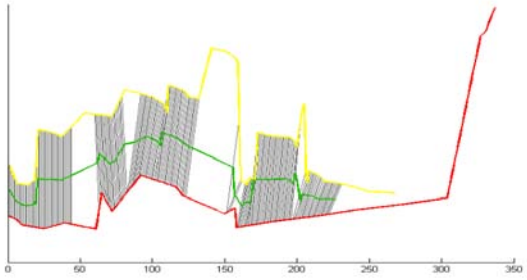


Fig. 2. Prototype based on match ponts

One of the recent affords in this area is [18] where authors formulated the prototype computation problem as a optimization task, and proposed an local search solution to solve it. They claim that this approach improves the *k*-medoids approach.

In this work, we present an optimal prototype based on LCSS measurement and then we show how using this prototype raises the accuracy of final result rather than using average or median approach.

The rest of this paper is organized as follows. In section 2, the terminology is described, and then in section 3 the methodology is presented. The algorithm is applied on real time series data sets and the experimental results are reported in sections 4. In section 5 the results are evaluated, and in section 6 conclusions and achievements are drawn.

II. TERMINOLOGY

We start by providing some basic notation and preliminary definitions.

Definition 1. Time series: A time series $F_i = \{f_1, .., f_t, .., f_T\}$ is a ordered set of flow vectors which indicate the spatiotemporal characteristics of moving objects at any time t of the total track life T [19]. A flow vector or feature vector $f_t = [X, Y, Z, ...]$ generally represents location and dynamics in the domain. However, we limit ourselves to just a spatial location $f_t = [X]$ in this work for the sake of simplicity. We assume $M = \{F_1, .., F_i, .., F_n\}$ is a collection of time series in a domain, where F_i represents i-th time series (i = 1,..,n) in the domain.

Definition 2. Similar time series: Two time series F_i and F_j are defined as similar if and only if $D(F_i, F_j) < \varepsilon$, where $D(F_i, F_j)$ is a function or process for calculating similarity between F_i and F_j, and ε is a specified threshold value [20].

A. Longest common Sub-sequence

In this section we shortly explain the LCSS employed as distance measure for our methodology.

Definition 3. Longest Common Sub-Sequence: Given F_i as a time series and f_t as feature vector at time t in time series F_i, if f_{qt} is the feature q-th of time series for $q = \{1, .., p\}$ at time t and if p is number of features describing each object, then the LCSS distance is defined as [21]:

$$LCSS(F_i, F_j) = \begin{cases} 0, & T_i = 0 \mid T_j = 0 \\ 1 + LCSS\left(F_i^{T_i - 1}, F_j^{T_j - 1}\right), & d_E(f_i, T_i, f_j, T_j) < \varepsilon \text{ and } |T_i - T_j| < \delta \\ \max\left(LCSS\left(F_i^{T_i - 1}, F_j^{T_j}\right), LCSS\left(F_i^{T_i}, F_j^{T_j - 1}\right)\right), & otherwise \end{cases} \quad (1)$$

Where the $LCSS(F_i, F_j)$ value states the number of matching points between two time series and $F_i = \{f_1, .., f_t\}$ specifies all the flow vectors in time series F_i up to time t. Additionally, in this formula, δ is an integer value which constricts the length of the warping and $0 < \varepsilon < 1$ is a real number as the spatial matching threshold to cover elements with real values. More precisely, ε is a tolerance threshold to find the set of flow vectors in a time series that are within distance ε from a point (flow vector) in another time series. The LCSS also has the ability of computing efficiently using dynamic programming like similar to what has been done with DTW.

In this paper, a customized distance measure is defined based on LCSS as:

$$D_{LCSS}(F_i, F_j) = 1 - \frac{LCSS(F_i, F_j)}{mean(T_i, T_j)} \quad (2)$$

Where, using $mean(T_i, T_j)$ instead of $min(T_i, T_j)$ results in taking the length of both time series into account.

B. Fuzzy C-Means (FCM) algorithm

One of the most extensively used clustering algorithms is the Fuzzy C-Means (FCM) algorithm presented by Bezdek [22]. FCM works by partitioning a collection of n vectors into c fuzzy groups and finds a cluster center in each group such that the cost function of dissimilarity measure is minimized. Bezdek introduced the idea of a "fuzzification parameter" (m) in the range [1, n] which determines the degree of fuzziness (weighted coefficient) in the clusters. Essentially, the parameter m controls the permeability of the cluster horizon which can be viewed as an n-dimensional cloud moving out from a cluster center [23].

Given c as number of classes, v_j, centre of class j for $j = \{1, .., c\}$, n as the number of time series and μ_{ij} as the degree of membership of the time series i to cluster j for $i = \{1, .., n\}$, distance of each time series F_i from each cluster center(v_j) is denoted as such:

$$d_{ij} = (F_i, v_j) = D_{LCSS}(F_i, v_j) \quad (3)$$

Let the centers be shown by $v_j = \{v_1, .., v_c\}$ and each time series by F_i that $i = \{1, .., n\}$ and d_{ji} as distances between centers and time series. Therefore, the membership values μ_{ij} are obtained with:

$$\mu_j(x_i) = \frac{\left(\frac{1}{d_{ji}}\right)^{\frac{1}{m-1}}}{\sum_{k=1}^{p}\left(\frac{1}{d_{ji}}\right)^{\frac{1}{m-1}}} \quad (4)$$

And the sum of cluster memberships for a time series equals 1:

$$\sum_{j=1}^{c} \mu_{ij}(F_i) = 1 \quad ,\forall i \in \{1,..,n\} \quad (5)$$

The FCM objective function (standard loss) that is attempted to be minimized takes the form:

$$J = \sum_{j=1}^{c} J_j = \sum_{j=1}^{c}\sum_{i=1}^{n} [\mu_{ij}]^m d_{ij} \quad (6)$$

where μ_{ij} is a numerical value between [0; 1]; d_{ij} is the Euclidian distance between the j_{th} prototype and the i_{th} time series; and m is the exponential weight which influences the degree of fuzziness of the membership matrix.

In different iterations, the membership values of the time series are calculated, and then the prototypes (cluster centers) are recomputed. In order to update a new cluster center value, the following formula is employed:

$$v_j = \frac{\sum_{i=1}^{n}(\mu_{ij})^m (F_i)}{\sum_{i=1}^{n}(\mu_{ij})^m} \quad \forall j \in \{1,..,c\} \quad (7)$$

III. Prototype Calculation Based on Existing Time Series Inside a Clusters

In this methodology each time series is assigned to a cluster, whose prototype (centroid) is the nearest. The prototype of a cluster has to be constructed in such a way that:

1. The prototype has to be changed based on changes of time series inside the cluster
2. Time series inside a cluster should have most similarity to their cluster's prototype

This problem is break down in two sub-problems:

A. Making a centre for a group of existing time series inside a cluster
B. Moving existing centers based on time series inside the clusters

A. *Defining a prototype for a group of time series*

In order to construct a prototype based on exist time series in a cluster, a new time series is defined as prototype of each cluster. In this step, we use the Shortest Common Super Sequence (SCSS) to make the prototype.

Definition 4. Shortest Common Super Sequence: Given two sequences $F_i =< f_{i1},..,f_{it},..,f_{im} >$ and $F_j =< f_{j1},..,f_{jt},..,f_{jn} >$, a sequence $U_k = < u_{k1},..,u_{kt},..,u_{km} >$ is a common super sequence of F_i and F_j, if U_k is a super sequence of both F_i and F_j . The SCSS is a common super sequence of minimal length.

In the SCSS problem, the two sequences F_i and F_j are given and the target is to find the shortest possible common super sequence of these sequences. In general, the SCSS is not unique. The SCSS problem is closely related to the Longest Common Sub-Sequence (LCSS) problem. That is, for two input sequences, an SCSS can be formed from an LCSS easily.

In the first step all dimensionally reduced time series are normalized. In the next figures three original (Figure 3) and normalized (Figure 4) time series inside a cluster are depicted.

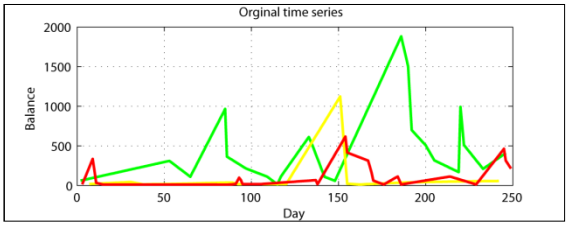


Fig. 3. Raw time series before normalization

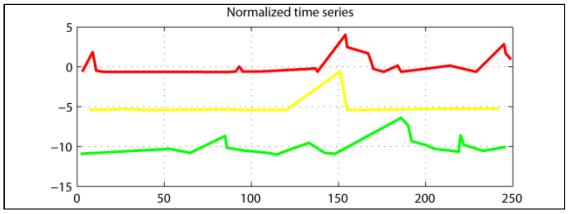


Fig. 4. Normalized time series

In order to find SCSS among time series, the LCSS among the time series are used. The following pictures illustrate the match points (Figure 5) among two time series calculated by dynamic programming.

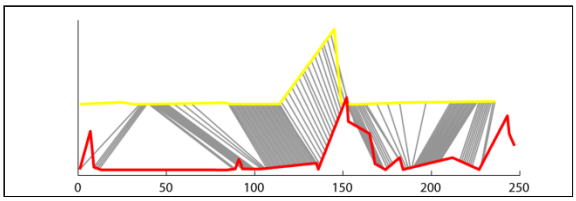


Fig. 1. Match points between time series based on LCSS

In order to make the prototype based on the clusters' members, a novel algorithm is presented in table 1. In this algorithm, cluster C is denoted as $C = \{F_1,..,F_i,..,F_n\}$ and F_i as a member of cluster C. For each $F_i = \{f_{i1},..,f_{it},..,f_{ip}\}$, f_{it} is a point of time series F_i and $LCSS(F_i, F_j)_{x*2}$ is the longest common sub sequence of time series F_i and F_j and x is the total number of common points between them.

TABLE 1
PSEUDO CODE FOR PROTOTYPE CALCULATION BASED ON THE LONGEST COMMON SUB SEQUENCE.

Initialize n=number of time series
Initialize U with $P_{k,1}$=LCSS $(F_1,F_2)_{k,1}$, $P_{k,2}$=LCSS $(F_1,F_2)_{k,2}$, $1 < k < length(\text{LCSS}(F1, F2))$
  for all pair time series i and j (i <j) :
    x←1; y←1;
    while there are unvisited rows in $\text{LCSS}(F_i, F_j)$ do
      initial a new pair Q_x= < $Q_{x,i}$, $Q_{x,j}$ > where $Q_{x,i}$ = $\text{LCSS}(F_i, F_j)_{x,1}$; $Q_{x,j}$= $\text{LCSS}(F_x, F_y)_{x,2}$;
      Add Q_x to U in such a way that:
      If there is not any pair include $P_{y,i}$ in P_y
        increase y;
      elseif $Q_{x,i}$< $P_{y,i}$ then
        insert Q_x before P_x in U; increase x,y;
      elseif $Q_{x,i}$ == $P_{y,i}$ then
        $P_{y,j}$= $Q_{x,j}$;
        increase x,y;
        elseif $Q_{x,i}$>$P_{y,i}$ then
        increase y;
      end if
    end while
end for
for each P_j in U
  $V_j = mean\left(f_{p_{j,x}}, f_{p_{j,y}}\right)\ \forall x, y\ which\ \ < p_{j,x}, p_{j,j} > \in P_j$
end for
return V_j as prototype of cluster O

In this algorithm, we define an ordered set $U =< P_1, \dots, P_k, \dots >$ for showing the match points' indices which construct the prototype. The P_k is defined as a none-ordered set as $P_k =< (p_{k,1}, p_{k,2}), \dots, (p_{k,i}, p_{k,j}), \dots >$ which includes a set of pair points of time series i and j which construct the k-th point of prototype.

E.g. $P_3 =< (i2, j2)(i2, k4)(j2, k2) >$ denotes that P_3 is made from second points of time series i, j ,k ,and the fourth point of k.

At first a common super sequence is made based on the LCSS of each pair of cluster members. Then intermediate points between each pair of time series are considered in order to turn the common time series into a SCSS. This time series is denoted as the SCSS of all match points among the time series in the same cluster. Based on this definition, the prototype can be regarded as the shortest sequence that includes all LCSSs among the time series within the cluster. Then the prototype of cluster j is defined as:

$$V_j = SCSS(F_1, \dots, F_i, \dots, F_z),\quad \forall i \in \{memberance\ of\ cluster\ j\}, \forall j \in \{1, \dots, c\} \qquad (7)$$

and the value of each point of V_j is calculated by the average of the value of each common pair points in the SCSS. For the sake of simplicity, an example of a prototype calculated for a cluster with three sample time series is provided in table 2.

TABLE 2
EXAMPLE OF CALCULATING OF PROTOTYPE OF A CLUSTER BASED ON THE LONGEST COMMON SUB SEQUENCE

If we consider F_i, F_j and F_k as example time series within a cluster,
F_i=<f_{i1},..,f_{i7}>=<2.1, 3.2, 1.1, 2.7, 4.8, 2.9, 1.5>
F_j=<f_{j1},..,f_{j5}>=<2.8, 3.3, 2.8, 2.2,1.4>
F_k=<f_{k1},..,f_{k9}>=<2.7, 3.2, 2.2, 3.2, 2.8, 2.8, 2.3, 1.5, 1.6>
and, if LCSS of each pairs of the time series is assumed as follow:

$$LCSS(F_i, F_j) = \begin{bmatrix} i2 & j2 \\ i4 & j3 \\ i7 & j5 \end{bmatrix}, \qquad LCSS(F_i, F_k) = \begin{bmatrix} i1 & k3 \\ i2 & k4 \\ i6 & k5 \\ i7 & k9 \end{bmatrix},$$

$$LCSS(F_j, F_k) = \begin{bmatrix} j1 & k1 \\ j2 & k2 \\ j3 & k5 \\ j4 & k7 \\ j5 & k8 \end{bmatrix}$$

Then, the prototype indexes is calculated in three steps as:
(1*) the U_{y*n} is defined based on first common piecewise (F_i) between $LCSS(F_i, F_j)$ and $LCSS(F_i, F_k)$
(2*) Updating based on second common piecewise(F_j) between $LCSS(F_i, F_j)$ and $LCSS(F_j, F_k)$

$$(1^*) = \begin{bmatrix} i1 & - & k3 \\ i2 & j2 & k4 \\ i4 & j3 & - \\ i6 & - & k5 \\ i7 & j5 & k9 \end{bmatrix} \rightarrow (2^*) = \begin{bmatrix} i1 & - & k3 \\ - & j1 & k1 \\ i2 & j2 & k4, k2 \\ i4 & j3 & k5 \\ i6 & - & k5 \\ - & j4 & k7 \\ i7 & j5 & k9, k8 \end{bmatrix}$$

(3*) common piecewise among F_i, F_j and F_k

$$(3^*) = SCSS(F_i, F_j, F_k) = \begin{bmatrix} (i1, k3) \\ (j1, k1) \\ (i2, j2)(i2, k4)(j2, k2) \\ (i4, j3)(j3, k5) \\ (i6, k5) \\ (j4, k7) \\ (i7, j5)(i7, k9)(j5, k8) \end{bmatrix}$$

(4*)Value of prototype based on common piecewise among F_i, F_i and F_k:

$$(4^*) = V_j = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix} = \begin{bmatrix} mean(f_{i1}, f_{k3}) \\ mean(f_{j1}, f_{k1}) \\ mean(f_{i2}, f_{i2}, f_{j2}, f_{j2}, f_{k4}, f_{k2}) \\ mean(f_{i4}, f_{j3}, f_{j3}, f_{k5}) \\ mean(f_{i6}, f_{k5}) \\ mean(f_{j4}, f_{k7}) \\ mean(f_{i7}, f_{i7}, f_{j5}, f_{j5}, f_{k9}, f_{k8}) \end{bmatrix}$$

$$= \begin{bmatrix} \frac{2.1 + 2.2}{2} \\ \frac{2.8 + 2.7}{2} \\ \frac{3.2 + 3.3 + 3.2 + 3.2 + 3.3 + 3.2}{6} \\ \frac{2.7 + 2.8 + 2.8 + 2.8}{4} \\ \frac{2.9 + 2.8}{2} \\ \frac{2.2 + 2.3}{2} \\ \frac{1.5 + 1.4 + 1.5 + 1.6 + 1.4 + 1.5}{4} \end{bmatrix} = \begin{bmatrix} 2.15 \\ 2.75 \\ 3.23 \\ 2.77 \\ 2.85 \\ 2.25 \\ 1.48 \end{bmatrix}$$

The figure 4 illustrates calculated prototype for a cluster include three time series presented in figure 3.

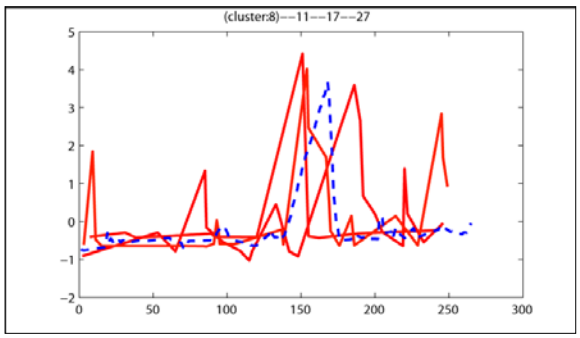


Fig. 5. The calculated prototype of normalized time series

B. *Moving prototypes*

In order to move the prototypes, the prototypes to be updated according to the membership of all objects. Similarly, the fuzziness of the time series is utilized in order to update prototypes in this approach. In table 3, a new algorithm is presented to explain the details of updating prototypes. In this algorithm, $F_i = \{f_{i1}, .., f_{it}, .., f_{ip}\}$ is a time series with length p, and f_{it} is a point of the time series F_i at time t, and matrix $LCSS(F_i, V_j)_{r*2}$ indicates match points (LCSS) of time series Fi and prototype Vj, where $LCSS(F_i, V_j)$ has a dimension of r×2 (r is the number of match points)

We define an ordered set $U =< P_1, \dots, P_k >$ correspond to V_j' as prototype. The P_k is defined as a none-ordered set as $P_k =< p_{k,1}, p_{k,2}, .., p_{k,b} >$, $b < n$ which includes a set of index of points of time series (which have an specific condition explained further). E.g. $P_3 =< i7, y3, z6 >$ that indicates the f_{i7} point of time series F_i and f_{y3} point of time series F_y and so on.

TABLE 3
PSEUDO CODE TO UPDATE PROTOTYPE BASED ON TIME SERIES INSIDE A CLUSTER

initialize $\mu_{min} = \frac{1}{c}$ (determine a threshold equal to the inverse of the class number for c as number of clusters)
initialize $\mu_{mean,j} = \frac{\sum_{i=1}^{z}[\mu_{ij}]}{z}$ $\forall i \in \{members\ of\ cluster\ j\}, z = number\ of\ members$,
initialize a set U, length=length (V_j)
for each time series F_i with $\mu_{ij} > \mu_{min}$;
  for each pair of match point in $LCSS(F_i, V_j)$, r: the row number in matrix $LCSS(F_i, V_j)$
    initialize $P_{LCSS(F_i,V_j)_{r,1},i} = LCSS(F_i, V_j)_{r,2}$;
  end for
end for
/// extending the prototype
for every time series F_i that $\mu_{ij} > \mu_{mean}$;
 h←1; r←1
 while r <= length(F_i)
  initialize a new pair Q as Q=< f_{il} , μ_{ij}>;
  initialize set Q_x= < $Q_{x,1}$,.., $Q_{x,n}$ > $Q_{x,i}$ = r;
  read P_h from U
  if r <$P_{h,i}$
    insert P before P_h in U;
    increase h,r;
  elseif r == $P_{h,i}$
    increase h,r;
  elseif r > U_{hi}
    increment h;
 end
end
for each P_h in v
  $V_{jt}' = \frac{\sum_{i=1}^{b}(\mu_{ij})^m(f_{ix})}{\sum_{i=1}^{b}(\mu_{ij})^m}$, b=length(P_h)
end if

In the mentioned algorithm above, a threshold μ_{min} is defined and then prototypes are updated based on the time series with memberships more than μ_{min}(candidate time series). This threshold is required in order to ignore the noise by omitting time series with a lower fuzziness value. Additionally, it prevents prototypes from stretching incrementally over time. μ_{min} equals to the inverse of the class number value:

$$\mu_{min} = \frac{1}{c} \tag{8}$$

In accordance with the definition, only a specific part of time series (candidate time series) is considered in the calculation of prototypes. For each candidate time series, corresponding points must be found, that is, match points (LCSS) between time series and prototypes. The set U of the prototype is initialized according to following equation:

$$P_{LCSS(F_i,V_j)_{r,1},i} = LCSS(F_i, V_j)_{r,2} \tag{9}$$

where $P_{LCSS(F_i,V_j)_{r,1},i}$ indicates the index of common points (match points) between all of the time series assigned to a cluster with an acceptable membership.

Until now, only the common points among all candidate time series have been considered. In the next step, the U set is updated based on some parts of candidate time series which, although they do not have match points with the prototype, they have higher memberships than $\mu_{mean,j}$ ($\mu_{ij} > \mu_{mean,j}$). These points are inserted between common points of U in order to take only sub-sequences that have acceptable membership into account. For cluster j with z members, $\mu_{mean,j}$ is calculated as:

$$\mu_{mean,j} = \frac{\sum_{i=1}^{z}[\mu_{ij}]}{z} \quad \forall i \in \{\text{ members of cluster j }\}, \tag{10}$$
$$z = \text{number of members of cluster j}$$

In corresponding with matrix U, matrix S stores the value of each point f_{it} and its membership μ_{ij}. That is, for each record of U (each point of main centre), there are a set of point values and their membership's value in S. Let t be one of these points in U. Now if point t of the new center is matched with h different points with value X_i and membership μ_{ij} from h different time series, point t of the updated prototype V_j' is shown as:

$$V_{jt}^{'} = \frac{\sum_{i=1}^{b}(\mu_{ij})^{m}(f_{ix})}{\sum_{i=1}^{h}(\mu_{ij})^{m}} \quad \forall i \in \{1,\dots,h\}, \quad \forall t \in \{1,\dots,n\} \qquad (11)$$

Table 4 shows an example of selecting candidate time series among four time series and then updating prototypes based on $\mu_{mean,j}$

TABLE 4
EXAMPLE OF UPDATING PROTOTYPE BASED ON EXISTING TIME SERIES IN A CLUSTER

Given four assumed time series, candidate time series are declared for cluster j as:

Time series:	μ_{min}	$\mu_{mean,j}$
F_i=<$f_{i1},..,f_{i7}$>	$\mu_{i,j} > \mu_{min}$	$\mu_{i,j} > \mu_{mean,j}$
F_Y=<$f_{y1},..,f_{y5}$>	$\mu_{y,j} > \mu_{min}$	$\mu_{y,j} > \mu_{mean,j}$
F_x=<$f_{x1},..,f_{x7}$>	$\mu_{x,j} < \mu_{min}$	$\mu_{x,j} < \mu_{mean,j}$
F_Z=<$f_{z1},..,f_{z9}$>	$\mu_{z,j} > \mu_{min}$	$\mu_{z,j} < \mu_{mean,j}$

The LCSS between each pairs of time series are assumed as :

$$LCSS(F_i, V_j) = \begin{bmatrix} i2 & v2 \\ i4 & v3 \\ i7 & v5 \end{bmatrix}, \quad LCSS(F_y, V_j) = \begin{bmatrix} y1 & v1 \\ y2 & v2 \\ y3 & v5 \\ y4 & v7 \end{bmatrix}$$

$$LCSS(F_z, V_j) = \begin{bmatrix} z1 & v3 \\ z2 & v4 \\ z6 & v5 \end{bmatrix}, \quad LCSS(F_x, V_j) = \begin{bmatrix} x1 & v1 \\ x3 & v2 \\ x7 & v5 \\ x8 & v6 \end{bmatrix}$$

(1*) step 1: Find common points between V and time series with membership more than μ_{min} ,in this case:(F_i, F_y, F_z)

(2*) step 2: Extend the matrix U with some parts of time series that has membership more than $\mu_{mean,j}$ (F_i, F_y)

$$V_j = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ . \\ . \\ v_n \end{bmatrix} \quad U = (1^*) \begin{bmatrix} - & - & - \\ - & y1 & - \\ i2 & y2 & - \\ - & - & - \\ i4 & - & z1 \\ - & - & z2 \\ - & - & - \\ - & - & - \\ - & - & - \\ - & - & - \\ i7 & y3 & z6 \\ - & y4 & - \\ . & . & . \\ . & . & . \end{bmatrix} \rightarrow (2^*) \begin{bmatrix} i1 & - & - \\ - & y1 & - \\ i2 & y2 & - \\ i3 & - & - \\ i4 & - & z1 \\ - & - & z2 \\ i5 & - & - \\ i6 & - & - \\ - & - & z3 \\ - & - & z4 \\ i7 & y3 & z6 \\ - & y4 & - \\ . & . & . \\ . & . & . \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} i1 \\ y1 \\ i2, y2 \\ i3 \\ i4, z1 \\ z2 \\ i5 \\ i6 \\ z3 \\ z4 \\ i7, y3, z6. \\ y4 \\ . \\ . \end{bmatrix} \begin{bmatrix} \mu_{i,j} \\ \mu_{y,j} \\ \mu_{i,j}, \mu_{y,j} \\ \mu_{i,j} \\ \mu_{i,j}, \mu_{z,j} \\ \mu_{i,j} \\ \mu_{i,j} \\ \mu_{z,j} \\ \mu_{z,j} \\ \mu_{y,j}, \mu_{z,j}, \mu_{i,j} \\ \mu_{y,j} \\ \mu_{z,j} \\ . \\ . \end{bmatrix} \rightarrow V_j^{'} = \begin{bmatrix} v_1^{'} \\ v_2^{'} \\ v_3^{'} \\ v_4^{'} \\ v_5^{'} \\ . \\ . \\ . \\ . \\ v_n^{'} \end{bmatrix}$$

IV. EXPERIMENTAL RESULTS

In this paper, we focus on segmentation of bank customers as identifiable objects to show how we utilize the presented method for clustering of customers.

For a bank in Malaysia, similar customers based on their daily transactions are desirable. The profile of similar customer is used for decision making, fraud detection, campaigns, etc. In order to find accurate similar transactions on all accounts, we try to cluster cards based on their balance on each day. We have applied fuzzy clustering algorithm on different cardinality of dataset of the customer time series databases, but with the proposed prototype approach.

Our dataset is a collection of time series which includes 365 days of outstanding amount of 10,000 credit cards. Each time series in this dataset is presented by 200 to 365 observations.

V. EVALUATION

We call our prototype calculation approach FPT (Fuzzy Prototype of Time series), and compare its accuracy with different clustering algorithms.

clustering of time series is an unsupervised process and there are no predefined classes and no examples that can show that the clusters found by the clustering algorithms are valid [24]. Therefore, it is necessary to use some validity criteria. In order to prove that our approach is more efficient than utilized conventional prototypes, we employed the FCM algorithms with different prototypes (median, mean and FPT) for comparison purpose. We have collected different amount of records from our dataset to show how it works in terms of accuracy.

In order to evaluate clusters in terms of accuracy, we used Squared Error (SSE), the most common measure. For each time series, the error is the distance to the nearest cluster. To get SSE, we used following formula:

$$SSE = \sum_{j=1}^{c} \sum_{F_i \in C_j} (dist(F_i, V_j))^2 \qquad (12)$$

Where, F_i is a time series in cluster C_j and V_j is the prototype for cluster C_j.

The results illustrated in figure 10 and 11 are related to average of SSE for 10 times run of the FCM algorithms with three different approaches. The results show that the presented algorithm is competitive with other algorithms in terms of accuracy.

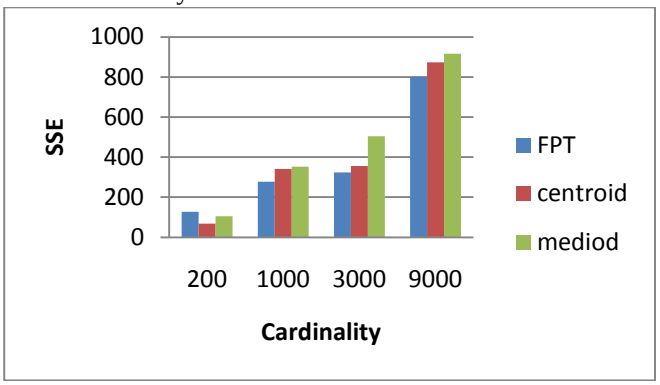


Fig. 6. Accuracy of using FPT in clustering crosses different cardinalities.

VI. Conclusion

The purpose of the current study was to present a method to present a prototype for time series clusters efficiently. We developed a novel method for constructing prototypes based on its ability to be accurate enough. The defined prototype can be updated based on fuzzy concept through iterations as well. We discussed about this fact that if the prototype for partitioning algorithms is computed precisely, the clusters is more accurate.

In order to show experimental results, PFT methodology was implemented on time series data of a bank to perform the segmentation. Moreover, we applied two more frequently used prototypes in clustering algorithms on our dataset to compare them with the developed approach (FPT) in terms of accuracy. The results of this study indicate that this method is much more efficient than conventional prototypes used in clustering algorithms. It is because of considering the common points of time series in clusters instead of whole data points.

However, further research needs to be done in order to evaluate FPT in terms of execution time and accuracy of data clusters in different datasets with different dimensions to understand its potentials and limitations.

VII. References

[1] M. Halkidi, *et al.*, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, pp. 107-145, 2001.

[2] P. Cotofrei and K. Stoffel, "Classification rules+ time= temporal rules," *Computational Science—ICCS 2002*, pp. 572-581, 2002.

[3] G. Das, *et al.*, "Rule discovery from time series," *Knowledge Discovery and Data Mining*, pp. 16–22, 1998.

[4] T. Fu, *et al.*, "Pattern discovery from stock time series using self-organizing maps," 2001, pp. 26-29.

[5] M. Gavrilov, *et al.*, "Mining the stock market: Which measure is best," 2000.

[6] X. Jin, *et al.*, "Indexing and mining of the local patterns in sequence database," *Intelligent Data Engineering and Automated Learning—IDEAL 2002*, pp. 39-52, 2002.

[7] E. Keogh and C. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, pp. 358-386, 2005.

[8] P. Tino, *et al.*, "Temporal pattern recognition in noisy non-stationary time series based on quantization into symbolic streams: Lessons learned from financial volatility trading," *Report Series for Adaptive Information Systems and Management in Economics and Management Science*, 2000.

[9] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: A survey and empirical demonstration," *Data Mining and Knowledge Discovery*, vol. 7, pp. 349-371, 2003.

[10] V. Kavitha and M. Punithavalli, "Clustering Time Series Data Stream-A Literature Survey," *Arxiv preprint arXiv:1005.4270*, 2010.

[11] T. Warren Liao, "Clustering of time series data--a survey," *Pattern Recognition*, vol. 38, pp. 1857-1874, 2005.

[12] H. Ding, *et al.*, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, vol. 1, pp. 1542-1552, 2008.

[13] S. Hirano and S. Tsumoto, "Empirical comparison of clustering methods for long time-series databases," *Active Mining*, pp. 268-286, 2005.

[14] Z. Zhang, *et al.*, "Comparison of Similarity Measures for Trajectory Clustering in Outdoor Surveillance Scenes," presented at the Proceedings of the 18th International Conference on Pattern Recognition - Volume 03, 2006.

[15] D. Sankoff and J. Kruskal, "Time warps, string edits, and macromolecules: the theory and practice of sequence comparison," 1983.

[16] S. Chu, *et al.*, "Iterative deepening dynamic time warping for time series," 2002.

[17] V. Vuori and J. Laaksonen, "A comparison of techniques for automatic clustering of handwritten characters," *Pattern Recognition*, vol. 3, p. 30168, 2002.

[18] V. Hautamaki, *et al.*, "Time-series clustering by approximate prototypes," 2008, pp. 1-4.

[19] B. Morris and M. Trivedi, "Learning trajectory patterns by clustering: Experimental studies and comparative evaluation," 2009, pp. 312-319.

[20] C. Lai, *et al.*, "A novel two-level clustering method for time series data analysis," *Expert Systems with Applications*, vol. 37, pp. 6319-6326, 2010.

[21] M. Vlachos, *et al.*, "Discovering Similar Multidimensional Trajectories," presented at the Proceedings of the 18th International Conference on Data Engineering, 2002.

[22] J. Bezdek, "Fuzzy Mathematics in Pattern Classification," Cornell University, Ithaca, 1973.

[23] E. Cox, *Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration*, 2008.

[24] M. Halkidi, *et al.*, "Clustering algorithms and validity measures," 2002, pp. 3-22.

SESSION

REGRESSION, CLASSIFICATION

Chair(s)

Wolfram-M. Lippe
Robert Stahlbock

Threshold Value Based Traffic Congestion Identification Method

Zhanquan Sun, Weidong Gu, Jinqiao Feng, Xiaomin Zhu
Key Laboratory for Computer Network of Shandong Province, Shandong Computer Science Center, Jinan, Shandong, 25014, China.

Abstract - *Traffic congestion identification is a popular research topic of Intelligent Transportation System (ITS). Many identification methods, such as threshold value based methods, California, McMaster method and so on, have been studied. But the threshold values of these methods are difficult to be determined. A novel threshold value based traffic congestion identification method is proposed in this paper. In the method, traffic flow parameters are divided into sections according to threshold values that are determineded with mutual information maximization theory. Congestion identification rules are extracted with decision tree. At last, the efficiency of the proposed method is illustrated through analyzing Jinan urban transportation data.*

Keywords: Traffic congestion identification; Threshold value; Mutual information; Decision tree; Generic algorithm

1 Introduction

Traffic congestion has become one of the most major and costly problems in the world, especially in many big cities and metropolitan areas. Many efforts have been done to reduce the impact of traffic congestion. The most efficient means is to apply advanced technologies to traffic management field, such as sensor, communication, and compute processing etc. technologies [1]. Traffic flow data can be detected automatically with these technologies. For enhancing the capability of decision-making, it is significant to get the traffic congestion state according to the real-time traffic flow data. Traffic congestion is straightforward and easy to be understood. Traffic congestion identification is an important component in the design and deployment of advanced transportation management systems and advanced traveler information systems [2]. It has become one of the major issues in most countries and has been widely studied. Traffic flow parameter based detection methods have been widely accepted in that they are not affected by weather circumstance and can be performed automatically [3]. Many identification methods based on traffic flow parameters have been studied. Dudek, Messer and Nuckles developed the california method in 1974 , which has been widely applied in traffic congestion identification and incident detection. It is the algorithm mostly used as a basis of comparison [4]. Persaud developed McMaster incident detection algorithm method in 1990. These methods are convenient to be used in practice. Traffic congestion identification rules can be extracted from them. They are widely applied to traffic congestion identification and incident detection. But how to define the threshold values of these methods is a difficult problem. They are often prescribed subjectively according to experience [5]. With the development of artificial intelligent, many machine learning methods are adopted to resolve traffic congestion identification problem. Payne (1978) used decision tree to detect freeway incident [6]. But it dose not consider how to discretize the traffic flow parameters. Zhang (2006) uses SOM method to orgnize flow data of links into physically relevant clusters with each cluster corresponds to a kind of traffic state [7]. In ref. [8], BP neural network is used to identify the traffic congestion state. Porikli and Li (2004) use Hidden Markov Model to estimate traffic congestion with MPEG video data[9]. Zhang (2006) used Wavelet method to identify traffic congestion state [10]. These artificial methods are not easy to be used in practice.

A novel threshold value based traffic congestion identification method is proposed in this paper. In the method, mutual information maximization principle is used to determine the threshold values, i.e. the mutual information between the discretized traffic parameters and traffic state should be maximum[11]. Mutual information based on Shannon entropy can measure arbitrary statistical correlation between variables[12]. It is used to measure the correlations between traffic flow parameters and traffic state. Traffic flow parameters, such as flow volume, speed, and occupancy, can be discretized according to the determined threshold values. Then how to extract traffic congestion identification rules with these discrete traffic flow parameters is an important task. Decision tree is a powerful and popular tool for classification[13]. It is used to identify traffic congestion and extract the identification rules.

The scheme of the paper is organized as follows. In part two, traffic flow parameters' discretization method based on mutual information maximization is presented in detail. In part three, traffic congestion identification rules extraction method based on decision tree is introduced. In part four, traffic congestion identification procedure is summarized. In

part five, an example is analyzed with the proposed identification method. In the last part, a conclusion is summarized.

2 Threshold value determination

Threshold values are determined through maximizing the mutual information between traffic parameters and traffic state. Traffic parameters, such as volume, speed, and occupancy and so on, are taken as continuous variables. Their probability distributions can be determined according to historical traffic flow data. The estimation of continuous probability distribution and the calculation of mutual information are described in detail as follows.

2.1 Maximum Likelihood Estimation

Maximum likelihood method is used to estimate the probability distribution of continuous variable. The method is sumamrized as follows. Let $\boldsymbol{X}$ be a continuous random variable with probability density function $p(\boldsymbol{x};\theta_1,\theta_2,\cdots,\theta_k)$, where $\theta_1,\theta_2,\cdots,\theta_k$ are k unknown constant parameters to be estimated. Given N independent observation samples $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N$ of the variable $\boldsymbol{X}$. The likelihood function is given by

$$\begin{aligned} L &= L(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N \mid \theta_1,\theta_2,\cdots,\theta_k) \\ &= \prod_{i=1}^{N} p(\boldsymbol{x}_i;\theta_1,\theta_2,\cdots,\theta_k) \end{aligned} \tag{1}$$

The logarithmic likelihood function is

$$\Lambda = \ln L = \sum_{i=1}^{N} \ln p(\boldsymbol{x}_i;\theta_1,\theta_2,\cdots,\theta_k) \tag{2}$$

The maximum likelihood estimators of $\theta_1,\theta_2,\cdots,\theta_k$ are estimated through maximizing L. They are the simultaneous solutions of following k equations

$$\frac{\partial(\Lambda)}{\partial\theta_i} = 0, i = 1,2,\cdots,k \tag{3}$$

Traffic flow parameters usually obey normal probability distritubtion i.e.

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(x-\sigma)^2}{2\sigma^2}\right\} \tag{4}$$

Then the maximum likelihood estimation of mean value μ and covariance σ are

$$\hat{\mu} = \frac{1}{N}\sum_{k=1}^{N}\boldsymbol{x}_k \tag{5}$$

$$\hat{\sigma} = \frac{1}{N}\sum_{k=1}^{N}(\boldsymbol{x}_k - \hat{\mu})^2 \tag{6}$$

After parameters being estimated, the distributions of traffic flow parameters are determined.

2.2 Mutual Information

When the probability distritubtions of traffic flow parameters are determined, the probability distributions of the discretized traffic flow parameters can be calculated according to the continuous probability densities. Let traffic flow parameter X be divided into m segments and each segment's threshold value be $x_1,x_2,\cdots,x_{m+1}$ respectively. The discretized traffic flow parameter is denoted by Z and each section $[x_i,x_{i+1}]$ corresponds to a discrete value $z_i, i = 1,2,\cdots,m$. The probability of discrete variable Z can be calculated as follows.

$$P_Z(Z = z_i) = P_X(X \le x_{i+1}) - P_X(X < x_i) \tag{6}$$

where $P_Z(\cdot)$ denotes the probability of variable Z and $P_X(\cdot)$ denotes the probability density of variable X.

Let traffic state be denoted by discrete variable Y. The mutual information between discretized traffic flow parameters and traffic state can be calculated as follows.

The Shannon entropy of discrete variable Y is described as

$$H(Y) = -\sum_{i=1}^{l} p(y_i)\log p(y_i) \tag{7}$$

where $y_i, i = 1,2,\cdots,l$ are possible values of variable Y.

Let $p(y,z)$ denote the joint probability of variable Y and Z. The joint entropy of variable Y and Z is described as

$$H(Y,Z) = -\sum_{i=1}^{l}\sum_{j=1}^{m} p(y_i,z_j)\log p(y_i,z_j) \tag{8}$$

The entropy of Z conditional on the variable Y taking a certain value y_j is

$$H(Z \mid y_j) = -\sum_{i=1}^{m} p(z_i \mid y_j)\log p(z_i \mid y_j) \tag{9}$$

It denotes the remaining uncertainty of variable Y given that the value of varaible Z is known.

Mutual information between Y and Z is described as

$$\begin{aligned} I(Z;Y) &= H(Z) - H(Z \mid Y) \\ &= \sum_{i=1}^{l}\sum_{j=1}^{m} p(y_i, z_j) \log \frac{p(y_i, z_j)}{p(y_i)\,p(z_j)} \end{aligned} \tag{10}$$

Mutual information is a quantity that measures the mutual dependence of the two variables. It can measure arbitrary statistical correlation between variables. The mutual information between traffic state Y and discretized traffic flow parameter Z can be calculated as follows.

The Shannon entropy of variable Z is

$$\begin{aligned} H(Z) &= -\sum_{i=1}^{m} P(z_i) \log P(z_i) \\ &= -\sum_{i=1}^{m} \int_{x_i}^{x_{i+1}} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{(x-\mu)^2}{2\sigma^2}\right) \mathrm{d}x \log \int_{x_i}^{x_{i+1}} \exp\left(\frac{(x-\mu)^2}{2\sigma^2}\right) \frac{1}{\sqrt{2\pi}\sigma} \mathrm{d}x \\ &= -\sum_{i=1}^{m} (\Phi(x_{i+1}) - \Phi(x_i)) \log(\Phi(x_{i+1}) - \Phi(x_i)) \end{aligned} \tag{11}$$

where $\Phi(\cdot)$ denotes integral value of normal distribution, i.e. $\Phi(\cdot) = \int_{\Omega} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{(x-\mu)^2}{2\sigma^2}\right) dx$.

The conditional Shannon entropy of variable Z is

$$\begin{aligned} H(Z \mid Y) &= -\sum_{j=1}^{l} P(y_j) \sum_{i=1}^{m} P(z_i \mid y_j) \log P(z_i \mid y_j) \\ &= -\sum_{j=1}^{l} P(y_j) \sum_{i=1}^{m} \int_{x_i}^{x_{i+1}} \frac{1}{\sqrt{2\pi}\sigma_{X|y_j}} \exp\left(\frac{(x-\mu_{X|y_j})^2}{2\sigma_{X|y_j}^2}\right) \mathrm{d}x \log \int_{x_i}^{x_{i-1}} \frac{1}{\sqrt{2\pi}\sigma_{X|y_j}} \exp\left(\frac{(x-\mu_{X|y_j})^2}{2\sigma_{X|y_j}^2}\right) \mathrm{d}x \\ &= -\sum_{j=1}^{l} P(y_j) \sum_{i=1}^{m} (\Phi(x_i \mid y_j) - \Phi(x_{i-1} \mid y_j)) \log(\Phi(x_i \mid y_j) - \Phi(x_{i-1} \mid y_j) \end{aligned} \tag{12}$$

where $\mu_{x|y_i}, \sigma_{x|y_i}$ are probability distribution parameters determined according to each data group divided according to Y values with maximum likelihood estimation method.

The mutual information between discretized traffic flow parameter Z and traffic congestion Y can be calculated according to (10), i.e.

$$\begin{aligned} I(Z;Y) = &-\sum_{i=1}^{m} (\Phi(x_{i+1}) - \Phi(x_i)) \log(\Phi(x_{i+1}) - \Phi(x_i)) + \\ &\sum_{j=1}^{l} P(y_j) \sum_{i=1}^{m} (\Phi(x_i \mid y_j) - \Phi(x_{i-1} \mid y_j)) \log(\Phi(x_i \mid y_j) - \Phi(x_{i-1} \mid y_j) \end{aligned} \tag{13}$$

2.3 Threshold value calculation

The discretization procedure of traffic flow parameters is to maximize the mutual information between discretized traffic flow parameter and traffic state, i.e. $\max I(Z,Y)$. The procedure can be summarized as the following optimization problem.

$$\begin{aligned} &obj \quad \max I(Z;Y) \\ &s.t. \quad x_1 < x_2 < \cdots < x_{m+1} \end{aligned} \tag{14}$$

From the expression of Eq. (13), we can find that the optimization problem is difficult to get an analytical solution. Genetic algorithm is a search technique used in computing to find exact or approximate solutions to optimization and search problems[14]. Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover. It is an efficient method to solve complicated optimization problems. It is used to solve the discretization optimization problem. The procedure can be summarized as follows.

(1) Representation

The variables $x_i, i = 1,2,\cdots,m+1$ are denoted by a binary vector, i.e. chromosome. The vector length depends on the range of variable and required precision. Initial n population of potential solutions should be created.

(2) Evaluation function

The evaluation function plays the role of environment, rating potential solutions in terms of their fitness. Eq (13) is taken as the evaluation function.

(3) Crossover

After prescribing probability of crossover, chromosomes are selected to cross in terms of their fitness. The chance to be chosen is bigger if the fitness value is bigger. Commonly used method is one-point crossover, two-point crossover and multi-point crossover through roulette wheel method. Here one-point crossover is taken.

(4) Mutation

Mutation rate is prescribed firstly. Each mutation alters one or more genes with probability equals to the mutation rate. Random point mutation is a commonly used mutation method. Mutation of binary chromosome is the process of changing between 0 and 1.

(5) Selection

Enlarged sample space composed of chromosomes generated through crossover, mutation, and current generation chromosomes. After competition, *n* best chromosomes are selected as the parent generation of the next iteration.

(6) Stopping criterion

When iteration epoch reaches prescribed threshold value T , iteration stops. The chromosome corresponds to the maximum fitness value is taken as the optimum resolve. Corresponding $x_1, x_2, \cdots, x_{m+1}$ value are taken as the best threshold value of discretization.

Based on above discretization procedure, traffic flow parameters, volume, speed, occupancy etc., can be transformed into discrete variables. Traffic congestion can be identified according to the discrete variables.

3 Extract traffic congestion identification rules with decision tree

Decision tree is used to extract traffic congestion identification rules. A decision tree is a classification data mining tool aimed to extract useful information contained in large data sets. A decision tree is made up of a set of nodes that classify the samples of an objective variable. Each classification is achieved by separation rules according to the numerical or categorical values of variables. Information gain is a measure of selecting an attribute node. The information gain can be expressed as

$$Gain(X_i;Y) = H(Y) - H(Y \mid X_i) \quad (15)$$

Big information gain value means that the attribute variable has close correlation with the classification variable. A decision tree can be constructed top-down using the information gain in the following way [15]:

Step1: begin at the root node.

Step2: determine the attribute with the highest information gain that is not used in an ancestor node.

Step3: add a child node for each possible value of that attribute.

Step4: attach all examples to the child node where the attribute values of the examples are identical to the attribute value attached to the node.

Step5: if all samples attached to the child node can be classified uniquely, add that classification to that node and mark it as leaf node.

Step6: go back to step 2 if there is at least one more unused attribute left, otherwise add the classification of most of the examples attached to the child node.

When decision tree is used to extract traffic congestion identification rules, discretized traffic flow parameters are taken as attributes and traffic state variable is taken as classification variable. Information gain is calculated according to collected historical samples.

4 Traffic congestion identification

The traffic congestion identification procedure based on the proposed method is summarized as in figure 1.

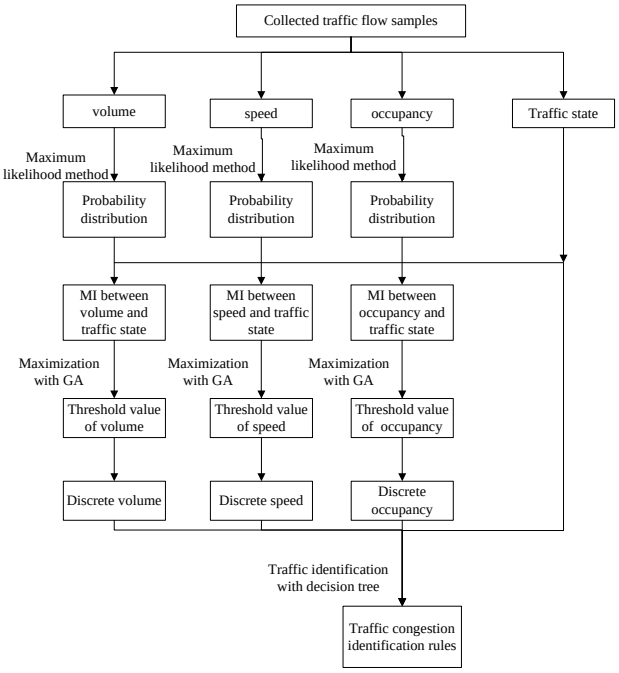


Fig. 1. Flowchart of traffic congestion identification. Notes: MI (Mutual information) and GA(Genetic Algorithm).

Firstly, traffic flow samples are collected. These samples include both traffic flow parameters' historical data and corresponding traffic congestion state. The actrual traffic congestion state can be obtained manually through observing video recorders collected with cameras located at crossways. To simplify, traffic state is divided into two states, i.e. normal flow and congestion. Traffic state is taken as normal flow

when cars can pass the intersection within two signal cycle. Otherwise, traffic state is taken as congestion.

Secondly, probability distribution of traffic flow parameters are estimated according to historical traffic flow data with maxim likelihood method. The probability distribution style can be determined with hypothesis testing method. According to experience, traffic flow parameters approximately obey normal distribution.

Thirdly, the threshold value of each traffic flow parameters are determined with the mutual information maximization method proposed in this paper. The number of segment of each traffic flow parameter can be determined according to practical requirement. For simplicity, the traffic flow parameters are divided into two segments, i.e. only one threshold value needs determining.

Finally, traffic congestion identification rules should be extracted. Different intersection's traffic congestion rules should be extracted respectively.

5 Example

5.1 Data source

Jinan traffic police branch provides us with traffic flow data and video data of Jinan Jingshi Road expressway. There are 14 intersections in the expressway. Inductance loop vehicle detectors are located beneath the roadway approach each intersection and are used to collect traffic flow data. Two intersections are selected to be studied. They are the intersection of Jingshi Road and Qingnian East Road, and the intersection of Yuhan Road and Jingshi Road. The former is taken as current intersection and the later is taken as upstream intersection. The road map is shown as in figure 2. 5 days' historical samples are sorted out. The traffic states of samples are judged through observing video data manually. Traffic flow data are aggregated over 5min interval. In these samples, traffic flow parameters are volume, speed, and occupancy denoted by variables Q,V,O respectively. These traffic flow data corresponding to the period from 6:30 AM to 7:30 PM are chosen. There are total 780 samples in the example, where 118 samples corresponds to congestion and the others corresponds to normal flow. 700 samples are selected randomly to model and the others are selected to test.

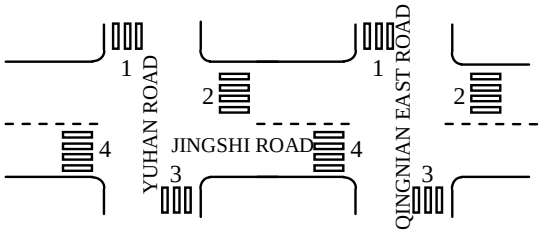


Fig. 2. Map of the intersections.

5.2 Threshold values of traffic flow parameters tables

Traffic flow parameters, volume, speed, and occupancy, are discretized according to above proposed method. They are all supposed to obey normal probability distributions. The probability distributions are tested with hypothesis testing method. For simplicity, each parameter is divided into two segments. Three threshold values corresponding to each traffic flow parameter should be determined. Among them, two values, i.e. the minimum and maximum threshold values, can be prescribed previously according to each parameter's range. The other threshold value is determined with mutual information maximization method. The first segment is denoted by 0 and the second segment is denoted by 1. Traffic state is taken as a binary variable denoted by 0 and 1. 0 denotes normal flow and 1 denotes congestion.

Each traffic flow parameter's probability distributions under different condition are estimated with maximum likelihood method. The probability distributions of volume, speed, and occupancy corresponding to all samples are denoted by $p(q), p(v)$, and $p(o)$ respectively. The samples are divided into two groups according to traffic state value. Conditional probability distributions corresponding to normal flow group are denoted by $p(q \mid y=0), p(v \mid y=0)$, and $p(o \mid y=0)$ respectively. Conditional probability distributions corresponding to congestion group are denoted by $p(q \mid y=1), p(v \mid y=1)$, and $p(o \mid y=1)$ respectively. The estimated probability distribution parameters under different conditions are listed in table 1. The probability distribution of traffic state variable can be determined according to statistical method, i.e. $p(y=1)$=0.151, $p(y=0)$=0.849.

Table 1. Distribution parameters of traffic flow parameters.

	Global distribution $p(\cdot)$		Distribution condition on normal flow $p(\cdot \mid y=0)$		Distribution condition on congestion state $p(\cdot \mid y=1)$	
	σ	μ	σ	μ	σ	μ
Volume	32.61	158.54	33.11	156.56	25.89	171.28
Speed	4.03	18.40	4.12	18.69	1.79	16.54
Occupancy	0.12	0.73	0.12	0.72	0.06	0.77

Based on the probability distributions, mutual information function can be generated. Then mutual information maximization is used to obtain threshold values. In the optimization process, the parameters of GA are set as follows. Based on the range of each parameter, the length of each binary chromosome is set to 8. The crossover rate is set to 0.4, and mutation rate is set to 0.02. The stopping epoch is set to 300. After optimization, the threshold values of each traffic flow parameters are obtained. The obtained threshold values are listed in table 2. Each parameter is divided into two segments according to the threshold values. Each segment is

denoted by 0 or 1, where 0 denotes the first segment and 1 denotes the second segment respectively.

Table 2. Threshold values of each traffic flow parameter.

Volume		Speed(km/h)		Occupancy	
Q_1	0	V_1	0	O_1	0
Q_2	116	V_2	19.1	O_2	0.684
Q_3	250	V_3	100	O_3	1

5.3 Extract identification rules with decision tree

The mutual information between each discrete traffic flow parameter and traffic state variable is calculated according to (13). The parameter with maximum mutual information is selected as the first node to divide the samples into two groups according to its value. In the example, the mutual information between traffic state and volume, speed, and occupancy are 0.0505, 0.2031, and 0.0575 respectively. Speed is selected as the first node. The samples are divided into two groups according to speed's threshold value. The traffic states of the samples corresponding to 1, i.e. $v > 19.1$ km/h, are all normal flow. The samples corresponding to 0, i.e. $v \le 19.1$ km/h, need further classification. According to these samples, the mutual information between traffic state and volume is 0.0603, and the mutual information between traffic state and occupancy is 0.0325. Volume is selected as the second node. The samples to be classified are divided into two groups according to volume's threshold value. The traffic states corresponding to 0, i.e. $q \le 116$ are almost normal flow. It does not need further classification. The samples corresponding to 1, i.e. $q > 116$, need to be classified according to occupancy's value. The traffic states of samples corresponding to 1, i.e. $o > 0.684$, are normal flow. The traffic states of samples corresponding to 0, i.e. $o \le 0.684$, are congestion. The identification procedure is shown in figure 3. The identification rules of traffic state can be extracted, i.e. the traffic state is congestion when speed's value is less than 19.1km/h, volume's value is greater than 116 and occupancy's value is greater than 0.684. Otherwise the traffic state is normal flow. For illustrated the efficiency of the identification rules, 80 test samples are used to test. According to the obtained identification rules, the identification rate is 0.925, shown as in table 3.

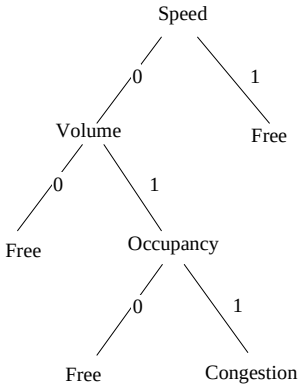


Fig. 3. Traffic congestion.identification process

5.4 Commonly used identification methods

For comparison, commonly used traffic congestion identification methods are used to analyze the traffic flow data. Firstly, California algorithm is used to identify the traffic congestion state. In the method, upstream traffic flow parameter data are used. The threshold value of traffic occupancy difference between upstream and current is set at 0.3. The relative threshold value corresponding to the initial time of congestion is set at 0.2. They are prescribed subjectively. No strict rules can be used. The 80 test samples are used to test the method. Through analyzing, the identification rate is 0.89, shown as in table 3.

Secondly, three-layer BP neural network is used to identify the traffic state. Upstream traffic flow parameter data are used. The setting of the BP neural network is as follows. The activation function of the second layer is sigmoid function, and of the third layer is linear function. The number of input unit is 6, i.e. the traffic flow parameters of current intersection and upstream intersection. The number of hidden unit is set 10. Learning rate is set to 0.02, error is set to 0.001, and echo number is set to 100000. 700 samples are selected as the training samples to train the neural network and the 80 test samples are used to test. After training, the identification rate is 0.90, shown as in table 3.

Table 3. Identification results.

	The proposed method	California method	BP network
Identification rate	0.925	0.89	0.9

6 Conclusions

Traffic state identification is an important content of ITS. Threshold value based traffic identification method is a commonly used one. But none threshold value determination method has been developed. The traffic state identification method proposed in this paper gives a novel threshold value determination method. Mutual information is used to measure the correlation between discrete traffic flow parameter and traffic state. Genetic algorithm is used to solve the mutual

information maximization problem. Decision tree can extract an intuitive identification rules to be used in practical. It is easy to be operated. Through example analysis, we can find that the identification rate is higher than that of commonly used threshold value based identification method. With the proposed method, each intersection's traffic congestion identification threshold values can be determined.

Acknowledgment

This work is partially supported by national youth science foundation (No. 61004115) and Provincial Outstanding Research Award Fund for youth scientist (No. BS2009DX016).

References

[1] J. Lu, L. Cao. "Congestion evaluation from traffic flow information based on fuzzy logic". IE Intelligent Transportation Systems. 1, 50-53, 2003.

[2] Z. S. Yang. "Technology and Application of Basis Traffic Information Fusion". China Railway Publishing House, 2005.

[3] G.Y. Jiang. "Road Traffic State Detection'. China Communications Press, 2004.

[4] C. L. Dudek, C.J. Messer, N. B. Nuckles. "Incident Detection on Urban Freeways". Transportation Research Record, Washington D C, 12-24, 1974.

[5] B. N. Persaud, F. L. Hall, L. M. Hall. "Congestion Identification Aspects of the McMaster Incident Detection Algorithm". Transportation Research Record, 1287, 167-175, 1990.

[6] H. J. Payne, S. C. Tignor. "Freeway Incident Detection Algorithms Based on Decision Trees with States". Transportation Research Record, 682, 30-37, 1978.

[7] Y. D. Chen, Y. Zhang. "Pattern Discovering of Regional Traffic Status with Self-Organizing Maps". IEEE Intelligent Transportation Systems Conference Toronto, Canada, pp: 647-652, 2006.

[8] G. Y. Jiang, L. H. Gang, J. F. Wang."Traffic Congestion Identification Method of Urban Expressway".Journal of Traffic and Transportation Engineering, 6 (3), 87-91, 2006.

[9] F. Porikli, N. Li."Traffic Congestion Estimation Using Hmm Models Without Vehicle Tracking".IE Intelligent Vehicles Symposium, 188-193, 2004

[10] J. L. Zhang, X. Y. Wang. "Study on Traffic Flow Condition Identification Using Wavelet Method". Journal of Wuhan University of Technology, 30 (5), 820-823, 2006.

[11] J. N. Kapur. "Entropy optimization principles with applications". Boston: Academic Press, 1992.

[12] Z. Q. Sun, G. C. Xi, J. Q. Yi, D. B. Zhao. "Select Informative Symptoms Combination for Diagnosing Syndrome". Journal of Biological Systems, 15 (1), 27-37, 2007.

[13] F. J. Shao, Z. Q. Yu. "Data Mining Principle and Algorithm". China WaterPower Press, 2003.

[14] G. Mitsuo, R. W. Cheng. "Genetic Algorithms and Engineering Design". John Wiley & Sons, Inc., 1997

[15] T. M. Mitchell. "Machine Learning". The Mc-Graw-Hill Companies, 1997.

Comparison of Single Image Processing and Bilateral Image Feature Subtraction in Breast Cancer Detection

Aijuan Dong and **Sinanovic Senad**
Department of Computer Science, Hood College, Frederick, MD 21701

Abstract: Although the concept of bilateral asymmetry analysis is appealing and techniques have been applied in automatic breast cancer detection, its application is compromised due to the difficulty in accurately aligning left and right breast images. This study developed a computerized method for automated breast cancer detection using bilateral image feature subtraction and compared it with single image processing technique. Experiment showed that bilateral image feature subtraction method performed better than single image processing technique. The weighted averages of TP (true positive) and FP (false positive) for bilateral image feature subtraction approach were 0.712 and 0.288; while the weighted averages of TP and FP for single image processing technique were 0.519 and 0.486, respectively.

Keywords: single image processing, bilateral image feature subtraction, mammogram, and breast cancer.

1. Introduction

Breast cancer continues to be a major public health problem around the world. The American Cancer Society estimates that invasive breast cancer would be found in 207,090 women in the United States in 2010, the second most frequent cancer diagnosis among women in the United States after skin cancer. Additionally, it is estimated that 40,230 women would die from the disease during this period, second only to lung cancer [1]. Currently the etiologies of breast cancer are unclear, and there is no generally accepted therapy for preventing it. Therefore, the best way to improve the prognosis for breast cancer is early detection when treatment is more effective, and a cure is more likely.

Mammography is one of the most reliable and widely used methods for breast cancer early detection [1]. A variety of abnormalities may appear in mammograms, including masses, microcalcifications, and architectural distortions. In this study, we are particularly interested in mass detection. Masses are groups of cells that are clustered together. They are often denser than the surrounding tissue. Based on shape, masses can be classified as circumscribed, spiculated, or ill-defined. The circumscribed ones usually have a distinct, round boundaries; the spiculated ones have star-shaped boundaries; and the ill-defined ones have irregular shapes. Researchers have found out that the main obstacle of mass detection is the great variability of mass appearance[2].

Various techniques have been proposed for mass detection in the literature. A complete overview can be found in this paper [3]. Since the study reported here is based on bilateral image asymmetry analysis, the discussion will be focused on this aspect. To identify suspicious regions in mammograms, radiologists often observe and compare the left and right breast images of a given patient. Bilateral image asymmetry analysis of mammograms is an important clinical procedure in breast cancer diagnosis [4]. The assumption is that asymmetries between left and right breasts may represent the presence of early signs of breast cancer.

The bilateral image asymmetry analysis techniques generally consist of two steps: (1) align the left and right breast images; and (2) detect asymmetry between left and right images by bilateral image subtraction [3]. Several studies have applied the bilateral image subtraction technique in mammographic image analysis. Yin et al. [5] investigated and compared the performance of a nonlinear bilateral subtraction technique on image pairs with a local gray-level threshold technique on single images. The study found that the nonlinear bilateral-subtraction technique performed better than the local gray-level threshold technique with Az=0.530 for bilateral subtraction and Az=0.385 for local gray-level threshold technique, where Az measures the area under free-response operating characteristic analysis (FROC) curve. Méndez et al. [6] developed a bilateral subtraction method for mass detection. The breast border and nipple were used to align left and right breast images, then two images were subtracted and a threshold was applied on difference image to identify suspicious region. To reduce false-positive rate, texture features were extracted, and size and eccentricity tests were performed. An area under the FROC curve of Az = 0.667 was obtained. Zheng [7] et al. tested and compared single-image segmentation based on Gaussian filtering and bilateral-image subtraction based on left-right image pairs to identify mass regions. The study found that both the single-image segmentation method and the bilateral-image

subtraction method achieved more than 90% sensitivity at a false-positive rate of approximately 0.8 per image. Wang et al. [8] developed an automated scheme to detect breast tissue asymmetry depicted on bilateral mammograms and use the computed asymmetric features to predict the risk of women having or developing breast abnormalities or cancer. Using a single feature, the maximum classification performance level measured by the area under the receiver operating characteristic curve was 0.681 ± 0.038. Using the GA-optimized ANN, the classification performance level increased to 0.754 ± 0.024.

Although the concept of bilateral image asymmetry analysis is appealing and techniques have been applied in automatic mass detection, its application is compromised due to the difficulty in accurately aligning left and right breast images. The left and right breasts are not always symmetric in size and shape. Even they are physically symmetric, the left and right mammograms may not exactly symmetric due to different image acquisition, orientation, and compression [9]. Current bilateral methods are often based on simple, subtraction-based techniques and compute a number of thresholds in an ad hoc manner. Choosing an appropriate threshold that would work across a large set of images is difficult [10].

The aim of this study is to develop a computerized method for automated mass detection by bilateral image feature subtraction. Instead of accurately aligning left and right breast images, performing bilateral image subtraction, and then working on the difference image for asymmetry analysis, this study proposes to (1) extract features from the left mammogram and right mammogram, respectively. This will generate two feature vectors for any given patient, one from left mammogram, the other from right mammogram; (2) calculate difference between feature vectors. Take the two feature vectors from one patient and subtract one feature vector from another. The process generates one difference feature vector per patient; (3) apply classification algorithms on difference feature vectors to classify patients into one of two groups: with normal mammogram and with abnormal mammogram. The rationale behind this approach is that the multitude of difference feature vectors is related to the degree of asymmetry between left and right breasts, which in turn may indicate the presence of or possibility of developing breast abnormalities.

The rest of the paper is organized as follows. Section 2 details the proposed bilateral image feature subtraction method and single image processing technique. Section 3 presents the experiments conducted. Data sets and evaluation criteria are first introduced, and then results are presented and discussed. Section 4 concludes the paper.

2. Methodology

2.1. Bilateral image feature subtraction method

The proposed bilateral image feature subtraction method consists of a few phases as shown in Figure 1.

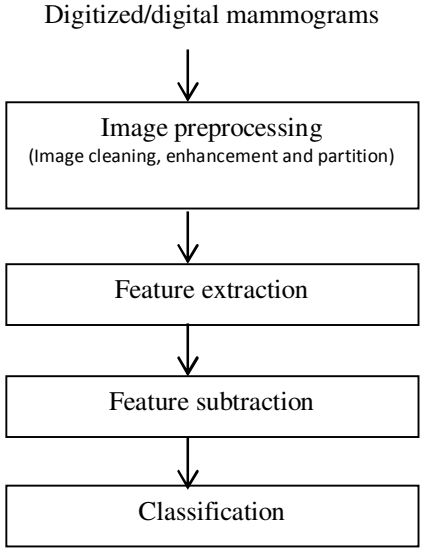


Figure 1. Bilateral image feature subtraction for mass detection

In image preprocessing phase, first unwanted background noise is removed and image contrast is improved. Then each image is split into four quadrants. In feature extraction phase, a group of intensity features are extracted from selected quadrant images. In feature subtraction phase, feature vectors from corresponding left and right quadrants are used to generate difference feature vectors. In the last phase, i.e. classification phase, the classification is performed on difference feature vectors and patients are classified into one of two groups: with normal mammograms and with abnormal mammograms. In the following sections, the details for each step are presented.

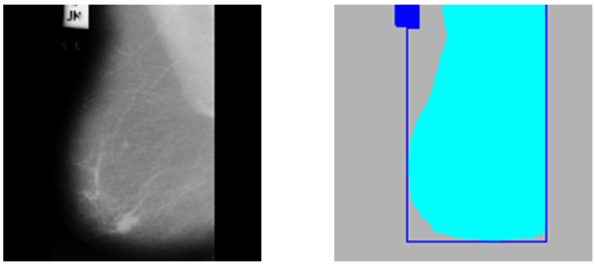
Image preprocessing

A typical mammogram has two kinds of background noises: black background and artifacts such as medical labels (Figure 2(a)). The goal of image cleaning is to remove black background and unwanted artifacts from mammograms as much as possible. Thus, the first step in image cleaning is to find breast region and cropped out the unwanted image portions.

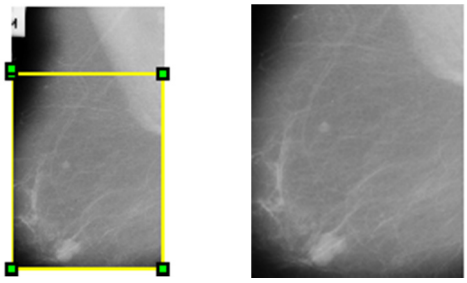
To locate breast region in a mammogram, global thresholding is applied to segment the mammogram (Figure 2(b)). The global threshold of a mammogram is determined using the algorithm proposed in [11]. Since the breast is generally the largest region in a mammogram, the issue of locating breast region is converted to the task of finding the

largest segment. The smallest rectangle containing the largest segment, i.e. the breast region, is then used to remove the background and unwanted medical labels (Figure 2(c)). Since the intensity difference between black background and breast regions and size difference between artifacts and breast regions are large, the global thresholding segmentation can locate breast regions with satisfying accuracy.

Based on the statistics we gathered from Mammographic Image Analysis Society (MIAS) data set [12], the majority of, if not all, mass lesions appear in the lower, round part of the breasts. For better location of region of interest, a cropping operation is then performed. Given an extracted breast region (Figure 2(c)), first eight extrema points, i.e. top-left, top-right, right-top, right-bottom, bottom-right, bottom-left, left-bottom, and left-top, of the breast region are identified. The extrema points of two different regions are illustrated in Figure 3. Depending on the actual region shapes, there often are overlapping points. Then, a bounding box that encloses the breast region is generated by using minimum and maximum X and Y coordinates calculated from extrema points (in yellow). At last, the lower portion of the breast region is extracted (Figure 2(d)). All the preprocessing operations are performed automatically using MATLAB Image Processing Tool Box.



2(a) Original 2(b) Segmented

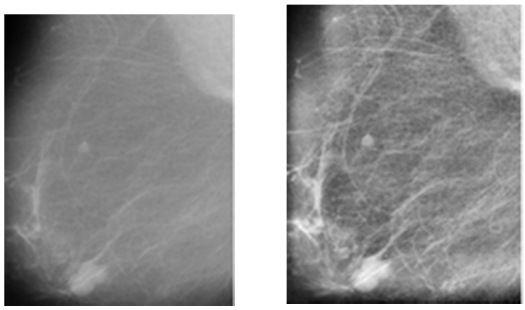


2(c) Cropped 2(d) Extracted

Figure 2. Image cleaning

Figure 3. Extrama points of two different regions

Mammographic images often have low contrast. Over brightness and over darkness diminish the image features. In addition, the illumination conditions are generally different for mammograms acquired at different time. In this study, contrast-limited adaptive histogram equalization, or CLAHE, is applied to stretch image contrast range by increasing the dynamic range of grey levels and change brightness distribution so that all values are equally probable. This operation accentuates image features. Figure 4 shows the effect of histogram equalization.



4(a) Before 4(b) After

Figure 4. Image enhancement

Masses in mammograms are very subtle. They are difficult to detect because they are similar to normal breast tissue and their features are often obscured by normal breast tissue. For better location of region of interests, the cleaned, cropped and enhanced mammograms are split into quadrants. For left and right mammograms from a normal patient, all quadrants are kept. For left and right mammograms from an abnormal patient, only quadrant pairs with abnormalities are kept. For example, Figure 5 is the abnormal right mammogram of a given patient. Only NW and SW quadrants are kept when building the data set.

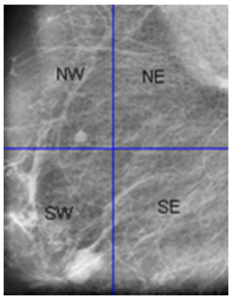


Figure 5. Image partition

Feature extraction

Feature extraction is an important step. The main goal of this step is to extract features that can accurately characterize masses and thus discriminate normal mammograms from abnormal ones. Based on previous study [14], four statistical measures based on intensity are used to describe mammograms. They are mean, variance, skewness, and kurtosis. In the formulae, x^i is the intensity value for ith data point, n is the total number of data points, σ is the standard deviation.

$$m = \frac{\sum_{i=1}^{n} x_i}{n} \quad (1)$$

$$v = \frac{\sum_{i=1}^{n} (x_i - m)^2}{n} \quad (2)$$

$$sk = \frac{\sum_{i=1}^{n} (x_i - m)^3}{n\sigma^3} \quad (3)$$

$$kurt = \frac{\sum_{i=1}^{n} (x_i - m)^4}{n\,\sigma^4} - 3 \quad (4)$$

Feature subtraction

All four intensity features are extracted from quadrant images. For every quadrant pair, two feature vectors are generated, one from left breast quadrant and the other from corresponding right breast quadrant. One difference feature vector is then obtained by calculating the difference between the two feature vectors. Thus, if the total number of quadrant images is n, then $n/2$ difference vectors will be generated. To address the issue of one relatively larger feature value biasing the distance calculation, Gaussian normalization [15] is performed on difference feature vectors to put equal emphasis on each feature.

We expect the values of difference feature vector represent the asymmetries of left and right breast images. The larger the values are, the bigger the asymmetries are, and possibly the more likely potential cancer will be diagnosed.

Classification

The Naïve Bayes classifier is chosen for this study due to the size of data set. The Naïve Bayes classifier is a simple probabilistic classification technique with assumption that all features to be independent. Assume that C is the dependent class variable such that $C = \{c_1,\ldots, c_k\}$ with k known classes and $A_1,\ldots, A_n$ are n features of a given record, the fundamental equation for the Naïve Bayes classifier is:

$$P(C \mid A_1,\ldots,A_n) = P(C)\prod_{i=1}^{n} P(A_i \mid C) \quad (5)$$

The goal of Naïve Bayes is to accurately predict class C, thus C is chosen such that the posterior probability, $P(C \mid A_1,\ldots,A_n)$, is maximized. Hence, this gives us the Naïve Bayes classification rule:

$$C = \arg\max_c P(C = c)\prod_{i=1}^{n} P(A_i \mid C = c) \quad (6)$$

where $c \in C$.

This study is two-class classification problem, i.e. patients with normal mammogram and patients with abnormal mammogram. The features used are mean, variance, skewness, and kurtosis as described in the section of "Feature extraction".

2.2.Single image processing technique

Single image processing technique (Figure 6) employed in this study is similar to bilateral image feature subtraction method. The only difference between the two is that single image processing does not perform feature subtraction step (Figure 1). Features are extracted from quadrant images and classification is applied on normalized feature vectors directly.

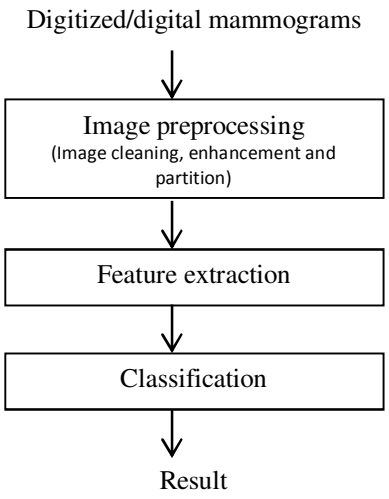


Figure 6. Single image processing method

3. Experiments

3.1. Data sets

The image collection used in this study comes from the Mammographic Image Analysis Society (MIAS). The corpus has a total of 161 paired images, with each image falling into one of seven categories: calcification, circumscribed masses, spiculated masses, architectural distortion, asymmetry, other ill-defined masses and normal. The first six categories are considered abnormal. The mammograms were originally digitized with 50 micron pixel edge but then reduced to 200 micron pixel edge. The size of each image is 1024 x 1024 and the bit depth per pixel is 8. Other ground truth information, such as breast tissue type, severity of abnormality, and location of abnormalities, is also provided.

In this study, we were interested in classifying patients as either with normal mammograms or with abnormal mammograms. All patients with non-dense mammograms containing circumscribed and speculated masses, and a random selection of patients of normal mammograms were considered. After images were cleaned, enhanced and partitioned, there were 104 quadrant images participating in feature extraction and subtraction, which generates 52 difference feature vectors that participated in final classification. Half of 52 features were labeled as abnormal, the other half were labeled as normal.

3.2.Evaluation criteria

For performance evaluation, the following indicators were used: True Positive (TP), False Positive (FP), precision, recall, F-score, and the area under Receiver Operating Characteristic curve. The detailed explanation of each indicator can be found in Weka documentation [14].

3.3.Results and discussions

In our study, we used Naïve Bayes classifier from the WEKA software suite. WEKA is a workbench for machine learning and data mining that is programmed in Java and allows for the domain specialist to select and investigate the application of different machine learning techniques to real world problems [16].

Table 1. Comparison between single image processing and bilateral image feature subtraction

Evaluation criteria		bilateral image feature subtraction	single image processing
TP	Abnormal	0.577	0.538
	Normal	0.846	0.5
	Weighted Avg.	0.712	0.519
FP	Abnormal	0.154	0.5
	Normal	0.423	0.462
	Weighted Avg.	0.288	0.481
ROC	Weighted Avg.	0.75	0.574
Classification accuracy		71.15%	51.92%

In general, when data set is not large enough to have separate training and testing data sets, cross validation is preferred to splitting data set into training and testing sets. Therefore, we used cross validation in the experiments. For all runs, 10-fold cross validation was used.

As Table 1 shows, the weighted averages of TP and FP for bilateral image subtraction method were 0.712 and

0.288, while the weighted averages of TP and FP for single image processing technique were 0.519 and 0.481, which showed the proposed bilateral image feature subtraction method performed better.

From Table 1, we can also see that TPs of abnormal cases were lower compared to those of normal cases, and the FPs of normal cases were higher than those of abnormal cases. This indicates patients with normal mammograms tend to be classified as with abnormal ones. This reflects close resemblance between mass and normal breast tissue and the need for further investigation of image features with more discriminating power.

4. Conclusions

In this paper, we proposed an automatic mass detection method using bilateral image feature subtraction and compared it with single image processing technique. Instead of accurately aligning left and right breast images, performing bilateral image subtraction, and then working on the difference image for asymmetry analysis, this study proposed bilateral asymmetry analysis based on image feature difference.

Experiment results showed that bilateral image feature subtraction performed better than single image processing, with weighted averages of 0.712 and 0.288 for TP and FP for the testing data set, respectively.

The future work will be targeted at reducing false positives and further test the bilateral image subtraction approach on larger data set.

.

5. References

[1]. American Cancer Society, "Cancer facts and figures", 2010.

[2]. I. Christoyianni, E. Dermatas, G. Kokkinakis, "Fast detection of masses in computer-aided mammography", *IEEE Signal Process. Mag*., 17(1), pp. 54-64, 2000.

[3]. H. D. Cheng, X. J. Shi, R. Min, L.M. Hu, X.P. Cai and H.N. Du, "Approaches for automated detection and classification of masses in mammogram", *Pattern Recognition*, 39(4), pp. 646-668, 2006.

[4]. G. Cardenosa, *Breast Imaging Companion*. Lippincott-Raven, Philadelphia, NY, 2007.

[5]. F. F. Yin, M. L. Giger, C.J. Vyborny, K. Doi, R.A. Schmidt, "Comparison of bilateral-image subtraction and single-image processing techniques in the computerized detection of mammographic masses", *Invest. Radiol.*, 28(6), pp.473–481, 1993.

[6]. A. J. Mendez, P. G. Tahoces, M. J. Lado, M. Souto, J. J. Vidal, "Computer-aided diagnosis: automatic detection of malignant masses in digitized mammograms", *Med. Phys.*, 25 (6), pp. 957–964, 1998.

[7]. B. Zheng, Y. Chang and D. Gur, "Computerized detection of masses from digitized mammograms: Comparison of single-image segmentation and bilateral-image subtraction", *Academic Radiology,* 2(12), pp. 1056-61, 1995.

[8]. X. Wang, D. Lederman, J. Tan, X. Wang and B Zheng, "Computerized Detection of Breast Tissue Asymmetry Depicted on Bilateral Mammograms: A Preliminary Study of Breast Risk Stratification", *Academic Radiology,*17(10), pp.1234-1241, 2010.

[9]. Y. Yuan, M. L. Giger, H. Li, N. Bhooshan and C. A. Sennett, "Multimodality Computer-Aided Breast Cancer Diagnosis with FFDM and DCE-MRI", *Academic Radiology,* 17(9), pp.1158-1167, 2007.

[10]. M. P. Sampat, M. K. Markey, and A. C. Bovik, "Computer-Aided Detection and Diagnosis in Mammography", in *Handbook of Image and Video Processing*, By Alan C. Bovik (ed), pp.1195-1217, 2005.

[11] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, pp. 62-66, 1979.

[12]. J Suckling et al., "The Mammographic Image Analysis Society Digital Mammogram Database*", Exerpta Medica*, International Congress Series 1069, pp.375-378, 1994.

[13]. MATLAB Image Processing Toolbox, URL: http://www.mathworks.com/help/toolbox/images/, last accessed in May, 2011.

[14]. O. R. Za¨ıane, M. L. Antonie, A. Coman, "Mammography Classification by an Association Rule-based Classifier", *Proc. Of the International Workshop on Multimedia Data Mining*, pp. 62-69, 2002.

[15]. Q. Iqbal and J. Aggarwal, "Combining Structure, Color and Texture for Image Retrieval: A Performance Evaluation", *Proceeding of the 16th International Conference on Pattern Recognition*, Quebec City, Canada, vol. 2, pp. 438-443, 2002.

[16]. G. Holmes, A. Donkin, and I. H. Witten. "WEKA: A Machine Learning Workbench", *Proc. of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems*, pp. 357-361, 1994.

Simple R-Tree for Temporal Searches

Paul te Braak[1], **Richi Nayak**[2]
[1,2]School of Computer Science, Queensland University of Technology, Brisbane, QLD, Australia

Abstract - The R-Tree and variant indexes have been used for multidimensional indexing since 1984 and applied to temporal data since 1994. Despite this, there are few functional implementations of the index that can be applied directly to the temporal data. This paper explores the use of the index for temporal subsequence searches and identifies two implications that must be considered so the index can be used in the temporal domain. We empirically evaluate the requirements for equi-node splits. The tree performs well and is comparable to other methods eliminating approximately 86% of search space.

1 Introduction

The R-tree index has been used to improve the performance of searches over temporal data since 1994. Despite this use, there are a small number of functional implementations for the Tree that are readily available. Additionally, the use of the tree in temporal searches presents a unique set of problems which makes the use of such an index structure troublesome.

The R-tree was proposed by Guttman [1] as a method for spatial indexing and has been used [2-4] as a method for indexing temporal data. However, despite the use of the structure in their work, we found no functioning implementation of the tree for temporal recognition. This project is motivated by the lack of implementation detail for the index on temporal data and we seek to share our implementation with others. The project builds a static tree which performs well against random length tests, eliminating approximately 86% of the search space.

2 Background

The R-tree was proposed by Guttman [1] as a method for spatial indexing and since a spatial index is essentially a multidimensional (multi-axis) the tree is commonly referred to as a multi-dimensional index. Similar to a B-tree, the index is a height-balanced tree with index records in its leaf nodes containing pointers to data objects [1].

Within the index, an object is contained and reduced to the ranges of its axis, that is, its bounding rectangle. More formally, I = <I0, I 1, I 2 … I n> where I is the bounding rectangle and each subscript definition of I represents a dimensional axis and the upper and lower bounds of the all data points along that axis. The tree takes the general form of <I, pointer or identifier>, which either defines lower order rectangle (pointer) or the tuple extractor (for a leaf in the tree).

The structure aggregates definitions of nodes up paths in the tree and thus allows navigation in a hierarchical manner. These concepts are better explained in Guttmans original work however, for our purpose and brevity, the tree is a height balanced B-tree that aggregates axis dimensional bounds. The work of Guttman was refined by Beckmann [5] who focused on tree development and the splitting of tree nodes based on margin and overlap. The authors argue that splitting a node on the basis of minimal shared space does not guarantee the best index performance, and, the splitting of the node should be done on the basis of margin, overlap and storage utilization. Assent et al [6] continued tree development by implementing the TS-tree, a tree developed especially for time series data. Unlike the prior trees proposed but Guttman and Beckmann, the TS-tree boasts zero overlaps of bounding rectangle nodes and is built in a bottom up manner.

3 Application of the Tree

Temporal data mining is concerned with mining sequential datasets where records are ordered. This concept plays an important role in temporal mining because it permits the inclusion of many domains which are not directly included in the temporal field. For example, there are the standard applications for temporal data mining which include a time dependent view of data and include prediction for weather and financial stocks [7]. Antunes et al [8] include inventory evaluation, ECG and treatment effectiveness in the medical domain and engineering applications arising from sensor data. However, other, non-traditional applications include shape recognition and classification [9, 10], handwriting classification [11], motion capture and speech recognition.

Temporal pattern recognition generally takes two forms. Either the search identifies sequences that are within a given threshold of the query, or the search involves a KNN style search which identifies the N most similar candidates to the query. Regardless of which technique is applied, a common approach to performance improvement has been the use of an index (or index type structure based on attribute reduction) to reduce candidate sets prior to computationally expensive distance measures (such as Euclidean distance or Dynamic Time Warping).

The method of reducing search space for temporal mining has been used by various authors and forms. Faloutsos [2] used an R-tree to index sequences that had been reduced using a Fourier transformation. Yi [12] used an upper and lower bound to identify candidates that could not be close to the existing set. Park [13] defined the sequence using a feature

vector which was indexed using an R-tree (or variant). Vlachos [4], Keogh [3] have used an R-tree (or variant) to describe a sequence and then used the index bounds to eliminate candidate sequences in a pre-pruning step.

Despite the reference for R-tree structures by the authors above, we found no working implementation of the tree for temporal data. This paper discusses our implementation of the tree and identifies what issues are specific to the temporal domain. The tree was designed to satisfy queries of varying length.

4 Tree Design

The following sections discuss the static tree that was implemented. The main concepts of the tree are that;

1. Each root child (the child of the root) represents the temporal information for an entire class.
2. Nodes are split without overlap. Each node (at a given level defines) a distinct temporal proportion of its class.
3. The tree is unbalanced. The data used in the project represented classes with lengths between 1 and 3,636 data points. The use of an unbalanced tree reduced space requirements and eliminated redundancy.

4.1 Index Overview

A crucial consideration in our index construction is the relationship (and maintenance) consecutive bounding rectangles. This relationship must be preserved because it permits aggregation across the time axis. If a sequence is split in half then, the area between the first and second half can only be examined if the first half follows the second half.

Each node in the index includes the dimensional area for that node and a pointer to leaf tuple(s). This pointer is the class name, the start of the sequence (commencement pointer) and the node length.

The tree is built in a top-down manner. Each node at the first level of the tree defines a class which is split along the temporal axis until a stop condition is met (based node length). Unlike the trees of Guttman and Beckmann, this tree does not require leaves appear at the same level. In fact, since each child of the root defines a specific class of varying length, each child node would have a depth relative to the number of data points in the sequence.

4.2 Tree Build Algorithm

The tree is built through recursive node addition. From the root of the tree, a node is added to define a class, then, this node is split until some condition is satisfied. Splitting merely halves the parent across the time axis with the left hand split (LHS) being the first half of the existing sequence and the RHS is the remaining. The algorithm for building the tree is shown in Algorithm 1.

Algorithm 1 - Tree Creation

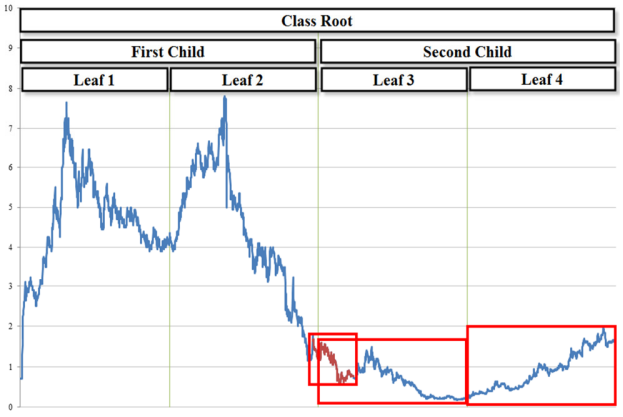
```
Tree -> new tree
MetaData -> Set of Bounding MBR for each class

Foreach Bounding MBR in MeataData Set
    Add_Node_To_Tree (Bounds.MBR)
Next Bounding MBR in MetaData Set
Save Tree and Exit

Function Add_Node_To_Tree(MBR, [Optional] ParentNode)
/* Creates a Node (N) for MBR */

    Append N to ParentNode (if exists) or Root of tree

    /* Test if N can be split */
    If N can not be split
        Exit
    Else
        LHS -> New MBR defining LHS of MBR
        RHS -> New MBR defining RHS of MBR
        Add_Node_To_Tree(LHS, N)
        Add_Node_To_Tree(RHS, N)
    End if

End Function
```

4.3 C. Split Consideration

Vlachos et al [4] define methods of splitting as K-Optimal, Equi, Random and Greedy. The Tree used an equi-split to maximize the coverage of each node. Nodes were split until a condition based on node length was met. Our testing showed that there are two important considerations for the choice of split length. Firstly, when the size of the split threshold is much larger than the query, the index may not identify correct nodes because the expected sequence was split and this resulted in distorted MBR(s). That is, the query space is not defined in the index because splitting. Secondly, when the general area is correctly identified, the split may have striped the actual sequences leading or trailing edge.

Both these examples are demonstrated in Figure 1 where the node containing the majority of the query has a smaller bounding definition that the query (and is not returned as a match) and, an alternate node (from a disjoint time area) is identified as a suitable node because its bounds are greater than the query.

Figure 1 - Split Consideration



To circumvent these issues we choose a small split threshold

(in comparison to the expected query length) and extend the detailed search domain by the query length.

4.4 Tree Search

The search returns the lowest level set of nodes that are bound by the query. We note that because the query length is expected to be greater than that of a leaf, a leaf node will generally not be returned by the search. Rather, the search will return a node which lies between the root and a leaf.

To identify whether a node bounds the query, we define the logical function InBounds. The predicate for InBounds is that all the data points for the query lie with the node bounds.

The tree search algorithm is shown in Algorithm 2. The use of recursion allows for unknown branch lengths.

Algorithm 2 - Tree Search

```
Function Query_Tree /* Populates Leaf_List */
Query -> Query to conduct against the tree
Tree -> R-tree
Node_List -> Empty set of nodes (candidate set)

Foreach child_node under Tree Root
    Query_Node(child_node)
Next child_node

Function Query_Node(Node)

    If Node is Not InBounds(Query) then
        Exit
    Else
        Foreach child_node under Node
            If child_node is Inbounds(Query) then
                Query_Node(child_node)
        Next child

        If no child_node were InBounds then
            Append Node.Leaves to Node_List

    End If

End Function
```

4.5 Detail Search

The tree search provides a set of candidate nodes (with pointers to data points) that can be searched using a comprehensive distance calculation and a sliding window. At this point, we consider the sequence starting point to be the nodes starting point ***less the length of the query***, and extend its length by this amount. The algorithm for the detail search is shown in Algorithm 3.

Algorithm 3 - Detail Search

```
Function Query_Data
    Forech node in Node_list Nodes
        DataStream = new Stream for node
        While DataStream has data
            Add data to window
            if window is valid then /* Conduct Some Test */
    Next node
End Function
```

5 Empirical Evaluation

The data used in this work was the daily price information for the Australian Stock Exchange between 1st Jan 1993 and 8th Dec 2006 for stocks commenting with 'A'. Daily price information includes each stocks opening, closing, maximum, minimum value and the volume of stocks traded on a given day. This represents 338 stocks and 424,749 data points. Testing was conducted on a Dual Core (2.34 GHz) desktop with 8GB RAM running Windows 7 with SQL server 2008.

5.1 Approach

Testing examined two aspects, firstly, the efficiently of the tree in its ability to reduce the search space. This test generated fifty thousand random test cases and validated the Tree by ensuring that the test existed in search results. The second tests performed an exact retrieval on twelve tests from the randomly generated test data.

5.2 Fifty Thousand Tests

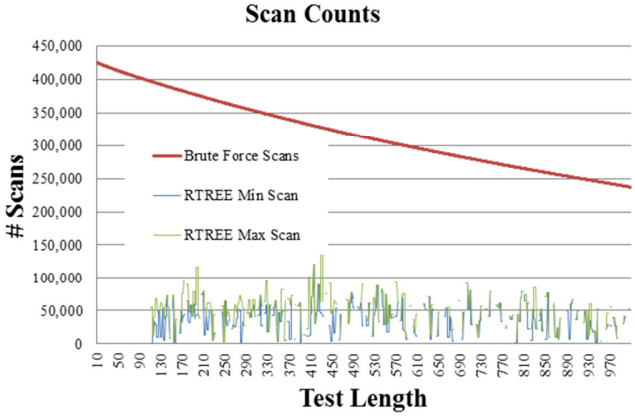
We conducted a large number of tests to determine the trees general ability to reduce search space, because the reduction has a direct consequence on the time required to conduct a thorough search. We generated 50,000 random tests with lengths between 100 and 1,000 data points and compared the number of brute force scans to the number of scans returns from the tree search. A scan is a valid subsequence that need be searched (and hence requires a distance calculation).

A brute force search requires a scan of each for each window that can be formed over the data and therefore is a function of the number of data points in the candidate set. This is given by the formula (1).

$$\sum_{Stock(i=0)}^{All\ Stocks} Stock(i).DataPoints - n + 1 \quad (1)$$

The plot of the test length verse search space (scans) is shown in Chart 1.

Chart 1 - Scan Efficency



The maximum tree scans performed for any query was 140,360 (test length of 423). The equivalent brute force

requirements would be 327,317 equates to a reduction of 57.12%. We define this as scan efficiency (#{scans eliminated by index} / #{brute force scans}). In some instances, the tree performed remarkably well. The smallest number of scans conducted was 140 (achieved 294 times), on test with lengths between 101 and 104 data points. A brute force scan would require between 396,324 and 397,070 scans and this represents an efficiency of 99.96%.

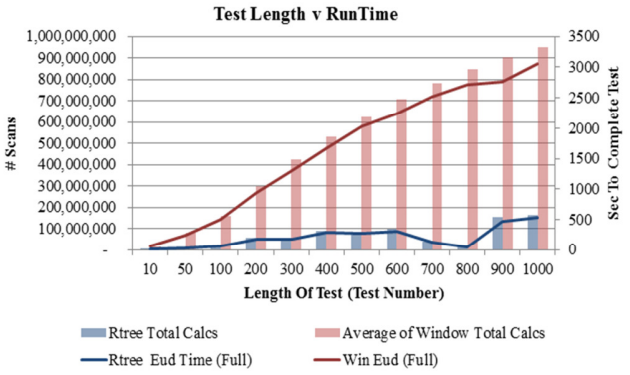
The average maximum number of scans per test was 51,882. Brute force requirements would be 308,617 and the tree has an efficiency of 83.2%.

5.3 Effect using a Distance Calculation

Twelve tests were used to examine the time requirements for exact matching. Each test was run twice using the Euclidean distance measure.

The average combined pruning power these tests was 86.90%. Average scans for the tree was 42,873 compared to 327,277 using brute force. Similarly, the scan time reduction was 87.73% with the tree scan time being 204 sec and the brute force requiring 1,668 sec. For all tests, the tree performed significantly better than the brute force approach (ranging from an increase in performance of 81.18% to 98.40%). This is demonstrated in Chart 2.

Chart 2 - Scan Time Requirements



5.3.1 Calculation Abandonment

There is a correlation between test length and calculation time which can be reduced through the use of early abandonment. This technique aborts the distance calculation when the accumulative distance has reached some threshold (usually the best result so far). To demonstrate the significance of this technique, the twelve tests were also run using early abandonment. The average performance increase for the tree was 98.25% with average search time reduced to 29 sec from 204 sec and brute force searches improved by 85.72% (from 1,668 sec to 397sec).

6 Conclusion

The implementation of the R-tree for time series data presents some unique challenges for maintaining series for searches of multiple query lengths. There are considerations regarding the split of sequences which form tree branches and the length of sequences that will form leaves. Despite these issues, the R-tree performs well as a simple implementation and achieves pruning rates similar to its original implementation.

7 References

[1] Guttman, A., R-trees: A Dynamic Index Structure for Spatial Searching, in Proceedings of the 1984 ACM SIGMOD international conference on Management of data1984, ACM: Boston, Massachusetts. p. 47-57.

[2] Faloutsos, C., M. Ranganathan, and Y. Manolopoulos, Fast Subsequence Matching in Time-Series Databases, in Proceedings of the 1994 ACM SIGMOD international conference on Management of data1994, ACM: Minneapolis, Minnesota, United States. p. 419-429.

[3] Keogh, E. and C. Ratanamahatana, Exact Indexing of Dynamic Time Warping. Knowledge and Information Systems, 2005. 7(3): p. 358-386.

[4] Vlachos, M., et al. Indexing Multi-Dimensional Time-Series with Support for Multiple Distance Measures. 2003. ACM.

[5] Beckmann, N., et al., The R*-tree: an Efficient and Robust Access Method for Points and Rectangles, in Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data1990, ACM: Atlantic City, New Jersey, United States. p. 322-331.

[6] Assent, I., et al., The TS-tree: Efficient Time Series Search and Retrieval, in Proceedings of the 11th international Conference on Extending database technology: Advances in Database Technology2008, ACM: Nantes, France. p. 252-263.

[7] Laxman, S. and P. Sastry, A survey of Temporal Data Mining. Sadhana, 2006. 31(2): p. 173-198.

[8] Antunes, C. and A. Oliveira. Temporal Data Mining: An Overview. in KDD 2001 Workshop on Temporal Data Mining. 2001. San Francisco, USA: Citeseer.

[9] Keogh, E., et al. LB_Keogh Supports Exact Indexing of Shapes Under Rotation Invariance with Arbitrary representations and Distance Measures. 2006. VLDB Endowment.

[10] Ye, L. and E. Keogh, Time Series Shapelets: A New Primitive for Data Mining, in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining2009, ACM: Paris, France. p. 947-956.

[11] Wei, L. and E. Keogh, Semi-Supervised Time Series Classification, in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining2006, ACM: Philadelphia, PA, USA. p. 748-753.

[12] Yi, B.-K., H.V. Jagadish, and C. Faloutsos, Efficient Retrieval of Similar Time Sequences Under Time Warping, in Proceedings of the Fourteenth International Conference on Data Engineering1998, IEEE Computer Society. p. 201-208.

[13] Kim, S.-W., S. Park, and W.W. Chu, An Index-Based Approach for Similarity Search Supporting Time Warping in Large Sequence Databases, in Proceedings of the 17th International Conference on Data Engineering2001, IEEE Computer Society. p. 607-61

On Sample Selection Bias in Large-Scale Online Stream Mining: a Model Indexing Approach

Xiong Deng, Moustafa M. Ghanem, and Yike Guo
Department of Computing, Imperial College London, London, United Kingdom, SW7 2AZ

Abstract - *Large-scale data stream applications including* ***network intrusion detection*** *pose the non-trivial problem of* ***sample selection bias*** *to online data mining. The problem greatly degrades state-of-the-art data mining models including C4.5 and soft margin SVM, incremental data mining algorithms including CVFDT, and online ensemble model methods including the weight-by-accuracy approach. Inspired by the web search engine technology, this paper proposes an index-based ensemble model approach to ease this problem. During online training, we summarize the variable distribution characteristics of each data sample before training a model on it, and then index the models to their corresponding distribution characteristics. During online classification, we indentify the most appropriate models for chunks of incoming unlabeled instances by matching their variable distribution characteristics. Experiments studied on both synthetic datasets and* ***network intrusion detection*** *domain demonstrate substantial advantages of the proposed approach over state-of-the-art stream mining algorithms in terms of easing the* ***sample selection bias*** *problem and therefore improving classification accuracy.*

Keywords: Stream mining, sample selection bias, online learning, algorithm, network intrusion detection

1 Introduction

Sample selection bias refers to the learning challenge that training variable distributions differ from test ones [1; 2]. It is a non-trivial problem in *batch learning* scenarios, where datasets are partitioned into training and test subsets and models trained on the training subset are evaluated on the test ones. This is because many state-of-the-art supervised data mining models including C4.5, *Na ïve Bayes* and soft margin SVM are highly sensitive to biased training samples [1].

This challenge turns much more substantial in online stream mining scenarios where the instances arrive in an online manner as time elapses. A model's observation is, thus, restricted by what have been collected. For example, in *network intrusion detection* [3; 4], there are always continuous bursts of different intrusion types against a learned model along with the time step. Furthermore, given the huge and possibly indefinite number of instances, learning a model on all historical distributions from scratch becomes too time-consuming and therefore almost impractical. In addition, note that, although the variable distributions in the presented applications are changing over time, their data-generating mechanism may often keep stable (This is different from the popularly-known *concept drift* problem [5]). In *network intrusion detection*, the rules for describing a specific intrusion type (*e.g. buffer overflow* and *smurf* [6]) are not likely to change [3; 4]. Thus, a model trained on a biased sample may still perform effectively on samples of similar variable distributions.

Large bodies of studies have been devoted to large-scale online stream mining including incremental mining algorithms [7-11] and online ensemble model methods [3; 12-15]. Through incremental mining algorithms, outdated data distributions are gradually discarded, while updated data distributions are continuously retrieved according to most recent instances. CVFDT {Hulten, 2001 1640 /id} is a famous example of these algorithms. It works by keeping a decision tree model consistent with an adjustable *sliding window* of instances. Every time a new instance arrives, it updates the sufficient statistics at its nodes by incrementing the counts corresponding to the new instance and decrementing the counts corresponding to the oldest instance in the *window*. It grows alternative sub-trees immediately after a great increase of classification error and then performs replacement of the original sub-trees if necessary.

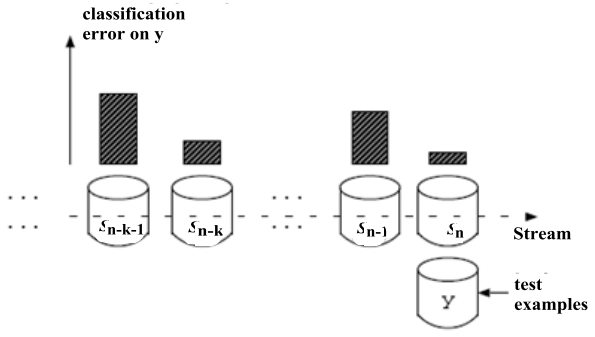


Figure 1. The weight-by-accuracy ensemble model approach [14].

Online ensemble model methods train multiple base models on the biased data distributions and then combine some of them in order to best match the data distributions of incoming instances. A well-known example of this kind is the weight-by-accuracy approach (WBA) proposed by Wang *et al*. [14]. As shown in Figure 1, this approach practically maintains a number of base models built on recent labeled data chunks (*i.e.* $S_{n-k-1}, \ldots, S_n$) of same size. Each time before a classification on an incoming unlabeled data chunk (*i.e.* y. y has the same size as the labeled chunks), WBA estimates the

prediction accuracy of each base model on y with its validation accuracy on S_n. This validation accuracy is then used as the weight of the model during classification.

However, when dealing with *sample selection bias*, there are some critical issues in these approaches. In a single incremental mining model, frequent changes of distinct incoming variable distributions may result in dramatic updating of the model and greatly deteriorates its effectiveness. In many online ensemble model methods, there are big flaws to determine how the historical data distributions can be matched to the unknown data distributions of incoming examples. The weight-by-accuracy ensemble model approach is excellent if the estimation of prediction accuracy can be "accurate". Nevertheless, this assumption is violated under *sample selection bias*.

This paper proposes to maintain variable distribution characteristics of the biased samples in addition to training multiple base models on these samples. The variable distribution characteristics are used to distinguish these base models. Thereby appropriate models can be selected and combined to match incoming variable distortions. The general idea is borrowed from the state-of-the-art web-search engine mechanisms where huge quantity of web pages are crawled, processed and indexed in real time and during online search, an ordered web-page subset that can optimally match a query request are selected [16].

This approach is straightforward but effective. It can avoid both the frequent updating issue in incremental data mining algorithms and the inaccurate estimation issue (due to the biased samples) in many other online ensemble model methods. This approach is also relatively efficient compared to other ensemble model methods including the weight-by-accuracy approach.

This rest paper is organized as follows. Section 2 defines the *sample selection bias* problem in data stream context and distinguishes it from the *concept drift* problem. Section 3 introduces the proposed model indexing approach. Section 4 experimentally studies its effectiveness and efficiency. Section 5 presents related work and contributions of this paper. Section 6 concludes this paper.

2 Preliminaries

2.1 Concept Drift and Sample Selection Bias

In a stream environment, two problems may lead to great degradation of model performance: *sample selection bias* and *concept drift*. They are due to either limitations of the data-collecting mechanisms or changes of the data-generating mechanisms. They can be fully expressed as changes of the joint distributions $P(X, c)$, where $X = \langle v_1, v_2, \dots, v_n \rangle$ denotes any vector in the pre-defined n-dimensional variable space and $c \in \{c_1, c_2, \dots, c_m\}$ denotes any label in the pre-defined class label set.

Let $D = \{D_t | t = 1,2, \dots\}$ denote a data stream that are sequentially split into an indefinite number of data block D_t according to the time step, t. The *concept* of the streaming data in a given sequential data block D_t is determined by the data-generating or data-collecting mechanisms during t, but is naturally reflected in the joint distributions, $P_{D_t}(X, c)$, of the examples, X and labels, c, in D_t. Note that $P(X, c) = P(X) * P(c|X)$. The changes of *concept* can be transformed to the two kinds: changes of $P(X)$ and/or $P(c|X)$. They are termed *sample selection bias* and *concept drift* respectively. More specifically, we define them as follows.

Definition 1. ***Concept*** *of D_t is defined as the joint distributions $P_{D_t}(X, c)$.*

Definition 2. ***Concept drift*** *happens between D_{t1} and D_{t2}, if the posterior probabilities change, i.e. $P_{D_{t1}}(c|X) \neq P_{D_{t2}}(c|X)$.*

Definition 3. ***Sample selection bias*** *between D_{t1} and D_{t2} happens if and only if the variable distributions are different, i.e. $P_{D_{t1}}(X) \neq P_{D_{t2}}(X)$ and $P_{D_{t1}}(c|X) = P_{D_{t2}}(c|X)$.*

As can be observed, *concept drift* normally reflects an actual change of the data-generating mechanism, while *sample selection bias* expresses the situations where although there is a stable data-generating mechanism, there are differing variable distributions in the streaming data.

Sample selection bias is a non-trivial problem in stream mining. It is common since it is often difficult for a data-collecting method to accumulate sufficient and unbiased training samples in the context of data streams. It also greatly deteriorates classification performance of a great number of state-of-the-art data mining models including C4.5, *Naïve Bayes* and soft margin SVM [1], incremental data mining algorithms including CVFDT [9], and online ensemble model methods including the weight-by-accuracy approach [14].

2.2 Sample Selection Bias in Network Intrusion Detection

Sample selection bias can be widely identified in real-world stream mining applications, a typical instance of which is *network intrusion detections* [6] where there are always bursts of different intrusion types against a learned model along with the time step. As the rules for describing each intrusion type always keep the same once the type is identified [3; 4], there is no *concept drift*. However, in the context of streaming data, there is *sample selection bias* due to the limitation of a model's observation [3; 4].

3 The Model Indexing Approach for Sample Selection Bias in Data Streams

3.1 Framework Definition

Under *sample selection bias*, since the data-generating mechanism keeps stable, $P_{D_t}(X, c)$ can be fully expressed as $P_{D_t}(X)$. This motivates the model indexing approach.

More specifically, as a data mining model, M_t, is supposed to fully represent the *concept* of a data block, D_t. First, we train a base model models, M_t, on $P_{D_t}(X, c)$ and also extract a *data*

pattern,K_t, to express $P_{D_t}(X)$. The model set and *data pattern* set can be denoted as:

$$M = \{M_t | t = 1,2 \dots, \} \quad (1)$$

$$K = \{K_t | t = 1,2 \dots, \} \quad (2)$$

Next, we organize each base model in line with its corresponding *data pattern*, resulting in the *index* system, denoted as:

$$I = \{\langle K_t, M_t \rangle | K_i \in K, M_i \in M, t = 1,2 \dots, \} \quad (3)$$

Then we introduce a similarity measure which determines whether two given *data pattern*, K_i, K_j , represent similar *concepts*, denoted as $s(K_i, K_j) \in [0,1]$. The larger the value is, the more similar the *data patterns* are. Given a new *data pattern*, K_n , extracted from a block of incoming unlabeled instances, then *model selection* mechanism is denoted as:

$$O = \{\langle K_n, K_t \rangle | S(K_n, K_t) > \theta, 1 > \theta > 0, K_t \in K\} \quad (4)$$

where θ is a pre-defined threshold of the acceptable similarity of the *data patterns*. If a unique *data pattern* is desired, the *data pattern* having the largest similarity value is chosen, given by:

$$S(K_n, K_t) = \begin{cases} 1, & if\ t = argmax_i(S(K_n, K_i)), \\ 0, & otherwise \end{cases} \quad (5)$$

Finally, the *model indexing* approach (MI), is defined as:

$$MI(K_n) = \{M_i * w_i | \langle K_n, K_i \rangle \in O, \langle K_i, M_i \rangle \in I, w_i = S(K_n, K_i)\} \quad (6)$$

In addition, the *knowledge base* is a dynamically-updatable knowledge repository storing historical *data mining models*, *data patterns* and *index*, denoted as a triple:

$$KB = \langle M, K, I \rangle \quad (7)$$

3.2 The Online Training and Prediction Process

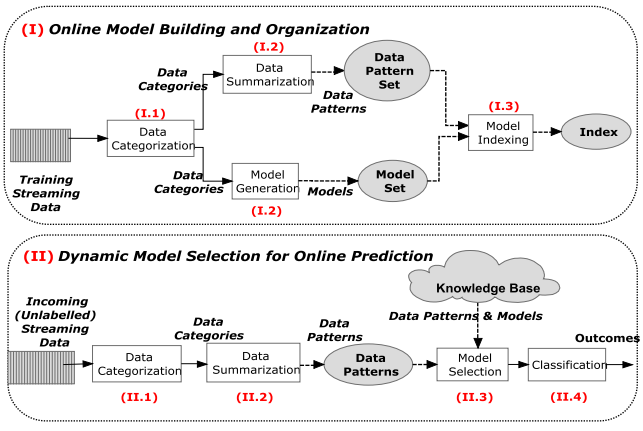


Figure 2. Overview of the model indexing approach.

Figure 2 shows an overview of the online training and prediction processes. During the *online model building and organization* process, there is a sequence of operations on the training streaming data: (I.1) the *data categorization* partitions each the training streaming data into data categories of equal size; (I.2) a base model is trained on each individual category, while the *data patterns* of each category is also summarized; (I.3) the base models are indexed to their corresponding *data patterns*, resulting in the *index*.

During the *dynamic model selection for online prediction* process, there is a sequence of operations on incoming unlabelled streaming data: (II.1) a similar data categorization strategy is used to partition the streaming unlabeled instances into categories; (II.2) the *data pattern* of each data category is summarized using the same strategy as that used in (I.2); (II.3) appropriate historical base models are selected by comparing the target *data pattern* with the historical *data patterns* in the *knowledge base*. The models are dynamically weighted by how much their *data patterns* are similar to a target *data pattern*; (II.4) the selected models are combined with calculated weights to classify the corresponding data category.

3.3 Algorithms for Online Model Building and Indexing

The key challenge in the online model building and indexing process is to choose an appropriate exploratory data analysis technique in the data summarization. Note that in the context of data streams, one may not assume the data can be retained for long and accessed in multiple runs. In our work, *principal component analysis* (PCA) [17] is used to roughly represent $P_{D_t}(X)$. There are three benefits: (1) PCA is able to reduce the dimensionality of the dataset into a smaller number of *principal components*; (2) the *principal components* reveals the internal structure of the data in a way which best explains its spatial orientation; (3) the similarities between the *principal components* are measurable. Figure 3 outlines the algorithm. We sequentially fetch categories of training examples of fixed size to train models and to summarize the *data patterns*. In the work, J48 is used as the base model and *principal components* are used the *data pattern*.

Algorithm 1 *Categorize-Index*
Input:
stream *: a training data stream;*
m: *a fixed size of each data category;*
Output:
knowledgeBase; *the* ***knowledge base*** *used to store data patterns, models and index;*
Begin
category ← ∅; // *Initializing an empty* ***category*** ;
do
category ← *sequentially fetching* ***m*** *training data from* ***stream*** ; // *the data categorization*
model ← *training a new model on* ***category*** *using* J48 ; // *the model generation*
dataPattern ← *summarizing the principle components from* ***category*** *using PCA ;* // *the data summarization*
add ***model*** *and* ***dataPattern*** *into* ***knowledgeBase ;***
indexing ***model*** *to* ***dataPattern ;*** //*the model indexing*
end for
*unitl (****dataset*** ==∅)
return ***knowledgeBase***;
End.

Figure 3. Online model building and indexing process

3.4 Algorithms for Dynamic Model Selection

The key challenge in the *dynamic model selection for online prediction* process is to choose an appropriate

similarity measure for comparing the *data patterns*. There have been a number effective PCA similarity measures in the literature [18; 19]. In our work, we choose the popular PCA Similarity Factor [17] as the similarity measure. It compares any two groups of *principal components* by calculating the angles between them. Figure 4 outlines the process. Figure 5 presents the comparison procedure. It is noted that the chosen models are the most similar to target variable distributions in terms of the similarities of their corresponding *principal components*. The models are also weighted by their similarity values respectively.

Algorithm 2 ***Select-Classify***
Input:
stream: *an unlabelled data stream;*
m *; a fixed size of each data category;*
N: *number of models to choose ;*
knowledgeBase: *the* ***knowledge base*** *used to store all the data patterns, models and index;*
Begin
category ← ∅ ; *// Initializing an empty* ***category*** ;
do
category ← *sequentially fetching* ***m*** *unlabelled data from* ***stream*** ; *// the data categorization*
targetPattern ← *summarizing the principle components from* ***category*** *using PCA ; // the data summarization*
// ---- the model selection begins --- //
patterns ← *selecting first* ***N*** *data patterns which are the most similar to* ***targetPattern*** *; // the similarity value is measured by Algorithm 3*
models ← *retrieving the corresponding* ***N*** *models indexed to* ***patterns*** *in* ***knowledgeBase***;
weighting ***models*** *by their corresponding similarity values ;*
// ---- the model selection ends ----- //
applying ***models*** *to classificaiton of* ***category*** *; // the classification*
unitl (***dataset*** *== ∅)*
End

Figure 4. Online dynamic model selection process

Algorithm 3 ***Measure***
Input:
// the ***pattern*** *and* ***targetpattern*** *are represented as the principle components*
targetPattern *: a target data pattern;*
pattern *: an historical data pattern;*
Output:
similarity: *a numeric value between* 0 *and* 1 *representing the consistence*
Begin
similarity ← *using PCA similarity Factor to measure* ***targetPattern*** *and* ***pattern ;***
return ***similarity*** *;*
End

Figure 5. Measuring similarities of the *data patterns*

3.5 Comparing the Efficiency between MI and WBA

We analyze efficiency of the proposed approach (MI) by comparing it with the other ensemble model method (WBA) mentioned previously.

Firstly, we show that given the same number of base models, when the data chunk (category) size increases[1], WBA shows constant classification time, while MI costs classification time inversely-proportional to the data category size.

More specifically, let m_t be the number of test instances, s be the data chunk size and m_b be the number of base models.

WBA. Given m_t test instances, there are $\frac{m_t}{s}$ test chunks. Before each classification of a test chunk, any base model in WBA is validated on s validation instances. That is, there are $m_b * s$ instances to be validated before one classification. Thus, the total count of validation instances is $\frac{m_t}{s} * m_b * s = m_t * m_b$. Note the validation data chunk size is equal to the test data chunk size.

MI. Given m_t test instances, there are $\frac{m_t}{s}$ test chunks. Before each classification of a test chunk, any base model in MI is validated on s instances. That is, there are $m_b * s$ instances o be validated before one classification. Thus, the total count of validation instances would be $\frac{m_t}{s} * m_b * s = m_t * m_b$. Note the validation data chunk size is equal to the test data chunk size.

As the performance validation or similarity measure is the key operation in WBA or MI respectively. Their running times can generally determine the classification time complexity of their corresponding approaches.

Secondly, it can be easily observed that both methods consume comparative raining time, although MI contains extra operations for building principle components and indexing models. Note that PCA can be computationally-efficient [20] (cf. our experimental analysis section below).

4 Experiments

We experimentally demonstrate effectiveness and efficiency of MI on data streams with *sample selection bias*. We compare MI with WBA, CVFDT and another *incremental data mining* algorithm, UNB. We analyze MI from the following three perspectives: (1) we demonstrate that although the proposed PCA-based model selection strategy is simple, they are effective and efficient; (2) in particular, we examine advantages of PCA-based model selection and weighting strategy over the weight-by-accuracy model selection and weighting strategy to ease *sample selection bias*; (3) we further examine properties of classification effectiveness and/or efficiency of MI by varying its algorithm parameters and datasets.

4.1 Data Streams

We create synthetic datasets and use the KDDCup'99 *network intrusion detection* dataset in our experiments.

--Synthetic datasets can simulate specific properties we require in experiments, thus enable us to clearly observe the specific behaviours of our approach under different experimental parameters. We apply the widely-used Random RBF stream benchmark [21; 22] to simulating *sample selection bias*. In the datasets, a fixed number of distant centres are built. Along with the time, the biased examples are created by continuously selecting a centre in a dominate way and then randomly switching to next centre. We simulate a number of these datasets by varying the number, d, of the centers, the number, v, of variables and the number, c, of class labels. In each dataset, there are 100,000 instances;

[1] The data chunk in WBA is called the data category in MI.

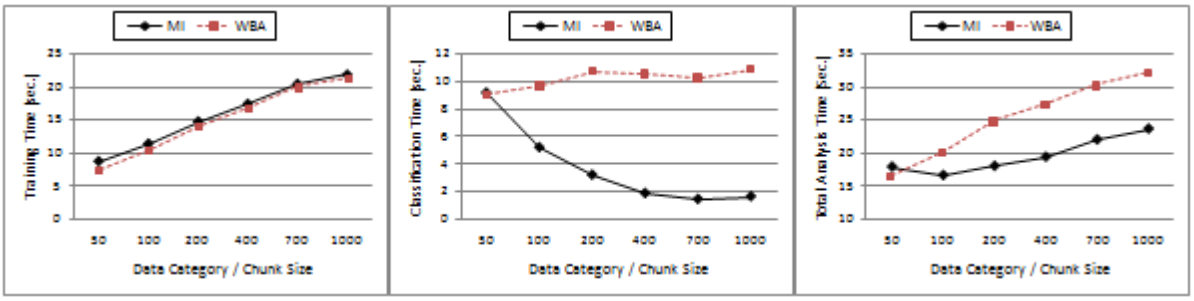


Figure 6. Comparison of training and classification time (seconds) of MI and WBA.

--We also test our approach on the *network intrusion detections* domain. We apply KDDCup'99 *internet intrusion detection* dataset [6] in our experiments. The dataset contains 100,000 instances randomly selected from the original dataset but retains all of its intrusion types.

4.2 Algorithm Opponents

State-of-the-art stream mining algorithms are chosen as the opponents. There are two incremental model algorithms and one ensemble model method, as follows.

--CVFDT [9] is a famous decision tree algorithm for mining evolving data streams. It is reported in KDD'01 and has been widely used as a benchmark stream mining algorithm in many studies [3; 23; 24]. The algorithm has been implemented in C on open-source VFML [25]. We use its default algorithmic settings[2] on VFML. The parameter settings are also aligned with its reporting paper;

--UpdatableNaiveBayes (UNB) is an incremental version of *Naïve Bayes*. It is implemented in Java on WEKA [26]; The presented results of UNB are the best evaluation results;

--WBA [14] is an excellent ensemble model approach for mining non-stationary data streams. It is reported in KDD'03 and has been widely considered a benchmark stream mining algorithm in related work [3; 4].. We implement it in Java.

4.3 Results on Synthetic Data

We first study effectiveness and efficiency of MI by comparing it with the opponents on synthetic datasets. J48 is used as base model for MI and WBA in the experiments.

4.3.1 Time Complexity Analysis

A series of experiments compare analysis performance and time complexity of MI with WBA on the RBF stream with $d = 50$ ($v = 10$ and $c = 2$). We vary the data chunk size from 50 to 1000 and set $N = 8$ in Algorithm 2.

Figure 6 compares the time complexity of the two methods. As can be seen, MI generally costs less analysis time than WBA. This advantage becomes more substantial when the chunk size becomes larger. More specifically,

--On one hand, MI consumes slightly more training time than WBA. This is due to the extra operations of MI for summarizing *principal components* and indexing the base models to the *principal components*. In addition, when the data chunk size increases, both show increasing training time in a near-linear manner. This result complies with WBA's reporting paper [14].

--On the other hand, MI achieves substantially lower classification time compared to WBA. When the chunk size increases, WBA shows no substantial change of its classification time but WBA becomes less time-consuming. The results comply with the analysis mentioned previously.

4.3.2 Classification Accuracy Analysis

Figure 7 examines the error rate of MI and WBA under the same settings as above. The results demonstrate that, MI outperforms WBA at every data category/chunk size and on every data stream in terms of classification error rate. When data category/chunk size is small, MI produces much lower error rate than WBA. For example, there is 31.02% (*i.e.* (29.40-20.28)/29.40) less error rate when the chunk size is 50. As the size grows bigger, both error rates decrease and converge gradually to each other. These experimental observations give us two insights about properties in the proposed model selection strategy over the weight-by-accuracy strategy in WBA.

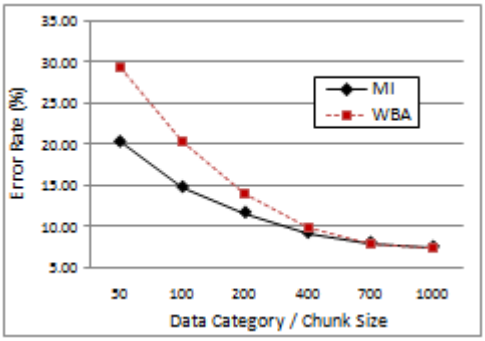


Figure 7. Comparison of effectiveness of MI and WBA.

--Under *sample selection bias*, when the data chunk size is small, the data distribution discrepancy between validation and incoming chunks is more likely to be large (*i.e.* the validation sample tends to be very biased). This leads to the validation accuracy of a historical base model is more likely to differ from its actual accuracy on the incoming instances. Thus, WBA suffers. However, as the proposed model selection strategy is performed by comparing actual variable distributions of incoming chunks with historical ones, it can largely reduce this effect.

--Provided that there are an increasing number of examples in each data chunk, observations of the underlying data distributions tends to be complete. Thus, the base models tend

[2] We test on a number of different parameter settings, among which the default parameters produces best performances.

to be more accurate and the validation data distributions tend to be similar to incoming data distributions. This leads to not only more accurate base models can be selected in MI, but also the validation accuracy of a base model tends to be close to its classification accuracy in WBA. Thus, both methods tend to agree with each other about the classification accuracy as the data category / chunk size increases.

4.3.3 Impact of Ensemble Size

Figure 8 examines the impact of ensemble size (*i.e.* the number, N, of models chosen in Algorithm 2) on error rate and time complexity of MI. The experiment is carried on the RBF stream with $d = 50$ ($v = 10$ and $c = 2$). J48 is used as the base model. The category size is set to 400. We vary the ensemble size from 4 to 12. The results indicate that, when N increases, the error rate reduces, while the classification time gradually increases. In addition, the error reduction is dramatic from 4 to 6, while it becomes flatter when N tends to be larger than 10.

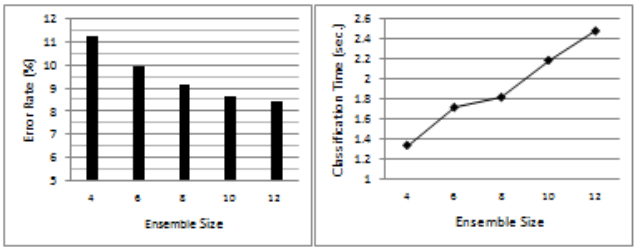


Figure 8. Impact of ensemble size on error rate (%) and classification time (seconds) of MI.

4.3.4 Impact of Ensemble Size

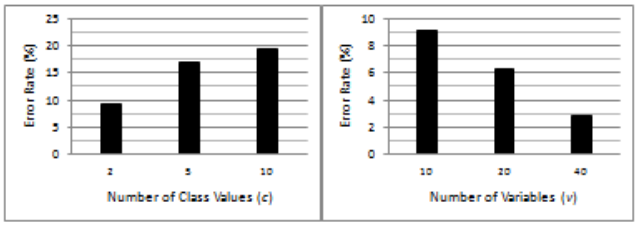


Figure 9. Impact of dataset complexity on error rate of MI.

Figure 9 studies the impact of number of variables (v) or class labels (c) on error rate of MI. In the experiments, we change v and c respecitvely and keep the other parameters fixed: the category size is set to 400; the ensemble size is set to 8; there are 50 centers ($d = 50$) in the data stream. As can be seen, the error rate rises as the number of class labels increases, while the error rate reduces as the number of variables grows. This is because the base models become increasingly accurate on the datasets of an increasing number of variables and/or a decreasing number of class labels.

4.3.5 Comparing with Incremental Mining Algorithms

We also compare the effectiveness and efficiency of MI and WBA with CVFDT and UNB on the synthetic datasets. We create synthetic datasets with $d = 30,40,50,60, v = 10,$ and $c = 2$. We keep ensemble size to be 8 for MI, and vary the data chunk size from 100 to 1000 ($s = 100, 200,400,700,1000$) for MI and WBA. Table 1 presents the results, based on which the following three observations can be achieved.

--The results indicate that ensemble model methods are clearly less efficient than incremental algorithms. This is because the data analysis process of a single incremental algorithm may not only save substantial training time by incrementally updating the single model according to each incoming instance instead of learning from scratch on all historical dataset, but also save substantial classification time without the model selection and weighting operations. In contrast, in ensemble model methods, a great deal of time has to be consumed to continuously train new base models according to blocks of new labelled instances and to dynamically select and weight base models for classification.

--The results also indicate that ensemble model methods can be more effective than incremental algorithms. In addition, MI is able to significantly outperform all the others in terms of classification accuracy.

--MI is more efficient than WBA, as analyzed previously.

Table 1. Error rate (%) and analysis time (seconds) on synthetic datasets

Methods		RBF Streams							
		d =30		d =40		d =50		d =60	
		Error	Time	Error	Time	Error	Time	Error	Time
MI	s =100	8.56	15.99	12.47	17.02	14.76	8.84	18.91	16.60
	s =200	7.43	16.26	10.91	18.02	11.66	11.52	13.48	17.44
	s =400	6.26	19.26	8.76	21.59	9.19	14.88	11.32	19.33
	s =700	5.10	20.84	7.32	23.68	8.00	17.56	9.37	21.20
	s =1000	**4.67**	23.07	**6.62**	25.41	**7.48**	20.57	**8.64**	24.38
WBA	s =100	16.55	19.15	19.65	20.65	20.29	20.23	23.17	19.96
	s =200	11.00	22.67	11.08	24.87	14.08	24.94	16.09	24.57
	s =400	6.27	27.35	8.55	30.39	9.86	27.45	12.10	27.77
	s =700	5.18	28.97	7.39	32.13	7.99	30.30	9.58	29.53
	s =1000	4.71	31.46	6.63	34.64	7.48	32.20	8.79	33.05
UNB		18.47	2.53	24.85	2.42	27.97	2.39	25.02	2.34
CVFDT ǂ		10.59	**1.05**	16.04	**1.34**	15.53	**1.13**	18.87	**1.21**

ǂ As CVFDT is implemented in C and UNB is implemented in Java, CVFDT's analysis time is evaluated shorter than UNB.

4.4 Results on Network Intrusion Detection Domain

We also study the effectiveness and efficiency of MI by comparing it with the opponents on *network intrusion detection* domain. J48 is used as base model for MI and WBA in the experiments. We vary the data chunk size from 100 to 1000 and set $N = 8$ in Algorithm 2.

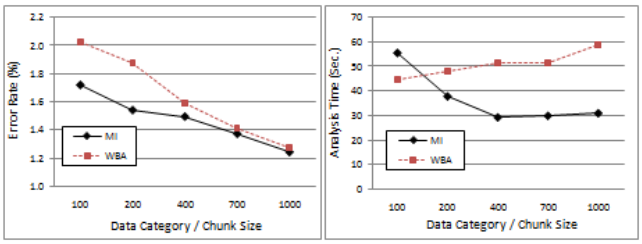


Figure 10. Comparison of error rate (%) and analysis time (seconds) of MI and WBA on KDDCup'99.

4.4.1 Comparison with WBA

Figure 10 compares the classification error rate and analysis time consumed in classification and training of MI and WBA. The results generally comply with those on synthetic datasets.

--When the data chunk size is small, MI produces lower error rate than WBA. For example, there is 14.9% less error rate when the chunk size is 100. As the size grows bigger, both error rates decrease and converge gradually to each other.

--MI consumes less time than WBA. This advantage becomes more substantial when the chunk size becomes larger.

4.4.2 Comparison with CVFDT and UNB

Table 2 compares the effectiveness and efficiency of MI and WBA with CVFDT and UNB. The results still indicate ensemble model methods are less efficient but more effective than single incremental algorithms. MI is clearly the best in terms of classification error rate.

Table 2. Error rate (%) and analysis time (seconds) on KDDCup'99

	MI		WBA		CVFDT ǂ		UNB	
	Error	Time	Error	Time	Error	Time	Error	Time
s = 100	1.72	55.35	2.02	44.71	15.45	0.74	1.50	38.4
s = 200	1.54	37.73	1.87	48.24				
s = 400	1.49	29.22	1.59	51.69				
s = 700	1.37	29.85	1.41	51.59				
s = 1000	1.25	30.90	1.27	58.85				

ǂ As CVFDT is implemented in C and UNB is implemented in Java, CVFDT's analysis time is evaluated shorter than UNB.

5 Related Work

Supervised stream mining is greatly different from classic batch (*supervised*) *learning* scenarios. In the latter, once the models are built, they are applied to test examples stably without considering (or knowing) any possible actual class labels of the test instances during classification. However, in the former, the actual class labels of the incoming unlabelled data can be continuously retrieved as soon as the class labels are available. This feature enables state-of-the-art stream mining methods effective: the models built on historical training distributions can be continuously updated or new models can be built based on these newly-labeled data in order to effectively adapt to or track the evolving *concept* [3; 4; 8; 9; 30; 31]. Based on this feature, we categorize state-of-the-art supervised stream mining approaches into two kinds of: *tracking* and *adaption*.

The *tracking* approach passively renews models after stimulated by certain *concept* change detection mechanisms. These detection mechanisms continuously receive the most recent labeled streaming instances and attempt to monitor certain changes caused by these instances during their online prediction processes. Examples of these algorithms include the FLORA family [32], the OLIN series [8; 30], and the VFDT series [9; 31; 33].

The *adaption* approach automatically adapts to the changing *concept* by updating its models without performing obvious *concept* change detections. However, this kind of algorithms is still relied on the online retrieved labeled examples during their online prediction processes. WBA is a well-known example of this kind. Other algorithms include RePro [3], High-order [4], Bifet's online *Bagging* algorithms [15].

6 Conclusions and Contributions

6.1 Conclusions

We summarize the experimental results in this subsection. The experimental results demonstrate that MI is able to improve classification accuracy under *sample selection bias* in the context of data streams, compared to state-of-the-art ensemble model approaches and incremental mining algorithms including WBA and CVFDT. MI is also capable of improving classification efficiency compared to other ensemble model approaches. More specifically, the following conclusions can be achieved.

--Compared to WBA, MI is able to reduce impact of biased training samples resulting from the *sample selection bias* problem and therefore to improve classification accuracy.

This advantage can be more substantial when data chunks are smaller (*i.e.* samples are more biased).

--MI is able to outperform the weight-by-accuracy strategy of WBA in terms of efficiency. Although MI consumes slightly more training time for summarizing *principal components* and indexing the base models to the *principal components*, the PCA-based model selection and weighting strategies greatly reduce classification time costs compared to WBA.

--MI (and other ensemble model methods) is able to outperform single incremental algorithms in terms of classification accuracy. However, the data analysis process of a single incremental algorithm can be faster than ensemble model methods including MI.

--MI can be more effective if the ensemble size (*i.e. N* in algorithm 2) becomes bigger. In addition, MI demonstrates rising classification accuracy as the number of class labels in the dataset reduces or as the number of variables increases (*i.e.*, as the dataset complexity reduces).

6.2 Contributions

Sample selection bias has been widely studied on classic datasets, but receives little attention in online data stream mining. This paper studies the *sample selection bias* problem in the context of data streams and proposes a new ensemble model approach for easing the *sample selection bias* problem. The contributions lie in three folds.

-- We formally distinguish this problem with the well-known *concept drift* problem in the literature. We argue that in the context of non-stationary data streams, *concept drift* is due to changes of data-generating mechanisms and can be fully expressed as changes of the posterior probabilities, while *sample selection bias* is due to the limitations of data-generating mechanisms and can be fully expressed as changes of variable distributions only. The clear understandings of

these different problems helps design better algorithms and evaluate the performance of these algorithms.

--Addressing *sample selection bias* on data streams, we introduce the strategy of structuring multiple historical data mining models in line with the variable distributions of their training datasets.

This strategy is beneficial for locating the most appropriate models for classification in terms of both effectiveness and efficiency.

--In addition, attempts have been devoted to predicting labels of test examples by first revealing their underlying data distributions (or the underlying *concept* or the underlying data generating mechanism) instead of making predictions of the labels only.

This strategy is beneficial for improving classification accuracy because data mining models are supposed to be more effective on its training data distributions (or training *concepts*) instead of on distinct data distributions

7 References

[1] Bianca, Z., "Learning and evaluating classifiers under sample selection bias," ACM, Banff, Alberta, Canada, 2004, pp. 114.

[2] Steffen, B., Michael, B., ckner, and Tobias, S., "Discriminative learning for differing training and test distributions," ACM, Corvalis, Oregon, 2007, pp. 81-88.

[3] Yang, Y., Wu, X., and Zhu, X., "Mining in Anticipation for Concept Change: Proactive-Reactive Prediction in Data Streams," *Data Mining and Knowledge Discovery*, Vol. 13, No. 3, 2006, pp. 261-289.

[4] Chen, Shixi, Wang, Haixun, Zheng, Shuigeng and Yu, Philip S., "Stop Chasing Trends: Discovering High Order Models in Evolving Data," IEEE Computer Society, 2008, pp. 923-932.

[5] A.Tsymbal, "The problem of concept drift: Definitions and related work," Department of Computer Science, Trinity College, Technical Report TCD-CS-2004-15, Dublin, 2004.

[6] Blake, C. and Merz, C.. UCI machine learning repository. 1998.

[7] Klinkenberg, R. and Renz, I., "Adaptive information filtering: learning in the presence of concept drift," *AAAI/ICML-8 Workshop on Learning for Text Categorization*, AAAI Press, Madison, WI., 1998, pp. 33-40.

[8] L.Cohen, G.Avrahami, and M.Last, "Incremental Info-Fuzzy Algorithm for Real Time Data Mining of Non-Stationary Data Streams," *TDM Workshop*, Brighton UK, 2004.

[9] Geoff, H., Laurie, S., and Pedro, D., "Mining time-changing data streams," ACM, San Francisco, California, 2001, pp. 97-106.

[10] Gama J., Medas, P, and Rodrigues, P, "Learning decision trees from dynamic data streams," ACM, Santa Fe, New Mexico, 2005, pp. 573-577.

[11] Gama.J, Medas, P., Castillo, G., and Rodrigues, P., "Learning with Drift Detection," 2004, pp. 286-295.

[12] Fan,Wei, "StreamMiner: a classifier ensemble-based engine to mine concept-drifting data streams," VLDB Endowment, Toronto, Canada, 2004, pp. 1257-1260.

[13] Fan, Wei, "Systematic data selection to mine concept-drifting data streams," ACM, Seattle, WA, USA, 2004, pp. 128-137.

[14] Wang, Haixun, Fan, Wei, Yu, Philip S., and Han, Jiawei, "Mining concept-drifting data streams using ensemble classifiers," ACM, Washington, D.C., 2003, pp. 226-235.

[15] Albert, B., Geoff, H., Bernhard, P., Richard, K., and Ricard, G., "New ensemble methods for evolving data streams," ACM, Paris, France, 2009, pp. 139-148.

[16] Broder, A., "A taxonomy of web search," *SIGIR forum*, Vol. 36, No. 2, 2002, pp. 10.

[17] Jolliffe, I. T., *Principal component analysis*, 2 ed., Springer-Verlag New York, Inc, New York, 2002.

[18] Krzanowski, W. J., "Between-groups comparison of principal components," *Journal of the American Statistical Association*, Vol. 74, No. 367, 1979, pp. 703.

[19] Singhal, A. and Seborg, D. E., "Matching patterns from historical data using PCA and distance similarity factors," *American Control Conference, 2001. Proceedings of the 2001*, Vol. 2, 2001, pp. 1759-1764.

[20] Sam, R., "Em algorithms for pca and spca," *Proceedings of the 1997 conference on Advances in neural information processing systems*, MIT Press, 1998, pp. 626-632.

[21] G.Holmes, , R. K., and B.Pfahringer.. MOA:Massive Online Analysis. http://sourceforge.net/projects/moa-datastream . 2007.

[22] Bifet, A., Holmes, G., Pfahringer, B., and GavaldÃ , R., "Improving Adaptive Bagging Methods for Evolving Data Streams," 2009, pp. 23-37.

[23] Bifet, A. and GavaldÃ , R., "Adaptive Learning from Evolving Data Streams," *Advances in Intelligent Data Analysis VIII*, edited by N. Adams, C. Robardet, A. Siebes, and J. F. o. Boulicaut Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2009, pp. 249-260.

[24] Wang, Haixun, Y., Jian, P., Jian, Yu, Philip S., and Yu Jeffrey, "Suppressing model overfitting in mining concept-drifting data streams," ACM, Philadelphia, PA, USA, 2006, pp. 736-741.

[25] Pedro Domingos and Geoff Hulten. VFML-A toolkit for mining high-speed time-changing data streams. http://www.cs.washington.edu/dm/vfml/cvfdt.html . 2003.

[26] Weka 3.7. http://www.cs.waikato.ac.nz/ml/weka/ . 2009.

[27] Phua, C., Lee, V., Smith, K., and Gayler, R., "A comprehensive survey of Data Mining-based Fraud Detection Research," *Artificial Intelligence Review*, 2005.

[28] Alexandr Seleznyov, Seppo Puuronen, and R Seleznyov, "Anomaly Intrusion Detection Systems: Handling Temporal Relations between Events," 1999.

[29] Lior, C., Gil, A., Mark, L., Abraham, K., and Oscar, K., "Real-time data mining of non-stationary data streams from sensor networks," *Inf.Fusion*, Vol. 9, No. 3, 2008, pp. 344-353.

[30] Last, M., "Online classification of nonstationary data streams," *Intelligent Data Analysis*, Vol. 6, No. 2, 2002, pp. 129-147.

[31] Gama J., Ricardo, R., and Pedro, M., "Accurate decision trees for mining high-speed data streams," ACM, Washington, D.C., 2003, pp. 523-528.

[32] Widmer, G., "Learning in the presence of concept drift and hidden contexts," *Machine Learning*, Vol. 23, No. 1, 1996, pp. 69.

[33] Pedro, D. and Geoff, H., "Mining high-speed data streams," ACM, Boston, Massachusetts, United States, 2000, pp. 71-80.

HIOPGA: A New Hybrid Metaheuristic Algorithm to Train Feedforward Neural Networks for Prediction

Masoud Yaghini[1], Mohammad M. Khoshraftar[2], Mehdi Fallahi[3]
[1]School of Railway Engineering, Iran University of Science and Technology, Tehran, Iran
[2] School of Railway Engineering, Company / University of Science and Technology, Tehran, Iran
[3] School of Railway Engineering, Company / University of Science and Technology, Tehran, Iran

Abstract - *Most of neural network training algorithms make use of gradient-based search and because of their disadvantages, researchers always interested in using alternative methods. In this paper to train feedforward, neural network for prediction problems a new Hybrid Improved Opposition-based Particle swarm optimization and Genetic Algorithm (HIOPGA) is proposed. The opposition-based PSO is utilized to search better in solution space. In addition, to restrain model overfit with training pattern, a new cross validation method is proposed. Several benchmark problems with varying dimensions are chosen to investigate the capabilities of the proposed algorithm as a training algorithm. The result of HIOPGA is compared with standard backpropagation algorithm with momentum term.*

Keywords: PSO, GA, Prediction, Hybrid Algorithm

1 Introduction

Neural network (NN) is one of the most important data mining techniques. It is used with both supervised and unsupervised learning [1]. Training NN is a complex task of great importance in problems of supervised learning. Most of NN training algorithms make use of gradient-based search. These methods have the advantage of the directed search, in that weights are always updated in such a way that minimizes the error, which called NN learning process. However, there are several negative aspects with these algorithm such as dependency to a learning rate parameter, network paralysis, slowing down by an order of magnitude for every extra (hidden) layer added and complex and multi-modal error space, Therefore, these algorithms most likely gets trapped into a local minimum, making them entirely dependent on initial (weight) settings which make the algorithms not guaranteed to be universally useful [2]. Metaheuristic global search strategy makes them able to avoid being trapped into secondary peak of performance and can therefore provide effective and robust solution to the problem of NN and training [3]. Metaheuristics have the advantage of being applicable to any type of NN, feedforward or not, with any activation function, differentiable or not [2]. Metaheuristics provide acceptable solutions in a reasonable time for solving hard and complex problems; they are particularly useful for dealing with large complex problems, which generate many local optima. They are less likely to be trapped in local minima than traditional gradient-based search algorithms. They do not depend on gradient information and thus are quite suitable for problems where such information is unavailable or very costly to obtain or estimate [4]. The outline of this paper is as follows. Section 2 presents literature review about metaheuristic algorithm for training NN. In section 3, the proposed particle and chromosome, criterion for accuracy evaluation, component and operator of the proposed algorithm, the proposed cross validation, steps of the algorithm, and the termination criterions is completely described. In section 4, experimental results, the value of parameter and convergence graph is presented. In section 5 summery, conclusion and some hints for the future research is given.

2 Literature Review

Metaheuristic algorithms for training NN could divide into single-solution based and population-based algorithms (S-Metaheuristic and P-Metaheuristic). In training NN with S-Metaheuristic [5], [6] used tabu search approach and [7], [8] used simulated annealing approach. One could divide NN training with P-Metaheuristic into two main groups, which are train with Evolutionary Algorithms (EA) and train with swarm intelligence algorithms, respectively. Learning and evolution are two fundamental forms of adaptation. There has been a great interest in combining learning and evolution with NN and combinations between NN's and EA's can lead to significantly better intelligent systems than relying on NN's or EA's alone [9]. In Training NN with EA [10] and [11] make a comparison among proposed EA and a gradient-based algorithm, [12] and [13] combine EA with gradient-based local search algorithm to obtain better result. Another class of P-Metaheuristic, which is used as training algorithm, is swarm intelligence. They originated from the social behavior of those species that has a common target (e.g. compete for foods) [4]. Among swarm intelligence inspired optimization algorithms Particle Swarm Optimization (PSO) is one the most successful one. Unlike Genetic Algorithm (GA), PSO has no complicated evolutionary operators such as crossover, selection, and mutation and it is highly dependent on stochastic processes [2]. The PSO was introduced by [14] for the first time. [15] proposed a method to employ PSO in a cooperative configuration which is achieved by splitting the input vector into several sub-vectors, each which is optimized cooperatively in its own swarm.[16] and

[17] make use of PSO to train neural network. In these research authors just use a very simple problem that did not reveal outperformance of their method.[18] presents a modified PSO which adjust the trajectories (positions and velocities) of the particle based on the best positions visited earlier by themselves and other particles, and also incorporates population diversity method to avoid premature convergence. [19] analyzes the use of the PSO algorithm and two variants with a local search operator. [20] use multi-phase PSO algorithm (MPPSO) which simultaneously evolves multiple groups of particles that change their search criterion when changing the phases, and also incorporates hill-climbing.

In addition to the modifications made to basic PSO algorithm, a variety of other PSO variations have also been developed. Among these variations are those which incorporate opposition-based learning into PSO is capable of delivering better performance as compared to the standard PSO. Opposition-based learning was first introduced by [21] later applied to PSO. Opposition-based learning is based on the concept of opposite points and opposite numbers. [22] proposed a modified PSO algorithm for noisy problems which utilized opposition-based learning. [23] proposed an opposition-based comprehensive learning PSO which utilized opposition-based learning for swarm initialization and for exemplar selection. [24] Presented the improved PSO which utilized a simplified form of opposition-based learning. In this approach, the particle having worst fitness in each iteration is replaced by its opposite particle. Opposition-based learning was only applied to one particle instead of the whole swarm and was also not used at the time initialization. Apart from PSO researcher employed other swarm intelligence but none of the is successful as PSO. [25] presented a continuous version of ACO algorithm (i.e., ACO_R) also [26] proposed a novel hybrid algorithm based on Artificial Fish Swarm Algorithm and PSO both compare their proposed algorithm with specialized gradient based algorithms for NN training.

3 The Proposed Algorithm

3.1 Proposed Particle and Chromosome

A good detail on basic version PSO algorithm is in [27] and for GA is in [4]. In this research we employ fully connected layered feedforward networks. All units have a bias except for input units. In the proposed algorithm (HIOPGA) to utilize a combination of PSO and GA a structure as Fig. 1 is employed. For simplification in this figure a NN with one hidden layer, three input units, one hidden units and two output units is considered.

3.2 Criterion for Accuracy Evaluation

For classification problems, *classification error percentage* (CEP) is utilized as shown in (1) and (2) to evaluate the accuracy. op and tp are predicted value and target value, p is input pattern and P is the number of pattern.

$$\psi(\vec{p}) = \begin{cases} 1 & if \quad \vec{o}_p \neq \vec{t}_p \\ 0 & otherwise \end{cases} \tag{1}$$

$$CEP = 100 \times \left(\sum_{p=1}^{P} \psi(p) \Big/ P \right) \tag{2}$$

For approximation problem *Normalized Root Mean Squared Error* (NRMSE) is utilized as shown in (3) and (4).where N is number of the output units, P number of pattern, opi and tpi are predicted value and target value of ith output unit for pattern p.

$$RMSE = \sqrt{\sum_{p=1}^{P}\sum_{i=1}^{N} \left(t_{pi} - o_{pi}\right)^2 \Big/ P \times N} \tag{3}$$

$$NRMSE = 100 \times \frac{RMSE}{\sum_{p=1}^{P}\sum_{i=1}^{N} t_{pi} \Big/ P \times N} \tag{4}$$

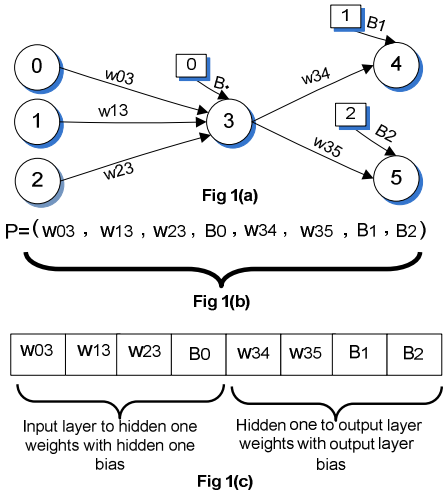


Fig. 1. (a) a NN structure (b) the particle for PSO (c) the chromosome for GA.

3.3 Improved PSO

Although PSO is capable of locating a good solution at a significantly fast rate, but its ability to fine tune the optimum solution is comparatively weak, mainly due to the lack of diversity at the end of the evolutionary process. To improve the search ability of standard PSO time-varying parameter is utilized. Suppose that t and m is current and final iteration number and $C_1(t)$, $C_2(t)$, $C_1(m)$ and $C_2(m)$ are cognitive and social component of current and final iteration then time-varying parameter is calculated as (5) and (6). If each of parameter reaches to final values, it set to initial value again. By using the time-varying parameter, we can implement large cognitive component and small social component at the beginning of the search to guarantee particles' moving around the search space and to avoid particles moving toward the population best position. On the other hand, a small cognitive

component and a large social component allow the particles to converge to the global optima in the latter of the search [28]. *K* is another parameter that utilized along with these parameter and called *constriction coefficient* with the hope that it can insure a PSO to converge [29]. *K* is calculated as (6), $\phi(t)=C_1(t)+C_2(t)$ and $\phi(t)\geq 4$.

$$C_1(t+1)=\frac{t}{m}\times\left(C_{1m}-C_1(t)\right)+C_1(t) \quad (4)$$

$$C_2(t+1)=\frac{t}{m}\times\left(C_{2m}-C_2(t)\right)+C_2(t) \quad (5)$$

$$K(t)=2\Big/\left|2-\phi(t)-\sqrt{\phi^2(t)-4\phi(t)}\right| \quad (6)$$

3.4 Opposition-based Learning Components

The proposed *HIOPGA* implement this method two ways. First, after population initialization, to start with a better population, the algorithm calculate the opposite position and velocity of each particle, then for each particle the better one (current particle or its opposite) is inserted into the population. Second, during the iteration, when the algorithm finds a new velocity and position for a particle, the opposite position and velocity of each particle is calculated and the better one is inserted into the current generation. When creating opposite particles an important question that arises is, what should be the velocity of these particles? Either we can have the same velocity as that of the original particle or we can randomly reinitialize the velocity. Alternatively, we can calculate the opposite of the velocity of the original particle. We cannot use the velocity of the original particle because that velocity was calculated using the current position of the original particle which would be invalid for the opposite particle. Reinitializing the opposite particles velocity randomly is not such an inviting option because we would not be taking advantage of the experience gained by original particle. Other researchers have not investigated this question and use random initialization of velocity. We have decided to use the opposite velocity of the original particle. We believe that by using opposite velocity we would be able to achieve better performance as we do with utilizing opposite positions. The opposite velocity is calculated in exactly the same way as we calculate the opposite particles. The pseudocode of opposite particle calculation is illustrated in Fig 2. [x_{min}, x_{max}] is the initial interval of the particle position (initial weight of NN) and [v_{min}, v_{max}] is the velocity interval. The poisons and velocity of *i*th particle at iteration *t* are $X_i(t)=(x_{i1}(t),\ldots,x_{id}(t))$ and $V_i(t)=(v_{i1}(t),\ldots,v_{id}(t))$.

For all particles *i*
 For all dimensions *d*
  opposite position: $\bar{x}_{id}(t)=x_{\max}+x_{\min}-x_{id}(t)$
  opposite velocity: $\bar{v}_{id}(t)=v_{\max}+v_{\min}-v_{id}(t)$
  if($\bar{v}_{id}(t)>=v_{\max}$) **then** $\bar{v}_{id}(t)=v_{\max}$
  if($\bar{v}_{id}(t)<=v_{\min}$) **then** $\bar{v}_{id}(t)=v_{\min}$
  if ($E(\bar{X}_i(t))<E(X_i(t))$) **then** $\bar{x}_{id}(t)=\bar{x}_{id}(t)$ **and** $v_{id}(t)=\bar{v}_{id}(t)$;
  Update $(x_i(t),v_i(t))$;
 End for *d*
End for *i*

Fig. 2. Pseudocode of opposite particle calculation

3.5 Random Perturbation

PSO can quickly find a good local solution but it sometimes suffers from stagnation without an improvement [28]. Therefore, to avoid this drawback of basic PSO, the velocity of particles is reset in order to enable particles to have a new momentum. Under this new strategy, when the global best position is not improving with the increasing number of generations, each particle *i* will be selected by a predefined probability (0.5 in this study) from the population, and then a random perturbation is added to each dimension v_{id} (selected by a predefined probability 0.5 in this study) of the velocity vector v_i of the selected particle *i*. The velocity resetting is presented as follows: where r_1, r_2 and r_3 are separately generated, uniformly distributed random numbers in range (0, 1), and v_{max} is the maximum magnitude of the random perturbation to each dimension of the selected particle.

For all particles *i*
Generate a random number($r_1\in[0,1]$)
 If (r_1>0.5) **then** select particle *i*;
  For all dimensions of selected particle *i*
   Generate a random number ($r_2\in[0,1]$)
   If (r_2>0.5) **then** $vi_d=v_{\max}\times(2r_3-1)+v_{id}$
   If ($vi_d>V_{\max}$) **then** $vi_d=V_{\max}$
   If($\bar{v}_{id}(t)<=v_{\min}$) **then** $\bar{v}_{id}(t)=v_{\min}$
  End for *d*
End for *i*

Fig. 3. Pseudocode of opposite particle calculation

3.6 Evolutionary Operators

All illustrations, on some evolutionary schemes of GA, several effective mutation and crossover operators have been proposed for PSO. [30] proposed a crossover operator, and [31] proposed a Gaussian mutation operator to improve the performance of PSO. Utilizations of these operators in PSO have potential to achieve faster convergence and to find better solutions. During iteration of HIOPGA, if the best personal position for each particle *i* ($Pbest_i$) is not improved for *maxPbestPersistence* successive iteration, we suppose that these particles are get stuck in local minima of the problem. Therefore, crossover and mutation operator is utilized to improve the performance of algorithm and obviate the aforementioned problem. These trapped particles establish a sub-population of particle. Then according to Stochastic Universal Sampling (SUS) two individual are selected. In addition, a crossover point is selected randomly. For example, for a NN with one hidden layer, crossover point is a number between 1 and 2, for a NN with two hidden layers crossover point is a number between 1 and 3, and for a NN with three hidden layer crossover point is a number between 1 and 4. To make new offspring if 1 is considered as crossover point, then connections between input layer and hidden layer one is used for crossover operator. Suppose $parent_1(x_i)$ and $parent_2(x_i)$ is the *i*th component of selected individuals. The crossover operator is conducted by the (7) for position crossover and the (8) for velocity crossover ($r_i\in(0,1)$).

$$Child_1(x_i) = r_i \times Parent_1(x_i) + (1-r_i) \times (Parent_2(x_i))$$
$$Child_2(x_i) = (1-r_i) \times Parent_1(x_i) + r_i \times (Parent_2(x_i)) \quad (7)$$

$$Child_1(v_i) = |Parent_1(v_i)| \times \frac{(Parent_1(v_i) + Parent_2(v_i))}{|Parent_1(v_i) + Parent_2(v_i)|}$$
$$Child_1(v_i) = |Parent_2(v_i)| \times \frac{(Parent_1(v_i) + Parent_2(v_i))}{|Parent_1(v_i) + Parent_2(v_i)|} \quad (8)$$

In each generation, the mutation operator is conducted by the (9), $r_i \in (-1,1)$. $(m\text{-}t)/m$ makes algorithm to have great jump at the early phase of algorithm and small jump at the latter phase.

$$Child_1(x_i) = \frac{m-t}{m} \times r_i \times x_{\max} + Parent(x_i)$$
$$Child_1(v_i) = \frac{m-t}{m} \times r_i \times v_{\max} + Parent(v_i) \quad (9)$$

3.7 The Proposed Cross Validation Method

The training error of an NN may reduce as its training process progresses. However, at some point, usually in the later stages of training, the NN may start to take advantage of idiosyncrasies in the training data. Consequently, its generalization performance may start to deteriorate even though the training error continues to decrease. Early stopping in cross validation [32] is one common approach to avoid overfitting. In this method, the training data is divided into training and validation sets. The training process will not terminate when the training error is minimized instead it stopped when the validation error starts to increase. This termination criterion is deceptive because the validation set may contain several local minima. In HIOPGA to decrease negative effect of multimodal validation space on model generalization ability, we use a simple criterion that terminates the training process of the NN. At the end of each L training iterations, the validation error is evaluated and the process repeated, when this error increases for T successive times in comparison to first L training iterations (independent of how large the increases actually are), training process is terminated. The idea behind the termination criterion is to stop the training process of the NN when its validation error increases not just once but during T consecutive times. It can be assumed that such increases indicate the beginning of the final overfitting not just the intermittent.

3.8 The Steps of HIOPGA

The steps of HIOPGA are explained as follows.

Step1) according to initial value of parameters, specify starting position and velocity of particles. Set iteration counter to zero (*iter* =0).

Step 2) to establish a better population, calculate opposite position and velocity of particles, and insert better particle into population (Pseudocode in Fig. 2).

Step 3) for each particle specify best personal position and calculate the number of no improvements in it ($pbest_i$ and $pbest_i Counter$). Also, specify best global position and calculate the number of no improvements in it (*gbest* and *gbsetCounter*).

Step 4) if ($E_{train}(gbest(iter)) < \mathcal{E}$) then go to step 28, otherwise go to Step 5.

Step 5) calculate the best global position of particle validation error ($E_{val}(gbest(iter))$.

Step 6) according to equation (4), (5), and (6) calculate $C_1(iter+1)$, $C_1(iter+1)$ and $K(iter+1)$.

Step 7) calculate new position and velocity of particles (training by PSO).

Step 8) for each particle, specify best personal position and calculate the number of no improvements in it ($pbest_i$ and $pbest_i Counter$). Also, specify best global position and calculate the number of no improvements in it (*gbest* and *gbsetCounter*).

Step 9) if ($E_{train}(gbest(iter)) < \mathcal{E}$) then go to Step 28 otherwise go to Step 10.

Step 10) do the proposed cross validation.

Step 11) identify a sub-population for genetic algorithm by specifying the particles that number of no improvements in the best personal position is greater than maximum allowed number ($pbest_i Counter$>max*Pbest*) and go to Step12.

Step 12) if the number of individuals in sub-population is grater than 1+[*m*/*iter*] then go to Step 13, otherwise go to Step 23.

Step 13) set genetic counter to zero (*GeneticCounter*=0) **Step 14)** select two individuals by using SUS methods.

Step 15) call crossover operator using equation (7) and (8), and the parents are replaced with their better offsprings in main population.

Step 16) call mutation operator using equation (9) and the new better offspring take the place of its parents in main population.

Step 17) if number of individual in the new generation is equal to number of individual in the current generation go to Step 18, otherwise, go to Step 14.

Step 18) establish next generation with best individual in the current and new generation and add one to genetic algorithm counter (*GeneticCounter*++)

Step 19) if genetic algorithm counter is equal to number of individual in the sub-population go to Step 20, otherwise go to Step 14.

Step 20) for each particle, specify best personal position and calculate the number of no improvements in it ($pbest_i$ and $pbest_i Counter$). Also, specify best global position and calculate the number of no improvements in it (*gbest* and *gbsetCounter*).

Step 21) if ($E_{train}(gbest(iter)) < \mathcal{E}$) then go to Step 28, otherwise, go to Step 22.

Step 22) if the number of no improvements in the best global position is greater than maximum allowed number (*gebestCounter*>*Maxgbest*) then call random perturbation (Pseudocode in Fig. 3)

Step 23) increase iteration counter (*iter*++).

Step 24) if iteration counter is greater than maximum allowed number (*iter*>*m*) then go to Step 28, otherwise, go to Step 25.

Step 25) if the remaining for number of iteration divided by the proposed cross validation strip length (*iter*/*L*) become zero go to Step 27, otherwise, go to Step 2.

Step 26) if validation error of global position for the current iteration is grater than pervious iteration ($E_{val}(gbest(iter))$> E_{val}

(*gbest*(*iter*-1)) then increase overtraining counter (*Tcounter*++), otherwise set it to zero (*Tcounter* =0).
Step 27) if overtraining counter is greater than maximum allowed number (*Tcounter*>*T*) then go to step 28 (termination because of overfitting), otherwise go to Step 2. **Step 28)** stop the training.

3.9 Termination Criterion

The algorithm simultaneous uses three criterions as termination conditions. First termination condition is based on training error. In this approach, at the end of each iteration *t* if the error on the training pattern is less than ε the training process will terminate (for the classification problems $\varepsilon = 10^{-2}$ for the approximation problems $\varepsilon = 10^{-6}$). Second, if number of iteration becomes greater than a predefined number, the training process will be terminated. Third, according to the proposed cross validation method, if the algorithm meets the over training condition, the training process will be terminated.

4 Experimental Studies

In this section, a comparison between the performance of HIOPGA and backpropagation algorithm (BP) with momentum term on several well-known benchmark problems is presented. The characteristics of these problems are summarized in Table 1, which show a considerable diversity in the number of examples, attributes, and classes. The detailed description of these problems can be obtained from the University of California Irvine, Machine Learning Repository. For each benchmark problem, entropy is calculated according to (10). Where $P(C_i)$ is the probability of class C_i in the data set, determined by dividing the number of pattern of class C_i by the total number of pattern in data set. Entropy of a data set is the average amount of information needed to identify the class label of a pattern in data set. In fact, entropy explores class distribution information in its calculation and show impurity of data set. It could considered as a criterion for difficulty of the problems.

$$E = -\sum_i P(C_i) \times \log_2 \left(P(C_i) \right) \qquad (10)$$

TABLE 1
CHARACTERISTICS OF BENCHMARK PROBLEMS

problem	Number of					Entropy
	Input attributes	Output classes	Training pattern	Validation pattern	Testing pattern	
Cancer	9	2	350	175	174	0.93
Card	51	2	345	173	172	0.99
Diabetes	8	2	348	192	192	0.93
Glass	9	6	107	54	53	2.18
Heart	35	2	460	230	230	0.99
Horse	58	3	182	91	91	1.32
Iris	4	3	75	38	37	1.58
Mushroom	125	2	4062	2031	2031	1
Thyroids	21	3	3600	1800	1800	0.45

4.1 Experimental Methodology

The proposed algorithm is implemented with Java programming language and a personal computer with Intel(R) Pentiume (R) CPU 2.66 GHz 2.68GHz, 32 Bits Windows 7 Ultimate operating system and 1.50 GB installed memory (RAM) is used to achieve all of the results. In both algorithm (HIOPGA and BP) we consider 150 as maximum number of iteration and observably both algorithm has less iteration to converge. The metaheuristic and gradient-based algorithms are sensitive to the value of their parameters. The parameters are the configurable components of HIOPGA and BP. Parameter tuning may allow a larger flexibility and robustness to the algorithm, but requires a careful initialization. Those parameters may have a great influence on the efficiency and effectiveness of the search. It is not obvious to define a priori which parameter setting should be used. The optimal values for the parameters depend mainly on the problem and even the instance to deal with and on the search time that the user wants to spend in solving the problem. One main step of this research is fine parameter tuning of the algorithm. To tune the parameters of each algorithm, by random we selected three benchmark problems with different sizes and only one parameter is modified at a time for each algorithm, while the other are not changed. Then proper values of the parameters determined through running the algorithm 15 times over different values of the parameters and calculating average of the objective function for these 15 runs. The criteria for modifying parameters are the quality of solutions and CPU time to find them. The final value for the parameters is as follows. Initial NN weights (initial position of particles or x_{min} and x_{max}) is [-1,1], the velocity interval is [-3,3], number of particles (n_p) is 15, initial and final value for cognitive component are 5 and 1, respectively. Initial and final value for social component are 1 and 5. GA mutation probability (P_m) is 0.02 and crossover probability (P_c) is 0.7. For the proposed cross validation method, *L* and *T* are 5 and 3. In addition, the learning rate and momentum term for BP are 0.1 and 0.9, respectively. The max number of training epochs, i.e., *m*, for both algorithms is set to 150. One bias neuron with a fixed input +1 was connected to the neurons of the hidden layers and output layers. The logistic sigmoid function was used for the neurons in the hidden layers and output layer.

4.2 Experimental Results

To reduce the effect of random parameter initialization on prediction ability of the models, we run each model 100 times, independently and take the average and standard deviation of results in table 2. The BP needs much more time and iteration to converge but less disperses solution. The reason for this matter is the random essence of metaheuristic algorithm.

4.3 The Convergence Graph

Fig. 4(a)-4(i) represent number of iteration-testing error for the best so far network from the beginning of the algorithms. As these figures reveal, the proposed HIOPGA with doing big jump in the testing space to find better solutions converge faster than BP.

5 Conclusion

In this research, a hybrid algorithm based on particle swarm optimization and genetic algorithm was presented. During iteration of the proposed algorithm, when some NN (or particles position) in the *d*-dimensional space cloud not be improved through the IOPSO, a sub-population of such NNs is established and be sent to the GA, to with utilizing the GA crossover and mutation operators, the HIOPGA finds better NN for replacing in the population. The comparison between the proposed algorithm and standard BP with momentum term reveals the superiority of the algorithm, however HIOPGA in the final latter steps and tuning the final solution need more iteration than BP. For example, in Cancer problem, the testing error of the proposed algorithm in iteration 40 was 4.02 and 10 iterations later i.e. at the iteration 50 error decreased to 2.99, but in the BP the testing error in iteration 79 is 3.21 and in the iteration 80 is 3.

TABLE 2
PERFORMANCE OF HOIPGA ON NINE BENCHMARK PROBLEMS FOR DIFFERENT PARAMETER VALUES. ALL RESULTS WERE AVERAGED OVER 50 INDEPENDENT RUNS

Name of Problem	Algorithm	Model Accuracy on Training Set Mean	Training Set SD	Validation Set Mean	Validation Set SD	Test Set Mean	Test Set SD	Number of Iteration Mean	Number of Iteration SD	Average of Training Time(second)
Cancer	HOIPGA	98.23	1.13	97.21	1.26	97.01	1.22	50.06	7.23	1.43
	BP	98.11	1.01	97.23	1.00	97.00	1.02	79.11	6.54	2.25
Card	HOIPGA	87.02	2.24	86.09	2.21	86.92	1.42	25.32	2.45	6.94
	BP	87.12	1.89	87.09	1.64	86.85	1.33	45.75	2.32	12.38
Diabetes	HOIPGA	69.19	1.23	69.11	1.14	68.87	1.13	26.59	1.45	1.68
	BP	68.99	0.98	68.87	1.09	68.54	0.95	45.11	1.40	2.81
Glass	HOIPGA	68.42	1.33	68.42	1.17	67.71	1.27	51.54	9.66	0.28
	BP	68.11	1.24	68.09	1.12	67.11	1.14	84.23	7.78	0.46
Heart	HOIPGA	79.34	1.41	79.28	1.45	77.39	1.42	45.21	5.21	2.20
	BP	78.98	1.20	78.89	1.07	77.02	1.15	59.75	3.98	2.91
Horse	HOIPGA	67.69	1.17	66.76	1.31	65.44	1.03	58.01	8.34	3.15
	BP	67.06	0.97	66.80	0.93	64.99	0.80	67.56	6.11	3.69
Iris	HOIPGA	98.02	0.18	97.01	0.32	97.02	0.39	10.18	1.02	0.12
	BP	98.12	0.02	97.21	0.19	97.01	0.18	30.51	0.87	0.34
Mushroom	HOIPGA	96.10	0.22	96.08	0.14	96.02	0.17	28.22	3.71	6.39
	BP	96.15	0.05	96.07	0.09	96.00	0.08	42.14	2.65	9.45
Thyroids	HOIPGA	93.68	0.13	93.61	0.17	92.69	0.22	17.12	1.83	48.87
	BP	93.48	0.21	93.52	0.24	92.46	0.21	25.37	1.02	69.97

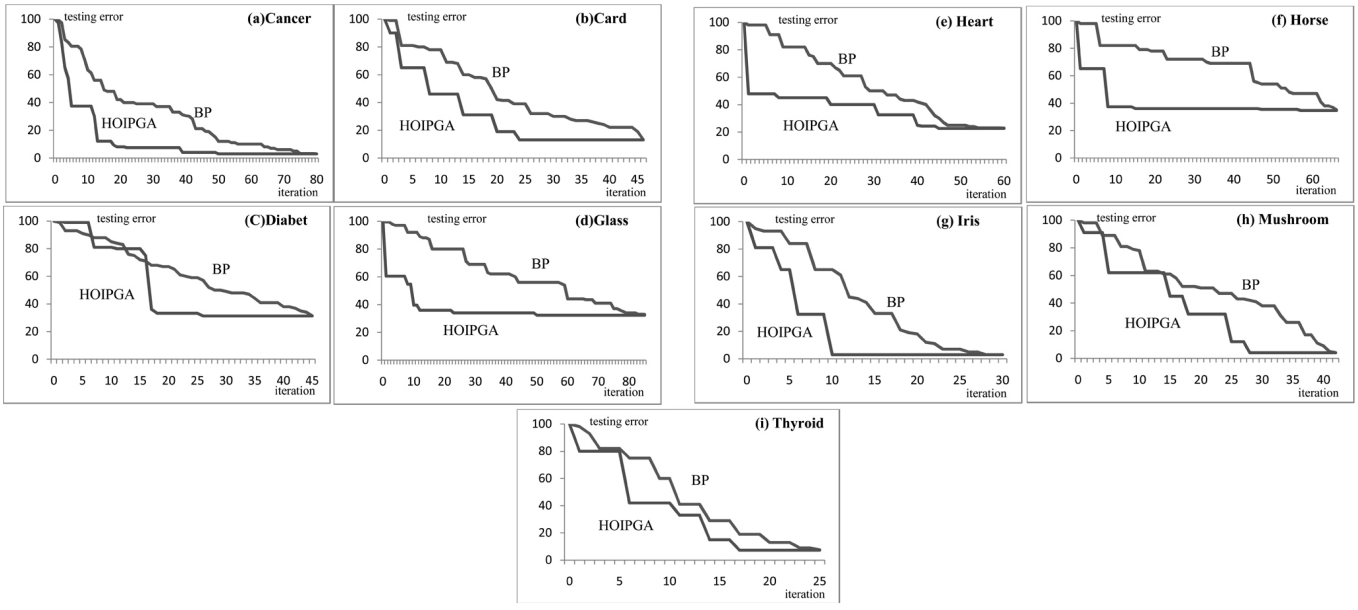


Fig. 4. The best so far network iteration- testing error graph for the Benchmarking problems

Future research will progress in two directions including improving training time and prediction accuracy of the proposed algorithm by modifying the final iteration of algorithm or the architecture of NN. The model training time or accuracy may be improved through incorporation gradient-based local search methods with the proposed algorithm especially at the final training iteration of the algorithm; also, prediction accuracy of the algorithm could be improved by using another metaheuristic to optimize NN architecture.

6 References

[1] M. Yaghini, M. M. Khoshraftar, M. Seyedabadi, "Predicting Passenger Train Delays Using Neural Network," The 4th International Seminar on Railway Operations Modelling and Analysis (*RAILROME* 2011), Feb, 2011,vol.10, no.3, Juan, 1995,16–22.

[2] S. Kiranyaz, T. Ince, A. Yildirim, M. Gabbouj, "Evolutionary artificial neural networks by multi-dimensional particle swarm optimization, "*Neural Networks*, vol. 22, no. 10, December, 2009, pp. 1448-1462.

[3] M. Castellani, H. Rowlands, "Evolutionary Artificial Neural Network Design and Training for woodveneer classification", *Engineering Applications of Artificial Intelligence*, vol. 22, 2009, pp. 732–741.

[4] E-G. Talbi, *Metaheuristic: From Design to Implementation*, University of Lille – CNRS – INRIA, a John Wiley & sons, Inc., 2009

[5] R. Battiti and G. Tecchiolli, "Training neural nets with the reactive tabu search", *IEEE Transaction on Neural Network*, vol. 6, no. 5, Sept. 1995, pp. 1185–1200.

[6] R. S. Sexton, B. Alidaee, R. E. Dorsey, and J. D. Johnson, "Global optimization for artificial neural networks: A tabu search application," *European Journal of Operational Research*, vol. 106, no. 2–3, April, 1998, pp. 570–584.

[7] N. K. Treadgold and T. D. Gedeon, "Simulated annealing and weight decay in adaptive learning: The SARPROP algorithm," *IEEE Transaction on Neural Network.*, vol. 9, no. 4, Jul. 1998, pp. 662–668.

[8] S. Chalup and F. Maire, "A study on hill climbing algorithms for neural network training," in Proc. Congress on Evolutionary Computation, vol. 3, 1999, pp. 2014–2021.

[9] X. Yao, "Evolving Artificial Neural Network", *Proceedings of the IEEE*, vol. 87, no. 9, 1999,pp. 1423–1447.

[10]V. W. Porto, D. B. Fogel, L. J. Fogel, "Alternative neural network training methods", *IEEE Expert* ,vol. 10, no. 3, Juan, 1995, pp.16–22.

[11]M. Mandischer, "A comparison of evolution strategies and backpropagation for neural network training," *Neurocomputing*, vol.42, Jan, 2002, pp. 87–117.

[12] E. Alba, J. F. Chicano, "Training Neural Networks with GA Hybrid Algorithms", K. Deb (ed.), *Proceedings of GECCO'04*, Seattle, Washington, LNCS 3102, 2004, pp. 852-863.

[13] P. Malinak, R. Jaksa, "Simultaneous Gradient and Evolutionary Neural Network Weights Adaptation Methods," *IEEE Congress on Evolutionary Computation (CEC)*, Sept, 2007, pp. 2665-2671.

[14] J. Kennedy, R. Eberhart, "Particle swarm optimization," *In Proc. IEEE international conference on neural networks*. vol. 4, Nov, 1995, pp. 1942-1948.

[15] A. P. Engelbrecht, F. V.D. Bergh, "Cooperative Learning in Neural Networks using Particle Swarm Optimizers," *South African Computer Journal*, vol. 26, 2000, pp. 84-90.

[16] Mendes R., Cortez P., Rocha M., Neves J., "Particle Swarm for Feedforward Neural Network Training," *IEEE, in Proc. International Joint Conference on Neural Networks*, 2002, pp. 1895-1899.

[17] V. G. Gudise, G. K. Venayagamoorthy, "Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithms for Neural Networks, "*IEEE Swarm Intelligence Symposium*, April, 2003, pp.110-117.

[18] F. Zaho, Z. Ren, D. Yu, Y. Yang, "Application of An Improved Particle Swarm Optimization Algorithm for Neural Network Training," *International Conference on Neural Networks and Brain (ICNN&B '05)*, Oct, 2005, pp.1693-1698.

[19] M. Carvalho, T. B. Ludermir, "Particle swarm optimization of neural network architectures and weights," *In Proc. of the 7th international conference on hybrid intelligent systems*, Sept, 2007, pp. 336-339.

[20] B. Al-Kazemi , C. K. Mohan, "Training Feedforward Neural Networks using Multi-Phase Particle Swarm Optimization", *in Proc. Ninth International Conference on Neural Information Processing*, vol. 5, 2002, pp. 2615-2619.

[21] H. R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence," *in Proc. International Conference Computational Intelligence Modeling Control and Automation,* Vienna, Austria, vol. 1, Nov, 2005, pp. 695–701.

[22] L., Han, X. He, "A novel Opposition-based Particle Swarm Optimization for Noisy Problems," *in Proc. Third International Conference on Natural Computation* (*ICNC*), *IEEE* Press, vol. 3,Aug, 2007, pp. 624 – 629.

[23] Z. Wu, Z. Ni, C. Zhang, L. Gu, "Opposition based comprehensive learning particle swarm optimization", *in Proc. 3rd International Conference on Intelligent System and Knowledge Engineering* (*ISKE*), Nov, 2008, pp. 1013-1019.

[24] M. G. H. Omran, "Using Opposition-based Learning with Particle Swarm Optimization and Barebones Differential Evolution," *Particle Swarm Optimization, InTech Education and Publishing*, 2009.

[25] C. Blum, K. Socha, "Training feed-forward neural networks with ant colony optimization: An application to pattern classification", *Fifth International Conference on Hybrid Intelligent Systems* (*HIS'*05), 2005, pp. 233-238.

[26] X. Chen, J. Wang, D. Sun, J. Liang,"A Novel Hybrid Evolutionary Algorithm Based on PSO and AFSA for Feedforward Neural Network Training", *IEEE 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08.*, Oct, 2008, pp.1-8.

[27] J. Yu, S.Wang, L. Xi, "Evolving artificial neural networks using an improved PSO and DPSO," *Neurocomputing*, vol.71, January, 2008, pp. 1054–106.

[28] A. Ratnaweera, K.Saman, H.C. Watson,"Self–organizing hierarchical particle swarm optimizer with time–varing acceleration coefficients," *IEEE Trans Evol Comput* vol.8 (3), June, 2004, pp.240–255.

[29] M. Clerc, J. Kennedy, "The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, 2002, pp. 58-73.

[30] M. Lovbjerg, T. K. Rasmussen, T. Krink, "Hybrid particle swarm optimiser with breeding and subpopulations," *In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). San Francisco*, CA, July, 2001.

[31] N. Higashi, H. Iba, "Particle swarm optimization with Gaussian mutation," *In: Proc. of the IEEE Swarm Intelligence Symp*. Indianapolis,April, 2003, pp. 72–79.

[32] L. Prechelt, "Automatic early stopping using cross validation: Quantifying the criteria," *Neural Network*, vol. 11, no. 4, Jun. 1998, pp. 761–767

Incremental Classification Based on Association Rules Algorithm (ICBA)

S. Tanarat[1], **and W. Kreesuradej**[2]
[1]Information Technology, King Mongkut's Institiute of Technology Ladkrabang, Bangkok,Thailand
[2]Information Technology, King Mongkut's Institiute of Technology Ladkrabang, Bangkok,Thailand

Abstract - *In this study, an incremental updating technique is applied to associative classification for constructing classification system when a new training dataset is appended to an old training dataset. The proposed algorithm, called Incremental Classification Based on Association Rules (ICBA). ICBA has 2 phases which are rule generator phase (ICBA-RG) and classifier building phase (ICBA-CB). In order to reduce the execution time, we applied the concept of Fast Update algorithm (FUP) algorithm to both phases of our algorithm. The experiment results show that the proposed algorithm has execution time better than CBA algorithm.*

Keywords: Incremental Associative Classification, Associative Classification, Class Association Rule.

1 Introduction

Associative Classification [3] is a framework that integrates classification and association rule mining [1,2]. The goal of associative classification is to build a model that uses association rules for classification to predict future data objects for which the class label is unknown.

Model Construction generally consists of two major phases: rule generation and classifier building. Firstly, the rule generation is discovering the set of class association rules (CARs) which satisfy the user specified constraints denoted respectively by minimum support and minimum confidence thresholds. Secondly, a classifier is built by choosing a subset of the generated class association rules (CARs). Many studies have shown that Associative Classification is often more accurate than do traditional classification techniques.

When a new training dataset is appended to an old training dataset, the classifier that uses association rules may need to be changed in order to reflect any change in the new training dataset. As a brute force technique to deal with this situation, both old training dataset and a new training dataset are merged into an updated training dataset. Then, the model construction process starts building a classifier based on the updated training dataset. This brute force technique is time consuming and inefficiency.

Therefore, a new algorithm, called Incremental Classification Based on Association Rules (ICBA) algorithm, is proposed. The objective of this algorithm is to solve these problems more efficiently. As a result, the proposed algorithm has faster execution time faster than that of the previous algorithm.

2 Related Work

2.1 Associative Classification (AC)

Associative Classification is considered as a new approach for classification. The framework of associative classification is integration of classification and association rule mining. The first associative classification algorithm is called Classification Based on Association Rules (CBA) [3].The algorithm has two major phases:

- CBA – Rule Generator (CBA-RG)
- CBA – Classifier Building (CBA-CB).

CBA-RG algorithm generates a complete set of class association rules (CARs) that satisfy the minimum support and minimum confidence thresholds. To generate the set of class association, CBA-RG algorithm finds all large ruleitem by making multiple pass over data similar to Apiori algorithm. Ruleitems are large ruleitem if their supports are greater than or equal to minimum support. For all ruleitems with the same condset, the ruleitem that have the highest confidence is chosen as possible rule. The result of this step is the set of CARs.

CBA-CB algorithm sorts the set of CARs according to the precedence relation ($\succ$). The rule ranking is defined as follows:

Given two rule r_i and r_j, $r_i \succ r_j$ (r_i has higher precedence over r_j), if one of the following holds good:

1. The confidence of r_i is greater than that of r_j
2. Their confidence are the same but support of r_i is greater than that of r_j
3. Both confidences and supports of r_i and r_j are the same, but r_i is generated before r_j

After rule ranking, each training instance is covered by a rule having the highest precedence among the rules that can

cover the case. The rule that do not cover any training instances are removed. Then, training instances that do not fall into any of the observed classed are added to a default class. Finally, rules that do not improve the accuracy of the classifier are discarded. The remaining rules and the default class of the last rule are formed as associative classifier.

2.2 An Incremental Updating Technique

When new transactions are added to the database shown in figure 1, association rules may be changed. For dynamic databases, several incremental updating techniques have been developed for mining association rule. An Incremental Updating Technique [5,6] is proposed for dynamic database which new transactions are appended.

The concept of incremental updating technique is to reuse large itemset of previous mining to obtain the update large itemset of an incremental database. Fast Update algorithm (FUP) was first introduced in [4]. The algorithm handles database with transactions insertion only. An efficient algorithm FUP is presented for computing the large itemset in the updated database. It is shown that the information from the old large itemset can be reused.

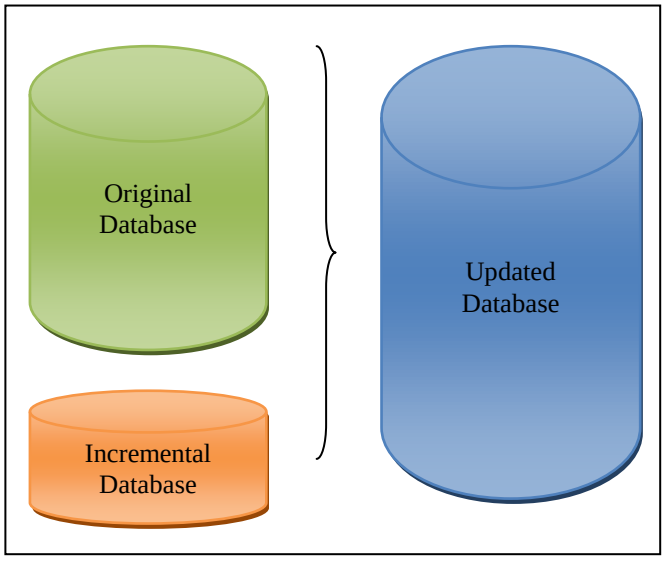


Fig. 1. Incremental Database

3 Incremental Classification Based on Association Rule Algorithm

When a new training dataset is appended to an old training dataset, an associative classifier may need to be changed in order to reflect any change in the new training dataset. However, when the training dataset changed, the existed Associative Classification algorithm always scans the changed training dataset in order to reflect the changes done. So far, there are rarely researches on the incremental learning of associative classification but there had been some studies on incremental association rule discovery algorithms that we can use their ideas for reference to solve the incremental associative classification problem. We propose an efficient incremental associative classification algorithm, a new algorithm is proposed to update an associative classifier when a new training dataset is appended to an old training dataset. The algorithm called Incremental Classification Based on Association Rules (ICBA) algorithm [8], is based on the concept of FUP algorithm to solve this problem.

The algorithm is divided into two parts. As the first part, ICBA-RG algorithm shown in figure 2 discovers the set of class association rule (CARs′) in updated training datasets. Then, the second part is building a classifier for an updated training datasets. The algorithm for the second part shown in figure 3 is called ICBA-CB algorithm.

According to Figure 2, the steps of rule generation part are outlined as follows. An incremental training dataset (t) is scanned to determine large 1-ruleitem of an updated training dataset (UT) shown at line 1-16. If a candidate 1-ruleitem is a member of previous large 1-ruleitem, its support is updated. On the other hand, if a candidate 1-ruleitem is not a member of old large 1-ruleitemset, our algorithm checks the support of the ruleitem in an incremental training datasets (t). If the support of the ruleitem in an incremental training datasets is equal or above minimum support of incremental training datasets i.e., support $\geq$ (s×d) when s is minimum support and d is size of incremental training dataset, then the algorithm scans original training datasets (T) to update support count of ruleitem ($X.support_{UT}$). In this paper, we called ruleitem which its support is greater than minimum support "winner" and ruleitem which its support is lower than minimum support "loser". As shown at line 17-33, the large k-ruleitems of updated training datasets are determined when k is greater than or equal to 2. Candidate k-ruleitems are generated by applying candidateGen function shown in line 18, this function is joining step similar to Apriori algorithm (see this function details in [2]). At k-th iteration, loser in L_k will be filtered out in a scan of t. The filtering is done by two steps. Firstly , a large k- ruleitems in L_k containing the ruleitem that cannot be the winner in the k-th iteration will be filtered out by ruleSubset function shown in line 19.

And the second , ICBA-RG filtered out loser in L_k without checking it against t. The set of losers $Y= L_k - L_{k-1}$ have been identified in line 21. Therefore, any sets of $X \in L_k$, which have subset Y such that $Y \in L_k - L_{k-1}$, cannot be large ruleitem and are filtered out from L_k. Then, if a ruleitem is a member of L_k and its support is equal or above s×(D+d) it becomes large k-itemset of update training datasets (L'_k). On the other hand, if a ruleitem is a member of C'_k and its support less than s×d, the item will be removed from candidate k-ruleitem. If the support of ruleitem is equal or above (s× (D+d), the ruleitem is inserted into L'_k which will be generated to class association rules (CARs) by genRule function[3.] at line 14 and 32. Finally, pruneRule function shown at line 15 and 33 is prune CARs by minimum confidence same as Apriori

algorithm. All rules in CARs′ which have their confidence less than minimum confidence ICBA-RG will be filter out.

```
Input : UT =  The updated training datasets
        T  =  The original training datasets with the total number of
              transactions, equal to D.
        t  =  The incremental training datasets with the total of
              number transaction equal to d.
        L′k =  The set of large-ruleitems in UT
        W  = L1 (large 1-ruleitem of T)
        s  =  minimum support

Output : CARs′ = The set of class association rules of updated
                 training datasets.

1.  for all X ∈ t do
2.     if  X ∈ W do
3.       scan t to update X.supportUT then
4.          if  X.supportUT ≥ s × (D + d)  do
5.            if  X.supportUT ≥ s × (D + d)  do
6.              insert X at the end of L′1
7.     else
8.        if  X ∉ W do
9.          for all X ∉ W do
10.           if  X.supportUT ≥ (s × d)  do
11.             scan UT to update X.supportUT then
12.               if  X.supportUT ≥ s × (D + d)  do
13.                 insert X at the end of L'1
14. CARs1 = genRules(L'1)
15. prCARs1′ = pruneRules(CARs1)
16. end
17. for (k=2; Lk-1 ≠∅; k++) do
18.    C′k = candidateGen (L′k-1) - Lk
19.    C′s  = ruleSubset (C′k)
20.         for all k-1 ruleitem in C′s  do
21.            Y = Lk-1 - L′k-1  do
22.              if  Y ⊆ X then W = W-{X}
23.    for all X ∈ C′s do
24.       if  X.supportUT ≥ s × (T + t)  do
25.          scan UT to update X.supportUT then
26.             insert X at the end of L′k
27.    for X ∈ W do
28.       scan UT to update X.supportUT then
29.          if  X.supportUD ≥ s × (T + t)  do
30.             insert X at the end of L′k
31.    end
32.  CARs′k = genRules(L′k)
33.  prCARs′k = pruneRules(CARs′k)
34. end
```

Fig. 2 ICBA-RG algorithm

For the last phase, ICBA-CB algorithm shown in figure 3 builds a classifier using CARs′. In this phase, in order to reduce the execution time we try to scan transactions of training datasets as less as we can.

ICBA-CB has three steps to build the classifier. The first step which is at line 1 is sorting the set of CARs′ according to the relation "≻".

Then, the second step at line 2-24 is selecting rules for classifier following the sorted sequence. For each rule which is member of CARs, we go through t to find those cases covered by the rule. If selected rules are not member of CARs, our algorithm goes through UT to find those covered case instead. We mark selected rules if they correctly classify a case (line 6 and line 17). If selected rules can correctly classify at least one case, they may be our potential rule in our classifier. The rules that do not cover any case are removed and the cases that do not fall into any of the observed classes are added to a default class (in case we stop selecting more rule for our classifier (C′)). The algorithm computes and records the total number of error made by classifier and default class. When there is no rule of training case left, the rule selection process is completed.

```
Input : CARs  = Set of class association rules of original training
                datasets
        R′    = Class association rule is CARs′
        UT    = Updated training datasets
        T     = Original training datasets
        t     =  Incremental training datasets
Output :  C′  = Classifier

1.  R′ = sort (R′)
2.  for each rule  r′ ∈ R′ in sequence do
3.     if  r′ ∈ CARs then
4.       for each case  X in t do
5.          if  X satisfies the condition of  r′ do
6.            store X.id in temp and mark  r′ if it
              correctly classifies t ;
7.       if  r′ is marked  then
8.       insert  r′ at the end of C′ (our classifier)
9.         delete all the ruleitem with the id in temp from db
10.        selecting a default class for the current C′
11.        compute the total number of errors of C′
12.    end
13.  else
14.    if  r′ ∉  CARs  then
15.       for each  case  X in  UT  do
16.         if  X satisfies the condition of  r′ do
17.           store X.id in temp and mark  r′ if it
              correctly classifies X;
18.      if  r′ is marked  then
19.      insert  r′ at the end of C′ (our classifier)
20.      delete all the ruleitem with the id in temp from UT
21.      selecting a default class for the current C′
22.      compute the total number of errors of C′
23.   end
24. end
25. Find  the  first  rule p′ in C′ with  the lowest  total
    number  of  errors  and drop  all the rule after p′  in C′
26. Add  the default  class  associated  with  p′ to end  of  C′
    and  return C′
```

Fig. 3 ICBA-CB algorithm

The third step at line 25-26 is discarding those rules in the classifier that do not improve the accuracy. The cutoff rule is the first rule at which there is the least number of errors recorded on UT. The remaining rules and the default class of the last rule in the classifier form our classifier.

4 Experiments

The comprehensive experiment is conducted to evaluate the efficiency of the proposed algorithm. We compare ICBA performance with CBA algorithm and we use 3 datasets (2

different minimum support thresholds and 2 different minimum confidence thresholds) from UCI Machine Learning Repository *[http://archive.ics.uci.edu/ml*]. The execution time show in figure 4, 5 and 6.

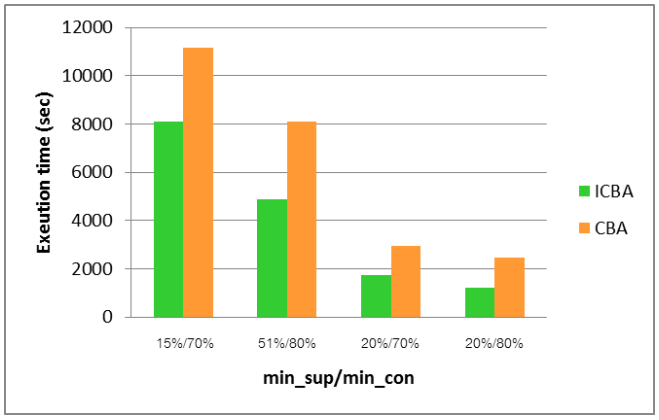


Fig. 4 Execution time of Adult dataset

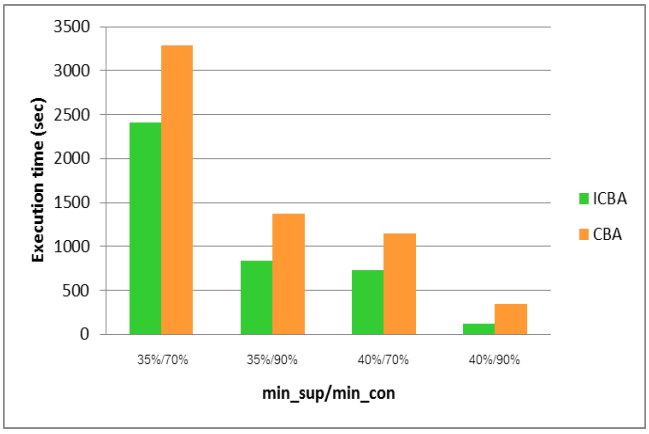


Fig. 5 Execution time of Mushroom dataset

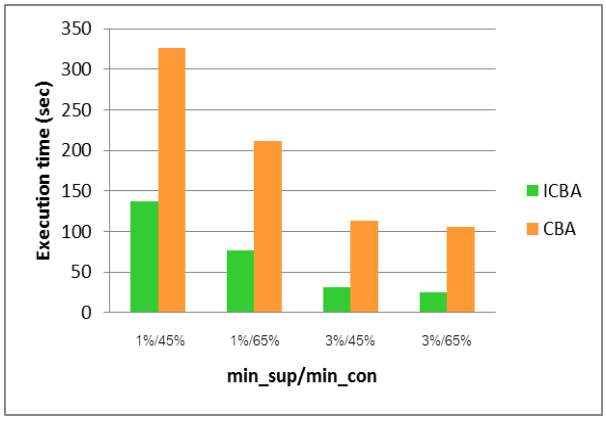


Fig. 6 Execution time of Nursery dataset

5 Conclusion

In this study, we propose an improved new classification based on association rules algorithm called Incremental classification based on association rules (ICBA). The experiment results show that our algorithm is more efficient than CBA algorithm. In the future, further researches and experiments on the proposed algorithm will be presented.

6 Reference

[1] R. Agrawal., T. lmielinski, and A. Swami, " Mining association rule between sets of items in large database", In Proceeding of the ACM SIGMOD Int'l Conf. on Management of Data, Washington, USA, May 1993, pp. 207-216.

[2] R. Agrawal, and R. Srikant, "Fast Algorithm for Mining Association Rules," Proceedings of the International Conference on Very Large Database, Santiago, Chile, 1944, pp. 487-499.

[3] Bing Liu., Wynne Hsu., Yimming Ma. "Integrating Classification and Association Rule Mining", In Proc. Of the Fourth International Conference on Knowledge Discovery and Data Mining, New York, NY, pp.80-86, 1998

[4] D. Cheung, J. Han, V. Ng, and C. Y. Wong. "Maintenance of Discovered Association Rules in Lange Database: An Incremental Updating Technique," Proceedings of the 12th IEEE International Conference on Data Engineering, 1996, pp. 106-114.

[5] D. Cheung, S.D. Lee, and B. Kao. "A General Incremental Technique for Maintaining Discovered Association Rules," Proceedings of the 5th International Conference on Database System for Advanced Applications, Melbourne, Australia, 1997, pp. 185-194.

[6] F. Thabtah. "Challenges and Interesting Research Directions in Associative Classification," Proceeding of the Sixth IEEE International Conference on Data Mining Workshops, 2006, pp.785-792.

Constrained Multi-Label Classification: A Semidefinite Programming Approach

Hui Wu*, Guangzhi Qu*, Hui Zhang†, Craig T. Hartrick‡
*Computer Science and Engineering Department, Oakland University Rochester, MI, 48309 USA
{hwu, gqu}@oakland.edu
†State Key Laboratory of Software Development Environment
School of Computer Science, Beihang University, Beijing, China 100191,
hzhang@nlsde.buaa.edu.cn
‡Anesthesiology Research, School of Medicine, Oakland University Rochester, MI, 48309 USA
chartrick@beaumont.edu

***Abstract*—Multi-label classification is more general in practice because it allows one instance to have more than one label simultaneously. In this paper, we focus on one type of multi-label classification in that there exist constraints among the labels. We formulate this kind of multi-label classification into a minimum cut problem, where all labels and their correlations are represented by a weighted graph. To attain the solutions of the minimum cut problem, we propose a semidefinite programming (SDP) approach. The experimental evaluation results show that our multi-label classification approach works much better than SVM+BR method.**

I. Introduction

Compared with single-label classification, multi-label classification is more general in practice, since it allows one instance to have more than one label simultaneously. For example, a CNN news report can be related with *politics* and *economy* at the same time. Here *politics* and *economy* are usually used as tags (labels). Because of its generality, multi-label classification has attracted much attention from researchers. Existing approaches on multi-label classification can be categorized into two main groups: program transformation method and algorithm adaptation method [26]. Program transformation method transforms a multi-label classification problem into multiple single-label classification problems. There are two typical transformation methods, *Binary Relevance* (BR) and *Label Powerset* (LP) ([25], [27]). In BR, labels are assumed to be independent ([11], [12], [16], [18], [24], [30]). Every label has its own single-label classifier. During testing, each single-label classifier determines whether its corresponding should be selected, or it gives a confidence score for further judgment. In LP, multi-labels are considered as new single labels. An apparent problem of this approach is that many generated labels are supported by very few instances. Be different from program transformation method, algorithm adaptation method extends existing single-label classification approaches to handle multi-label classification directly. Many single-label classifiers have been extended, such as Logistic regression ([5]), K-Nearest Neighbors (KNN) ([23], [35]), decision trees ([31]), and Support Vector Machine (SVM) ([8], [9], [13], [21]). Ensemble methods have also been applied for multi-label classification ([7], [19], [20], [22], [29]). Though there exist such efforts, two issues remain critical in improving multi-label classification performance.

The first issue is how to deal with the correlations among labels. A unique characteristic of multi-label classification is that there may exist correlations among labels. Some previous works have shown that these correlations have positive impact on improving the performance of multi-label classification ([5], [12]). Particularly, Cheng and Hüllermeier assume that label correlations do not change in different contexts [5]. But Ghamrawi and McCallum think that these correlations are associated with contexts [12]. In their CMLF model, a tri-tuple $< feature, label_1, label_2 >$ is defined to describe the relationship between features and label pairs. For example, given the labels of *politics*, *economy* and *people*, if '*Barack Obama*' appears in an instance, the correlation between $politics$ and $people$ will be increased. On the other side, if the feature is '*Bill Gates*', $economy$ and $people$ will have intensified correlation. In our approach, we proposed an approach to represent labels, which makes it easy to calculate label correlations while considering contexts.

Another challenge in multi-label classification is what strategy will be used to select the final labels. Filtering on single label strategy has been widely used in the literature. For example, *Label Ranking* (LR) based methods compare the confidence value of each label to the customized threshold in making the decision on whether the label will be selected into the final label set. Since the existence of the correlations among labels, when we consider the selected labels, we need a way to evaluate the merit of the label set rather than individual label.

Besides the above issues, for some applications, there are constraints among the labels in that the appearance of some labels will prohibit the others. In this work, we propose to solve multi-label classification problems with label constraints. In our approach, a weighted graph is created to represent all labels and their correlations, where vertices denote labels and edge weights reflect the label correlations. Multi-label classification is then converted into finding a partition of this weighted graph. Specifically, all labels are partitioned into two sets: one set includes the labels that are related

to the data instance, while the other set contains the labels with less relation to the data. The desired situation is that all selected labels are highly correlated with each other but have low connectivities with the un-selected labels. This idea is formulated as a minimum cut problem. To maintain the label constraints, we introduce them into the minimum cut formulation. As semidefinite programming (SDP) has been successfully used to solve cut problems [14], we also employ SDP to solve this minimum cut problem. In the experiments, we evaluate the performance of the multi-label classification, and the results show that our approach works much better than SVM+BR on all popular metrics used on measuring multi-label classification performance.

The rest of paper is organized as follows. Related work is described in Section II. Section III presents the detailed methodology, where Section III-A introduces our minimum cut system and the semidefinite programming approach, and Section III-B describes our proposed multi-label classification approach. The experimental design and results are given in Section IV, and finally Section V concludes the work.

II. Related Work

In this section, we first review different multi-label classification methods based on their implementation models: probabilistic model, logistic regression, KNN, decision tree, and Bayesian.

In probabilistic model-based multi-label classification methods, each label is generally represented by a feature distribution. But label correlations are treated differently. Some researchers assume that all single labels are independent, and multi-labels are selected by selecting labels with high individual confidence values ([34]). Other researchers utilize a mixture model to combine single label models ([12], [16], [18], [24], [30], [32]). In [16], McCallum presented a Bayesian-based model for multi-label. In the model, every document is represented by a hybrid model, which guides the creation of the class distribution and class weight distribution. In light of Bayesian rule, the classification goal is to find out the original class distribution :

$$\vec{c}^{+} \approx \arg\max_{\vec{c}} P(\vec{c}) \prod_{w \in d} \sum_{c \in C} \vec{\lambda}_c^{(\vec{c})} P(w|c) \qquad (1)$$

Where d denotes a tested document, $\vec{c}$ is a class distribution, and $\vec{\lambda}_c^{(\vec{c})}$ is the mixture weight of class c in mixture weight distribution $\vec{\lambda}^{(\vec{c})}$. They assumed that features are independent, and so are labels.

Logistic regression is another popular technique applied for multi-label classification. Cheng and Hüllermeier presented an approach to combine both instanced-based learning and logistic regression [5]. The instanced-based learning method - K-Nearest Neighbors (KNN), is used to form candidate labels, where they improved KNN by weighting votes through the similarities between an instance and its neighbors. Logistic Regression is employed to combine the correlations among labels and calculate the final occurrence probability of each label. In another work [11], Fujino and Isozaki built the binary classifier of each label with Logistic Regression.

KNN has been extended for handling multi-label classification in many ways. Wpyromitros $et\ al.$ computed each label's confidence through KNN, and the multi-label is formed according to these confidence values [23]. Brinker and Hüllermeier used KNN-based binary relevance method for multi-label ranking [4]. Zhang and Zhou proposed ML-KNN to combine both KNN and Bayesian rule [35]. ML-KNN's objective function is :

$$\vec{y_t}(l) = \arg\max_{b \in \{0,1\}} P(H_b^l | E^l_{\vec{C}_t(l)}), l \in Y \qquad (2)$$

Where $\vec{C}_t(l)$ denotes the number of t's neighbors having label l, and Y is the entire label set. H_b^l is the event that whether an instance t is labeled l. If b is equal to 1, it means that t has label l; otherwise, l is not assigned to t. Since b has only two options, 0 and 1, then if $P(H_1^l|E^l_{\vec{C}_t(l)}) \geq P(H_0^l|E^l_{\vec{C}_t(l)})$, l is assigned to t; otherwise, t is equal to 0. According to their experiments, ML-KNN has better performance than Boostexter [22], multi-label decision tree ADTboost.MH [7] and the multi-label kernel method Rank-SVM [8].

Celine Vens $et\ al.$ analyzed the techniques of decision trees for hierarchical multi-label classification [31]. They classified these techniques into three sorts. One is Single-label Classification (SC) approach, where each class has one binary classifier. The second is Hierarchical Single-label Classifier(HSC). The third approach offers the labels of one example at once and then uses the hierarchical category to give final multiple labels, named as HMC, which works best according to their experiments. Amanda Clare and Ross D.King improved C4.5 by modifying the formula of *entropy* to deal with multi-label classification [6].

In [34], Zhang $et\ al.$ proposed a Naive Bayes based approach for multi-label classification. In their method, features all follow Gaussian Distribution. To meet the assumption of Naive Bayes features independence, a feature selection procedure was conducted.

In this paper, we commence by using a weighted graph to represent labels and their relations. Then we use semidefinite programming to develop an effective algorithm to solve the minimum cut problem for furthering the multi-label classification.

III. Methodology

A. Minimum Cut with Semi-definition Programming

Let $G = (V, E, W)$ be an undirected weighted graph, where V is a vertex set, E is an edge set, and W is the edge weight set of G. w_{ij} indicates the weight of edge e_{ij}, and $w_{ii} = 0$. In graph theory, a cut is a partition of the vertices of a graph into two disjoint subsets. The cut-set of the cut is the set of edges whose end points are in different subsets of the partition. Edges are said to be crossing the cut if they are in the cut-set. In a weighted graph, the weight of a cut is defined by the sum of the weights of the edges crossing the cut [33]. The

minimum cut aims to minimize the weight of a cut, whose objective function is:

$$\arg\min(\sum_{v_i\in A, v_j\in B} w_{ij}). \quad (3)$$

In Equation 3, A, B are two disjoint sets of a partition, which means $A \cup B = V$ and $A \cap B = \emptyset$ [15]. According to the end points of an edge, we classify all edges into three groups: $A-A$, $B-B$ and $A-B$:

$$\begin{aligned} S &= \sum_{v_i,v_j\in V, v_i\neq v_j} w_{ij} \\ &= \sum_{v_i\in A, v_j\in B} w_{ij} + \sum_{v_i\in A, v_j\in A} w_{ij} + \sum_{v_i\in B, v_j\in B} w_{ij} \quad (4) \end{aligned}$$

In a given graph G, S is a constant. Therefore, we can reform Equation 3 as:

$$\begin{aligned} & \arg\min(\sum_{v_i\in A, v_j\in B} w_{ij}) \\ = \ & \arg\min(S - (\sum_{v_i\in A, v_j\in A} w_{ij} + \sum_{v_i\in B, v_j\in B} w_{ij})) \\ = \ & S + \arg\min(-1 * (\sum_{v_i\in A, v_j\in A} w_{ij} + \sum_{v_i\in B, v_j\in B} w_{ij})) \\ = \ & S - \arg\max(\sum_{v_i\in A, v_j\in A} w_{ij} + \sum_{v_i\in B, v_j\in B} w_{ij}) \quad (5) \end{aligned}$$

According to Equation 5, the optimal solution of Equation 3 is also the optimal solution of:

$$\arg\max(\sum_{v_i\in A, v_j\in A} w_{ij} + \sum_{v_i\in B, v_j\in B} w_{ij}), \quad (6)$$

which is to say :

$$\begin{aligned} & \arg\min(\sum_{v_i\in A, v_j\in B} w_{ij}) \\ \Leftrightarrow \ & \arg\max(\sum_{v_i\in A, v_j\in A} w_{ij} + \sum_{v_i\in B, v_j\in B} w_{ij}). \quad (7) \end{aligned}$$

Combine Equation 5 and 7, we get:

$$\begin{aligned} & \arg\min(\sum_{v_i\in A, v_j\in B} w_{ij}) \quad (8) \\ \Leftrightarrow \ & \arg\max(\sum_{v_i, v_j\in A} w_{ij} + \sum_{v_i, v_j\in B} w_{ij} - \sum_{v_i\in A, v_j\in B} w_{ij}). \end{aligned}$$

Let $\mathbf{x}$ be the indicator vector of the partition of A, B. One item in $\mathbf{x}$ corresponds to a vertex, and this item's value indicates the partition assignment of this vertex. If the value is equal to $+1$, it means the corresponding point is assigned to set A, and if the value is -1, then this corresponding vertex is assigned to set B. With this representation, the right side of Equation 8 can be calculated through :

$$\begin{aligned} \arg \quad & \max \mathbf{x}^T W \mathbf{x} \\ s.t. \quad & \mathbf{x} \in \{-1, 1\}^{|V|}. \quad (9) \end{aligned}$$

In Equation 9, $\mathbf{x}^T W \mathbf{x}$ is equal to $\sum_{i,j} w_{ij} x_i x_j$. If v_i and v_j are in the same set, $x_i x_j$ is equal to $+1$; otherwise, its value is -1.

Constraint is defined as a partition of $S \subseteq V$, denoted as $\alpha(S)$ and $\beta(S)$, which satisfy $|\alpha(S)| = t$ ($t \in \mathbb{R}^+$). Since $\alpha(S)$ and $\beta(S)$ are a partition of S, then $\alpha(S) \cup \beta(S) = S$ and $\alpha(S) \cap \beta(S) = \emptyset$. Combine all this kind of constraints together, we form a *constraint matrix*, denoted as M, where each row indicates a constraint and each column corresponds one vertex. Therefore, $M(i, j)$ denotes the value of vertex v_j in i-th constraint. Let S_i be the vertex set involved in i-th constraint, then :

$$M(i,j) = \begin{cases} 1 & \text{if } v_j \in S_i; \\ 0 & \text{Others.} \end{cases} \quad (10)$$

Add the constraint matrix M to Equation 9, our new system is defined as:

$$\begin{aligned} \arg \quad & \max \mathbf{x}^T W \mathbf{x} \\ s.t. \quad & M\mathbf{x} = \mathbf{k} \\ & \mathbf{x} \in \{-1, 1\}^{|V|}, \quad (11) \end{aligned}$$

where $\mathbf{k}$ is a vector. Let $|\alpha(S_i)| = 1$ and the vertices of $\alpha(S_i)$ have the value of -1 in $\mathbf{x}$, then $|\beta(S_i)| = |S_i| - 1$ and the vertices of $\beta(S_i)$ have the value of -1 in $\mathbf{x}$. Under this context, $\mathbf{k}_i = (-1) + (|S_i| - 1) = |S_i| - 2$. The system in Equation 11 is not linear. Since semidefinite programming has been successfully employed to solve a maximum-cut problem [14], and this minimum cut has the similar formulation with maximum-cut problem, so we will employ semidefinite programming to solve the problem.

We firstly transfer the first constraint equation as the following :

$$\begin{aligned} (M\mathbf{x})^T M\mathbf{x} &= \mathbf{x}^T M^T M \mathbf{x} \\ &= \mathbf{k}^T \mathbf{k}. \quad (12) \end{aligned}$$

Let $X = \mathbf{x}\mathbf{x}^T$, then the system is reformed as

$$\begin{aligned} \arg \quad & \max W \bullet X \\ s.t. \quad & (M^T M) \bullet X = \mathbf{k}^T \mathbf{k} \\ & X = \mathbf{x}^T \mathbf{x} \\ & x_i \in \{-1, 1\}, i = 1, \ldots, n. \quad (13) \end{aligned}$$

where $\bullet$ is the notation of computing dot product. The last constraint set in Equation 13 are equivalent to $X_{ii} = 1$ ($i = 1, \ldots, n$) [10]. X=$\mathbf{x}^T\mathbf{x}$, which is a symmetric rank-1 positive semi-definite matrix, is relaxed by removing the rank-1 restriction. Consequently, we get the following system

$$\begin{aligned} \arg \quad & \max W \bullet X \\ s.t. \quad & (M^T M) \bullet X = \mathbf{k}^T \mathbf{k} \\ & \text{diag}(X) = \mathbf{e} \\ & X \succeq 0, \quad (14) \end{aligned}$$

where $\mathbf{e}$ is an all-one vector, and $\text{diag}(X)$ indicates the items on X's diagonal. Equation 14 has the standard format of positive semidefinite programming. Consequently, we can easily

get the value of X by a positive semidefinite programming solver, such as CSDP [1] and SeDuMi [2]. As X is a real symmetric square matrix, it has particular properties, which are given in Property 1.

Property 1: [17] Let X be a symmetric n-by-n matrix. The following are equivalent:

1) There exists an n-by-n matrix Y such that $YY^T = X$.
2) For all $y \in \mathbb{R}^n$, $y^T X y \succeq 0$.
3) All eigenvalues of X are non-negative.

Since there is no direct way to get x from X, which satisfies $\mathbf{x}^T\mathbf{x} = X$, we utilize the first property in Property 1 to attain a n-by-n matrix Y. Y is computed by Cholesky decomposition, which is described in Theorem 1.

Theorem 1: (**Cholesky decomposition**) [3]. Let X be a real symmetric positive definite matrix. There exists a unique real lower triangular matrix Y, having positive diagonal entries, such that

$$X = YY^T. \tag{15}$$

In conclusion, the proposed approach of minimum cut with semidefinite programming is described in Algorithm 1.

Algorithm 1 Minimum Cut with Semidefinition Programming

Input:
W - correlation matrix among vertices;
M - constraint matrix;
k - constraint vector.
Output:
x - vertex assignment vector.
Process:
Solve the system in Equation 14 by CSDP, and get X;
Get Y by running Chlolesky decomposition on X (Assume Y is an upper triangle matrix);
Attain **y** by selecting the first row in Y;
Set $\mathbf{x} = \mathbf{e}$
for i-th constraint **do**
  Get $sa = \{v_j | v_j \in S_i, \mathbf{x}_j = -1\}$;
  if $|sa| < 1$ **then**
    $v_j = \arg\min_{v_t \in S_i} \mathbf{y}_t$
    Set $\mathbf{x}_j = -1$
  end if
end for

B. Application on Multi-Label Classification

Let $L = \{l_1, l_2, \ldots, l_n\}$ be a label set, and d be one test instance. The goal of multi-label classification is to find a label set $s \subseteq L$, which is able to best describe the label information of d. In this paper, we further present the concept of label constraint for multi-label classification. Let $s \subset L$ be a label subset, and ml be a multi-label. The *label constraint* is stated as that there is one and only one label l satisfying $l \in s$ and $l \in ml$. Figure 1 gives one example of label constraints in the multi-label classification.

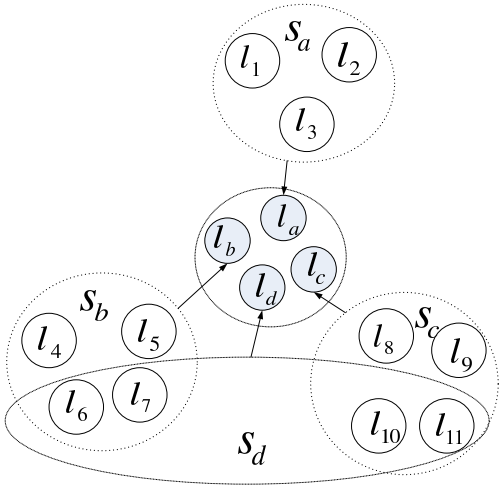


Fig. 1. One Example of Multi-Label Classification with Label Constraints

In Figure 1, there are four label constraints. Each of S_a, S_b, S_c and S_d corresponds to the label set of one constraint. For a test instance d, the constrained multi-label classification will find a label set $\{l_a, l_b, l_c, l_d\}$, which can best describe the label information of d and satisfy all the constraints. Here, l_p ($p \in \{a, b, c, d\}$) comes from S_p, i.e. $l_p \in S_p$. The constrain on S_p means that there is a partition of $\alpha(S_p)$ and $\beta(S_p)$, where $\alpha(S_p) \cup \beta(S_p) = S_p$, $\alpha(S_p) \cap \beta(S_p) = \emptyset$ and $\alpha(S_p) = 1$. $\alpha(S_p)$ includes the selected label. For example, $\alpha(S_1) = \{l_1\}$ and $\beta(S_1) = \{l_2, l_3\}$ are one eligible partition of S_1, but $\alpha(S_1) = \{l_1, l_2\}$ and $\beta(S_1) = \{l_3\}$ are not, since $|\alpha(S_1)| = |\{l_1, l_2\}| \neq 1$. Furthermore, S_b and S_d have common labels, so do S_c and S_d. Then, if l_6 is selected, any label in $S_b \cup S_d \setminus \{l_6\}$ will not be selected. For instance, if l_4 is also be selected, the constraint on S_6 will not be satisfied, as there are two labels $\{l_4, l_6\}$ are selected. In this case, l_d is equal to either l_b or l_c. $\{l_1, l_7, l_8\}$ is one eligible multi-label, where $l_a = l_1$, $l_b = l_7$, $l_c = l_8$ and $l_d = l_7$.

Our goal of multi-label classification is to find such a label set that all labels in this label set are highly correlated with each other, and they have low correlations with the other labels. Let all labels and their correlations be represented by a weighted graph, where vertices denote labels and edge weights reflect correlations. Therefore, the constrained multi-label classification problem is transfered into a minimum cut problem, where all the selected labels are assigned to set A. There are three steps in building the weighted graph: weighting single labels, computing label correlations and normalization.

The label importance is a key factor to judge whether one label should be selected or not. We do not limit specific computation method on calculating the label importance. All kinds of single-label classification methods are able to offer label importances, such as Support Vector Machine (SVM), Naive Bayes and Logistic Regression. A high label importance value means this label will probably be selected. Let h_{l_i} denote the importance of $l_i \in L$.

While computing these correlations, we make that assump-

tion that all labels are dependent, and their correlations are closely related with contexts. Let $F = \{f_1, f_2, \ldots, f_t\}(t \in \mathbb{R})$ be a feature set. We represent each label with a feature vector, say l_i's feature vector is $\mathbf{v}^i$. $\mathbf{v}^i_j$ indicates the relevance between feature $f_j \in F$ and l_i, and its value is calculated by :

$$\mathbf{v}^i_j = \frac{I(l_i, f_j)}{H(l_i)}, \tag{16}$$

where $I(l_i, f_j)$ is the mutual information of l_i and f_j, and $H(l_i)$ is l_i's entropy. Let $\mathbf{d}$ be the feature vector of d, where $\mathbf{d}_i$ is the term frequency of f_i. Given $\mathbf{d}$, l_i's specified feature vector is $\mathbf{r}^i$, which is defined as:

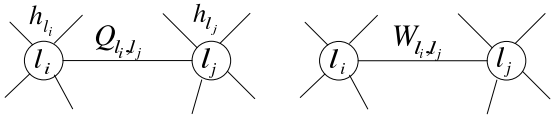
$$\mathbf{r}^i_j = \mathbf{v}^i_j * \mathbf{d}_j. \tag{17}$$

Let Q be the correlation matrix, which is a m-by-m matrix, and Q_{l_i,l_j} indicates the correlation between l_i and l_j. In the context of $\mathbf{d}$, Q_{l_i,l_j} is calculated by:

$$Q_{l_i,l_j} = \exp(\frac{-1}{2\sigma^2}\mathrm{Ed}(\mathbf{r}^i, \mathbf{r}^j)), \tag{18}$$

where σ is a parameter, whose value is equal to 0.5 in this paper, and $\mathrm{Ed}(\mathbf{r}^i, \mathbf{r}^j)$ is the Euclidean distance between $\mathbf{r}^i$ and $\mathbf{r}^j$.

After the first two steps, all vertices and edges are weighted.



(a) Unnormalized Weighted Graph (b) Weighted Label Graph

Fig. 2. Weighted Graph Normalization

In Figure 2(a), l_i and l_j both have their own importances, h_{l_i} and h_{l_j}, and they also have their correlation value Q_{l_i,l_j}. Finally, we normalize this weighted graph to let only edges be weighted, and the normalization formula is given by:

$$W_{l_i,l_j} \quad = h_{l_i} \times Q^*_{l_i,l_j} + h_{l_j} \times Q^*_{l_j,l_i}, \tag{19}$$

where W_{l_i,l_j} is the normalized correlation between l_i and l_j, and

$$Q^*_{l_i,l_j} = \frac{Q_{l_i,l_j}}{\sum_{l \in \{l_1,\overline{l_1},\ldots,l_m,\overline{l_m}\}} Q_{l_i,l}}. \tag{20}$$

Figure 2(b) gives the normalized result of Figure 2(a).

After getting M, W and $\mathbf{k}$, $\mathbf{x}$ will be outputted by Algorithm 1. The labels whose values are -1 in $\mathbf{x}$ will be selected to form a multi-label.

C. Analysis

Though our approach derives from handling the multi-label classification with label constraints, it can also deal with multi-label classification without label constraints. Let l_1, l_2 and l_3 be three labels. We expand l_i to two labels l^+_i and l^-_i, where l^+_i means l_i is selected, and l^-_i means l_i is not selected. Therefore, l_i forms a label constraint. Figure 3 describes this design.

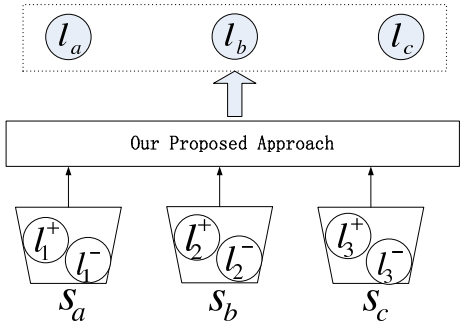


Fig. 3. One Example of Our Proposed Approach on Handling Multi-Label Classification Without Label Constraints

As shown in Figure 3, l_i ($1 \leq i \leq 3$) is replaced by a label set including l^+_i and l^-_i. In each label set, one and only one label will be selected, since l_i is either selected or not selected. If $l_a = l^+_1$, $l_b = l^-_2$ and $l_c = l^+_3$, then the multi-label is $\{l_1, l_3\}$.

Furthermore, our constrained multi-label classification model is also able to represent the multiclass-multilabel classification, where each label has different classes and the classes of one label are exclusive. If we make this mapping that the multiclass-multilabel classification's label corresponds to our constrained multi-label classification's label set, and its class corresponds to our label, then the multiclass-multilabel classification is a constrained multi-label classification. Basically, multiclass-multilabel classification is one specific case of the constrained multi-label classification, where the label sets of any two label constraints have no mutual components.

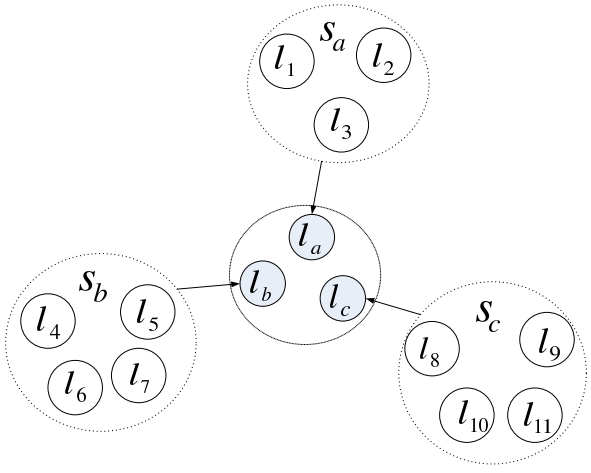


Fig. 4. One Example of Our Proposed Approach on Handling Multiclass-Multilabel Classification

Figure 4 describes one example of our proposed approach on handling multiclass-multilabel classification, where S_a, S_b and S_c are three different labels, S_a has three classes $\{l_1, l_2, l_3\}$, S_b has four classes $\{l_4, l_5, l_6, l_7\}$ and S_c has four classes $\{l_8, l_9, l_{10}, l_{11}\}$. Any two labels have no mutual classes.

IV. EXPERIMENTS

A. Evaluation Metrics

On evaluating performance of multi-label classifiers, lots of criteria have been created ([13], [22]). As there is no ranking score of each label in multi-labels predicted by our approach, then we focus on example-based metrics [28], including $Hamming\ loss$, $Precision$, $Recall$, $F1$ and $Accuracy$. Let $T = \{(x_1, Y_1), (x_2, Y_2), \ldots, (x_q, Y_q)\}$ ($q \in \mathbb{R}$) be a test data set, where x_i ($1 \leq i \leq q$) is a test instance, and $Y_i \in C^*$ is x_i's multi-label. In addition, let $h(x_i)$ be the predicted multi-label for x_i by one multi-label classifier.

- *Hamming loss* calculates the percentage of mis-predicted labels:

$$HL(T) = \frac{1}{q}\sum_{i=1}^{q}\frac{1}{m}|h(x_i)\Delta Y_i| \tag{21}$$

  Where Δ is the notation for differentiating two sets.
- *Precision* comes from the metrics for single-label classifiers in Information Retrieval (IR):

$$P(T) = \frac{1}{q}\sum_{i=1}^{q}\frac{|Y_i \cap h(x_i)|}{|h(x_i)|} \tag{22}$$

- *Recall* corresponds to *recall* in single-label metrics:

$$R(T) = \frac{1}{q}\sum_{i=1}^{q}\frac{|Y_i \cap h(x_i)|}{|Y_i|} \tag{23}$$

- *F1* combines *precision* and *recall*:

$$F1(T) = \frac{1}{q}\sum_{i=1}^{q}\frac{2 * P(x_i) * R(x_i)}{P(x_i) + R(x_i)} \tag{24}$$

- *Accuracy* is similar to *accuracy* in single-label metrics:

$$A(T) = \frac{1}{q}\sum_{i=1}^{q}\frac{|Y_i \cap h(x_i)|}{|Y_i \cup h(x_i)|} \tag{25}$$

B. Experimental Results and Discussions

We run the experiments on a real medical data set. In the field of pain medicine, a demanding problem is how to predict the potential effects of a specific treatment plan. These effects are usually closely related with each other, and each effect has different levels. A general case is that every effect has 3 different levels, which range from 1 to 3. Level 1 means the pain is light, and level 3 means the pain is heavy. If each pain level is regarded as a label, these three labels should be in a label set, and only one of them will be selected by the multi-label.

Our real data set includes 128 patient records. Each record has five different aspects of pain, including pain intensity, sharp, hot, dull and sensitive. Besides these pains, one record also has symptoms, such as genes, anxiety levels and depression levels. All these symptoms are regarded as features in our experiments. We compare our approach with SVM+BR. In SVM+BR, each label set has its own SVM single-label classifier. The experiments use 8-folder crossing validation, where each folder has 21 records. The performance comparisons of all folders are given in Figure 5. As shown in Figure 5, our proposed approach works better than SVM+BR among 75% data sets.

It should be noticed that $h(x_i)$ and Y_i(refer to Equations 22 to 24), have the same values in our experiments. They are both equal to the number of label sets. Therefore, their experimental results are identical, which is demonstrated by Figures 5(b) to 5(d). The summarized experimental results are given in Table I.Table I shows that our proposed approach's performances are higher than SVM+BR's with a big gap in all metrics. These experiments demonstrate the effectiveness of our approach.

	SVM+BR	Our Proposed Approach
Hamming loss	0.1432±1.2875e-004	**0.1292±2.1373e-004**
Precision	0.2838±0.0032	**0.3542±0.0053**
Recall	0.2838±0.0032	**0.3542±0.0053**
F1	0.2838±0.0032	**0.3542±0.0053**
Accuracy	0.1922±0.0018	**0.2492±0.0040**

TABLE I
COMPARISONS OF PERFORMANCES, AND THE FORMAT IS *'mean±variance'*.

V. CONCLUSION

In this paper, we focus on the multi-label classification with label constraints. This kind of multi-label classification is a challenge to most current approaches. To solve this problem, we transfer the multi-label classification problem into a minimum cut problem, where labels and their correlations are represented by a weighted graph. We further propose a semidefinite programming approach to solve this minimum cut problem. According to the experiments, our approach works much better than SVM+BR method in all metrics. In the future, we will apply our approach into more applications.

REFERENCES

[1] Csdp, a c library for semidefinite programming. Jan. 26, 2011. https://projects.coin-or.org/Csdp/.

[2] Sedumi. Jan. 26, 2011. http://sedumi.ie.lehigh.edu/.

[3] Gregoire Allaire and Sidi Mahmoud Kaber. In *Numerical Linear Algebra*. Springer, 2008.

[4] Klaus Brinker and Eyke Hüllermeier. Case-based multilabel ranking. In Manuela M. Veloso and Manuela M. Veloso, editors, *IJCAI*, pages 702–707, 2007.

[5] Weiwei Cheng and Eyke Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2-3):211–225, 2009.

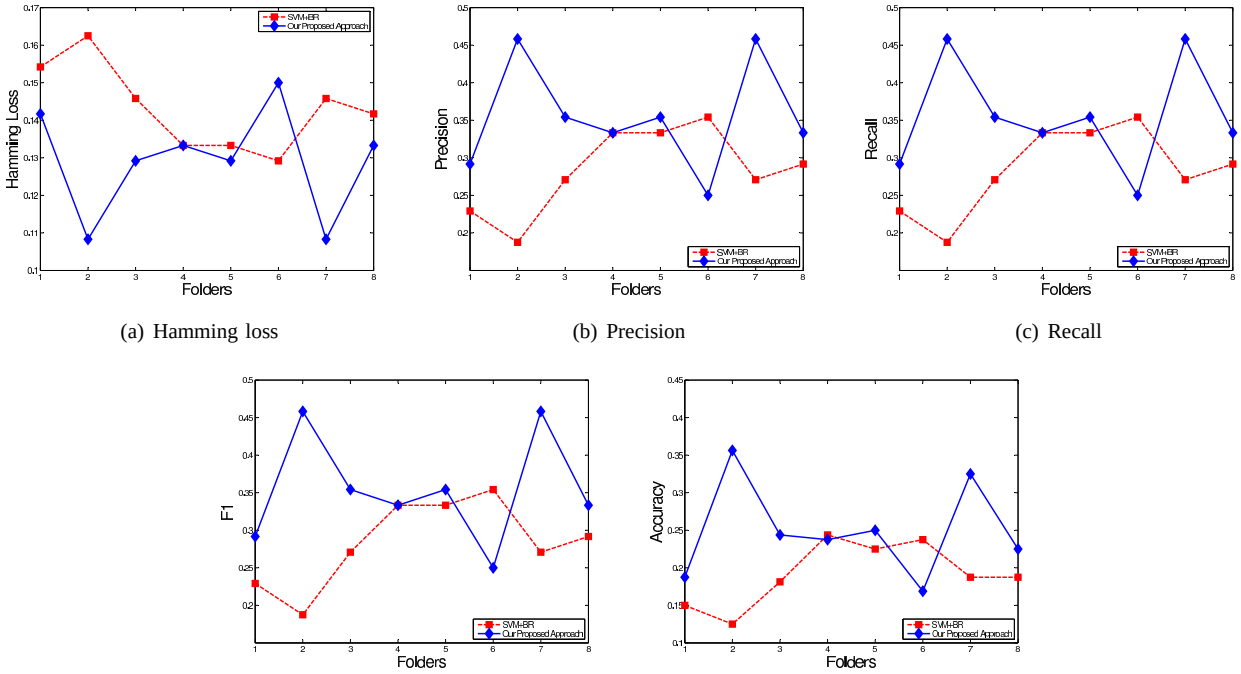
[6] Amanda Clare and Ross D. King. Knowledge discovery in multi-label phenotype data. In *PKDD '01: Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 42–53, London, UK, 2001. Springer-Verlag.

[7] Francesco De Comité, Rémi Gilleron, and Marc Tommasi. Learning multi-label alternating decision trees from texts and data. pages 251–274. 2003.

[8] Andre Elisseeff and Jason Weston. Kernel methods for multi-labelled classification and categorical regression problems. In *In Advances in Neural Information Processing Systems 14*, pages 681–687. MIT Press, 2001.

[9] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *Annual ACM Conference on Research and Development in Information Retrieval*, pages 274–281, 2005.

[10] Robert M. Freund and Brian Anthony. Introduction to semidefinite programming (sdp). 2004.



(a) Hamming loss (b) Precision (c) Recall

(d) F1 (e) Accuracy

Fig. 5. Experimental Results .

[11] Akinori Fujino and Hideki Isozaki. Multi-label classification using logistic regression models for ntcir-7 patent mining task. In *Proceedings of NTCIR-7 Workshop Meeting*, 2008.

[12] Nadia Ghamrawi and Andrew McCallum. Collective multi-label classification. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 195–200, New York, NY, USA, 2005. ACM.

[13] Shantanu Godbole and Sunita Sarawagi. Discriminative methods for multi-labeled classification. In *In Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 22–30. Springer, 2004.

[14] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42:1115–1145, November 1995.

[15] Jonathan Gross and Jay Yellen. Graph theory and its applications. CRC Press, 1998.

[16] Andrew Kachites McCallum. Multi-label text classification with a mixture model trained by em, 1999.

[17] Ryan O'Donnell. Semidefinite programming and max-cut. 2008.

[18] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 248–256, Singapore, August 2009. Association for Computational Linguistics.

[19] Jesse Read, Bernhard Pfahringer, and Geoff Holmes. Multi-label classification using ensembles of pruned sets. In *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, volume 0, pages 995–1000, Washington, DC, USA, 2008. IEEE Computer Society.

[20] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. In *ECML PKDD '09: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 254–269, Berlin, Heidelberg, 2009. Springer-Verlag.

[21] Juho Rousu, Craig Saunders, Sandor Szedmak, and John Shawe-Taylor. Kernel-based learning of hierarchical multilabel classification models. *JOURNAL OF MACHINE LEARNING RESEARCH*, 7:1601–1626, 2006.

[22] Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[23] E. Spyromitros, G. Tsoumakas, and Ioannis Vlahavas. An empirical study of lazy multilabel classification algorithms. In *SETN '08: Proceedings of the 5th Hellenic conference on Artificial Intelligence*, pages 401–406, Berlin, Heidelberg, 2008. Springer-Verlag.

[24] Andreas Streich and Joachim Buhmann. Classification of multi-labeled data: A generative approach. pages 390–405. 2008.

[25] Lena Tenenboim, Lior Rokach, and Bracha Shapira. Identification of label dependencies for multi-label classification. In *MLD 2010 : Second International Workshop on learning from Multi-Label Data*, 2010.

[26] G. Tsoumakas and I. Katakis. Multi label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

[27] G. Tsoumakas, I. Katakis, and I. Vlahava. Mining multi-label data. *Data Mining and Knowledge Discovery Handbook, O. Maimon, L. Rokach (Ed.), Springer, 2nd edition*, 2009.

[28] G. Tsoumakas, I. Katakis, and I. Vlahavas. A review of multi-label classification methods. *Proceedings of the 2nd ADBIS Workshop on Data Mining and Knowledge Discovery (ADMKD 2006)*, 2006.

[29] Grigorios Tsoumakas and Ioannis Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *ECML '07: Proceedings of the 18th European conference on Machine Learning*, pages 406–417, Berlin, Heidelberg, 2007. Springer-Verlag.

[30] N. Ueda and K. Saito. Parametric mixture models for multi-labeled text, 2002.

[31] Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, and Hendrik Blockeel. Decision trees for hierarchical multi-label classification. *Machine Learning*, 2(73):185–214, August 2008.

[32] Hongning Wang, Minlie Huang, and Xiaoyan Zhu. A generative probabilistic model for multi-label classification. In *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 628–637, Washington, DC, USA, 2008. IEEE Computer Society.

[33] Wikipedia. Cut (graph theory). Jan. 9, 2011. http://en.wikipedia.org/wiki/Cut_(graph_theory).

[34] Min-Ling Zhang, José M. Pe na, and Victor Robles. Feature selection for multi-label naive bayes classification. *Inf. Sci.*, 179(19):3218–3229, 2009.

[35] Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, July 2007.

An Empirical Study of Class Noise Impacts on Supervised Learning Algorithms and Measures

Victor S. Sheng, Rahul Tada, and Abhinav Atla
Department of Computer Science, University of Central Arkansas, Conway, AR, USA

Abstract - *Noise in data is an effective cause of concern for many machine learning techniques. Researchers have studied the noise impacts only on some particular learning algorithm. We empirically study the noise impacts on four different representative learning algorithms and the two popular measures (accuracy and AUC) under different intensities of noise, particularly decision tree, naïve bayes, support vector machine, and logistic regression. Our empirical results show that AUC is more tolerant to noise. Among the four algorithms, naïve bayes is the most resistant to noise, but it performs the worst in accuracy. The other algorithms perform much better than naïve bayes especially after the noisy level is lower than 40%. When we develop approaches to improve the data quality (reduce the noise level) and build model with higher accuracy, decision tree is the most preferred one, followed by logistic regression and support vector machine. However, logistic regression performs the best in AUC.*

Keywords: Noisy learning, noise impact, supervised learning, machine learning, data mining

1 Introduction

Data extracted from real-world problems using supervised learning techniques frequently contains noise. Noise decreases the quality of the data and might affect the learning process leading to inaccurate models. This research is to discover the performance of learning algorithms with noisy training data, and the noise sensitivity of learning algorithms with various percentages of noise. The results of this research can assist choosing proper learning algorithms. It can recommend which learning algorithm is preferred under what level of noise. Furthermore, if we can reduce the noise level, which algorithm could bring higher benefit?

Cleaning data and improving the data quality is one of the preprocessing of KDD process [1]. Cleaning the mislabeled instances before training a learning model is very common. Mislabeled instances hurt the performance of supervised learning algorithms. Great efforts contribute to identify the potential mislabeled data and further to handle these mislabeled data properly [2] [3]. There are several approaches addressed to improve the label quality of training instances. The knowledge of the impacts of noise on different supervised learning algorithms can help us choose the learning algorithms, which are sensitive to noise reduction and perform better with the data quality improvement. On the other hand, try not to choose the noise insensitive algorithms.

Real-world data contains attribute noise and/or class noise. We focus on the class noise. Specifically, we are concerned with inaccuracies that migrate from data contaminated with class noise. We study the performance of four representative algorithms: decision tree, naïve bayes, support vector machine, and logistic regression, under different noise levels from 0% (no noise) to 50% (complete noise for binary classification). This empirical study guides the algorithm choice under different noise levels, particularly the two extreme cases: very noise or no noise at all. Our empirical results show that naïve bayes is the most resistant to noise, but it performs the worst in accuracy. The other algorithms perform much better than naïve bayes especially after the noisy level is lower than 40%. Decision tree is the preferable one. However, in AUC (area under the ROC curve) [4] [5] [6], naïve bayes performs not bad, although it follows logistic regression. Logistic regression is the best.

The investigation of the noise sensitivity of the basic algorithms under different noise levels is the preliminary study, which helps the study of improving data quality (reduce the noise level via removing/correcting identified noise). Studying data quality improvement needs to choose noise reduction sensitive algorithms. There exist numerous articles about handling noisy data, including removing the noise instances directly and correcting the noise instances. But, according to our knowledge, none of them emphasizes the difference of noise impacts on different learning algorithms, and none of them explicitly explained why the certain algorithms were chosen. Either for removing or for correcting, it can be expected that the noise reduction sensitive learning algorithms would be proper ones to be chosen, not the noise reduction insensitive learning algorithms. Our experimental results show that decision tree is the most preferred one, followed by logistic regression and support vector machine, in accuracy.

This paper also investigates the noise sensitivity of the measures under different noise levels. Accuracy and AUC are two common measures for supervised learning algorithms. In this paper, we investigate their noise sensitivity via comparing the performance of the supervised learning algorithms based on these two measures. It has been shown that the two measures are not completely identical [7]. Our experimental results show that accuracy is more sensitive to noise, comparing to AUC. That is, AUC is more tolerant to the noise impact on learning algorithms, particularly on naïve bayes and

logistic regression. When we study the data improvement, accuracy is more preferable than AUC.

The paper is organized as follows. A brief summary of the related literature is provided in Section 2. Section 3 reviews the different learning algorithms used to analyze their sensitivity under different noise levels, and the measures of supervised learning algorithms. Sections 4 and 5 presents an explanation of the mechanism used for generating noise and the analysis of the experimental results for the four supervised learning algorithms. Finally Section 6 provides conclusions about the performance of the algorithms under various levels of noise and future research directions.

2 Related Work

Real-world data always contain noise. In order to apply learning algorithms, data preprocessing is the first step. There are many tasks during data preprocessing. Among them, data cleaning is one of the important tasks. There exists much work on data cleaning, including detecting noise and reducing noise [8] [9] [10] [11] [2] [3]. There are two types of noise: attribute noise and class noise [9] [10]. Brodley [2] [3] detected and removed the class noise. Most of the work on noise detection is on supervised classification problems. However, Xiong et al. [8] studied the approach of remove noise for unsupervised learning. Kubica & Moore [11] studied how to identify and remove the attribute noise, such that the remaining information in the training examples can still be used in modeling. Different from previous work, this paper focuses on the impacts of class noise on supervised learning algorithms and investigates the noise reduction sensitivity of different learning principles through analyzing the performance of learning algorithms under different level of noise. Thus, we can choose the proper learning algorithm. This is also needed when the detecting and reducing noise approaches are applied to improve the data quality.

Improving the quality of the training data with mislabeled instances is important for supervised learning. Most of previous work focuses on removing the mislabeled instances on a specific learning algorithm – k-nearest neighbor (e.g., [3] [12] [13] and [14] created an instance selection mechanism for nearest neighbor classifiers). The algorithm proposed by [15] removes noise by retaining only those instances that have good impact on classifiers, which extends the nearest neighbor algorithm. Brodley and Friedl [3] identified and eliminated the mislabeled training instances by classifying each instance by an ensemble of classifiers. The ensemble of classifiers is built by three different learning algorithms (a 1-nearest neighbor, a linear machine, and a decision tree C4.5). An instance is removed if its original label is against the predictions of all three classifiers. Except the ensemble approach [3], the specific learning algorithm k-nearest neighbor is chosen for studying noise removing. To get under the noise impacts on different learning algorithms, we are going to investigate the performance of different supervised learning algorithms under the different noise levels, and their noise reduction sensitivity. It is expected to find which is better to be chosen among the popular supervised learning algorithms. We have done the investigation on four algorithms and will continue to investigate others.

Accuracy and AUC (area under the ROC curve) are two common measures for supervised learning algorithms. AUC is introduced into machine learning and data mining by [5] [6] from signal detection [4]. Bradley [16] indicates that AUC is more preferable to measure the performance of supervised learning algorithms. It discusses that AUC has several desirable properties compared to accuracy, such as increasing sensitivity in Analysis of Variance (ANOVA) tests, the independence of the decision threshold, and invariant to a priori class probability distributions. Ling [17] further compared the two measures with two formal criteria: (statistical) consistency, and (statistical) discriminancy. They formally prove that AUC is consistent with, and more discriminant (or finer) than accuracy, for the binary balanced datasets (which have the same number of positive and negative examples). In this paper, we compare AUC against accuracy under the noise situation. We investigate their noise reduction sensitivity and figure out which measure is preferable during improving data quality process.

3 Learning Algorithms

There exist more than 20 different learning algorithms, including the basic ones and improved variations. According to the categorization of WEKA [18], we choose the fundamental one from each category. That is, we are going to investigate the noise reduction sensitivity of four basic learning algorithms. They are decision tree, naïve bayes, support vector machine, and logistic regression.

A decision tree algorithm (DT in short) [19] partitions the input space into small sets, and labels them with one of the various output categories. That is, it iteratively selects a specific attribute to extend the learning model. According to the values of the specific attribute, a large group of cases are categorized into sub-groups. The essence of the algorithm is the criteria of selecting a specific attribute from the set of attributes available. There exist several criteria, such as accuracy improvement, entropy reduction, information gain, and gain ratio (details of these criteria can be found in [20]). The noisy label information will directly affect the estimation of all the criteria. Thus, the model built on decision tree algorithm would be affected by the noisy label.

Naïve bayes (NB in short) [21] is based on bayes theorem. Specifically, it is based on the properties of estimating the frequency of each value of every attribute under a given class from the obtained dataset. We can image that there is no difference of these estimation with/without noisy labels if both the class distribution and the distribution of the noisy label follow the complete random distribution. That is, naïve bayes is noisy-insensitive, particularly in estimating the conditional probabilities of each attribute value under given classes. We will see that there is little difference on the performance of naïve bayes under different noisy levels. Of

course, it is not always true that the conditions are satisfied for the real-world applications. Noise still impacts the performance of the model built on naïve bayes.

Support Vector Machine (SVM in short) [22] is one of the kernel approaches for classification. It constructs a hyperplane in high dimension space to classify cases into different classes. Its intuition is to find the hyperplane that has the largest distance to the nearest training cases. These nearest training cases are commonly referred as support vectors. According to the vectors on each side, a sub-hyperplane can be built for each side. The maximum margin between the two sub-hyperpanes is achieved to reduce the general classification errors. When the data have noise, it is possible that these support vectors could have noise too. Thus, the hyperplane and the two sub-hyperplanes (found based on support vectors) can vary. That is why support vector machine is very sensitive to noise. We further discuss this after describing experimental results.

Logistic regression (logistic in short) [23], like naïve bayes discussed above, is part of a category of statistical models called generalized linear models. However, unlike the independent assumption among the variables in naïve bayes, logistic regression has no such assumption. In addition, it also makes no assumption about the distribution of the independent variables. Although both logistic regression and naïve bayes are statistical models, the basic ideas of them are different. Logistic regression estimates the probability of being positive case (for binary classification). It can be inferred that the probability migrates toward the uniform distribution when more noise labels are involved. Thus, it is more noise sensitive. The experimental results also show this (refer to Section 5).

4 Experiment Setup

To study the noise reduction sensitivity of the four different learning algorithms (decision tree, naïve bayes, support vector machine, and logistic regression), we run experiments on all the classification datasets downloaded from WEKA website. In the experiments, we assume that these datasets are clean (no noise, especially no class noise). In order to have the datasets to study the noise reduction sensitivity, we inject noise into these datasets. Since we focus on investigation of the class noise, we only introduce noise into the class labels.

Here is the simulation procedure:

1. Repeat 10 times
   a. Divide a dataset into training part (70%) and test part (30%)
   b. Keep the original class labels in array for all training examples in the training part
   c. For each training example (simulate the noise labels with the control of the noise level)
      i. If it needs a wrong label, we randomly generate a label from the labels other than the original label to replace the original label
   d. Build classification model using a specific learning algorithm
   e. Make prediction for the test examples in test part, and output the classification accuracy
2. Output the average classification accuracy over 10 repeats

We studied the noise reduction sensitivity on the four learning algorithms step by step. We first focus on the impact of noise labels on binary classification. The features of the binary datasets from WEKA website are shown in Table 1.

TABLE 1
FEATURES OF THE 9 BINARY DATASETS USED IN THE EXPERIMENTS

	#Attributes	*#Examples*	*Class dist. (P/N)*
bmg	41	2417	547/1840
expedia	41	3125	417/2708
kr-vs-kp	37	3196	1669/1527
mushroom	22	8124	4208/3916
qvc	41	2152	386/1766
sick	30	3772	231/3541
spambase	58	4601	1813/2788
tic-tac-toe	10	958	332/626
travelocity	42	8598	1842/6756

We further study the performance of each learning algorithm for multiclass classification to see whether the noise impact on the performance on binary classification stays. The features of the multiclass datasets from WEKA website are shown in Table 2.

TABLE 2
FEATURES OF THE 18 MULTICLASS CLASSIFICATION DATASETS USED IN THE EXPERIMENTS

	#Attributes	*#Examples*	*#Classes*
anneal	39	898	6
audiology	70	226	24
autos	26	205	7
balance	5	625	3
Glass	10	214	7
heart-c	14	303	5
heart-h	14	294	5
iris	5	105	3
letter	17	20000	26
lymph	19	148	4
primary-tumor	18	339	21
segment	20	2310	7
splice	62	3190	3
thyroid	30	3772	4
vehicle	19	846	4
vowel	14	990	11
waveform	41	5000	3
zoo	18	101	7

From the procedure above, we can see that we have a parameter (noise level) to control the amount of noise introduced. It should be noted that the noise introduced here are random. In details, in the step 1.c.i of the simulation procedure above, for each training example, we randomly generate the noise labels to replace the original labels in the training data. Specifically, for binary classification, if the label of a training example is wrong, its opposite label should be

used to replace the original label. For multiclass classification, any one from the rest labels (different from the original label) has the same probability to be chosen to replace the original one.

5 Experimental Results

In this section, we compare the performance of different machine learning algorithms on the datasets with different percentages of label noise injected. The performance of different learning algorithms is measured in accuracy and AUC. Noise degree is controlled from 50% to 10%. In addition, to see the noise label impact, we also have the experimental results for each dataset under the four learning algorithms without noise. The results are obtained from the implementation of WEKA for the four learning algorithms default parameter settings. That is, we use J48 for decision tree, NaiveBayes for naïve bayes, SMO for support vector machine, and Logistic for logistic regression [18].

5.1 Binary classification

First, we investigated the performance of the learning algorithms on binary classification datasets listed in Table 1. The accuracy and AUC of each algorithm on each dataset is shown in Table 3 and Table 5 respectively. We counted the number of champions (the highest result among the four algorithms, *in italic*) for each learning algorithm from Table 3 and Table 5 respectively and showed it in Table 4 and Table 6 (the number in the bracket means the tied champion). Table 4 shows that DT wins more than other algorithms, measured in accuracy. When the noise level is reduced, the number of wins for DT increases. Overall, DT performs the best, followed by Logistic, followed by SVM. NB performs the worst. Table 6 shows that Logistic wins more than other algorithms, measured in AUC. Logistic performs the best, followed by DT, followed by SVM. NB performs the worst. This shows that the two measures (accuracy and AUC) do not perform the same under the noise situation. Taking both measures into consideration, DT and Logistic are more preferable than SVM and NB.

Besides, in order to visualize the average performance of each learning algorithm under difference noise levels, we plotted the average accuracy and AUC in Figure 1 and Figure 2 respectively. Both figures show that NB is the most noise tolerant among the four learning algorithms. Its performance does not change much when the noise level decreases (that is, the label quality increases). Its curve over the noise level is almost flat. According to average accuracy, Figure 1 shows that NB performs the best when the noise level is the highest (50%), followed by DT and Logistic, followed by SVM. However, when the noise level reduces, other three algorithms perform better. Especially, when the noise level reduces from 50% to 40%, the performance of all the three algorithms improves quickly. Although their performance keeps improving when the noise level continues to reduce, the acceleration of the improvement slows down. Among the three noise reduction sensitive learning algorithms, DT performs the best, followed by Logistic, followed by SVM. In other words, if we have approaches which can improve the quality of labels, DT is preferable, followed by Logistic and SVM.

TABLE 3
THE ACCURACY OF THE FOUR LEARNING ALGORITHM ON BINARY CLASSIFICATION

Dataset	%Noise	50%	40%	30%	20%	10%	0%
bmg	DT	*80.23*	*81.70*	*82.94*	*85.20*	*86.28*	*86.07*
	NB	69.92	67.60	68.92	66.74	66.41	65.54
	SVM	76.87	76.87	76.87	76.87	76.87	76.94
	Logistic	76.87	76.81	77.59	78.25	79.08	78.95
expedia	DT	87.39	87.42	90.61	*91.97*	*92.89*	*93.34*
	NB	85.88	85.68	85.93	85.40	84.60	83.54
	SVM	87.45	87.77	88.59	90.85	91.42	91.88
	Logistic	*89.20*	*90.04*	*91.11*	91.32	91.55	91.55
Kr-vs-kp	DT	*72.55*	*83.86*	*95.70*	*98.22*	*99.12*	*99.24*
	NB	66.48	72.93	79.58	83.88	86.10	87.30
	SVM	57.55	83.26	90.65	90.86	94.16	95.38
	Logistic	60.99	73.36	85.71	92.11	95.22	97.44
mushroom	DT	74.69	90.23	99.61	99.87	*100.0*	*100.0*
	NB	*96.14*	96.71	97.32	97.56	97.21	95.87
	SVM	74.09	*97.89*	*99.86*	*99.98*	99.98	*100.0*
	Logistic	74.63	96.07	99.74	99.93	99.98	*100.0*
qvc	DT	*83.97*	*85.47*	*86.95*	*87.81*	*88.88*	*88.95*
	NB	69.24	69.77	68.03	67.02	65.86	65.92
	SVM	82.40	82.40	82.54	82.84	83.27	83.26
	Logistic	82.79	82.98	83.50	83.64	83.26	83.12
sick	DT	*47.52*	*84.08*	*95.03*	*97.24*	*98.62*	*98.75*
	NB	40.37	64.17	83.16	82.93	88.74	92.14
	SVM	24.79	80.52	93.58	94.08	94.08	94.08
	Logistic	35.59	84.04	94.18	95.20	95.28	96.61
spambase	DT	47.57	*86.65*	*88.09*	89.41	91.17	92.38
	NB	*85.22*	86.02	86.36	85.35	77.22	79.45
	SVM	41.68	59.72	87.49	90.43	90.58	90.68
	Logistic	55.77	76.96	86.40	*90.46*	*91.72*	*92.88*
tic-tac-toe	DT	65.51	71.74	77.77	80.24	83.07	84.18
	NB	67.32	69.16	71.71	73.94	73.07	70.94
	SVM	65.75	*95.61*	*96.48*	*97.70*	*98.22*	*98.22*
	Logistic	*80.59*	91.15	95.26	96.17	97.91	*97.91*
travelocity	DT	50.09	83.14	95.87	*99.08*	*99.63*	*99.71*
	NB	*90.54*	*94.32*	94.90	94.87	94.89	95.01
	SVM	19.64	90.47	93.02	93.09	93.10	93.18
	Logistic	40.02	91.89	*96.45*	95.81	95.76	97.20
average	**DT**	**67.72**	**83.81**	***90.29***	***92.12***	***93.30***	***93.62***
	NB	***74.57***	**78.48**	**81.77**	**81.97**	**81.57**	**81.75**
	SVM	**58.91**	**83.83**	**89.90**	**90.74**	**91.30**	**91.51**
	Logistic	**66.27**	***84.81***	**89.99**	**91.43**	**92.20**	**92.85**

TABLE 4
NUMBER OF CHAMPIONS OF THE FOUR LEARNING ALGORITHM ON THE NINE BINARY CLASSIFICATION DATASETS (ACCURACY)

	%Noise	50%	40%	30%	20%	10%	0%
#Wins	DT	4	5	5	6	7	6+(1)
	NB	3	1	0	0	0	0
	SVM	0	2	2	2	1	1+(1)
	Logistic	2	1	2	1	1	1+(1)

5.2 Multiclass classification

We further study the noise reduction sensitivity of the four learning algorithms in multiclass classifications. Note that the implementation (SMO and Logistic) for support vector machine and logistic regression respectively can be directly applied to multiclass classifications. We have done the experiments on 18 datasets listed in Table 2. Because of the space limitation, we plotted the figures, instead of showing the results in Tables which occupy more space. Figure 3 shows the accuracy of each learning algorithm on each dataset, and

Figure 4 shows the AUC. We summarize the performance of the four learning algorithms by counting the champions and show the number of champions in Table 7 (in accuracy) and Table 8 (in AUC). Besides, we show the average performance of the four learning algorithms in Figure 5 (average accuracy) and Figure 6 (average AUC).

TABLE 5
THE AUC OF THE FOUR LEARNING ALGORITHM ON BINARY CLASSIFICATION

Dataset	%Noise	50%	40%	30%	20%	10%	0%
bmg	DT	0.705	*0.758*	*0.783*	*0.816*	*0.835*	*0.851*
	NB	0.723	0.723	0.726	0.729	0.732	0.731
	SVM	0.500	0.500	0.500	0.500	0.500	0.503
	Logistic	*0.767*	0.757	0.769	0.771	0.774	0.776
expedia	DT	0.521	0.544	0.782	0.826	0.865	0.877
	NB	0.805	0.814	0.821	0.822	0.821	0.822
	SVM	0.515	0.527	0.566	0.655	0.687	0.726
	Logistic	*0.904*	*0.904*	*0.909*	*0.911*	*0.912*	*0.914*
kr-vs-kp	DT	0.943	0.955	*0.982*	*0.991*	*0.996*	*0.997*
	NB	0.928	0.928	0.930	0.934	0.938	0.948
	SVM	0.592	0.839	0.908	0.909	0.941	0.954
	Logistic	*0.970*	*0.976*	0.980	0.984	0.990	0.996
mushroom	DT	0.970	0.987	0.997	0.999	1.000	1.000
	NB	0.997	0.997	0.997	0.997	0.998	0.998
	SVM	0.750	0.980	0.999	1.000	1.000	1.000
	Logistic	*1.000*	*1.000*	*1.000*	*1.000*	*1.000*	*1.000*
qvc	DT	0.687	0.768	0.794	0.822	*0.844*	*0.867*
	NB	0.733	0.745	0.745	0.746	0.744	0.746
	SVM	0.500	0.500	0.505	0.525	0.542	0.547
	Logistic	*0.812*	*0.818*	*0.823*	*0.827*	0.828	0.831
sick	DT	0.848	0.897	0.906	0.891	0.926	0.930
	NB	*0.884*	0.902	0.905	0.915	0.915	0.923
	SVM	0.575	0.694	0.504	0.500	0.500	0.500
	Logistic	0.880	*0.910*	*0.917*	*0.932*	*0.937*	*0.933*
spambase	DT	0.893	0.891	0.898	0.908	0.919	0.934
	NB	0.922	0.927	0.929	0.931	0.936	0.937
	SVM	0.524	0.67	0.883	0.897	0.895	0.893
	Logistic	*0.950*	*0.953*	*0.956*	*0.959*	*0.964*	*0.972*
tic-tac-toe	DT	0.500	0.641	0.745	0.794	0.840	0.895
	NB	0.739	0.758	0.756	0.76	0.762	0.764
	SVM	0.504	0.938	0.948	0.966	0.974	0.974
	Logistic	*0.983*	*0.981*	*0.980*	*0.984*	*0.992*	*0.997*
travelocity	DT	0.927	0.959	0.978	*0.991*	*0.995*	*0.995*
	NB	0.934	0.932	0.925	0.911	0.920	0.918
	SVM	0.544	0.579	0.558	0.556	0.557	0.563
	Logistic	*0.972*	*0.980*	*0.984*	0.979	0.986	0.972
average	**DT**	**0.777**	**0.822**	**0.874**	**0.893**	**0.913**	**0.927**
	NB	**0.852**	**0.858**	**0.859**	**0.861**	**0.863**	**0.865**
	SVM	**0.566**	**0.692**	**0.708**	**0.723**	**0.733**	**0.740**
	Logistic	***0.915***	***0.920***	***0.924***	***0.927***	***0.931***	***0.932***

TABLE 6
NUMBER OF CHAMPIONS OF THE FOUR LEARNING ALGORITHM ON THE NINE BINARY CLASSIFICATION DATASETS (AUC)

	%Noise	50%	40%	30%	20%	10%	0%
#Wins	DT	0	1	2	3	3+(1)	3+(1)
	NB	1	0	0	0	0	0
	SVM	0	0	0	(1)	(1)	(1)
	Logistic	8	8	7	6+(1)	6+(1)	6+(1)

Figure 5 verifies the conclusions we made from the experimental results for binary datasets: naïve bayes is the most noise tolerant learning algorithm and it performs well when the noise level is high (50%). Although it performs better when the noise level reduces, the performance of the other three learning algorithms improves more significantly. When the noise level is higher than 20%, SVM performs the best in average, followed by DT, followed by Logistic. However, when the noise level is further reduced, DT performs the best, followed by SVM, followed by Logistic. Table 7 also shows that DT wins most champions, especially after the noise level is reduced to less than 30%. Surprisingly, NB wins more champions than SVM and Logistic, especially when the noise level is not lower than 40%.

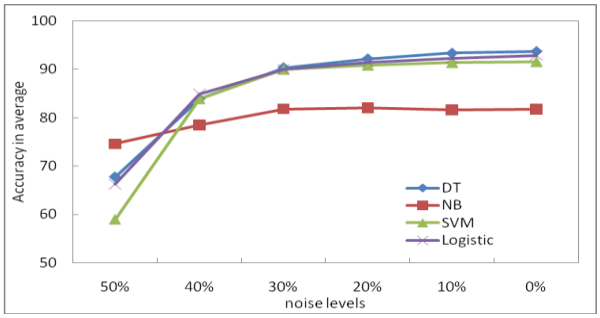


Figure 1: The average accuracy over the nine binary datasets for the four learning algorithms.

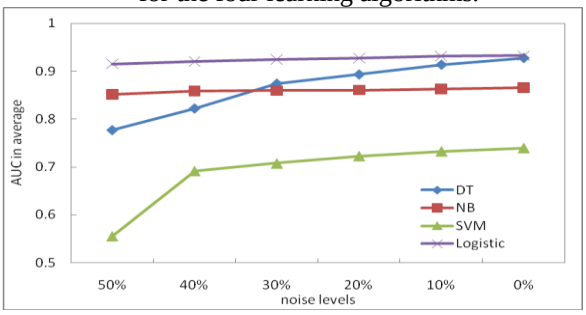


Figure 2: The average AUC over the nine binary datasets for the four learning algorithms.

TABLE 7
NUMBER OF CHAMPIONS OF THE FOUR LEARNING ALGORITHM ON THE 18 MULTICLASS CLASSIFICATION DATASETS (ACCURACY)

	%Noise	50%	40%	30%	20%	10%	0%
#Wins	DT	4	4	5	8	10	6
	NB	6	6	4	5	5	2
	SVM	4	3	6	3	2	3
	Logistic	4	5	3	2	1	7

The experimental results in AUC are shown in Figure 4, with the average AUC shown in Figure 6. The summarization of the number of champions of each learning algorithm in AUC is displayed in Table 8.

TABLE 8
NUMBER OF CHAMPIONS OF THE FOUR LEARNING ALGORITHM ON THE 18 MULTICLASS CLASSIFICATION DATASETS (AUC)

	%Noise	50%	40%	30%	20%	10%	0%
#Wins	DT	0	0	1	2	2	2
	NB	10	9	9	8	6	6
	SVM	1	2	2	1	1	1
	Logistic	7	7	6	7	9	9

Surprisingly, the experimental results in AUC show us different observations. NB wins most champions showing in Table 8. When the noise level is lower than 20%, Logistic wins most champions. DT and SVM only win a few. Figure 6 also shows that NB dominates other in AUC, followed by Logistic, SVM, and DT. Again, the different observations from accuracy and AUC further confirm that the two measures

perform differently in noise situations. The surprising observation based on AUC motivates us to further study NB and Logistic under noise in the future. It can be expected that we can improve the accuracy of NB and Logistic by adjusting the classification threshold, instead of using the classical 0.5. We also observe that SVM performs well in accuracy on binary and multiclass classification, but its AUC is very low. What is the reason? We are going to investigate this and to further improve its performance in AUC.

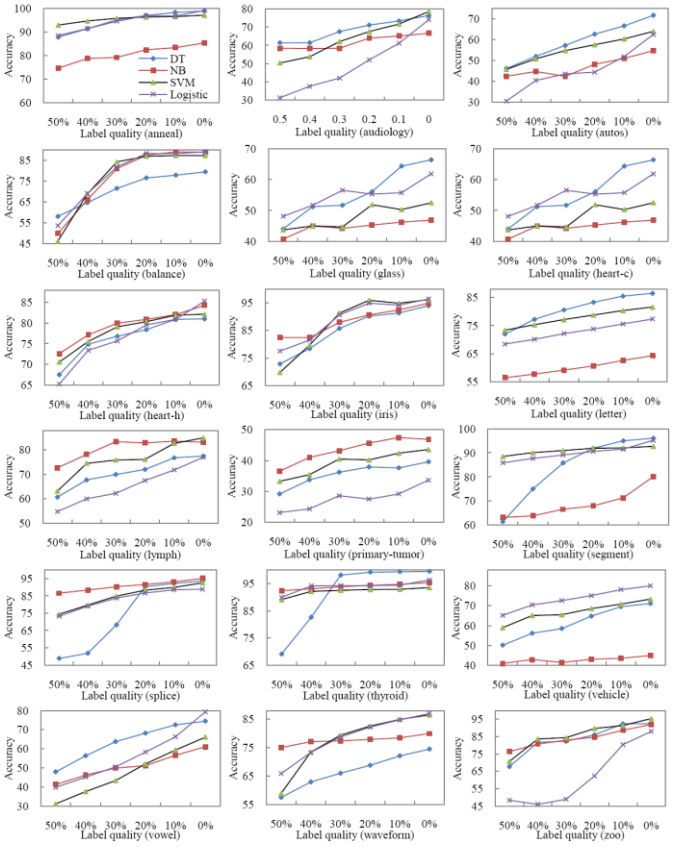


FIGURE 3. THE ACCURACY OVER THE 18 MULTICLASS DATASETS FOR THE FOUR LEARNING ALGORITHMS.

6 Conclusions and Future Work

In this research, we studied on how the quality of models is affected by different amounts of noise for different machine learning algorithms. The study was performed on four different classifiers called decision tree, naïve bayes, support vector machine, and logistic regression. A detailed experimentation proves that the behavior of each algorithm depends on the percentage of noise injected and the characteristics of different datasets.

We also investigated the noise reduction sensitivity of the two measures (accuracy and AUC). It is observed that AUC is more noise tolerant. The improvement of AUC is slower with the noise reduction. When we study the noise reduction (quality improvement), accuracy is the preferable measure.

This study is very useful in situations of real-world data processing that may contain implicit and explicit errors. The results in accuracy show that naïve bayes is the most noise tolerant algorithm. However, decision tree performs the best overall under different noise level for most datasets (binary and multiclass), followed by logistic regression and support vector machine. However, logistic regression performs the best in AUC. When we develop approaches to improve the data quality (reduce the noise level), they are more preferred than naïve bayes. Besides, we are also interested in studying how to improve the accuracy performance of naïve bayes, as it performs well in AUC; and how to improve the AUC performance of support vector machine, as it performs well in accuracy.

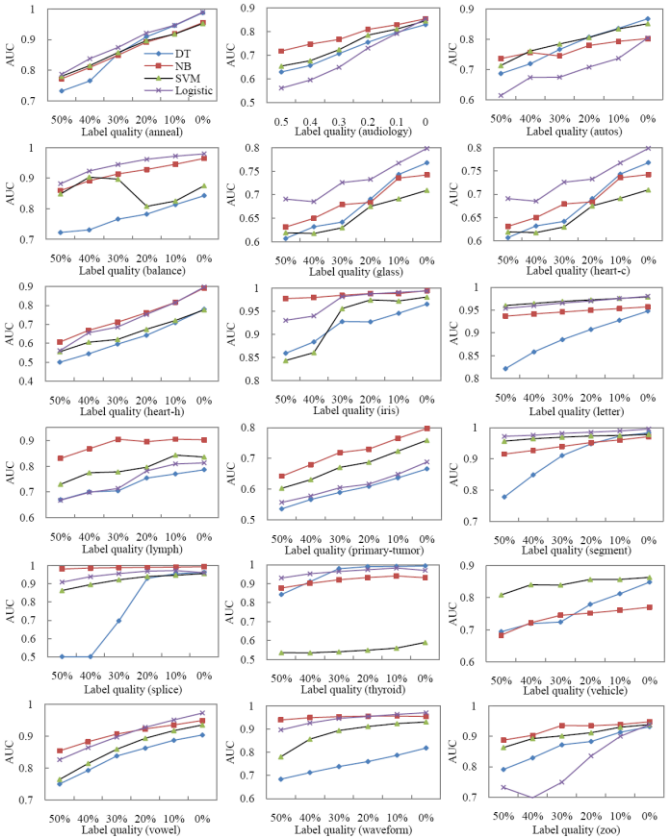


FIGURE 4. THE AUC OVER THE 18 MULTICLASS DATASETS FOR THE FOUR LEARNING ALGORITHMS.

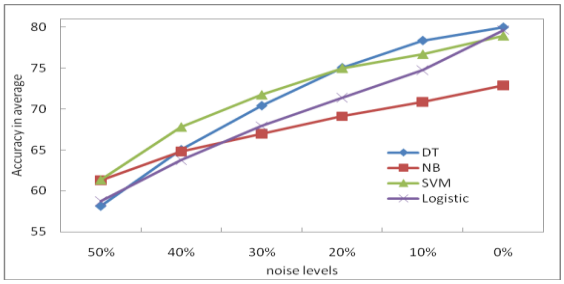


Figure 5. The average accuracy over the 18 multiclass datasets for the four learning algorithms.

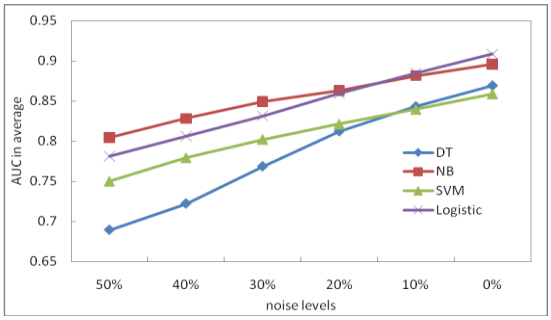


Figure 6. The average AUC over the 18 multiclass datasets for the four learning algorithms.

7 Acknowledgment

We thank the anonymous reviewers for the valuable comments. The work was supported by the National Science Foundation (IIS-1115417).

8 References

[1] Sheng, V.S., Provost, F. and Ipeirotis, P. Get another label? Improving data quality and data mining using multiple, noisy labelers. Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2008), 2008, 614-622.

[2] Brodley, C.E. and Friedl, M.A. Identifying and eliminating mislabeled training instances. Proceedings of 13th National Conf. on Artificial Intelligence, 1996, 799-805.

[3] Brodley, C.E. and Friedl, M.A. Identifying mislabeled training data, Journal of Artificial Intelligence Research, 1999, 11, 131-167.

[4] D.M. Green and J.A. Swets. Signal Detection Theory and Psychophysics. Wiley, New York, 1966.

[5] F. Provost and T. Fawcett. Analysis and visualization of classifier performance: comparison under imprecise class and cost distribution. Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1997, 43-48.

[6] F. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In Proceedings of the Fifteenth International Conference on Machine Learning, Morgan Kaufmann, 1998, 445-453.

[7] J. Huang and C.X. Ling. Constructing New and Better Evaluation Measures for Machine Learning. The Twentieth International Joint Conference on Artificial Intelligence (IJCAI 2007), 859-864.

[8] Xiong, H., Pandey, G., Steinbach, M. and Kumar, V. Enhancing data analysis with noise removal. IEEE Transactions on Knowledge and Data Engineering, 2006, 18, 304-319.

[9] Zhu, X., Wu, X. and Chen, Q. Eliminating Class Noise in Large Datasets. Proceedings of the 20th ICML International Conference on Machine Learning (ICML 2003). Washington D.C., 2003, 920-927.

[10] Zhu, X., Wu, X. and Chen, Q. Bridging Local and Global Data Cleansing: Identifying Class Noise in Large, Distributed Data Datasets. Data Mining and Knowledge Discovery, 2006, 12, 275-308.

[11] Kubica, J., and Moore, A. Probabilistic Noise Identification and Data Cleaning, Proceedings of the third IEEE International Conference on Data Mining (ICDM'03), 2003, 131-138.

[12] Gamberger, D.; Lavrac, N.; and Dzeroski, S. 1996. Noise elimination in inductive concept learning: A case study in medical diagnosis. In In Proc. of the 7th International Workshop on Algorithmic Learning Theory, 199-212. Springer.

[13] Wilson, D. R., and Martinez, T. R. 2000. Reduction techniques for instance-basedlearning algorithms. Journal of Machine Learning 38:257-286.

[14] Skalak, D. B. 1994. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In Proceedings of the Eleventh International Conference on Machine Learning, 293-301.

[15] Aha, D. W.; Kibler, D.; and Albert, M. K. 1991. Instance based learning algorithms. Journal of Machine Learning 6:37-66.

[16] A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition, 30:1145-1159, 1997.

[17] C. X. Ling, J. Huang, and H. Zhang. AUC: a statistically consistent and more discriminating measure than accuracy. In Proceedings of 18th International Conference on Artificial Intelligence (IJCAI-2003), 2003, 519-526.

[18] Witten, I.H., and Frank, E. Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed. Morgan Kaufmann Publishing, 2005.

[19] J.R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann: San Mateo, CA, 1993.

[20] Tom Mitchell, Machine Learning, McGraw Hill, 1997

[21] John, G. H., and Langley, P. Estimating Continuous Distributions in Bayesian Classifiers. In Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, 1995, 338-345.

[22] Corinna Cortes and V. Vapnik, Support-Vector Networks, Machine Learning, 20, 1995.

[23] Hilbe, Joseph M. Logistic Regression Models. Chapman & Hall/CRC Press, 2009.

A Novel Protein Secondary Structure Intelligent Prediction System

Bingru Yang*
School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China

Abstract – *Protein secondary structure prediction is one of major challenges in bioinformatics, data ming. In this paper, we propose a novel intelligent prediction system model--Compound Pyramid System Model, which may become the classic model for predicting protein secondary structure. It consists of four components by intelligent interfaces and synthesizing several methods such as SAC (Structural association classifier), AAC (Attribute association classifier) and KDTICM. The model is applied to the domain knowledge, and the effective attributes are chosen by Causal Cellular Automata. Assessments using RS126 and CB513 datasets indicate that the CPSM method can achieve average Q3 accuracy approaching 84.31% and 86.78%. The prediction result of in CASP8 dataset shows that its performance is better than previously reported methods and accessible prediction servers. The result shows that our method has strong universality ability. The fully automated prediction server of CPSM is available at http://kdd.ustb.edu.cn/protein_ Web/. which has a significant international impact.*

Keywords: Compound Pyramid Model, Protein secondary structure Prediction, Data Mining, Hybrid Prediction Model

1 Introduction

Recently, more and more protein sequence data show explosive growth[1,2], while the increasing numbers of protein sequences are much greater than that of already known[3], it's urgent to find out some effective data mining methods for solving this problem.

Reviewing the development history of protein secondary structure prediction, many approaches are continuously benchmarked, such as data mining [4,5], support vector machines [6], hidden Markov models [7,8], and so on. Classical algorithms include: PSIPRED, SAM, PORTER and PROF. But we can find that these methods, to some extent, possess problems with low accuracy which are less than 80% and the prediction stability is not very good.

To solve these problems, we propose a novel protein secondary structure intelligent prediction system model—Compound Pyramid System Model (CPSM), which combine KDD* process model based on Knowledge Discovery Theory based on Inner Cognitive Mechanism Theory (KDTICM) [9] with structural association classifier (SAC) [10], attribute association classifier (AAC) [10] and other technical method. Experiments show that the CPSM can achieve excellent Q3 accuracy on RS126, CB513 and CASP8.

2 Theory Basis of CPSM

2.1 KDTICM

The author originally brought forward a new research direction which data mining should be done based on inner cognitive mechanism that data mining is considered as a procession of cognizing, and the knowledge discovery system is considered as cognize system, applying system information and cognitive science to research complicated knowledge discovery system. The author also constructs Knowledge Discovery Theory based on Inner Cognitive Mechanism[9].

1).Double bases cooperation mechanism. Based on "intention creation" and "psychology information restore" in cognitive psychology, we find the relationship of database and knowledge base under specific construction in the process of KDD, demonstrate the conformation mapping theorem, design the heuristic coordinator and the maintaining coordinator, and resolve puzzles of "directional searching", "directional mining", independent discovery and real time maintenance.

2) Conformation mapping theorem:There is an equivalent relations between the inferential category of universe X, Cr(N) and complete data substructure reachable category $C\propto<\gamma,\Re c(\gamma)>$. We obtain this theorem by applying two methods, one is category theory, and the other is the extension of continuous maps theory.

3) The theorem establishes one-to-one correspondence between knowledge single node and "data substructure" in database. Double bases cooperation mechanism resolves the problem of "directional searching" and "directional mining", we proposes and realizes two cooperated algorithms: firstly, real time maintenance for domain knowledge base through maintenance cooperate algorithm and component; secondly, find knowledge shortage independently to produce intention creation through the heuristic coordinate algorithm and component.

2.2 KDD*--new process model derived from double cooperation mechanism

We integrate the double basis cooperation mechanism and construction of two cooperators into classical KDD

process, form the new process model- KDD* independently, then change the old knowledge discovery process essentially. KDD*process model is shown in Figure 1.

The basic feature of KDD* process model are as follows:

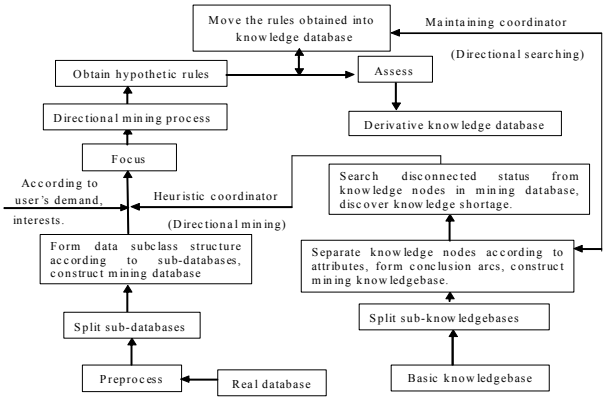


Figure 1. Compound Pyramid System Model

① Domain knowledge is applied in the mining process directly through two cooperators; this idea is derived from synchronization evolution and cooperation computation.

② System can produce directional focus through the adjacency matrix of directed hyper graph, and mining knowledge shortage independently.

③ Focus problem: direction and process of directional mining will only produce when user's interests match knowledge shortage system find independently. So it will not mine a great deal of repeated, redundant knowledge, decrease rule evaluation greatly. The purpose is to decrease search space, improve the algorithm efficiency, and provide necessary technology support.

④ With accumulation of knowledge, in order to reflect application quickly, the maintenance coordinator is added to the new model, process the repetition, redundancy, confliction, circle and hypostasis effectively, dynamically on real time.

⑤ The new model is based on the two cognitive features –"intention creation" and "psychology information restore" in cognitive psychology, so the new model has its solid theory base; and the implement of this model is based on the theory.

2.3 Associated analysis—Maradbcm algorithm derived from double bases cooperation mechanism and KDD*

Based on KDD* process model, we proposed a new association analysis algorithm, Maradbcm (for short, we call it as M algorithm), Process as followed:

Input: Rule strength threshold Min_Intensity, support threshold Min_Sup, confidence threshold Min_Con;

Output: Association rule base KD.

1. Data Preprocess;
2. When "shortage of knowledge" is detected
3. Create K2; //Km denotes the shortage knowledge whose length is m, namely Km={r| Len(r)=m}.
4. m = 2;
5. Create hypothesis of knowledge Km; // Directional mining the shortage of knowledge ri in Km.
6. Repeat
7. For every ri in Km
8. If (ri is conformed with present knowledgebase && the measure of ri is qualified)
9. move ri into KD, update reachable matrix;
10. Else
11. delete ri;
12. End for;
13. m = m+1;
14. Until Km=ϕ;
15. EndWhen;

2.4 Testing Datasets

We have used three different datasets to test our novel method:

RS126[11]: This original dataset containing 126 sequences is provided by Rost and Sander.

CB513[12]: This dataset containing 513 sequences is developed by Cuff and Barton, it is one of the most widely used independent dataset in protein secondary structure prediction.

CASP8[13]: Critical Assessment of Techniques for Protein Structure Prediction (CASP) is an international competition for protein structure prediction. the CASP8 dataset from the Protein Prediction Center http://predictioncenter.org/ .

2.5 Evaluation of Prediction Accuracy

There are different measures used to assess secondary structure prediction methods. The most common measure is Q3 accuracy.

Q3 accuracy: the percentage of the number of correctly estimated structures in the overall predictions. This measure depends on the accurately predicting numbers of three-state per-residue. The formula is as follows:

$$Q_3 = \frac{\sum_{i\in\{H,E,C\}} \mathrm{Prd}_i}{\sum_{i\in\{H.E.C\}} \mathrm{Class}_i} \times 100\% \qquad (1)$$

Prd_i denotes the correctly estimated residue numbers, in which i means three secondary structures. Class_i denotes the three different residue numbers.

3 Compound Pyramid System Model

Modern research shows that multi-hierarchical configuration is an effective method for reducing system complexity, and ordered granularity space theory is one of the effective methods for establishing multi-hierarchical configuration of complex system. Its configuration is shown in Figure 2.

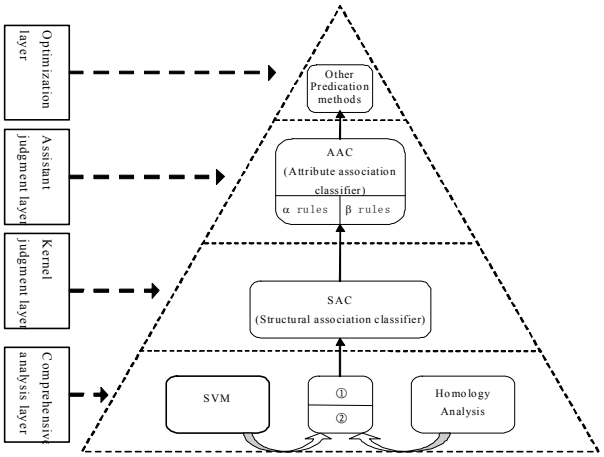


Figure 2. Compound Pyramid System Model

As Figure 2 shows, CPSM contains four layers, which are comprehensive analysis layer, kernel judgment layer, assistant judgment layer and optimization layer. CPSM has higher accuracy than those traditional prediction models and methods.

3.1 Comprehensive analysis layer

This layer is the basic layer of the whole model, integrate improved homologous analyze and optimized SVM multi-classifying method, and complete more than 70% percent of the prediction of amino acids.

3.1.1 Improved Homologous Analysis

The homologous sequence method is widely used in the field of the protein secondary structure prediction. In our experiment, we use two neural network packages: SNNS neural network and BPJoone neural network. We apply a neural network with a sliding window of 13 residues over each amino acid in multi-sequence alignment result produced by the PSI_BLAST algorithm, plus three physicochemical properties (hydrogen bond, hydrophobia and electricity) as the input nodes. The above network comprises nine, twenty hidden nodes respectively and three output nodes. The input of the standard neural network is a window, and the output is three kinds of secondary structures (H, E and C).

3.1.2 Optimized SVM Multi-classifying Method

Avoid using too many capital letters. All section headings including the subsection headings should be flushed left.

SVM multi-classification method applies combined classifying methods such as structure and attribute[14].

Our substantial tests show that the RBF (radial basis function) kernel, defined as,

$$K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2) \qquad (2)$$

RBF is appropriate for complex classification problems, when parameters and C are selected from the optimization process. The optimized parameters are chosen based on the results of each step and in both binary classifiers, are γ = 0.05 and C = 2.0 on RS126 , γ = 0.05 and C = 1.0 on CB513, γ = 0.05 and C = 1.0 on CASP8. We construct the SVM classifiers with these parameters respectively.

Before constructing a tertiary classifier, we firstly construct several SVM binary classifiers including three one-versus-rest classifiers (say," one": positive class," rest": negative class) named H/~H,E/~E and C/~C, and three classifiers named H/E,E/C and C/H which distinguish the sample between each of two states.

3.2 Kernel judgment layer and Assistant judgment layer

These two layers are the core parts of CPSM, which aim for processing those un-classified data in the Comprehensive analysis layer. We propose a novel algorithm named KAAPSSP (KDD* Association Analysis Protein secondary structure prediction), which combines the Maradbcm algorithm and ICBA algorithm.

3.2.1 Inter-Sequence Interactions Theory

In 2004, Simossis and Heringa[15] found that the correlations between neighboring amino acids are essentially uninformative and only 1/4 of the total information which needed to determine the secondary structure is available from local inter-sequence correlations. These observations support the view that the majority of most proteins fold via a cooperative process where secondary and tertiary structure form concurrently. Generally the methods based on knowledge for protein secondary structure prediction start from the primary structure, namely by considering the primary structure about the sides of the goal residue to predict its secondary structure. Firstly the sequence of amino acids in the peptide chain should be spitted into several windows, whose length is usually between 13 and 17. On the basis of knowledge learned from those windows, from the properties about the sides of center residue, the secondary structure of the center residue could be predicted .

3.2.2 KAAPSSP1---SAC(Structural association classifier) algorithm

We mine high accuracy and good fitness association rules from training dataset, and use the above theory to design structural association classifier algorithm SAC. We use SAC algorithm to mine dataset RS126, and obtain the following results, which is shown in Table 1.

TABLE I. A PART OF RS126 STRUCTURE DATABASE MINING RESULTS

RuleID	Condition	Result	Support	Confidence
1	Negative6H	CenterStructureH	0.180932	0.580645112
2	Negative5H,Positive4H	CenterStructureH	0.134312	0.682655712
3	Negtative4E	CenterStructureE	0.101231	0.994865125
4	Negtative3E	CenterStructureE	0.102318	0.987121524
5	Negtative2C	CenterStructureC	0.381257	0.890542132
6	Positive2C	CenterStructureC	0.386423	0.998095924
7	Negtative1C	CenterStructureC	0.432474	0.783655676
8	Positive1C	CenterStructureC	0.436790	1.000000000
9	Negtative2H	CenterStructureH	0.271264	0.990615372
10	Negtative2H,Negtative1H	CenterStructureH	0.261689	0.980275131

3.2.3 KAAPSSP2---AAC (Attribute association classifier) algorithm

Hua and Sun [16] indicate that the physicochemical properties of amino acids decisively affect the spatial conformation of protein. However, as we known, there are many physicochemical properties of amino acids, such as hydrogen bond, carbon circle, hydrophobia, electricity, residue size, fat and electric quantity et al. It is impossible to consider all of these properties in the process of learning. Based on Causal Cellular Automata Theory [17], we choose the hydrogen bond, hydrophobia and electricity as considered properties, which are shown in Figure 3, and then analyze the relationship between protein physicochemical properties and secondary structure.

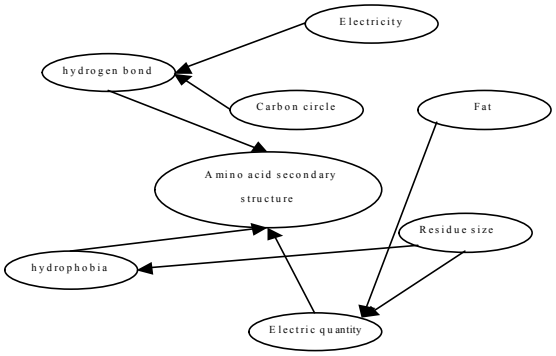


Figure 3. Causal Cellular Automata Based on Knowledge Discovery

Similarly, we mine high accuracy and good fitness association rules from training dataset, and use the above theory to design attribute association classifier algorithm AAC. We use AAC algorithm to mine dataset RS126, and obtain the following results, which is shown in Table 2.

TABLE II. A PART OF RS126 STRUCTURE AND PHYSICOCHEMICAL PROPERTIES DATABASE MINING RESULTS

RuleID	Condition	Result	Support	Confidence
1	HydropgenGlobal(0.57.0.60]	CenterStructureH	0.090935	0.995805865
2	HydrophobicLeft3(2.5]	CenterStructureH	0.281751	0.99577306
3	HydrogenGlobal(0.62,0.65],IsoeletricpointLeftrange(24,36]	CenterStructureH	0.432767	0.988526462
4	HydropgenGlobal(0.62,0.65],HydrogenRight5(0.6,0.7]	CenterStructureH	0.370103	0.98951407
5	HydrophobicLeft4(2,5],HydrogenRight6(0.7,0.7]	CenterStructureH	0.199569	0.98951407
6	HydrophobicLeft4(2,5],IsoeletricpointLeftrange(4,6]	CenterStructureH	0.218082	0.99498938
7	HydropgenTightrange(2,3],HydropgenGloabal(0.62,0.65],.....	CenterStructureH	0.076872	0.99281417
8	HydropgenTightrange(3,4.5],HydropgenGloabal(0.62,0.65],.....	CenterStructureH	0.394451	0.990146451
9	HydropgenTightrange(3,4.5],HydropgenGloabal(0.62,0.65],.....	CenterStructureH	0.147333	0.992268371
10	HydropgenTightrange(3,4.5],HydropgenGloabal(0.62,0.65],.....	CenterStructureH	0.08266	0.995399918

3.2.4 Adoption of KAAPSSP

KAAPSSP not only makes the compound pyramid prediction system model keep a high global accuracy, but also provides several interpretable association rules (as shown in Table.1 and Table.2), which are very useful for study of protein spatial structure folding.

3.3 Optimization layer

This layer mainly designs three methods such as tendency factor, potential function and reasonable inference. The first two kinds of methods are belong to bioinformatics methods, these methods predict the structure using bioinformatics background; the reasonable inference method is on the basis of the physical and chemical properties rules. These three kinds of methods optimize the results of three layer, then it can improve the whole predict accuracy.

4 Experiment and Analysis

We select RS126, CB513 and CASP8 datasets in our experiment. At the same time, we use Q3 accuracy as evaluation standard, which is denoted as percentage of accurately predicting amino acid in total amino acid. Each layer's results of CPM are shown in Table 3 and Table 4.

TABLE III. EACH LAYER PREDICTION ACCURACY AND SCALE OF CPM ON THE RS126 DATA SET

Module	Accuracy	Percent
Comprehensive Analysis Layer	16937/18646= 91.15%	18646/24806=75.17%
Kernel judgment layer	3885/6053= 64.18%	6053/24806= 24.40%
Assistant judgment layer	15/107= 14.02%	107/24806=0.43%
Total	20896/24806 = 84.31%	

TABLE IV. EACH LAYER PREDICTION ACCURACY AND SCALE OF CPM ON THE CB513 DATA SET

Module	Accuracy	Percent
Comprehensive Analysis Layer	106342/113638= 93.58%	113638/146233=77.71%
Kernel judgment layer	19961/32194= 62.00%	32194/146233= 22.02%
Assistant judgment layer	86/ 401= 21.45%	401/146233= 0.27%
Total	126901/146233 = 86.78%	

The accuracy comparison with other research methods separately on the RS126 and CB513 datasets, experimental results are shown in Table 5 and Figure 4.

TABLE V. ACCURACY COMPARISON WITH OTHER RESEARCH RESULTS ON THE RS126 AND CB513 DATA SET.

Method	Q3(RS126)	Q3(CB513)
Ref[7]	76.10%	76.60%
Psipred[18]	81.01%	79.95%
Prof[19]	76.95%	77.13%
PHD Expert[20]	76.92%	77.61%
SSPRO[21]	77.01%	79.07%
JPRED[22]	73.82%	73.37%
SAM[23]	78.81%	78.17%
Predator[24]	80.06%	80.04%
Ref[25]	81.65%	80.99%
CPSM	84.31%	86.78%

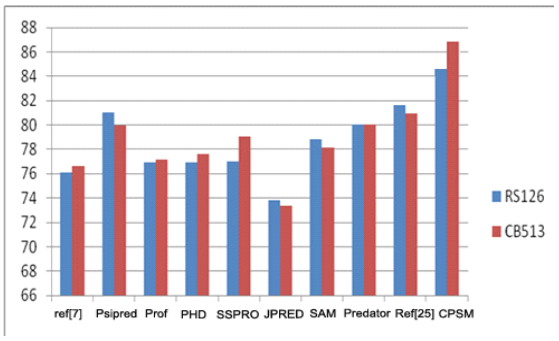


Figure 4. Q3 Accuracy comparison with other research results on the RS126 and CB513 dataset.

We select the best eight prediction servers to conduct experiments on dataset CASP8, the prediction results are shown in Figure 5. From the experimental results, we can see that Q3 accuracy and standard deviation of CPSM secondary structure prediction server are better than that of the other 8 international secondary structure prediction server.

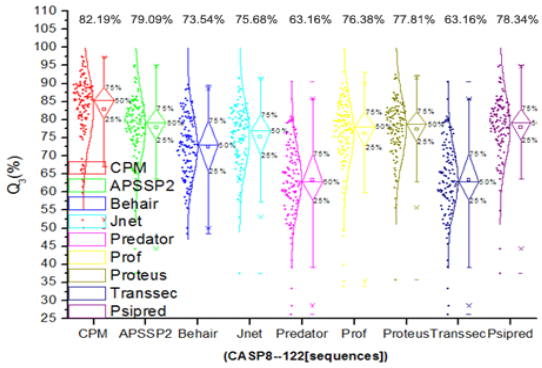


Figure 5. Testing result distribution of secondary prediction server CASP8 (numbers on figure are Q3 accuracy)

Now we can see that CPSM achieves the excellent predict accuracy compared to other methods and servers, it is because of the following model's innovation.

1) CPSM is a strictly innovated, auto and general-service system model. all models and algorithms in CPSM are innovated or improved. We neither use any research results from other severs nor apply any users' subjective experience. The sever can provide auto and open service, which has the function of predicting protein database or protein unit, and efficiently improve the universality of our intelligence system.

2)CPSM is a novel intelligent prediction system model. It is neither a common model nor a combined prediction method. It contains model, method and optimization components. Our method is proved to be effective in resolving the problem of protein secondary structure prediction. we try to breakthrough the bottle-neck of technological path and methodology.

3)From the point of system model, there isn't any other researchers constructing prediction model similar with composing pyramid system model, and no researchers combine physical attribute determinant and structure sequence determinant to form method, no researcher apply field knowledge and background knowledge in the system model.

4)From the point of methodology, we have present a system methods, which are combined by many relevant predicting methods. Among those methods, some are original, such as KDTICM innovative technology (SAC algorithm and AAC algorithm); some are improved methods, such as SVM apperception analysis. Additionally, the Causal Cellular Automata theories can improve the predicting accuracy.

5)From the point of system optimization, in the deduction of CPSM, from comprehensive analysis layer, kernel judgment layer and assistant judgment layer to optimization layer, granularity space of every layer is becoming thinner, which shows perforation of field knowledge and background knowledge, and this can ensure the prediction accuracy.

5 Conclusions

The CPSM method is both an integrated web server and standalone application that exploits recent advancements in data mining and machine learning to perform very accurate protein secondary structure predictions. The CPSM method combines four different high-performing prediction methods such as KDTICM innovative technology (SAC algorithm and AAC algorithm). For protein sequence the CPSM is able to achieve a very high level of accuracy higher than that previously reported and accessible prediction servers. The program's performance was extensively tested and compared to both available programs and publicly accessible web servers using a variety of test proteins and test scenarios. In most cases the CPSM appears to perform better than existing tools. At the present time, the CPSM website is accessible at http://kdd.ustb.edu.cn/protein_Web/, the prediction results are sent automatically to the user via email. In the future, we will continue to improve and perfect the prediction server.

6 References

[1] P. Baldi, S. Brunak, P. Frasconi, G. Soda, G. Pollastri, Exploiting the past and the future in protein secondary structure prediction, Bioinformatics 15(11):937–946, 1999.

[2] Fischer, D. and Eisenberg, D. Protein fold recognition using sequence-derived prediction. Prot. Sci.5, 947-955, 1996.

[3] Longfei Yang, Zhi Sun. Protein molecule structure. Tsinghua university publishment, 1999.

[4] Haoudi,A. Bensmail,H. Bioinformatics and data mining in proteomics. Expert Review of Proteomics, 3(3):333-343, 2006.

[5] J Y Li, L S Wong, Q Yang. Data mining in Bioinformatics. Intelligent Systems, 20 (6):16-18, 2005.

[6] Hua, S. J. and Sun, Z. R. A novel method of protein secondarystructure prediction with high segment overlap measure: support vector machine approach. J. Mol. Biol., 308 (2):397–407, 2001.

[7] K. Karplus, R. Karplus, J. Draper, et al, Combining local-structure, foldrecognition,and new fold methods for protein structure prediction, Proteins., 53(6) :491-496,2003.

[8] H N Lin, J M Chang, K P Wu, T Y Sun, W L Hsu. HYPROSP II--a knowledge-based hybrid method for protein secondary structure prediction based on local prediction confidence. Bioinformatics, 21(15):3227-3233,2005.

[9] Bingru Yang. Knowledge discovery theory based on inner cognitive mechanism. Electron industry publishment, 2004.
[10]Bingru Yang, Wei Hou, et al. KAAPRO: An approach of protein secondary structure prediction based on KDD* in the compound pyramid prediction model, Expert Systems with Applications, 36(5): 9000-9006,2009.
[11]B. Rost, C. Sander. Prediction of protein secondary structure at better than 70% accuracy. J. Mol. Biol., 232(2): 584-599,1993.
[12]J. A.Cuff, and G. J Barton,. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. Proteins: Structure, Function and Genet,. 34(4): 508-519,1999.
[13]http://predictioncenter.org/
[14]Jian Guo,Hu Chen,Zhirong Sun,Yuanlie Lin. A novel method for protein secondary structure prediction using dual-layer SVM and profiles. Protein, 54(4):738-743,2004.
[15]V. A. Simossis, J. Heringa, Integrating protein secondary structure prediction and multiple sequence alignment, Current Protein and Peptide Science, 5(4) :249–266,2004.
[16]Sujun Hua, Zhirong Sun.. A Novel Method of Protein Secondary Structure Prediction with High Segment Overlap Measure:Support Vector Machine Approach. J. Mol. Biol. 308 (2):397-407,2001.
[17]Bingru Yang, Xin Li, Wei Song. Generalized Causal Inductive Reasoning Model Based on Generalized Causal Cellular Automata. In:Proc. of ICNN&B'05,375-378, 2005.
[18]D. T. Jones, Protein secondary structure prediction based on position specific scoring matrices, J. Mol. Biol.,292: 195–202,1999,.
[19]M. Ouali, R. D. King, Cascaded multiple classifiers for secondary structure prediction, Protein Sci., 9(6):1162-1176, 2000.
[20]D. Frishman, P. Argos, Seventy-five percent accuracy in protein secondary structure prediction, Proteins, 27(3) :329-335,1997.
[21]G. Pollastri, D. Przybylski, B. Rost, P. Baldi, Improving the predictionof protein secondary structure in three and eight classes using recurrent neural networks and profiles, Proteins, 47(2):228‑235,2002.
[22]C. Cole, J. D. Barber, G. J. Barton, The jpred 3 secondary structure prediction server, Nucleic Acids Res. 36,197-201, 2008.
[23]K. Karplus, R. Karplus, J. Draper, et al, Combining local-structure, fold recognition, and new fold methods for protein structure prediction, Proteins,53(6):491‑496,2003.
[24]D. Frishman, P. Argos, Seventy-five percent accuracy in protein secondary structure prediction, Proteins,27(3):329‑335,1997.
[25]Ouali M, King R. Cascaded multiple classifiers for secondary structure prediction. Protein Sci, 9(11):62‑76,2000.

Bankruptcy Prediction with Missing Data

Q. Yu[1], **Y. Miche**[1], **A. Lendasse**[1] **and E. Séverin**[2]
[1]Department of Information and Computer Science, Aalto University, Espoo, Finland
[2]Department GEA, Université Lille 1, Lille, France

Abstract— *Bankruptcy prediction have been widely studied as a binary classification problem using financial ratios methodologies. When calculating the ratios, it is common to confront missing data problem. Thus, this paper proposes a classification method Ensemble Nearest Neighbors (ENN) to solve it. ENN uses different nearest neighbors as ensemble classifiers, then make a linear combination of them. Instead of choosing k in original k-Nearest Neighbors algorithm, ENN provides weights to each classifier which makes the method more robust. Moreover, using a adapted distance metric, ENN can be used directly for incomplete data. In a word, ENN is a robust and a comparatively simple model while providing good performance with missing data. In the experiment section, four financial datasets which are publicly available are used to prove this conclusion.*

Keywords: Missing data, Ensemble model, Nearest Neighbors, Bankruptcy Prediction

1. Introduction

The business failure has been widely studied, trying to identify the various determinants that can affect the existence of firms. Especially due to the recent changes in the world economy and as more firms, large and small, seem to fail now more than ever. The prediction of the bankruptcy, is then of increasing importance.

In most of the studies, bankruptcy prediction is treated as a binary classification problem. The target (output) variable of the models is commonly a dichotomous variable where 'firm filed for bankruptcy' is set to 1 and 'firm remains solvent' is set to 0. The reference (input) variables are often financial ratios drawn from financial statements and include measures of profitability, liquidity, and leverage. The pioneer study using univariate statistic of financial ratios originated from Beaver (1966) [4] and Altman's work (1968) [5]. Using multivariate discriminate analysis to assess predictive power of ratio analysis, financial ratios methodologies are becoming indispensable tools for modeling, analysis and prediction. The other main steam is employing Artificial Intelligence (AI) methods, which have been applied to bankruptcy prediction problem from 1990âĂŹs, including decision tree [24], [25], fuzzy set theory [26], case-based reasoning [27], [28], genetic algorithm [29], support vector machine [30], several kinds of neural networks such as BPNN (back propagation trained neural network) [32], [31], [34], PNN (probabilistic neural networks) [33], SOM (self-organizing map) [35], [36].

However, when calculating the financial ratios, for example, from companies' annual reports, it is very common to encounter the problem of missing value [1]. Some classification methods choose to remove all the ratios (variables) and the observations (samples) which contain missing values to train the model. The drawback is that it loses data, especially when the quantity of the observations is originally small. Furthermore, the new observations with missing values are no longer predictable. On the other hand, a great number of methods have been already developed for solving the problem by filling this missing values (also named imputation), for example, Kriging [6] and several other Optimal Interpolation methods, such as Objective Analysis [7].

In this paper, a third approach is proposed: a classification model which is directly applied to datasets with missing values. In order to predict whether the target company is healthy or not, this method provides an Ensemble model of nearest-neighbors (ENN) aimed at solving the classification task. Since it is impossible to calculate a standard Euclidean distance in NN algorithm when the sample have missing data, this method uses a new distance metric to measure the closeness between incomplete samples[10]. Thus the observations with missing values can be used to train the model, and moreover, the incomplete new observations can also be predicted. Besides, how to choose the suitable 'k' is always an issue when using KNN methods. In this paper, an ensemble method [9] is used. Instead of choosing a specific k, different nearest neighbors are treated as several different classifiers. The ensemble strategy assigns different weights to each classifier and then a linear combination of the nearest neighbors is used as the global output of the ensembles. This method is robust as the ensemble of classifiers has a smaller variance than each single classifier and then will reach better prediction performances [23].

The following section introduces the ensemble concept in general and particular strategy used in this paper with k Nearest Neighbors. It is followed by a presentation of incomplete data problem and a feature-weighted distance metric measurement in Section 3. In Section 4, four data sets are performed using random interpolation of missing

[1]Missing data, or missing values, occur when no data value is stored for the variable in the current observation. If a input data has N observations (samples) with d dimensions (variables). Then, when we say a missing data in this data, it implies one missing point among the original $(N * d)$ points.

data and the Monte-Carlo cross test. Finally, Section 5 shows more discussion about the experiment results and some conclusion.

2. Ensemble Nearest neighbors (ENN)

In machine learning, ensemble methods use multiple models to obtain better predictive performance than could be obtained from any of the constituent models [1], [2], [3]. It is a supervised learning algorithm, because it can be trained and then used to make predictions. Empirically, ensembles tend to yield better results when there is a significant diversity among the models [11], [12]. Many ensemble methods, therefore, seek to promote diversity among the models they combine [13]. On the other hand, ensembles can be shown to have more flexibility in the functions they can represent. This flexibility can, in theory, enable them to over-fit the training data more than a single model would, but in practice, some ensemble techniques (for example bagging) tend to reduce problems related to over-fitting of the training data. In the following from this section, more details are presented about ensemble of different k nearest neighbors.

2.1 The classifiers used for ensembles

An effective way to improve a classification method's performance is to create ensembles of classifiers. Two elements are believed to be important in constructing an ensemble: (a) the performance of each individual classifier; and (b) diversity among the classifiers. Therefore, different k nearest neighbors are chosen to perform such tasks.

In the original k-NN algorithm, the main difficulty is how to choose k properly. To solve this problem, we use Nearest Neighbors with each specific k as classifiers in this method. Therefore, the method will choose or weight each k automatically, using the ensemble technique. Besides, k-NN algorithm itself is proved to be an efficient classifier [14]. Another advantage of using different k NN is that NN is a distance-based algorithm, which provides us the opportunity to solve missing data problem simultaneously with the corresponding distance metric.

This part is shown as step 1 in Fig 1.

2.2 Linear optimization strategy

There exist some common types of ensembles:

- Bayes optimal classifier. The Bayes Optimal Classifier is an optimal classification technique. It is an ensemble of all the hypotheses in the hypothesis space. On average, no other ensemble can outperform it, so it is the ideal ensemble [17]. Unfortunately, Bayes Optimal Classifier cannot be practically implemented for any but the most simple of problem.
- Bootstrap aggregating (bagging). It involves having each model in the ensemble vote with equal weight. In order to promote model variance, bagging trains each model in the ensemble using a randomly-drawn subset of the training set. As an example, the random forest algorithm combines random decision trees with bagging to achieve very high classification accuracy [18].
- Boosting. Boosting involves incrementally building an ensemble by training each new model instance to emphasize the training instances that previous models misclassified. In some cases, boosting has been shown to yield better accuracy than bagging, but it also tends to be more likely to over-fit the training data. By far, the most common implementation of Boosting is Adaboost, although some newer algorithms are reported to achieve better results.
- Linear combination. The reason for linear combination is that taking a weighted average over several models reduces the error by decreasing the variance around the target. On the other hand, linear ensemble makes the final model relatively simple and easier to interpret. Therefore, linear combination is used in this paper.

However, it is not easy to determine the weights in practice. In this paper, Non-Negative Least Square (NNLS) algorithm is used. According to Miche et al. [19], the advantage of NNLS is that it is efficient and fast. The square of the difference between the actual output and the weighted leave-one-outputs of the classifiers is minimized such that the weights ω_j are positive, as seen in Equation 1.

$$\min_{\omega_j} \|y - \sum_j y_{loo}^j \omega_j\|^2, \qquad s.t. \quad \omega_j \geq 0 \tag{1}$$

This linear combination using non-negative constraints of the weights also helps to avoid over-fitting. This part is illustrated on step 2 of Fig 1.

2.3 Leave-One-Out

LOO is an special case of k-fold cross-validation where k equals to the number of observations. In this paper, LOO method is used in the training set to get even better prediction and meantime, to reduce the risk of over-fitting. One problem with the LOO method is that it can get very time consuming, especially if the dataset tends to have a high number of observations. Fortunately, the PRESS (or PREdiction Sum of Squares) statistics provide a direct and exact formula for the calculation of the LOO error for linear models.

$$\epsilon^{\text{PRESS}} = \frac{y_i - \hat{y}_i \omega}{1 - \hat{y}_i \mathbf{P} \hat{y}_i^T}, \tag{2}$$

where $\mathbf{P}$ is defined as $\mathbf{P} = (\mathbf{Y}^T \hat{\mathbf{Y}})^{-1}$ and $\hat{\mathbf{Y}}$ is the estimated output matrix, and ω is the ensemble weight for each model. Read from [15] and [16] for details on this formula and implementations.

This LOO part can be found on step 3 of Fig 1.

3. Distance metric with missing data

In this section, a distance measurement is introduced using as much the existing data as possible.

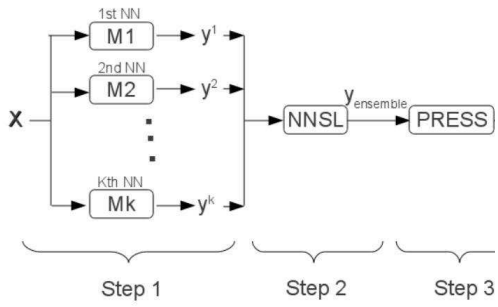


Fig. 1: Ensemble Nearest Neighbors (ENN) framewoFrk. X represents the Input data, $y^i, i = 1, ..., k$ represents the estimated output of each classifiers.

3.1 Incomplete data

We have already defined 'missing data' previously in footnote. Besides, the work in this paper assumes that the missing data is missing completely at random (MCAR) or missing at random (MAR) [20], meaning that the values of the data have no affect on whether the data is missing or not. MCAR occurs when the probability that a variable is missing is independent of the variable itself and any other external influence.

3.2 Measuring distance

Euclidean distance is normally used to measure closeness in NN series algorithms. But when confronting the incomplete data, some changes should be made to handle the missing data. Instead of making use of all the features between two observations, the adaptation of Euclidean distance is calculated by taking into account only the features with no missing values in both observations [10]. The distance is then normalized with respect to the number of features used to compute. The normalization is important to reduce the effect of the missing data. Otherwise, more features used, larger distance computed.

It may be more clear to use an example to explain. Suppose we have two observations $[2, ?, 4, 6, 8]$ and $[3, 5, 7, ?, 2]$. '?' represents a missing data. According to our new distance metric, the distances between these two observations is computed by using only the first, third, and fifth features. The second and fourth features are ignored because they contain missing values. Thus, the distance would be computed like this: $\sqrt{\frac{5}{3}((2-3)^2 + (4-7)^2 + (8-2)^2)}$. If there is no missing data in both observations, then the distance calculated is exactly the Euclidean distance in between.

4. Experiments with four datasets

In order to test the proposed method for bankruptcy prediction, four datasets are chosen in this paper. I would like to thank Dr du Jardin, Dr. Pietruszkiewicz, Dr. Atiya and Laura Kainulainen again for sharing these dataset which I know are expensive to obtain. The other reason for using these dataset is they have been used in some published articles which the results can be easily compared.

On the other hand, how to get a more general performance of the model remains to be a problematic issue. A common solution is to split the whole dataset into training, validating and testing sets, which is a good practice. In this paper, we only need to separate training and testing set because Leave-One-Out validation is used with the training set, i.e. the error we get from the trainning set is actually the LOO error. Furthermore, Monte-Carlo method is performed to split the data in order to reduce the effect of limited data size.

4.1 Monte-Carlo preprocessing

Monte-Carlo methods refer to various techniques. In this paper, Monte-Carlo methods are used to preprocessing the data, aiming to two tasks. Firstly, training set are drawn randomly about 75% of the whole data sets, the rest leaves for test set. Meantime, the proportion on the two class (healthy or bankruptcy) of both the training and testing set remain the same as the original one. Secondly, this Monte-Carlo preprocessing are repeated for many times for each dataset independently. Therefore, after these rounds of training and testing, a average test error is computed to represent the more general performance of the method.

4.2 Generating the missing date

There is no missing value originally in these four dataset. Therefore, missing data is artificially added in each datasets, in order to test the performance on incomplete data of the method. More precisely, the missing data is added one by one at randomly position till each observation has only one feature left. For example, if we have training set with N observations and d features ($N \times d$ data point totally), missing data is added till there is only N data points left (each sample has one variable). So that the model is trained and tested $N \times (d-1)$ times.

Moreover, in the following experiments, missing data is also added to the test set. The goal is to evaluate if the model trained with incomplete data can fits on the the incomplete new observations.

4.3 Pietruszkiewicz dataset

Wiesław Pietruszkiewicz has developed a data set of 240 cases of which 112 are bankrupted companies and 128 healthy. In total there are 120 companies, because the data comes from two years in a row. The possible bankruptcy occurred from two up to five years after the observations [38], [39]. The 29 variables consist of ratios of different financial variables. These variables are presented in Table 1.

Since this dataset is relatively small, Monte-Carlo cross test is used in order to present more general performance. In each round of Monte-Carlo test, the same size of samples (180 out of 240) are randomly chosen to train the model

Table 1: The variables used in the Pietruszkiewicz dataset

X1	cash/current liabilities
X2	cash/total assets
X3	current assets/current liabilities
X4	current assets/total assets
X5	working capital/total assets
X6	working capital/sales
X7	sales/inventory
X8	sales/receivables
X9	net profit/total assets
X10	net profit/current assets
X11	net profit/sales
X12	gross profit/sales
X13	net profit/liabilities
X14	net profit/equity
X15	net profit/(equity + long term liabilities)
X16	sales/receivables
X17	sales/current assets
X18	(365*receivables)/sales
X19	sales/total assets
X20	liabilities/total income
X21	current liabilities/total income
X22	receivables/liabilities
X23	net profit/sales
X24	liabilities/total assets
X25	liabilities/equity
X26	long term liabilities/equity
X27	current liabilities/equity
X28	EBIT (Earnings Before Interests and Taxes)/total assets
X29	current assets/sales

and the rest of samples are used to be test, keep the same proportion of each classes. In this experiment, 1000 times of Monte-Carlo tests are performed, and the average accuracy is calculated and shown in Fig 2.

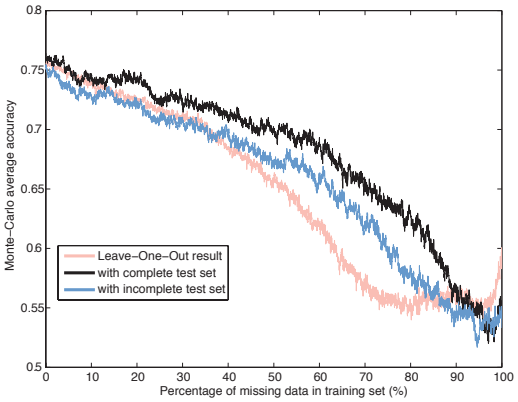


Fig. 2: Results of Pietruszkiewicz dataset

In general, this data set is very challenging to predict. It has been used and tested in many papers, for example, in Kainulainen's work [23], the best accuracy it achieved is around 75% without any missing data. Fig 2 shows the Leave-One-Out accuracy in red (the lowest curve), and test accuracies are presented in blue (curve in middle) and black (curve on the top). Incomplete test set contains one third of the missing values for each observitions. Both complete and incomplete test set performances start to decrease drastically for more than 60% missing data in the training set.

4.4 Philippe du Jardin datasets

The second and third data sets are somewhat similar. They were both used in the thesis of Philippe du Jardin. The dataset of 2002 comprises companies that have accounting data from the year 2002 and net equity data from the year 2001. The bankruptcy decisions, or more accurately, decisions of reorganization or liquidation, are from the year 2003. The dataset of 2003 was constructed similarly. In both datasets, the proportion of healthy and bankrupted companies is 50:50. In total, there were 500 and 520 samples, respectively. The companies are all from the trade sector and they have a similar structure, juridically and from the point of view of the assets. In addition, the healthy companies were still running in 2005, and had activities at least during four years. The ages of the companies were also considered, in order to obtain a good partition of companies of different ages [40]. Both of the datasets have 41 variables. The labels of the variables are presented in Table 2.

Jardin dataset and the Pietruszkiewicz dataset are fairly similar in terms of the variables. Both of them use financial ratios. The ratios are not exactly the same in all the cases, but very similar.

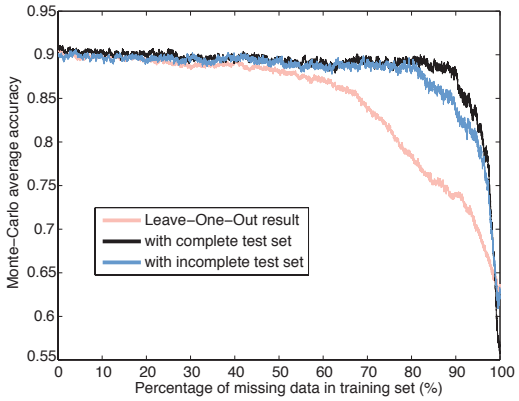


Fig. 3: Results of Philippe du Jardin datasets 2002

For the Jardin dataset, year 2003 is surely more difficult to predict than year 2002. There are some published results for this [40], [41]. Same as the previous dataset, in each Monte-Carlo round, about 75% of the observations are selected for training and the rest for testing, keep the same proportion of each classes in both training and testing set. Fig 3 shows the average classification results of 30 times Monte-Carlo processes. In general, the results remain on a high level (around 90% accuracy) and being relatively stable. More precisely, blue and black curve are interweaved together from 0% of missing data till about 80%, i.e. even the model built using only 20% of the training data, is still trustable.

Table 2: The variables used in the du Jardin datasets. EBITDA = Earnings Before Interest, Taxes, Depreciation and Amortization.

X1	Profit before Tax/Shareholders' Funds
X2	Net Income/Shareholders' Funds
X3	EBITDA/Total Assets
X4	EBITDA/Permanent Assets
X5	EBIT/Total Assets
X6	Net Income/Total Assets
X7	Value Added/Total Sales
X8	Total Sales/Shareholders' Funds
X9	EBIT/Total Sales
X10	Total Sales/Total Assets
X11	Gross Trading Profit/Total Sales
X12	Operating Cash Flow/Total Assets
X13	Operating Cash Flow/Total Sales
X14	Financial Expenses/Total Sales
X15	Labor Expenses/Total Sales
X16	Shareholders' Funds/Total Assets
X17	Total Debt/Shareholders' Funds
X18	Total Debt/Total Assets
X19	Net Operating Working Capital/Total Assets
X20	Long Term Debt/Total Assets
X21	Long Term Debt/Shareholders' Funds
X22	(Cash + Marketable Securities)/Total Assets
X23	Cash/Total Assets
X24	(Cash + Marketable Securiti es)/Total Sales
X25	Quick Ratio
X26	Cash/Current Liabilities
X27	Current Assets/Current Liabilities
X28	Quick Assets/Total Assets
X29	Current Liabilities/Total Assets
X30	Quick Assets/Total Assets
X31	EBITDA/Total Sales
X32	Financial Debt/Cash Flow
X33	Cash/Total Debt
X34	Cash/Total Sales
X35	Inventory/Total Sales
X36	Net Operating Working Capital/Total Sales
X37	Accounts Receivable/Total Sales
X38	Accounts Payable/Total Sales
X39	Current Assets/Total Sales
X40	Change in Equity Position
X41	Change in Other Debts

Moreover, there is no significant differences to predict a complete new observation or a observation with one third data missing.

Result from year 2003 is similar as year 2002. 30 times of Monte-Carlo process is done so far and shown in Fig 4. Since the size of Jardin data is larger, compared to Pietruszkiewicz dataset, it takes more time to compute in each round. More rounds of test will be done in order to further reduce the effect of randomness when adding missing data. Results will be updated later on.

4.5 Atiya dataset

The data set developed by Amir Atiya consists of 983 firms. 607 of them were solvent and 376 defaulted, but the prediction for the defaulted firms was performed at two or four instants before default. The observations of the defaulted firms come from a time period of 1 month to 36

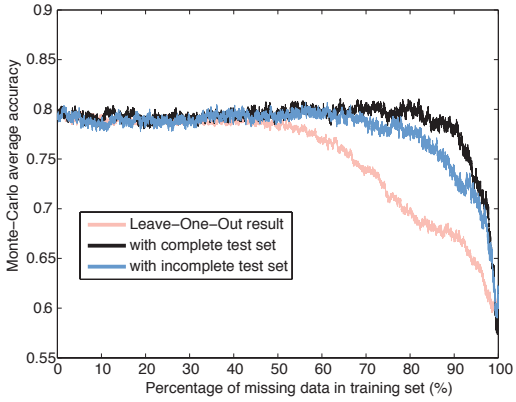


Fig. 4: Results of Philippe du Jardin datasets 2003

months before the bankruptcy, the median time being 13 months [37]. In total, there were 63 variables. The data was standardized to 0 mean and variance 1 before performing classification task. The values of the Atiya dataset are presented in Tables 3 and 4

Since the Atiya dataset is unbalanced with regards to the number of healthy companies and number of bankrupted companies, a different measure for mean accuracy is used. That measure is defined in Equation 3.

$$\frac{\frac{True\ positive}{Total\ positive} + \frac{True\ negative}{Total\ negative}}{2} \tag{3}$$

This Atiya dataset (983 observations and 63 variables) is relatively larger than previous three datasets. Thus, after each round of Monte-Carlo split, there are 737 samples (about one third) using for training. Missing data is added from 1 to 45694 $(737 \times (63-1))$ to the training set, i.e., the model have to be trained and tested 45694 times for each Monte-Carlo round. It is very time consuming. Therefore, 3 rounds is done so far, more rounds of experiments are still running and more results will be updated.

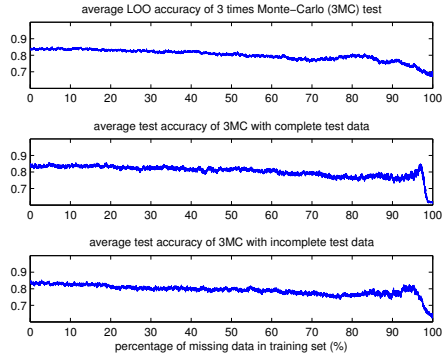


Fig. 5: Results of Atiya dataset

Fig 5 illustrates the results using three separate figures

Table 3: The variables used in the Atiya dataset, part 1. ROC=rate of change (usually over 4 year period), CFPS=cashflow per share, EPS=earning per share, GOI=gross operating income (i.e.before taxes, interest and other deductions), profit mgn=profit margin, TA=total assets, gross profit mgn=profit margin as related to GOI, EQ=shareholders equity (also called book value), NOI=net operating income (after taxes, etc), P/CF=price cashflow ratio, PE = price earnings ratio.

X1	cash/tot assets
X2	working capital/tot assets (TA)
X3	working capital/curr assets
X4	equity (EQ)/TA
X5	1-(long term debt/TA)
X6	rate of chg of cash flow per share (CFPS)
X7	rate of chg (ROC) of earnings per share (EPS)
X8	ROC(EPS from cont. operations)
X9	ROC(gross operating income GOI)
X10	ROC(net oper. Inc NOI)
X11	ROC(sales)
X12	ROC(gross profit margin)
X13	ROC(net profit margin)
X14	a measure of share price chg
X15	a measure of chg of gross oper mgn
X16	one year chg in net profit mgn
X17	ROC(TA)
X18	one year chg in EQ
X19	other ROC(CFPS) (other measure of chg)
X20	other ROC(EPS)
X21	other ROC(EPS cont oper)
X22	other ROC(GOI)
X23	other ROC(NOI)
X24	other ROC(sales)
X25	gross profit mgn
X26	net profit mgn
X27	a measure of dividend incr/decr
X28	cash flow (CF)/TA
X29	earnings/TA
X30	earnings cont oper/TA
X31	GOI/TA
X32	NOI/TA
X33	sales/TA
X34	PE ratio
X35	P/CF ratio
X36	price sales ratio
X37	price book value ratio
X38	return on assets ROA
X39	return on equity
X40	current ratio

(one curve each). The reason is because these three curves are interweaved together which is impossible to see clearly in White-Black print. The curve is not as smooth as previous ones because only three Monte-Carlo are used. However, the tendency is similar as previous results. The models built with up to at least 50% of missing data keep stable at a high level, and test with complete data or incomplete data (one third missing) doesn't make obvious differences.

Table 4: The variables used in the Atiya dataset, part 2. ROC=rate of change (usually over 4 year period), CFPS=cashflow per share, EPS=earning per share, GOI=gross operating income (i.e.before taxes, interest and other deductions), profit mgn=profit margin, TA=total assets, gross profit mgn=profit margin as related to GOI, EQ=shareholders equity (also called book value), NOI=net operating income (after taxes, etc), P/CF=price cashflow ratio.

X41	Quick ratio
X42	market capitalization/(long term debt LTD)
X43	relative strength indicator
X44	gross profit mgn
X45	net profit mgn
X46	one-year rel chg of CF
X47	one-year rel chg of GOI
X48	one-year rel chg og NOI
X49	4 yr ROC(CF)
X50	4 yr ROC(GOI)
X51	4 yr ROC(NOI)
X52	3 yr ROC(CF)
X53	3 yr ROC(GOI)
X54	3 yr ROC(NOI)
X55	TA
X56	sector default prob
X57	one year ROC(price)
X58	4 yr ROC(price)
X59	3 yr ROC(price)
X60	price
X61	a measure of ROC(price)
X62	volatility
X63	3 yr ROC(EQ)

5. Conclusions

In this paper, a new methodology to achieve classification for bankruptcy prediction with incomplete data is introduced. The approach ENN assembles k different Nearest Neighbor classifiers, and makes a linear combination of them. The most significant advantage is that ENN uses a modified Eulidean distance metric to solve the missing data problem while keeping the comparable performance.

In the experiments, Monte Carlo test is used in order to reduce variability of the performances casued by limited data size. Results on the four financial datasets illustrate that the performances of the proposed methodology are not deteriorating significantly with missing data from a percentage going from 0 to at least 50% of missing data in both the training and the testing data. The test results remain on the same level with both complete testing observations and incomplete testing ones (one third of the data are missing for each observations).

The results confirm the advantages of this method: being robust while providing good performance with missing data and a comparatively simple model.

References

[1] D. Opitz, R. Maclin, "Popular ensemble methods: An empirical study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.

[2] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, pp. 21–45, 2006.

[3] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, 1–39,2010.

[4] W.H. Beaver, " Financial ratios as predictors of failure," *Journal of Accounting Research*, vol. 4, pp. 71–111, 1966.

[5] E.I. Altman, " Financial ratios, discriminant analysisi and the prediction of corporation bankruptcy," *The Journal of Finance*, vol. 23, pp. 589–609, 1968.

[6] H.Wackernagel, "Multivariate Geostatistics - An introduction with applications," Springer, Berlin, 1995.

[7] L.S. Gandin, "Objective analysis of meteorological fields," *Israel Program for Scientific Translations*, Jerusalem, pp. 242, 1969.

[8] G.E. Batista, M.C. Monard, "A study of k-nearest neighbour as an imputation method, *Second International Conference on Hybrid Intelligent Systems*, vol. 87, pp 251–260, Santiago, Chile, 2002.

[9] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, pp. 21–45, 2006.

[10] G. Doquire, M. Verleysen, "Mutual information for feature selection with missing data," *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Bruges (Belgium), 27–29 April 2011, to appear.

[11] L. Kuncheva and C. Whitaker, "Measures of diversity in classifier ensembles," *Machine Learning*, vol. 51, pp. 181–207, 2003.

[12] P. Sollich and A. Krogh, "Learning with ensembles: How overfitting can be useful," *Advances in Neural Information Processing Systems*, vol. 8, pp. 190–196, 1996.

[13] G. Brown, J. Wyatt, R. Harris and X. Yao, "Diversity creation methods: a survey and categorisation," *Information Fusion*, vol. 61, pp. 5–20, 2005.

[14] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, 21–27, 1967.

[15] R. H. Myers, "Classical and Modern Regression with Applications," Duxbury, Pacific Grove, CA, 1990. USA

[16] G. Bontempi, M. Birattari, and H. Bersini, "Recursive lazy learning for modeling and control," *Euro- pean Conference on Machine learning*, pp. 292–303, 1998.

[17] T. M. Mitchell, *Machine Learning*, pp. 175, 1997.

[18] L. Breiman, "Bagging Predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.

[19] Y. Miche, E. Eirola, P. Bas, O. Simula, C. Jutten, A. Lendasse and M. Verleysen, "Ensemble modeling with a constrained linear system of LOO outputs," *ESANN2010: 18th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine learning*, bruges, Belgium, pp. 19–24, Apr 2010.

[20] R. J. A. Little and D. B. Rubin, "Statistical Analysis with Missing Data (second ed.)," Wiley, NJ, USA, 2002.

[21] http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

[22] http://www.pietruszkiewicz.com/datasets

[23] L. Kainulainen, Q. Yu, Y. Miche, E. Eirola, E. Séverin and A. Lendasse, " Ensembles of Locally Linear Models: Application to Bankruptcy Prediction," *In Proceedings of the 2010 International Conference on Data Mining*, pp. 280–286, July, 2010.

[24] H. Frydman, E.I. Altman and D. Kao, "Introducing recursive partitioning for financial classification: The case of financial distress, Journal of Finance," vol 40 , pp. 269–291, 1985.

[25] M.L. Marais, J. Patel and M. Wolfson, "The experimental design of classification models: An application of recursive partitioning and bootstrapping to commercial bank loan classifications," *Journal of Accounting Research*, vol. 22, pp. 87–114, 1984.

[26] J. Zimmermann, "Fuzzy set theory and its applications," *Kluwer Academic Publishers*, London, 1996.

[27] S.M. Bryant, "A case-based reasoning approach to bankruptcy prediction modeling," *International Journal of Intelligent Systems in Accounting, Finance and Management*, vol. 6, pp. 195–214, 1997.

[28] Park and I. Han, "A case-based reasoning with the feature weights derived by analytic hierarchy process for bankruptcy prediction," *Expert Systems with Applications*, vol. 23, pp. 255–264, 2002.

[29] K.S. Shin and Y.J. Lee, "A genetic algorithm application in bankruptcy prediction modeling," *Expert Systems with Applications*, vol. 23, pp. 321–322, 2002.

[30] J.H. Min and Y.C. Lee, "Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters", *Expert Systems with Applications* vol. 28, pp. 603–614, 2005.

[31] M. Lam, "Neural networks techniques for financial performance prediction: Integrating fundamental and technical analysis," *Decision Support Systems*, vol. 34, pp. 567–581, 2004.

[32] A.F. Atiya, "Bankruptcy prediction for credit risk using neural networks: A survey and new results," *IEEE Transactions on Neural Networks*, vol. 12, pp. 929–935, 2001.

[33] Z.R. Yang, M.B. Platt and H.D. Platt, "Probability neural network in bankruptcy prediction," *Journal of Business Research*, vol. 44, pp. 67–74, 1999.

[34] P. Swicegood and J.A. Clark, "Off-site monitoring systems for predicting bank underperformance: A comparison of neural networks, discriminant analysis and professional human judgment," *International Journal of Intelligent Systems in Accounting, Finance and Management*, vol. 10, pp. 169–186, 2001.

[35] S. Kaski, J. Sinkkonen and J. Peltonen, "Bankruptcy analysis with self-organizing maps in learning metrics," *IEEE Transaction on Neural Networks*, vol. 12, pp. 936–947, 2001.

[36] K. Lee, D. Booth and P. Alam, "A comparison of supervised and unsupervised neural networks in predicting bankruptcy of Korean firms," *Expert Systems with Applications*, vol. 29, pp. 1–16, 2005.

[37] A. F. Atiya, "Bankruptcy prediction for credit risk using neural networks: A survey and new results," *Neural Networks, IEEE Transactions*, vol. 12, pp. 929–935, Jul, 2001.

[38] W. Pietruszkiewicz, "Application of Discrete Predicting Structures in an Early Warning Expert System for Financial Distress," Ph.D. thesis, Fac- ulty of Computer Science and Information Technology, Szczecin University of Technology, Dec 2004.

[39] W. Pietruszkiewicz, "Dynamical systems and nonlinear kalman ïňĄlter- ing applied in classiïňĄcation," *In Proceedings of 2008 7th IEEE International Conference on Cybernetic Intelligent Systems*, pp. 263–268, 2008.

[40] P. D. Jardin, "Prévision de la défaillance et réseaux de neurones: léapport des méthodes numériques de sélection de variables," PhD thesis, Université de Nice-Sophia-Antipolis, 2007.

[41] D. Sovilj, A. Sorjamaa, Q. Yu, Y. Miche and E. Séverin, "OPELM and OPKNN in long-term prediction of time series using projected input data," *Neurocomputing*, vol. 73, pp. 1976–1986, June, 2010.

An EM-based Multi-Step Piecewise Surface Regression Learning Algorithm

Juan Luo and Alexander Brodsky

***Abstract*— A multi-step Expectation Maximization based (EM-based) algorithm is proposed to solve piecewise surface regression problem which has typical applications in market segmentation research, identification of consumer behavior patterns, weather patterns in meteorological research, and so on. The multiple steps involved are local regression on each data point of the training data set and a small set of its closest neighbors, clustering on the feature vector space formed from the local regression, regression learning for each individual surface, and classification to determine the boundaries for each individual surface. An EM-based iteration process is introduced in the regression learning phase to improve the learning outcome. The reassignment of cluster identifier for every data point in the training set is determined by predictive performance of each submodel. Cross validation technique is applied to the scenario in which the number of piecewise surfaces is not given in advance. A few clustering quality validity indexes such as Silhouette index and Davis-Bouldin index are adopted to estimate the number of piecewise surfaces as well. A set of experiments based on both artificial generated and benchmarks data source are conducted to compare the proposed algorithm and a few widely-used regression learning packages to show that the proposed algorithm outperforms those packages in terms of root mean squared errors (RMSE) of test data set.**

I. Introduction

The solution to any learning problem involves the reconstruction of an unknown function $f : X \to Y$ from a finite set S of sample of f (training set), possibly affected by noise. Different approaches are usually adopted when the range of Y only contains a reduced number of disjoint elements, typically without a specific ordering among them (classification problems) or when Y is an interval of the real axis with the usual topology (regression problems). The real world application areas can be the determination of market segments in a marketing research study, the identification of distinct spending patterns in studies of consumer behavior, the detection of different types (clusters) of documents in text mining or weather patterns in meteorological research.

Regression analysis attempts to build a model based on the relationship of several independent variables and a dependent variable [1]. Let $x_1, ..., x_n$, be independent variables, and y, be dependent variable, both range over the set of R. The latter is a random variable defined over the underlying distribution of sample tuples in $I_n = R \times R \times ... \times R$. Suppose the learning set contains m tuples. Let us denote such a tuple as $x_h = (x_{h1}, ..., x_{hn})$ for $h = 1, ..., m$. The collection of data, $c = (x_h, y_h)$ for $h = 1, \ldots, m$, represent the available training data to estimate the values of the random variable $y = f(x_h, \beta)+N$ for $h = 1, \ldots, m$, where β represents a set of coefficients and N is a random noise. We assume that N is distributed as a Gaussian with 0 mean and variance σ such that: $E(y) = E(f(x_h, \beta) + N) = E(f(x_h, \beta)) = f(x_h, \beta)$, where E is the expected value. The standard least squares method is used to find coefficients β of f that minimize σ.

Application can be found, which lie on the borderline between classification and regression; these occur when the input space X can be subdivided into disjoint regions X_i characterized by different behaviors of the function f to be reconstructed. One of the simplest situation of such kind is piecewise surface regression: in this case X is a polyhedron in the n-dimensional space R^n and $\{X_i\}_{i=1}^k$ is a polyhedral partition of X, i.e. $X_i \cap X_j = \emptyset$ for every $i, j = 1, \ldots, k, i \neq j$ and $\bigcup_{i=1}^k X_i = X$. The target of a piecewise surface regression problem is to reconstruct an unknown function $f^* : X \to R$ having a linear behavior in each region X_i

$$f^*(x) = f_i(x_j, \beta_i) \qquad \text{if} \quad x \in X_i \tag{1}$$

when only a training set D containing m samples (x_h, y_h), $h = 1, \ldots, m$, is available. The output y_h gives a noisy evaluation of $f(x_h)$, being $x_h \in X$; the region X_i to which x_h belongs is not given in advance. The parameters set $\beta_1, \beta_2, \ldots, \beta_i$ for $i = 1, 2, \ldots, k$, characterizes the function set f_i and their estimate is a target of the piecewise surface regression problem. The regions X_i are polyhedral, i.e., they are defined by a set of l_i linear inequalities, which can be written in the following form:

$$A_i \begin{pmatrix} 1 \\ x \end{pmatrix} \geq 0 \tag{2}$$

where A_i is a matrix with l_i rows and $n + 1$ columns and their estimate is another target of learning process for every $i = 1, 2, \ldots, k$. According to (1) and (2), the target of the learning problem is actually two-fold: to generate both the regions X_i and the parameter set β_i for the unknown function set f_i, utilizing the information contained in the training set.

There has been work on the learning of piecewise surface regression problem. The quality of a piecewise regression algorithm heavily depends on the accuracy of the partition of input space. In the Local Linear Map (LLM) of [2] and combinatorial regression learning of [3], only information about the input space is used for partition of the data. However, when data points can not be separated within the input space but a meaningful separation can still be

Juan Luo is with the Department of Computer Science, George Mason University, Fairfax, VA 22030, USA phone: 703-993-1531 email: jluo2@gmu.edu).

Alexander Brodsky is with the Department of Computer Science, George Mason University, Fairfax, VA 22030, USA (email: brodsky@gmu.edu).

obtained by considering the target variable together. Different approaches have been proposed to solve the problem of partitioning data by incorporating the target variable. Some of them focuses on solving problems in two dimensional spaces as in [4] and [5].

In [6], the data is clustered based on the local model parameters learned from a set of small size local data set. In [7], a connectionist model, i.e., a three layer neural network is constructed to learn the parameter set in the regression problem. While neural networks show high accuracy on training data set, they typically do not perform well on testing data set which is caused by over-fitting problem. An approach is outlined in [8] that uses hierarchical clustering to cluster data points into segments that represent the individual regimes of the piecewise function and perform standard linear regression on them.

An EM-based algorithm (EMPRR) has been proposed proposed in [9] to solve general piecewise surface regression model. It is developed based on Levenberg-Marquardt (LM) algorithm [10] and [11], an iterative technique which helps in locating the discrepancy between a given model and the corresponding data and has become a standard technique for nonlinear least-square problems. Since EMPRR is a nonlinear optimization technique, an absolute bound on the time complexity is not feasible since there is no way of knowing exactly how long it takes for the method to converge. At the same time, an initial guess need to be made for every unknown parameter in the piecewise surface regression model as inputs. This brings more uncertainty into the optimization process and increases the chance of getting trapped in a local minimum. Another EM-like piecewise linear regression algorithm has been proposed in [12] as well. It describes an EM-like piecewise linear regression algorithm that uses information about the target variable to determine a meaningful partitioning of the input space. The main goal of this approach is to incorporate information about the target variable in the prototype selection process of a piecewise regression approach. The drawback of this approach lies in the fact that it randomly assigns an initial cluster index for each data point in the data set. It makes the learning process hard and ineffective to converge and at the same time, the learning outcome is unpredictable.

Some algorithms which have been developed for learning decision trees are variations of the algorithms which employs a top-down, greedy search through the space of decision trees. The Classification and Regression Trees (CART) system [13] is a tree learning technique that assigns constant values at the leaves of the tree. Consequently it can fit piecewise constant data well but fit piecewise linear data with errors. The package M5P, which combines conventional decision trees with linear regression functions at the leaves, is developed by [14] and [15]. These model trees are similar to piecewise linear functions. The M5P will be run as part of our experiment as a comparison approach.

The contribution of this paper can be summarized as follows. First, we propose an EM-based multi-step approach (EMMPSR) for the general piecewise surface regression problem described in (1) and (2), no matter what dimension the input domain is and considering the target variable in the clustering process; Second, multiple steps involved are local regression, clustering, regression learning on each individual surface and classification to determine the boundaries of each surface. The clustering process is performed based on the feature vector space of input data set, instead of the data set itself. The feature vector for each data point is calculated by local regression on a small subset of data which are closest to that data point. The estimation of submodels in (2) is learned by robust regression learning which can effectively detect "outliers". The estimation of boundaries for each region in (1) is performed by a multi-category classification algorithm; Third, an EM-based iteration process is introduced in the regression learning phase to improve the learning outcome. The clustering process assigns a cluster index to each data point. However, the assignment may not be the correct and can be adjusted in the next iteration; Fourth, in the scenario that the number of regions is not known in advance, a few clustering quality validity indexes are calculated to detect the optimal number of surfaces contained by the input data set; Finally, a set of experiments are conducted to compare a few currently used regression packages and the proposed algorithm.

The paper is organized as following. The problem definition is given and related literatures are discussed in section I. The detailed algorithm is described in section II. Experiment setups on both synthetic and benchmark data are discussed in section III. Section IV concludes the paper and points out possible future work.

II. The EM-based Multi-Step Piecewise Surface Regression Algorithm

The Expectation Maximization (EM) algorithm [16] has been adapted in our algorithm. The input space is partitioned by applying a double-fold k-means clustering algorithm, incorporating the value of target variable. After the polyhedral regions have been identified, a multi-category SVM library [17] is called to calculate the boundary matrix A_i in (2) which represents a polyhedral region. For each polyhedral region, its surface regression model can be learned by robustfit [18]. Similar to EM algorithm, an iteration process is involved in our approach as well. First, the surface models are learned from the resulted clusters of clustering process. Then all data points in every cluster will be re-assigned to the local model which has the best predictive performance. The local model will be updated again based on the newly created clusters of polyhedral regions. The iteration process will be repeated until termination criterion has been reached. The algorithm is given in Algorithm 1.

A. Local Regression

As discussed in the introduction, the learning effect of piecewise regression problem depends on the accuracy of the partition of input space. Given a training set D, the intuitive way to do partitioning is classical K-means clustering [19].

Algorithm 1: The EM-based Multi-step Piecewise Surface Regression Algorithm

Input: Data set D with size m, number of clusters k
Output: Surface function model f_i and boundary matrix A_i for i = 1, ..., k

(Local regression) **foreach** *h = 1, ..., m* **do**
  1.1 Build the local dataset E_h containing the sample (x_h, y_h) and the pairs $(x, y) \in D$, together with the $e-1$ closest neighbors x to x_h.
  1.2 Perform a linear regression to obtain the feature vector v_h of a linear unit fitting the samples in E_h.

(Clustering) Perform clustering process in the feature vector space.
  2.1 Run regular k-means on feature vector space R_{n+1} with an assigned feature vector centroid set CV to subdivide the set of feature vectors v_h into k groups $U_i, i = 1, ..., k$.
  2.2 Build a new training set D' containing m pairs (x_h, i_h) being U_{i_h} the cluster including v_h
repeat
  (Regression) For every j = 1,..., k, run a linear regression on the samples $(x, y) \epsilon D$ with $x \epsilon X_i$. The parameter set β_i returned represents the i_{th} surface function f_i.
  Update cluster index of each data point, further the training set D', according to the minimal predictive error among surface models f_i for i = 1, ..., k.
until *Maximum number of iterations has been reached or no cluster index is reassigned*;
Multi-category classification on training set D' to compute the boundary matrix A_i for every surface X_i.

However, k-means clustering results are sensitive to the initial centroid which are randomly picked, when clusters are of differing sizes, densities or non-globular shapes. Instead of clustering on training set, we proposes a different way of clustering, i.e. clustering on the feature (parameter) vector space of training set. The feature vector for each data point is learned by local linear regressor based on small subset of the whole training set D. It is observed that points close to one another are more likely to belong to the same region X_i than those are not. For each sample (x_h, y_h), with h = 1, ..., m, a local data set E_h is built to contain (x_h, y_h) and its $e-1$ nearest neighbors $(\hat{x}, \hat{y})$ that satisfy

$$\|x(h) - \hat{x}\|^2 \leq \|x(h) - \tilde{x}\|^2 \qquad \forall (\tilde{x}, \tilde{y}) \in D \quad \setminus \quad E_h, \quad (3)$$

The distance between points is calculated by $\|x_h - x\|$ where $\|.\|$ is the Euclidean norm. Note that each E_h can be labeled by the point (x_h, y_h). This way a bijective map between data points and local data sets is formed. Most sets E_h contain data points belonging to the same region X_i, while the rest, called mixed, include data points from different regions X_j. The local regression step is trying to obtain a first estimate of the parameter set β_i set which characterize the functional set f_i. Local linear regression is run on small subsets of the whole training set D based on the fact that points x_h which are close to one another are more possible to belong to the same region X_j than those not close. The feature vector v_h learned from pure local data set E_h is a good estimate of parameter β_i which represents the region function f_i, while the feature vectors learned from mixed local set lead to the wrong estimate of β_i so their number need to be kept at the lowest possible level.

The number of mixed local data set depends both on the sampling schedule of input space and choice of parameter e. As to the sampling schedule, an implicit assumption for good results from our algorithm is that the sampling is fair, i.e., the data points drawn are not all concentrated around the boundary of the sets X_i. The parameter e should be chosen well in order to obtain non-overlapping clusters of feature spaces and minimize, at the same time, the number of outliers. If the parameter e is low, the ratio between the number of mixed and non-mixed local data sets is low. However, when the noise level is not negligible, a low e produces poor estimates of the feature vectors, i.e. estimates with high variance, thus preventing a good partitioning of the feature vectors. So the value of e can not be assigned too low. On the other hand, if e is too high, a large percent of mixed local data sets (further outliers in the feature space) will be generated. In the extreme case is that when e is equal to the number of sample data set D, all local sets built are mixed and every feature vector collapse in a single hyperplane fitting all the data. In order to have a well-defined clusters, the value of the parameter e need to be tuned in experiments with cross-validation techniques.

The least-squared estimation can be used to compute the feature (parameter) vector of every local data set E_h which contains data points $(x_h^1, y_h^1), (x_h^2, y_h^2), ..., (x_h^e, y_h^e)$. We can define ϕ_h and ψ_h as

$$\phi_h = \begin{bmatrix} x_h^1 & x_h^2 & ... & x_h^e \\ 1 & 1 & ... & 1 \end{bmatrix}', \quad \psi_h = \begin{bmatrix} y_h^1 & y_h^2 & ... & y_h^e \end{bmatrix}' \quad (4)$$

The $'$ is the transpose operator for matrix. The feature vector v_h can be computed by the formula

$$v_h = (\phi_h' \phi_h)^{-1} \phi_h' \psi_h \quad (5)$$

Another bijective map can be formed between feature vectors V computed for each data point and each local data set. Given the bijective map between data points and local data sets, a new bijective map between each data point and each feature vectors is formed as well.

B. Clustering

After the feature vector space has been generated, the next step of the algorithm is to cluster the feature vectors into k disjoint subsets U_i. Principally, any clustering algorithm can be used but the performance of classical clustering algorithms like k-means is often spoiled by poor initialization of centroid which are randomly picked, when clusters are of differing

sizes, densities or non-globular shapes [19]. In our case, we propose a two fold k-means clustering algorithm which can decrease the misclassification rate of k-means and at the same time, the computational efficiency of K-means will still be kept. The clustering process is described in Algorithm 2.

Algorithm 2: Two-fold k-means clustering algorithm

Input: data set D with size m, feature vector set V with size m, number of clusters k

Output: Feature vector set V with cluster index assigned for every feature vector

1 Do regular k-means clustering on the data points with randomly picked initial centroid
2 For each cluster $X_1, X_2, ..., X_k$ returned by 1, calculate its mean $\bar{X_1}, \bar{X_2}, ..., \bar{X_k}$
3 For each cluster mean $\bar{X_i}$, $i = 1, ..., k$, among data points in cluster X_i, find the data point which is most close to $\bar{X_i}$ and save it to the centroid set C with size k
4 For each $c_i \in$ centroid set C, i=1,...,k, find the corresponding feature vector cv_i in the feature vector set V having the same index as centroid in the data set to form a feature vector centroid set CV.
5 Run k-means clustering on the feature vector set V with initially assigned centroid set CV to subdivide feature vector set into k groups U_i for $i = 1, ..., k$.

The difference between Algorithm 2 and the classical k-means is the initial picking of centroid. In Algorithm 2, the k-means clustering is run the first time to estimate the centroid of clusters of data point. Due to the bijective mapping between each data point and each feature vector, a corresponding feature vector can be found for each centroid resulted from the first step. This centroid set will be a much better initial input for the second k-means clustering which will be run on the feature vector space, compared to the randomly picked centers among feature vectors. Finally, by using the bijective maps between feature vectors and data points, the original data can now be classified. In fact, each data point (x_h, y_h), h=1,...,m is assigned a cluster index i_h being U_{i_h} the cluster including v_h. A new data set D' is formed as m pairs (x_h, i_h), h=1,...,m.

C. EM-based iteration process

The EM-based iteration process consists of two main steps: First, surface regression models f_i, i = 1,..., k for each region are determined according to the cluster assignment i_h of each data point (x_h, y_h), h=1,...,m; Second, each data point is re-assigned to one of the clusters X_i, i=1,...,k where the predictive error of the corresponding regression model is minimal. In our algorithm, each sample data point (x_h, y_h), h=1,...,m is assigned to only one cluster X_i, i=1,...,k. The assignment can be defined as a mapping:

$$CI(h) = i, \qquad \text{with} \quad 1 \leq h \leq m, 1 \leq i \leq k \tag{6}$$

which assigns the h^{th} data point to the i^{th} cluster. Each surface model is represented by function f_i, i = 1,..., k. The surface regression model of the i^{th} cluster is trained on all data points that are assigned by the mapping CI to the i^{th} cluster:

$$f_i(x_h) = y_i \quad \text{where} \quad CI(h) = i \tag{7}$$

1) Estimation of sub-models: To learn the surface regression models f_i, i = 1,..., k, least squares can accomplish this task. However, one of the main drawbacks of least squares lies in the sensitivity of the method to outliers [18] that may be present due to classification errors. The robust regression techniques are less sensitive to outliers than least squares, especially when the number of outliers is a small fraction of the data points [18]. The estimation of each sub-model is solved by the robust regression learning.

2) Cluster index reassignment: The second main step of EM-based iteration process is to update cluster index of each data point. The mapping defined in Equation (7) is updated according to the minimum predictive error among all submodels f_i, i = 1,..., k

$$CI(h) = argmin_{i=1,...,k} |y_h - f_i(x_h)| \tag{8}$$

If any data point which is misclassified at the beginning, it is possible that its cluster index will be re-assigned by Equation (8). The reassignment process changes region in terms of data point, furthermore, the estimation of the sub-models will be re-estimated as well. The iteration process is repeated until no more cluster reassignment occurs or the maximum number of iterations has been reached.

D. Estimation of boundary matrices

After the local model has been identified, the next step is to obtain an approximation of the unknown polyhedral regions which are specified by a set of matrices A_i, i = 1,..., k in Equation (2). The matrices is solved by multi-category classification technique derived from the support vector machine [17].

E. Detection of the number of regions

So far we assume that the number of clusters for data set is known in advance. However for some real data sets, this is not known a priori and, in fact, there might be no definite or unique answer as to what value k should take. In other words, k is a nuisance parameter of the clustering model. Numerous techniques can be applied to determine this k value. Among them, cross validation [20] and a few clustering validity indexes (silhouette index [21], Davis-Bouldin index [22], Calinski-Harabasz index [23] and Dunn index [24]) are computed and compared.

1) V-fold cross validation: The v-fold cross-validation algorithm is applied to clustering. The general idea of this method is to divide the overall sample into a number of v folds. The same type of analysis is then successively applied to the observations belonging to the v-1 folds (training sample), and the results of the analysis are applied to sample v (the sample or fold that was not used to estimate the parameters, i.e, the testing sample) to compute some index of predictive validity. The results for the v replications are

averaged to yield a single measure of the stability of the respective model, i.e., the validity of the model for predicting new observations. We can apply the v-fold cross-validation method to a range of numbers of clusters in k-means, observe the resulting average distance of the observations (in the testing samples) from their cluster centers.

2) Clustering quality validity indexes: Four different indexes are calculated to estimate the optimal number of clusters in benchmark data sets. The Silhouette index calculates the silhouette width for each sample, average silhouette width for each cluster and overall average silhouette width for a total data set. It uses average dissimilarities between points to identify the structure of the data and highlights possible clusters. It is suitable for estimating the first choice or the best partition. The value range of Silhouette index is between [-1, 1]. If silhouette value is close to 1, it means that sample data set is well-clustered and it was assigned to a very appropriate set of clusters. If silhouette value is close to -1, it means that sample data set is misclassified and is merely somewhere in between the clusters. The Davis-Bouldin index is a function of the ratio of the sum of within-cluster scatter to between-cluster separation. The ratio is small if the clusters are compact and far from each other. Consequently, Davis-Bouldin index will have a small value for a good clustering. Calinski-Harabasz index is the pseudo F statistic calculating the quotient between the intracluster average squared distance and intercluster average squared distance. The higher the Calinski-Harabasz index, the better the clustering quality. Dunn's index is based on geometrical considerations for hard clustering. This index is designed to identify sets of clusters that are compact and well separated. The main goal of this measure is to maximise the intercluster distances and minimize the intracluster distances. Therefore, the number of clusters that maximize the Dunn's index is taken as the optimal number of clusters.

III. Experiments

To evaluate our EM-based multi-step piecewise surface regression algorithm EMMPSR, we generate synthetic high-dimensional data which is piecewise-defined. We compare the performance of EMMPSR with those of M5P (weka.classifier.trees) [25], classregtree (matlab statistical toolbox) [26], and MultilayerPerceptron (three layer neural network) (weka.classifier.functions) [25] on a set of experimental data set. These three packages are currently wide-used regression learning tools. The data set includes three set of synthetic data and four benchmark data set.

A. Synthetic data sets

Three data sets are generated using four different piecewise models. Each model has linear boundaries between regions and linear functions within each region. Model 1 and model 2 each has three regions and two independent variables. Model 3 has five regions and nine independent variables with linear boundaries and linear functions as well. Data in each model are generated with additive Gaussian noise with zero mean and 0.1 variance. We generated 300 sample points for model 1, 900 data points for model 2 and 1500 data points for model 3. The second data set is generated from the following piecewise functions:

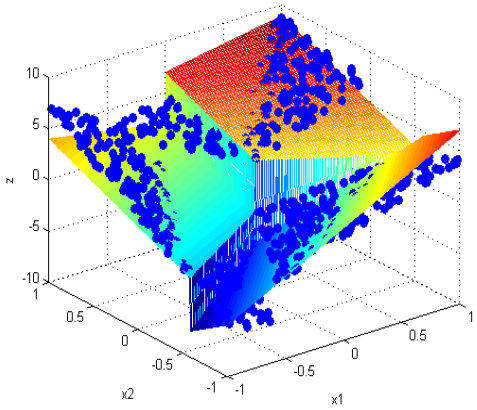


Fig. 1. Synthetic Data Set Generated in Model 2

$$f(x_1,x_2)=\begin{cases}3+4x_1+2x_2\\ \text{if } 0.5x_1+0.29x_2\geq 0 \; and \; x_2\geq 0\\ -5-6x_1+6x_2\\ \text{if } 0.5x_1+0.29x_2<0 \; and \; 0.5x_1-0.29x_2<0\\ -2+4x_1-2x_2\\ \text{if } 0.5x_1-0.29x_2\geq 0 \; and \; x_2<0\end{cases} \tag{9}$$

The target function is depicted in Figure 1. Total 900 samples are drawn uniformly from $I_2 = [-1,1]\times[-1,1]$ and y is determined as $y = f^*(x_1,x_2)+\varepsilon$, where $\varepsilon \sim N(0,0.1)$. In this setting, the target value need to combined with independent variables to determine the appropriate cluster prototypes.

The following function estimate is yielded by the EMMPSR algorithm:

$$f(x_1,x_2)=\begin{cases}3.0067+3.9940x_1+1.9977x_2\\ if\ 0.5x_1+0.32x_2\geq 0.005 and x_2\geq 0\\ -5.0217-6.0201x_1+6.0056x_2\\ if\ 0.5x_1+0.32x_2<0.005 and\\ 0.5x_1-0.31x_2<0.01\\ -2.0035+3.9793x_1-2.0330x_2\\ if\ 0.5x_1-0.31x_2\geq 0.01 and x_2<0\end{cases} \tag{10}$$

As noted, the generated model is a good approximation of the unknown function to learn in Equation (9). Five-fold cross validation is adopted to evaluate the learning performance by randomly dividing the data set into 5 equal parts. Each part is held out in turn and the remaining four is trained for the learning method. The root mean squared error (RMSE) [27] will be calculated on the unseen data. The results are summarized in Table I.

Another matrix to be compared among different methods is average number of rules generated by each model for a data set. In EMMPSR it is the number of regions, while in

TABLE I
RMSE VALUES FOR PERFORMANCE COMPARISON EXPERIMENTS ON SYNTHETIC DATA SETS

Model	M5P	MultilayerPerceptron	Classregtree	EMMPSR
Model1	1.0925	3.0657	2.8899	**0.3759**
Model2	0.7599	1.8773	0.4995	**0.2538**
Model3	37.6910	47.8030	33.3755	**30.8755**

M5P and Classregtree it is the number of rules generated during the process of building the tree. EMMPSR only uses a fraction of the rules that are generated by M5P and Classregtree. It is obvious that EMMPSR outperforms other methods as to RMSE as well.

B. Benchmark Data Set

Benchmark data sets are obtained from the Repository of Regression Problems at [28]. This repository is actually a collection of data from other sources, however we still choose it because the data sets have been preprocessed to meet our specifications – nominal attributes and samples with missing attributes have been removed. Five-fold cross validation is adopted to evaluate the learning performance as well.

Auto MPG Data Set: The task of this data set is to predict the fuel consumption in miles per gallon (MPG) of different cars. Five attributes of the original data set are used as input dimensions acceleration', displacement', horsepower', model-year, and weight. The data set consists of 398 instances. Six instances with missing values are ignored within the experiments.

Delta Ailerons Data Set: This data set is also obtained from the task of controlling the ailerons of an F16 aircraft, although the target variable and attributes are different from the ailerons domain. The target variable here is a variation instead of an absolute value, and there is some pre-selection of the attributes. 7129 cases with 6 continuous attributes.

California Housing: This data set contains information on block groups in California from the 1990 Census. The target variable is median house value. Independent attributes are median income, housing median age, total rooms, total bedrooms, population, households, latitude, and longitude. 20640 cases with 8 continuous attributes.

Stock Data Set: Daily stock prices from January 1988 through October 1991, for ten aerospace companies. 950 cases with 10 continuous attributes.

The number of clusters for the real data set is not known in advance so first the v-fold cross-validation algorithm described in section II.D.1 is adopted to determine the number of piecewise surfaces which are involved in the piecewise regression problem. The optimal number of clusters in Stock Data Set is calculated to describe the determination process. From Figure 2, it is observed that as the number of clusters increases, the sum of squared errors within the test set goes down fast until the number of cluster is equal to 7. Then the sum of squared errors goes down very slightly and gets stable when the number of clusters is equal to 8, 9 or even more. Consequently we can set the tentative number

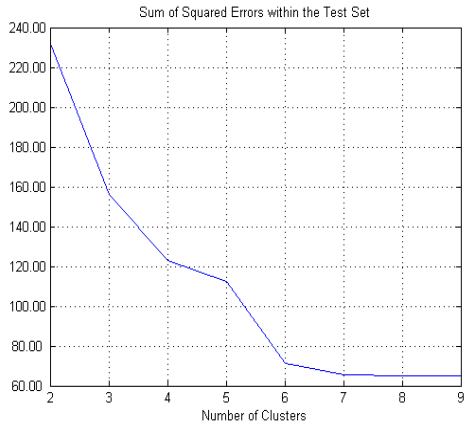


Fig. 2. Sum of squared errors within the test data set vs. number of clusters

of clusters to value either 8 or 9. However, the v-fold cross-validation is very time consuming due to the fact that the EMMPSR algorithm is involved in each run of the v-fold cross-validation process.

The clustering validity indexes are calculated as well to determine the optimal number of clusters in the Stock data set. Each index value is plotted in Figure 3 against the number of clusters. The star which represents the optimal number of clusters in the figure is circled with a small rectangle box. For Silhouette index, when the number of cluster is equal to 8, its value reaches the peak point, 0.52. Davis-Boudin index and Dunn index both display the optimal number of clusters as 8. The Calinski-Harabasz index shows that the optimal number is 9. We observe very similar result to what is observed in the cross validation.

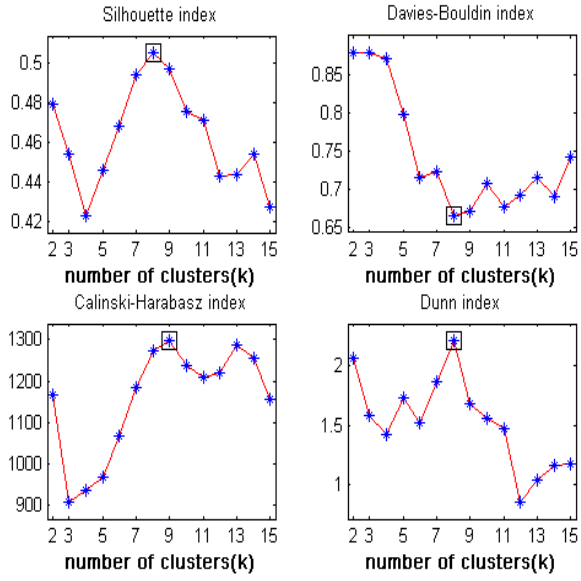


Fig. 3. Clustering Quality Validity Index vs. The Number of clusters

After we determine the optimal number of clusters for ev-

ery benchmark data set, the learning outcome is summarized in Table II. Five-fold cross validation is adopted to evaluate the learning performance of different packages / algorithms as well.

TABLE II
RMSE VALUES FOR PERFORMANCE COMPARISON EXPERIMENTS ON BENCHMARK DATA SETS

Model	M5P	Multi	Classregtree	EMMPSR
Auto	3.8419	4.6238	4.0455	**2.5332**
Delta Ailerons	0.0002	0.0002	0.0003	**0.0002**
California Housing	0.4838	0.6022	0.5998	**0.1617**
Stock	1.0151	1.3441	0.9746	**0.4876**

It is observed from the Table II that the EMMPSR algorithm can mostly achieve the best learning outcome among the four algorithms and at the same time, has the simplest format of representation for the piecewise regression problem.

IV. Conclusion and Future Work

A EM-based multi-step piecewise surface regression learning algorithm is proposed in this paper. The multiple steps involved are local regression, clustering, regression learning and classification for each surface. An EM-based iteration process is introduced for each run of regression learning phase. A set of experiments are compared to show in most cases the EMMPSR algorithm outperforms other popular packages used for classification and regression. Future research topic will be the selection of main features which will used for regression, and how to apply the EMMPSR algorithm to more general form of piecewise regression learning.

References

[1] Draper, N. and Smith, H., Applied Regression Analysis Wiley Series in Probability and Statistics, 1998

[2] Ritter, H, Learning with the self-organizing map, Artificial Neural Networks, pp. 379–384, 1991

[3] Luo, Juan and Brodsky, Alexander, An Optimal Regression Algorithm for Piecewise Functions Expressed as Object-Oriented Programs, the Ninth International Conference on Machine Learning and Applications, pp. 937–942, 2010

[4] Cherkassky, V and Lari-Najafi, H., Constrained topological mapping from non-parametric regression analysis, Neural Networks, vol.4, pp. 27–40, 1991

[5] Luo, Juan and Brodsky, Alexander, A Heaviside-based Regression of Piecewise Functions Expressed as Object-Oriented Programs, the Third International Conference on Machine Learning and Computing, vol. 1, pp. 296–301, 2011

[6] Hathaway, R.J. and Bezdek, J.C., Switching regression models and fuzzy clustering, IEEE Transactions on Fuzzy Systems, vol. 3, no. 1, pp. 195–204, 1993

[7] Ferrari-Trecate, Giancarlo and Muselli, Marco, A New Learning Method for Piecewise Linear Regression, Artificial Neural Networks ICANN, Lecture Notes in Computer Science, Springer Berlin Heidelberg, vol.2415, pp.135–135, 2002

[8] McGee, Carleton, Piecewise regression, Journal of the Society for Industrial and Applied Mathematics, vol.11(2), pp.431–441, 1963

[9] Manimozhiyan Arumugam and Stephen Scott, EMPRR: A High-Dimensional EM-Based Piecewise Regression Algorithm, *The 2004 International Conference on Machine Learning and Applications*, Louisville, Kentucky, pp. 264–271, 2004

[10] Levenberg, K., A method for the solution of certain non-linear problems in least squares, *Quarterly Journal of Applied Mathmatics*, vol. II, no. 2, pp. 164–168, 1944

[11] Donald W. Marquardt, An Algorithm for Least-Squares Estimation of Nonlinear Parameters, *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963

[12] Nusser, Sebastian, Otte, Clemens and Hauptmann, Werner, An EM-Based Piecewise Linear Regression Algorithm, *Proceedings of the 3rd international workshop on Hybrid Artificial Intelligence Systems*,Burgos, Spain, pp. 466–474, 2008

[13] Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J., Classification and Regression Trees, Wadsworth Publishing Company, 1984

[14] Quinlan, J.R., Learning with continuous classes, Proceedings of the Second Australian Conference on Aritificial Intelligance, pp. 343–348, 1992

[15] Wang, Y. and Witten, I.H., Inducing Model Trees for Continuous Classes, In Proc. of the 9th European Conference on Machine Learning Poster Papers, pp. 128–137, 1997

[16] Dempster, P., Laird N. and Rubin D., Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, Series B, pp. 1–38, 1977

[17] Chang, C. and Lin, C., LIBSVM: a library for support vector machines, http://www.csie.ntu.edu.tw/˜cjlin/libsvm, 2001

[18] Huber, P. and Ronchetti, E., Robust statistics, Wiley, New York, NY, U.S.A., 1981

[19] MacKay, D., An Example Inference Task: Clustering, *Information Theory, Inference and Learning Algorithms*, ch.20, num.284, Cambridge University Press, 2003

[20] Hill, T. and Lewicki, P., STATISTICS Methods and Applications. Ch.10, pp. 122–123, StatSoft, Tulsa, OK, 2007

[21] Rousseeuw, P.J., Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics*,vol. 20, pp. 53–65, 1987

[22] Davies, D.L. and Bouldin, D.W., A cluster separation measure, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 1(4), pp. 224–227, 1979

[23] Calinski, R.B. and Harabasz, J., A dendrite method for cluster analysis, *Communications in Statistics*, vol. 3, pp. 1–27, 1974

[24] Dunn, J.C., Well separated clusters and optimal fuzzy partitions, *Cybernetics and Systems: An International Journal*, vol. 4, Issue 1, pp. 95–104, 1974

[25] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann and Ian Witten, The WEKA Data Mining Software: An Update, vol.22, *SIGKDD Explorations*, 2009

[26] MATLAB, version 7.10.0 (R2010a), The MathWorks Inc., Natick, Massachusetts, 2010

[27] Alpaydin, E., Introduction to Machine Learning, MIT Press, 2004

[28] LIACC, http://www.liacc.up.pt/˜ltorgo/Regression/, 2006

SESSION

EXPLORATIVE DATA MINING, DATA PREPROCESSING, FEATURE SELECTION

Chair(s)

Nikolaos Kourentzes
Gary M. Weiss
Robert Stahlbock

A Computerized Feature Reduction Using Principal Component Analysis for Accident Duration Forecasting on Freeway

Ying Lee

***Abstract*—This study creates an Artificial Neural Network-based model and provides a forecast of accident duration at the accident notification. With this model, the estimated duration time can be provided by plugging in relevant traffic data as soon as an accident is notified. To reduce data feature, principal component analysis can decrease the number of model inputs and preserves the relevant traffic characteristics with fewer inputs. This study shows proposed model is feasible ones in the Intelligent Transportation Systems (ITS) context.**

***Index Terms*—Sequential forecast; Freeway accident duration analysis; Artificial neural networks; Principal component analysis**

I. INTRODUCTION

This study provides a forecast of the duration of an accident at the time of its notification. An Artificial Neural Network based model is deployed to input the relevant traffic data during the accident. The traffic data feature was reduced by the Principal Component Analysis (PCA) before inputting to the neural network. Figure 1 illustrates the time points when this model is employed to perform forecast in accident duration. When an accident is being notified for the first time, Model provides a preliminary forecast by using the traffic data which represent the traffic situation right before the accident.

II. LITERATURE REVIEW

Accident duration is often a major component of providing real-time traffic information. Usually, the incident duration varies with the traffic conditions. In most studies, however, the forecasted incident duration is not periodically updated according to the traffic conditions at the time point of forecast. The accident duration is usually defined as the time between accident occurrence and roadway clearance. This duration can be divided into three parts, namely, reporting time (the time between accident occurrence and accident notification), response time (the time between accident notification and rescuer arrival) and clearance time (the time between rescuer arrival and accident road clearance). Garib et al. [1] employed the multiple regression analysis to predict the clearance time of the accident. Nam and Mannering [2] applies hazard-based analysis to build three accident duration models from the three moments, i.e. the accident occurrence, the accident notification and the rescue people arrival. Wei and Lee [3] developed the accident duration model by the data fusion techniques. This model provides a forecasted duration at the accident notification moment.

In this study, data fusion is concerned with the problem of combining traffic data from multiple sensors in order to make inferences about traffic condition on a roadway of interest. Relevant techniques are needed to extract information from an individual database as well as to merge the data collected from different databases. The multi-source dataset is often composed of different statistical units or levels since it is normally gathered from different administrative sources. In developing our data fusion model, the artificial neural network (ANN) has been chosen as the key technique. As one of the most prominent approaches widely used for solving complex problems [3, 4, 5], ANNs have recently been gaining popularity for transportation studies.

Many studies demonstrate that ANNs have the potential to accurately predict freeway traffic conditions [3, 5, 6, 7], vehicle classification [8], incident detection [9] and civil engineering [10, 11].

In many research, high-dimensional data are involved, because large feature vectors are generated to be able to describe complex objects and to distinguish them. But large feature vectors may cause some disadvantages to the model, such as more time in model training and more noises in modeling. To avoid these problems, the feature vectors must be reduced. Principal components analysis (PCA) is a technique for forming new variables which are linear composites of the original variable. The maximum number of new variables that can be formed is equal to the number of original variables, and the new variables are uncorrelated among themselves [12]. PCA is used widely in different study field, such as medical science [13] and images processing [14].

This paper was derived from a research project sponsored by the National Science Council, Taiwan under the contract of NSC99-2628-E-451-001. I gratefully acknowledge the data providers for this research, including Taiwan Area National Freeway Bureau and Police Radio Station.

Ying Lee is Assistant Professor of the Department of Hospitality Management at the Ming Dao University, holds a Ph.D. degree from the National Cheng Kung University (NCKU), Tainan, Taiwan. (phone: +886-4-8876660 ext. 7829; fax: +886-4-88879035; e-mail: yinglee1017@gmail.com; No. 369, Wen-hua Rd., Pi-tou Township, Chang-Hua County, Taiwan ROC, 52345)

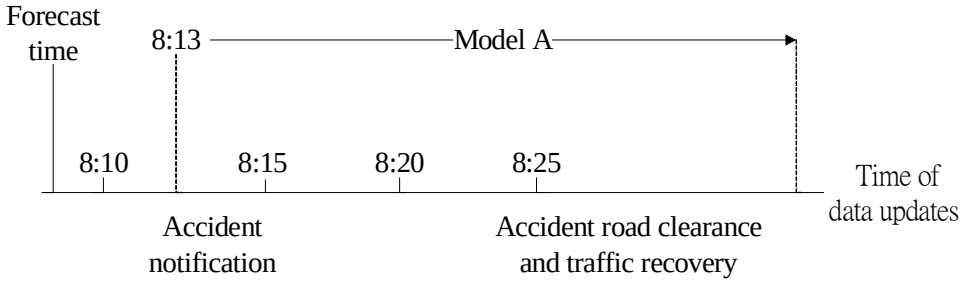


Fig. 1. The concept of accident duration forecasting

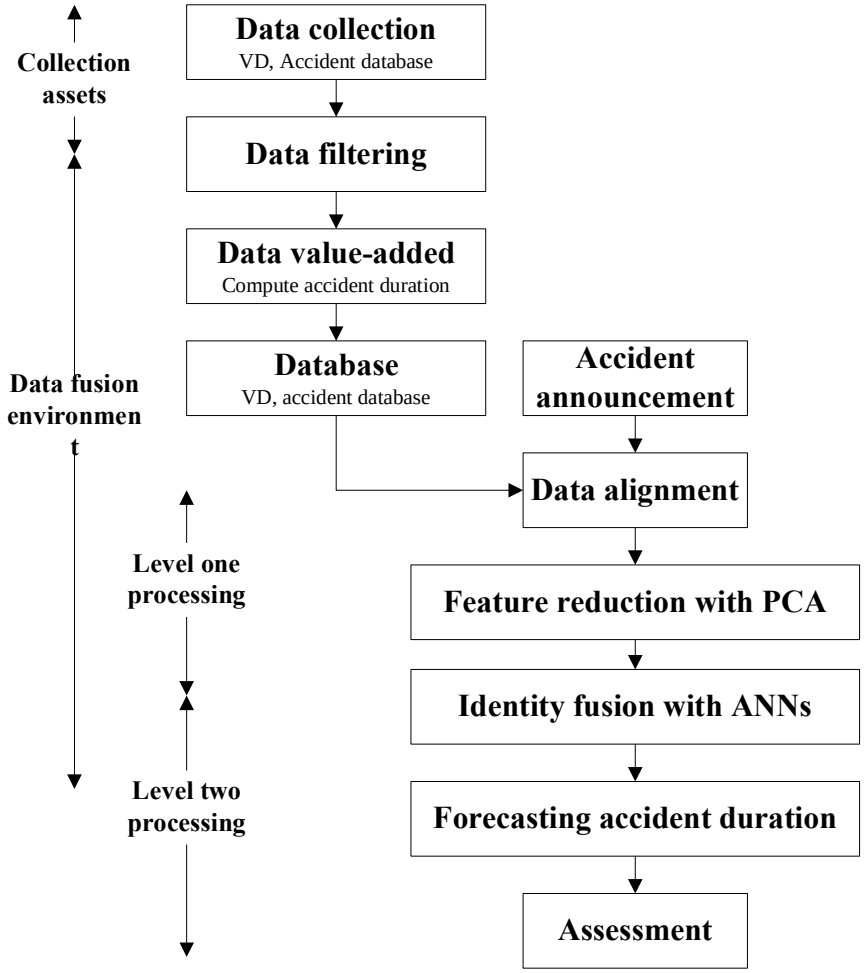


Fig. 2. Accident duration forecast flowchart

III. MODEL BUILDING

Previous studies found several highly significant factors for the present development of an accident duration model. Different from the hazard-based model and the regression approach, the study employs PCA method and Artificial Neural Networks (ANNs) techniques in modeling and builds model to forecast the accident duration at the moment of accident notification.

3.1 Model Structure

The flowchart of accident duration model is shown in Figure 2. The data sources are mostly from Vehicle Detector (VD) and accident database. The VD is the electronic equipment installed under the road and can record the time, the vehicle running speed, the vehicle volume and the vehicle occupancy. The VD data and the accident data are updated regularly for a fixed period of time. Relevant data are aligned according to available inputs and accident locations.

TABLE 1 INPUTS OF ACCIDENT DURATION FORECAST MODELS

	Factors	Variables	Descriptions
Inputs	Accident	Turn over	1:Yes, 0:No
		Occupied lane	1:Shoulder, 2:One lane, 3:Two lane
		# vehicles and types of vehicle involved in accident	# passenger car
			# non-passenger car
	Traffic data	Data from the first VD upstream before accident occurrence	Volume, Speed
		Data from the second VD upstream before accident occurrence	Volume, Speed
		Data from the first VD downstream before accident occurrence	Volume, Speed
	Time	The occurrence time	1:Day, 2:Night
		The time gap between the accident occurrence and the recording time of VD data before accident occurrence	Seconds
	Space	Distance between the accident and the first VD upstream	Kilometers
		Distance between the accident and the second VD upstream	Kilometers
		Distance between the accident and the first VD downstream	Kilometers
		Distance between the accident and the interchange upstream	Kilometers
		Distance between the accident and the interchange downstream	Kilometers
	Geometry	Location of accident	1:Interchange area, 2:Service facility area, 3:Other areas
		An interchange exists between the accident and the first VD upstream	1: Yes, 0: No
		An interchange exists between the accident and the first VD downstream	1: Yes, 0: No
		A toll plaza or service area exists between the accident and the first VD upstream	1: Yes, 0: No
		A toll plaza or service area exists between the accident and the first VD downstream	1: Yes, 0: No
Output	Accident Duration		Seconds

is the number of.

3.2 Model Inputs and Output

In Model, input selection is based on the reporting time at accident notification moment. In summary, as illustrated in Table 1, input variables include accident characteristics, traffic data, time relationship, space relationship, and geometry characteristics.

3.3 Feature reduction with principal comment analysis

The VD data are recorded and accumulated every 300sec for each lane. The features of traffic data from VD will exceed 48 items (8(car speed, bus speed, trailer speed, average speed, car volume, bus volume, trailer volume, occupancy)*2(Lanes)*3(upstream1, upstream2, downstream) =48). Using a large quantity of data features as the model inputs without careful processes may bring into the model significant noise. Therefore, data feature reduction with PCA method aims to decrease the number of model inputs and to preserve the relevant traffic characteristics with fewer inputs.

PCA is a technique for forming new variables which are linear composites of the original variable. The maximum number of new variables that can be formed is equal to the number of original variables, and the new variables are uncorrelated among themselves [12]. The equation of the PCA to reduce the data feature is as follows.

$$\begin{aligned}\xi_1 &= w_{11}x_1 + w_{12}x_2 + \cdots + w_{1p}x_p \\ \xi_2 &= w_{21}x_1 + w_{22}x_2 + \cdots + w_{2p}x_p \\ &\vdots \\ \xi_p &= w_{p1}x_1 + w_{p2}x_2 + \cdots + w_{pp}x_p\end{aligned} \quad \text{é (1)}$$

$$w_{i1}^2 + w_{i2}^2 + \cdots + w_{ip}^2 = 1 \qquad i=1, \text{é}\ , p \quad \text{é (2)}$$

$$w_{i1}w_{j1} + w_{i2}w_{j2} + \cdots + w_{ip}w_{jp} = 0 \quad \text{for all } i \neq j \quad \text{é (3)}$$

Where $\xi_1, \xi_2, \cdots, \xi_p$ are the principal components and w_{ij} is the weight of the jth variable for the ith principal component. p is the number of variables. x is the original variable.

The eigenvalue, percent of total variance and cumulate percent of total variance of each principal component are show in Table 2. The first principal component accounts for 23% of the total variance of the original data. The first two principal components account for 46% of the total variance of the original data. Through the PCA, the fewer number of variables, principal components, also can explain the most information of original data. Fig 3 shows the scree plots of eigenvalues. From

this figure, the elbow of scree plot appears at sixth principal component. Either first six or above first six principal components can be chosen as the model inputs [15].

Because the first ten principal components accounts over 90% of the total variance of the original data, this research choose ten principal components as the model inputs to put into ANNs model for identity fusion.

3.4 Identity fusion

In identity fusion, ANNs are the key technique. The ANNs approach is a data-driven, self-adaptive, and nonlinear methodology. After model training, the mapping between variables and accident duration is formed. Then by inputting the data from tested examples into this trained model, the forecasted accident duration is produced.

TABLE 2. EIGENVALUES OF CORRELATION MATRIX

Principal component	Eigenvalues	% of Total Variance	Cumulative % of Total variance
1st	5.0233	23.335	23.3349
2nd	3.7854	22.845	46.1801
3rd	2.4995	13.109	59.2893
4th	1.8135	9.502	68.7913
5th	1.1691	6.129	74.9202
6th	0.8263	4.332	79.2519
7th	0.6959	3.648	82.9002
8th	0.6205	3.253	86.1535
9th	0.5094	2.670	88.8239
10th	0.4236	2.221	91.0447
11th	0.3757	1.970	93.0144
12th	0.2302	1.207	94.2214
13th	0.1903	0.998	95.2193
14th	0.1725	0.904	96.1234
15th	0.1389	0.728	96.8514
é	é	é	é

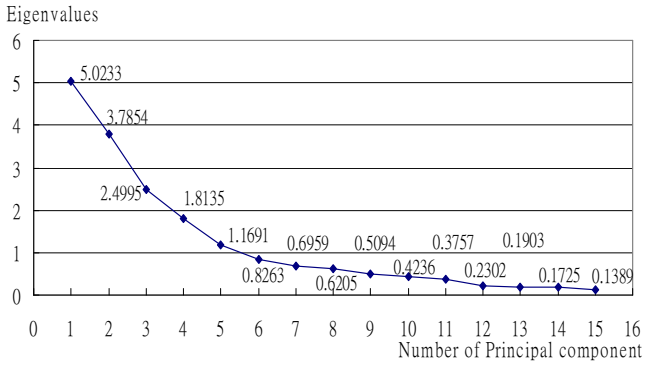


Fig. 3. Scree plots of eigenvalues

When inputting variables into the network of Multilayer Perceptron, the weights from input layer to hidden layer are calculated. Through the transfer function in the hidden layer, the input data are rescaled as inputs to the output layer. Since a discrepancy might occur between the estimated output and the actual accident duration, the weights are adjusted repeatedly by a suitable training method until the resulting error is stabilized and negligible. The accident duration function inside the ANN model is formed after this training procedure is completed [16].

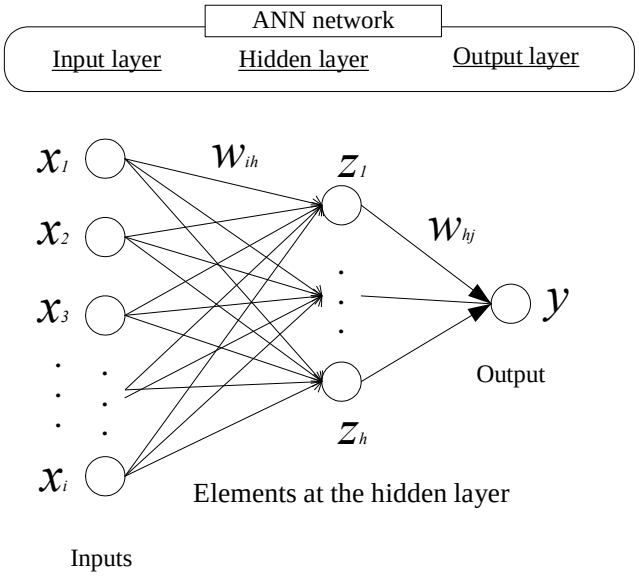


Figure 4 provides a schematic representation of a general ANN network. This study only estimates one output y, which is accident duration.

Fig. 4. Scheme of an ANN model

$$y = g\left(\sum_j w_{hj} \times f\left(\sum_i w_{ih} \times x_i - \theta_h\right) - \theta_j\right) \quad \text{é} \quad (4)$$

Where, y = Output variable (Accident duration); i = Elements at input layer; x_i = Input variables (Based on Table 1 to extract ten principal components through PCA); h = Elements at hidden layer; θ_h = Threshold values at hidden layer; w_{ih} = Weights between input layer and hidden layer; f = Transfer function at hidden layer; θ_j = Threshold values at output layer; w_{hj} = Weights between hidden layer and output layer; g = Transfer function at output layer.

There are 38, 20 and 1 nodes in the input, hidden and output layers, respectively. The conventional method (i.e., the average value of the numbers of input and output nodes) was selected to determine the number of hidden nodes. The data were scaled between [-1, 1]. The sigmoid method was chosen as the transfer function. In model training, the train-error slightly decreases from 10-2 in 300 epochs to 10-3 in 1000 epochs. The performance improvement from 300 epochs to 1000 epochs is less than 0.09. Therefore, the difference of training performance is not significant between 300 epochs and 1000 epochs. The epoch was set as 300 for training in this study. Training time is generally less than 17 sec.

3.5 Model results

The 39 accidents are divided into two parts, 24 accidents for model training and 15 accidents for model testing. In this section, the model performance is compared by MAE, R2 and MAPE.

MAE value is smaller, and the forecasted accident duration is closer to the actual accident duration.

$$MAE = \frac{1}{M}\sum_{k=1}^{M}\left|\hat{x}(k) - x(k)\right| \quad \text{é (5)}$$

Where, M = Total number of examples; k = The k_{th} example; $\hat{x}$ = Forecasted accident duration; x = Actual accident duration. As the R2 is closer to 1, and the information supplied by inputs is more helpful to output.

$$R^2 = \frac{\sum_{k=1}^{M}\left(\hat{x}(k) - \bar{x}(k)\right)^2}{\sum_{k=1}^{M}\left(x(k) - \bar{x}(k)\right)^2} \quad \text{é (6)}$$

Where, $\bar{x}$ = Mean of actual accident duration

The MAPE has been chosen as the primary criterion for model evaluation, as shown in equation 7. Typical MAPE values for performance assessment are shown in Table 3 [17]. As the MAPE approaches 0, the forecasted value becomes more accurate.

$$MAPE = \frac{1}{M}\sum_{k=1}^{M}\left|\frac{\hat{x}(k) - x(k)}{x(k)}\right| \times 100\% \quad \text{é (7)}$$

TABLE 3. MAPE CRITERIA FOR MODEL EVALUATION

MAPE (%)	Assessment
<10	Highly accurate forecasting
10-20	Good forecasting
20-50	Reasonable forecasting
>50	Inaccurate forecasting

Table 4 shows the model performance by MAE, R^2, MAPE and computing time. In terms of the MAE, R^2 and MAPE, the performance of PCA method is better. This result indicates that PCA method could decrease the numbers of features and preserve the data meaning which is similar as the data with no reduction.

TABLE 4. MODEL PERFORMANCE

	No reduction	PCA (10 principal components)
MAE(sec)	585.838	553.044
R^2	0.896	0.908
MAPE(%)	12.371	10.760
Computing time- feature reduction(sec)	0.000	0.117
Computing time- fusion(sec)	10.235	10.136
The number of the variables from VD	48	10

IV. CONCLUSIONS AND COMMENTS

4.1 Conclusions

This study presents accident duration forecasting model using Artificial Neural Networks. In terms of MAE, R^2 and MAPE, the proposed model has shown good and stable model performance. The MAPE percentages are mostly under 13%. This shows that the developed model fits the actual accident duration well during the accident, and that Artificial Neural Networks effectively smooth data noise of the model.

In terms of the model effect, the accident characteristics, VD data, time relationship, space relationship, and geometry characteristics are feasible as the inputs of accident duration forecasting model.

PCA method can decrease the number of model inputs and preserve the relevant traffic characteristics with a few inputs.

The travelers and traffic management units can generally realize the impact by the forecasted accident duration. From the assessment of model effects, this study shows proposed models are feasible ones in the Intelligent Transportation Systems (ITS) context.

4.2 Comments for future study

This research considers the elbow of scree plot and cumulate percent of total variance to choose first ten components as the

model inputs to put into ANNs model for identity fusion. Future study may discuss the model performance with the number of principal components.

There are a lot of feature reduction methods, such as Cluster, Independent Component Analysis, Genetic Algorithm. Future study may consider other methods to reduce the feature and preserve the information for modeling.

Besides ANN, future study may consider adopt other algorithms to conduct model for identity fusion and compare the model performance.

REFERENCE

[1] Garib, A., Radwan, A. E. and Al-Deek, H. (1997) Estimating magnitude and duration of accident delays, *Journal of Transportation Engineering*, 123(6), 459-466.

[2] Nam, D. and Mannering, F. (2000) An exploratory hazard-based analysis of highway accident duration. *Transportation Research, Part A*, 34(2), 85-102.

[3] Wei, C. H. and Lee, Y. (2005) Appling data fusion techniques to traveler information services in highway network, *Journal of the Eastern Asia Society for Transportation Studies*, 6, 2457-2475.

[4] Dougherty, M. S. (1997) Applications of neural networks in transportation. *Transportation Research, Part C*, 5(5), 255-257.

[5] Dharia, A. and Adeli, H. (2003) Neural network model for rapid forecasting of freeway link travel time. *Engineering Application of Artificial Intelligence*, 16(7-8), 607-613.

[6] Wei, C. H. & Chen, Y. C. (2001), Review of artificial neural network research and applications in transportation, *Transportation Planning Journal Quarterly*, **30**(2), 324-348.

[7] Chen, M., Liu, X. B., Xia, J. X. & Chien, S. I. (2004), A Dynamic Bus-Arrival Time Prediction Model Based on APC Data, *Computer-Aided Civil and Infrastructure Engineering*, **19**(5), 364-376.

[8] Sun, C., Ritchie, S. G. & Oh, S. (2003), Inductive Classifying Artificial Network for Vehicle Type Categorization, *Computer-Aided Civil and Infrastructure Engineering*, **18**(3), 161-172.

[9] Ghosh-Dastidar, S. & Adeli, H. (2003), Wavelet-Clustering-Neural Network Model for Freeway Incident Detection, *Computer-Aided Civil and Infrastructure Engineering*, **18**(5), 325-338.

[10] Xu, H. & Humar, J. M. (2006), Damage Detection in a Girder Bridge by Artificial Neural Network Technique, *Computer-Aided Civil and Infrastructure Engineering*, **21**(6), 450-464.

[11] Lam, H. F., Yuen, K. V. & Beck, J. L. (2006), Structural Health Monitoring via Measured Ritz Vectors Utilizing Artificial Neural Networks, *Computer-Aided Civil and Infrastructure Engineering*, **21**(4), 232-241.

[12] Sharma, S. (1996) *Applied Multivariate Techniques*. Willey, New York.

[13] Sengur, A. (2008) An expert system based on principal component analysis, artificial immune system and fuzzy k-NN for diagnosis of valvular heart diseases. *Computers in Biology and Medicine*, 38(3), 329-338.

[14] Phillips, R. D., Watson, L. T., Wynne, R. H., and Blinn, C. E. (2009) Feature reduction using a singular value decomposition for the iterative guided spectral class rejection hybrid classifier. *Journal of Photogrammetry and Remote Sensing*, 64(1), 107-116.

[15] Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate Behavioral Research*, 1, 245-276

[16] Zurada, J. M. (1992) *Introduction to Artificial Neural Systems*. West publishing company, St. Paul.

[17] Lewis, C. D. (1982), *Industrial and Business Forecasting Method*. Butter worth Scientific, London.

ACKNOWLEDGMENT

This paper was derived from a research project sponsored by the National Science Council, Taiwan under the contract of NSC99-2628-E-451-001. I gratefully acknowledge the data providers for this research, including Taiwan Area National Freeway Bureau and Police Radio Station.

Ying Lee, Assistant Professor of the Department of Hospitality Management at the Ming Dao University, holds a Ph.D. degree from the National Cheng Kung University (NCKU), Tainan, Taiwan. His current research activities and interests include: sequential accident duration forecasting and dynamic travel time prediction in congested traffic networks. He is a member of the East Asia Society for Transportation Studies.

Example Labeling Difficulty within Repeated Labeling

Victor S. Sheng
Computer Science Department, University of Central Arkansas, Conway, AR, USA

Abstract— *This paper addresses the repeated acquisition of labels for data items when the labeling is imperfect. We examine the improvement (or lack thereof) in data quality via repeated labeling, and focus especially on the improvement of training labels for supervised induction. We show that the repeated-labeling strategies proposed acquire more labels for difficult examples automatically. Among these strategies, the integrated one, combining the uncertainty of the multiple label with the uncertainty predicted from classification models, performs better than the strategies based on the two uncertainties respectively.*

Keywords: Repeated-labeling, noisy labeling, data preprocessing

1. Introduction

Over the past decade we have witnessed a major change in how people can interact with computers to solve problems. The development of marketplaces for *micro-outsourcing*, such as RentaCoder and Amazon's Mechanical Turk, allows the outsourcing of small tasks to workers over the Internet. Researchers have studied for example the task of identifying the sentiment of investors when discussing a specific stock [1]. An important step for building a document classifier is to have enough training data, typically thousands of cases. Amazon Mechanical Turk can be used to scale the annotation process, by allowing hundred of human labelers to look at articles and label them, using an interface as the one in Figure 1. Using such marketplaces, it is possible to outsource small parts of the process at very low cost—parts that prior to the introduction of such systems would have incurred much higher (in-house) cost, or would have been avoided altogether.

Fig. 1: An example of a micro-task submitted to the Amazon Mechanical Turk marketplace.

In this paper, we consider issues of data acquisition and data quality for decision making and modeling. Micro-outsourcing provides a new alternative for the large-scale acquisition of data from non-experts. These data can be used directly in decision making or for modeling. For example, NASA in the "*Clickworkers*" project[1] used 85,000 volunteers to classify landmarks in Mars images as crater, gully, dust devil track, hill, ridge, etc. Such classifications can be used directly by query systems, or can be used to train models to recognize landmark types automatically.

With micro-outsourcing, an important consideration is the accuracy of the data acquired. For image classification and text labeling, even a non-expert human brain often can provide very useful labeling. However, labeling error rates may still be significant, due to lack of expertise, dedication, attention, interest, or other factors. This paper is related to *repeated labeling* (re-labeling) [2]: taking advantage of the low cost of micro-outsourcing to acquire *multiple* labels for data points, in order to improve data quality and the quality of models built from the data. This preliminary work [2] discusses some basic *repeated labeling* strategies, where it assumed that the labeling difficulty of each example is the same. However, this is not true in real-world applications. This paper focuses on this issue. It studies the performance and the properties of the repeated-labeling strategies proposed in [2] when the labeling difficulty is example dependent.

2. Related Work

Repeatedly labeling the same data point is practiced in applications where labeling is not perfect (e.g., [3], [4]). We are not aware of a systematic assessment of the relationship between the resultant quality of supervised modeling and the number of, quality of, and method of selection of data points for repeated-labeling. To our knowledge, the typical strategy used in practice is what we call "round-robin" repeated-labeling, where cases are given a fixed number of labels—so we focus considerable attention in the paper to this strategy. A related important problem is how in practice to assess the generalization performance of a learned model with uncertain labels [3], which we do not consider in this paper. Prior research has addressed important problems necessary for a full labeling solution that uses multiple noisy labelers, such as estimating the quality of labelers [5], [3], [6], and learning with uncertain labels [7], [8], [9]. So we treat these topics quickly when they arise, and lean on the prior work.

[1] http://clickworkers.arc.nasa.gov

Repeated-labeling using multiple noisy labelers is different from multiple label classification [10], [11], where one example could have multiple *correct* class labels. As we discuss in Section 5, repeated-labeling can apply regardless of the number of true class labels. The key difference is whether the labels are noisy. A closely related problem setting is described by Jin and Ghahramani [12]. In their variant of the multiple label classification problem, each example presents itself with a set mutually exclusive labels, one of which is correct. The setting for repeated-labeling has important differences: labels are acquired (at a cost); the same label may appear many times, and the true label may not appear at all. Again, the level of error in labeling is a key factor.

The consideration of data acquisition costs has seen increasing research attention, both explicitly (e.g., cost-sensitive learning [13], utility-based data mining [14]) and implicitly, as in the case of active learning [15]. Turney [13] provides a short but comprehensive survey of the different sorts of costs that should be considered, including data acquisition costs and labeling costs. Most previous work on cost-sensitive learning does not consider labeling cost, assuming that a fixed set of labeled training examples is given, and that the learner cannot acquire additional information during learning (e.g., [16], [17], [18]).

Active learning [15] focuses on the problem of costly label acquisition, although often the cost is not made explicit. Active learning (cf., optimal experimental design [19]) uses the existing model to help select additional data for which to acquire labels [20], [21], [22]. The usual problem setting for active learning is in direct contrast to the setting we consider for repeated-labeling. For active learning, the assumption is that the cost of labeling is considerably higher than the cost of obtaining unlabeled examples (essentially zero for "pool-based" active learning).

Some previous work studies data acquisition cost explicitly. For example, several authors [23], [24], [25], [26], [27], [28], [29] study the costly acquisition of feature information, assuming that the labels are known in advance. Saar-Tsechansky et al. [27] consider acquiring both costly feature and label information.

None of this prior work considers selectively obtaining multiple labels for data points to improve labeling quality, and the relative advantages and disadvantages for improving model performance. An important difference from the setting for traditional active learning is that labeling strategies that use multiple noisy labelers have access to potentially relevant additional information. The multisets of existing labels intuitively should play a role in determining the examples for which to acquire additional labels. For example, presumably one would be less interested in getting another label for an example that already has a dozen identical labels, than for one with just two, conflicting labels.

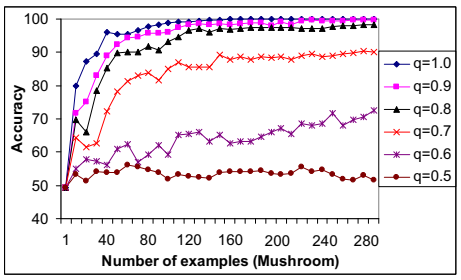


Fig. 2: Learning curves under different quality levels of training data (q is the probability of a label being correct).

3. Example Labeling Difficulty d

Figure 2 illustrates that the quality of the labels can have a marked effect on classification accuracy. Intuitively, using multi-labeling to shift from a lower-p curve to a higher-p curve can, under some settings, improve learning considerably. In order to treat this more formally, we first introduce some terminology and simplifying assumptions.

In a noisy labeling environment, we need careful strategies to decide which examples to (re-)label. Should we get more unlabeled training examples and label them, or should we get additional labels for already labeled examples? And which examples should be (re-)labeled? In this section we review the strategies [2] and study their performance where the labeling difficulty of each example is different, not the same which studied in [2]. Now let us briefly review these strategies.

3.1 Existing Repeated-labeling Strategies

A basic choice in a noisy labeling environment is to decide whether to acquire a large number of noisy examples, with one label each (single labeling, *SL*), or whether we should acquire multiple noisy labels for each example, hoping to improve the label quality and the quality of the learned model. The simplest re-labeling strategy is to acquire the same number of labelel per example (round robin). We call this strategy *fixed round-robin* (*FRR* in short). A slight generalization of *FRR* is to always give the next label to the example with the fewest labels; we call this labeling strategy *generalized round-robin* (*GRR* in short). In [2] we showed that round-robin repeated-labeling can give significantly better results over single-labeling, especially under high levels of noise.

Going beyond simple round-robin schemes, more effective strategies for re-labeling ideally should focus on examples with high levels of *uncertainty*. For example, intuitively it would seem better to get more labels for an example with the label multiset $\{+,-,+\}$ compared to an example with the label multiset $\{+,+,+,+,+,+\}$ or one with $\{+,+,+,-,-,-,+,+,+\}$. In our preliminary work, we showed how to compute the *label uncertainty (LU)* using Bayesian estimation $B(p+1,n+1)$. The LU score S_{LU} is,

$$S_{LU} = \min\{I_{0.5}(p+1,n+1), 1-I_{0.5}(p+1,n+1)\} \quad (1)$$

where

$$I_x(\alpha,\beta) = \sum_{j=a}^{\alpha+\beta-1} \frac{(\alpha+\beta-1)!}{j!(\alpha+\beta-1-j)!} x^j (1-x)^{\alpha+\beta-1-j} \tag{2}$$

with the decision threshold is $x = 0.5$.

Another type of uncertainty is *model uncertainty (MU)*, initially inspired by active learning. The MU score S_{MU} is,:

$$S_{MU} = 0.5 - \left| \frac{1}{m} \sum_{i=1}^{m} Pr(+|x, H_i) - 0.5 \right| \tag{3}$$

where $Pr(+|x, H_i)$ is the probability of classifying the example x into $+$ by the learned model H_i, and m is the number of learned models.

MU focuses on examples that are the most uncertain according to the learned classifier, aiming to improve their quality by acquiring additional noisy labels. Importantly, our most recent preliminary results suggest that MU works for a different reason than active learning—MU essentially is a self-healing process. Including the currently estimated y_i's for training and then applying the model back to the same x_i's leads MU to select examples whose erroneous classification dilutes the model's estimate of its probability of class membership, illustrating the advantage of the additional source of information (the current label multisets) over the traditional active learning setting. We will develop this line of reasoning further, both looking more deeply at the existing MU method, and asking whether the results suggest an improved MU. For example, it may be useful to look not only at the uncertainty of the model-estimated class of an example, but whether it disagrees with the integration of the current label multiset. Along this line, in our experiments [2], the strategy *LMU* that *combined both label and model uncertainty using geometric mean* performed best. Therefore as we develop new methods for LU and MU, we will also experiment with combinations of the novel techniques. Of course, we also plan to examine multiple learning algorithms, to examine their sensitivity to noise; we expect our techniques, though, to be agnostic and orthogonal to the underlying learning algorithm.

4. Repeated-labeling and Modeling

We have studied the performance of the repeated-labeling strategies under the assumption that the labeling difficulty of all examples is the same. Actually, this is not true in real-world applications. In the following, we will study the repeated-labeling strategies under the real-world situation. That is, the labeling difficulty varies among different examples. We are wondering what kinds of examples will obtain more labels during active learning: the more difficult ones, or the easier ones?

4.1 Experimental Setup

Practically speaking, the answers to these questions rely on the conditional distributions being modeled, and so we shift to an empirical analysis based on experiments with benchmark data sets.

To investigate the questions above, we present experiments on 12 real-world datasets from [30]. These datasets were chosen because they are classification problems with a moderate number of examples, allowing the development of learning curves based on a large numbers of individual experiments. If necessary, we convert the target to binary (for *thyroid* we keep the negative class and integrate the other three classes into positive; for *splice*, we integrate classes IE and EI; for *waveform*, we integrate class 1 and 2.)

For each dataset, 30% of the examples are held out, in every run, as the test set from which we calculate generalization performance. The rest is the "pool" from which we acquire unlabeled and labeled examples. To simulate noisy label acquisition, we first hide the labels of all examples for each dataset. At the point in an experiment when a label is acquired, we generate a label according to the labeling difficulty d ($d \in [0, 0.5]$): we assign the example's original label with probability $1 - d$ and the opposite value with probability d.

After obtaining the labels, we add them to the training set to induce a classifier. For the results presented, models are induced with J48, the implementation of C4.5 [31] in WEKA [32]. The classifier is evaluated on the test set (with the true labels). Each experiment is repeated 10 times with a different random data partition, and average results are reported.

4.2 Repeated-labeling Strategies with Example Difficulties

In [2], we assumed that the difficulty of labeling an example is a constant across all examples. In reality, some examples are more difficult to label than others. That is, the labeling difficulty is example-dependent. How does the labeling difficulty affect the selective repeating labeling strategies? Does more difficult example need more labels?

To investigate the questions above, we introduce the degree of difficulty d of labeling an example and study the performance of the selective repeating strategies (*MU*, *LU*, and *LMU*) under this case. Since the labeling difficulty of each example is not available in the benchmark datasets, we simulate this by assigning a difficulty (d, a decimal value) to each example. In this paper, we focus on binary classification. The difficulty d is setting in the range $[0, 0.5]$. 0 means no difficult to label an example. The label obtained for this example is correct. It is no need to acquire repeated labels for it. 0.5 means the most difficult to label an example. For multiple classifications, it can be extended easily.

Although the difficulty distribution is unknown for all the datasets in our experiments, we can simulate the labeling

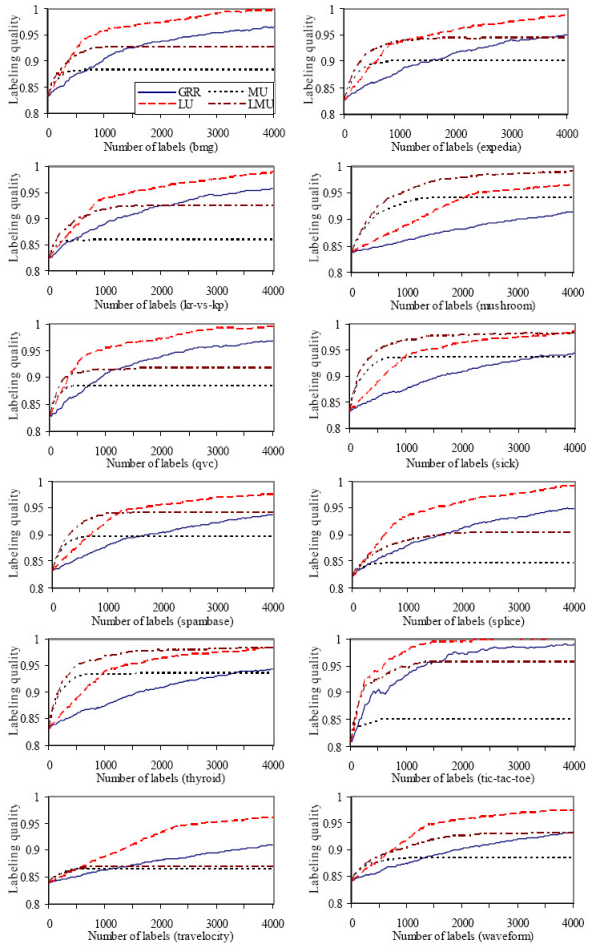


Fig. 3: The data quality improvement of the four strategies (*GRR*, *LU*, *MU*, and *LMU*) for the labeling quality d under beta distribution.

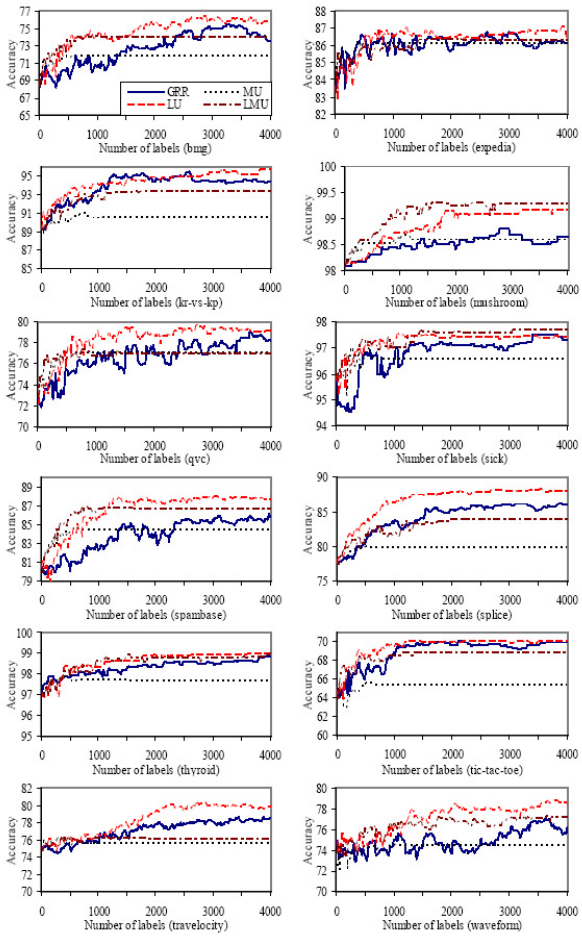


Fig. 4: Accuracy as a function of the number of labels acquired for the four selective repeated-labeling strategies(*GRR*, *LU*, *MU*, and *LMU*). The labeling quality d is under beta distribution.

difficulty in different distributions, such as uniform distribution, beta distribution, and so on. Furthermore, we also use learning model to predict the labeling difficulty for each example. Details are shown in the following subsections.

4.2.1 Beta Distribution

First we simulate the label difficulty d follows uniform distribution, then follows beta distribution. For uniform distribution, we randomly generate decimal values in the range $[0, 0.5]$ as the difficulties to assign to examples in each datasets. For beta distribution, the labeling difficulty d (again in the range $[0, 0.5]$) is generated from the specific beta distribution (i.e., Beta function $B(\alpha, \beta)$). In our experiment, we choose $\alpha = \beta = 5$.

We evaluate the performance the four strategies under the two distributions using over 10 fold opposite cross validation (OCV in short). The process of OCV is similar to cross validation (CV). It is known that CV uses one fold for testing, the rest for training. On the contrary to CV, OCV uses one fold for training, the rest for testing. We did this to avoid the impact of large training sets. Large training size can reduce the impact of noise data. One of the reason is the data redundancy. For example, there are many redundancies in the dataset Mushroom. Besides, the impact of the limited repeated labeling does not produce significant improvement over the whole training data.

Our experimental results show that the four strategies perform similar under different distributions. Because of the space limitation, we only show the experiment results of beta distribution to represent. Besides, beta distribution can represent different distributions when different values for α and β in the beta function are chosen. For example, if we assign $\alpha = \beta = 0$, then it represents a uniform distribution. If we assign $\alpha = \beta = \infty$ or a big value, then it goes to one value. This is the case where all examples have the same difficulty. Note that the labeler quality value equals to the constant difficulty one. The experiment results are shown in Figures 3 and 4.

From Figure 3 above, we can see that the four strategies still improve the data quality for the labeling difficulty d under beta distribution. The four strategies perform similar

under uniform distribution. However, the relationship among the four strategies are different from the labeling difficulty d under uniform distribution. For uniform distribution, the three selective strategies consistently perform better than *GRR*. Among the three strategies, *LMU* performs the best, followed by *MU*, followed by *LU*, in term of accuracy and labeling quality. Under beta distribution, except *LU*, *MU* and *LMU* do not outperform *GRR* consistently. *LMU* and *MU* perform worse than *GRR* on some datasets (bmg, qvc, splice, tic-tac-toe, and travelocity), although they outperform *GRR* consistently at the beginning of the learning curves. Particularly, *MU* performs worse than *GRR* on most datasets, except that it outperforms *GRR* on *mushroom*. It is obvious that the learning curve of *MU* on all datasets becomes flat after certain amount labels acquired. It is also ostensible that *LMU* is affected by *MU*. Its learning curve also becomes flat after certain amount labeled acquired. However, the turning point of *LMU* is delayed because of the impact of *LU*. *LU* performs the best over all datasets. Particularly the budget of acquiring more labels is high. If the budget is very limited (about less than 1000 labels), *LMU* is the best choice. From Figure 4, we can observe the similar relationship.

4.2.2 Model Prediction

Either uniform distribution or beta distribution, the labeling difficulty d for each example is generated from the given distribution. It may not reflect the reality. In reality, some examples are hard to label and others are not in a dataset. To simulate the real labeling difficulty for each example, we build learning models from the training dataset, where each example has its true label. In this experiment, the original label from the dataset is assumed the true label. Based on the training dataset, we apply the active learning technique through building an ensemble of random trees to predict the uncertainty of each training example. The uncertainty is assigned as the labeling difficulty. The labeling difficulty does not change in the later label acquisition process. The others of the experiment settings as the same as uniform distribution and beta distribution. Our experiment results are shown in Figures 5 and 6.

From Figure 5 we can see the relationship among the four strategies is similar to beta distribution overall. *LU* performs consistently better than *GRR*. *LMU* performs better than *GRR* on most datasets, except (bmg, splice, and tic-tac-toe). *MU* performs better than *GRR* at the beginning of the learning curve and performs worse than *GRR* after certain amount of labels acquired on most datasets, except three datasets (mushroom, sick, and thyroid). On these three datasets, *LMU* performs the best. *LU* performs the best on others. Again, If the budget is very limited (about less than 1000 labels), *LMU* is the best choice. Otherwise, *LU* is. These discussions are also fit in the model performance of the four strategies shown in Figure 6.

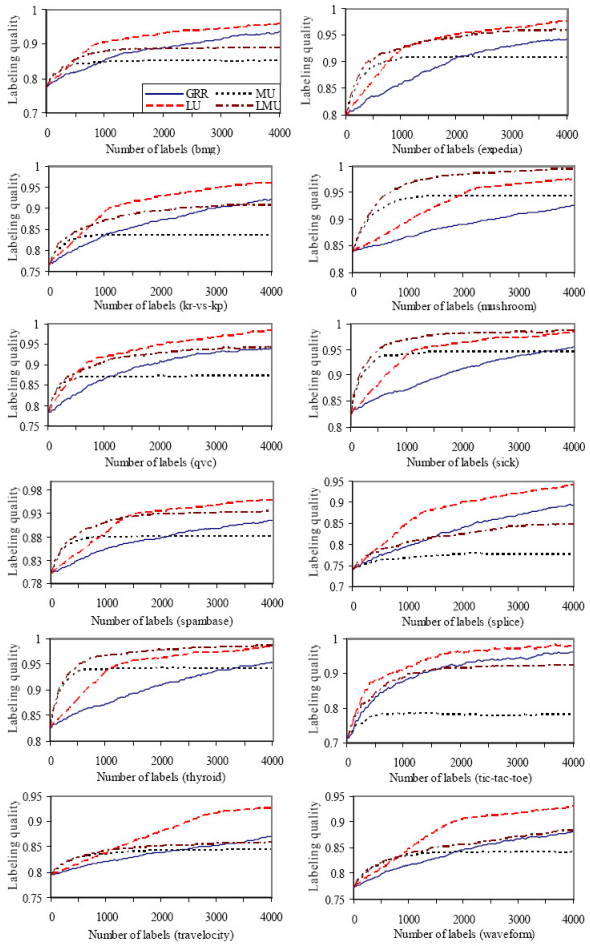


Fig. 5: The data quality improvement of the four strategies (*GRR*, *LU*, *MU*, and *LMU*) for the labeling quality d under model prediction.

4.2.3 Distribution of Acquired Label Against Labeling Difficulties

The previous subsections examined the performance of the four selective repeated labeling strategies under different labeling difficulty distributions (Uniform distribution, Beta distribution, and Model uncertainty). We wonder what examples are chosen by these strategies. Is there any relation with the labeling difficulty d? The answer of this question can be illustrated by the label (acquired) distribution along with difficulties (Seeing Figure 7).

From the figure above, we can see that *LU* extremely acquires many labels for examples with high labeling difficulties ($d \in [0.4, 0.45)$) (Note: there is no examples in the segment ($[0.45, 0.5]$)). However, *GRR* uniformly acquires more labels for all different labeling difficulties. *MU* is a typical *'M'* shape. It acquires more labels for examples with labeling difficulties in two segments ($d \in [0.1, 0.15)$ and $d \in [0.35, 0.4)$), particularly the later segment. The distribution of *LMU* is affected by *LU* and *MU*. However, its shape is close to that of *LU*.

We further look insight of the relationship the label

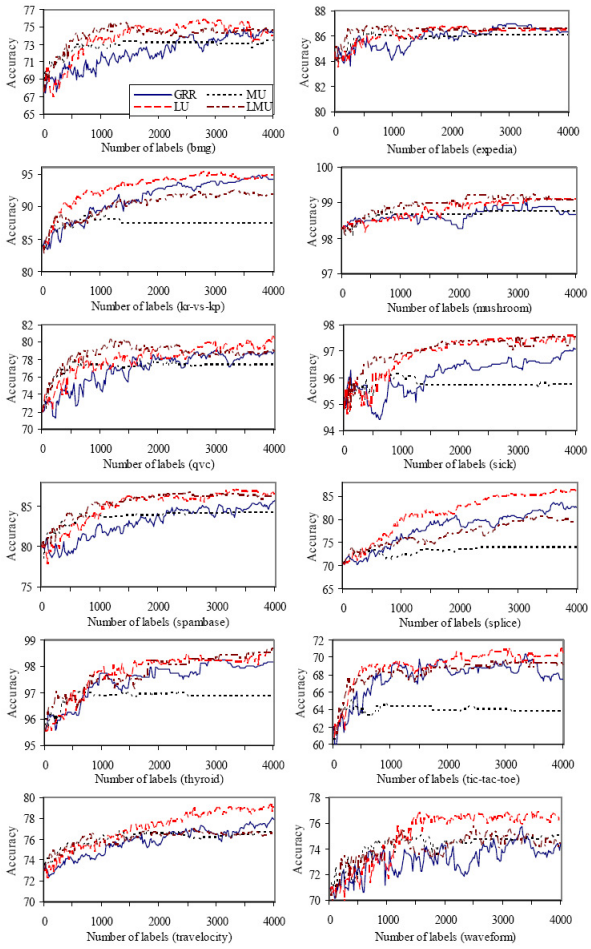


Fig. 6: Accuracy as a function of the number of labels acquired for the four selective repeated-labeling strategies(*GRR*, *LU*, *MU*, and *LMU*). The labeling quality d is assigned in terms of model prediction.

distribution and the labeling difficulty under beta distribution for at the different learning steps of the four strategies. The experiment results on the *mushroom* dataset are shown in Figures 8, 9, 10, and 11. From these figures we can see that *GRR* uniformly distributes the acquired labels on each segment in each learning step. *MU* gradually puts high weight on the two summits of the *'M'* shape. Like *MU*, *LU* gradually acquires more labels for the examples in the segment with high labeling difficulty. It is excited that *LMU* can find the high labeling difficult examples quickly. From Figure 11 we can see that there is a huge jump under the most difficult segment ($[0.4, 0.45]$)from the learning curve *step-10* to the learning curve *step-50*. This could be the reason why *LMU* performs the best at the beginning stages of the label acquisition (refer to Figure 3). As each step there are 20 labels acquired, there are 1000 labels acquired after 50 steps. Thus, this shows that when the budget is very limited (about less than 1000 labels), *LMU* is the best choice.

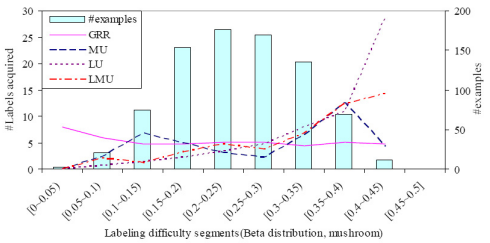


Fig. 7: The acquired label distribution along with the labeling difficulty d (under beta distribution, specifically B(5,5)) under the four strategies (*GRR*, *LU*, *MU*, and *LMU*) for the *mushroom* dataset.

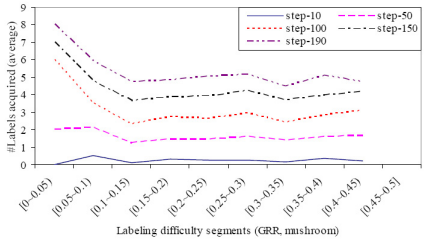


Fig. 8: The acquired label distribution along with the labeling difficulty d (under beta distribution, specifically B(5,5)) at the different learning steps of the strategy *GRR* for the *mushroom* dataset.

5. Conclusions and Future Work

Repeated-labeling is a tool that should be considered whenever labeling might be noisy, but can be repeated. We showed that the selective repeated-labeling strategies can improve both the quality of the labeled data directly, and the quality of the models learned from the data, under different labeling difficulty distribution (Uniform distribution, Beta distribution, and Model uncertainty). In particular, the repeated-labeling strategy *LMU* seems to be preferable, taking into account both labeling uncertainty and model uncertainty.

In future, we will study the development and analysis of more sophisticated re-labeling techniques, which makes use of the example labeling difficulty. We will also propose to look carefully at estimating labeler quality and case difficulty, extending techniques developed in statistics, in order that these quantities be available to the selective relabeling strategies.

Acknowledgment

We thank the anonymous reviewers for the valuable comments. The work was supported by the National Science Foundation (IIS-1115417).

References

[1] W. Antweiler and M. Z. Frank, "Is all that talk just noise? the information content of internet stock message boards," *Journal of Finance*, vol. 59, no. 3, pp. 1259–1294, 2004.

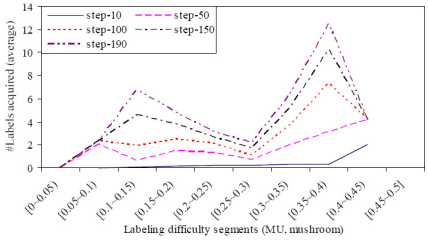


Fig. 9: The acquired label distribution along with the labeling difficulty d (under beta distribution, specifically B(5,5)) at the different learning steps of the strategy *MU* for the *mushroom* dataset.

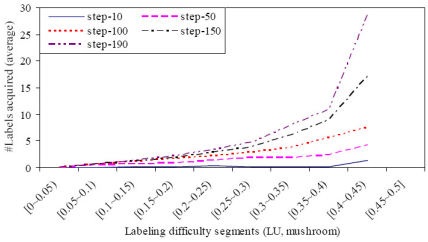


Fig. 10: The acquired label distribution along with the labeling difficulty d (under beta distribution, specifically B(5,5)) at the different learning steps of the strategy *LU* for the *mushroom* dataset.

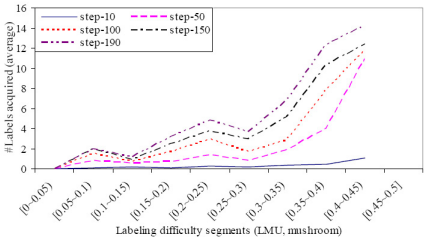


Fig. 11: The acquired label distribution along with the labeling difficulty d (under beta distribution, specifically B(5,5)) at the different learning steps of the strategy *LMU* for the *mushroom* dataset.

[2] V. S. Sheng, F. Provost, and P. Ipeirotis, "Get another label? Improving data quality and data mining using multiple, noisy labelers," in *Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2008)*, 2008.

[3] P. Smyth, U. M. Fayyad, M. C. Burl, P. Perona, and P. Baldi, "Inferring ground truth from subjective labelling of Venus images," in *Advances in Neural Information Processing Systems 7 (NIPS 1994)*, 1994, pp. 1085–1092.

[4] P. Smyth, M. C. Burl, U. M. Fayyad, and P. Perona, "Knowledge discovery in large image databases: Dealing with uncertainties in ground truth," in *Knowledge Discovery in Databases: Papers from the 1994 AAAI Workshop (KDD-94)*, 1994, pp. 109–120.

[5] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *Applied Statistics*, vol. 28, no. 1, pp. 20–28, Sep. 1979.

[6] P. Smyth, "Bounds on the mean classification error rate of multiple experts," *Pattern Recognition Letters*, vol. 17, no. 12, pp. 1253–1257, May 1996.

[7] P. Smyth, "Learning with probabilistic supervision," in *Computational Learning Theory and Natural Learning Systems, Vol. III: Selecting Good Models*, T. Petsche, Ed. MIT Press, 1995.

[8] G. Lugosi, "Learning with an unreliable teacher," *Pattern Recognition*, vol. 25, no. 1, pp. 79–87, Jan. 1992.

[9] B. W. Silverman, "Some asymptotic properties of the probabilistic teacher," *IEEE Transactions on Information Theory*, vol. 26, no. 2, pp. 246–249, Mar. 1980.

[10] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, Sep. 2004.

[11] A. McCallum, "Multi-label text classification with a mixture model trained by EM," in *AAAI'99 Workshop on Text Learning*, 1999.

[12] R. Jin and Z. Ghahramani, "Learning with multiple labels," in *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, 2002, pp. 897–904.

[13] P. D. Turney, "Types of cost in inductive concept learning," in *Proceedings of the ICML-2000 Workshop on Cost-Sensitive Learning*, 2000, pp. 15–21.

[14] F. Provost, "Toward economic machine learning and utility-based data mining," in *Proceedings of the 1st International Workshop on Utility-based Data Mining (UBDM'05)*, 2005, pp. 1–1.

[15] D. A. Cohn, L. E. Atlas, and R. E. Ladner, "Improving generalization with active learning," *Machine Learning*, vol. 15, no. 2, pp. 201–221, May 1994.

[16] P. Domingos, "MetaCost: A general method for making classifiers cost-sensitive," in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-99)*, 1999, pp. 155–164.

[17] C. Elkan, "The foundations of cost-sensitive learning," in *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)*, 2001, pp. 973–978.

[18] P. D. Turney, "Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm," *Journal of Artificial Intelligence Research*, vol. 2, pp. 369–409, 1995.

[19] P. Whittle, "Some general points in the theory of optimal experimental design," *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 35, no. 1, pp. 123–130, 1973.

[20] M. Saar-Tsechansky and F. Provost, "Active sampling for class probability estimation and ranking," *Journal of Artificial Intelligence Research*, vol. 54, no. 2, pp. 153–178, 2004.

[21] D. D. Margineantu, "Active cost-sensitive learning," in *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, 2005, pp. 1622–1613.

[22] Y. Baram, R. El-Yaniv, and K. Luz, "Online choice of active learning algorithms," *Journal of Machine Learning Research*, vol. 5, pp. 255–291, Mar. 2004.

[23] G. M. Weiss and F. J. Provost, "Learning when training data are costly: The effect of class distribution on tree induction," *Journal of Artificial Intelligence Research*, vol. 19, pp. 315–354, 2003.

[24] D. J. Lizotte, O. Madani, and R. Greiner, "Budgeted learning of naive-bayes classifiers," in *19th Conference on Uncertainty in Artificial Intelligence (UAI 2003)*, 2003, pp. 378–385.

[25] P. Melville, M. Saar-Tsechansky, F. J. Provost, and R. J. Mooney, "Active feature-value acquisition for classifier induction," in *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004)*, 2004, pp. 483–486.

[26] P. Melville, F. J. Provost, and R. J. Mooney, "An expected utility approach to active feature-value acquisition," in *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005)*, 2005, pp. 745–748.

[27] M. Saar-Tsechansky, P. Melville, and F. J. Provost, "Active feature-value acquisition," University of Texas at Austin, McCombs Research Paper Series, Tech. Rep. IROM-08-06, Sep. 2007.

[28] A. Kapoor and R. Greiner, "Learning and classifying under hard budgets," in *ECML 2005, 16th European Conference on Machine Learning*, 2005, pp. 170–181.

[29] X. Zhu and X. Wu, "Cost-constrained data acquisition for intelligent data preparation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 11, pp. 1542–1556, Nov. 2005.

[30] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.

[31] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, Inc., 1992.

[32] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.

Mining Frequent Item Sets Efficiently by Using Compression Techniques

Selim Mimaroglu, Cagri Cubukcu, Emin Aksehirli, and Ertunc Erdil
Department of Computer Engineering, Bahcesehir University, Ciragan Cad. 34353 Besiktas, Istanbul, Turkey

Abstract— *Mining frequent item sets in a data set is a significant problem of data mining that can only be solved in exponential time. For especially very large data sets, finding frequent item sets in practical run times is extremely important. A market basket data set can be represented by a set of bit vectors, which enables fast computation and low memory requirements. Space requirements can be reduced and better run time results can be obtained if bit vectors can be compressed, and binary operations can be applied on compressed bit vectors. In this paper, we study the advantages and disadvantages of using compression techniques for finding frequent item sets. Experimental evaluations clearly show that applying special purpose compression techniques has many benefits on a wide range of data sets.*

1. Introduction

Frequent item sets can be defined as subsets of items that appear together frequently in a data set, where frequency value is set by the user and is determined with regards to business, type of items, location and season.

Although identification of frequent item sets has been studied widely, it is still a major research topic because of its computationally expensive nature. Finding frequent item sets plays an essential role to disclose the hidden relations between items. Furthermore, frequent item set detection can be used in other data mining tasks like classification and clustering [1].

Apriori [2] is a seminal frequent item set detection algorithm which starts by scanning the data set to count each item and regards the items existing frequently enough as frequent 1-item sets. Like all the level-wise algorithms, Apriori uses frequent 1-item sets and rescans the data set for discovering frequent 2-item sets. Larger frequent item sets are discovered in level-wise manner, and the algorithm halts when all the frequent item sets are discovered. Scanning the data set at each iteration and generating the candidate item sets are the most important shortcoming of Apriori algorithm. Therefore, using Apriori on very large data sets is impractical.

FP-growth [3] algorithm mines frequent item sets without candidate generation. It also compresses the data set into a data structure called FP-tree, which generally fits into the main memory, for efficient scanning. FP-growth finds all the frequent item sets by searching the FP-tree, recursively.

ECLAT, which is another efficient frequent item set detection technique using vertical data set format, is introduced in [4]. Advantages of vertical representation of a data set is presented in this work as well.

Since finding all the frequent item sets is computationally challenging, there has been studies for finding a preponderant portion of the actual frequent item sets. A very good method which is based on agglomerative clustering of items for finding approximative frequent item sets is known as AFISA [5].

Our paper is organized as follows: Section 2 presents definitions and notations on frequent item sets and association rules. Section 3 shows some benefits of vertical data set format in binary. Special purpose binary compression techniques are presented in Section 4. Following, we present mining frequent item set process with compressed bit vectors in Section 5. Experimental test data sets and experimental results can be found in Section 6. Finally, we conclude with Section 7.

2. Frequent Item Set Detection and Association Rules

Let $I = \{I_1, I_2, \ldots, I_m\}$ be a set of items in data set D. A transactional data set consists of transactions $T = \{T_1, T2, \ldots, T_n\}$, where each T is a set of items such that $T \subseteq I$. Let A be a set of items which satisfies the condition $A \subseteq I$. For a transaction $T_i \in T$, if the condition $A \subseteq T_i$ is satisfied, then it is said that T_i contains A. Number of transactions that contains A is called $support\ count$ of A and is denoted with $support\_count(A)$. Frequency is determined by a user specified threshold called $minimum\ support\ count$, min_sup. A is said to be frequent k-item set, if and only if $|A| = k$, and $support\_count(A) \geq min\_sup$ hold.

As mentioned earlier, finding frequent item sets enables inferring relationships between set of items. Association rule analysis can be performed after detecting frequent item sets, and it is widely used in market basket data sets. Given two frequent item sets A and B, an association rule of the form $A \Rightarrow B$ indicates that customers who bought the item set A are likely to by the item set B. Of course, association rules are not limited to market basket data sets, wireless telecommunication companies can use association rules in their operational data sets for increasing quality of service

as well. The rule $A \Rightarrow B$ has the following confidence

$$confidence(A \Rightarrow B) = \frac{support\_count(A \cup B)}{support\_count(A)} \quad (1)$$

3. Mining Frequent Item Sets on Vertical Data Set Format

Most of the frequent item set detection algorithms operate on horizontal data sets as shown in Table 1, where each row represents a transaction and each transaction contains a set of items. It is possible to represent this data set in binary format as shown in Table 2. In binary representation 1 indicates the existence of the corresponding item in the corresponding transaction.

Table 1: A transactional Data Set in Horizontal Format

Transaction ID	Items
T_1	I_1,I_2,I_3
T_2	I_2, I_3
T_3	I_1, I_2, I_4
T_4	I_1, I_4
T_5	I_3, I_4, I_5
T_6	I_1, I_2, I_3, I_4

Table 2: Binary Representation of a Horizontal Transactional Data Set

Transaction ID	Items				
	I_1	I_2	I_3	I_4	I_5
T_1	1	1	1	0	0
T_2	0	1	1	0	0
T_3	1	1	0	1	0
T_4	1	0	0	1	0
T_5	0	0	1	1	1
T_6	1	1	1	1	0

Table 3: A vertical data set

Items	Transaction ID
I_1	T_1, T_3, T_4, T_6
I_2	T_1, T_2, T_3, T_6
I_3	T_1, T_2, T_5, T_6
I_4	I_3, I_4, T_5, T_6
I_5	T_5

Vertical data set format is introduced in many resources, such as [4]. Table 1 can be transformed into vertical format as shown in Table 3. In vertical data set format it is easier to obtain the occurrence information of an item, and perform set operations such as AND, OR, XOR. A vertical data set can be represented in binary as shown in Table 4. In this representation each item constitutes a bit vector, and bitwise set operations such as AND, OR, XOR can be performed. It should be noted that bitwise set operations are extremely fast, on CPUs with 64 bit architecture 64 bits are ANDed, ORed, or XORed at once. Most of the existing algorithms can be implemented on binary representation of vertical data set format to utilize the advantages of bitwise set operations.

Table 4: A vertical data set with binary representation

Transaction ID	Items				
	I_1	I_2	I_3	I_4	I_5
T_1	1	1	1	0	0
T_2	0	1	1	0	0
T_3	1	1	0	1	0
T_4	1	0	0	1	0
T_5	0	0	1	1	1
T_6	1	1	1	1	0

Unfortunately, sparse data sets waste a lot of space when represented by bit vectors. But, very good compression ratios can be obtained on sparse data sets since they mostly consist 0s. In order to save space and speed up operations we use compression techniques on bit vectors. Constantly compressing items for saving storage space and decompressing them for performing any operation is very costly. Therefore, we investigate and use compression techniques that allow bitwise set operations on the compressed bit vectors which avoids redundant and costly compression and decompression operations.

4. Compressing Binary Data Sets

For very large data sets bit vector representation of the data can be troublesome to fit into the main memory. Although bit vectors can be compressed, using general purpose compression schemes are not preferred because of costly compression and decompression overheads at each operation.

Special purpose compression techniques have been studied in data mining, although not very widespread. MAFIA [6] uses a technique that compresses the data adaptively according to its context. Compression is done on the fly with respect to a cost estimation method. VIPER [7] utilizes a compression technique called *skinning* which is based on Golomb encoding scheme and a novel candidate generation method.

Some other compression techniques are specifically designed to compress bit vectors while being able to apply bitwise set operations directly on the compressed forms-without the overhead of decompression. Even though such compression techniques may not compress the item bit vectors as good as the general purpose compression methods, their overall performance is higher by eliminating considerable overheads.

Byte-aligned Bitmap Coding (BBC) [8] is a Run Length Encoding (RLE) method that compresses the bit vectors in bytes. BBC considers bytes instead of bits since CPUs work more effectively on bytes than arbitrary number of bits. Word Aligned Hybrid (WAH) [9] compression algorithm is another RLE based technique that uses word-length atomic units for compression and utilizes modern CPU architectures. Enhanced Word Aligned Hybrid (EWAH) [10] tries to improve the compression ratio of WAH by sorting these

words. Position List Word Aligned Hybrid (PLWAH) [11] decreases the storage space requirement and improves the performance of WAH.

The compression technique to be utilized has a major effect on speed. Even though, in theory, any lossless compression technique can be used for compression, choices are limited when aim is the speed improvement. Since uncompressed bit map operations are very fast, it is possible that the operational advantages of compressed bit maps are overshadowed by the compression/decompression cost. In general, soft computing compression techniques are not deterministic in terms of time and efficiency, therefore they are not eligible for the tests. Aligned Bit Map Compression techniques offers a good balance between compression ratio and compression/decompression time.

4.1 Aligned Bit Map Compression

BBC, WAH and EWAH implement a special version of RLE which encodes an input sequence by representing each element with its occurrence. RLE scheme works best on sequences having lots of repetitions. For a sequence $S = 1,1,1,1,1,3,3,4,4,4,2,2$, basic RLE compression produces a compressed sequence $S_C = 5,1,2,3,3,4,2,2$. In S_C elements with odd indices represent the number of occurrences of their consecutive element. For sequence S, compression ratio of basic RLE is $\frac{|S_C|}{|S|} = \frac{8}{12}$.

Modern CPUs process the data in byte or word sized units. Thus, in most situations processing the whole word can be faster than processing just one bit. Byte aligned and word aligned compression methods exploit this behavior and instead of working directly on bits they group the bit sequences into bytes or word sized chunks. If two or more consecutive chunks are same, they will be replaced with a *fill byte/word* which consists of number of repetition and the repeated chunk. If a chunk is unique compared with its left and right chunks then it is added to the compressed sequence unmodified as a *literal byte/word*. More details on bit vector compression algorithms can be found in [12].

5. Mining Frequent Item Sets by Using Compressed Item Bit Vectors

We represent data sets in binary format where each item is a bit vector. Furthermore, we compress each item and operate directly on the compressed bit vectors. Using compressed form of the data set has the advantages of improved processing speed and reduced storage space. Bitwise operations on WAH compressed bit vectors are much faster than operations on uncompressed bit vectors if the compression level reaches to a certain level [9]. Similar results are reported on EWAH as well. We choose to use EWAH for compression since it is superior to WAH and its implementation is publicly available by its author.

As explained in detail in Section 4, RLE compression techniques looks for the repetitious patterns in the data and express them as a repetition number and the repeated string. Even though the repetitions in a data set, thus compression ratio, can not be known exactly in advance, a correlation exists between the density of the data set and the compression ratio.

If the data set is sparse, or dense, there is a high probability for existence of long sequences of 0's, or 1's. Long sequences are effectively compressed by RLE compression techniques, therefore, ratio of compression is much better on the dense or sparse data sets.

6. Experimental Results

We conducted experiments for mining all the frequent item sets by representing binary formatted vertical data sets with compressed bit vectors. Experiments are performed on a computer having Quadcore Intel Xeon CPU, 32GB main memory which runs on Linux operating system. Our choice of implementation language is Java, which provides built-in support for bit vectors and bitwise operations. For compressing item bit vectors, we used the Java implementation of EWAH which is obtained from http://code.google.com/p/javaewah.

Properties of test data sets are presented in Table 5. Compressed and original size of bit vectors can be found in Table 6, where compression is calculated as the ratio of output size to the input size as described in [13]-smaller values indicate better compression. Table 6 clearly shows that compression dramatically reduces the space requirement on all the data sets.

Our experimental evaluations show that by using compression, execution time of frequent item set detection process improves considerably on some data sets. Run time results are shown in Figures 1, 2, 3, and 4. Data sets are organized according to their number of transactions and items. On test data sets having compression ratio of 6.5×10^{-3} mining process works 4 to 6 times faster with compressed bit vectors, which is remarkable. On the data sets having compression ratio of 32×10^{-3} results are comparable. And, on the data sets having compression ratio of 65×10^{-3} compression looses its speed advantage. We noticed that although compression is very useful for saving space on all test data sets, it reduces run time considerably for data sets with compression ratio of 32×10^{-3} or better.

Mining frequent item sets by using compression techniques is very useful on data sets having good compression ratios, which can be obtained with dense or sparse data sets. These kind of data sets are common in telecommunication, retail, banking, and finance industries as well as on some data streams.

Table 5: Properties of Test Data Sets

Data Set Name	Transactions	Items	Transactions/Item (average)
100MT-100I-0.01	100M	100	10099
100MT-100I-0.05	100M	100	50486
100MT-100I-0.1	100M	100	100949
100MT-1KI-0.01	100M	1000	10099
100MT-1KI-0.05	100M	1000	50487
100MT-1KI-0.1	100M	1000	100948
10MT-100I-0.01	10M	100	1009
10MT-100I-0.05	10M	100	5048
10MT-100I-0.1	10M	100	10095
10MT-1KI-0.01	10M	1000	1009
10MT-1KI-0.05	10M	1000	5048
10MT-1KI-0.1	10M	1000	10094

Table 6: Compression Ratios of Test Data Sets

Data Set Name	Original size in bytes	Compressed size in bytes	Compression ratio
100MT-100I-0.01	12500000	80788	6.5×10^{-3}
100MT-100I-0.05	12500000	403884	32×10^{-3}
100MT-100I-0.1	12500000	807588	65×10^{-3}
100MT-1KI-0.01	12500000	80788	6.5×10^{-3}
100MT-1KI-0.05	12500000	403892	32×10^{-3}
100MT-1KI-0.1	12500000	807580	65×10^{-3}
10MT-100I-0.01	1250000	8068	6.5×10^{-3}
10MT-100I-0.05	1250000	40380	32×10^{-3}
10MT-100I-0.1	1250000	80756	65×10^{-3}
10MT-1KI-0.01	1250000	8068	6.5×10^{-3}
10MT-1KI-0.05	1250000	40380	32×10^{-3}
10MT-1KI-0.1	1250000	80748	65×10^{-3}

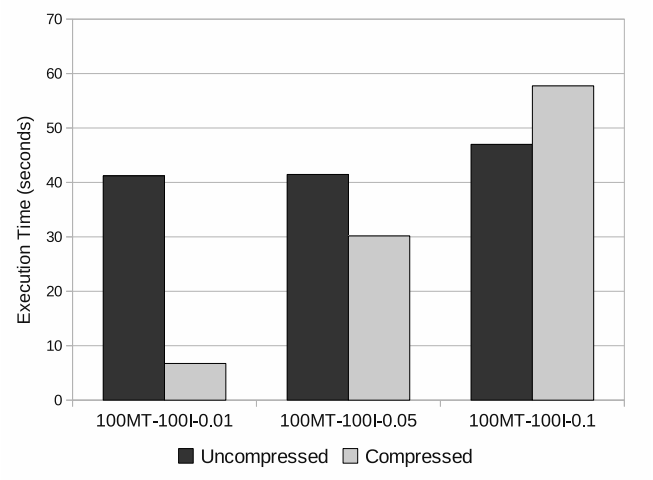


Fig. 1: Runtime results for the 100M-100I-0.01, 100M-100I-0.05, 100M-100I-0.1 data sets

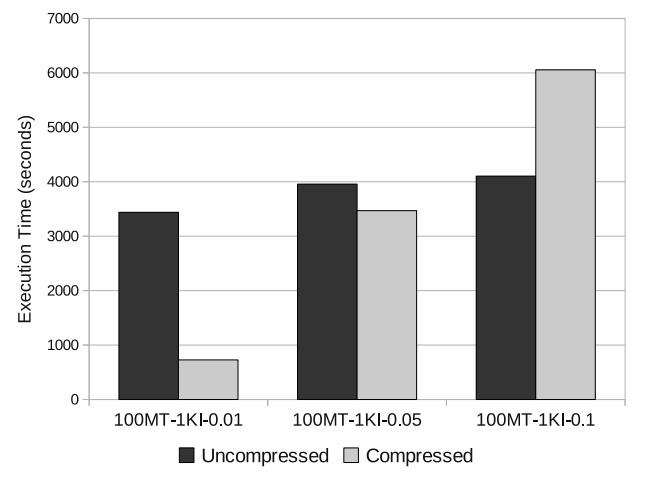


Fig. 2: Runtime results for the 100M-1KI-0.01, 100M-1KI-0.05, 100M-1KI-0.1 data sets

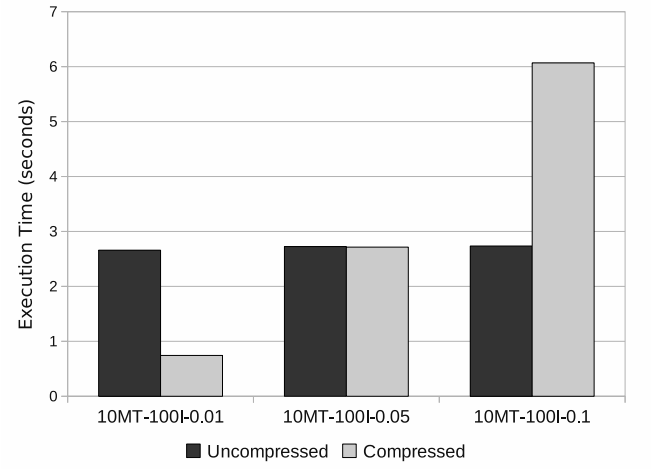


Fig. 3: Runtime results for the 10M-100I-0.01, 10M-100I-0.05, 10M-100I-0.1 data sets

7. Conclusion

We studied the benefits of using special compression techniques on binary data sets for mining frequent item sets which allow applying bitwise set operations directly on the compressed bit vectors. It is fair to say that compression is very useful for saving space. And, compression is very useful for obtaining better run times on data sets with good compression ratios. Our work can be widely applied in practice with very good results, since most of the data sets in telecommunication, retail, banking, and finance industries are very sparse with very good compression ratios.

References

[1] J. Han and M. Kamber, *Data mining: concepts and techniques*. Morgan Kaufmann, 2006.

[2] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*. IEEE, 2002, pp. 3–14.

[3] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," *Data mining and knowledge discovery*, vol. 8, no. 1, pp. 53–87, 2004.

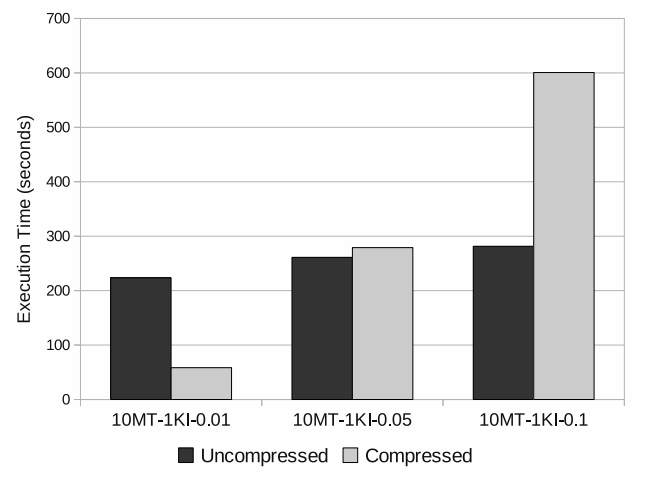


Fig. 4: Runtime results for the 10M-1KI-0.01, 10M-1KI-0.05, 10M-1KI-0.1 data sets

[4] M. Zaki, "Scalable algorithms for association mining," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 12, no. 3, pp. 372–390, 2002.

[5] S. Mimaroglu and D. Simovici, "Clustering and Approximate Identification of Frequent Item Sets," in *Proceedings of the Twentieth International Florida Artificial Intelligence Research Society Conference*, 2007, pp. 502–506.

[6] D. Burdick, M. Calimlim, J. Flannick, J. Gehrke, and T. Yiu, "MAFIA: a maximal frequent itemset algorithm," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 11, pp. 1490–1504, 2005.

[7] P. Shenoy, J. R. Haritsa, S. Sudarshan, G. Bhalotia, M. Bawa, and D. Shah, "Turbo-charging vertical mining of large databases," *ACM SIGMOD Record*, p. 22–33, 2000, ACM ID: 335376.

[8] G. Antoshenkov, "United states patent: 5363098 - byte aligned data compression," Nov. 1994. [Online]. Available: http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.htm&r=1&f=G&l=50&s1=5363098.PN.&OS=PN/5363098&RS=PN/5363098

[9] K. Wu, E. J. Otoo, and A. Shoshani, "Optimizing bitmap indices with efficient compression," *ACM Transactions on Database Systems (TODS)*, vol. 31, p. 1–38, Mar. 2006, ACM ID: 1132864.

[10] O. Kaser, D. Lemire, and K. Aouiche, "Histogram-aware sorting for enhanced word-aligned compression in bitmap indexes," *Proceeding of the ACM 11th international workshop on Data warehousing and OLAP*, p. 1–8, 2008, ACM ID: 1458434.

[11] F. Deliège and T. B. Pedersen, "Position list word aligned hybrid: optimizing space and performance for compressed bitmaps," *Proceedings of the 13th International Conference on Extending Database Technology*, p. 228–239, 2010, ACM ID: 1739071.

[12] K. Wu, E. Otoo, A. Shoshani, and H. Nordberg, "Notes on design and implementation of compressed bit vectors," Lawrence Berkeley National Laboratory, Technical Report LBNL/PUB-3161, Sept. 2001. [Online]. Available: http://crd.lbl.gov/~kewu/fastbit/publications.html

[13] K. Sayood, *Introduction to data compression*. Morgan Kaufmann, 2000.

Modeling functional outliers for high frequency time series forecasting with neural networks: an empirical evaluation for electricity load data

Nikolaos Kourentzes

Abstract— **This paper discusses and empirically evaluates alternative methodologies in modeling functional outliers for high frequency time series forecasting. In spite of several modeling and forecasting methodologies that have been proposed, there have been limited advancements in monitoring and automatically identifying outlying patterns and even less in modeling those for such times series. This is a significant gap considering the difficulty and the cost associated with manual exploration and treatment of such data, due to the vast number of observations. This study proposes and assesses the performance of different modeling methodologies focusing on two key aspects, the accuracy that the outliers are modeled and the impact of each methodology on modeling normal observations. The evaluated methodologies model functional outliers using binary, integer or trigonometric dummy variables, outlier profiles or isolate them into new time series and forecast them separately. Neural networks are employed to produce the forecasts, taking advantage of their flexible nature to accommodate the different methodologies and their superior performance in high frequency time series forecasting. Hourly electricity load data from the UK are used to empirically evaluate the performance of the different methodologies.**

Keywords: functional outliers, neural networks, multilayer perceptron, forecasting, electricity load.

I. Introduction

FORECASTS of electricity load data are required for a large variety of applications, such as trading electricity and scheduling production. In the forecasting literature such data are considered high frequency time series, where the data are collected and predicted in hourly or shorter time buckets. Although there is no strict definition of what constitutes a high frequency time series, in practice such time series are collected in daily or smaller time buckets and have vast amounts of data [1], introducing new issues in data handling, analysis and modeling. Use of conventional statistical modeling, designed for low frequency time series becomes problematic in these cases [2]. In the electricity load forecasting research several modeling methodologies for time series that exhibit these properties have been proposed [3], [4], [5], [6]; however, there have been limited advancements in both data monitoring and automatic outlier identification as well as modeling and automatic treatment of such outliers.

This is an important gap as time series models often require data cleaning, which involves modifying or removing outliers and obvious errors in the database [7], implicitly assuming that this information is a) available, which in fact requires costly manual collection and b) the analyst has a methodology to tackle outliers. Not cleaning the data can have substantial effects on model specification and parameters [7]. Outliers will introduce forecasting errors, as they do not follow the normal data generating process and they will also bias the model parameters, resulting in poor fit of the model to the data. In the literature, Taylor et al. acknowledges this issue and removes such days altogether [6], while Conejo et al. try to automatically correct outliers using conventional time series modeling approaches, but do not manage to improve the results [3], due to the high frequency nature of the data.

High frequency time series, and particularly electricity load data, typically exhibit periodic behavior. A new type of outlier appears in this family of time series. Whole periods may exhibit outlying behavior. For example electricity demand may be different throughout the day during a bank holiday in comparison to the normal demand profile. Such outliers can be analyzed as functional outliers [8]. In this case, we are interested in analyzing data providing information about curves, surfaces, etc as a whole varying over time. Such outliers can differ both in level, like normal outliers, but also in shape over the duration of a fixed period. To identify functional outliers there are different approaches, based on functional box- and bagplots [9], [10], time series clustering and classifications methodologies [11] belonging to a broader group of outlier detection research using unsupervised, supervised and semi-supervised learning algorithms, such as k-means, self-organizing maps, MLP networks, etc [12], [13], [14].

Once the outliers are known, one has to decide how to model them. In the case of electricity load forecasting, neural networks have shown good forecasting performance and is common to divide the time series into simpler ones and model those [15]. For instance, break the initial hourly time series into 24 new time series, one for each hour of the day. The functional outliers are now broken down to normal outliers, which can be modeled following conventional approaches [16]. However, [17] showed that this approach can lead to substantial loss of accuracy and increase the sensitivity to modeling decisions, concluding that it is preferable to forecast the complete time series, retaining all its dynamics. In this case one cannot avoid but model the functional outliers. In the context of time series forecasting

Nikolaos Kourentzes is with the Department of Management Science at Lancaster University Management School, Lancaster, LA1 4YX, United Kingdom. (email: n.kourentzes@lancaster.ac.uk).

there has been no focused research on the topic of modeling functional outliers for time series forecasting.

The contribution of this paper is to propose and evaluate a series of methodologies to model functional outliers on high frequency time series, specifically on hourly electricity load. These methodologies are based on both novel approaches and extensions of already existing conventional outlier modeling methods. In this paper the performance of each method is demonstrated and compared against a benchmark control model. This research concludes that using a trigonometric coding of the functional outliers results in the highest forecasting accuracy, while being robust to the stochasticity in the training of the neural networks.

The rest of this paper is organized as follows: Section II provides details of the proposed methodologies. In section III the experimental setup is described, while Section IV presents the empirical evaluation results. Section V concludes.

II. Methods

A. Multilayer Perceptrons for Time Series Forecasting

In this study multilayer perceptrons (MLP) are employed, which represent the most widely employed NN architecture [18], [19]. These have been well researched and have proven abilities in time series prediction and universal approximation [20]. They are able to approximate and generalize any linear or nonlinear functional relationship to any degree of accuracy without any prior assumptions about the underlying data generating process, providing a powerful forecasting method for linear or non-linear, non-parametric, data driven modeling [21], [22]. MLPs can be used to model time series in a univariate forecasting framework, using as inputs only time lagged observations of the time series, to predict the future values modeling nonlinear autoregressive NAR(p)-processes. Additional intervention variables and covariates can be used to capture additional information, modeling NARX(p)-processes. Data are presented to the network as disjunct vectors of a sliding window over the time series history. MLPs are organized in layers; the input layer, any number of hidden layers and the output layer that provides the predicted values for the time series. In forecasting applications one hidden layer is found to be adequate in most cases [23], which is also used here. The neural network learns the underlying data generating process by adjusting its connection weights $\mathbf{w} = (\beta, \gamma)$, minimizing an objective function on the training data, typically a squared error loss. Let $Y = (y_t)$ be the time series that needs to be predicted, with $t = (1, \ldots, T)$ observations and $X = (x_{tk})$ an array of $k = (1, \ldots, K)$ input variables, which can be lagged observations of either the time series or external variables. The predicted value $\hat{y}$ of the time series, one step ahead from time t using single hidden layer MLPs is:

$$\hat{y}_{t+1} = \beta_0 + \sum_{h=1}^{H} \beta_h g\left(\gamma_{h0} + \sum_{k=1}^{K} \gamma_{hk} x_{tk}\right), \qquad (1)$$

where $\beta = (\beta_1, \ldots, \beta_H)$, $\gamma = (\gamma_{11}, \ldots, \gamma_{HK})$ are the weights for the output and the hidden layer respectively. The β_0 and γ_{h0} are the biases of each neuron. The hidden nodes use a nonlinear transfer function $g(\cdot)$, which is usually either the sigmoid logistic or the hyperbolic tangent function. The modeler must choose the appropriate data pre-processing, the number of hidden nodes, the transfer function within nodes, the training algorithm and the cost function of the MLP. An adequate MLP architecture is routinely determined by running simulations on the time series; a set of candidate MLPs is trained using different architectural parameters and the architecture which shows the lowest in-sample error is selected. We provide further details in section III, where the experimental setup is discussed.

B. Methodologies for Modeling Functional Outliers

While conventional outliers are classified as additive or innovative outliers, requiring particular modeling in each case [7], functional outliers are not distinguished in separate classes [9]. In this study different alternative methodologies to model functional outliers are proposed. These are based on extensions of conventional outlier modeling or novel approaches, taking advantage of the unique nature of functional outliers in the context of time series forecasting. For all these methodologies it is assumed that the data generating process of normal observations is captured adequately and the outliers are already labeled as such.

1) Single Binary Dummy Variable: In conventional linear regression modeling persisting effects on the level of a time series or additive outliers can be captured by using a single indicator dummy variable I. Given a time series y_t following a process z_t such an event can be modeled as:

$$y_t = z_t + \omega I[t = \tau], \qquad (2)$$

where ω is the size of the shift or the additive outlier and $I[t = \tau]$ is the dummy variable taking a value of 1 when $t = \tau$, i.e. there is an outlier, and a value of zero otherwise. The process z_t is uncontaminated by outliers, but unobserved [24]. Although linear regression, as it is apparent from (2) always shifts z_t by the same amount ω, MLPs were shown to be able to output for the same binary indicator variable several different values for y_t [25]. Based on this finding, a MLP should be able capture the shape of a functional outlier using a single binary dummy variable that is equal to 1 when it is occurring and zero otherwise.

2) Multiple Binary Dummy Variables: Given that a functional outlier lasts for several observations S, one can employ several binary dummy variables to code each of its observations $s = 1, \ldots, S$ with a different shift from the unobserved underlying process z_t equal to ω_s. In the linear regression context one would be required to use S different binary variables, where each I_s would be equal to one if the observation is the s^{th} value of a functional outlier and zero otherwise. If a constant term is assumed outside of the process z_t then $S-1$ binary dummies can be used instead. Assuming that the differences of the functional outlier and

the normal observations are not significant for each s in S one could remove the corresponding indicator binary dummies for these periods, thus reducing the degrees of freedom of the model and simplifying its estimation. This process can be automated through the use of stepwise regression, where insignificant indicator variables are automatically dropped from the final model. In the context of neural networks the same principles can be applied directly, however there is higher incentive to remove insignificant indicators from the model, given that each variable increases the degrees of freedom of the model by H, i.e. the number of the hidden nodes. Also, due to the multiple bias terms $\gamma_{hk'}$ it is not straightforward to decide a-priori whether all S or $S-1$ binary dummies should be considered, where k' refers to the indicator of the binary dummy input variables.

3) Single Integer Dummy Variable: Capitalizing on the nonlinear mapping capabilities of neural networks one can use a single integer dummy variable to code the functional outliers. Such variable increases monotonically from 1 to S when there is a functional outlier and is zero otherwise. This coding resembles a sawtooth waveform. A similar technique has been employed to model deterministic seasonality with MLPs [19], [26]. The authors show that following this approach neural networks are able to capture deterministic seasonality in time series, although this approach underperforms in comparison to other methodologies. However, a key advantage of this approach is the minimum increase in the model's degrees of freedom, making it easier to train the MLPs.

4) Profile Dummy Variable: Instead of letting the neural network identify the nonlinear mapping between the integer dummy discussed above and the deviations of the functional outlier from z_t one could assist the network by providing an archetypal pattern that the functional outliers follow. Thus, by estimating the average profile of the functional outliers a dummy variable is constructed that is equal to the profile when there is an outlier and zero otherwise.

5) Trigonometric Dummy Variables: In modeling repeating patterns instead of using multiple indicator variables one can equivalently use trigonometric variables. This has been widely used in modeling seasonal time series with regression models, particularly for the case of deterministic seasonality [27]. When using MLPs it has been shown that due to their approximation capabilities only a pair of trigonometric variables (a single sine and a single cosine) can be used with minimal loss of fitting and predictive accuracy [19], [26]; yet reducing the additional degrees of freedom to only two for any longer seasonal periodicity. A analogous approach can be used to code functional outliers. The network is given a pair of sine and cosine with wavelength S, i.e. the duration of the functional outlier, when one is occurring and zero in other cases.

6) Model Separately: Instead of providing additional information and indicator variables to the MLP regarding the functional outliers, one can separate all the outliers in a new time series and replace those in the original time series with normal observations, based on the approximated process z_t. Although the modeling of the original time series is greatly simplified as there as no outliers, two separate time series have to be predicted. The newly constructed time series, containing all the functional outliers, can be relatively short in comparison to the original time series, leading to estimation issues. Once both time series are forecasted, the predicted functional outliers are replaced on the predicted series of z_t, resulting in a single series that both outlying and normal observations are predicted.

III. Experimental Design

A. Data

To assess the performance of the aforementioned functional outlier modeling methodologies electricity load time series data from the UK are used. These are sampled at hourly intervals, from the 1^{st} of April 2001 01:00 until the 1^{st} of November 2008 01:00, amounting to 66505 hourly observations or 2771 days. Figure 1 provides a plot of the first 3000 hourly observations. The time series exhibits triple seasonality, a daily, a weekly and an annual pattern.

The time series contains two leap years, 2004 and 2008, distorting the annual periodicity. The load profile of each day exhibits three distinct patterns, associated to winter, summer and transitional consumption profiles. This is demonstrated in figure 2, where data only for Thursday's are plotted. To avoid cluttering the figure, transitional profiles are plotted together with summer days. Finally, UK uses daylight saving, translated into one moving date day in either March or April having 23 hours and another moving date day in October having 25 hours every year.

Using the methodology outlined in [14] 63 functional outliers (or 1512 hourly observations) are identified, reflecting unusual electricity load profiles. These are illustrated in figure 3. Using these outliers the different methodologies discussed in section II will be applied and evaluated. A large number of these outliers can be explained by calendar events, such as bank holidays and are not connected to exogenous variables such as temperature, which is not used in this study.

The time series is split into three subsets for training the MLPs and evaluating their performance. The test set is years

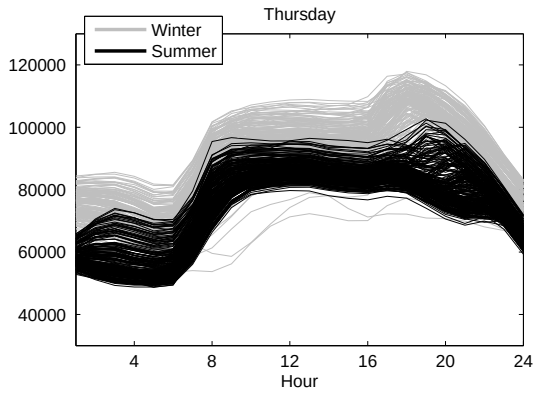


Fig. 2. Summer and winter consumption profiles for Thursday.

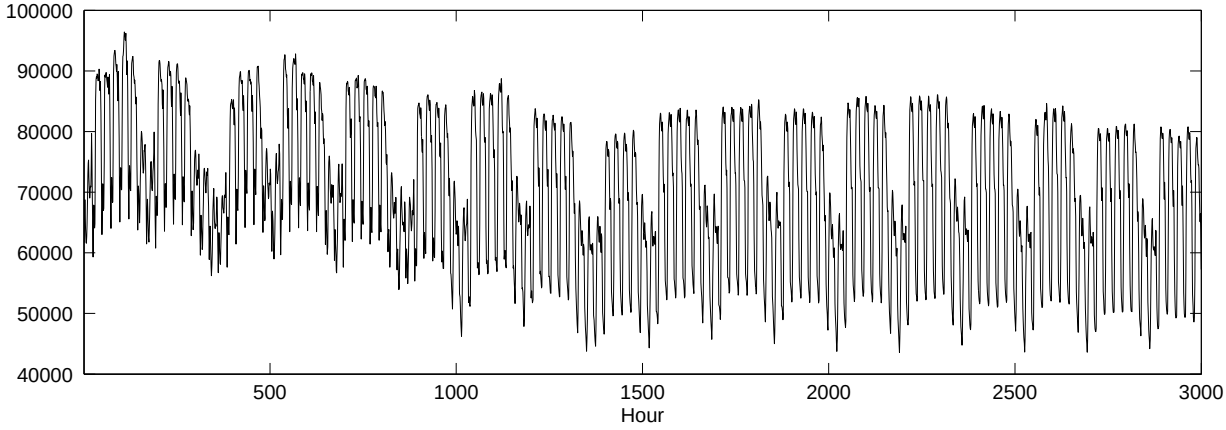


Fig. 1. Plot of the first 2880 observations of the time series.

2007 and 2008 (16080 observations). The validation set has equal number of observations and the training set is the remaining part of the time series.

B. Methods

A single MLP setup is used to model the time series under all methodologies. All model parameters are kept fixed with the exception of the input variables that change per methodology. The network has a single hidden layer. The number of hidden nodes is identified using a grid search from 5 to 40 hidden nodes with a step of 5. The search is stopped at 40 due to computational resources restrictions, as the network is trained using 34,345 observations. Using 40 nodes provides the best performance. All hidden nodes use the hyperbolic tangent function. The lagged observations of the time series are linearly scaled between $[-0.5, 0.5]$, before being inputted to the network. The networks are trained using the Levenberg-Marquardt algorithm, which requires setting the μ_{LM} and its increase and decrease steps. Here $\mu_{LM} = 10^{-3}$, with an increase step of $\mu_{inc} = 10$ and a decrease step of $\mu_{dec} = 10^{-1}$. For a detailed description of the algorithm and the parameters see [28]. The maximum training epochs are set to 1000. Mean squared error is used as a training cost function and is recorded for both training and validation sets. The training can stop earlier if μ_{LM} becomes equal of greater than $\mu_{max} = 10^{10}$ or the validation error increases for more than 25 epochs. This is done to avoid over-fitting. When training is stopped the network weights that give the lowest validation error are used. The limit of a 1000 training epochs is not reached in any of the simulations, due to the early stopping criterion. Each MLP is initialized 30 times with randomized starting weights to accommodate the nonlinear optimization. Once training is finished, the network initialization that exhibits the lowest error on validation set is chosen.

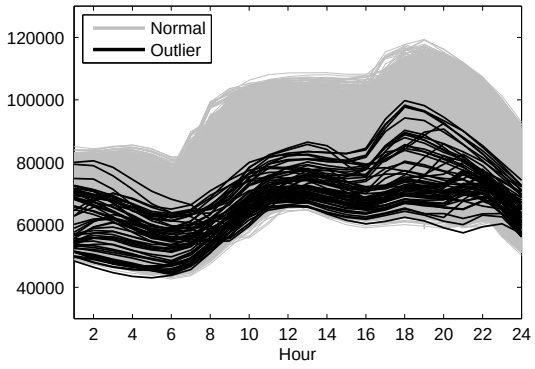


Fig. 3. Identified functional outliers.

In total, ten different sets of inputs are considered. First a $Control$ set of autoregressive lagged inputs is identified using stepwise regression. The high frequency nature of the time series (hourly observations) makes it challenging to automatically choose the relevant input variables, as most of the automatic input identification methodologies have been developed for lower frequency time series and either are associated with prohibitive computational cost or will not produce valid results [2], [19]. In [19] and [29] an extensive empirical evaluation of alternative input variables selection methodologies, on low and high frequency time series, showed that stepwise regression is a robust method to select variables for MLPs, superior to different forms of autocorrelation and partial autocorrelation analysis, selection by mutual information criterion, spectral analysis and random field regression. Based on this finding, regression is used in this study to identify the relevant autoregressive inputs. The resulting $Control$ set of input variables uses 36 lags, $X_{Control} = \{y_{t-1}, y_{t-2}, y_{t-3}, y_{t-5}, y_{t-6}, y_{t-8}, y_{t-9}, y_{t-10}, y_{t-11}, y_{t-12}, y_{t-13}, y_{t-15}, y_{t-16}, y_{t-17}, y_{t-19}, y_{t-20}, y_{t-21}, y_{t-22}, y_{t-23}, y_{t-24}, y_{t-48}, y_{t-72}, y_{t-120}, y_{t-144}, y_{t-164}, y_{t-165}, y_{t-166}, y_{t-167}, y_{t-168}, y_{t-169}, y_{t-170}, y_{t-171}, y_{t-172}, y_{t-8736}, y_{t-8760}, y_{t-8784}\}$. For this set of inputs the outliers are not modeled and will be used as a benchmark in assessing how much the proposed methodologies increase the accuracy of predicting both functional outliers and normal observations.

The first methodology employs a single binary dummy variable as described in section II and is named $Binary(1)$. The second methodology uses multiple binary dummies and four variants are created; $Binary(S)$ uses an equal number of dummy variables to the length of the functional outliers, in this case 24, while $Binary(S-1)$ uses one less. $Binary(Step)$ and $Binary(Back)$ use stepwise and backward regression to identify the number of useful binary dummy variables, starting from $Binary(S)$, resulting in 4 and 7 inputs respectively. Note that in this case stepwise and forward regression resulted in the same selection of variables; hence only one is evaluated here. The next methodology uses a single integer dummy variable and is named $Integer$, followed by $Profile$ that codes an archetypal functional outlier profile in a single dummy variable, which is scaled between -1 and 1. Methodology $SinCos$ uses a pair of trigonometric dummy variables. Finally the time series is separated into two, one for normal observations and one for outliers. To replace the outliers in the normal time series the following technique is used. First, the seasonal component of the time series is identified, through time series decomposition. Then the neighboring four seasons (of the highest frequency, daily in this case), two before and two after the functional outlier if available, are averaged to construct a local average profile, which is used to recreate a set of normal observations to replace the outlier. Replacing the outliers instead of removing them allows retaining the dynamics of the time series. A second time series is created and forecasted separately from all functional outliers. The forecasts of the latter series are used to replace the forecasts of the constructed local average profiles. The methodology is named $Replace$ and is the most complex, requiring to forecast two separate time series. To identify the input variables for the outlier time series stepwise regression is used, resulting in the following inputs: $X_{Replace} = \{y_{t-1}, y_{t-2}, y_{t-15}, y_{t-23}, y_{t-24}, y_{t-166}, y_{t-168}, y_{t-169}, y_{t-172}\}$. Note that all these methodologies use $Control$ to capture the underlying structure of the time series. Table I summarizes the described methodologies.

TABLE I
FUNCTIONAL OUTLIER MODELING METHODOLOGIES.

Methodology	No. of Inputs
$Control$	36
$Binary(1)$	37
$Binary(S)$	60
$Binary(S-1)$	59
$Binary(Step)$	40
$Binary(Back)$	43
$Integer$	37
$Profile$	37
$SinCos$	38
$Replace$	36, 9

C. Experimental Setup

The different methodologies are evaluated on forecasting the next 24 hours, considering the aggregate error from $t+1$ to $t+24$. Rolling origin evaluation is used, i.e. from each observation a trace of 24 consecutive forecasts is produced. This evaluation methodology has several advantages over the commonly employed fixed origin, where only a single out-of-sample measurement is done, collecting several error measurements, thus providing a richer and more reliable distribution of errors. For a discussion of rolling origin evaluation and its advantages see [30]. The forecasting accuracy is measured in Mean Absolute Percentage Error (MAPE) that is $MAPE = \sum_{t=1}^{h} (|y_t - f_t|/y_t)$, where y_t is the actual and f_t is the forecast at time t and h is the forecast horizon. MAPE is preferred to the commonly used MSE being more robust [31], but also due to its relevance to practice. Due to the high positive values of the time series none of the key issues of this metric are relevant here, while enjoying its very intuitive interpretation. For a detailed discussion on the choice of error measures for forecasting purposes see [30]. Once the MAPE has been calculated for each forecast origin it is average across origins to produce a single figure for each training, validation and test subsets. Furthermore, the MAPE is measured across all types of observations, only normal ones and only outlying ones. This allows tracking the performance of each methodology in improving forecasting accuracy in either normal or outlying observations.

IV. RESULTS

Table II provides the MAPE results for the best MLP training initialization for each methodology and in brackets the mean MLP across all initializations. The results are broken down in three categories, All, $Normal$ and $Outlier$ showing the errors filtered by the type of observation. The lowest error by column is highlighted in boldface. Any results worse than the benchmark $Control$ are underlined.

We will focus on the results after initialisation selection. First the results across all observations are analyzed. All methods outperform significantly the $Control$ in both training and validation sets. The $Replace$ methodology ranks first, while the remaining approaches follow with small differences among themselves. In the test set $SinCos$ is performing the best, closely followed by $Replace$. Note that $Binary(1)$ and $Binary(Step)$ fail to outperform the benchmark $Control$, pointing to over-fitting in the training set and poor generalization. It can also be observed that using a large number of binary dummy variables is preferable, with $Binary(S)$ being more accurate than $Binary(S-1)$, followed by $Binary(Back)$ and lastly by $Binary(Step)$. Both $Integer$ and $Profile$ that use only one additional, non-binary, dummy variable, perform better than the benchmark, with $Profile$ being best.

Looking at the performance only across normal observations we can see the impact of modeling adequately the outliers on the model coefficients and consequently on how well each methodology fits and predicts the time series. The $Replace$ methodology significantly outperforms all other approaches across all training, validation and test sets. In training and validation sets all methods perform better than the $Control$ benchmark, however once the test set is considered, all but $Replace$, $SinCos$ and $Profile$

TABLE II
MAPE RESULTS

Methodology	Trn	Val	Tst
All			
Control	1.83% (1.92%)	1.91% (1.98%)	1.92% (2.10%)
Binary(1)	1.72% (1.74%)	1.73% (1.83%)	1.93% (1.97%)
Binary(S)	1.60% (**1.70%**)	1.71% (**1.83%**)	1.86% (**1.96%**)
Binary(S-1)	1.64% (1.78%)	1.73% (1.90%)	1.86% (2.06%)
Binary(Step)	1.75% (1.89%)	1.80% (1.96%)	1.96% (2.10%)
Binary(Back)	1.73% (1.85%)	1.80% (1.93%)	1.89% (2.07%)
Integer	1.66% (1.74%)	1.71% (1.85%)	1.91% (1.97%)
Profile	1.74% (1.76%)	1.77% (1.86%)	1.86% (1.99%)
SinCos	1.70% (1.88%)	1.73% (1.93%)	**1.80%** (2.06%)
Replace	**1.49%** (1.81%)	**1.70%** (2.05%)	1.82% (2.08%)
Normal			
Control	1.70% (1.76%)	1.78% (1.86%)	1.78% (1.96%)
Binary(1)	1.68% (1.70%)	1.68% (1.78%)	1.88% (1.92%)
Binary(S)	1.59% (1.67%)	1.67% (1.77%)	1.82% (1.91%)
Binary(S-1)	1.62% (1.74%)	1.68% (1.83%)	1.82% (2.01%)
Binary(Step)	1.66% (1.76%)	1.72% (1.86%)	1.87% (1.98%
Binary(Back)	1.64% (1.75%)	1.73% (1.85%)	1.82% (1.98%
Integer	1.62% (1.68%)	1.66% (1.77%)	1.88% (1.91%
Profile	1.66% (1.70%)	1.68% (1.79%)	1.77% (1.93%
SinCos	1.65% (1.80%)	1.67% (1.85%)	1.75% (1.98%
Replace	**1.37%** (**1.57%**)	**1.59%** (**1.72%**)	**1.69%** (**1.81%**
Outlier			
Control	7.76% (8.78%)	7.75% (7.43%)	8.75% (9.01%
Binary(1)	3.21% (3.53%)	3.78% (**4.00%**)	4.51% (**4.18%**
Binary(S)	**1.98%** (**3.09%**)	**3.51%** (4.38%)	3.78% (4.43%
Binary(S-1)	2.38% (3.47%)	4.03% (4.79%)	3.77% (4.89%
Binary(Step)	5.76% (7.23%)	5.49% (6.13%)	6.17% (7.63%
Binary(Back)	5.52% (6.18%)	5.20% (5.49%)	5.65% (6.32%
Integer	3.54% (4.39%)	4.14% (5.10%)	**3.72%** (4.91%
Profile	4.97% (4.28%)	5.74% (4.93%)	6.16% (4.91%
SinCos	4.04% (5.11%)	4.33% (5.18%)	4.31% (5.63%
Replace	6.52% (11.40%)	6.53% (17.60%)	8.37% (15.50%

Best MLP initialization error. Values in bracket are mean MAPE over a initializations. The lowest error in each column is in boldface.

have higher errors in comparison to $Control$ demonstratir again lack of generalization. Similar to the ranking for a observations, $SinCos$ is marginally better than $Profil$ Considering the accuracy of the methods only for the funct onal outliers all methods outperform the benchmark signi cantly. $Binary(1)$, $Binary(S)$, $Binary(S-1)$, $Intege$ and $SinCos$ perform very well with errors in all se lower than 5%, while $Control$'s errors are around 8%. I sample $Binary(S)$ performs the best, however we can see significant degradation of accuracy between the training ar the validation and tests sets, implying potential over-fittin Note that this model has the highest degrees of freedor as it can be seen in table I. The same behavior to a less extend can be observed for $Binary(S-1)$ and $Binary(1$ $Integer$ gives the lowest test set error, demonstrating th the MLP can accurately map the functional outlier profi using a simple sawtooth waveform input, while there no evidence of over-fitting. $SinCos$ results in marginal higher errors, again with no evidence of over-fitting. $Profi$ has mediocre performance that degrades to the test set ar $Replace$, though still better than the $Control$ benchmar has poor performance. The latter can be explained by tl relatively small size of the outlier time series (1512 hour observations) and how erratic these are.

Across both normal values and outliers $SinCos$ has consistently good performance resulting in a superior overall accuracy. $Replace$ is the best method to predict the normal observations, but fails to improve the accuracy of outliers significantly. Due to the high number of normal observations (more than 97% of all values), its poor performance on outliers is masked when considering the aggregate accuracy across all observations, however it should be avoided, unless the objective is to focus only on the normal observations. $Profile$ performs in both cases better than $Control$, but worse than $SinCos$, while all other methods should be avoided as they lead to inferior performance in predicting the normal observation in comparison to the benchmark $Control$.

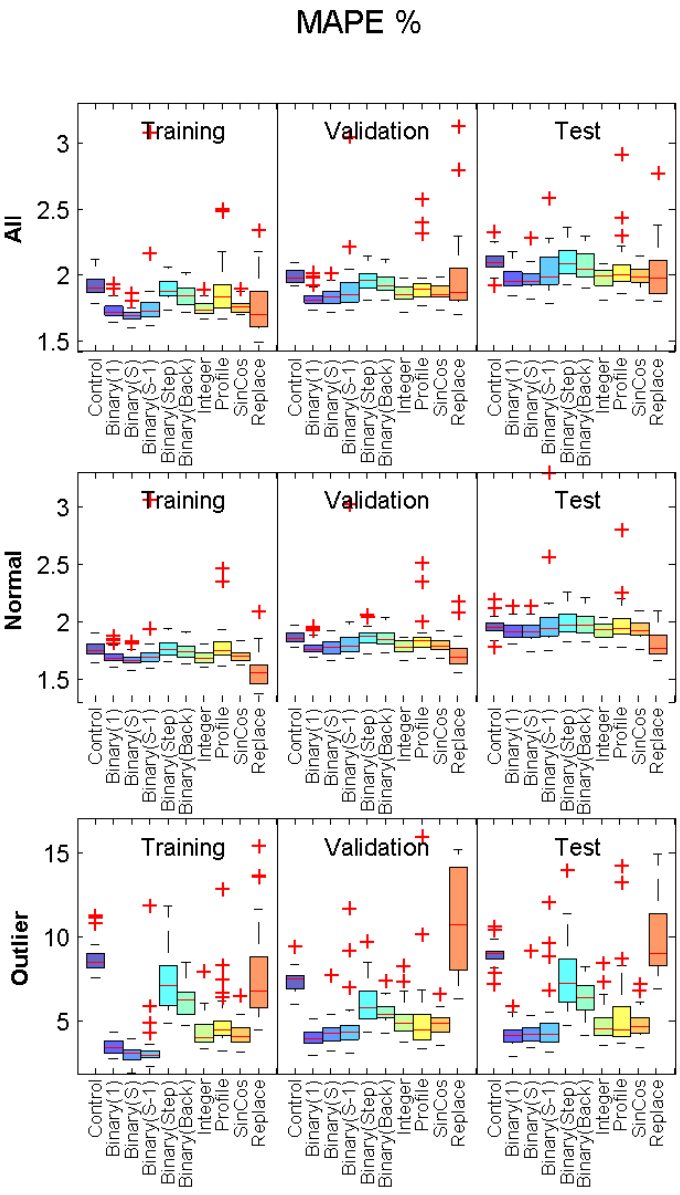


Fig. 4. Summer and winter consumption profiles for Thursday.

Figure 4 provides MAPE errors for each method across all initializations. This allows us to evaluate the stability and robustness of each method, reflecting the mean error provided in table II. The rankings of the models are in agreement with table II, though it is easier to assess the significance of the differences. Considering the results across all observations (1^{st} row in figure 4) we can see that the main body of the distributions of all methods but $Binary(Step)$ and $Binary(Back)$ and in some cases $Binary(S-1)$ and $Replace$, are well below the distribution of $Control$, indicating significant differences. Furthermore, across errors for normal and outlying observations the contrast in the performance of $Replace$ is clearly shown, as well as its wide distribution, implying more variability in the results, i.e. less robustness, in contrast to the other methods. On the other hand, note that with the exception of $Binary(Step)$, $Binary(Back)$ and $Replace$ the other methods result in tight distributions, meaning that the different training initializations resulted in similar accuracy, i.e. the methods are not sensitive to the initial training weights of the MLP.

Overall, $SinCos$ is the best compromise in performance across both normal and outlying observations, while $Replace$ has significantly superior accuracy in modeling the normal values.

V. Conclusions

In this paper alternative methodologies to model functional outliers with neural networks in high frequency time series are proposed in the context of electricity load forecast. A novel trigonometric coding of the outliers performs the best, improving the accuracy of both normal and outlying observations. Depending on the modeler's objective other approaches may perform well specifically in improving the performance of the network for normal values or outliers.

References

[1] R. F. Engle, "The econometrics of ultra-high-frequency data," *Econometrica*, vol. 68, no. 1, pp. 1–22, 2000.

[2] C. W. J. Granger, "Extracting information from mega-panels and high-frequency data," *Statistica Neerlandica*, vol. 52, no. 3, pp. 258–272, 1998.

[3] A. J. Conejo, J. Contreras, R. Espinola, and M. A. Plazas, "Forecasting electricity prices for a day-ahead pool-based electric energy market," *International Journal of Forecasting*, vol. 21, no. 3, pp. 435–462, 2005.

[4] H. Hahn, S. Meyer-Nieberg, and S. Pickl, "Electric load forecasting methods: Tools for decision making," in *International Conference on Information Systems, Logistics and Supply Chain*, vol. 199, Lyon, France, 2009, pp. 902–907.

[5] J. R. Trapero and D. J. Pedregal, "Frequency domain methods applied to forecasting electricity markets," *Energy Economics*, vol. 31, no. 5, pp. 727–735, September 2009.

[6] J. W. Taylor, L. M. de Menezes, and P. E. McSharry, "A comparison of univariate methods for forecasting electricity demand up to a day ahead," *International Journal of Forecasting*, vol. 22, no. 1, pp. 1–16, 2006.

[7] C. Chatfield, *The Analysis of Time Series: An Introduction*, 6th ed. Chapman & Hall/CRC, 2004.

[8] J. O. Ramsay and B. W. Silverman, *Functional Data Analysis*. Springer Science+Business Media Inc., 2002.

[9] R. J. Hyndman and H. L. Shang, "Rainbow plots, bagplots and boxplots for functional data," Monash University, Department of Econometrics and Business Statistics, Monash Econometrics and Business Statistics Working Papers 9/08, Nov. 2008.

[10] Y. Sun and M. G. Genton, "Functional boxplots," *Journal of Computational and Graphical Statistics*, vol. to appear, 2011.

[11] P. K. Chan and M. V. Mahoney, "Modeling multiple time series for anomaly detection," in *Proceedings of the Fifth IEEE International Conference on Data Mining*, ser. ICDM '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 90–97.

[12] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *SIGMOD Rec.*, vol. 29, pp. 427–438, May 2000.

[13] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, pp. 85–126, October 2004.

[14] N. Kourentzes and S. F. Crone, "Semi-supervised monitoring of electric load time series for unusual patterns," in *Proceedings of the 2011 International Joint Conference on Neural Networks*, ser. IJCNN 2011, Forthcoming.

[15] H. S. Hippert, D. W. Bunn, and R. C. Souza, "Large neural networks for electricity load forecasting: Are they overfitted?" *International Journal of Forecasting*, vol. 21, no. 3, pp. 425–434–, 2005.

[16] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*. New Jersey: Prentice Hall Inc., 1994, vol. 3rd.

[17] S. F. Crone and N. Kourentzes, "Segmenting electrical load time series for forecasting? an empirical evaluation of daily uk load patterns," in *Proceedings of the 2011 International Joint Conference on Neural Networks*, Forthcoming.

[18] G. Q. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[19] N. Kourentzes, "Input variable selection for time series forecasting with artificial neural networks an empirical evaluation across varying time series frequencies," Ph.D. dissertation, Department of Management Science, Lancaster University, 2009.

[20] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[21] G. P. Zhang, "An investigation of neural networks for linear time-series forecasting," *Computers and Operations Research*, vol. 28, no. 12, pp. 1183–1202, 2001.

[22] G. P. Zhang, B. E. Patuwo, and M. Y. Hu, "A simulation study of artificial neural networks for nonlinear time-series forecasting," *Computers and Operations Research*, vol. 28, no. 4, pp. 381–396, 2001.

[23] G. P. Zhang, "Neural networks for classification: a survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 30, no. 4, pp. 451–462, Nov 2000.

[24] P. H. Franses and van Dijk D., *Non-linear time series models in empirical finance*. Cambrid, 2006.

[25] N. Kourentzes and S. F. Crone, "Inference for neural network predictive models with impulse interventions," in *Proceedings of the 2010 International Conference on Data Mining*, ser. DMIN10, Las Vegas, USA, July 2010.

[26] S. F. Crone and N. Kourentzes, "Forecasting seasonal time series with multilayer perceptrons an empirical evaluation of input vector specifications for deterministic seasonality," in *Proceedings of the 2009 International Conference on Data Mining*, ser. DMIN09, Las Vegas, USA, July 2009, pp. 232–238.

[27] E. Ghysels and D. R. Osborn, *The econometric analysis of seasonal time series*. Cambridge: Cambridge University Press, 2001.

[28] M. Hagan, H. Demuth, and M. Beale, *Neural Network Design*. Boston: PWS Publishing, 1996.

[29] S. F. Crone and N. Kourentzes, "Input-variable specification for neural networks - an analysis of forecasting low and high time series frequency," in *Proceedings of the International Joint Conference on Neural Networks*, ser. IJCNN'09, Atlanta, USA, 2009, pp. 3221–3228.

[30] L. Tashman, "Out-of-sample tests of forecasting accuracy: An analysis and review," *International Journal of Forecasting*, vol. 16, pp. 437–450, 2000.

[31] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, pp. 679–688, 2006.

Feature Selection with Hybrid Mutual Information and Genetic Algorithm

Vahid Chahkandi*, Mehrdad Jalali, Mahsa Mirshahi, Ali Hosseini

Abstract— **Feature selection plays an important role in data mining and pattern recognition, especially for large scale data. During past years, various metrics have been proposed to measure the relevance between different features. Mutual information is nonlinear and can effectively represent the dependencies of features. In this paper, we proposed a combinatorial algorithm that uses the powerful metric mutual information and genetic algorithm. In this method, relevant features search with genetic algorithm by mutual information as fitness function. Totally, this method has better results than original mutual information in more situation and some of them the result of Hybrid Mutual Information and Genetic Algorithm[1] is equal with original mutual information.**

I. INTRODUCTION

High-dimensional datasets present many mathematical challenges as well as some opportunities, and are bound to give rise to new theoretical developments [1]. One of the problems with high-dimensional datasets is that, in many cases, not all the measured variables are important for understanding the underlying phenomena of interest. While certain computationally expensive novel methods [2] can construct predictive models with high accuracy from high-dimensional data, it is still of interest in many applications to reduce the dimension of the original data prior to any modeling of the data.

Ref.[3], reviewed traditional and current state-of-the-art dimension reduction methods that published in the statistics, signal processing and machine learning literature. Feature transform (or feature extraction) constructs new features by projecting the original feature space to a lower dimensional one. Principal component analysis and independent component analysis are two widely used feature transform methods. Although feature transform can obtain the least dimension, its major drawbacks lie in that its computational over- head is high and the output is hard to be interpreted for users.

Ref.[4], introduced a general criterion function about mutual information in feature selector, which can bring most information measurements in previous algorithms together. And another purpose of that, to propose a new feature selection algorithm based on dynamic mutual information, which is only estimated on unlabeled instances.

Feature selection (or variable selection) selecting a subset of relevant features for building robust learning models. Roughly speaking, there are three kinds of feature selection methods [5–7], i.e., wrapper, filter and embedded methods. In the embedded model, feature selection is integrated into the process of training for a given learning algorithm. One of the typical embedded methods is C4.5 [8]. Wrappers choose those features with high prediction performance estimated by specified learning algorithms. Since taking prediction capability into consideration, wrappers can achieve better results than others. Unfortunately, wrapper methods are less general and need more computational resources in learning, because they are tightly coupled with specified learning algorithms. Consequently, they are often intractable for large scale problems.

This paper is organized as follows. Section II introduces Mutual Information. Section III introduces Genetic Algorithm and after that in Section 4 we discuss on hybrid Mutual Information and Genetic Algorithm (HMIGA). Then in Section V shows Experimental results. Finally, conclusion are given in Section VI.

II. MUTUAL INFORMATION

A number of selection criteria, such as correlation coefficient and least square regression error, are available for the filter mechanism for feature selection. In our study, the mutual information (MI) [9] is chosen as the selection criterion because MI is capable of measuring a general dependence between two features without assuming the distributions of the features, and case based reasoning requires no assumption on the different project features to derive the solutions [10]. In addition, the MI function can be applied to both numerical and categorical features.

2.1. Entropy and mutual information

In feature selection problem, the relevant features have important information regarding the output, whereas the irrelevant features contain little information regarding the output. The objective of feature selection is to find those features that contain as much information about the output as possible. For this purpose, Shannon's information theory [9] provides a feasible way to measure the information of random variables with entropy and mutual information.

*V. Chahkandi is a student of Department of Artificial Intelligence faculty Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran (Chahkandi.vahid@gmail.com).

M. Jalali is now with the Department of Artificial Intelligence faculty Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran (mehrdadjalali@IEEE.org).

M. Mirshahi is a student of Department of Medical Engineering faculty Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran (mahsa_mirshahi@mshdiau.ac.ir).

A. Hosseini is now with the Department of Artificial Intelligence faculty Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran (ali.hosseini_ai@yahoo.com).

[1] HMIGA

The entropy H(X) is a measure of the uncertainty of a random variable X. For a discrete random variable X, with the p(x), the entropy of X is defined as

$$H(X) = -\sum_{x\epsilon X} p(x)logp(x).$$

Here, the log is to be base 2 and entropy is expressed in bits. The joint entropy of X and Y with joint pdf p(x,y),

$$H(X,Y) = -\sum_{y\epsilon Y}\sum_{x\epsilon X} p(x,y)logp(x,y).$$

When certain variables are known and others are not, the remaining uncertainty is measured by the conditional entropy,

$$H(Y|X) = -\sum_{x\epsilon X} p(x)H(Y|X=x) = -\sum_{y\epsilon Y}\sum_{x\epsilon X} p(x,y)logp(x|y).$$

Therefore, the joint entropy and conditional entropy has the following relation:

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y).$$

The information found shared by two random variables is important in our work and it is defined as the mutual information between two variables:

$$I(X;Y) = -\sum_{y\epsilon Y}\sum_{x\epsilon X} p(x,y)log\frac{p(x,y)}{p(x)p(y)}.$$

If the mutual information is large, the two variables are closely related. If the mutual information becomes zero, the two variables are independent. The mutual information and the entropy have the following relationships:

$$I(X;Y) = H(X) - H(X|Y),$$

$$I(X;Y) = H(Y) - H(Y|X),$$

$$I(X;Y) = H(X) + H(Y) - H(X,Y),$$
$$I(X;Y) = I(Y;X),$$
$$I(X;Y) = H(X).$$

The relationships are expressed in Fig. 1. The mutual information corresponds to the intersection of the information X with the information in Y.

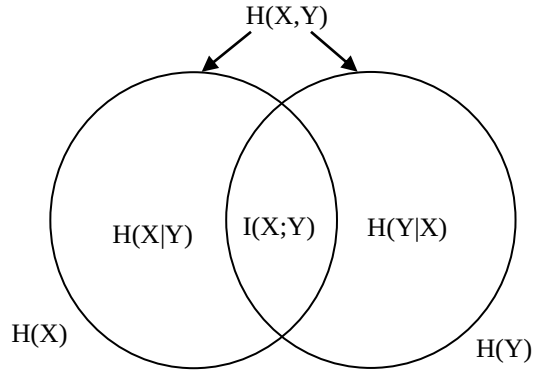


Fig. 1. The relations between mutual information and the entropy.

By far the concepts of entropy and mutual information are all described under the condition of discrete variables. But in software engineering databases many software project features are continuous, for continuous variables the entropy and mutual information are defined as follows:

$$H(X) = -\int p(x)logp(x),$$

$$I(X;Y) = \iint p(x,y)log\frac{p(x,y)}{p(x)p(y)}dxdy.$$

However, when the underlying pdfs (p(x), p(y) and p(x,y)) are continuous it is practically impossible to find exact integration. Therefore, the approximation estimators are proposed [11-12].

III. Genetic Algorithm

A genetic algorithm (GA) is a kind of searching algorithm in the global area. The Genetic Algorithms method [13] is an iterative search algorithm based on an analogy with the process of natural selection (Darwinism) and evolutionary genetics. The search aims to optimize a user-defined function (the function to be optimized) called the fitness function.

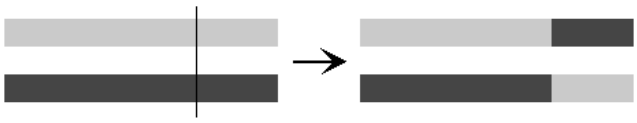
GA using three operators to generate a new population: selection, crossover and mutation.

Selection: Selection is the procedure by which good chromosomes are chosen in the next generation. Here we use the well-known roulette-wheel selection process, in which the selection probably of each individual is proportional to its fitness value. The probability of the individual *i* being selected is given by:

$$P_i = \frac{fit_i}{\sum_{i=1}^{n} fit_i}$$

Where fit_i is the fitness of *i* and *n* is the population size.

Crossover: In this process, the GA attempts to mate two parent together and product two child. These two parents are randomly selected from mating pool, and called solutions. Crossover operator can produce new chromosomes through combing partial structure of two parent individuals. We use one point crossover, and the crossover rate is applied with less than 80% probability.



(a) (b)

Fig. 2. One point crossover (a) parents (b) offspring

Mutation: Mutation is a common reproduction operator used for finding new points in the search space to evaluate. Mutation changes the characteristics of genetic material in a chromosome to sustain genetic diversity in the population. For each bit in the population, mutate with some low probability p_m. typically the mutation rate is applied with less than 2% probability.

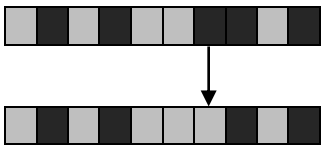


Fig. 3. Mutation Operator

IV. HYBRID MUTUAL INFORMATION AND GENETIC ALGORITHM (HMIGA)

In this method, first of all we generate a random binary population for GA, This population has as many as features in original dataset, and every bit in each row (each row is an individual) of population matrix that was one, we selected coordinate column (or feature) from the original dataset as a feature in new dataset. In this way, for every individual we will have a dataset with selected features, Then the value of mutual information calculates for every individuals in the dataset. For each individual one or more value obtained as mutual information of its selected feature. These values of mutual information represent measure of information of given feature and each value participates in calculation of individual fitness relative to its amplitude. we spotted a weight for each value of mutual information, every individual that has larger value of mutual information get a larger weight. Indisputable, each feature that has further mutual information plays more important role in calculation of final fitness of individual. After this, average the values use as fitness function for individual.

Since, every dataset has different number of features and we have to select some of them, in this paper we select 60% of whole features in original dataset.

In this way, fitness of all of individual in population obtained, and then by roulette wheel selection selected better individuals and produce a pool. After selection we have to select two individual and apply crossover operation on them and create two child in new pool after crossover. Then mutation rate with less than 2% probability is applied on pool that created with crossover. Fig.4. represented flowchart of hybrid proposed approach.

Where *runtime* is the number of calculation mutual information for all off individuals in population, in every iterative we attain a value as the best fitness, so use average of these values as final fitness. Note that runtime is optional and in our experiments is one.

V. SIMULATION EXPERIMENTS

In this section for evaluated proposed method, after selected features, with a classification method like Naive Bayes attain classification rate and this value is compared with one of that attain with MIFS (Mutual Information Feature Selection).

In our experiments we try to represent our method is better in some situations and some of them are equal. To serve for this purpose, brief description of benchmark datasets and experimental design will be firstly given, and then the simulation results will be presented and discussed.

To evaluate performance of this method we use 6 datasets. These datasets are all available in UCI Machine Learning Repository [14], and most of them are frequently used in literatures.

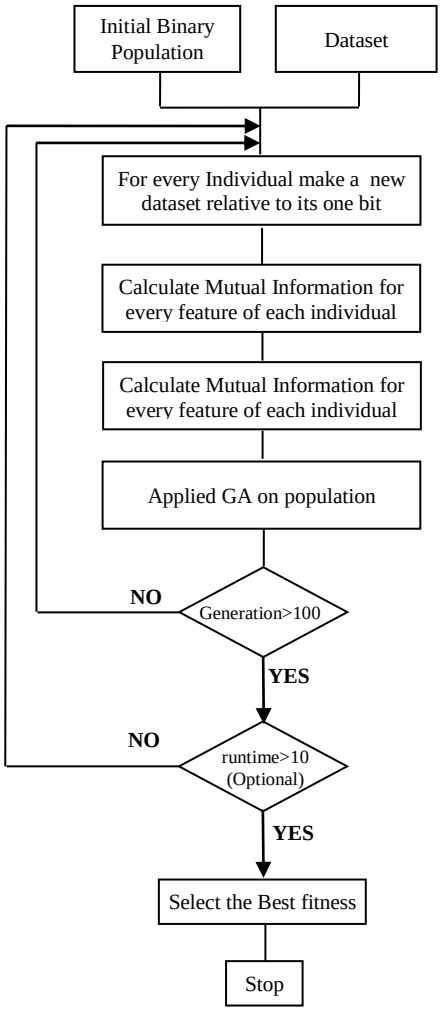


Fig. 4. Flowchart of hybrid Mutual Information and Genetic Algorithm (HMIGA).

TABLE I
THE DESCRIPTION OF 6 DATASETS IN OUR EXPERIMENTS

No.	Dataset	Instances	Features	Classes
1	Dematology	358	34	6
2	Diabet	768	8	2
3	Iris	150	4	2
4	Soybean-small	47	35	4
5	Spambase	4601	57	2
6	Wine	178	13	3

Table 1 summarizes some general information about these datasets. Their full documentations for original information can be obtained from the UCI website. Note that these datasets have numeric classes and differ in the sample size (range from 47 to 4601) and the number of features (from 4 to 57).

In simulation experiments, the number of feature chosen by our method is distinguished and in these experimental results is less than 60% of instances of datasets. After selection, datasets with newly selected features were passed to external learning algorithms to assess classification performance. In our experiments, one of the most popular classifiers, namely, NBC(naive Bayes) [18] is chosen to test prediction capability of the selected subset.

NBC [15] utilizes Bayes formula to distinguish which label an instance belongs to. The assumption behind it is that features are statistically independence with each other for the given target labels. Moreover, the conditional probability distribution of any given class satisfies normal distribution.

TABLE II
A COMPARISON OF CLASSIFICATION ACCURACIES OF NAÏVE BAYSE CLASSIFIER USING TWO FEATURE SELECTION ALGORITHMS ON 6 DATASETS
CCI: CORRECTLY CLASSIFIED INSTANCES, MI: MUTUAL INFORMATION

	10-Fold Cross validation			**66% Percentage split**		
No.	Unselected/ CCI	MI/CCI	HMIGA /CCI	Unselected/CCI	MI/CCI	HMIGA /CCI
1	48.20/172	50.97/184	**94.40**/337	55.37/67	56.10/69	**93.39**/113
2	20.60/158	68.84/528	**68.84**/528	18.77/49	65.52/171	**65.52**/171
3	97.99/146	100/149	**100**/149	98.04/50	100/51	**100**/51
4	95.65/44	**86.96**/40	82.61/38	93.75/15	93.75/15	**93.75**/15
5	**85.46**/3931	84.59/3891	84.79/3900	83.5038/1306	82.93/1297	**84.07**/1315
6	84.18/149	81.36/144	**86.84**/154	**86.67**/52	81.67/49	85.33/51

Many experiments have demonstrated that NB classifier has good performance compared with others on various real datasets.

The proposed algorithm has been implemented in Matlab 9.0. The experimental platform was Weka[2] (Waikato environment for knowledge analysis) [19], which is an excellent tool in data mining and brings together many machine learning algorithms under a common framework.

To achieve impartial results, ten10-fold cross validations and 66% percentage split had been adopted for each algorithm-dataset combination in verifying classification capability.

In Table 2 shows the result of experiment of 6 datasets with two test option.

VI. CONCLUSION

In this paper, we proposed a hybrid method for feature selection with combination of mutual information and genetic algorithm. In this method mutual information is calculated for each candidate feature of given an individual, since each individual has multiple candidate features, so we have multiple mutual information then we have to obtain a value as mutual information with a linear equation. After that these value plays fitness roll in genetic algorithm, and bring the best one for best individual.

[2] Weka is freely available at http://www.cs.waikato.ac.nz/~ml.

REFERENCES

[1] D.L. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. Lecture delivered at the "Mathematical Challenges of the 21st Century" conference of The American Math. Society, Los Angeles, August 6-11, http://wwwstat.stanford.edu/donoho/Lectures/AMS2000/AMS2000.html, 2000.

[2] L. Breiman. Random forests. Technical report, Department of Statistics, University of California, 2001.

[3] I.K. Fodor, A survey of dimension reduction techniques, Technical Report UCRL- ID-148494, Lawrence Livermore National Laboratory, US Department of Energy, 2002

[4] H. Liu, J. Sun, L. Liu, H. Zhang, "Feature selection with dynamic mutual information", Pattern Recognition 42 (2009) 1330 – 1339.

[5] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.

[6] A.L. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artificial Intelligence 97 (1997) 245–271.

[7] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Transactions on Knowledge and Data Engineering 17 (4) (2005) 491–502.

[8] R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[9] Shannon, C. E., & Weaver, W. (1949). The mathematical theory of communication. Urbana, IL: University of Illinois Press.

[10] Walkerden, F., & Jeffery, D. R. (1999). An empirical study of analogy-based software effort estimation. Empirical Software Engineering, 4, 135–158.

[11] Kwak, N., & Choi, C. H. (2002). Input feature selection by mutual information based on parzen window. IEEE Transactions on Pattern Analysis and Machine Intelligence,

24(12), 1667–1671.
[12] Moddemeijer, R. (1989). On estimation of entropy and mutual information of continuous distribution. Signal Processing, 16(3), 233–246.
[13] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. New York: Addison-Wesley, 1989.
[14] C.L.Blake, C.J.Merz, UCI repository of machine learning databases, Available from: (http://www.ics.uci.edu/~mlearn/ MLRepository.html), Department of Information and Computer Science, University of California, Irvine, 1998.
[15] G.H. John, P. Langley, Estimating continuous distributions in Bayesian classifiers, in: Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, San Mateo, 1995, pp. 338–345.
[16] I.H. Witten, E. Frank, Data Mining—Practical Machine Learning Tools and Techniques with JAVA Implementations, second ed., Morgan Kaufmann Publishers, Los Altos, CA, 2005.
[17] Y.F. Li , M. Xie, T.N. Goh, "A study of mutual information based feature selection for case based reasoning in software cost estimation", Expert Systems with Applications 36 (2009) 5921–5931.
[18] G. H. John, P.Langley, "Estimating continuous distributions in Bayesian classifiers", in: Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, San Mateo, 1995, pp.338–345.
[19] I. H. Witten, E. Frank, Data Mining — Practical Machine Learning Tools and Techniques with JAVA Implementations, seconded, Morgan Kaufmann Publishers, Los Altos, CA, 2005.

Finding Perfect-Predictor Feature Sets for Supervised Classification Using Genetic Algorithms

Alexander Liu and Cheryl Martin
Applied Research Laboratories, The University of Texas at Austin, Austin, TX, USA

Abstract - *In some supervised classification problems, the presence of a particular subset of features may be perfectly predictive of a data point's class label. Discovery of these perfect-predictor feature sets can be used to explore the data and help create new features that capture dependencies among existing features. In this paper, we test the capability of genetic algorithms for finding perfect-predictor feature sets and describe limitations.*

Keywords: Feature extraction, genetic algorithms

1 Introduction

In the field of machine learning, supervised classifiers can be used to predict the category of data where each data point's properties are described by a set of features or attributes. Many classifiers assume feature independence. If interaction between features is likely, it is often left for domain experts to create new features that properly capture the interaction between existing features. In this paper, we study problems where the presence of a subset of existing features is perfectly predictive of a particular class. The features may exist frequently in the data set, independently, but they are perfect predictors only when they co-occur. Such subsets violate the feature independence assumption.

For example, this problem may occur when recommending items to a user. In this case, there are two classes: recommend the item, or do not recommend the item. If the data points are movies and the features are genres (e.g., action, comedy, drama, historical, mystery, romance), the user might always be interested in movies that are either "action comedies" or "historical mysteries" (but not necessarily interested in movies that are only action, only comedy, only historical, or only mysteries). The genre(s) of other movies the user is interested in or not interested in may vary. However, the important pattern to discover is that "action comedies" and "historical mysteries" should always be recommended. Discovering these perfect-predictor feature sets is useful for understanding the data and potentially improving the accuracy of standard classifiers that assume feature independence.

This paper defines the concept of perfect-predictor feature sets and proposes a method of finding such feature sets. We will show that existing approaches designed for similar, related tasks are not always suitable for finding perfect-predictor feature sets. We propose a genetic algorithm approach specifically designed to find perfect-predictor feature sets and show that the proposed method can perform well in many cases. We will also examine factors such as the dimensionality of the problem that influence whether genetic algorithms perform well or poorly.

2 Problem definition

The problem of finding perfect-predictor feature sets is as follows. We are given a classification problem consisting of binary features (1 if the feature is present in the data point, 0 otherwise). In at least one class, there may exist one or more perfect-predictor feature sets. A perfect-predictor feature set for class y_c is a set of features that meet the following requirement: If all features in this perfect-predictor feature set are present in a data point, then the data point must be in class y_c, regardless of the values of the other features in that data point.

Points that do not contain features in any perfect-predictor feature sets may be in any class. In addition, the number of features in each perfect-predictor feature set is unknown, the number of perfect-predictor feature sets is unknown, and there is no ground truth mapping any data point to some perfect-predictor feature set.

Perfect-predictor feature sets can occur in many problems. As discussed in the introduction, these sets may occur when recommending items where features could correspond to genres, object properties, or user-created tags. Perfect-predictor sets also arise in text classification problems that are not purely topic based, such as automatic security classification, legal review, e-discovery, and FOIA requests.

Perfect-predictor feature sets do not occur in all supervised classification problems. In many problem domains, there is no subset of features perfectly predictive of class label. Moreover, the forced application of methods to discover perfect-predictor feature sets may result in the discovery of sets of features applicable to only a few data points in the training data due simply to noise or statistical aberration. As discussed in Section 5, thresholds may be specified for determining whether a discovered perfect predictor is significant.

3 Related work

The closest related work for finding small sets of interacting features is work on creating decision rules [1] and finding association rules [2]. The goal of the decision rule algorithm is to find a set of rules that best classify data points, where the rules are typically conjunctions of possible feature values (e.g., if feature 3 and feature 5 both appear, then the

point is positive). Thus, if perfect-predictor feature sets are present, then the hope is that these should be discovered as decision rules which can then be used as features for other classifiers or simply for understanding the data. However, decision rules are not designed to find perfect-predictor feature sets, and are instead designed to classify as many points correctly as possible. Thus, if there are highly informative (but fallible) features that tend to be correct for a large number of data points, these informative but fallible features will tend to dominate the discovered rules.

The goal of association rules is to find sets of features that co-occur frequently in the data set. Thus, a set of features that always co-occurs (i.e., has a confidence equal to one, where confidence is a term used in the association rule literature) could represent a perfect-predictor feature set if one of the features is the class label. However, in association rules, there is typically no concept of a class label used for classification purposes. Thus, there is no concept that one should find a set of features that co-occur for one class but do not co-occur for another class. Class association rule algorithms extend association rule algorithms to discover association rules useful for separating one class from another (e.g., [3], [4], [5], [6]). Thus, class association rule algorithms are algorithms that blend decision rule algorithms with association rule algorithms. One viewpoint of perfect-predictor feature sets is that they correspond to class association rules with confidence equal to one. A known problem with many association rule algorithms, however, is the difficulty in enumerating frequent itemsets. We will show that class association rule algorithms that enumerate frequent itemsets are limited when searching for perfect-predictor feature sets. A more thorough comparison against a wider variety of class association rule algorithms will be discussed in future work.

More broadly, there has been much previous work on feature selection and extraction in general. For example, [7] is one of the fundamental works on feature selection for text classification. While the goal of feature selection is to select a subset of features most useful for classification, this problem and the methods used are completely different from the problem of finding perfect predictors. Among other differences, most feature selection techniques, including those studied in [7] such as mutual information and information gain, consider each feature independently. There are approaches in feature extraction that create new features from existing features. However, many popular feature extraction techniques create a smaller dimensional subset of features consisting of linear transformations of existing features [8]. This is again a different problem than the one studied in this paper.

This paper proposes a genetic algorithm approach to finding perfect-predictor feature sets. Many genetic algorithm solutions to the related problems described above have been proposed in the past. For example, approaches such as [9] and [10] have been proposed for feature selection and extraction. Techniques also exist for using genetic algorithms for finding decision rules (e.g., [11], [12]) and association rules (e.g., [13], [14]). However, as discussed, the problems addressed in feature extraction and decision rule creation differ from the problem posed in this paper (regardless of whether genetic algorithms or some other approach are used to solve these problems). Class association rule algorithms are very related to the problem of perfect-predictor feature sets, and perfect-predictor feature sets can be captured by only looking for class association rules with confidence equal to one. However, class association rule algorithms can be problematic when the number of frequent itemsets becomes too large. Because existing methods for related problems can be unsuitable, a new approach specifically designed for finding perfect-predictor feature sets is proposed in the next section.

4 A genetic algorithm approach for finding perfect-predictor feature sets

In this paper, we study a genetic algorithm approach for finding a single perfect-predictor feature set for the positive class. Extensions to multiple perfect predictors and multiple classes are discussed as future work.

In general, genetic algorithms can be used to find a set of values that optimize some fitness function using a pool of candidate solutions (see [15] for a thorough introduction). In our approach, each candidate solution is a vector of n_f binary features. We will denote the *j*th member of the pool candidate vectors as $\boldsymbol{p_j}$ and will denote the *k*th feature of $\boldsymbol{p_j}$ as p_{jk}. If $p_{jk} = 1$, then it means that, according to the *j*th candidate solution, the *k*th feature is part of the discovered perfect-predictor feature set; if $p_{jk} = 0$, then it means that the *k*th feature is not believed to be part of the discovered perfect-predictor feature set.

It is desirable for an approach to finding perfect predictors to work on problems where the search space is extremely large. This would allow the approach to be applied to high-dimensional problems such as text classification. Thus, much of the proposed approach described below is designed to scale to a large number of possible features and handle cases where the number of features in the perfect-predictor feature set is small. The approach is summarized in Algorithm 1. The following subsections provide details relevant to the algorithm.

4.1 Fitness function

In machine learning, common metrics for classifier performance include precision, recall, and f1-measure. Let $\mathbb{f}(\boldsymbol{p_j})$ denote the fitness function of candidate solution $\boldsymbol{p_j}$. A fitness function based on f1-measure can be calculated if one classified all points in the data set with the following decision rule: classify a data point as positive if all features where $p_{jk} = 1$ are equal to 1, and negative otherwise. Since a point that contains a perfect-predictor of the positive class must be in the positive class, then precision for a candidate solution must be equal to 1 if the candidate solution matches a perfect predictor. We use a precision-gated f1-measure as the fitness function, where $\mathbb{f}(\boldsymbol{p_j})$ is equal to f1-measure if precision=1 and 0 otherwise. This is a better fitness function than f1-measure alone because it is possible for a candidate solution

to achieve a non-zero f1-measure without being a perfect-predictor. It is also better than using precision alone because the co-occurrence of any word with a word that appears only once in a single document in the positive class would have a precision equal to 1, even though candidate solutions representing these feature combinations would not be useful as created features.

4.2 Initialization

We initialize each pool member as follows: n_{init} features are chosen to equal 1, while the remaining features are set to 0. The n_{init} features chosen to equal 1 are chosen with probability proportional to the f1-measure obtained by classifying all data points in the data set into the positive class if that feature is present, and negative otherwise. In preliminary experimentation, other forms of initialization such as random initialization did not work well.

In applying the proposed approach, n_{init} should be chosen to be similar to the expected size of the perfect-predictor feature sets. For example, in our experiments, we use a value of $n_{init} = 5$ since the perfect-predictor feature sets are known to contain only a few (i.e., less than five) features. Note that we have not extensively tested cases where n_{init} is vastly different from the actual number of features in perfect-predictor feature sets, and examining the effects of varying values of n_{init} during initialization is planned as future work.

Algorithm 1: Finding a perfect-predictor feature set

Inputs:

- number of pool members
- number of iterations
- number of features in each pool member to initialize to 1 (n_{init})
- probability of mutation

1. **Initialize** pool
   a. Create pool with n_p members
   b. Each pool member consists of n_f bits, where a 1 in the string indicates that the feature is part of the perfect-predictor set, and 0 indicates the feature is not
   c. Initialize based on f1-measure of each feature
2. **Iterate** until user defined number of iterations is complete
   a. **Reproduction:** Select n_p members from current pool using expected value model; place selected members into a new pool
   b. **Crossover:** Randomly pair members of new pool; perform standard crossover on the pair of strings, where a crossover point is chosen uniformly at random from all possible crossover points; completely replace existing pool with new solutions obtained after crossover
   c. **Mutation:** Mutate new pool members, where the probability of mutating a feature is a user parameter

4.3 Reproduction

One method of reproduction is to select $\boldsymbol{p_j}$ with probability equal to $\frac{\mathbb{f}(\boldsymbol{p_j})}{\sum_{j'=1}^{n_p} \mathbb{f}(\boldsymbol{p_{j'}})}$. In this method of reproduction, the expected number of times $\boldsymbol{p_j}$ is selected is equal to $n_p \frac{\mathbb{f}(\boldsymbol{p_j})}{\sum_{j'=1}^{n_p} \mathbb{f}(\boldsymbol{p_{j'}})}$. However, since selection is probabilistic, if a few pool members have very high fitness compared to the remaining pool members, then those few pool members with high fitness can dominate the new pool, reducing the overall variety in the population. This is particularly problematic if the dimensionality is high and the number of features in the perfect-predictor feature set is small since many candidate solutions will tend to have low (or zero) fitness.

Thus, instead of the above, we use the expected value model of reproduction [15] when choosing existing pool members to populate a new candidate pool. In the expected value model, the number of times that $\boldsymbol{p_j}$ is selected is bounded such that it cannot be selected more than the expected number of times.

In preliminary experiments, we found that using other methods of reproduction (e.g., elitism [15]) did not work as well for the problem considered in this paper, particularly as the number of dimensions grew.

4.4 Crossover

Our proposed approach uses a standard method of crossover: members in the pool are randomly paired and a crossover point is chosen uniformly at random from all possible crossover points. The new points after crossover completely replace the current pool of solutions.

Other forms of crossover did not outperform standard crossover in preliminary experiments.

4.5 Mutation

The final step is to perform mutation, where the probability of performing mutation is a user specified parameter. Instead of randomly selecting features in candidate vectors for mutation, we select the *k*th feature for mutation with probability proportional to the correlation of the *k*th feature in all points in the data set with the class labels. This is useful if n_{init} is small compared to the number of dimensions since in this case most features in a pool member $\boldsymbol{p_j}$ tend to be zero. The proposed method of mutation will therefore tend to change features from zero to one. Thus, there will be a higher chance of including features that are highly correlated with the positive class label and thus are more likely to be part of one of the perfect-predictor sets of the positive class.

5 Results

There are three main goals of our experiments: (1) demonstrate limitations of existing algorithms, which work for related problems, for finding perfect-predictor feature sets, (2) determine how well genetic algorithms perform at finding perfect-predictor feature sets as a function of the dimensionality of the problem, and (3) determine the effect of the relative number of points that contain perfect predictors on

the proposed genetic algorithm approach. We will also discuss some effects of noise and feature rarity.

We create several artificial data sets in order to have ground truth necessary to analyze the results of our experiments. The experiments are based on standard text classification data in order to have realistic feature distributions. All experiments are based on the reviews data set [16], a standard benchmark text classification data set consisting of 2000 movie reviews. A bag-of-words model where stopwords have been removed is used as the set of potential features.

In all our experiments, we use 100 pool members and run the genetic algorithm for 100 iterations. Because genetic algorithms can get stuck in local minima, we run the proposed approach multiple times and take the result with largest fitness. Finally, we use $n_{init} = 5$ during initialization, and the probability that a feature will mutate is 0.01.

5.1 Existing versus proposed approach

In the first set of experiments, we compare decision rule and class association rule approaches to the proposed genetic algorithm for the problem of finding perfect predictors. The original class labels are used. To create known perfect-predictor feature sets, two new features are added. The two features are both equal to one for some number n_{ppfs} data points in the positive class (i.e., we add all features in a perfect-predictor feature set to n_{ppfs} positive class points). For all other data points, the two features are randomly set to both be zero or for only one feature to be equal to one, with an equal probability of any of the three possibilities occurring. We will use the case where there are 25 of the original features in the data set and 2 features corresponding to the injected perfect-predictor feature set, for a total of 27 dimensions. The 25 features selected from the original data correspond to the features that appear most frequently in the positive class. This method of selecting features was chosen since other methods, such as random selection, may pick features that occur infrequently either in the positive class or in both classes. Since a significant number of words occur infrequently in text corpora, randomly chosen features can tend to occur relatively infrequently. If the chosen features occur infrequently in the positive class relative to the features in the perfect-predictor feature set, the problem becomes unrealistically easy since the frequency of feature occurrences would then be highly indicative of whether or not the feature is part of a perfect-predictor feature set.

Empirically, we vary n_{ppfs} in order to test how many points must be part of a perfect-predictor feature set before it can be found. In the original class labels, 1000 points are in the positive class and 1000 points are in the negative class. Thus, when $n_{ppfs} = 1000$, the perfect predictor is in every positive class point.

The decision rule learning algorithm RIPPER [1], the association rule algorithm a priori [17], and the proposed approach are run on the data described above. In particular, we use the versions of RIPPER and a priori implemented in Weka[1]. The Weka implementation greatly extends the capability of the a priori algorithm. In particular, the Weka version of a priori allows the discovery of class association rules while the original a priori algorithm did not. The results are summarized in Table 1. "Yes" is listed if the perfect-predictor is found, and "No" is listed otherwise.

Table 1: Results based on original class labels and injected perfect-predictor feature set

n_{ppfs}	PERCENT OF POS CLASS	PROPOSED APPROACH	RIPPER	A PRIORI: CLASS ASSOCATION RULES
5	0.5%	Yes	No	No
25	2.5%	Yes	No	No
50	5%	Yes	No	No
100	10%	Yes	No	No
250	25%	Yes	No	Yes
500	50%	Yes	No	Yes
1000	100%	Yes	Yes	Yes

RIPPER is able to extract a rule that states that the class label should be the positive class if both the perfectly predictive features are equal to one only when $n_{ppfs} = 1000$ (i.e., when the perfect-predictor feature set is present in every point in the positive class). However, for all other tested values of n_{ppfs} (e.g., even when the perfect-predictor feature set occurs in half the positive class points), RIPPER does not extract a rule corresponding to the perfect-predictor feature set. This is not to say that RIPPER does not work well for classification (the purpose for which it was designed). However, this means that the rules learned by RIPPER should not be used for feature creation and data exploration when perfect-predictor feature sets are expected.

As mentioned, association rule algorithms do not make special use of the class label. Instead, one must use a class association rule algorithm. Moreover, to look for perfect-predictor feature sets, one must look for only rules with confidence equal to one. A priori can find the perfect-predictor feature sets when n_{ppfs} is large relative to the total number of points (e.g., when $n_{ppfs} \geq 250$). In the association rule literature, this is related to the support of the discovered rule (i.e., how many times the rule applies in the data). As the perfect-predictor occurs in less and less points, the support of the corresponding rule decreases. The minimum support for discovered rules is a user parameter to the a priori algorithm, and it is known that the number of frequent itemsets considered increases as the minimum support threshold decreases.

In our experiments, the number of frequent itemsets becomes too large when $n_{ppfs} < 250$, and a priori is unable to scale to handle this problem. Thus, class association rule algorithms that enumerate frequent itemsets cannot handle cases where the support threshold is too low. In addition,

[1] http://www.cs.waikato.ac.nz/ml/weka/

association rule algorithms that enumerate frequent itemsets are also hampered by the dimensionality of the problem. The run-time of algorithms that rely on frequent itemset discovery increases exponentially as a function of the number of dimensions since increased dimensionality will increase the number of frequent itemsets [18].

In contrast, the proposed approach is always able to find the perfect-predictor feature set regardless of the value of n_{ppfs} used. In experimentation, we varied n_{ppfs} from 5 to 1,000. Note that when $n_{ppfs} = 5$, the perfect-predictor feature set occurs in only 5 out of 1000 points in the positive class (i.e., 0.5% of the positive class), and the proposed approach is still able to exactly find the perfect-predictor feature set.

5.2 Effect of dimensionality and relative number of points

In the second set of experiments, we create eight new sets of class labels based on a single perfect predictor consisting of two existing features (i.e., two existing words). These perfect-predictor feature sets consist of pairs of words that were chosen in order to create data sets with a variety of different characteristics in terms of the number of times the perfect-predictor occurs and the number of times the features occur outside of a perfect-predictor. For example, the number of times the features co-occur (i.e., are a perfect predictor) ranges from 54 to 577. In addition, the number of times the features occur but are not part of a perfect predictor (i.e., occurs alone) varies widely for each data set (from 154 to 663 times). The number of times features co-occur and the number of times only one of the features occurs for each of the eight data sets are summarized in Table 2. As before, features that appear in perfect predictors can appear in both the positive and negative class (i.e., it can appear in the negative class if the other feature in the perfect-predictor feature set is not present).

In this set of experiments, all documents that contain features in a perfect predictor are placed in the positive class, and all but $n_{+,other}$ points are placed in the negative class. The $n_{+,other}$ points consist of randomly chosen points that are also included in the positive class in order to test cases where perfect-predictor feature sets exist for only some of the positive class points and to determine how well the genetic algorithm works as the relative number of points containing the perfect-predictor feature set changes (i.e., the same purpose as varying n_{ppfs} in the previous experiments). We will present results where $n_{+,other}$ varies from 0 to 1000.

We vary the number of features by selecting the n_f features that appear most frequently in the positive class in addition to the features that are in the perfect-predictor feature set. n_f is varied from 25 to 900.

In Figure 1 we plot results of applying the proposed approach to all the artificial data sets based on the new class labels. The two axes of the graph correspond to the two experimental variables being controlled: the relative number of points in the positive class containing the perfect-predictor feature set, and the number of dimensions. The marker used to plot the result of each combination of the two experimental controls is based on the Jaccard coefficient of the discovered set of features and the actual set of perfect-predictor features. The Jaccard coefficient is a standard method of measuring the similarity between two sets, and is equal to the number of elements in the intersection of the two sets divided by the number of elements in the union of the two sets. Thus, the Jaccard coefficient lies between zero and one, with a Jaccard coefficient of one indicating that the two sets are completely identical and a Jaccard coefficient of zero indicating that the two sets do not share any elements.

Table 2: Experimental data sets based on existing features

ID	NUM TIMES FEATURES CO-OCCUR	NUM TIMES ONLY FEATURE 1 OCCURS	NUM TIMES ONLY FEATURE 2 OCCURS
1	577	470	388
2	519	663	255
3	136	369	154
4	127	529	188
5	125	238	212
6	124	381	166
7	110	398	166
8	54	261	228

In Figure 1, results with larger Jaccard coefficient are plotted using larger, darker markers. These results indicate that when the number of dimensions is less than 300, the proposed approach almost always is able to exactly find the perfect-predictor feature set (i.e., running the proposed approach results in a Jaccard coefficient of 1). For dimensionality between 300 and 500 dimensions, the approach can still perform well. However, for greater than 500 dimensions, the search space becomes too large for the proposed approach to handle well. These results are true regardless of the relative number of points in the positive class with the perfect predictor. Thus, the main limit of the proposed approach is the dimensionality of the problem.

5.3 Analysis

On average, our proposed approach (i.e., initializing and running the genetic algorithm 100 iterations) took 37.5 seconds on a single core of a machine with two 3.2 GHz quad-core Intel Xeon CPUs and 16 GB of RAM. Note that, unlike association rule algorithms that enumerate frequent itemsets, the dimensionality of the problem does not affect the run time. Factors affecting the speed of the approach include the number of pool members, the number of data points, and the number of iterations. In particular, reducing the number of iterations directly reduces the time required for the proposed algorithm to run. In preliminary experimentation, we found that the proposed approach tended to converge to a solution before 100 iterations completed. Further steps for reducing the time taken to run our approach are also possible, such as more efficient implementations of our approach and parallelization.

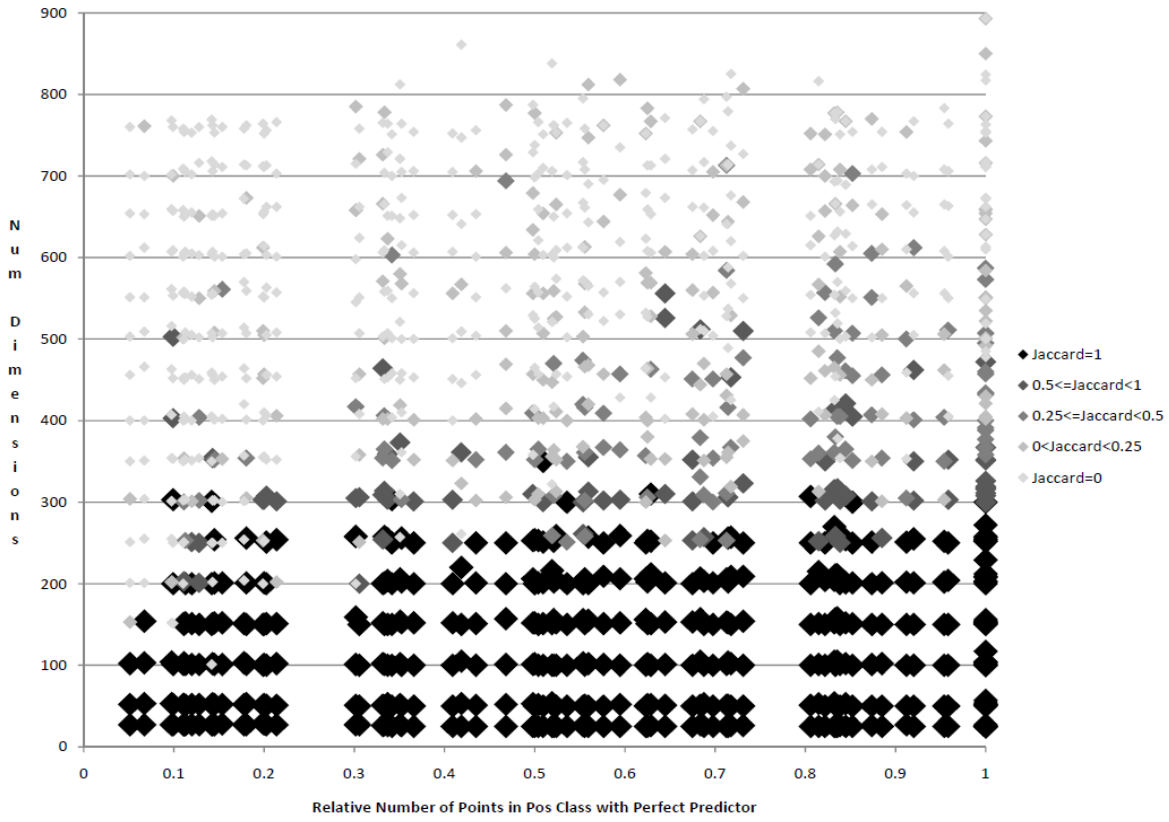


Figure 1: Results for artificial labels based on existing words

If the proposed approach is applied to a supervised classification problem where there is no set of features perfectly predictive of a large number of data points, two undesirable results could occur. First, a perfect predictor may be identified that is incorrect and based on noise or statistical rarity of features. Second, a perfect predictor may be found that is correct, but applies to so few data points as to be insignificant.

In the absence of rarely occurring features, we empirically verified that the proposed approach finds no perfect-predictor feature sets where they do not exist. The reviews data set is taken from a domain (sentiment mining) where perfect-predictor feature sets are not expected, and original features and class labels from this dataset were used to test this hypothesis. Using the 25 most commonly occurring features, the proposed approach was unable to find a set of features with fitness greater than zero. Since the fitness is precision-gated f1-measure, this means no set of features could be found that always co-occur in the positive class but not in the negative class. This is as expected, and indicates that if the fitness of all strings after the proposed approach is run is zero, then no perfect-predictors are present in the data.

However, rarely occurring features could cause the approach to identify incorrect or insignificant perfect predictors. To mitigate these cases, the method of reducing the dimensionality used in the empirical set-up can be applied in practice as follows.

Let n_+ be the number of positive class points and let n_s be the number of times a perfect-predictor feature set occurs in the positive class. If the perfect-predictor feature set occurs n_s times in the positive class, then each feature in the feature set must occur in at least n_s positive class points. Thus, to find a perfect-predictor feature set that occurs n_s times in the positive class, one can ignore all features that occur less than n_s times in the positive class. Since we do not know n_s exactly, we can prune all features that do not occur at least $t * n_+$ times in the positive class, where t is some user-chosen threshold. Note that this is a similar method as the one used by a priori to prune the number of frequent itemsets based on support.

Feature pruning is useful for two reasons. The first reason is straightforward: feature pruning reduces the dimensionality of the problem. As discussed above, the proposed approach is limited by dimensionality, so feature pruning can be helpful to get around this limit. Feature pruning also reduces the impact of rarely occurring words. In the degenerate case where there are no perfect-predictor feature sets or the near-degenerate case where the perfect-predictor occurs in a very small percentage of points, rarely occurring features may appear to be part of perfect-predictor feature sets if they only occur in one class. Feature pruning can therefore help guard against the presence of rarely occurring words by removing them from the feature space.

Note, however, that in the non-degenerate case, perfect predictors arising due to rarely occurring features and or statistical noise are only a problem if they occur in more points than "true" perfect predictors, where they would have higher fitness than the true perfect-predictor feature sets.

Furthermore, the use of validation sets can help distinguish between perfect-predictors arising due to chance and actual perfect-predictors. Moreover, in the near-degenerate case, the fitness of all discovered perfect-predictors will be quite low. Thus, the near-degenerate case can be guarded against by feature pruning, use of validation sets, or ignoring any perfect-predictors that only occur in a low percentage of points.

6 Future work

There are several areas of future work. One area of future work is to extend our approach to find several perfect-predictor sets. For example, this could be accomplished simply by removing all instances that contain the current discovered set of features and to iteratively re-run the approach, a similar approach used by many decision rule techniques. Another possibility is to remove the features in the discovered feature set and re-run the algorithm. Other extensions to the current algorithm include studying cases where there is a significant amount of noise in the data, imbalanced datasets, and cases where the features are not binary.

Finally, as mentioned in the related work, class association rules are very closely related, and many class association rule algorithms exist. The current experiments only compare the proposed approach against a single class association rule algorithm. A more thorough comparison would therefore be useful. Most likely, any approach that enumerates frequent itemsets in a method similar to the a priori algorithm will have similar problems with high dimensional datasets. However, other methods that find frequent itemsets in ways very different from a priori can potentially be used to find perfect-predictor feature sets on high dimensional data. In particular, the present approach should be compared against [19], which defines a concept called “top rules” that appear to be highly related to perfect-predictor feature sets.

7 Conclusion

In this paper, we introduced and defined the problem of finding perfect-predictor feature sets. We identify techniques for related problems that cannot be effectively applied to find perfect-predictor feature sets. In particular, decision rule algorithms solve a different problem, and a priori-based class association rule algorithms are limited by both the number of dimensions in the problem as well as the number of data points containing the perfect-predictor. We propose a genetic algorithm approach for finding perfect-predictor feature sets and show that the approach works well on problems under 300 dimensions. The approach does not seem limited by the percentage of points that must contain the perfect-predictor feature set.

8 References

[1] W. Cohen and Y. Singer, "Context-Sensitive Learning Methods for Text Categorization," *ACM Transactions on Information Systems*, vol. 17, pp. 141-173, 1999.

[2] R. Agrawal, T. Imielinski, and A. N. Swami, "Mining association rules between sets of items in large databases," *SIGMOD Record*, vol. 22, pp. 207-216,1993.

[3] B. Liu, W. Hsu, and Y. Ma, "Integrating Classification and Association Rule Mining," KDD, 1998.

[4] G. Dong, X. Zhang, L. Wong, and J. Li, "CAEP: Classification by Aggregating Emerging Patterns," 2nd International Conference on Discovery Science, 1999.

[5] W. Li, J. Han, and J. Pei, "CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules," IEEE International Conference on Data Mining, 2001.

[6] X. Yin and J. Han, "CPAR: Classification based on Predictive Association Rule," SIAM International Conference on Data Mining, 2003.

[7] Y. Yang and J. Pedersen, "A comparative study on feature selection in text categorization," *International Conference on Machine Learning*, pp. 412-420, 1997.

[8] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *JMLR*, vol. 3, pp. 1157-1182, 2003.

[9] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, "Dimensionality reduction using genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, pp. 164 -171, 2000.

[10] H. Chen and B. Zou, "Optimal feature selection algorithm based on quantum-inspired clone genetic strategy in text categorization," Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, 2009.

[11] A. Pietramala, V. L. Policicchio, P. Rullo, and I. Sidhu, "A Genetic Algorithm for Text Classification Rule Induction," Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II, 2008.

[12] E. Noda, A. A. Freitas, and H. S. Lopes, "Discovering interesting prediction rules with a genetic algorithm," *Evolutionary Computation*, vol. 2, 1999.

[13] P. D. Shenoy, K. G. Srinivasa, K. R. Venugopal, and L. M. Patnaik, "Dynamic Association Rule Mining using Genetic Algorithms," *Journal Intelligent Data Analysis*, vol. 9, pp. 439-453, 2005.

[14] P. P. Wakabi-Waiswa and V. Baryamureeba, "Extraction of Interesting Association Rules Using Genetic Algorithms," *International Journal of Computing and ICT Research*, vol. 2, pp. 26-33, 2008.

[15] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*: Addison-Wesley, 1989.

[16] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," Conference on Empirical Methods in Natural Language Processing, 2002.

[17] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," *20th International Conference on Very Large Databases*, pp. 487-499, 1994.

[18] F. Pan, G. Cong, A. Tung, J. Yang, and M. Zaki, "CARPENTER: Finding Closed Patterns in Long Biological Datasets," KDD, 2003.

[19] J. Li, X. Zhang, G. Dong, K. Ramamohanarao, and Q. Sun, "Efficient mining of high confidence association rules without support thresholds," Principles of data mining and knowledge discovery, 1999.

A novel knowledge-discovering approach from massive data

H. Bouhamed[1], A. Rebai[2], T. Lecroq[1] and M. Jaoua[3]
[1]Department of Computer Science, University of Rouen, Rouen, France
[2]Department of Bioinformatics, Biotechnologies Centre of Sfax, Sfax, Tunisia
[3]Department of Computer Science, University of Sfax, Sfax, Tunisia

Abstract - *The objective of our study lies in developing a new data-reducing approach whose useful application is crucial as a prerequisite to learning Bayesians Networks (BNs) structures. The application of our approach may, in some cases, turn out to be significantly effective in reducing the computational complexity of the BNs structures learning. Firstly, it is essential to define BNs and recall its widely-common relevant problem of learning structure from massive data. Secondly, we suggest a solution for optimizing the computational complexity by means of data organizational and optimization methods. As a matter of fact we have applied our approach to biological facts concerning hereditary complex illness where the literatures in biology identify the responsible variables for those diseases. Finally, we conclude by highlighting the limits arched by this work and proposing suggestions for further research.*

Keywords: optimization, automatic knowledge discovery, Bayesian Network (BN), score fusion, selection of clusters.

1 Introduction

It is worth noting that the immense amounts of diverse data, made recently available, pertaining to different research fields, has increasingly kindled interest for the training techniques, skills and efficiency to monitor and deal with the sophisticated data interdependences. Owing to their flexibility, comprehensive mathematical formulations, easy handling and manipulation, the BNs are most often regarded as the favourably chosen models to be applied to various fields and a wide array of applications: astronomy areas, web-mining as well as bioinformatics applications. Yet, training and handling the BNs structures within a large number of diverse variables remains a great challenge to retrieve in the contexts of powerful high-speed processing calculations, algorithmic complexities as well as application execution time [1]. In this respect, several algorithms have recently been devised and developed for the sake of applying and monitoring BNs structures from data [20] and [3]. In fact, a wide array of these algorithms rests on metric scoring methods, the widely compared and most frequently applied scoring methods [2] and [26].

Nevertheless these algorithms and scoring methods remain still insufficient and limited in scope as regards those cases in which the number of variables exceed some hundreds of thousands [18]. Moreover, they do not implement upstream processing treatment of cases where variables are either not sufficiently and entirely implicated or irrelevant and redundant with respect to a certain information system, in such a way as to exclude them from all-considerations in the modeling process. Most frequently, however, the fact of excluding non-implicated or irrelevant variables, or those whose implications, might considerably decrease the algorithmic complexities as well as the execution time. Thus, providing the possibility of extending the variable-modeling capacity during the initial step of the information system processing stage.

Noteworthy, a more developed and highly promoted type of algorithm has often been applied, using the hierarchical class of latent models (HCLM) [1], along with the double layer BN [25]. These types of algorithms are promising in so far as data reduction capacities are concerned. Yet, they turn out to be incapable of processing a quite large number of variables exceeding the range of about one thousand [18]. As for [10], they have set up a special method allowing to process quite a large amount of data (up to 6000 variables). This has been made possible by means of reducing the HCLM research and retrieving space to some possible connecting relationships among brother nodes. Nevertheless, the restriction imposed by this method is likely to deviate the model from actual and realistic facts [18].

As regards our research study, a novel approach has been devised. Designed to achieve a maximum reduction of the variables number prior to a BN structure learning implementation, this proposed model's effective usefulness lies in the fact that during its execution, neither data flow nor information loss could be engendered during its implementation. At this junction, it is worth noting that this new approach is to be tested and estimated on an intricate data base of genetics' variables pertaining to a complex genetic illness.

As a matter of fact, the present work turns out to be crucially important for a number of reasons. First, it helps determine and extract relevant information from a large set of data variables. Second, it enables to exploit the extracted information and reduce the scope of data ahead of BN modelization. Actually, this modelization is targeted to lessen the level and effects of algorithmic complexity as well as reduce the processing execution time without any loss of information flow nor any resulting imposed restriction as a prerequisite for the structure initiation or apprenticeship skills, as can be noticed in the elaborated works of [10].

In addition, on combining pioneering approach with other BN apprenticeship algorithms, one might well manage to modelize once non-modelizable information systems, thanks to the larges number of variables made available.

As for the remaining constituent sections of the present research work, they are organized as follows: the next section is allotted to the introductory exposition of the BN structure learning problem. As for the following section, a new data reduction approach is going to be presented, which is going to be applied and tested on a special biological data base. As regards the last section, it depicts our conclusion along with the perspectives for further future researches.

2 BN and data-structure learning problem

It is worth highlighting that knowledge representation and the related reasoning, thereof, have given birth to numerous models. The graphic probability models, namely, BN, introduced by Judea Pearl in the 1980s, have been manifested in to practical tools useful for the representation of uncertain knowledge, and reasoning process from incomplete information.

To note, a BN $B = (G, \Theta)$ is written under the form of:

- $G = (X, E)$ graph managed without circuit summits of whose associates a set of random variables $X = \{X_1,...,X_n\}$
- $\Theta = \{P(X_i|P_a(X_i))\}$, probability set of every knot X_i conditional upon the state of its parent relatives $P_a(X_i)$ in G.

Hence, the BN graphic representation indicates the dependences (or independences) between variables and provides a visual knowledge representation tool, that turns out to be more easily understood by its users. Furthermore, the use of probability allows to take into account the uncertainty, by quantifying the dependences between variables. These two properties have been at the origin of the first terms allotted, initially, of BN, "probabilistic expert systems", where the graph used to be compared with some set rules pertaining to a classic expert system, and conditional probability presented as a quantification measurement of the uncertainty related to these rules [16].

In this respect, Reference [13] has shown that BN have allowed to represent, in a compact way, the joint probability distribution relevant to all variables:

$$P(X_1, X_2,..., X_n) = \prod_{i=1}^{n} P(X_i|P_a(X_i)) \qquad (1)$$

Actually, this decomposition of a global function, undergone by a product of local terms exclusively depending on the considered knot and its relative parents in the graph, is a fundamental property of BN. It is on the basis of the early works pertaining to the development of inference algorithms, that the probability of any model variable could be calculated from observation, even partial, of the other variables. This problem was proved NP-hard, but ended in various algorithms which can be likened to the information distribution methods in a graph. As can be notes, these methods apply the notion of conditional probability, along with the theorem of Bayes, which allows to calculate the probability Xj from Xi, and vice versa, knowing that $P(Xi|Xj)$ [16].

The number of all BN possible structures has been shown to ascend sharply as a super-exponential on the number of variables. Indeed, Reference [24] derived the following recursive formula for the number of Directed Acyclic Graph (DAG) with n variables:

$$r(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-1)} r(n-i) = n^{2^{O(n)}} \qquad (2)$$

which gives: $r(1)=1$, $r(2)=3$, $r(3)=25$, $r(5)=29281$, $r(10)=4{,}2 \ 10^{18}$

This means that, it is impossible to perform an exhaustive search of all structures in a reasonable time in cases the number of nodes exceeds seven.

3 A New approach for optimizing the number of variables prior to modeling

3.1 Background

It is well recognized that the strategy based on single variable analyses has a very limited value in elucidating the mechanisms involved in complex phenomena [8]. In this respect, our proposed approach is fundamentally a four-step operating multivariate analysis. It starts by calculating a statistical score (test value or p-value) for each variable depicting its relevance to a certain phenomenon. It then, clusters variables according to their association to the studied phenomenon as well as their complementarity. In the third step, a global statistical score is calculated for each cluster of variables, which is a function of the correlation between the variables and their scores. Ultimately, the clusters will be ranked in a decreasing order based on their global score (following a logarithmic transformation in order to have a high score if the score statistic value is low), so that a number of them can be selected (Figure 1).

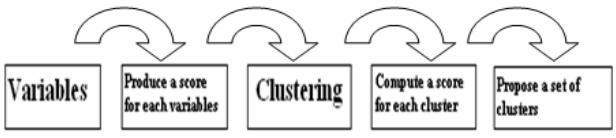


Figure 1. Approach Steps

It is worth mentioning that all the methods and statistical tests that used in this section are available in statistics as well as computer science literature. Noteworthy, however, the novelty lies in their nesting where they have never been built fit before, i.e, in the field of data reduction for learning BN structures.

3.2 Single-variable analysis

Chi-square test can validate assumptions raised regarding a certain property contained in the basis of a concrete case base [9]. It is a widely used test applied to measure the association between categorical variables [19]. For cases involving binary variables (two categories), for instance, the disease status and risk factor in epidemiological studies, the Chi-square is easily calculated.

3.3 Variables' Clustering

"Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters)" [11]. Clustering is an automated process to join related variables into a set in which they are grouped together on the basis of their attributes similar values. There is a variety of algorithms used for clustering, e.g., Generative Models, the Gaussian Mixture Model, C-Means Fuzzy Clustering, Reconstructive Models, K-means...[23]. Clustering can also be achieved based on expert's knowledge in the field [22].

3.4 Fusion of separate-cluster scores

In this junction, one might well wonder how to derive a score for each cluster based on the scores of variables within clusters.

Most of the methods used to combine the pertinent scores to computer science literature, and more specifically to knowledge discovery in database, are those that consist in merging such scores of independent variables as: Average and Maximum (MAX) scores [4] and [21], Sum, Minimum (MIN) and Product scores [12].

Yet, the statistical literature provides several score-combining methods by taking into account the correlations among variables. Among these is the Truncated Product Method (TPM) [27] which combines the correlated tests' p-values, whose algorithm is described below:

Truncated Product Method (TPM) Algorithm

For each cluster of variables, the following steps are to be undertaken:

1: Construct a correlation matrix for variables within the cluster.

2: Calculate the Cholesky matrix *C* for each correlation matrix

3: Choose the scores' maximum value π (p-values) to be selected.

4: Calculate $W_0 = \prod_i^L p_i^{I(p_i \le \pi)}$

Where *L* designates the number of variables in the cluster

5: Put *A=0*

6: Randomly generate *L* independent values from a uniform distribution generating the vector R^*: $u_1^*,.....,u_L^* \in [0,1]$

7: Transform the vector R^* into another vector *R* having the values with equation (2):

$$R = 1 - \Phi\left\{C\Phi^{-1}(1-R^*)\right\} \quad (3)$$

$$//\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz \quad (4) \quad \Phi^{-1}(x) = \int_{x}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz \quad (5)$$

8: Calculate $W = \prod_i^L R_i^{I(R_i \le \pi)}$

9: If $W \le W_0$, then *A=A+1*

10: repeat steps 6 to 9, B times

11: obtain the combined score (p-value) by means of *A/B*.

3.5 Ranking

In this way, Clusters of variables can be ranked on the basis of their scores. If a p-value is used as a score, then ranking is based on an increasing-order scoring (smaller p-values are indicative of higher significance). Most often, the score is calculated as the logarithmic transformation $-\text{Log}_{10}$(p-value) just as a high score value implies a high degree of significance (association). In this case, the score ranking will be done on decreasing-order.

3.6 Selection of a phenomenon closely-related clusters

The purpose of this step is to select the appropriate variables' clusters closely associated to the phenomenon. Actually, there exist numerous methods for selecting most influential variables available in statistical and computer science literature. Nevertheless, to our knowledge, few are those methods that deal with selecting clusters of variables. In this respect, we propose a method inspired from [14], described below.

We consider that among the *k* ranked scores obtained for each variables' cluster, the first *r* will be selected by applying the following steps.

1: Scores $S_1, S_2,..., S_k$ are used to compute sum statistics as follows: $T^i = \sum_{j=1}^{i} S^j$ where *i* varies from *1* to *k*.

2: P-values $(P_T^1,...,P_T^k)$ are estimated from the empirical distribution of each T^i (data are simulated, from a uniform distribution, for a number of times and the p-value is then estimated by the proportion of T^i values exceeding the observed value T_{obs}), thus:

$$P_T = \frac{Cardinality\left\{T^B \ge T_{obs}\right\}}{B} \quad (6)$$

3: The method selects the first *r*, where *r* corresponds to the first cluster in which a decrease in p-value is initially witnessed $P_T^{(i+1)} \le P_T^{(i)}$

that is, $r = \arg\min(P_T^{(i+1)} \ge P_T^{(i)})$.

3.7 Precepts of genetics and Experimentation

Single Nucleotide Polymorphisms (SNP), indicated in genetics, are variations of a single basic pair (of the same sort) of human genome among individuals. These variations are very frequent (1/1000 pairs of bases in the human genome). The SNP represent 90 % of all the human genetic variations, and SNP with an allelic frequency superior or equal to 1 %, are present in all 100 in 300 basic pairs on average in the human genome, where 2 SNP out of 3 substitute the cytosine with the thymine [28].

Generally speaking, the SNP are bi-allelic (a, A). Every individual will be carrier, at the level of an SNP, of one of the three possible genotypes:

- both homozygous genotypes (aa and AA)
- The heterozygous genotype (aA or Aa imperceptible one of the other). [28]

A gene is a sequence of deoxyribonucleic acid (DNA) which specifies the synthesis of a chain of polypeptide or a ribonucleic acid functional (ARN). We can also define a gene as a unit of genetic information. Therefore, we can say that the DNA is the support of the genetic information. Indeed, it can be considered as a book, an architectural plan of the alive, which directs, dictates the construction of the main constituents and cellular builders which are the proteins. The genotype of an individual (along with that of the animal, plant, bacteria or other) is the sum of the genes which it possesses. As for the phenotype, it corresponds to the sum of the morphological, physiological or behavioural characters which are recognizable from the outside. Consequently, two individuals can have the same genotype but not necessarily the same phenotype, depending on the conditions of expressions of the genes which confer a recognizable, discernible aspect [28].

Genome wide-association studies are which geneticists assess the association of thousands of molecular markers with a disease phenotype. The traditional way of analyzing the data consists in computing the chi-square association tests and the corresponding p-value for each marker. Then, those with the weakest p-values are selected as indicative of interesting genome region, as has been applied on 213 Canadian patients suffering from schizophrenia and 241 Canadian controls, both genotyped for 164 SNPs on chromosome 13.

Eventually, the reached database has been a text file formatted as follows:

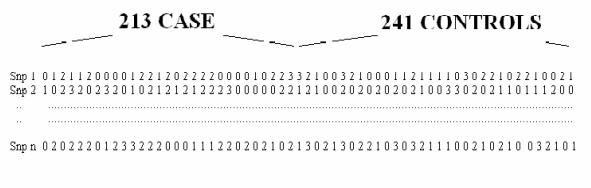


Figure 2. Data base Format.

Where 0: corresponds to the *aa* genotype,

1: corresponds to the *Aa* or *Aa* genotype

2: corresponds to the *AA* genotype

3: corresponds to missing data

Actually, our objective has been to select genomic regions (clusters of variables) which are most significantly associated to the disease (schizophrenia).

3.7.1 Data-Processing Steps

- Calculating a score corresponding to each variable (SNP) which is, in our case, the p-value derived from the chi-square test statistic.
- Variable clustering: we cluster the variable according to the genetic experts' suggesting that a gene might well represent a cluster of SNPs.
- Combine scores from each cluster using the different relevant strategies proposed in Sub-Section 3.4 and compare results.
- Rank the various clusters according to their scores.
- Select clusters involved in the disease (schizophrenia) using the approach described in Sub-section 3.6.

3.7.2 Results of different fusion-scoring methods

For the purpose of comparing the p-value combining methods, we have relied on the reference: the region "G72" described by [5], as being the region responsible for Schizophrenia.

TABLE I below depicts the results achieved via both the TPM as well as the MIN Methods. Actually, the discovered genes revealed by these two methods turn out to be very similar and are contained in the "G72" region.

TABLE I. BOTH METHODS' ACHIEVED RESULTS

	Rank	Gene name	Region	Score	Stati Sum	P-value
MIN	1	FOX01	151	1.33	1.33	0.70
	2	NARG1L	140-141	1.20	2.58	0.63
TPM	1	NARG1L	140-141	1.52	1.52	0.15
	2	FOX01	151	1.09	2.62	0.10

In terms of complexity, it clearly appears that the algorithm using the MIN method appears to be the more appropriate choice. Yet, to check the reliability of both methods' results, we turn to study the empirical distribution of the observed minimum p-value ($P_{\min}^{obs}$) via the Monte Carlo simulations, whose principle is the following:

We simulate B times the data by calculating each time relevant minimum p-value ($P_{\min}^{i}$) and, ultimately, we calculate the overall p-value corresponding to each step through the formula below:

$$P_G = \frac{cardinality\left\{P_{\min}^{(i)} \le P_{\min}^{obs}\right\}}{B}. \qquad (7)$$

Where $P_{\min}^{obs}$ represent P_T^i (already explained in the 3.6 sub-section) of the first cluster in which a decrease in p-value.

Thus, the p-value pertinent to the overall process using TPM is equal to 0.09, while it has been 0.41 with respect to the MIN method. One can conclude that the results achieved by means of the TPM turn out to be more significant and that this method is preferably convenient to a subsequent work.

3.7.3 Discussion

On applying our innovative approach, we have been able to successfully identify the most significant genes involved in the Schizophrenia disease. Indeed, our attained results have turned out to be conforming to, and to agree with, those published by the specialists in genetics. Above all, we have managed to exclude those genes having no implication, or a very weak relationship, with this illness. As a matter of fact, to our knowledge, the genes eliminated from the information system, subject of study, have not been mentioned by any specialized literature publication pertinent to the genetics field as being involved in the Schizophrenia disease.

Added to this, we have actually been able to reduce the number of initial variables, necessary to this study, from 165 (164 SNP plus one phenotype variable) into four (3 SNP plus a phenotype variable). Thus largely reducing the algorithmic complexity of applying the BN structure, as the number of possible graphs has gone down from $r(165)\approx10^{406}$ to $r(4)=576$, without any loss in data information. Yet, we reckon it necessary for our approach to be tested and applied to another data base of similar context, for the sake of validity consideration purposes of the achieved results.

It is worth highlighting, however, that data pertaining to the studies of complex genetic illnesses appear to fit particularly well to our devised approach, seeing the fact that a diverse number of genes would not be involved in certain genetic illnesses. Hence, it could be set, as a proposal for a prospective future research to test this model approach on another domain area data base. Actually, our targeted purpose is to check out the achieved results, above all on those cases where the total numbers of variables appear to be really implicated in the context of certain phenomena.

4 Conclusions and suggestions for further research

Our study has defined a novel and appropriately-useful approach for filtering the number of variables in respect of their degree of implications in a given phenomenon. This proposed approach enables to identify clusters of variables that are most frequently involved in a given phenomenon using several steps. This has been illustrated through a simple pertaining to a genetic study on schizophrenia. The comparison of the proposed approach, as a whole, with other similar methods available will be the objective of a prospective publication.

In a future research, we intend to present a new multi-purpose heuristics designed for learning BN structure. Such a process, aimed at reducing the search space for the possible graphs, should be able to combine with the already-existing algorithms and the classic metric-score methods of BN learning. Bound for learning a Bayesian network structure, this heuristic has actually been built on the formalism introduced by [15], called Multi-Entity Bayesian Networks (MEBN). The MEBN formalism unifies the first-order logic jointly with the probability theory. It contains fragments dubbed MFrags, which represent the joint distribution of a subset of variables. Our principle will be based on the fact that the complexity of learning Bayesian network structure is exponential giving the exponential, increase in the number of variables. Hence, the urgent need for methods allowing to learn the structure with all its contained variables, even when the number of variables is too large. The solution that we reckon to propose would be based on the modulation of learning structure: each cluster has its properly-allotted learning structure, before forming the final single structure encompassing all the variables.

A structure will be devised for a benchmark of databases that depicts the dominating relationships between the selected variables and the phenomenon. Noteworthy, Multi-Entity Bayesian Networks, despite the interest of their use in respect of the complex classical structure-learning algorithms, remain still liable to demonstration.

5 References

[1] V. Nefian, "Learning SNP using embedded Bayesian Networks," IEEE Computational Systems Bioinformatics Conference, 2006.

[2] L. Bouchaala, A. Masmoudi, F. Gargouri and A. Rebai, "Improving algorithme for structure learning in Bayesian Networks using a new implicit score," Expert Systems with Application, 37, 5470-5475, 2010.

[3] G. Cooper and E. Hersovits, "A Bayesian method for the induction of probabilistic networks from data," Machine learning, 9, 309-347, 1992.

[4] M. L. Damian and F. H. Donald, "Combining multiple scoring systems for target tracking using rank-score characteristics," Information Fusion, 10, 124-136, 2009.

[5] S. Detera-Wadleigh and F. McMahon, "G72/g30 in schizophrenia and bipolar disorder: review and meta-analysis," Biological Psychiatry, 60(2): 106-114, 2006.

[6] P. Dempster, N. Laird and B. D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," Journal of the Royal Stat Soc B 39: 1-38, 1977.

[7] O. Francois, and P. Leray, "Evaluation d'algorithmes d'apprentissage de structure pour les réseaux bayésiens," In Proceedings of 14ème Congrès Francophone Reconnaissance des Formes et Intelligence Artificielle, RFIA, pages 1453-1460, Toulouse, France, 2004.

[8] M. Geudj, J. Wojcik, D. Robelin, M. Hoebeke, M. Lamarine and G. Nuel, "Detecting Local High-Scoring Segments: a First-Stage Approach for Genome-Wide Association Studies," Statistical Applications in Genetics and Molecular Biology, Vol. 5, Iss. 1, Article 22 2006.

[9] C. Herman and E. L. Lehman, "The use of Maximum Likelihood Estimates in chi-square tests for goodness of fit," The annals of Mathematical Statistics volume 25, Number 3, 579-586, 1954.

[10] K. Hwang, B. H. Kim and B. T. Zhang, "learning hierarchical Bayesian Networks for large-scale data analysis," In ICONIP: 670-679, 2006.

[11] K. Jain, M. N. Murty and P. J. Flynn, "Data clustering: A review," ACM Computing Reviews, 264-323, 1999.

[12] Jain, K. Nandakumar and A. Ross, "Score normalization in multimodal biometric systems," Pattern Recognition, volume 38 Issue 12, Pages 2270-2285, Dec 2005.

[13] P. Judea and V. Tom, "A theory of inferred causation," In James Allen, Richard Fikes and Erik Sandewall, editors, KR' 91: Principles of knowledge representation and reasoning, pages 441-452, San Mateo, California, 1991.

[14] S. Karlin and S. Altshul, "Applications and statistics for multiple high-scoring segments in molecular sequences," Proceedings of the National Academy of Science USA 90, 5873-5877, 1993.

[15] K. B. Laskey, "MEBN: A language for first-order Bayesian knowledge bases," Artificial Intelligence, 172, 140-178, 2007.

[16] P. Leray, "Réseaux Bayésiens: apprentissage et modélisation de systèmes complexes," habilitation à diriger les recherches, Université de Rouen, 2006.

[17] R. Mourad, C. Sinoquet and P. Leray, "Learning hierarchical Bayesian Networks for genome-wide association studies," In 19th International Conference on computational statistics, (COMPSTAT): 549-556, 2010.

[18] R. Mourad, C. Sinoquet and P. Leray, "A hierarchical Bayesian Network approach for linkage disequilibrium modelling and data dimensionality reduction prior to genome-wide association studies," BMC Bioinformatics, ISSN 1471-2105, 2011.

[19] P. Naim, P. H. Wuillemin, P. Leray, O. Ponnet and A. Becker, "Réseaux Bayésiens," Eyrolles, Paris, 3 editions, 2007.

[20] R. E. Neapolitan, "Learning Bayesian Networks," Newyork, NY, USA: Prentice Hall 2003.

[21] H. N. Parkash and D. S. Guru, "Offline signature verification: An approach based on score level fusion," International journal of computer applications, 0975-8887, Article 10, No.18, 2010.

[22] A. Peter, G. Patrick and F. Geett, "On the potential of domain literature for clustering and Bayesian Networks leraning," Proceeding of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 405-414, 2002.

[23] X. Rui, and C. W. Donald, "Clustering," IEEE Press/Wiley, oct 2008.

[24] R. W. Robinson, "Counting unlabeled acyclic digraphs," Combinatorial Mathematics, 622, 28–43, 1977.

[25] Y. Zhang and L. Ji, "Clustering of SNPs by structural EM algorithm," International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing: 147-150, 2009.

[26] Y. Shulin and K. Chang, "Comparison of score Metrics for Bayesian Networks Learning," IEEE Transactions on Systems, Man and Cybermeties-part A: Systems and Human, 32(3), 419-428, 2002.

[27] D. Zaykin, L. Zhivotovsky, P. Westfall and B. Weir, "Truncated product method for combining P-values," Genet Epidemiol, 22(2), 170-85, Feb 2002.

[28] D. W. Watson, T. A. Baker, S. P. Bell, A. Gann, M. Levine, R. Losick, "Molecular Biology of the gene," Distributed in conjunction with Benjamin cummings, 841 pp, ISBN 978-080539592-1, 2008.

Improved Interpretability of the Unified Distance Matrix with Connected Components

Lutz Hamel and Chris W. Brown

Abstract— **Self-organizing maps have been adopted in many fields as the data visualization method of choice. The unified distance matrix is the *de facto* standard for evaluating and interpreting self-organizing maps. In large, high-dimensional problems clusters can be difficult to identify in the plain unified distance matrix. Here we introduce an enhanced version of the unified distance matrix in which clusters are easier to see and interpret. In this enhanced version we view the self-organizing map as a planar graph where the clusters are connected components of this graph. Using the transitive properties of connectedness and exploiting the fact that each component has a minimal node where the gradient on the unified distance matrix is equal to zero we can transform these connected components into stars with the minimal node as the internal node. In order to avoid unnecessary fragmentation of the components we apply a kernel based smoothing algorithm to the unified distance matrix. Our enhanced unified distance matrix is then the smoothed original unified distance matrix with the star components overlaid. The result is an easily interpretable self-organizing map. We perform a number of experiments on synthetic as well as real-world data that highlight the increased visual power of this enhanced unified distance matrix.**

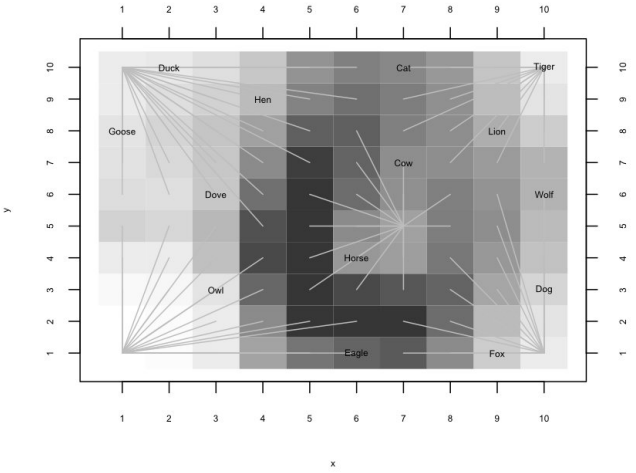


Fig. 2. An enhanced unified distance matrix.

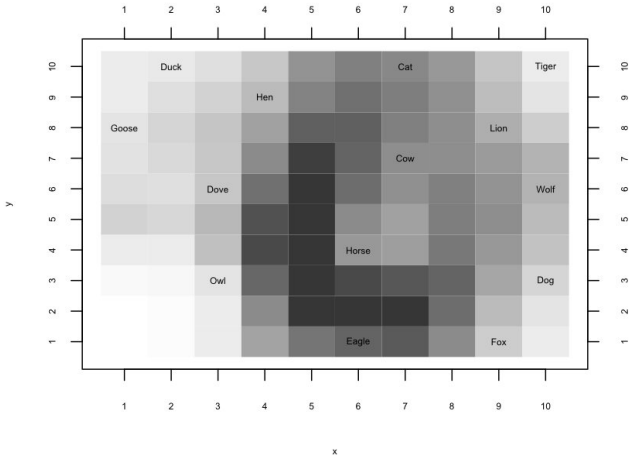


Fig. 1. A unified distance matrix.

I. Introduction

Data visualization is a powerful method to get an intuitive understanding of the data set at hand. Self-organizing maps (SOM) have been adopted as the method of choice for data visualization in many fields [1] [2]. The unified distance matrix (UMAT) is the de facto standard for evaluating and interpreting self-organizing maps. During the training phase of a SOM the values of its neural elements are computed in such a way that elements next two each other on the two-dimensional map are also close to each other in the space spanned by the attributes of the training data. In this way the two-dimensional map represents a topology preserving image of the high-dimensional training data. The UMAT visualizes the relative distances between the neural elements in training data space using a heat map: lighter colors represent neural elements that are close together in training data space and darker colors represents neural elements that are further apart. See Figure 1 for an example of an UMAT. The example is due to Kohonen [2] where each animal is described by a 10-dimensional feature vector with attributes such as the number of legs, if fur is present or not, and if the animal can swim. In Figure 1 we can clearly identify two strong clusters; one on the left side of the map clustering the birds and one on the right side of the map clustering the mammals. However, it turns out that there are more clusters that can only be seen on the UMAT by a very careful inspection: the bottom left corner represents birds of prey whereas the top left corner represents birds that do not hunt. On the right side of the map we have clusters of hunters with fur and in the middle we have animals that are not hunters with hooves.

Lutz Hamel is with the Department of Computer Science and Statistics, University of Rhode Island, Kingston, RI 02881, USA (email: hamel@cs.uri.edu).

Chris W. Brown is with the Department of Chemistry, University of Rhode Island, Kingston, RI 02881, USA (email: cbrown@chm.uri.edu).

In this paper we introduce connected components as a way to improve the visibility of such clusters. Figure 2 is the UMAT appearing in Figure 1 with the connected components of the map overlaid. Notice that the clusters become immediately visible as the eye is guided toward seeing the clusters. These guides proved essential in our own work dealing with high dimensional spectroscopic data where data sets with hundreds of attributes are not uncommon [3], [4]. In order to obtain reasonable interpretations of these data via UMATs we had to construct large SOMs with thousands of neural elements; an approach not unlike Ultschs approach to constructing maps with emergent SOM [5]. The connected components proved extremely helpful when interpreting these large maps.

The paper is structured as follows. Section II provides a very brief overview of the standard SOM algorithm. In section III we develop connected components and describe how they are implemented in our current library. In section IV we describe a number of experiments. The first set of experiments is based on Ultsch's FCPS library and the second set of experiments is based on real world data sets we used in some of our work. We discuss related work in section V and we conclude with section VI providing some observations and pointers to further research.

II. Self-Organizing Maps

Self-organizing maps [2] were introduced by Kohonen in 1982 and can be viewed as tools to visualize structure in high-dimensional data. Self-organizing maps are considered members of the class of unsupervised machine learning algorithms, since they do not require a predefined concept but will learn the structure of a target domain without supervision.

Typically, a self-organizing map consists of a rectangular grid of neural elements. Multidimensional observations are represented as vectors. Each neural element in the self-organizing map also consists of a vector called a reference vector. The dimensions of the reference vectors on the map match the dimensionality of the observations. The goal of the map is to assign values to the reference vectors on the map in such a way that all observations can be represented on the map with the smallest possible error. However, the map is constructed under constraints in the sense that the reference vectors cannot take on arbitrary values but are subject to a smoothing function called the neighborhood function. During training the values of the reference vectors on the map become ordered so that similar reference vectors are close to each other on the map and dissimilar ones are further apart from each other. This implies that similar observations will be mapped to similar regions on the map. Often reference vectors are referred to as centroids, since they typically describe regions of observations with similarities.

The training of the map is carried out by a sequential regression process, where $t = 1, 2, \ldots$ is the step index. For each observation $\mathbf{x}(t)$ at time t, we first identify the index c of some reference vector which represents the best match in terms of Euclidean distance by the condition,

$$c = \underset{i}{\operatorname{argmin}} \|\mathbf{x}(t) - \mathbf{m}_i(t)\|. \tag{1}$$

Here, the index i ranges over all reference vectors on the map. The quantity $\mathbf{m}_i(t)$ refers to the reference vector at position i on the map at time step t. Next, all reference vectors on the map are updated according to the following rule where index c is the reference vector index as computed above,

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}[\mathbf{x}(t) - \mathbf{m}_i(t)]. \tag{2}$$

Here h_{ci} is the neighborhood function that is defined as follows,

$$h_{ci} = \begin{cases} 0 & \text{if } |c-i| > \beta \\ \eta & \text{if } |c-i| \leq \beta \end{cases} \tag{3}$$

where $|c - i|$ represents the distance on the map between the best matching reference vector at position c and some other reference vector at position i, β is the neighborhood distance and η is the learning rate. It is customary to express η and β also as functions of time t. The above computation is usually repeated over the available observations many times during the training phase of the map. Each iteration is called a training epoch.

One of the advantages of self-organizing maps is that they have an appealing visual representation as the 2-dimensional unified distance matrix as seen in Figure 1. Each square in the map represents a reference vector. As before, the colors on the map represent the relative distances between reference vectors: light colors indicate short distances and dark colors indicate long distances. Contiguous areas of light colors represent strong clusters.

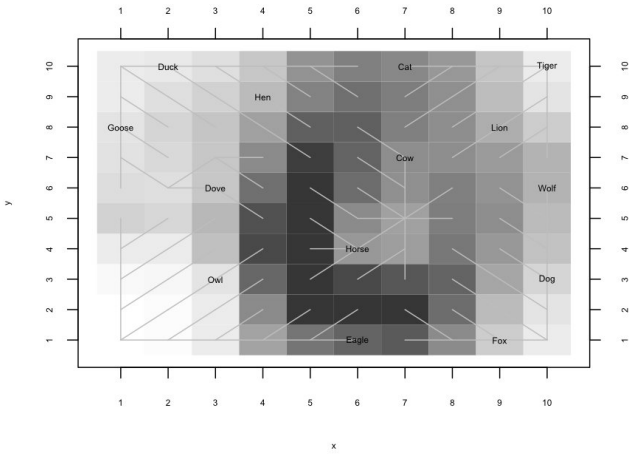


Fig. 3. Connected components of a SOM.

III. Connected Components

A. Development

Connected components are a graph theoretical concept and can be defined as follows,

Definition. *A connected component of an undirected graph is a subgraph in which any two vertices are connected to each other by a path [6].*

In our case, we view the SOM neural elements as the vertices of an undirected planar graph. The edges in this graph are defined as follows based on the UMAT: a node in the graph is connected to a neighboring node along the maximum gradient on the UMAT. We do not add edges for nodes where the gradient is equal to zero. For the animal example given in the introduction this construction gives rise to a graph with five connected components shown in Figure 3.

It is easy to see that connectedness is a transitive relation. That is, if node A is connected to node B and node B is connected to node C, then node A is connected to node C via the path from A to B and from B to C. Exploiting this transitivity property and the fact that there exists a path from every node in a connected component to the node where the gradient is equal to zero, we can transform each component into a star [7] with the node where the gradient is equal to zero at the center. This transformation gives rise to the components as we had shown in Figure 2.

The visual power of this representation derives from the fact that clustering relationships between labeled neural elements can be directly read from the connected components by again exploiting the transitivity of connectedness. Consider for example the cluster in the lower left corner in the UMAT of Figure 1. It is visually difficult to see that owl and eagle indeed belong to the same cluster. Now compare this to the UMAT with the overlaid connected components shown in Figure 2. Here it is immediately evident that owl and eagle belong to the same cluster: owl and eagle are connected by a path in the connected component via the internal node of the star.

B. Implementation

We have implemented this enhanced version of the unified distance matrix as a package in R [8] (the code is available by request; we intend to release it as a publicly available package as part of CRAN). At the core of the implementation is the function `find.internal.node` which, given the map coordinates of a neural element, will find the corresponding internal node of the associated star. Table I shows the pseudo code for this function. Given a position on the map this function first searches the adjacent nodes for the minimal UMAT value using the function `find.min`. If an adjacent node with a smaller UMAT value than the value of our current node exists and if this UMAT value is smaller than the UMAT values of all other adjacent nodes, then that node lies along the maximum gradient of the surface and we make this node our new current position. If no such node exists, then the gradient at our current position is zero and we are at an internal node.

The remainder of the implementation is straightforward; we first draw the traditional UMAT, we then iterate over all the neural elements of the map and compute their corresponding internal nodes with the `find.internal.node` function. Once this is complete we draw the paths from each neural element to its corresponding internal node giving rise to enhanced UMAT representations as shown in Figure 2.

TABLE I
PSEUDO CODE TO FIND THE INTERNAL NODE OF A CONNECTED COMPONENT.

```
function find.internal.node(int x,
                            int y,
                            real umat[xdim,ydim])

returns (int cx, int cy)

// x and y are the map coordinates of our current node
// umat is the unified distance matrix
// xdim and ydim are the dimensions of the map
// cx and cy are the map coordinates of the internal
// node of the star

begin
    // find the smallest value of umat in our immediate
    // neighborhood, including our current position.

    (minx,miny) = find.min(x,y,umat)

    // if minx and miny are our current position then
    // the gradient at our current position is zero
    // otherwise move to the new position along the
    // maximum gradient and call ourselves
    // recursively.

    if (minx == x and miny == y) then begin
        cx = minx
        cy = miny
        return
    end else begin
        (cx,cy) = find.internal.node(minx,miny,umat)
        return
    end
end
```

We found that smoothing the unified distance matrix before applying our algorithm to find the connected components results in larger, homogeneous components making the map easier to interpret. In our implementation we use a two-dimensional Gaussian kernel smoothing function where the bandwidth of the Gaussian controls the level of smoothing: the larger the bandwidth the more aggressive the smoothing. Picking the right level of smoothing then becomes a trade-off between the size of the connected components and their homogeneity.

IV. EXPERIMENTS

We discuss four different experiments using our enhanced version of the UMAT. The first two experiments are based on artificial data sets from Ultsch's Fundamental Clustering Problem Suite (FCPS) [9] and the last two experiments are real world spectroscopy data sets we have analyzed using self-organizing maps. The hallmark of the latter data sets is that they are high-dimensional. Our petroleum data set has 257 attributes and our bacteria data set has 300 attributes. This high dimensionality forces us to consider large maps, in this case, maps with up to a thousand neural elements.

In each of these experiments we compare the original UMAT with our enhanced version of the UMAT and we point out that the enhanced version of the UMAT facilitates the discovery and evaluation of clusters. Unfortunately, as with many graphical artifacts, their comparative evaluation is

highly subjective. However, it is our hope that this empirical study will convey some of the advantages of the enhanced UMAT representation, as we perceive them.

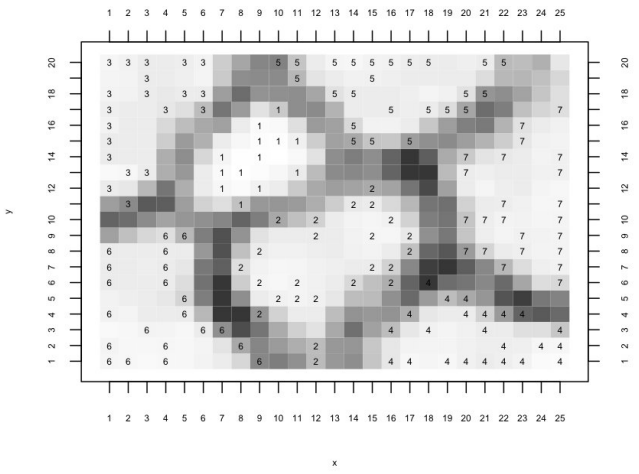


Fig. 4. UMAT for the Hepta data set.

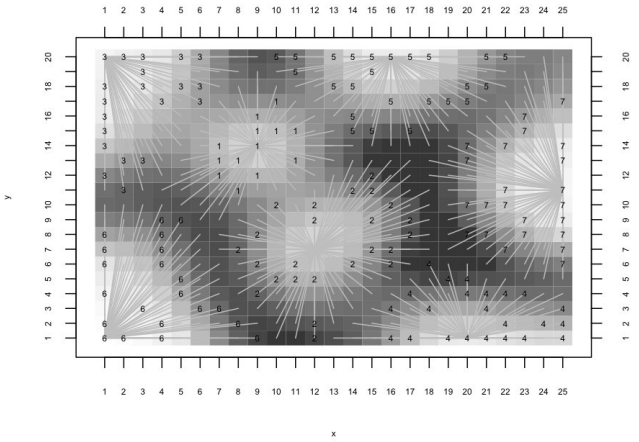


Fig. 5. Enhanced UMAT for the Hepta data set.

A. Experiment 1: Hepta

Hepta is the first FCPS data set. It is a three dimensional data set representing seven non-overlapping classes with different variances. The data set has 212 observations. Since the classes are non-overlapping we would expect that any clustering procedure, including SOM, would display seven individual clusters. Figure 4 shows the seven clusters in the traditional UMAT display and Figure 5 shows the clusters in our enhanced UMAT display. Since the seven clusters are completely separable, the traditional UMAT and our enhanced version show the clusters very nicely. The differences in the cluster layouts between UMAT and our enhanced version are due to the smoothing in our enhanced version.

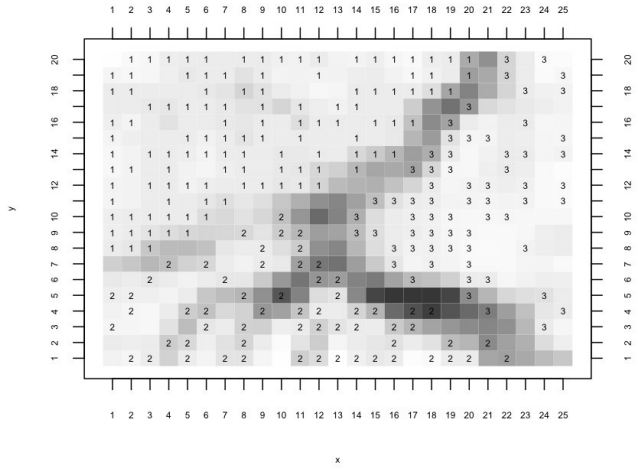


Fig. 6. UMAT for Lsun data set.

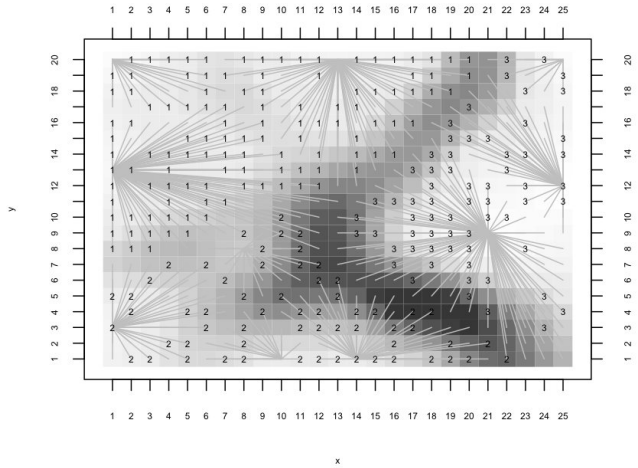


Fig. 7. Enhanced UMAT for the Lsun data set.

B. Experiment 2: Lsun

For our second experiment, we used the Lsun data set from Ultsch's problem suite. This is a two-dimensional data set with 400 observations containing three clusters. The hallmark of this data set is that the clusters have different variances and different inter-cluster distances. Figure 6 shows the UMAT for this data set. In this representation the border between cluster 1 and cluster 2 in the bottom left corner of the map is difficult to detect due to the fact that they lie very close together in the data space. However, in our enhanced UMAT the border is easily seen due to the component structure. This can be seen in Figure 7 even though the clusters themselves are composed of multiple connected components.

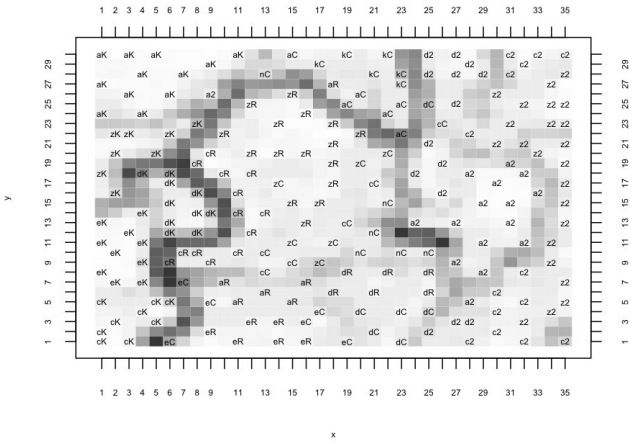


Fig. 8. UMAT for petroleum data set.

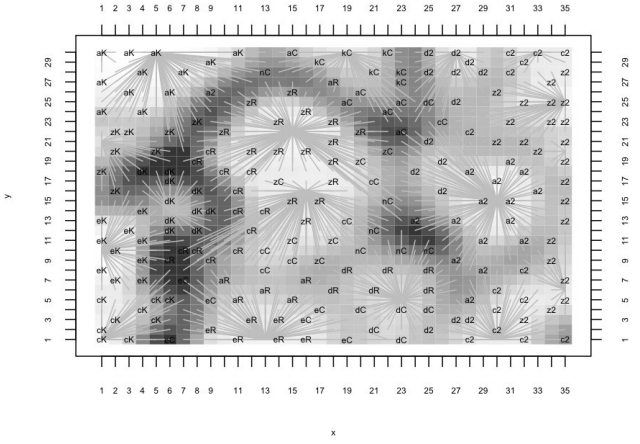


Fig. 9. Enhanced UMAT for the petroleum data set.

C. Experiment 3: Petroleum

Our petroleum data set consists of 235 observations of different petroleum products from various regions of the world. Each observation represents a spectrum in the infrared range and is labeled by a two letter label rP where r is a region identifier drawn from the set of region identifiers {a,c,d,e,k,n,z} and P is a product identifier: 2 – #2 fuel, K – kerosene, C – crude, and R – residue. Because of the large number of dimensions in the training data (257) we chose a map size with 1050 processing elements.

During our analysis we were interested to see if regions and petroleum products clustered based on the spectral information. Figure 8 shows the UMAT of our analysis. It turns out that there is significant clustering in this data set but it is difficult to see this here. Figure 9 shows the enhanced UMAT of the same data set. The clusters are easily visible. What this analysis shows is that both region of origin and fuel type carry significant identifying signatures in their corresponding spectra and this is easily seen in our enhanced UMAT.

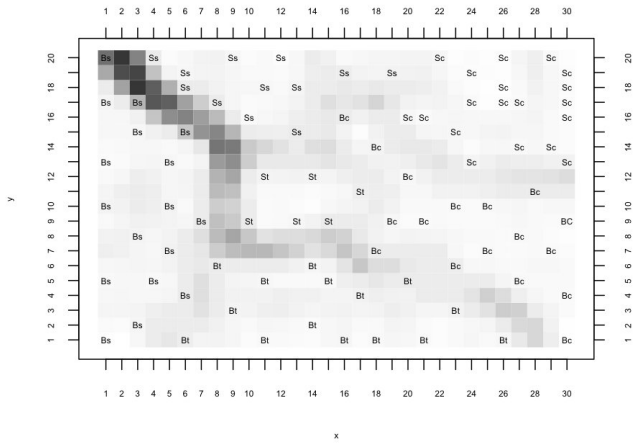


Fig. 10. UMAT for bacteria data set.

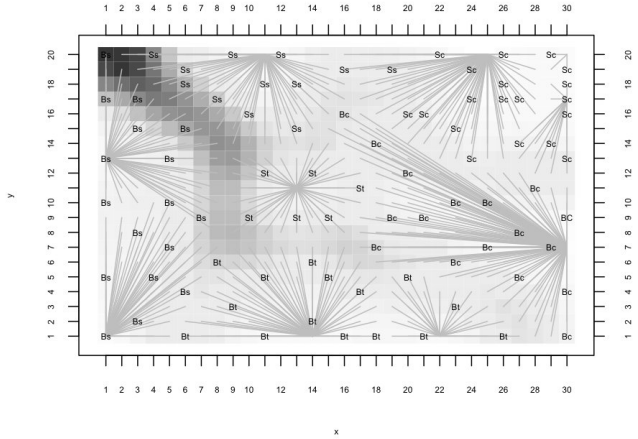


Fig. 11. Enhanced UMAT for the bacteria data set.

D. Experiment 4: Bacteria Spores vs. Vegetative States

Our final experiment concerns observations of three bacteria: Bacillus cereus, Bacillus thuringiensis, and Bacillus subtilis. For each bacterium we have spectral observations in its spore state as well in its vegetative state. Overall we have 89 observations. Each spectrum is composed of 300 wavelengths. The goal of this investigation was to see if the spectra of different bacteria are identifiable and if we can also distinguish between their spore and vegetative states. We labeled each observation with a two-letter label Qb where Q denotes the state of the bacterium, S for spore and B for the vegetative state, and b denotes the type of bacterium: c – cereus, t – thuringiensis, and s – subtilis. We analyzed the data with a 600 element SOM.

Figure 10 shows the traditional UMAT. Just as in the case of the petroleum data, the clusters are somewhat difficult to see even though the data clusters extremely well. Figure 11 is our enhanced UMAT and here the clusters are immediately visible due to the connected components. We are able to identify clusters of individual bacteria in the vegetative state along the bottom of the map and we can identify clusters of individual bacteria as spores in the top half of the map. This means that the spectra of the individual bacteria are characteristic with respect to whether they are spores or in the vegetative state as well as to their genus identity.

V. Related Work

The paper by Vesanto [10] provides a general overview of visualization techniques for self-organizing maps. Recently, a number of graph-based visualization techniques for self-organizing maps have been developed. From a visual perspective the work that is most closely related to our own is the work by Pölzlbauer *et. al* [11]. In their visualization they also plot gradient-based components on top of the SOM two-dimensional map, however, their gradients are derived from highly correlated groups of attributes in the training data. Thus, the meaning of the components is different from ours where a component simply connects all the neural elements that lie close together in data space. In [12] Tasdemir and Merenyi describe their graph-based visualization. In this visualization the graph edges overlaid on the SOM grid are color coded to convey more detailed information on the topology of the training data. What distinguishes our approach from these approaches is that we do not stray from the *de facto* standard interpretation of self-organizing maps via the unified distance matrix but instead enhance this interpretation.

VI. Conclusions and Further Work

We have shown that overlaying connected components based on the gradient information in the unified distance matrix improves the identification and interpretation of clusters on the self-organizing map. This was especially true in the case of our high-dimensional real world data.

We have found that smoothing the UMAT before applying our connected component identification algorithm results in less fragmented subgraphs. Given that the user can control the level of smoothing that is applied to the UMAT, constructing connected components then becomes a trade-off between the size of the components and their homogeneity. The more smoothing we apply, the larger the connected components, but this implies an increased likelihood that clusters will be merged that contain observations from different classes. Given that the connected components provide us with a tractable way to group observations into clusters, we envision that we can perform the optimization of map components via smoothing automatically. We can view this as a model fitting step. Highly fragmented components represent an overfit model responding to minor, perhaps random, nuances in the UMAT. Large, non-homogeneous components represent an underfit model; a model that is not refined enough to display the true clustering of the data. The optimization then is a model fitting step that attempts to find just the right trade-off between size and homogeneity of the components.

Furthermore we like to investigate why certain clusters consistently have multiple connected components regardless of the smoothing. We would like to understand if this is an artifact of the UMAT construction or if this in fact does describe topological features in the underlying data space.

Note

Publication format restrictions forced us to display all figures in gray scale. The actual displays in R are in color. A color version of this paper is available at the website: http://homepage.cs.uri.edu/faculty/hamel/pubs.

References

[1] A. Ultsch, "Self-organizing neural networks for visualization and classification", *Proc. Conf. Soc. for Information and Classification*, 1993, pp. 307-313.
[2] T. Kohonen, *Self-Organizing Maps*, 3rd ed., Springer, 2001.
[3] K. Judge and C. W. Brown and L. Hamel, "Sensitivity of Raman Spectra to Chemical Functional Groups," *Appl. Spectrosc.*, Vol. 62, pp. 1221-1225, 2008.
[4] K. Judge and C. W. Brown and L. Hamel, "Sensitivity of Infrared Spectra to Chemical Functional Groups," *Anal. Chem.*, Vol. 80, pp. 4186-4192, 2008.
[5] A. Ultsch, "Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series," *Kohonen Maps*, Elsevier Science, pp. 33-46, 1999.
[6] K. H. Rosen, *Discrete Mathematics and its Applications*, McGraw-Hill, New York, 2003.
[7] F. Harary, *Graph Theory*, Addison-Wesley, Reading, MA, 1994.
[8] R Development Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.
[9] A. Ultsch, "Clustering with SOM: U*C", *Proc. Workshop on Self-Organizing Maps*, 2005, pp. 75-82.
[10] J. Vesanto, "SOM-based data visualization methods", it Intelligent Data Analysis, Vol. 3, pp. 111-126, 1999.
[11] G. Pölzlbauer and M. Dittenbach and A. Rauber, "Gradient visualization of grouped component planes on the SOM lattice", WSOM05, pp. 331-338, 2005.
[12] K. Tasdemir and E. Merenyi, "Exploiting data topology in visualization and clustering of self-organizing maps", *IEEE Trans. Neural Netw.*, Vol. 20, pp. 549-562, Apr. 2009.

On Selecting The Number Of Bins For A Histogram

Sai Venu Gopal Lolla, Lawrence L. Hoberock
School of Mechanical and Aerospace Engineering
Oklahoma State University, Stillwater OK 74078

Abstract— *Histograms are widely used in exploratory data analysis for graphically describing datasets. This paper presents a new method for selecting the number of bins to be used for constructing a histogram for a given dataset. The improved performance of the proposed method is compared to the performances of methods proposed by Sturges, Scott, Freedman et al., Shimazaki et al., and Knuth.*

Keywords: histogram; bin selection;

1. Introduction

A histogram is a graphical representation of the frequency distribution of a dataset. Widely employed in exploratory data analysis, a histogram can be treated as a simple non–parametric density estimator. For a given dataset, a histogram can visually convey the information relating to shape, spread, location, modality and symmetry of the distribution of the underlying population, and is well suited for summarizing large datasets [10]. While more sophisticated kernel–based density estimators are available, histograms are widely employed due to the ease and simplicity of construction and interpretation [20], [16]. While histograms are used mainly for visualizing data and obtaining summary quantities such as entropy, the values of such quantities depend upon the number of bins used (or the bin width used) and the location of the bins [7].

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a univariate dataset with probability density function $f(x)$. We follow Martinez et al. [10]: To construct a histogram, an origin for the bins t_0 (also referred to as the anchor) and a bin width h are selected. Selection of these two parameters defines a mesh (position of all the bins) over which the histogram will be constructed. Each bin is represented by a pair of bin edges as $B_k = [t_k, t_{k+1})$, where $t_{k+1} - t_k = h$ for all k. Histograms using varying bin widths are not addressed in this paper. Let c_k represent the number of observations in B_k (bin count for B_k) given by:

$$c_k = \sum_{i=1}^{n} I_{B_k}(x_i) \tag{1}$$

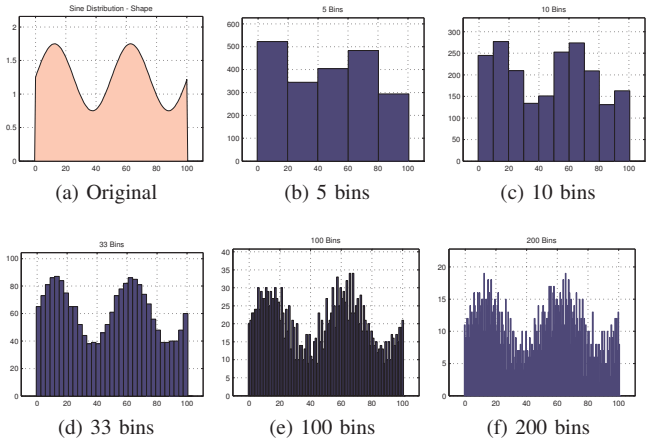
where I_{B_k} is defined as:

$$I_{B_k}(x_i) = \begin{cases} 1 & x_i \text{ in } B_k \\ 0 & x_i \text{ not in } B_k \end{cases} \tag{2}$$

While the density estimate for the underlying population (c_k for all k) satisfies the non–negativity condition necessary for it to be a *bona fide* probability density function, the summation of all the probabilities do not necessarily add to unity. To satisfy that condition, the probability density function estimate, $\hat{f}(x)$, as obtained from a histogram, is defined as:

$$\hat{f}(x) = \frac{c_k}{nh} \qquad \text{for } x \text{ in } B_k \tag{3}$$

This assures that $\int \hat{f}(x)dx = 1$ is satisfied, and $\hat{f}(x)$ represents a valid estimate for the probability density function of the population underlying the dataset.



(a) Original (b) 5 bins (c) 10 bins

(d) 33 bins (e) 100 bins (f) 200 bins

Fig. 1
ORIGINAL DISTRIBUTION AND SEVERAL HISTOGRAMS FOR A DATASET (≈ 2000 POINTS)

The information relating to shape, modality, symmetry and summary quantities estimated using a histogram will depend on the values that c_k (and $\hat{f}(x)$) assume, which in turn depend upon the parameters t_0 and h.

While histograms are commonly constructed using $t_0 = \min(X)$, it is known that modifying this parameter can sometimes cause a rather drastic change in the values assumed by c_k [21]. Simonoff et al. [16] provide a method to quantify the effects of changing the parameter t_0 during the construction of a histogram. However, in the work herein, we use $t_0 = \min(X)$.

A common method to determine bin width h is:

$$h = \frac{\max(X) - \min(X)}{m} \tag{4}$$

where m is the number of bins. From (1), (2), (3) and (4) it can be seen that the number of bins used to construct a histogram will influence c_k (and $\hat{f}(x)$) and any further information derived from them. Consider the following two extreme cases: (1) Using only one bin ($m = 1$) will cause all the data points in X to map to that bin, and information

Datafile	# of Datasets	# of Data points
DF–1	12	≈ 500
DF–2	12	≈ 1000
DF–3	12	≈ 2000
DF–4	12	≈ 5000

Table 1
DATAFILES USED FOR TESTING

Dataset	Distribution	Dataset	Distribution
DS–1	Uniform	DS–7	Gamma
DS–2	Sine	DS–8	Triangular
DS–3	Normal	DS–9	Custom-1
DS–4	Laplace	DS–10	Custom-2
DS–5	Semi-Circular	DS–11	Custom-3
DS–6	Exponential	DS–12	Custom-4

Table 2
DATAFILES USED FOR TESTING

relating to shape, modality, and symmetry will be lost (unless the underlying population distribution is Uniform); (2) Using n or more bins ($m \geq n$) will spread the data points over all the bins more or less uniformly, such that any information relating to shape, modality, and symmetry will again be lost. These two extreme cases suggest that an "optimal" number of bins should be used to construct a histogram that can effectively capture information relating to shape, modality, and symmetry and provide meaningful values for summary quantities. Using very few bins (small value for m) results in a large bin width, and hence a histogram that captures the shape of the underlying distribution "coarsely" (under–fitting). Using excessive bins (large value for m) results in a small bin width, and hence a "noisy" histogram that captures the shape of the underlying distribution "finely" and typically "noisily" (over–fitting). *Fig.1 illustrates that arbitrarily increasing the number of bins to construct a histogram does not necessarily result in "better" histograms.*

Thus, the problem of selecting an "optimal" number of bins refers to selecting an appropriate number of bins for constructing a histogram that achieves a "good" balance between "degree of detail" and "noisiness" for a given dataset. In other words, the number of bins should be large enough to capture all the major shape features present in the distribution, but small enough so as to suppress finer details produced due to random sampling noise [7].

Tables 1 and 2 and Fig.2 describe the datafiles and datasets used for testing our proposed method.

2. Existing Methods

Perhaps the earliest reported method for constructing histograms is due to Sturges [18]. It is based on the assumption that a good distribution will have binomial coefficients $\binom{m-1}{i}, i = 0, 1, 2, \ldots, m-1$ as its bin counts. It suggests the number of bins to be used as:

$$m = 1 + \log_2 n \tag{5}$$

Hyndman [6] suggests that the argument used by Sturges [18] is incorrect and should not be used. Scott [14] uses IMSE (Integrated Mean Square Error – which is equal to Mean Integrated Square Error MISE [11]) as the measure of error between the estimated probability density ($\hat{f}(x)$) represented by the histogram, and the actual (and unknown) probability density ($f(x)$) of the underlying population. MISE is defined as:

$$\begin{aligned} IMSE &= \int MSE(x)\, dx \\ &= \int E(\hat{f}(x) - f(x))^2\, dx \\ &= E \int (\hat{f}(x) - f(x))^2\, dx \\ &= MISE \end{aligned} \tag{6}$$

Using this error metric with Gaussian density as the reference for the actual probability density, Scott suggests a bin width of:

$$h = \frac{3.49s}{n^{1/3}} \tag{7}$$

where s is the estimated standard deviation. Freedman et al. [2] suggests a similar formula with a slight modification as:

$$h = \frac{2(IQR(X))}{n^{1/3}} \tag{8}$$

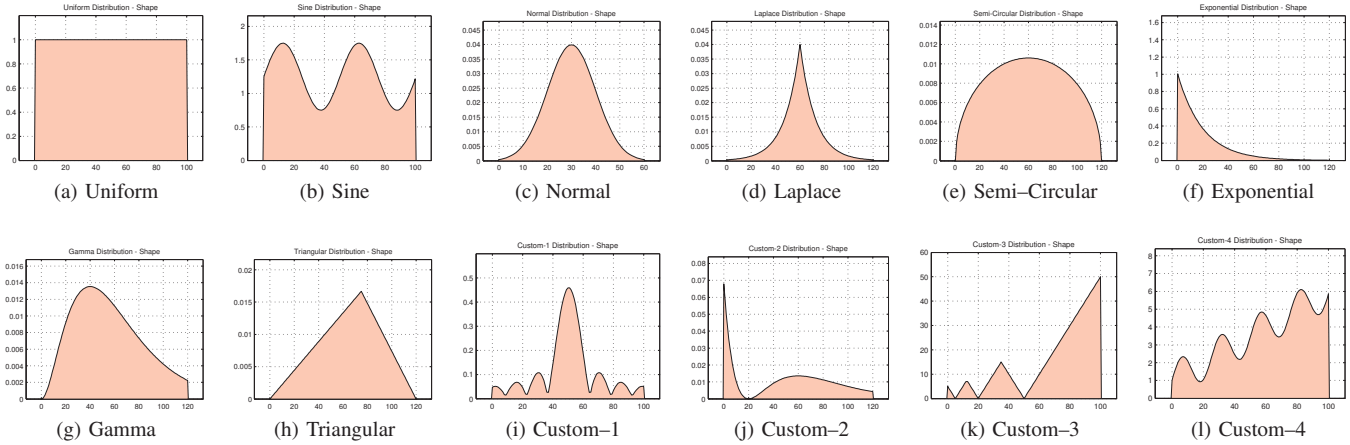
where $IQR(X)$ is the Inter–Quartile Range for the dataset X.

Methods proposed by Stone [17], Rudemo [12], and Wand [20] are also frequently encountered in the related literature. Stone [17] proposes a method based on minimization of a loss function defined on the basis of bin probabilities and number of bins. Rudemo [12] proposes a method based on Kullback–Leibler risk function and cross–validation techniques. Wand [20] extends Scott's method [14] to have good large sample consistency properties. Hall [5] investigates the use of Akaike's Information Criterion (AIC) and Kullback Liebler Cross Validation methods for constructing histograms.

More recently, Birge et al. [8] have proposed a method using a risk function based on penalized maximum likelihood. Knuth [7] has proposed a method based on maximizing the posterior probability for number of bins. Shimazaki et al. [15] have proposed a method based on minimizing an estimated cost function obtained by using a modified MISE. The method evaluates the estimated cost function using the implications of an assumption that the data are sampled independently of each other (assumption of a Poisson point process).

3. A New Proposed Method

Popular methods such as those given by Scott [14], and Freedman et al. [2] try to asymptotically minimize MISE. These methods make certain assumptions to allow estimating the value of MISE, since the actual density function of the underlying population itself is unknown. Knuth [7] suggests that it is not reasonable to extend these assumptions for all datasets. It is also known that MISE does not necessarily conform with the human perception of closeness of a density function to its target [21]. Marron et al. [9] provide a good introduction to the disconnect between classical mathematical theory and the practice of non–parametric density estimation due to the non–conformance of human perception of closeness with metrics such as MISE and MIAE. Methods employing risk functions based on penalized likelihood functions need not make assumptions about the underlying function, but their



(a) Uniform (b) Sine (c) Normal (d) Laplace (e) Semi–Circular (f) Exponential

(g) Gamma (h) Triangular (i) Custom–1 (j) Custom–2 (k) Custom–3 (l) Custom–4

Fig. 2
DATASETS USED FOR TESTING

performance will depend upon the form of the risk function selected.

In the new method proposed here, error metrics are defined on quantities observable or computable from the dataset. An intuitive balance between the error and the cost of computing the histogram is used to select the number of bins.

Motivation: A histogram for a given dataset can be interpreted as a compact representation of the dataset itself, obtained by a lossy compression process. A good histogram will provide enough information to recreate data whose Cumulative Distribution Function (CDF) approximately matches the Cumulative Distribution Function of the actual dataset itself (Statement–I). Also, a good histogram will have no significant shape information inside any bin (Statement–II).

Reflection will show that Statements I & II are axiomatic. They also indicate that data can be reconstructed from a given histogram. There are two simple ways to approximately reconstruct data from a histogram. For each bin B_k with bin count c_k: (1) recreate c_k data points equal to the bin center $((t_k+t_{k+1})/2)$ – equivalent to nearest neighbor interpolation; (2) recreate c_k data points spread uniformly over (t_k, t_{k+1}) – equivalent to linear interpolation.

Let $\hat{X}_{NN} = \{\hat{x}_{1_{NN}}, \hat{x}_{2_{NN}}, \ldots, \hat{x}_{n_{NN}}\}$ represent data reconstructed using the nearest neighbor equivalent described above, and let $\hat{X}_L = \{\hat{x}_{1_L}, \hat{x}_{2_L}, \ldots, \hat{x}_{n_L}\}$ represent data reconstructed using the linear interpolation equivalent. Fig.3 illustrates that for a histogram constructed using a given number of bins for a dataset, the CDF of the data recreated using linear interpolation matches the actual CDF more closely than the data recreated using the nearest neighbor approximation. Due to the Glivenko-Cantelli theorem [3], [1] both approximations will converge to the actual CDF itself as m increases.

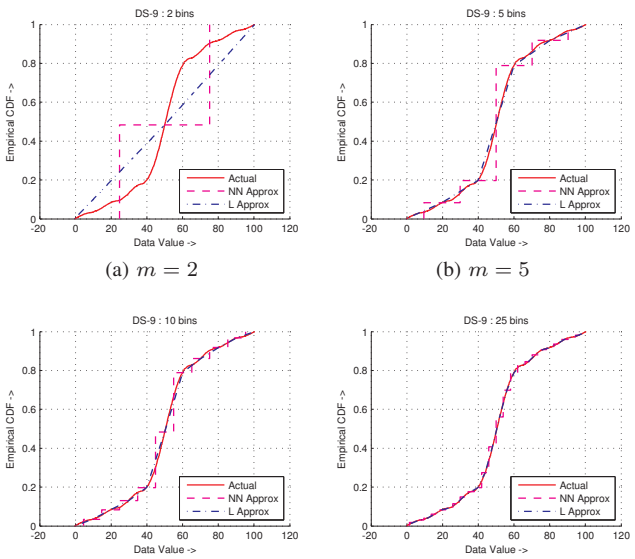
Define the error metrics E_{NN} and E_L for the nearest neighbor and linear interpolation reconstructions , respectively, by:

$$\begin{aligned} E_{NN} &= \sum_{i=1}^{n} |x_i - \hat{x}_{i_{NN}}| \\ E_L &= \sum_{i=1}^{n} |x_i - \hat{x}_{i_L}| \end{aligned} \tag{9}$$

Due to the aforementioned theorem, E_{NN} and E_L will converge to zero as the number of bins used to construct the histogram are increased ($m \to \infty$). In fact the convergence of the error metrics to zero is very likely once $m \geq n$. The CDF of data reconstructed using linear interpolation, which matches the actual data CDF more closely than the data reconstructed using the nearest neighbor approximation, indicates that E_L will converge faster than E_{NN}. Fig.4 shows plots of E_{NN} and E_L for various values of m. In these plots, the vertical axis represents the value of the error metrics, and the horizontal axis represents the value of the computational cost. The computational cost involved in constructing a histogram using m bins for n points will at the most be of order $O(mn)$. Since we are trying to select m for the same n points, the computational costs will be proportional to m and hence m is used as the computational cost.

Fig.4 uses square markers to indicate "elbow points" for both error metric curves. An elbow point marks the region where incurrence of further "costs" does not result in further significant "gains". Hence elbow points represent an intuitive trade–off between two conflicting quantities. The method is often traced to Thorndike [19] and has been used for similar purposes [22], [13]. The method described in [22] is used to compute the elbow points for the work done in this paper.

Let m_{NN} and m_L correspond, respectively, to the number of bins indicated by the elbow points on the E_{NN} and E_L metric curves. Using any m in $[m_L,m_{NN}]$ will result in a histogram that offers a reasonably good trade–off between the error metrics and the cost involved. In all the histograms constructed using an m in $[m_L,m_{NN}]$, the histogram having

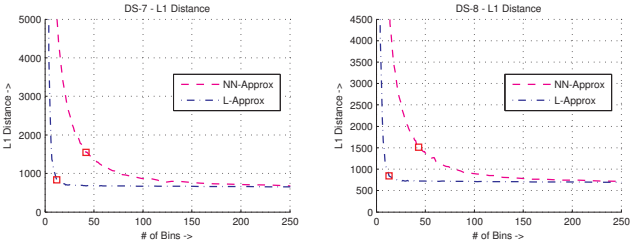


(a) $m = 2$ (b) $m = 5$

(c) $m = 10$ (d) $m = 25$

Fig. 3

EMPIRICAL CDF: DATA APPROXIMATIONS USING $m = 2, 5, 10, 25$ BINS FOR DS–9 (DF–3)



(a) DS–7 (b) DS–8

Fig. 4

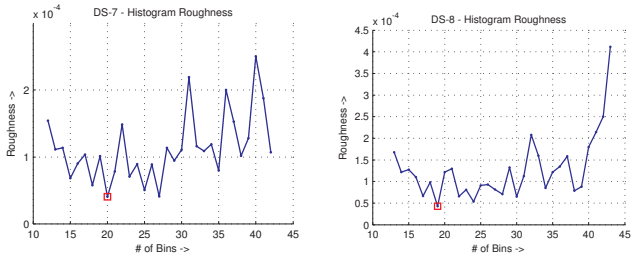
ERROR METRICS FOR DS–7 & DS–8 (DF–3)

the lowest roughness $\hat{R}$ is likely to be the most visually appealing. The roughness measure for a histogram is defined as [4]:

$$\hat{R} = \sum(\Delta^2 \hat{f}(x))h \tag{10}$$

where Δ^2 represents the second order finite difference for $\hat{f}(x)$. Fig.5 shows Roughness measures for histograms constructed with m in the corresponding $[m_L, m_{NN}]$ for DS–7 & DS–8.

In summary, to construct a histogram using our new method: (1) Define $M_1 = \{1, 2, \ldots, \sqrt{n}, \frac{n}{\sqrt{n}}, \ldots, \frac{n}{2}, \frac{n}{1}\}$; (2) Construct a histogram for X with m bins for all m in M_1; (3) Construct E_{NN} and E_L for each histogram; (4) Compute m_{NN} and m_L for the E_{NN} and E_L metric curves; (5) Define $M_2 = \{m_L, m_L + 1, \ldots, m_{NN} - 1, m_{NN}\}$; (6) For each m in M_2 construct a histogram for X with m bins; (7) Compute roughness metric $\hat{R}$ for each histogram; (8) Select as the optimal number of bins m_{opt}, the value of m that has the lowest $\hat{R}$.



(a) DS–7 (b) DS–8

Fig. 5

ROUGHNESS MEASURES FOR DS–7 & DS–8 (DF–3)

4. Experiments & Results

The method explained in Section 3 was coded in MATLAB for testing on the datafiles/datasets introduced in Section 1. Shimazaki et al. [15] and Knuth [7] provide MATLAB implementations of their methods. Methods due to Sturges [18], Scott [14], and Freedman et al. [2] were also coded in MATLAB. All the methods were tested on datafiles DF–1, DF–2, DF–3, and DF–4.

The following abbreviations are used in the tables and figures displaying results: StM – Sturges Method; ScM – Scott Method; FDM – Freedman Diaconis Method; SM – Shimazaki et al. Method; KM – Knuth Method; LHM – method proposed in this paper.

In order to measure the performance of the various methods mentioned above, the values of E_{NN}, E_L, and $\hat{R}$ are computed for the histograms generated by each method. It is desirable to have values as low as possible for all three metrics simultaneously. However, low values of $\hat{R}$ tend to result in relatively higher values of E_{NN} and E_L, and vice versa. E_{NN} and E_L indicate a given histogram's fidelity in representing the data, and $\hat{R}$ indicates the degree of over–fitting (or under–fitting) in the representation.

Tables 3 and 4 document values of m_{opt}, E_{NN}, E_L, and $\hat{R}$ for histograms generated by various methods for each dataset. The maximum values for m_{opt}, and the minimum values for E_{NN}, E_L, and $\hat{R}$ across all the methods are highlighted in blue boldface for easy reading. It can be seen from the tables that the method proposed herein (LHM) produces the lowest values of E_{NN}, E_L, and $\hat{R}$ simultaneously for a vast majority of the cases. This indicates that the proposed method does a better job of capturing shape-related information to a good degree of detail without admitting excessive noise as compared to the other methods.

Fig.6 to 11 display histograms constructed using various methods for the datasets in datafile DF–3. Visual examination of these plots and comparison to data distribution shapes in Fig.2 supports the aforementioned inference. Results for other datasets (and other datafiles) were found to be similar.

The method proposed in this paper also produces some results that the authors find less satisfying, in which case the shapes of the distributions underlying the population are not as well captured. However, as shown in Fig.12 and 13, results from the other methods are also less satisfying.

DS		StM	ScM	FDM	SM	KM	LHM
DS-1	m_{opt}	10	**13**	**13**	1	1	1
	E_{NN} (x 10^2)	12.75	**9.86**	**9.86**	127.46	127.46	127.46
	E_L (x 10^2)	1.84	**1.75**	**1.75**	1.90	1.90	1.90
	R (x 10^{-5})	2.19	16.95	16.95	**0.00**	**0.00**	**0.00**
DS-2	m_{opt}	11	12	12	4	4	**22**
	E_{NN} (x 10^2)	12.68	11.62	11.62	34.81	34.81	**6.43**
	E_L (x 10^2)	2.12	2.18	2.18	2.59	2.59	**1.87**
	R (x 10^{-4})	6.13	4.92	4.92	37.06	37.06	**2.20**
DS-3	m_{opt}	11	5	3	8	8	**20**
	E_{NN} (x 10^2)	7.39	15.91	26.28	10.04	10.04	**4.04**
	E_L (x 10^2)	1.97	4.81	9.00	2.30	2.30	**1.73**
	R (x 10^{-4})	18.00	142.05	499.85	43.61	43.61	**2.25**
DS-4	m_{opt}	11	8	5	15	11	**28**
	E_{NN} (x 10^2)	15.18	21.87	32.05	11.44	15.18	**6.17**
	E_L (x 10^2)	3.46	7.71	9.43	2.50	3.46	**2.03**
	R (x 10^{-3})	9.22	9.80	33.42	5.26	9.22	**1.05**
DS-5	m_{opt}	11	13	12	6	3	**15**
	E_{NN} (x 10^3)	1.53	1.27	1.40	2.77	5.43	**1.12**
	E_L (x 10^2)	2.36	2.07	**2.05**	3.14	9.08	2.10
	R (x 10^{-5})	14.59	12.91	19.88	19.04	166.64	**8.16**
DS-6	m_{opt}	11	11	7	14	7	**18**
	E_{NN} (x 10^2)	15.79	15.79	25.09	12.51	25.09	**9.87**
	E_L (x 10^2)	3.26	3.26	6.20	2.46	6.20	**2.33**
	R (x 10^{-4})	7.08	7.08	23.04	3.57	23.04	**2.03**
DS-7	m_{opt}	11	**12**	10	7	7	**12**
	E_{NN} (x 10^3)	1.56	**1.39**	1.67	2.41	2.41	**1.39**
	E_L (x 10^2)	2.77	**2.51**	2.62	3.65	3.65	**2.51**
	R (x 10^{-4})	1.67	**1.19**	2.69	8.47	8.47	**1.19**
DS-8	m_{opt}	11	11	9	9	6	**19**
	E_{NN} (x 10^2)	15.41	15.41	18.65	18.65	27.79	**8.92**
	E_L (x 10^2)	2.49	2.49	2.95	2.95	5.12	**2.13**
	R (x 10^{-4})	2.38	2.38	4.70	4.70	13.78	**1.17**
DS-9	m_{opt}	11	9	4	19	5	**27**
	E_{NN} (x 10^2)	12.61	15.43	40.00	7.30	25.34	**5.43**
	E_L (x 10^2)	4.47	5.05	20.38	2.35	5.54	**2.15**
	R (x 10^{-3})	12.56	27.60	6.77	5.83	51.06	**2.32**
DS-10	m_{opt}	11	15	17	**24**	16	23
	E_{NN} (x 10^2)	15.84	11.48	10.18	**7.29**	11.03	7.54
	E_L (x 10^2)	4.37	3.24	2.67	2.25	2.82	**2.17**
	R (x 10^{-4})	52.63	30.38	22.56	11.38	24.56	**8.65**
DS-11	m_{opt}	11	11	8	8	7	**20**
	E_{NN} (x 10^2)	12.57	12.57	17.50	17.50	20.02	**6.93**
	E_L (x 10^2)	2.75	2.75	3.76	3.76	4.20	**2.12**
	R (x 10^{-4})	15.65	15.65	19.44	19.44	36.89	**4.33**
DS-12	m_{opt}	11	**12**	11	4	4	4
	E_{NN} (x 10^3)	1.26	**1.16**	1.26	3.46	3.46	3.46
	E_L (x 10^2)	**2.25**	2.32	**2.25**	3.06	3.06	3.06
	R (x 10^{-6})	2036.87	1678.39	2036.87	**9.75**	**9.75**	**9.75**

(a) Results for DF–1 ($\approx$ 500 points)

DS		StM	ScM	FDM	SM	KM	LHM
DS-1	m_{opt}	**11**	10	10	1	1	1
	E_{NN} (x 10^3)	**2.34**	2.57	2.57	25.50	25.50	25.50
	E_L (x 10^2)	3.41	3.48	3.48	**3.41**	**3.41**	**3.41**
	R (x 10^{-5})	2.58	5.25	5.25	**0.00**	**0.00**	**0.00**
DS-2	m_{opt}	12	10	10	4	4	**28**
	E_{NN} (x 10^2)	22.23	26.39	26.39	66.19	66.19	**9.76**
	E_L (x 10^2)	3.97	4.60	4.60	5.72	5.72	**3.54**
	R (x 10^{-4})	6.59	10.03	10.03	41.49	41.49	**1.13**
DS-3	m_{opt}	12	4	3	14	10	**18**
	E_{NN} (x 10^2)	13.41	38.86	51.60	11.44	15.80	**9.13**
	E_L (x 10^2)	4.06	16.06	19.13	3.70	4.27	**3.62**
	R (x 10^{-4})	14.10	169.16	514.27	6.76	19.19	**2.94**
DS-4	m_{opt}	12	6	4	21	11	**37**
	E_{NN} (x 10^2)	27.22	56.19	88.13	15.52	28.77	**9.06**
	E_L (x 10^2)	7.22	25.40	55.74	3.97	6.59	**3.75**
	R (x 10^{-4})	55.57	156.80	103.28	25.63	99.27	**5.61**
DS-5	m_{opt}	12	10	10	6	6	**18**
	E_{NN} (x 10^3)	2.65	3.11	3.11	5.16	5.16	**1.75**
	E_L (x 10^2)	4.03	4.65	4.65	6.33	6.33	**3.68**
	R (x 10^{-5})	5.02	5.56	5.56	22.77	22.77	**4.07**
DS-6	m_{opt}	12	8	5	18	9	**28**
	E_{NN} (x 10^3)	2.69	4.09	6.66	1.81	3.61	**1.17**
	E_L (x 10^2)	5.36	10.56	24.50	**3.75**	7.59	3.83
	R (x 10^{-4})	6.01	16.52	50.55	3.25	12.33	**1.00**
DS-7	m_{opt}	12	9	8	13	9	**19**
	E_{NN} (x 10^3)	2.66	3.54	3.96	2.47	3.54	**1.69**
	E_L (x 10^2)	4.78	5.60	6.53	4.31	5.60	**3.82**
	R (x 10^{-5})	12.98	36.91	59.32	12.62	36.91	**4.39**
DS-8	m_{opt}	12	9	7	10	10	**19**
	E_{NN} (x 10^3)	2.63	3.49	4.46	3.15	3.15	**1.67**
	E_L (x 10^2)	4.21	5.68	8.19	4.96	4.96	**3.62**
	R (x 10^{-5})	20.62	42.13	81.31	28.39	28.39	**4.80**
DS-9	m_{opt}	12	7	3	30	18	**36**
	E_{NN} (x 10^2)	22.30	41.00	69.76	9.32	14.99	**7.76**
	E_L (x 10^2)	6.92	8.41	29.66	3.84	4.67	**3.77**
	R (x 10^{-3})	17.98	42.55	30.02	2.04	7.68	**1.27**
DS-10	m_{opt}	12	12	14	**47**	17	30
	E_{NN} (x 10^2)	27.27	27.27	23.49	**7.26**	19.30	11.01
	E_L (x 10^2)	7.98	7.98	6.30	**3.59**	5.05	3.80
	R (x 10^{-4})	53.64	53.64	40.98	6.88	24.53	**5.71**
DS-11	m_{opt}	12	9	6	16	13	**25**
	E_{NN} (x 10^3)	2.23	3.00	4.46	1.66	2.08	**1.10**
	E_L (x 10^2)	4.85	6.83	13.73	4.19	4.43	**3.66**
	R (x 10^{-4})	14.30	14.98	16.36	7.15	12.02	**3.77**
DS-12	m_{opt}	12	9	9	8	4	**24**
	E_{NN} (x 10^3)	2.22	2.95	2.95	3.32	6.61	**1.15**
	E_L (x 10^2)	4.52	5.59	5.59	4.31	6.68	**3.77**
	R (x 10^{-6})	1949.62	1609.92	1609.92	4123.74	**4.91**	413.81

(b) Results for DF–2 ($\approx$ 1000 points)

Table 3

RESULTS FOR DF–1 & DF–2 USING VARIOUS METHODS

DS		StM	ScM	FDM	SM	KM	LHM
DS-1	m_{opt}	**12**	8	8	1	1	1
	E_{NN} (x 10^3)	**4.29**	6.40	6.40	51.00	51.00	51.00
	E_L (x 10^2)	6.65	6.74	6.74	**6.60**	**6.60**	**6.60**
	R (x 10^{-6})	12.62	5.58	5.58	**0.00**	**0.00**	**0.00**
DS-2	m_{opt}	12	8	8	14	4	**33**
	E_{NN} (x 10^3)	4.33	6.48	6.48	3.72	12.94	**1.68**
	E_L (x 10^2)	7.69	10.57	10.57	7.41	10.68	**6.95**
	R (x 10^{-5})	72.26	209.96	209.96	42.42	514.19	**6.35**
DS-3	m_{opt}	12	3	2	17	12	**31**
	E_{NN} (x 10^3)	2.65	10.19	16.79	1.91	2.65	**1.20**
	E_L (x 10^2)	7.64	37.73	144.58	7.14	7.64	**6.99**
	R (x 10^{-4})	12.04	552.52	**0.00**	4.65	12.04	1.77
DS-4	m_{opt}	13	5	3	25	17	**48**
	E_{NN} (x 10^3)	4.74	11.53	17.14	2.54	3.65	**1.40**
	E_L (x 10^2)	10.98	39.56	78.06	7.40	8.57	**7.07**
	R (x 10^{-4})	73.21	410.54	470.35	18.01	39.07	**2.82**
DS-5	m_{opt}	13	8	8	11	7	**17**
	E_{NN} (x 10^3)	4.75	7.64	7.64	5.60	8.66	**3.60**
	E_L (x 10^2)	8.02	10.86	10.86	8.31	11.48	**7.24**
	R (x 10^{-5})	6.38	12.81	12.81	7.21	16.79	**2.34**
DS-6	m_{opt}	13	6	4	23	17	**30**
	E_{NN} (x 10^3)	4.78	10.48	16.10	2.77	3.68	**2.05**
	E_L (x 10^2)	9.44	34.01	72.55	7.24	7.61	**7.01**
	R (x 10^{-5})	46.38	354.75	737.01	12.10	27.92	**3.48**
DS-7	m_{opt}	13	7	6	14	12	**20**
	E_{NN} (x 10^3)	4.78	8.80	10.22	4.41	5.12	**3.11**
	E_L (x 10^2)	8.16	13.66	18.73	7.56	8.34	**7.35**
	R (x 10^{-5})	11.14	99.50	155.56	11.34	15.44	**4.04**
DS-8	m_{opt}	13	7	6	13	13	**19**
	E_{NN} (x 10^3)	4.75	8.69	10.18	4.75	4.75	**3.28**
	E_L (x 10^2)	8.41	16.53	21.76	8.41	8.41	**7.55**
	R (x 10^{-5})	16.78	90.27	141.93	16.78	16.78	**4.30**
DS-9	m_{opt}	12	5	2	38	18	**47**
	E_{NN} (x 10^3)	4.37	9.32	33.70	1.50	2.93	**1.23**
	E_L (x 10^2)	13.95	20.20	236.71	7.05	8.96	**6.96**
	R (x 10^{-4})	190.43	626.38	**0.00**	13.42	82.32	6.93
DS-10	m_{opt}	13	10	11	**45**	23	37
	E_{NN} (x 10^3)	4.88	6.41	5.79	**1.52**	2.78	1.77
	E_L (x 10^2)	14.06	20.45	17.42	**7.00**	7.90	7.06
	R (x 10^{-4})	49.83	73.80	64.99	4.19	14.59	**3.25**
DS-11	m_{opt}	13	7	5	13	13	**34**
	E_{NN} (x 10^3)	4.04	7.55	10.44	4.04	4.04	**1.61**
	E_L (x 10^2)	8.51	14.55	27.88	8.51	8.51	**7.17**
	R (x 10^{-4})	12.27	37.72	35.01	12.27	12.27	**2.17**
DS-12	m_{opt}	13	7	7	8	4	**31**
	E_{NN} (x 10^3)	4.01	7.40	7.40	6.49	12.88	**1.76**
	E_L (x 10^2)	8.35	14.10	14.10	8.38	12.50	**6.93**
	R (x 10^{-6})	1710.36	775.42	775.42	3847.28	**2.82**	208.35

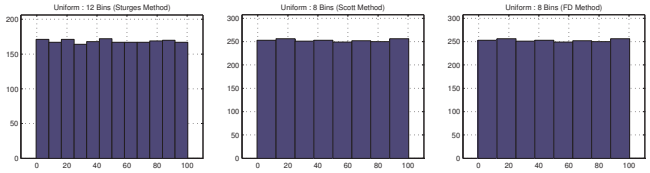
(a) Results for DF–3 ($\approx$ 2000 points)

DS		StM	ScM	FDM	SM	KM	LHM
DS-1	m_{opt}	**14**	6	6	1	1	1
	E_{NN} (x 10^3)	**9.24**	21.29	21.29	127.53	127.53	127.53
	E_L (x 10^3)	1.72	1.72	1.72	**1.72**	**1.72**	**1.72**
	R (x 10^{-6})	9.75	5.31	5.31	**0.00**	**0.00**	**0.00**
DS-2	m_{opt}	14	6	6	18	4	**30**
	E_{NN} (x 10^3)	9.24	21.28	21.28	7.22	31.89	**4.50**
	E_L (x 10^3)	1.84	3.76	3.76	1.73	2.70	**1.69**
	R (x 10^{-5})	39.66	256.35	256.35	18.99	503.03	**5.20**
DS-3	m_{opt}	14	2	2	22	19	**61**
	E_{NN} (x 10^3)	5.68	41.89	41.89	3.74	4.29	**1.35**
	E_L (x 10^3)	1.82	36.47	36.47	1.69	1.70	**1.66**
	R (x 10^{-5})	67.85	**0.00**	**0.00**	15.97	22.60	5.60
DS-4	m_{opt}	14	3	2	54	21	**63**
	E_{NN} (x 10^3)	11.13	42.14	99.51	3.20	7.40	**2.88**
	E_L (x 10^3)	2.84	19.42	90.85	**1.70**	1.90	1.70
	R (x 10^{-4})	49.05	490.78	**0.00**	2.59	25.14	1.84
DS-5	m_{opt}	14	6	6	15	12	**25**
	E_{NN} (x 10^3)	10.82	24.93	24.93	10.12	12.59	**6.17**
	E_L (x 10^3)	1.86	3.31	3.31	1.80	1.91	**1.70**
	R (x 10^{-5})	4.52	23.42	23.42	3.94	4.85	**1.66**
DS-6	m_{opt}	14	4	3	28	17	**40**
	E_{NN} (x 10^3)	11.01	39.73	54.54	5.64	9.03	**3.88**
	E_L (x 10^3)	2.21	18.23	31.41	1.74	1.91	**1.72**
	R (x 10^{-5})	35.02	750.14	1023.42	9.27	20.94	**2.58**
DS-7	m_{opt}	14	5	5	21	16	**28**
	E_{NN} (x 10^3)	10.89	29.98	29.98	7.34	9.54	**5.57**
	E_L (x 10^3)	1.94	6.85	6.85	1.72	1.84	**1.72**
	R (x 10^{-5})	8.72	240.39	240.39	2.63	5.40	**2.43**
DS-8	m_{opt}	14	5	4	15	15	**33**
	E_{NN} (x 10^3)	10.79	29.97	36.79	10.12	10.12	**4.78**
	E_L (x 10^3)	1.95	7.89	10.33	1.90	1.90	**1.72**
	R (x 10^{-5})	11.66	235.58	509.14	10.42	10.42	**2.70**
DS-9	m_{opt}	14	4	2	51	30	**67**
	E_{NN} (x 10^3)	9.12	37.21	83.41	3.05	4.53	**2.48**
	E_L (x 10^3)	3.26	20.41	59.32	1.70	1.80	**1.69**
	R (x 10^{-4})	124.90	79.24	**0.00**	7.60	21.94	2.89
DS-10	m_{opt}	14	7	8	50	30	**59**
	E_{NN} (x 10^3)	11.14	22.96	19.91	3.42	5.32	**2.94**
	E_L (x 10^3)	3.13	9.23	7.15	1.73	1.83	**1.73**
	R (x 10^{-4})	44.75	90.91	92.32	2.16	6.00	**1.74**
DS-11	m_{opt}	14	5	3	26	17	**44**
	E_{NN} (x 10^3)	9.27	25.71	45.70	5.09	7.66	**3.27**
	E_L (x 10^3)	2.27	7.01	14.85	1.76	1.95	**1.71**
	R (x 10^{-5})	84.47	372.90	636.10	31.79	70.25	**8.79**
DS-12	m_{opt}	14	6	5	20	8	**43**
	E_{NN} (x 10^3)	9.20	21.31	25.56	6.54	15.97	**3.35**
	E_L (x 10^3)	2.01	3.99	4.98	1.75	2.08	**1.69**
	R (x 10^{-5})	145.12	30.04	**3.40**	69.38	396.55	8.77

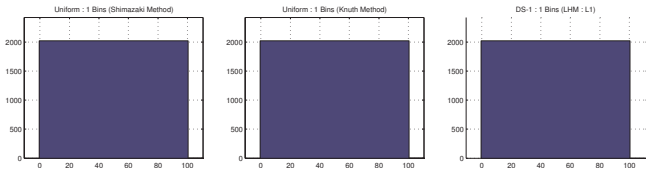
(b) Results for DF–4 ($\approx$ 5000 points)

Table 4

RESULTS FOR DF–3 & DF–4 USING VARIOUS METHODS



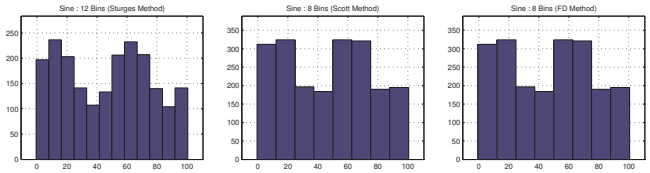
(a) StM (b) ScM (c) FD



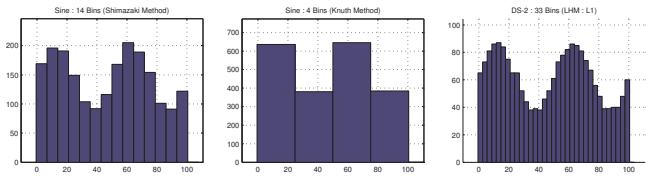
(d) SM (e) KM (f) LHM

Fig. 6

HISTOGRAMS GENERATED FOR DS–1 (FROM DF–3) USING VARIOUS METHODS.



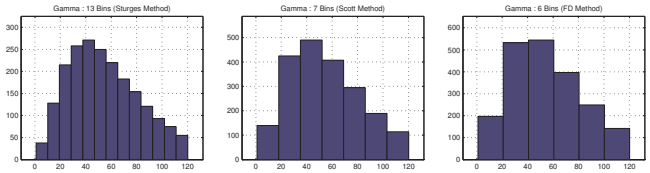
(a) StM (b) ScM (c) FD



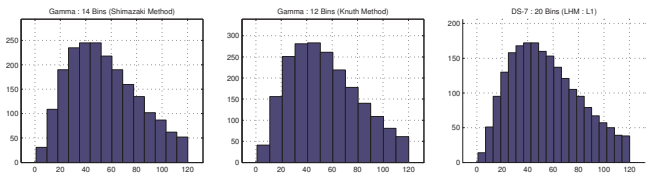
(d) SM (e) KM (f) LHM

Fig. 7

HISTOGRAMS GENERATED FOR DS–2 (FROM DF–3) USING VARIOUS METHODS.



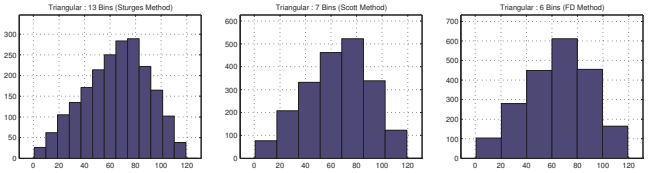
(a) StM (b) ScM (c) FD



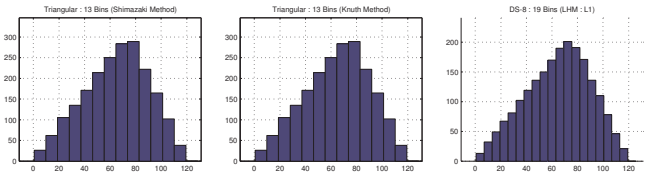
(d) SM (e) KM (f) LHM

Fig. 8

HISTOGRAMS GENERATED FOR DS–7 (FROM DF–3) USING VARIOUS METHODS.



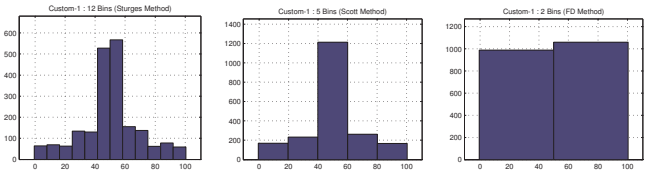
(a) StM (b) ScM (c) FD



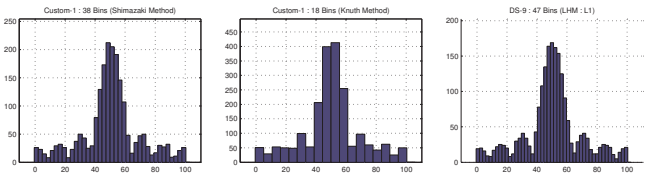
(d) SM (e) KM (f) LHM

Fig. 9

HISTOGRAMS GENERATED FOR DS–8 (FROM DF–3) USING VARIOUS METHODS.



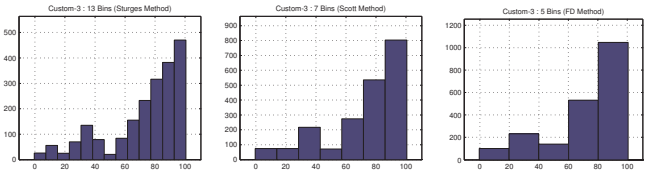
(a) StM (b) ScM (c) FD



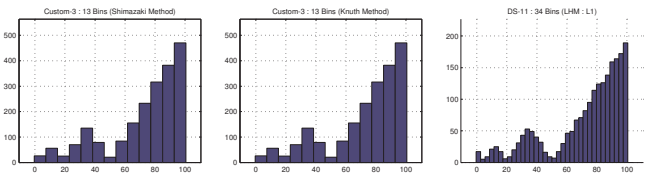
(d) SM (e) KM (f) LHM

Fig. 10

HISTOGRAMS GENERATED FOR DS–9 (FROM DF–3) USING VARIOUS METHODS.



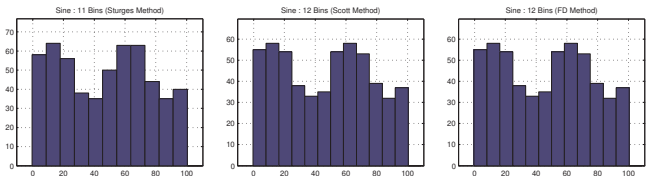
(a) StM (b) ScM (c) FD



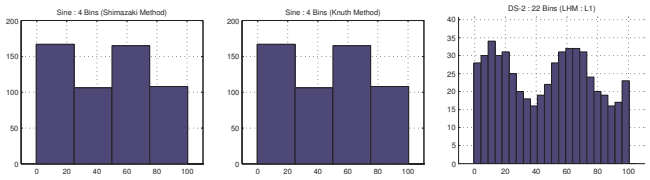
(d) SM (e) KM (f) LHM

Fig. 11

HISTOGRAMS GENERATED FOR DS–11 (FROM DF–3) USING VARIOUS METHODS.



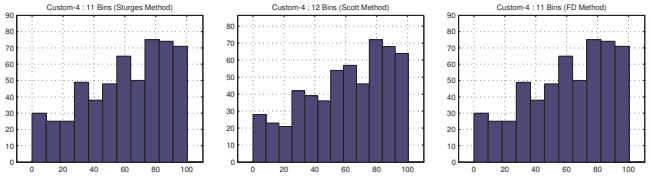
(a) StM (b) ScM (c) FD



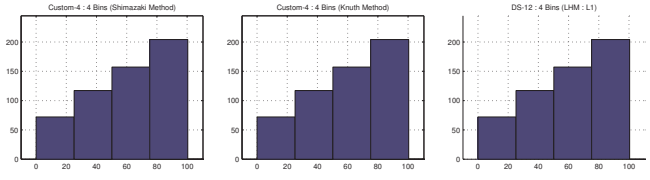
(d) SM (e) KM (f) LHM

Fig. 12

LESS SATISFYING RESULT (LHM): UNDESIRABLE SPIKE ON LEFT MODE (DS–2, DF–1).



(a) StM (b) ScM (c) FD



(d) SM (e) KM (f) LHM

Fig. 13

LESS SATISFYING RESULT (LHM): SHAPE NOT CAPTURED "WELL" (DS–12, DF–1).

5. Conclusions

This paper introduces a new method for selecting the number of bins for constructing a histogram for a given dataset. The performance of the proposed method is compared with the performance of five other methods in the literature. Comparison results show that the proposed method performs better than the other five methods, with the proposed method producing visually appealing histograms that reveal shape features of underlying distribution to a finer detail without admitting excessive noise.

We suggest that future investigations should explore the following issues: (1) Designing a metric to measure the performance of a histogram as evaluated by human perception; (2) Extension of ideas proposed herein to higher dimensional data; and (3) Optimizing the proposed method to reduce time and memory requirements.

References

[1] F. P. Cantelli. Sulla determinazione empirica delle leggi di probabilita. *Giornale dell'Istituto Italiano degli Attuari*, (4):221–424, 1933.

[2] D. Freedman and P. Diaconis. On the histogram as a density estimator:L2 theory. *Probability Theory and Related Fields*, 57(4):453–476, December 1981.

[3] V. I. Glivenko. Sulla determinazione empirica delle leggi di probabilita. *Giornale dell'Istituto Italiano degli Attuari*, (4):92–99, 1933.

[4] P. J. Green and B.W.Silverman. *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. Chapman and Hall/CRC, 1994.

[5] P. Hall. Akaike's information criterion and kullback-leibler loss for histogram density estimation. *Probability Theory and Related Fields*, 85:449–467, 1990.

[6] R. J. Hyndman. The problem with sturges rule for constructing histograms. *Business*, pages 1–2, July 1995.

[7] K. H. Knuth. Optimal Data-Based Binning for Histograms. *ArXiv Physics e-prints*, May 2006.

[8] Lucien Birgé and Yves Rozenholc. How many bins should be put in a regular histogram. *ESAIM: P&S*, 10:24–45, 2006.

[9] J. S. Marron and A. B. Tsybakov. Visual error criteria for qualitative smoothing. *Journal of the American Statistical Association*, 90(430):499–507, 1995.

[10] W. L. Martinez and A. R. Martinez. *Computational Statistics Handbook with MATLAB, Second Edition (Computer Science and Data Analysis)*. Chapman & Hall/CRC, 2 edition, December 2007.

[11] C. R. Rao, E. J. Wegman, and J. L. Solka. *Handbook of Statistics, Volume 24: Data Mining and Data Visualization (Handbook of Statistics)*. North-Holland Publishing Co., 2005.

[12] M. Rudemo. Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, 9(2):65–78, 1982.

[13] S. Salvador and P. Chan. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, pages 576 – 584, nov. 2004.

[14] D. W. Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979.

[15] H. Shimazaki and S. Shinomoto. A method for selecting the bin size of a time histogram. *Neural Comput.*, 19(6):1503–1527, 2007.

[16] J. S. Simonoff and F. Udina. Measuring the stability of histogram appearance when the anchor position is changed. *Comput. Stat. Data Anal.*, 23(3):335–353, 1997.

[17] C. J. Stone. An asymptotically optimal histogram selection rule. In *Proceedings of the Berkeley conference in honor of Jerzy Neyman and Jack Kiefer, Vol. II (Berkeley, Calif., 1983)*, Wadsworth Statist./Probab. Ser., pages 513–520. Wadsworth, 1985.

[18] H. A. Sturges. The choice of a class interval. *Journal of the American Statistical Association*, 21(153):65–66, 1926.

[19] R. L. Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953.

[20] M. P. Wand. Data-based choice of histogram bin width. *The American Statistician*, 51:59–64, 1996.

[21] M. P. Wand and M. C. Jones. *Kernel Smoothing (Chapman & Hall/CRC Monographs on Statistics & Applied Probability)*. Chapman and Hall/CRC, 1994.

[22] Q. Zhao, M. Xu, and P. Fränti. Knee point detection on bayesian information criterion. In *ICTAI '08: Proceedings of the 2008 20th IEEE International Conference on Tools with Artificial Intelligence*, pages 431–438, Washington, DC, USA, 2008. IEEE Computer Society.

SESSION

WEB AND TEXT MINING

Chair(s)

Peter Geczy
Nikolaos Kourentzes
Robert Stahlbock

A Secure Knowledge Discovery Framework for Clinical Informatics

Yueh-Hsun Shih[1], Chung-Yueh Lien[1], Chi-Hsien Chen[2], Chia-Hung Hsiao[3]*, and Woei-Chyn Chu[1]#
[1]Institute of Biomedical Engineering, National Yang Ming University, Taipei, Taiwan
[2] Office for Human Subjects Protection, Changhua Christian Hospital, Changhua, Taiwan
[3]Department of Medical Informatics, Tzu Chi University, Hualien, Taiwan
Corresponding authors: *chhsiao@mail.tcu.edu.tw ; #wchu@ym.edu.tw

Abstract - *This paper describes a secure framework for knowledge discovery from clinical data stored in the distributed healthcare systems. Two important integration profiles have been specified this purpose, namely, the "Integrating the Healthcare Enterprise (IHE) Cross-Enterprise Document Sharing (XDS)" and "IHE Quality, Research and Public Health (QRPH)." IHE XDS provides the cross-enterprise patient-centred documents sharing workflow and IHE QRPH describes how the quality measures are mapped to the clinical documents to create the Quality Reporting Document Architecture (QRDA). Providers can use the same data structures developed for information exchange to report on quality measures directly out of the Electronic Medical Records (EMRs). The paper proposed a secure Single Sign on (SSO) portal in this integrated framework for clinical oncologists to access data of cancer patients from various systems. The portal consists of the SSO layer, authentication layer, authorization layer, and audit trails layer to protect the patient's privacy. Therefore, the consumers who passed the security layers can apply other data mining methods, using the de-identification specific data for further clinical analysis and public health research. This paper addresses knowledge discovery processes in the integrated healthcare framework and security specifications for data mining from clinical records. Consequently, we are able to discover meaningful information from clinical data while protecting patient's privacy when we handle medical data for knowledge discovery.*

Keywords: Knowledge discovery, Medical informatics, IHE XDS, QRPH, SSO, security

1 Introduction

Cancer patients usually require several different types of therapies and lots of clinical examinations, which may endure for many years. The diagnosis data from a patient may be separated and isolated in different kinds of computer systems, as shown in Figure 1. The systems in each phase of a cancer therapy are usually not designed for long-term preservation of the clinical data or for information exchange with other systems. Thus, when a new patient comes, there might be no feedback mechanism to suggest the doctors the most suitable treatment plan for the patient of a specific cancer.

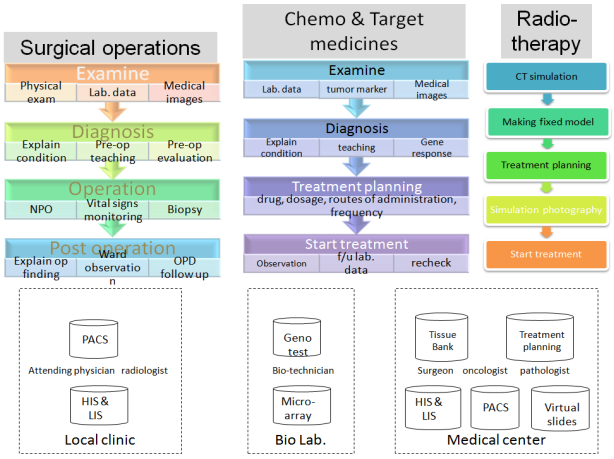


Figure 1. The exams and processes of different types of cancer therapy.

Some of cancer research organizations have been aware of the importance of the integration of various oncological data, and have been constructing specific information frameworks to realize this purpose. For example, there are two large-scale projects in U.S. and Germany, trying to implement nationwide healthcare integrated networks fed by heterogeneous information systems in Cancer healthcare domain. The cancer biomedical Informatics Grid (CaBIG) program of the National Cancer Institute (NCI) in U.S., which is to dynamically link applications, clients, and community provided resources. Its other goal is to build tools for collecting, analyzing, integrating, and providing information associated with cancer research and care [1,2]. On the other hand, the joint project of MediGRID in Germany unifies well known research institutes in the area of medicine, biomedical informatics and life sciences into a consortium. The four methodological modules (middleware, ontology, resource fusion and eScience) plan to incrementally develop and to provide a grid infrastructure while taking into account the need of the biomedical users. The user communities are represented in three research modules for biomedical informatics, image processing and clinical research [3].

Since the Taiwan Cancer Database (TCDB) program has been established, all the hospitals must register their cancer cases to TCDB for the following purpose: data collection, patient follow-up, education and research. The information about patients and their therapies is input manually, including 95 columns to record the each cancer

case. However, the current process is time-consuming and cannot provide enough information for education and research. Since the RT system in every hospital usually stores more details of treatment information for cancer patients, we could therefore combine these resources for our knowledge discovery workflow to solve these problems. In this way, all the information can be fed back in real time to the doctors, which can help them have a better understanding for planning a treatment, and evaluating the performance of cancer therapies.

This paper describes a secure framework for knowledge discovery from clinical data stored in the distributed healthcare systems. The framework is based on standard clinical document exchange architecture, the "Integrating the Healthcare Enterprise (IHE) Cross-Enterprise Document Sharing (XDS)" and "IHE Quality, Research and Public Health (QRPH)" profile. Using the system integration profile, clinical documents can be shared between systems located at different hospitals. The paper proposed a secure Single Sign on (SSO) portal in this integrated framework for clinical oncologists to access data of cancer patients from various systems. Therefore, they can apply other data mining methods for further clinical analysis and public health research. This paper addresses knowledge discovery processes in the integrated healthcare framework and security specifications for data mining from clinical records. Consequently, our framework will provide discovery meaningful information from clinical data while protecting patient privacy when we handle medical data for knowledge discovery.

2 Knowledge discovery framework for distributed EMRs

2.1 A Standardized Electronic Medical Records for Radiation Therapy

In our research, we used two standard formats as the Electronic Medical Records for Radiation Therapy (RTEMRs) archetype, including "the Health Level 7 (HL7) Clinical Document Architecture (CDA)", and 'the Digital Imaging and Communications in Medicine (DICOM) Structured Reporting (SR)". The CDA is the guild line for standardizing our clinical documents, which can be used to generate an XML format document with definition tags. In our work, we followed the standard of HL7 CDA to store cancer patients' information and to generate their RTEMRs chart, as shown in Figure 2.

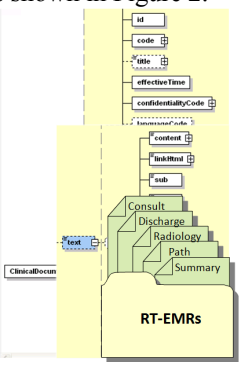


Figure 2. Cancer patient's RTEMRs chart

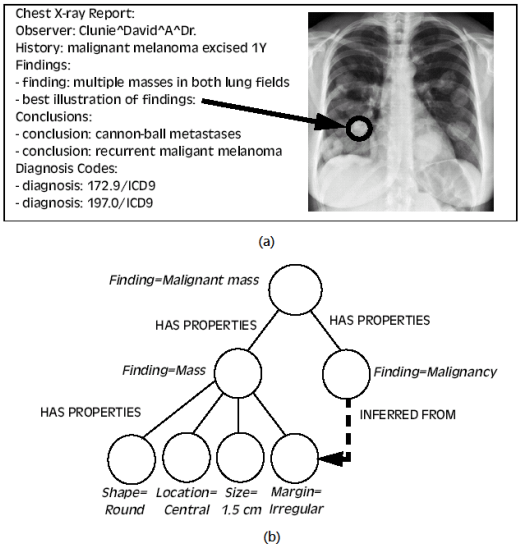


Figure 3. Simple example of a DICOM Structured report (SR).

In an RT planning, there exist lots of findings within medical images during clinical diagnosis and treatment processes. The findings may be identified manually by domain experts or automatically by image processing tools. However, the records for these findings may be stored in isolated systems with proprietary formats, which would be difficult for other systems to handle. To conquer this problem, we followed Content Mapping Resource in Part 16 of DICOM standard to design a clinical information architecture and to generate SR. XML formatted SR provides a mechanism for encoding and interchanging structured information, including annotations, measurements, and formatted text data related to medical images. Documents stored in DICOM SR and image objects constitute open standard facilitate us to develop standard IT systems in a clinical oncology department to acquire data from cancer patients. A simple example of the presentation and data structure of SR is demonstrated in Figure 3a. It combines the reference image, the overlapped contour of the finding with the structured text data in the presentation. The diagnosis results are represented with coded terminology for a precise expression. The data structure of a finding in a structured report (SR) is of tree type with attributes in each node, representing relation between root and child nodes, as shown in Figure 3b.

2.2 Knowledge Discovery Framework

Different from the other centralized knowledge management systems, the proposed architecture was built on an environment that has centralized query and distributed storage for sharing EMRs. The core technology was based on IHE XDS that provides the cross-enterprise patient-centred documents sharing workflow. The framework, as shown in Figure 4, demonstrates the workflow of the SSO Oncology Healthcare Portal. The framework is composed of the following three layers.

2.2.1 Data Acquisition and Standardization Layer

A major part of RTEMRs contains various kinds of information with different structures and data types. In this

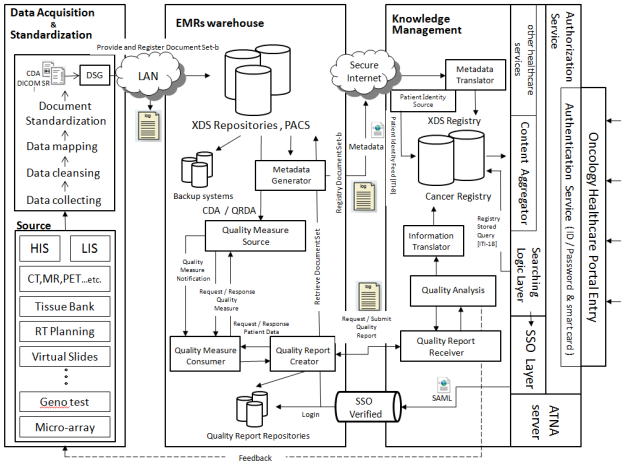


Figure 4. The Secure Knowledge Discovery Framework

layer, we acquired the data from different sources and used data mapping, standardization services to generate the standardization format documents. Next, through XDS [Provider and Registry Document set-b] workflow, these documents were uploaded to the repository.

2.2.2 EMRs warehouse Layer

In this layer, every healthcare provider establishes the repository that must provide storage and must ensure an effective link for consumers to retrieve to the saved documents. The metadata generator is responsible to extract the information from the standardized documents (e.g. CDA, DICOM SR). Through the [Registry Document Set-b] workflow, the generator sends metadata for a set of documents to the registry.

IHE QRPH is established in this layer by using the CDA and the IHE XDS environment, the quality measures are mapped to the documents to create the Quality Reporting Document Architecture (QRDA). Providers can use the same data structures developed for information exchange to report on quality measures directly out of the EMRs [4]. We can draft automated rules at this point to check the conditions of care and data collection for providing immediate quality reports on the completeness of submission data sets and can also provide feedback on adherence with related practice guidelines. In this paper, the rules we developed were focused on dealing with problems of cancer treatment.

2.2.3 Knowledge Management Layer

We implemented a SSO portal to combine the XDS/Cancer Registry, other healthcare services, and the security issue. Via the translation service, we can translate the information (e.g. metadata, quality report notification) into our relational database model, and then manage the registered information more efficiently.

According to our framework, when the patient's cancer type is diagnosed, the radiologist can use the cancer type or some conditions as keywords to query the registry, and then the distributed repositories from different healthcare providers will request the qualified documents. IHE QPRH workflow can exploit the de-identified documents to create a quality report that come from past experiments with the same cancer type. Then the report provides the knowledge to suggest the

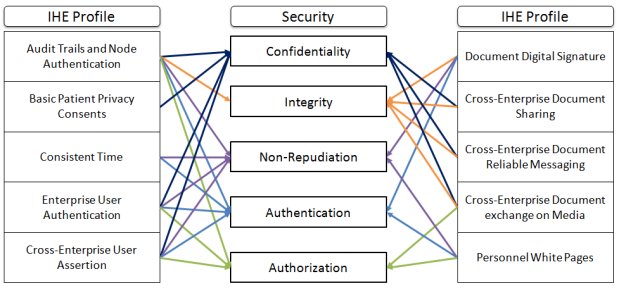


Figure5. IHE IT Infrastructure profile and the security related issues

radiologists a better treatment plan for a specific cancer patient.

3 Security overview

In our framework, we do not dedicate our research to developing a new security model; rather, we dealt with the security issue by referencing the IHE profiles, as shown in Figure 5. However, these profiles are scattered for dealing with different issues without systematic planning. Some of the profiles are repeatedly referred to the same security issue and could possibly increase the cost for development. In this paper, we will show how the existing security components can satisfy the framework for security needs.

According to our framework, we proposed a portal solution to integrate with the XDS/Cancer registry. The portal consists of the SSO layer, authentication layer, and authorization layer. The authorization layers were implemented based on Role Base Access Control (RBAC) model. Under the RBAC framework, users are granted membership into roles based on their competencies and responsibilities in the organization. We designed the authenticate layer and authorized layer by using a new access control model named simplified RBAC(s-RBAC) which was modified from the original RBAC module for access control for participants. The s-RBAC combined with IHE ITI profiles built a secure oncology healthcare infrastructure in the internet environment for healthcare use. In order to achieve the SSO, the Security Assertion Markup Language (SAML) 2.0 was used to communicate between the portal and various repositories or among systems after the user's authority is verified. And the security protection in this environment was maintained by the ATNA server. The ATNA server was used to ensure node authentications and secure communication between the nodes in this infrastructure. In addition, other mechanisms can be used to enforce the security and consistency for sharing clinical documents, such as Consistent Time (CT), Patient Demographic Query (PDQ), Patient Identifier Cross-referencing (PIX), Document Digital Signature (DSG) and patient synchronized application (PSA).

4 Conclusions

In our research, we proposed a cancer knowledge discovery framework for the clinical informatics in the RTEMRs system based on the standardization clinical document exchanging architecture. Different from other data mining systems where the resources of mining are all

stored in one database, the IHE XDS has provided an environment that has centralized query and distributed storage for distributed systems, which helps us request data from distributed systems. Because of the use of the proposed secure SSO portal, the system can be developed in highly heterogeneous environment. Consequently, we are able to discover meaningful information from clinical data while protecting patient privacy when we handle medical data for knowledge discovery.

5 Acknowledgments

This work was supported by the Ministry of Education, Aim for the Top University Plan and the National Science Council (NSC99-2218-E-241-001, NSC99-2321-B-010-013, NSC97-2320-B-010-003-MY3).

6 References

[1] G. A. Komatsoulis, et al., "caCORE version 3: Implementation of a model driven, service-oriented architecture for semantic interoperability," J. of Biomedical Informatics, vol. 41, pp. 106-123, 2008.

[2] S. Langella, et al., "Sharing data and analytical resources securely in a biomedical research Grid environment," J Am Med Inform Assoc, vol. 15, pp. 363-373, 2008.

[3] M. Vossberg, et al., "DICOM Image Communication in Globus-Based Medical Grids," Information Technology in Biomedicine, IEEE Transactions on, vol. 12, pp. 145-153, 2008.

[4] AHIC Quality Workgroup Meeting, (2007, Oct. 3), Requirements Analysis Update, PowerPoint presentation. [Online]. Available: http://www.dhhs.gov/healthit/ahic/quality/

[5] IHE IT Infrastructure (ITI) Technical Framework. (2008, Oct. 10). Supplement 2007-2008, cross-enterprise document sharing-b (XDS.b), draft for trial implementation. [Online]. Available: http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_Supplement_Cross_Enterprise_Document_Sharing_XDS-b_TI_2008-10-10.pdf

[6] IHE IT Infrastructure (ITI) Technical Framework. Quality, Research and Public Health, (2011, Mar. 9) [Online]. Available: http://www.ihe.net/Technical_Framework/index.cfm#quality

Pattern-based Aggregation of Named Entity Extractors

T. Lemmond[1], P. Kidwell[1], K. Boakye[1], N. Perry[2], J. Guensche[1], J. Nitao[1], W. Hanley[1], R. Prenger[1], and R. Glaser[1]
[1]Lawrence Livermore National Laboratory, Livermore, CA, USA
[2]Mathematics Department, Brigham Young University, Provo, UT, USA

Abstract - *Despite significant advances in named entity extraction technologies, state-of-the-art extraction tools achieve insufficient accuracy rates for practical use in many operational settings. However, they are not all prone to the same types of error, suggesting that substantial improvements may be achieved via appropriate combinations of existing tools, provided their behavior can be accurately characterized and quantified. In this paper, we present an inference framework that leverages the joint characteristics of their error processes via a pattern-based representation of extracted entity data. This approach has been shown to produce statistically significant improvements in entity extraction relative to standard performance metrics and to mitigate the weak performance of entity extractors operating under suboptimal conditions. Moreover, this aggregation methodology provides a framework for quantifying uncertainty in extracted entity output, and it can readily adapt to sparse data conditions.*

Keywords: Knowledge discovery, text mining, named entity extraction, probabilistic aggregation, ensemble learning

1 Introduction

Since the 1980s, the sophistication of machine learning and computer technologies has increased dramatically, enabling the development of solutions to a wide variety of challenges facing the Natural Language Processing (NLP) community. These problems range from the development of search engines that can interpret simple natural language queries to the construction of knowledge discovery systems predicated upon reliable information extraction from heterogeneous data sources. Often, the construction of such a knowledge base depends to a large degree upon the automatic recognition and extraction of complex relational information and, more fundamentally, related named entities (e.g., people, organizations) from a collection, or *corpus*, of text documents (e.g., e-mail, news articles, medical records, weblogs, intelligence reports). Consequently, the fidelity of knowledge discovery systems is particularly susceptible to errors introduced during the automatic extraction process.

However, even state-of-the-art entity extraction tools are vulnerable to variations in (1) the source and domain of a corpus and its adherence to conventional lexical, syntactical, and grammatical rules; (2) the availability and reliability of manually annotated data; and (3) the complexity of entity types targeted for extraction. Under these conditions extractors produce a range of interdependent errors and often fail to achieve high accuracy rates in operational settings. However, many extraction technologies, distinguished by the nature of their underlying algorithms, possess complementary characteristics that may be combined to selectively amplify their most attractive attributes (e.g., low miss or false alarm rates) and mitigate their respective weaknesses.

Many extractor combination methods that aim to leverage these characteristics have relied upon variations of a "voting" mechanism (e.g., majority vote [1]). In practice, such approaches often fall short, as they depend heavily upon the number and type of extractors chosen, and they do not account for the differing characteristics of their errors. Moreover, such systems tend to be limited in their ability to assess uncertainty, a critical capability for evaluating reliability in downstream analysis and decision-making. Proposed enhancements to the basic voting mechanism include weighting of the constituent (i.e., *base*) extractors' output [2]; stacking of entity extractors [3]-[5]; establishing a vote "threshold" [6]; and bagging of entity data [7].

Even more sophisticated combination techniques, such as that described in [8], fail to adequately account for text within a local neighborhood of a word of interest. Indeed, a method based on the Conditional Random Field (CRF) model presented by [9] demonstrated that performance may be enhanced by incorporating the classification structure of nearby words. More recently, Lemmond, et al. [10] utilized a fine-grained hierarchical error space to characterize named entity extractors' error processes and aggregate their output entity data.

The aggregation methodology described in this paper, called the *pattern-based meta-extractor (PME)*, utilizes a pattern-based representation of named entity data to evaluate the joint performance characteristics of its base entity extractors. The resulting characterization is utilized to determine the most likely truth, given base extractor output. Section 2 describes the pattern representation, along with its use in characterizing base extractor performance and aggregating entity output. In Section 3, we discuss enhancements that enable the PME to adapt to sparse data conditions. Finally, experimental results are presented in Section 4, with conclusions and future research given in Section 5.

2 Extractor characterization

In the following discussion, we assume that an entity can be expressed as a text string that is associated with a *location* in the source text. To enable the characterization of base extractor performance, we assume an annotated set of documents is available (distinct from those used for training) to serve as an "evaluation corpus" for the base extractors. The *ground truth* entity data, G, consists of the true (i.e., manually annotated) entities identified in the evaluation corpus. The meta-extractor aggregates the output of $K > 1$ base entity extractors, where D_k

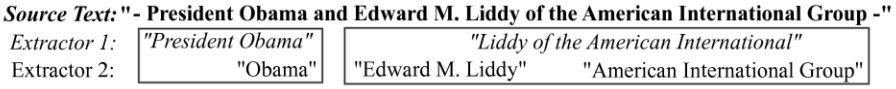


Fig. 1. Meta-entities formed from extracted data: "President Obama", "Edward M. Liddy of the American International Group".

denotes the output of extractor *k* relative to a corpus. When the locations of a ground truth entity and an extracted entity intersect, we say that the entities *overlap*.

2.1 The pattern representation

Named entity extractors leverage a variety of different methodologies to correctly extract fragments from text that represent real-world entities, such as people, organizations, or locations. Many extractors are proprietary, and hence, direct analysis of the characteristic error processes of their underlying algorithms is often infeasible. Therefore, we choose to treat each extractor as a "black box". However, when the base entity extractors are applied to a corpus for which the ground truth, *G*, is known, mistakes in their output, D_k, represent an observable transformation of the truth that is driven by their underlying error processes. The PME utilizes an encoding of the combined base extractor output, **D**, that encodes the joint characteristics of the extractors' output and resultant errors.

To lay a foundation for this encoding, we revisit a construct originally proposed in [10] called the *meta-entity*. This meta-extraction methodology assumed that the combined entity output of the base extractors at a given location in the corpus encapsulates all available information regarding the ground truth. Hence, to facilitate discovery of the truth, mutually overlapping entities output by the *K* base extractors may be concatenated to form a *meta-entity*, which in turn can be used to generate a space of hypotheses over the ground truth. For example, in Figure 1, the extracted data within each rectangle can be concatenated to form two distinct meta-entities consisting of the following fragments of text:

(i) "President Obama"
(ii) "Edward M. Liddy of the American International Group"

Let D_{mk} denote the entity output of base extractor *k* used to form meta-entity m, and let $\mathbf{D}_m = \{\mathbf{D}_{m1},...,\mathbf{D}_{mK}\}$. Note that $\mathbf{D}_m$ consists of the *K*-way joint entity output of the *K* base extractors and possesses a distinctive structure that can be characterized by the boundaries of its individual entities. Specifically, the locations of its entity boundaries collectively define a *K*-way pattern, $\mathbf{d}_m$, relative to m that can be encoded numerically via the following process (illustrated in Figure 2):

(A) Meta-entity *m* is partitioned into *s* segments terminating at the $s+1$ unique entity boundaries in $\mathbf{D}_m$.
(B) For each extractor *k*, a string of length *s* (a 1-way or *simple* pattern denoted d_{mk}) is constructed, in which "2" indicates the beginning of an entity, "1" represents the middle or end of an entity, and "0" indicates that the segment was not extracted by extractor *k*.
(C) We represent the *K*-way pattern corresponding to the segmented meta-entity *m* by $\mathbf{d}_m = \{d_{m1},...,d_{mK}\}$.

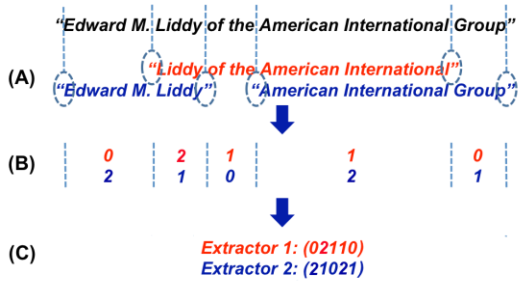


Fig. 2. The pattern-based encoding associated with extracted data relative to a meta-entity.

Note that this segmentation strategy is motivated by the assumption that, if two words in the meta-entity remain "unbroken" by the base extractors (e.g., "American International" in Figure 2), then they most likely remain unbroken in ground truth. Empirically, we have found that the performance of the PME appears to benefit from this assumption.

When the ground truth, G_m, associated with a meta-entity *m* is known and the above assumption is made, an analogous simple pattern representation of ground truth can be derived from the meta-entity segmentation. For example, in Figure 2, the ground truth is given by G_m = {"Edward M. Liddy", "American International Group"}, and its associated pattern is given by $g_m = (21021)$.

2.2 The pattern dictionary

The pattern-based encoding described in the previous section relies solely on the joint *structure* of the entity data being encoded relative to a given segmented meta-entity. Consequently, a particular *K*-way pattern of extracted data may be repeatedly observed in a corpus regardless of the actual text involved in the associated meta-entities. For example, in Figure 3, the extracted data are associated with a joint pattern identical to that shown in Figure 2. However, despite the similar encoding of the extracted data, their associated ground truths differ. In particular, the ground truth in Figure 3 is given by G_m ={"Joe Biden", "Delaware"}, with the associated pattern $g_m = (02002)$. Hence, a particular pattern of extracted data, $\mathbf{d}_m$, may be associated with many different ground truth patterns; in fact, the total number a_s of unique ground truth hypotheses that may be encoded for a meta-entity of length *s* segments is given by $a_0 = 1, a_1 = 2, a_s = 3a_{s-1} - a_{s-2}$. Clearly, only a subspace of the possible encodings will be observed in the training data for long patterns. Indeed, in practice, as pattern length increases, the relative size of this observed subspace shrinks rapidly. Some implications of this behavior will be discussed in later sections.

In an operational setting, the base entity extractors are applied to a corpus for which ground truth is unknown. With access to *only* the extracted entity output of its *K* extractors, the PME must determine the most likely ground truth (i.e., the set of *true* named entities, *G*). This process involves forming a collection of meta-entities from the extractor output, $\mathbf{D}$, and for each meta-entity *m*, determining the ground truth hypothesis that is most plausible in a Bayesian sense among the a_s possible hypotheses. We will show that the optimal ground truth hypothesis H_m^*, given $\mathbf{D}_m$, is that most frequently associated with the *K*-way pattern $\mathbf{d}_m$ in the evaluation data set.

Evaluation of base extractor performance relative to an annotated data set consists of constructing a database, or *pattern dictionary*, from the evaluation data that stores counts of observed ground truth patterns for each *K*-way pattern derived from the extracted data. For example, a final entry in the pattern dictionary might resemble that shown in Figure 4 for the *2*-way pattern presented in Figures 2 and 3.

Consider a particular meta-entity *m* of size *s* having the *K*-way pattern $\mathbf{d}_m$ and unknown ground truth. Let $\theta_1,...,\theta_n (\sum\theta_j = 1)$ denote the respective probabilities of the $n = a_s$ hypothesized ground truths, $H_{m1},...,H_{mn}$. Suppose there are a total of $N = N^{(K)} \geq 1$ occurrences in the pattern dictionary of the pattern $\mathbf{d}_m$. Since the corresponding collection of *N* meta-entities may be regarded as a random sample from the population which generates the pattern $\mathbf{d}_m$, the resulting pattern dictionary counts, i.e., the observed frequencies $f_1,...,f_n (\sum f_j = N)$ of the set of possible ground truths, may be modeled as following a multinomial distribution. The frequency f_j may be viewed as the number of "votes" for the ground truth hypothesis H_{mj}.

The conjugate prior for the multinomial distribution is the Dirichlet distribution $D(\alpha_1,...,\alpha_n)$. For our application, we used a noninformative Dirichlet prior, $D(\alpha_1 = ... = \alpha_n = 1 \setminus n)$, which, in effect, splits a single *a priori* vote evenly among the candidate ground truths.

The posterior distribution of $\theta_1,...,\theta_n$ then, given the observed frequencies $f_1,...,f_n$, is $D(1/n + f_1,...,1/n + f_n)$. These frequencies have the effect of updating the number of votes for hypothesis H_{mj} to $1/n + f_j$. Hence, the marginal posterior distribution of θ_j is the beta distribution with parameters $A_j = 1/n + f_j$ and $B_j = 1 + N - (1/n + f_j)$. It is this distribution that should be used to model the credibility of the hypothesized ground truth H_{mj}. In particular, the posterior mean for θ_j is given by

$$\tilde{\theta}_j = E(\theta_j \mid f_1,..., f_n) = \frac{1}{1+N}\frac{1}{n} + \frac{N}{1+N}\frac{f_j}{N},$$

which is a weighted average of the prior mean, 1/*n*, for θ_j and the sample proportion, $\hat{\theta}_j = f_j / N$, of observed patterns associated with H_{mj}.

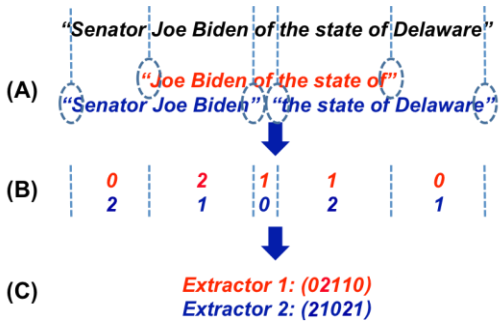


Fig. 3. The joint pattern representation for a different collection of extracted data, identical to that in Fig. 2.

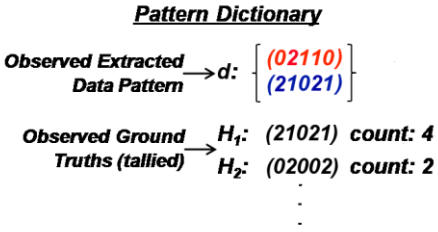


Fig. 4. Example pattern dictionary entry.

The Bayesian optimum ground truth hypothesis H_m^* is the H_{mj} that maximizes the posterior mean $\tilde{\theta}_j$. Moreover, it is apparent from the formulation that it is equivalent to maximize $\hat{\theta}_j$. Hence, the optimal hypothesis is simply that most frequently associated with the *K*-way pattern $\mathbf{d}_m$ in the evaluation data set, easily determined via the pattern dictionary.

3 Unprecedented patterns

When new extractor output $\mathbf{D}_m$ is encountered in the field, it may happen that the associated *K*-way pattern, $\mathbf{d}_m$, was not observed in the evaluation data set and, consequently, cannot be found in the pattern dictionary ($N^{(K)} = 0$). We present two enhancements of the PME that enable it to adapt to these challenging conditions.

3.1 Stepping down

The *K*-way pattern described above is a joint model over the *K* extractors and their corresponding behavior with respect to a given meta-entity. It is reasonable to assume that the pattern algorithm, if necessary, can utilize progressively weaker marginal models in an effort to capture some patterns that would not otherwise be observed. We call this process "stepping down".

Stepping down involves reducing the number of extractors represented by the patterns in the dictionary in an effort to increase the likelihood that a given joint pattern will have been observed. This means that, in building the pattern dictionary, we must additionally store counts of observed ground truth patterns for each *k*-way pattern derived from the extracted data, $1 \leq k \leq K$ - 1. During operation of the PME, when a *K*-way pattern cannot be found in the dictionary, frequencies of these smaller *k*-way patterns, $k < K$, are used to determine plausible

ground truth. The particular value of k employed will be referred to as the stepping down *level*.

Here, we focus chiefly upon two approaches to implementing this stepping down procedure, simple k-way and LBM.

3.1.1 Simple k-way decision

A straightforward implementation of stepping down involves querying the dictionary for all possible k-way patterns, for successively smaller k, $k < K$, until one or more patterns is found. A K-way pattern $\mathbf{d}_m$ induces $T = \binom{K}{k}$ k-way patterns $\mathbf{d}_{mt}$, $t = 1, \ldots, T$, according to the combination of extractors represented. As shown in Figure 5, each k-way pattern $\mathbf{d}_{mt}$ and its associated ground truth patterns are reconfigured, if necessary, to comply with the segmentation induced by the s-segment K-way pattern $\mathbf{d}_m$. Again, let $\theta_1,\ldots,\theta_n$ denote the respective probabilities of the $n = a_s$ possible ground truths, $H_{m1},\ldots,H_{mn}$. Suppose there are a total of $N_t \geq 0$ occurrences in the pattern dictionary of the pattern $\mathbf{d}_{mt}$, with $N = N^{(k)} = \sum N_t \geq 1$. Since we regard the corresponding collection of N meta-entities as a random sample from the population which generates patterns from $\cup_t \mathbf{d}_{mt}$, the resulting pattern dictionary counts, i.e. the observed frequencies $f_1, \ldots, f_n$ ($\sum f_j = N$) of the set of possible ground truths, may again be modeled as following a multinomial distribution. Here the frequencies are pooled over the T k-way pattern dictionaries. Bayesian inferences proceed as in the full K-way case, with the same expressions for $\tilde{\theta}_j$ and $\hat{\theta}_j$. Analogous Bayesian intervals may be constructed.

While this approach has been shown to be reasonably effective, it does not explore and compare probability estimates for all extractor combinations at all values of k. To this end, we have developed an alternative approach that does so.

3.1.2 Lower Bound Maximization (LBM)

The essence of the LBM method consists of stepping down to the "best" combination of extractors, subject to a constraint on the reliability of the estimated probability of the top-ranking hypothesis associated with each combination. The LBM method uses the lower Bayesian bound as a metric to compare hypotheses' probability estimates. Specifically, for each combination of base extractors i, the lower bound on the estimated probability of hypothesis H_{mj}, denoted by $x = l^{(i)}(H_{mj})$, is the solution to

$$I_x(A_j^{(i)}, B_j^{(i)}) = \alpha ,$$

where I_x denotes the incomplete beta function, and the parameters of the corresponding beta distribution are computed in a fashion similar to that described in the preceding section.

The parameter α < 0.5 is pre-specified such that $1-\alpha$ indicates the desired degree of confidence in a bound. Since higher bounds suggest greater plausibility, by comparing the bounds over all levels and hypotheses, we effectively are able to rank the ground truth probabilities. The LBM optimum ground truth hypothesis, H_m^*, achieves the largest bound, i.e.

$$H_m^* = \arg\max_{H_{mj}} \left(\max_i l^{(i)}(H_{mj}) \right).$$

Empirically, we have found the LBM method to be fairly insensitive to the choice of α.

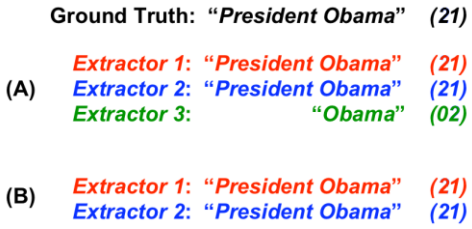


Fig. 5. The 2-way pattern representation formed by Extractors 1 and 2 (B), as well as that of its associated ground truth, maintains the sementation of the original 3-way pattern (A), despite the lack of disagreement between the two extractors.

In a similar fashion as stepping down, LBM simultaneously addresses both the quality and uncertainty of estimates by assigning heavier weights to hypotheses associated with more observations $N^{(i)}$. Moreover, by introducing a confidence metric, it provides an avenue for directly comparing the estimates arising from the totality of possible extractor combinations.

3.2 A Sequential Meta-Entity Model

Although the marginal models utilized in Section 3.1 enhance the PME's ability to make decisions under sparse data conditions, there certainly remain cases in which even these techniques are unsuccessful.

Recall from our previous discussion that the K-way pattern encodes joint information among the errors as well as among the base extractors. In many cases, the rarest of meta-entities consist of lengthy patterns, which represent a complex sequence of errors and disagreement among the extractors. Moreover, the underlying dependencies among extractors is unknown. Thus, it is reasonable to incrementally break down a K-way pattern across errors, rather than across extractors, so that the patterns arising from a single meta-entity are represented by progressively fewer segments. We can address this approach via a sequential modeling technique that is often used in other language-based applications. For example, let us consider a 3-way pattern $\mathbf{d}_m$, together with a hypothesis H_{mj}, as a sequence of columns as shown in Table 1.

We can decompose the joint probability of the pattern ($\mathbf{d}_m$, H_{mj}) in Table 1 as follows:

$$P(\mathbf{d}_m, H_{mj}) = P(c_1) \prod_{t=2}^{4} P(c_t \mid c_{t-1},\ldots,c_1)$$

Table 1: Columnwise representation of a pattern and corresponding hypothesis.

	c_1	c_2	c_3	c_4
d_{m1}	2	1	2	1
d_{m2}	2	1	0	2
d_{m3}	2	0	0	2
H_{mj}	2	1	0	2

where each column pattern is dependent upon those that precede it. Hence, when a complex pattern is encountered that cannot be handled by the previously described methods, we make the assumption that each column pattern is dependent only upon the preceding *n* columns, with $n < s-1$, giving

$$P(\mathbf{d}_m, H_{mj}) = P(c_1)\prod_{t=2}^{s} P(c_t \mid c_{t-1},...,c_{t-n}).$$

Under this framework, we select the hypothesis H_m^* that satisfies

$$H_m^* = \arg\max_{H_{mj}} P(\mathbf{d}_m, H_{mj}).$$

Note that taking $n=1$ in this sequential modeling approach yields a standard Markov model. We have generally found this small window size to be fairly effective, requiring the least amount of data to obtain reliable probability estimates.

4 Empirical studies

In this section, we present results from three aggregation experiments using the output of (1) GATE, a rule-based extraction tool [11]; (2) LingPipe, an extraction tool based on Hidden Markov Models (HMMs) [12]; (3) Stanford Named Entity Recognizer (SNER), based on CRFs [13]; and (4) BALIE, an extraction tool that utilizes unsupervised learning [14]. These experiments were carried out using two publicly available annotated data sets, MUC6 (Wall Street Journal) and MUC7 (New York Times), as well as a small operational data set called TAI consisting of 40 annotated documents (containing approximately 700 ground truth entities).

The following studies compare the performance of the PME where stepping down is implemented up to *n* levels, $n = 0,...,3$ (i.e., "PA*n*"), together with the LBM method ("LBM"). In all cases, when a pattern could not be found in the pattern dictionary after stepping down or LBM was employed, we utilized the Sequential Modeling algorithm to determine a winning hypothesis.

We focused on two relevant real-world scenarios. The first involved a test in which the base extractors and the PME used identical training data. The PME, which requires annotated data for evaluation, necessarily used base extractors trained on less data, thus pitting these weak learners against their stronger, standalone versions. To this end, MUC6 and MUC7 were used in a 10-fold cross-validation procedure where, for each fold, 10% of the corpus was set aside for testing, and the remaining 90% was used to train and evaluate the base extractors (via 9-fold cross-validation). The resulting ten performance estimates were bootstrapped (2000 samples) and presented in box plots (Figures 6 and 7).

The second scenario involved more challenging conditions in which the base extractors were not trained using representative data. We simulated these conditions by training the base extractors on MUC6 and then evaluating their performance and aggregating their output on TAI. As in the first scenario, we performed 10-fold cross-validation, and the resulting estimates were bootstrapped and plotted (Figure 8).

4.1 Results

In the following figures, we have presented our results in terms of *F Measure*, where the Precision, *P*, and Recall, *R*, given by

$$P = \frac{c + 0.5 * p_E}{E},\ R = \frac{c + 0.5 * p_G}{G},$$

where *G* and *E* are the number of ground truth and extracted entities, respectively; p_G and p_E are partial matches of the ground truth and extracted entities, respectively, and *c* is the number of correct extractions (i.e., true positives). This formulation for Precision and Recall is motivated by an interest in quantifying the usability of extracted data, under the assumption that a partially correct extraction is more valuable than a miss, but less valuable than a correctly extracted entity.

In addition to *F* Measure, we have presented our results in terms of Exact Match (EM) rates, and the combined Miss and False Alarm rates for each base extractor and the PME variants. These error types are often traded off to address operational requirements, but here we focus on their combined impact.

We also assessed statistical significance relative to F Measure via a nonparametric pairwise test performed on the results from the original ten folds.

Figures 6 and 7 present the results generated for the first experimental scenario. For both MUC6 and MUC7, the base extractors founded upon statistical methodologies, LingPipe and SNER, produced *F* measures that significantly exceeded those of GATE and BALIE (p = 0.001). In general, we expected this behavior, since statistical methodologies often excel when they are trained on representative data. However, the performance of GATE greatly exceeded that of BALIE. BALIE was trained on a set of prepackaged untagged websites, negatively impacting its performance in our experiments.

Note that, although the EM rate of the LBM method was roughly equivalent to the EM rate of LingPipe for the MUC6 experiment, LBM produced a lower error rate than SNER and, consequently a significantly higher F measure (for MUC6, p = 0.001; for MUC7, p = 0.005).

Note that for both the MUC6 and MUC7 experiments, stepping down with respect to the number of base extractors results in a significantly improved *F* measure (with a p-value $\leq$ 0.002 in

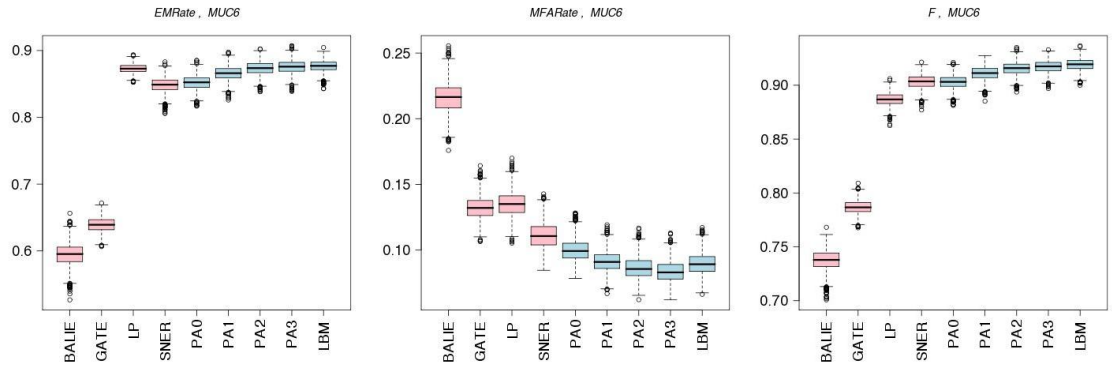


Fig. 6. Left to Right: Exact match rates, Miss + FA rates, F measure on MUC6 for the first experimental scenario. "PA*n*" represents the pattern algorithm, using the simple *k*-way decision stepping down process to step down up to *n* levels. "LBM" presents results from the LBM method, alpha = 0.3. Patterns not found are processed using the Sequential Modeling method.

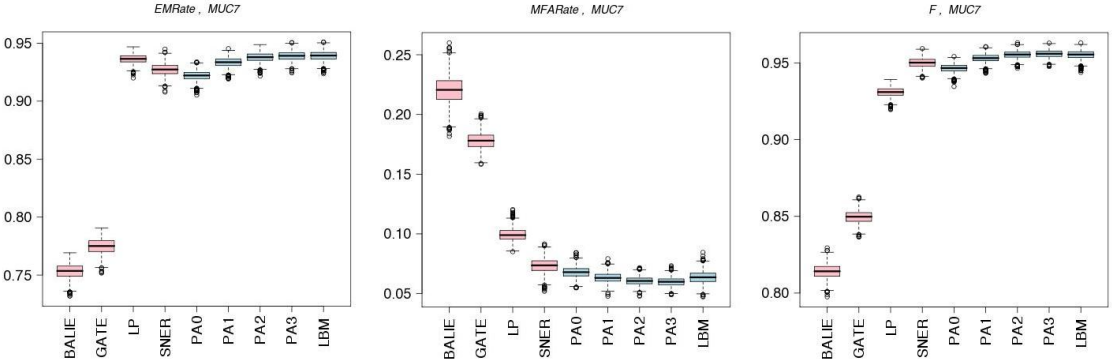


Fig. 7. Left to Right: Exact match rates, Miss + FA rates, F measure on MUC7 for the first experimental scenario. BALIE and GATE performed poorly relative to LP and SNER, much like MUC6. The LBM again uses alpha = 0.3.

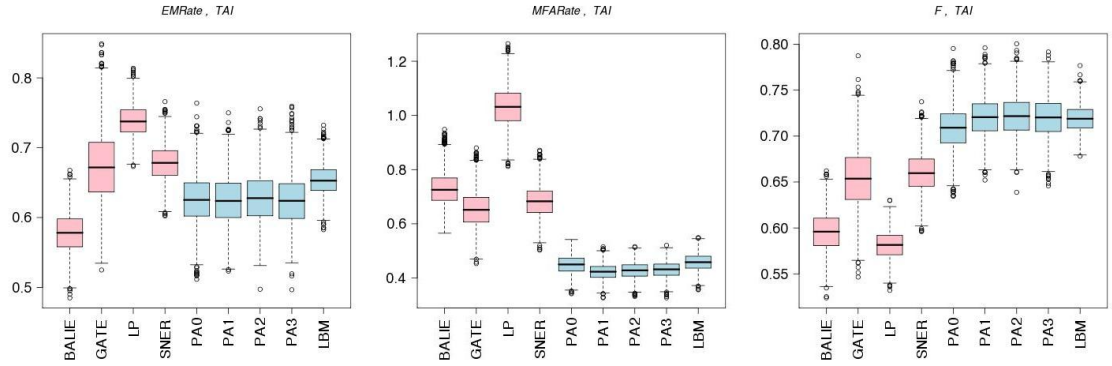


Fig. 8. Left to Right: Exact match rates, Miss + FA rates, F measure on TAI for the second experimental scenario. The performances of BALIE and GATE were more robust relative to LP and SNER. The LBM again uses alpha = 0.3.

each case). These results suggest that the PME benefits from stepping down as far as possible before reverting to the Sequential Modeling method (i.e., when a pattern cannot be found at the lowest level). However, we have observed that it is sometimes advantageous to interrupt the stepping down process and defer the decision to the sequential method, particularly when the data are sparse.

In the second experimental scenario, we examine results from the TAI data set. The TAI data set was roughly one-tenth the size of MUC6 (which is roughly half the size of MUC7), and was annotated according to MUC6 guidelines. As it turns out, this annotation was poorly performed with many underlying true entities unidentified. Hence, this situation mimics those of the second condition described above (i.e., annotations used to train the base extractors are flawed). Specifically, we may regard the incomplete TAI annotations as a relevance-based annotation, in which only entities of interest have been identified relative to some operational need. In such a case, MUC6 turns out to be nonrepresentative, and base extractors trained on MUC6 are poorly equipped to perform effectively when applied to TAI.

The results for TAI are presented in Figure 8. It is clear from the plot that, as in other experiments, the PME successfully mitigated the decreased performance of the base extractors.

Note that GATE's performance remained relatively robust, as it does not require training and, hence, is not susceptible to the flaws in the training data set. SNER's performance degraded significantly, but at least produced results comparable to GATE. The performance of LingPipe dropped precipitously, largely because its error rate increased by nearly an order of magnitude. Indeed, it produced roughly one false alarm per ground truth entity. We have observed that our version of LingPipe tends, in general, to produce more false alarms than other methods.

With respect to the PME and LBM in Figure 8, their respective performance was not found to be statistically different (p = 0.55), but the results again indicate that the LBM is competitive with the PME.

5 Conclusions and future work

In this paper, we have presented a pattern-based aggregation methodology – the PME – that implicitly incorporates the joint behaviors of extractors and their error processes. Through the integration of marginal models and corresponding representations of extracted data, the PME has proven to be highly effective. Specifically, it has been shown to achieve statistically significant improvements in the summary metric, *F* Measure, over its base entity extractors in multiple experimental scenarios and on multiple data sets. Even under sparse data conditions, where marginal models become more critical, the PME remains highly effective.

Strategies for integrating across multiple marginal models under these conditions were also presented and their relative performance compared. The simple *k*-way decision, though generally effective, makes the decision to step down based only upon the absence of a pattern in the pattern dictionary, without regard to uncertainty or accuracy across levels. As a consequence, decisions may sometimes be made by few or highly variable data.

An alternative approach to the *k*-way decision, the LBM method, is able to account for the uncertainty across the various extractor combinations. Specifically, this method selects an optimum hypothesis according to a Bayesian lower bound metric appropriate and applicable across all of the combinations. As a result, it is competitive with the best-performing PA*n* algorithm in each of these empirical studies relative to *F* Measure.

Both of the methods require that a parameter be specified for optimal performance. Specifically, the *k*-way decision requires the selection of the minimum level *k*, while the LBM method requires that the parameter α be specified. However, our studies have shown that the LBM method is fairly insensitive to the choice of α, and for the *k*-way decision, the choice of $k = 1$ as the minimum level is often the most effective.

In text applications, a wide variety of meta-entities is observed. These meta-entities can be distinguished by structural features derived from their underlying patterns of base extractor text. Other research we have performed has demonstrated that the effectiveness of different aggregation algorithms can be linked directly to these characteristic features. Consequently, our future efforts will investigate systems that can assign meta-entities to the most favorable given specific operational conditions and meta-entity features.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

6 References

[1] Kozareva, Z., Ferrández, O., et al. 2007. Combining data-driven systems for improving named entity recognition. *Data & Knowledge Engineering*, 61-3 (Jun. 2007), 449-466. doi:10.1016/j.datak.2006.06.014.

[2] Duong, D., Goertzel, B., et al. 2006. Support vector machines to weight voters in a voting system of entity extractors. In *Proc. IEEE World Congress on Computational Intelligence* (Vancouver, Canada, 2006), 1226-1230. doi:10.1109/IJCNN.2006.246831.

[3] Wang, H. and Zhao, T. 2008. Identifying named entities in biomedical text based on stacked generalization. In *Proc. 7th World Congress on Intelligent Control and Automation* (Chongqing, China, 2008), 160-164. doi:10.1109/WCICA.2008.4592917.

[4] Wu, D., Ngai, G. and Carpuat, M. 2003. A stacked, voted, stacked model for named entity recognition. In *Proc. CoNLL-2003*, 4 (Edmonton, Canada, 2003), 200-203. doi:10.3115/1119176.1119209.

[5] Florian, R. 2002. Named entity recognition as a house of cards: classifier stacking. In *Proc. 6th Conference on Natural Language Learning*, 20 (Taipei, Taiwan, 2002), 1-4. doi:10.3115/1118853.1118863.

[6] Kambhatla, N. 2006. Minority vote: at-least-N voting improves recall for extracting relations. In *Proc. COLING/ACL on Main Conference Poster Sessions* (Sydney, Australia, 2006), 460-466.

[7] Kegelmeyer, P. and Goldsby, M. Massive ensembles for mindlessly improving named entity recognition. Unpublished.

[8] Florian, R., Ittycheriah, A., et al. 2003. Named entity recognition through classifier combination. In *Proc. CoNLL-2003*, 4 (Edmonton, Canada, 2003), 168-171. doi:10.3115/1119176.1119201.

[9] Si, L., Kanungo, T. and Huang, X. 2005. Boosting performance of bio-entity recognition by combining results from multiple systems. In *Proc. 5th International Workshop on Bioinformatics* (Chicago, IL, 2005), 76-83. doi:10.1145/1134030.1134044.

[10] Lemmond, T., et al. 2010. Enhanced Named Entity Extraction via Error-Driven Aggregation. In *Proc. Intl. Conference on Data Mining* (Las Vegas, NV, Jul., 2010), 31-37.

[11] Cunningham, H., Maynard, D., et al. 2002. GATE: a framework and graphical development environment for robust NLP tools and applications. In *Proc. 40th Anniversary Meeting of the Assoc. for Computational Linguistics* (Philadelphia, PA, 2002).

[12] Alias-I, LingPipe 3.8.2, 2008. http://alias-i.com/lingpipe.

[13] Stanford University, Stanford Named Entity Recognizer 1.1, 2008. http://nlp.stanford.edu/software/CRF-NER.shtml.

[14] University of Ottawa, Baseline Information Extraction (BALIE) 1.81, 2004. [Online] http://balie.sourceforge.net/.

Sentiment Detection with Character n-Grams

Tino Hartmann, Sebastian Klenk, Andre Burkovski and Gunther Heidemann

***Abstract*—Automatic detection of the sentiment of a given text is a difficult but highly relevant task. Application areas range from financial news, where information about sentiments can be used to predict stock movements, to social media, where user recommendations can determine success or failure of a product.**

We have developed a methodology, based on character n-grams, to detect sentiments encoded in text. In the course of this paper we will present the founding idea and the algorithms as well as a usage scenario with an evaluation. We discuss the the obtained results in detail and a compare them with those of other popular sentiment detection methodologies.

I. Introduction

Sentiment detection is an important aspect of unstructured text analysis. Automatically determining the feelings that a text is expressing is becoming increasingly important as more and more content is generated. Especially for companies the knowledge about consumer sentiment is of high value. Social media and user generated content are more and more forming public opinion. A decade ago consumer decisions were mostly based on experiences of close friends and a selective list of publications. Today, social media gives access to experiences of several thousand consumers and public opinion is formed by a vast network of users contributing and sharing information. One aspect of this social opinion generation process is that the overall sentiment is not determined by a few individuals but by an aggregation of all the available sentiments. Therefore it is necessary to be able to automatically analyze user generated content.

In this paper we want to contribute to this reseach task by presenting a sentiment detection method based on character n-grams. Here, as opposed to word n-grams, one is capable – by a suitable choice of n – to detect interword dependencies without overboarding combinatorics. Word n-grams require an exact n-tuple word match, character n-grams require only n characters to match, which (*i*) is more likely (for small n) and (*ii*) allows to match text stems without any sophisticated language model.

Character n-grams are a rather popular and simple method in natural language processing and information retrieval [1], [2]. In the course of this paper we will present a method to compare character n-grams based on the cosine distance. There the so called "Length Delimited Dictionary Distance" (LD^3) forms a very simple but efficient way to measure the distance between documents. The originating idea thereof stems from the Normalized Compression Distance [3], [4].

In the course of this paper we will present character n-grams as a means to determine sentiments of texts. We will present the rationale behind it, the algorithm as well as some experiments that demonstrate its applicability. We will further analyze, whether character n-grams are suited for determining text sentiment. For this purpose, we try to classify the popular IMDb dataset [5], using n-grams as terms with a Naive Bayes classification, and compare the results with other existing methods.

Tino Hartmann Sebastian Klenk, Andre Burkovski and Gunther Heidemann are with the Intelligent Systems Department, University of Stuttgart, Stuttgart, 70569, Germany (phone: +49 (0)711 685 88-241; email: klenksn@vis.uni-stuttgart.de).

II. Related Work

Because of the inherent complexity of the task and the lack of a model that is generally agreed on, there is a vast variety of approaches trying to tackle the text sentiment problem. The most basic approach is to model text as a *bag of words*, neglecting all compositional structure. Every single word gets labeled with a polarity score, which represents the probability of the word being in a positive or a negative text. The polarity of the text is then defined as the sum of all word polarities. Polarity scores for terms can either be manually constructed [6] or inferred via machine learning techniques [7], [8]. Manually constructed reference sets have always the problem of coverage, which means that most of the domain specific words will not be enclosed in an universal reference set. Some work focuses on the construction of domain specific sets and the adaption of existing ones of other domains [9]. Such a domain specific set can be inferred with a number of techniques, for example seeds, i.e. words with known polarity like *"good"* and *"poor"* and a proximity measure between words, like mutual information [7] or the WordNet[10], [11].

One major problem is that most of the sentences of a document do not express any sentiment. They will only add noise to the classification process. Therefore, there has been the attempt to classify the objectivity on sentence level [12]. The polarity estimate is then based only on the sentences which were classified as *subjective* beforehand. It is possible to go even further and try to determine which topic a given sentiment addresses. Instead of assuming, that a text only consists of sentiments about a single topic, every document is modelled as a collection of sentiments about many topics. A review of a book may contain sentiment about the author, which can be a different from the sentiment about the book. For example, Mullen [11] tries to determine topic proximity via the open ontology tool [13].

All of these basic approaches have a good baseline performance, but there seems to be a certain barrier of accuracy that all of them can not overcome. Ironically, they perform only slightly better than simple machine learning approaches. It seems to be obvious that the reason for this lies in the neglection of words interdependencies, i.e. the structure and the context of the text. To model this structure, the compositional semantics, there is a variety of approaches.

A very basic approach to model sentence interdependencies is to look at negation only [14], more sophisticated is it to try to build semantic hierarchies via manually constructed rules [15] or to improve the classification based on words by simple linguistic rules [16]. The most promising approach seems to be a combination of machine learning techniques (like SVMs), pattern and sub pattern recognition and analysis of the grammatical structure of sentences [17]. Further information on the subject of sentiment detection can be found in the very extensive survey on sentiment detection, that has been done by Pang and Lee [12]. In this paper we are not concerned with more advances models. We are trying to detect sentiments with as little prior knowledge as possible.

III. The Movie Database

Sentiment detection, like any other pattern recognition and machine learning problem is highly depending on the quality of the data. We chose the IMDb movie review database as test scenario, because it is probably one of the most commonly used data sets for sentiment detection. The IMDb is a freely accessible library containing information to countless movies. Besides featured actors or information on the director the site also contains movie reviews which can be found at http://reviews.imdb.com/Reviews. There, one has access to over 41,000 movie reviews, written in plain English. Unfortunately, the data format varies and there is no common rating scale, which makes an automated use of this dataset difficult. However, a formatted dataset has been made available at http://www.cs.cornell.edu/people/pabo/movie-review-data/ which has grown very popular among sentiment detection researchers (used in [12], [17] for example). It consists of 1,000 both positive and negative reviews.

The IMDb dataset has proven to be especially difficult. One problem all algorithms have, that try to tackle sentiment detection with word counting, is that for example *"good"* and *"not good"* have opposite meanings. Algorithms based on word occurrence will match *"good"* both times which means that both phrases get a high positive weighting due to the occurrence of *"good"* which in the latter case is plain wrong. Das and Chen [14] tried to eliminate this problem by marking all words between a negating word and the next punctuation with a special tag, so that "good" and the word good in "not good" will count as different word. We will call the IMDb dataset, which is tagged with this rule IMDb-NOT.

IV. Text classification

When classifying text there are a large number of possible methods to choose from. Probably the most well known is the Naive Bayes classifier [18] with its simplistic approach. Besides that we will present two rather new approaches to text classification based on character n-grams.

A. Naive Bayes

The Naive Bayes classifier is a very common approach to statistical text classification. It is based on the – obviously naive – assumption that the occurrence of one term $t \in D$, given a document class C, is independent of the occurrence of any other term. Therefore, if we disregard the interdependency of term $t \in D$ with all other terms $t' \in D$ the conditional probability of the document D being a member of class C is simply:

$$P(C|D) = P(C) \prod_{t \in D} P(t|C) \tag{1}$$

The prior probability $P(C)$ of any document being in class C is estimated as follows:

$$P(C) = \frac{\#D_C}{N} \tag{2}$$

Where $\#D_C$ is the number of documents in the training set that are in class C and N is the number of documents in the training set. If we use a balanced dataset, $P(C)$ is identical for all classes. $P(t|C)$ is estimated as the relative frequency of term t in all documents belonging to class C.

$$P(t|C) = \frac{\#t_C}{\sum_{t' \in C} \#t'_C}$$

Here $\#t_C$ is the number of occurrences of term t in class C. To obtain a probability we are normalizing it with the sum of the occurrence of all terms in C. Although the assumption of positional independence is far from reality, Naive Bayes performs quite well for sentiment detection.

B. Length Delimited Dictionary Distance

In this section we will introduce what we are calling the Length Delimited Dictionary (LDD_k). It is based on the idea of compression based pattern recognition [4], where it is possible to determine dissimilarity of two objects by looking at the ratio of joint compression against individual compression. Let C be a compression algorithm and $C(s)$ be the length of the compressed string s. The normalized compression distance (NCD) is defined as follows:

$$NCD(s_1, s_2) = \frac{C(s_1, s_2) - min\{C(s_1), C(s_2)\}}{max\{C(s_1), C(s_2)\}} \tag{3}$$

Most discrete compression algorithms generate a dictionary $W(D)$ to compress a document D. This dictionary is simply a list of substrings (*words*) w, all of which have preferably high frequency in D. If the compression algorithm finds a word w of the dictionary in the string, it replaces this word with a shorter one. If there is no occurrence of w in D, w does not contribute to the compression of w. If there is no occurence of *any* word of the dictionary, the document D will not be compressed or might even get larger. If a dictionary can be used to highly compress a document D_1, but does not compress another document D_2, we can assume that D_1 and D_2 are very dissimilar.

The joint compression of two strings can be very effective, if a dictionary $W(D_1, D_2)$ can be found that compresses both strings effectively. We assume that in this case the dictionaries $W(D_1)$ and $W(D_2)$ are very similar and it is sufficient to compare the dictionaries to determine dissimilarity.

In order to have an intuitive and highly flexible dictionary that can be used to measure the distance of any type of data,

we use a very basic approach for the generation of W: the Length Delimited Dictionary (LDD). Formally speaking the LDD_k of a document D is the set of all substrings of length k (character k-grams).

The Length Delimited Dictionary Distance LD_k^3 of two documents D_1 and D_2 is the number of elements that are common to both dictionaries normalized by the number of unique elements in both dictionaries joint together:

$$LD_k^3(D_1, D_2) = 1 - \frac{|LDD_k(D_1) \cap LDD_k(D_2)|}{|LDD_k(D_1) \cup LDD_k(D_2)|} \quad (4)$$

Interesting to note here is that the LD^3 is identical with the Jaccard similarity coefficient [19] and as such related to the cosine distance for character n-grams.

For sentiment detection we create two dictionaries (consisting of k-grams) $LDD_k(D_0)$ and $LDD_k(D_1)$ where D_i is the class document represented by the concatenation of all documents of class $i \in \{0|1\}$. For each class $C \in \{0, 1\}$ we determined the class membership $C_k(D)$ of document D by calculating the dissimilarity between D and each of the class documents D_0 and D_1.

$$C_k(D) = \arg\min_{i \in \{0,1\}} \{LD_k^3(D_i, D)\} \quad (5)$$

C. Character n-grams with Naive Bayes

LD^3 determines dissimilarity in a black and white manner, either an n-gram exists or not. Naive Bayes on the other hand weights the existence or non-existence. Unfortunately it is too restrictive in such a way that only words or even worse word n-grams are used. We implemented Naive Bayes with character n-grams as a trade-off between the flexibility of the Length Delimiting Dictionary – depending on the length, LDD is capable of representing inter word dependencies – and the problem adaption of Naive Bayes which learns the relevance of strings. This way, as we will demonstrate later on, we are capable to increase the recognition performance beyond that of either one of them alone.

The algorithm is as follows: instead of calculating $P(C|D)$ with word occurrences within a document D, we define d_k to be a LDD_n dictionary element, i.e. a substring of length n. Thus the Naive Bayes formula will be rewritten to

$$P_n(C|D) = P(C) \prod_{d \in LDD_n(D)} P(d|C)$$

with

$$P(d|D) = \frac{\#d_C}{\sum_{d' \in C} \#t'_C}.$$

Here $\#d_C$ is the number of occurrences of dictionary element d in the dictionary consisting of all documents in class C. We will call this classifier $NB(LD_n)$ as opposed to $NB(n)$ for plain Naive Bayes.

V. Evaluation

We tested the $NB(n)$ classifier with 10-fold cross validation on the two datasets IMDb and IMDb-NOT. We compared the results to a Naive Bayes classifier using word n-grams as feature. We call the classifier that uses Naive Bayes with word n-grams $NB(n)$, so $NB(1)$ is a Naive Bayes approach operating on unigrams, $NB(2)$ on bigrams and so forth.

The results of our evaluation can be observed in Figures 1,2 and 3. Detailed information can be found in Table I. For the IMDb-NOT dataset details are presented in Table II. Here there is a slight increase in performance due to the prior information (inform of the encoded negation) stored in the data.

It turns out that the simple LD_n^3 classifier cannot outperform Naive Bayes, but $NB(LD_n)$ performs slightly better than $NB(n)$, i.e. character n-grams are better features than word n-grams. This is interesting, because character n-grams make less assumptions on the underlying data than regular n-grams. The document does not have to be tokenized into words, a simple substring routine is sufficient. As a result, character n-grams can be used on all kinds of data.

As a baseline for the evaluation of the $NB(LD_n)$ classifier we are referencing Pang and Lee. They are classifying the exact same dataset with a number of different classifiers [20]. There Support Vector Machines with unigram feature presence got the best result of 82.9% accuracy. It should be mentioned though that they evaluated the classifier with 3-fold cross-validation which generally yields worse results.

Better results were obtained by Matsumoto *et. al.* [17] with an accuracy of 88.3%. Their solution uses much more knowledge about the underlying data. They incorporate information on text and language, like the grammatical structure of sentences returned by a natural language parser which is not always available or even desirable.

We are also comparing the LD_k^3 distance measure with its origin, the Normalized Compression Distance. Therefore we are creating an intuitive classifier based on compression. The document D was classified as a positive review if the average normalized compression distance to all positive reviews was smaller than the average distance to all negative reviews of the training data. Ten-fold cross validation result obtained for the NCD classification is 63.5%.

VI. Discussion

Originating from the Normalized Compression Distance, the LD_k^3 distance measure does fairly well at the text sentiment classification task. Whereas an NCD classification with 10-fold cross validation reached only 63.5 percent, the $ld(17)$ classifier can achieve an accuracy of 80.4. This is not much compared to other classification solutions done by other authors, but it shows, that the LD_k^3 distance is a suitable and efficient substitution for the NCD distance when it comes to large documents.

A very interesting result is, that character n-grams are a better choice than word n-grams when used with Naive Bayes. This could mean that word n-grams are either too strict in counting evidence or, which is more probable, are requiring a larger training data set. A classification with tri-grams, given the size of the training set, is not optimal, because there simply are not enough tri-gram intersections

TABLE I
ACCURACY RESULTS FOR DIFFERENT CLASSIFIERS BASED ON THE IMDB DATASET.

C	ld1	ld2	ld3	ld4	ld5	ld6	ld7	ld8	ld9	ld10	ld11	ld12	ld13	ld14	ld15	ld16	ld17	ld18
μ	14.7	49.7	50.9	50.3	50.8	51.3	52.2	54.0	57.3	62.3	68.0	73.0	76.5	78.3	79.4	80.3	80.4	79.8
σ	13.20	1.59	0.19	0.02	0.03	0.01	0.04	0.06	0.06	0.14	0.12	0.19	0.07	0.18	0.03	0.15	0.20	0.24

C	nb.ld1.	nb.ld2.	nb.ld3.	nb.ld4.	nb.ld5.	nb.ld6.	nb.ld7.	nb.ld8.	nb.ld9.	nb.ld10.
μ	50.0	52.8	75.6	80.7	82.0	83.4	84.5	84.7	84.9	84.8
σ	0.01	0.06	0.08	0.06	0.10	0.03	0.10	0.09	0.05	0.09

nb.1.	nb.2.	nb.3.
81.7	83.5	81.2
0.08	0.13	0.13

TABLE II
ACCURACY RESULTS FOR DIFFERENT CLASSIFIERS BASED ON THE IMDB-NOT DATASET.

C	ld1	ld2	ld3	ld4	ld5	ld6	ld7	ld8	ld9	ld10	ld11	ld12	ld13	ld14	ld15	ld16	ld17	ld18
μ	14.7	49.3	50.7	50.9	50.9	51.6	52.5	54.8	58.8	63.8	68.9	72.9	76.2	78.4	79.5	80.1	80.0	79.4
σ	18.82	7.46	0.27	0.12	0.02	0.01	0.04	0.05	0.11	0.13	0.14	0.16	0.22	0.12	0.13	0.14	0.23	0.09

C	nb.ld1.	nb.ld2.	nb.ld3.	nb.ld4.	nb.ld5.	nb.ld6.	nb.ld7.	nb.ld8.	nb.ld9.	nb.ld10.
μ	50.0	51.6	74.2	79.8	82.0	83.3	84.1	84.8	85.2	84.9
σ	0.004	0.032	0.206	0.083	0.050	0.120	0.108	0.063	0.139	0.115

nb.1.	nb.2.	nb.3.
81.2	83.8	81.7
0.06	0.02	0.18

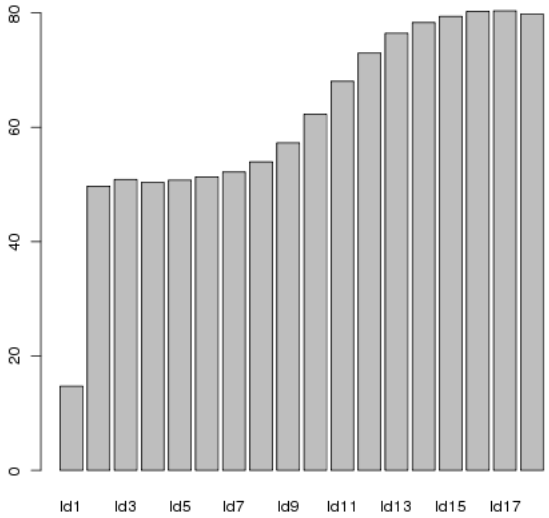


Fig. 1. Accuracy results for the LD_k^3 classifier.

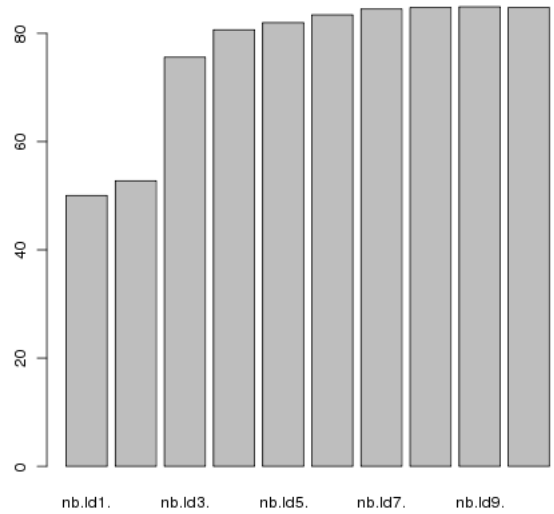


Fig. 2. Accuracy results for the Naive Bayes classifier based on character n-grams.

between the test documents and the training set. The character n-gram perform better, because the probability of finding an exact string of length n in the training set is much higher than finding an n-gram, which makes the character n-gram parameter more flexible. Here one has to keep in mind that in a text of length m there are almost $m * n$ character n-grams whereas only about $(m * n)/k$ word n-grams (given the average word length is about k). Here it would be very interesting to work with much larger labeled datasets and compare the performance.

We have also observed that the quality of the classification is strongly correlated with the number of reviews used, i.e. a classification with a 0.7 training/test ratio will lead to much less classification accuracy than a ratio of 0.9. This leads us to the assumption that there may not be enough reviews in the IMDb dataset, compared to the difficulty of the task. The variance in the data is too high in relation of the amount of data available. This also means that one has to be careful when comparing classification results from different authors even though they are using the exact same dataset. For datasets of similary size to the IMDb dataset classification accuracies calculated with different cross-validation strategies result in difference accuracy values for the exact same classifier.

VII. CONCLUSION

We have demonstrated that for text sentiment classification, character n-grams perform at a high level and are capable of achieving results comparable to highly sophisticated methods. This is especially interesting as character n-gram require an almost minimal amount of prior knowledge, even

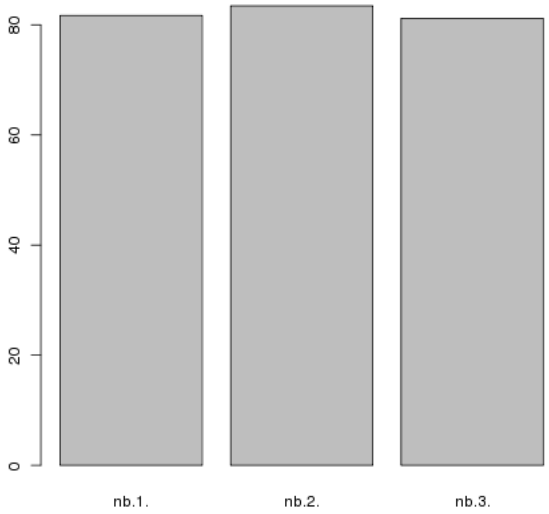


Fig. 3. Accuracy results for the Naive Bayes classifier based on word n-grams.

compared to word n-grams. Character n-grams make less assumptions about the data than, because they model a text as a collection of characters rather than words. Given the size of the available datasets we demonstrated that character n-grams are efficient than more intuitive approaches such as word n-grams. Here it would be of great interest to repeat the presented evaluation on much larger datasets.

REFERENCES

[1] P. Mcnamee and J. Mayfield, "Character n-gram tokenization for european language text retrieval," *Inf. Retr.*, vol. 7, no. 1-2, pp. 73–97, 2004.

[2] Y. Miao, V. Kešelj, and E. Milios, "Document clustering using character n-grams: a comparative evaluation with term-based and word-based clustering," in *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*. New York, NY, USA: ACM, 2005, pp. 357–358.

[3] M. Li, X. Chen, X. Li, B. Ma, and P. Vitanyi, "The similarity metric," *Information Theory, IEEE Transactions on*, vol. 50, no. 12, pp. 3250–3264, Dec. 2004.

[4] R. Cilibrasi and P. Vitanyi, "Clustering by compression," *Information Theory, IEEE Transactions on*, vol. 51, no. 4, 2005.

[5] H. Tang, S. Tan, and X. Cheng, "A survey on sentiment detection of reviews," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10 760–10 773, 2009.

[6] M. Hurst and K. Nigam, "Retrieving topical sentiments from online document collections," in *Document Recognition and Retrieval XI*, 2004, pp. 27–34.

[7] P. Turney, "Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews," 2002, pp. 417–424. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.1992

[8] S.-M. Kim and E. Hovy, "Determining the sentiment of opinions," in *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2004.

[9] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, Bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *Proceedings of the Association for Computational Linguistics (ACL)*, 2007.

[10] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[11] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, July 2004, pp. 412–418, poster paper.

[12] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *In Proceedings of the ACL*, 2004, pp. 271–278. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.9.9144

[13] N. Collier, K. Takeuchi, A. Kawazoe, T. Mullen, and T. Wattarujeekrit, "A framework for integrating deep and shallow semantic structures in text mining," in *KES*, 2003, pp. 824–834.

[14] S. R. Das and M. Y. Chen, "Yahoo! for Amazon: Sentiment extraction from small talk on the Web," *Management Science*, vol. 53, no. 9, pp. 1375–1388, 2007.

[15] A. Fahrni and M. Klenner, "Old Wine or Warm Beer: Target-Specific Sentiment Analysis of Adjectives," in *Proc.of the Symposium on Affective Language in Human and Machine, AISB 2008 Convention, 1st-2nd April 2008. University of Aberdeen, Aberdeen, Scotland*, 2008, pp. 60 – 63.

[16] X. Ding and B. Liu, "The utility of linguistic rules in opinion mining," in *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM, 2007, pp. 811–812.

[17] S. Matsumoto, H. Takamura, and M. Okumura, "Sentiment classification using word sub-sequences and dependency sub-trees," in *PAKDD*, 2005, pp. 301–311.

[18] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, 1st ed. Cambridge University Press, July 2008.

[19] J. Han and M. Kamber, *Data mining*. Morgan Kaufmann Publ., 2001.

[20] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002, pp. 79–86.

Objective Words Can Improve Sentiment Classification for Word of Mouth

Chihli Hung*, Hao-Kai Lin, Chih-Fong Tsai

***Abstract*—Word of mouth (WOM) has a strong effect on consumer behavior. A sentimental lexicon, i.e. SentiWordNet is useful for the task of WOM sentiment classification. However, most current related research ignores the problem that too many objective words are defined in SentiWordNet. In this research, we focus on the effect of objective words on the performance of WOM sentiment classification, and propose a novel sentimental relevance approach. We analyze the co-relation of each objective word and its associated sentences in order to modify the sentimental score and tendency for the objective word. This semantic-oriented approach integrated with a machine learning-oriented approach, i.e. support vector machine, is capable of making an improvement in WOM sentiment classification.**

I. Introduction

Word of mouth has become the main information resource while making business or buying strategies and has been proven to have a powerful effect on consumer behavior [1-3]. Esuli and Sebastiani [4] built a sentimental WordNet [5] lexicon, namely SentiWordNet, which is useful for mining WOM efficiently. SentiWordNet provides each synonym set (synset) of WordNet with three sentimental labels regarding positivity, objectivity and negativity. The score of each sentimental label is between 0 and 1. The greater sentimental score contains the greater sentimental tendency and the total score of three sentimental labels is equal to 1 as in (1).

$$pw_i + ow_i + nw_i = 1, \text{ where } pw_i, ow_i, nw_i \in [0,1] \quad (1)$$

For synset *i*, pw_i, ow_i and nw_i indicate positive, objective and negative score respectively. A word with a stronger positive, objective or negative sentimental tendency is defined as a positive, objective, or negative word respectively and a positive or negative word is further defined as a sentimental word.

SentiWordNet has become a public and popular sentimental lexicon for sentiment classification research [6]. This lexicon includes 117,659 synsets but 93.75% of them are objective words. Most current related research into the use of SentiWordNet for sentiment classification simply ignores objective words [7-9]. However, such objective words may have some effects on sentiment classification [10].

As a result, the aim of this research is to make use of the effect of objective words on sentiment classification. We propose a novel technique based on sentimental relevance of words and sentences to re-evaluate objective words in SentiWordNet and integrate with the support vector machine in order to improve the performance of sentiment classification for word of mouth.

Chihli Hung is with the Department of Information Management, Chung Yuan Christian University, Chung-Li, Taiwan 32023 R.O.C. (email: chihli@cycu.edu.tw)

Chih-Fong Tsai with the Department of Information Management, National Central University, Chung-Li, Taiwan 32023 R.O.C. (email: Cftsai@mgt.ncu.edu.tw)

Hao-Kai Lin is with the Department of Information Management, Chung Yuan Christian University, Chung-Li, Taiwan 32023 R.O.C. (email: f750502@gmail.com)

II. Methodology

Our research methodology is divided into three modules, which are document preprocessing, extraction of relevant sentimental words and sentiment classification.

A. Document Preprocessing

The module for document preprocessing includes sentence segmentation, part of speech filtering, lemmatizing and removal of stop words. As a sentence usually presents a specific sentimental tendency, we treat a sentence as a processing unit. We segment a document into sentences with some punctuation, such as ".", ";", "!" and "?" . As SentiWordNet contains only open class words, i.e. nouns, verbs, adjectives and adverbs, we keep such words by using the Brill tagger. Finally, we lemmatize a word into its lemma and remove stop words.

B. Extraction of Relevant Sentimental Words

The aim of this module is to re-define objective words by a proper sentimental tendency. Each word may have more than one sense in SentiWordNet. In this case, we look up this word by its first sense as this is the most common sense. We then define the sentimental tendency of a sentence by accumulation of sentimental tendency from its sentimental words. The sentimental score for a sentence j is defined as (2).

$$ss_j = \frac{\sum_{i=1}^{n} (pw_i - nw_i)}{n}, \quad (2)$$

where *n* indicates the total number of sentimental words in sentence *j*.

We define a sentence with a positive sentimental score as a positive sentence, *ps*, and one with a negative sentimental score as a negative sentence, *ns*. The sentimental score of a word is proportional to the significance of its associated sentimental sentences. Generally speaking, a positive sentence should contain more positive words than negative words and vice versa. We therefore propose a novel technique based on the sentimental relevance between an objective word and its associated sentences to re-evaluate the sentimental score of this objective word. We modify the positive score and the negative score of an original objective word, *i*, as (3) and (4) respectively.

$$pw_i = \frac{pr(ps_j, word_i)}{pr(word_i)} \tag{3}$$

$$nw_i = \frac{pr(ns_j, word_i)}{pr(word_i)}, \tag{4}$$

where *ps* indicates a positive sentence, *ns* indicates a negative sentence and $pr(i, j)$ indicates a co-occurring probability of *i* and *j*.

However, modifications of (3) and (4) may not be meaningful enough when the sentimental score of a sentence *j*, i.e. ss_j and sentimental score of a word *i*, i.e. pw_i and nw_i are not great enough. Thus, we set two thresholds for these modifications. One is for the sentimental score of a sentence and the other is for the sentimental score of a word. It is obvious that a greater threshold will modify a smaller number of objective words. It should be noted that the total number of objective words is decreased through this module.

C. Sentiment Classification

The module of sentiment classification integrates a traditional machine learning algorithm, i.e. support vector machine (SVM) [11] with a modified vector space representation model for document classification. A document vector *k* is represented as $D_k = [W_1, W_2, ..., W_m]$, where *m* indicates the total number of words in the data set and W_i is the weight value of word *i*. Such words are filtered by the document preprocessing module and shown in SentiWordNet. The weight value of a positive word is equal to its term frequency multiplied by its positive score while the weight value of a negative word is equal to its term frequency multiplied by its negative score multiplied by (-1). The remaining objective words are assigned a neutral value as shown in (5).

$$W_i = \begin{cases} tf_i \times pw_i, \text{ where word } i \in [\text{positive words}] \\ tf_i \times nw \times (-1), \text{ where word } i \in [\text{negative words}] \\ 0, \text{ where word } i \in [\text{objective words}] \end{cases} \tag{5}$$

III. Experiment Designs

This research uses a movie word of mouth data set, i.e. iMDB in our initial experiments. The iMDB data set includes 27,886 movie review articles and 2,000 of them have been assigned a sentimental tag manually (1,000 positive and 1,000 negative).

We use the whole iMDB data set to modify the objective words in SentiWordNet. As there are only 2,000 articles with a pre-assigned sentimental tag in iMDB, we treat these articles as the data set for the sentiment classification task. The strategy of 10-fold cross validation is used in order to obtain more general results. We compare our proposed approach with a traditional approach and evaluate them based on the criterion of classification accuracy, Acc, is shown in (6) with a help of Table 1.

TABLE I
The Confusion Matrix of Positive and Negative Documents

	Predictive Positive Doc #	Predictive Negative Doc #
Positive Doc #	A	B
Negative Doc #	C	D

$$Acc = \frac{A + D}{A + B + C + D} \tag{6}$$

where *A*, *B*, *C* and *D* are referred to Table 1.

IV. Experiment Results

The concept underlying our proposed approach is that a sentimental sentence is made of its associated sentimental words and therefore this sentence should have some sentimental effect on its associated objective words. In the module for extraction of relevant sentimental words, we set up two thresholds, one for sentimental score of a sentence and the other for sentimental score of a word. We show the experiment results in Table 2. The symbol *v* indicates the threshold of sentimental score for a sentence and *w* indicates the threshold of sentimental score for a word. According to our experiments, only 61 objective words are modified when *v* is 0.5 and *w* is 0.7. Thus, our proposed approach may not work well when these thresholds are too great. Table 2 shows that our proposed approach outperforms the traditional approach for various thresholds and concludes that this approach is able to make an improvement for sentiment classification.

TABLE II
The Experiment Results of Traditional and Proposed Approaches Based on Different Thresholds

acc (v / w)	0.5		0.6	
	Traditional Approach	Proposed Approach	Traditional Approach	Proposed Approach
0.5	74.05%	76.90%	74.05%	75.60%

0.6	74.10%	78.65%	74.05%	74.75%
0.7	74.05%	77.20%	74.05%	76.10%

V. CONCLUSIONS

Word of mouth (WOM) has become the main information resource while making business or buying strategies. The technique of WOM sentiment classification is a way to help users to manage and analyze WOM efficiently. SentiWordNet provides each synonymous set of WordNet three sentiment scores regarding positivity, negativity, and objectivity respectively, and has become a public and popular sentiment lexicon for sentiment classification research. This research proposes a novel sentimental relevance approach to re-evaluate objective sentiment words in this electronic sentiment lexicon. According to our experiments, our proposed approach is able to improve the performance of sentiment classification.

REFERENCES

[1] J. Arndt, "Role of product-related conversations in the diffusion of a new product. *Journal of Marketing Research*," vol. 4, no. 3, 1967, pp. 291-295.

[2] D. Godes, and D. Mayzlin, "Using online conversations to study word-of-mouth communication," *Marketing Science*, vol. 23, no. 4, 2004, pp. 545-560.

[3] M. Abulaish, Jarhiruddin, M.N. Doja, and T. Ahmad, "Feature and opinion mining for customer review summarization," *Lecture Notes in Computer Science*, vol. 5909, 2009, pp. 219-224.

[4] A. Esuli, and F. Sebastiani, "SentiWordNet: a publicly available lexical resource for opinion mining," in *Proceedings of LREC*, 2006, pp. 417-422.

[5] G.A. Miller, "WordNet: a dictionary browser," in *Proceedings of the First International Conference on Information in Data*, 1985, pp. 25-28.

[6] A. Devitt, and K. Ahmad, "Sentiment polarity identification in financial news: a cohesion-based approach," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2007, pp. 984-991.

[7] K. Denecke, "Using SentiWordNet for multilingual sentiment analysis," in *Proceedings of the IEEE 24th International Conference on Data Engineering Workshop (ICDEW)*, 2008, pp. 507-512.

[8] K. Denecke, "Are SentiWordNet scores suited for multi-domain sentiment classification?" in *Proceedings of the 4th International Conference on Digital Information Management(ICDIM)*, 2009

[9] S. Agrawal, and T. Siddiqui, "Using syntactic and contextual information for sentiment polarity analysis," in *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, 2009, pp. 620-623.

[10] M. Koppel, and J. Schler, "The importance of neutral examples for learning sentiment," *Computational Intelligence*, vol. 22, no. 2, 2006, pp. 100-109.

[11] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.

Privacy-Preserving Profiling

Thomas Barnard, Adam Prügel-Bennett
Information: Signals, Images, Systems (ISIS),
School of Electronics and Computer Science,
University of Southampton, United Kingdom

Abstract— *With the rise of social networking, and other sites which collect vast amounts of user data, the issue of user privacy has never been more important. When creating user profiles care must be taken to avoid collecting sensitive information, while ensuring that these profiles are fit for purpose. In this paper we present a specific instance of the privacy-preserving profiling problem in an expert-finding application. We present a dataset of profiles, as well as several datasets for contaminating these profiles, and provide experiments to test data quality and privacy-preserving performance. We present a simple solution based on training an LSA model on a clean profile corpus, which maintains performance and provides a moderate level of privacy.*

Keywords: User Profiling, Information Retrieval, Privacy

1. Introduction

People spend an increasing amount of their time using social networking sites. In building and maintaining social networking profiles, users provide large amounts of information to these sites. Of course these users expect something in return; providing this information may help to find new friends or business contacts, and strengthen existing relationships, while the social networking provider gains access to profiles which it can use to provide personalized advertisements.

There have been a number of cases of privacy being compromised or potentially compromised by user profiling. Facebook have been criticized for their use of profiling in providing personalized adverts, which may allow advertisers and others to discover the sexual orientation of users[1]. Privacy concerns also led to the second Netflix recommendation prize being cancelled, and the dataset for the first prize being made unavailable for download[2].

While the user shares information about their interests and contacts, they may unwittingly disclose private information about themselves. Relying on a user to ensure their own privacy is an unacceptable solution, both because it places an additional burden on the user, and because the user may not be the best judge of what information about themselves should be made available. They may also be broadcasting their details more widely than they realise; privacy settings may be set incorrectly, and third-party applications may collect data from profiles without users' knowledge or consent.

In this paper we will introduce the problem of privacy-preserving profiling. We will look at the specific problem of generating profiles within the *Instant Knowledge* project. We will describe a series of experiments to determine the preservation of privacy, and use these experiments to evaluate our early attempts to solve this problem.

2. Instant Knowledge

The *Instant Knowledge* (IK) project aims to provide as solution to the problem of finding experts within an organization. It can be difficult to keep track of expertise within an organization, which can limit collaboration, or make it difficult to find the appropriate people to work on a new project. In academia researchers often find out too late that somebody was working on a similar problem in the same department, with each unaware of the other's work.

The IK system is a keyword-based information system utilizing a client-server architecture. Users' personal devices collect context information, and generate queries based on user activity. Keywords relating to an area of expertise are sent to the server which returns a ranked list of experts. In this paper we will focus on the generation of profiles, and ignore more complex aspects of the system such as context awareness, distributed algorithms, and query augmentation.

The IK system requires accurate, up-to-date profiles of expert interests in order to provide the best responses to user queries. The simplest method of generating these profiles would be for the experts to enter free-text describing their professional interests. This may, however, lead to profiles which are poorly maintained as the user loses interest in the task.

The next step would be for users to manually provide documents which they feel represent their interests, for example technical reports or academic publications. This approach is not without its problems, as it still requires user effort. Even if documents are added automatically, for example if they are added to a publication repository, if these documents are added infrequently, they might not fully represent a user's interests. Certain approaches to a problem may not lead to a publication but may nonetheless help enrich a user's profile.

Instead we favour a fully automatic approach, building a profile from all the documents authored or collected by a user, as well as other sources of information, such as email, web browsing activity, and social networking. By including these

additional sources of information we hope to build profiles which are more accurate and up-to-date than those produced manually.

This approach does, however, present some challenges; some of the information collected will be irrelevant or private. In the case of irrelevant data, recommendation performance may be reduced, in the case of private information disclosure may have serious negative consequences.

3. Privacy

Profiles within the IK system are assumed to be private in the sense that their exact contents is only known to the user they belong to and the system itself. In this paper we will assume that there are no third parties who can peek at the profile, or observe it in transit from the expert to the IK server. The user profile is however assumed to be accessible, either publicly or within an organization, through the profile recommendation system.

The main attack vector we consider is profile reconstruction through repeated queries. By making a series of carefully constructed queries it may be possible to infer the presence and weights of certain terms and concepts within a profile, by observing how highly a given user is ranked for these queries. The construction of such an attack will not be addressed in this paper.

While a notion of privacy in data mining and user profiling can have a number of different interpretations, from anonymity to an uncertainty in the particular values of an attribute, we consider profiles to be made up of public and private information, and it is our job to remove the private information while leaving the public information intact. This is in contrast to some applications where the whole profile is assumed to be private; the need to recommend specific, named users is incompatible with absolute privacy.

We aim to conceal two main types of private information within a profile: passwords, bank account details, usernames, and other private tokens; interests which would be embarrassing, controversial, or would cause some harm to the user should they be disclosed. We are also interested in removing irrelevant information from a profile, for example non-professional interests such as musical tastes, or hobbies.

4. Privacy-Preserving Profiling

Our goals in automatic privacy-preserving profiling are the production of an useful user profile, and the preservation of user privacy. These goals are to some extent at opposition with each other: as we remove private information we will remove useful profile which will reduce performance; as more information is retained in a profile the greater the risk of disclosing sensitive information will be.

Our task is made harder as our privacy-preserving techniques must operate without user input. It would be much simpler to train a classifier to identify public and private documents by using user labelled documents, building a model for each user. We could consider building a global model using a profile corpus and examples of private information.

The problem here is that what each user considers private may vary considerably. It could be argued that there are subjects that most users would consider private for example sexual preferences and habits, political affiliation, or health concerns. For some users, however, these controversial topics may be their main area of expertise, so we cannot filter them outright.

Determining the nature of information without help from the owner of that information requires us to rely on patterns in the data itself, and the overall properties of public and private data in general. It is difficult and may be impossible to build a privacy-preserving profile by analysing an expert's documents *in vacuo*.

5. Methodology

As our focus is on the automatic production of profiles and their privacy preserving attributes we have implemented a very simple information retrieval system.

The documents belonging to a user are converted into a bag-of-words representation, removing structure, turning them into an unordered collection of words. Commonly words with little discriminative power, called stop words, are removed. We use the list provided by Fox in [3]. Finally words are reduced to their root form using a stemming algorithm, for example 'computer' and 'computation' may be reduced to the stem 'comput'. Finally these processed words are counted to produce a term frequency representation of the original document. While this processing removes some information from the documents and may result in reduced performance, it should also remove private information.

We could produce profiles by adding together term frequency representations of their constituent documents, however this could lead to larger documents dominating the profile. Instead we normalize these document representations by their length before adding them together,

$$\mathrm{TW}_{p,i} = \sum_{j \in D_p} \frac{\mathrm{TF}_{j,i}}{N_j}, \tag{1}$$

where $\mathrm{TW}_{p,i}$ is the weight of term i in profile p, D_p is the set of documents that profile p contains, $\mathrm{TF}_{j,i}$ is the frequency of term i in document j, and N_j is the size of document j.

We then use a vector-space model (VSM)[4], treating each profile as a multidimensional vector, where each dimension corresponds to the weight of a particular term in the profile. We apply a weighting scheme to the raw frequency based weights called TF-IDF, here the term frequency weight is normalized by the profile length, and multiplied by the inverse document frequency (IDF), giving a higher weighting to terms which occur in fewer documents. The TF-IDF

weighting equations are given below,

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}},$$
$$IDF_i = log\frac{|D|}{|\{d : t_i \in d\}|},$$
$$TFIDF_{i,j} = TF_{i,j} \times IDF_i,$$

where $tf_{i,j}$ is the term frequency of term i in document j, $n_{i,j}$ is the number of times term i occurs in document j, idf_i is the inverse document frequency of term i, D is the collection of documents, and $tfidf_{i,j}$ is the TF-IDF weighting of term i in document j.

The process described so far will produce vectors which may have many thousands of dimensions. Differences in the terms used means that documents which concern similar topics may have few terms in common. In addition high-dimensional vectors require more resources to manipulate and compare. To solve these problems we apply a dimensionality reduction technique to our profile vectors.

Latent Semantic Analysis (LSA), or Latent Semantic Indexing (LSI) is a technique for taking document vectors and projecting them into a lower dimensional space[5]. As well as reducing the dimensions, LSA has the advantage of projecting the term vectors into a concept space, where concepts are represented rather than specific terms. This means that terms with similar meanings are close in this space, where in term space there would be no match.

LSA is implemented using a singular value decomposition (SVD) of the profile matrix. The details of this process are beyond the scope of this paper, but essentially the matrix is factorized into a form capturing the directions of maximal variance in the data,

$$A = USV^T, \tag{2}$$

where U and V are matrices corresponding to rows (terms) and columns (profiles) in the matrix respectively, and S contains the singular values. By retaining only the top singular values it is possible to reduce the dimensionality of the matrix. This also has the effect of removing noise in the matrix at the expense of fine detail.

To compare profiles and queries we must first project them into concept space,

$$\hat{D} = S^{-1}U^T D, \tag{3}$$

where D is the document, and $\hat{D}$ is its concept-space representation. We compare vectors by using the cosine similarity which gives a value between 0 and 1 indicating the degree to which two vectors point in the same direction,

$$similarity(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{||\mathbf{A}||||\mathbf{B}||}. \tag{4}$$

To recommend a profile given a query, we simply order profiles in descending order of their similarity to the query vector.

While these simple techniques may lead to useful information being removed from profiles, such as the context of words in a bag-of-words model, or the same word being used with a different meaning being ignored by an LSA model, their simplicity makes analysis of the privacy-preserving aspects easier. Removing structural information from documents should also lead to an increase in privacy.

5.1 Privacy Preservation

The projection of data onto a lower dimensional concept space provides some blurring of information, and innocuous documents and terms will share some similarity with private documents and terms. This provides some measure of plausible deniability, at the expense of loss of fine detail. Some evidence of private terms may remain.

We propose a method of privacy-preserving profiling using a technique we have already described in this paper, *Latent Semantic Analysis*. LSA works by finding a concept space representing a collection of documents, whose dimensions represent the directions of greatest variability in the collection of documents.

Making the assumption that public information differs significantly from private information, and using the fact that the LSA projection depends on the corpus of documents that was used to create it, we propose the simple technique of using a corpus of public information to build a projection, and then projecting all information into this concept space.

Our hypothesis is that as the public concept space has been learned using public documents it will be less well suited to representing private information. In this way private information will be "projected out" of the profiles.

6. Related Work

While there has been quite a lot of research into privacy in data mining in profiling generally, there has been surprisingly little research into the problems described earlier in the paper. That is profiles are usually treated as objects which are either wholly private or public.

Reichling et al.[6] presented a similar approach to user profiling for the purpose of finding experts, using an LSA model to represent profiles. In their approach privacy is dealt with manually: the user is responsible for selecting directories which the system is allowed to search for documents.

Privacy preserving data mining (PPDM) is a growing area of research which aims to ensure that data mining activities can be conducted while safeguarding user privacy[7]. While there are some overlaps with what we are doing, most research in PPDM seems to deal with anonymity[8], hiding precise values of data[9], and cryptographic methods.

While the problem may at first appear to be superficially similar to the problem of spam filtering, except the aim is to prevent information leaking out rather than being received, there are some important differences. Firstly with spam filtering it is possible to maintain a global model of

spam which can be used to filter incoming messages for every user, this may then be tweaked by user feedback (e.g. identifying misclassified messages), but large changes to the global model seem unlikely. Secondly, instead of filtering documents out completely, we may have documents which contain a mixture of private and public information and it would be ideal to have this public information added to the profile.

7. Experiments

Bertino et al.[10] describe five criteria with which to evaluate PPDM algorithms:

- Efficiency
- Scalability
- Data Quality
- Hiding Failure
- Privacy Level

Of these criteria the most applicable to our problem are data quality and hiding failure.

Data quality describes the effect that the privacy preserving process has on the original data. They suggest that this can be tested by the change in data mining performance on the when using the processed data versus the original dataset. Hiding failure relates to the amount of private data that can be recovered from the sanitized data.

In the following sections we will describe the experiments we performed to test our techniques given these criteria.

7.1 Datasets

In order to test our hypothesis and carry out experiments in user profiling we require both a source of user profiles, and of private information with which to "poison" them. It would be difficult and time consuming to obtain samples of real user profile data, as well as real private information, so instead we have created profiles from academic publications data and obtained surrogate private information from a different source.

The RKBExplorer website[1] which is part of the ReSIST project at the University of Southampton provides a semantic web database containing information from a number of institutions where authors of academic papers have self-archived their publications in ePrints repositories. This dataset has information on authors and their publications, including titles and abstracts, but unfortunately not full document texts. We have sampled this database to create a dataset with around 750 profiles and a total of around 14,000 documents. We believe this is a good representation of a set of expert profiles.

We decided to create a dataset of "poison" documents from another source; a collection of text files obtained from BBS (Bulletin Board Systems), grouped broadly by topic. Amongst these groups were collections of files categorized as "Anarchy" and "Drugs". We processed these documents in the same way as our profile data to create datasets with around 1500 and 500 documents respectively.

[1] www.RKBExplorer.com

7.2 Data Quality

For each experiment we first split our collection of academic publications randomly in two, holding back half the data for the creation of a corpus and using the rest of the data for training and testing.

We performed two experiments, the first was to determine the appropriate number of dimensions to retain in our LSA model. For this experiment we compared the performance of the corpus derived LSA model, with one built using the documents themselves, and another model built using the documents filtered to remove terms which are not present in the corpus. For the corpus derived model we looked at a model built from individual documents, as well as one built from profiles in this withheld data. At this stage no poison is added to the documents.

We used ten fold cross-validation, using the withheld documents as queries. Relevance is binary (i.e. a document is relevant or not) and will be determined by authorship of each query. This leads to very low scores, as many documents only have a single author, and if this author is not at the top of recommendation list then performance will be less than perfect. Additionally some experts who are not authors of the query document may nonetheless be relevant to it.

We use *Mean Absolute Precision* (MAP) to measure performance, which is the *Average Precision* averaged over all queries. The *Average Precision* is simply the precision of the top-r results of a query averaged over each relevant result at rank r. The equations are given below,

$$P(r) = \frac{|\{\text{relevant retrieved documents} <= \text{rank r}\}|}{r}, \quad (5)$$

$$AP = \frac{\sum_{r=1}^{N}(P(r)relevant(r))}{|R|}, \quad (6)$$

$$MAP = \frac{\sum_{q=1}^{Q} AP(q)}{|Q|}, \quad (7)$$

where R is the set of relevant documents, r is the rank, N is the number of relevant documents retrieved, and Q is the set of queries.

The results for our first experiment are shown in Figure 1. From these results we decided to use a model rank of 500 for good performance, but note that most of the performance is retained down to a model rank of around 100. Additionally we note that a corpus derived model LSA built on profiles performs better than one built on individual documents.

The second experiment involved testing the effect of profile poisoning on performance. For these experiments we used a model size of 100 and 500. Increasing amounts of poison was added to the documents. A poison level of 1 meaning that the number of poison documents added to a profile was equal to the number of documents already in the profile.

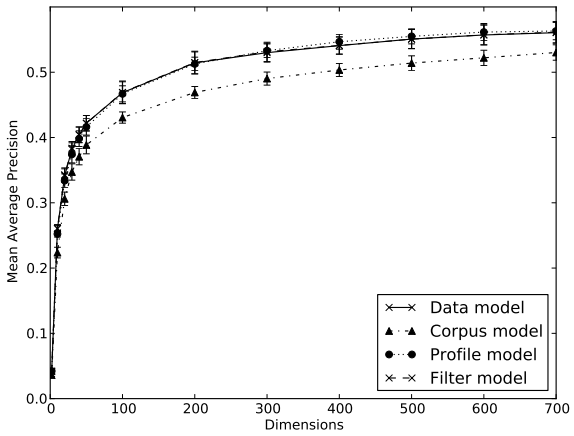


Fig. 1: Experiment one.

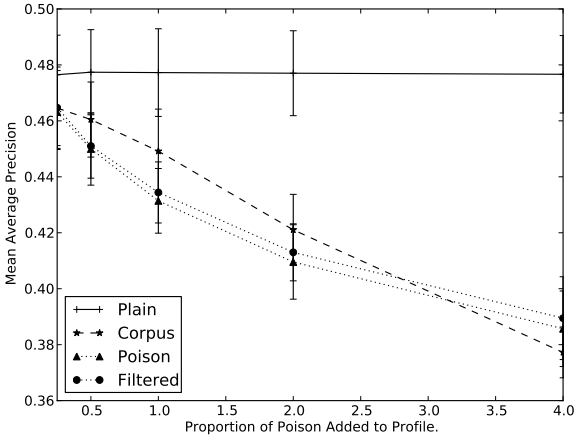


Fig. 2: Data Quality Experiment k = 100, Anarchy

7.3 Hiding Failure

Our privacy experiment is based on possible mining attacks that could be used to extract information about experts from the system. As attackers will not have direct access to profile vectors it does not seem sensible to look at the change in profile vectors with and without private projection, but instead to look at what information can be obtained through the query interface.

The scenario we consider is an attacker trying to find experts with interest in topics which may be controversial, embarrassing, or incriminating. We add poison to a certain proportion of profiles and attempt to detect these profiles by using a different set of poison documents as queries. In this experiment we add four times as many poison documents to each selected profile as the public documents that profile contains.

For these experiments we follow a similar approach to the performance experiments, except in this case success will be judged by how poorly the system performed in the experiment. Relevant profiles are all of the profiles which have had poison added to them, regardless of the specific documents used.

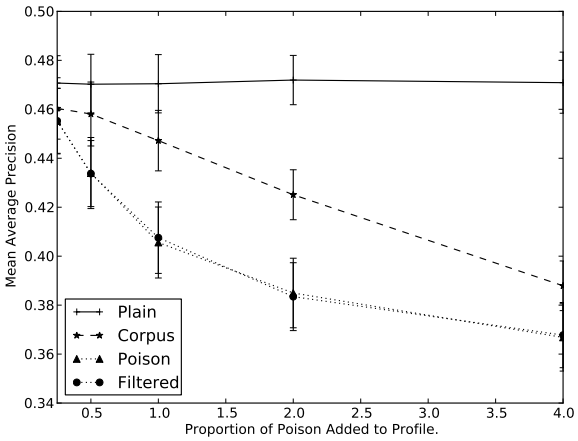


Fig. 3: Data Quality Experiment k = 100, Drugs

7.4 Results

Figure 2 shows the results for the data quality experiment using the anarchy dataset with a model size of 100, and Figure 3 shows the results of the same experiment with the drugs dataset. Figure 4 shows the results for the data quality experiment using the anarchy dataset with a model size of 100, and Figure 5 shows the results of the same experiment with the drugs dataset.

The results of the data quality experiment are roughly the same for both datasets. The quality of results degrades much more slowly when the higher rank model is used, and the corpus derived model performs the best on these tasks. It is interesting that simply removing words which are not in the corpus does not help maintain performance levels. This is probably because many of the important terms in the poison documents are present in the corpus.

Figure 6 shows the results for the hiding failure experiment using the anarchy dataset with a model size of 100, and Figure 7 shows the results of the same experiment with the drugs dataset. Figure 8 shows the results for the hiding failure experiment using the anarchy dataset with a model size of 100, and Figure 9 shows the results of the same experiment with the drugs dataset.

In each case the corpus derived model performs better than the poisoned and filtered models, which reach a MAP of almost 1 at certain points. While the corpus model does provide some level of privacy protection, it is slight, and much worse than the untainted profiles tested against the same queries. A higher level of privacy is provided using

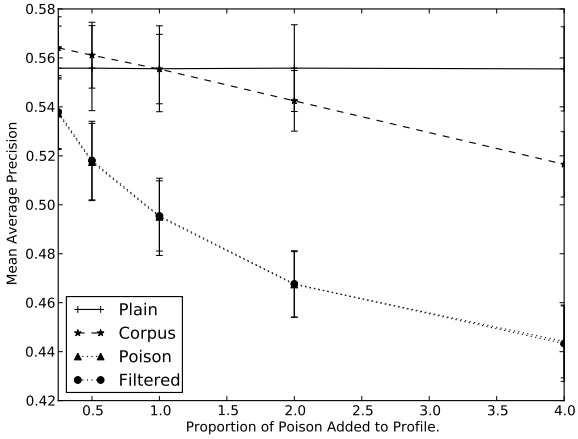


Fig. 4: Data Quality Experiment k = 500, Anarchy

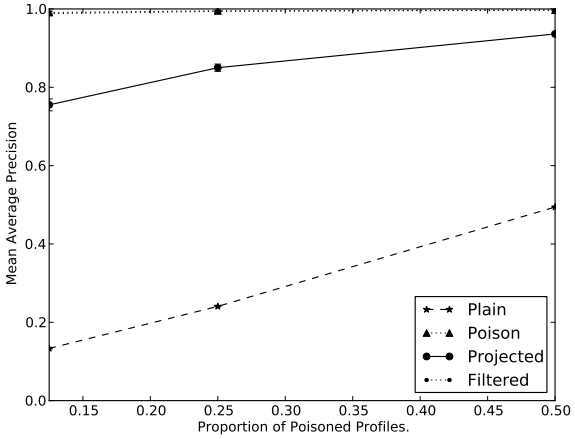


Fig. 6: Hiding Failure Experiment k = 100, Anarchy

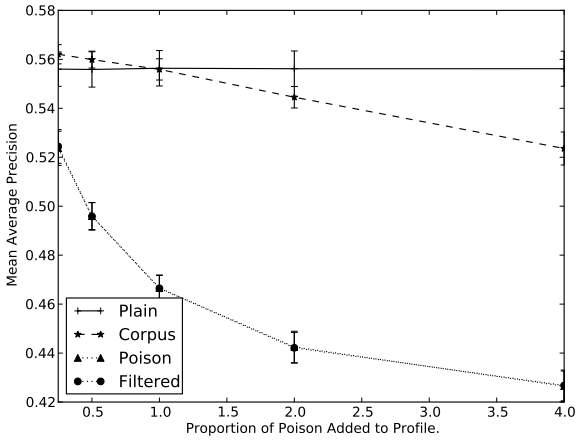


Fig. 5: Data Quality Experiment k = 500, Drugs

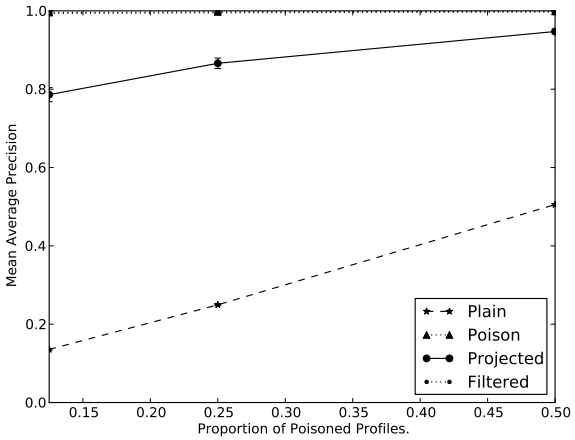


Fig. 7: Hiding Failure Experiment k = 100, Drugs

a lower-dimensional model.

8. Conclusion

In this paper we presented a specific instance of a privacy-preserving profiling problem relating to the Instant Knowledge expert recommendation system. Our main goals are the automatic generation of expert profiles, while preserving user privacy with little or no user feedback.

We presented a set of datasets and experiments which can be used to evaluate performance on this task. While our simple initial solution to the problem failed to hide private data adequately it significantly reduced the degradation of performance caused by polluting a profile with poison data.

We believe that the model failed to preserve privacy adequately as the LSA model was sufficient to represent most of the public and private information. The private information may be closer to public information than we had anticipated. While performance can be improved by reducing the rank of the profile matrix approximation, this affects the performance of the model on the data quality tasks.

8.1 Future Work

The simple privacy-preserving method we applied in this paper was largely passive. The intention was to create a model which was incapable of adequately representing the private information, which would lead to such data being filtered or reduced in magnitude.

Active filtering is more difficult without user feedback to guide the classification of documents or terms in a profile. We could, however, make better use of the profile corpus to train a filtering model. While private information may be different for each user, we should be able to make an educated guess about what makes a coherent profile.

For example we might not expect papers on sexually-transmitted infections to be present in the profile of a

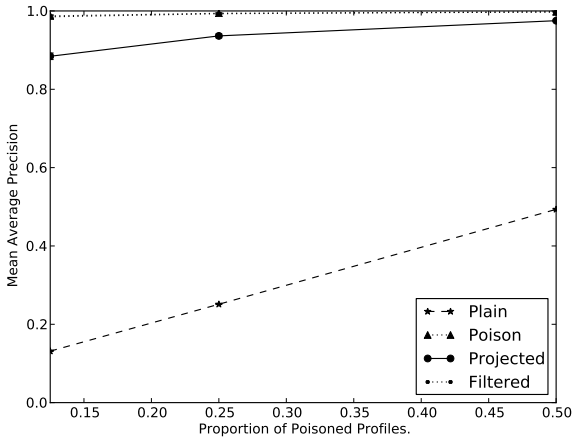


Fig. 8: Hiding Failure Experiment k = 500, Anarchy

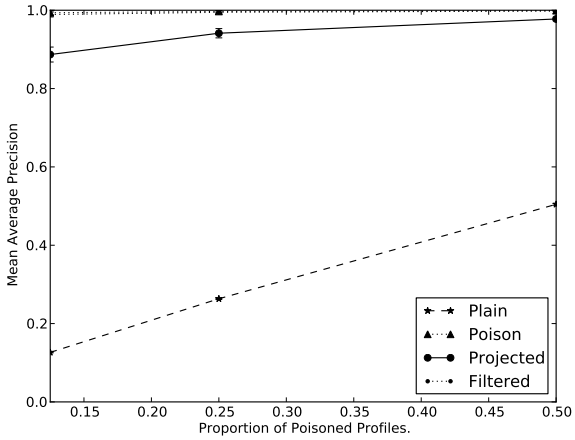


Fig. 9: Hiding Failure Experiment k = 500, Drugs

computer science researcher. While this researcher may have coauthored a paper on STIs, it is unlikely. Using the corpus we could calculate the probability of different interests co-existing in the same profile and use this information to filter out dubious interests.

The assumption of zero user input is perhaps too strong, and a wider range of techniques could be applied even if we have only a small number of labelled documents. We would also like to look at the issue of updating profiles with new documents, and how an existing profile can be used to preserve privacy.

Acknowledgment

The work reported in this paper has formed part of the Instant Knowledge Research Programme of Mobile VCE, (the Virtual Centre of Excellence in Mobile & Personal Communications), www.mobilevce.com. The programme is co-funded by the UK Technology Strategy Boards Collaborative Research and Development programme. Detailed technical reports on this research are available to all Industrial Members of Mobile VCE.

References

[1] S. Guha, B. Cheng, and P. Francis, "Challenges in measuring online advertising systems," in *Proceedings of the 10th annual conference on Internet measurement*, ser. IMC '10. New York, NY, USA: ACM, 2010, pp. 81–87. [Online]. Available: http://doi.acm.org/10.1145/1879141.1879152

[2] R. Singel, "Netflix cancels recommendation contest after privacy lawsuit," http://www.wired.com/threatlevel/2010/03/netflix-cancels-contest/, March 2010, retrieved on Wednesday 16th February 2011.

[3] C. Fox, "A stop list for general text," *SIGIR Forum*, vol. 24, pp. 19–21, September 1989. [Online]. Available: http://doi.acm.org/10.1145/378881.378888

[4] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, pp. 613–620, November 1975. [Online]. Available: http://doi.acm.org/10.1145/361219.361220

[5] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, vol. 41, no. 6, pp. 391–407, 1990.

[6] T. Reichling and V. Wulf, "Expert recommender systems in practice: evaluating semi-automatic profile generation," in *Proceedings of the 27th international conference on Human factors in computing systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 59–68. [Online]. Available: http://doi.acm.org/10.1145/1518701.1518712

[7] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis, "State-of-the-art in privacy preserving data mining," *SIGMOD Rec.*, vol. 33, pp. 50–57, March 2004. [Online]. Available: http://doi.acm.org/10.1145/974121.974131

[8] L. Sweeney, "k-anonymity: a model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, pp. 557–570, October 2002. [Online]. Available: http://portal.acm.org/citation.cfm?id=774544.774552

[9] D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, ser. PODS '01. New York, NY, USA: ACM, 2001, pp. 247–255. [Online]. Available: http://doi.acm.org/10.1145/375551.375602

[10] E. Bertino, I. N. Fovino, and L. P. Provenza, "A framework for evaluating privacy preserving data mining algorithms*," *Data Min. Knowl. Discov.*, vol. 11, pp. 121–154, September 2005. [Online]. Available: http://portal.acm.org/citation.cfm?id=1095655.1095681

SESSION

FEATURE SELECTION + CLUSTERING METHODS + TESTING APPLICATIONS

Chair(s)

Robert Stahlbock

Perspective of Feature Selection Techniques in Bioinformatics

Satish Kumar David[1], Mohammad Khalid Siddiqui[2]
[1]IT Department, [2]Department of Basic Sciences
[1,2]Strategic Center for Diabetes Research, King Saud University, Riyadh, Saudi Arabia

Abstract

The availability of massive amounts of experimental data based on genome-wide association and mass spectroscopy studies have given motivation in recent years to a large effort in developing mathematical, statistical and computational techniques to infer biological models from data. In many bioinformatics problems the number of features is significantly larger than the number of samples (high feature to sample ratio data sets) and feature selection techniques have become an apparent need in many bioinformatics applications. In addition to the large pool of techniques that have already been developed in the data mining fields, specific applications in bioinformatics have led to a wealth of newly proposed techniques. This assessment provides the aware of the possibilities of feature selection, providing a basic taxonomy of feature selection techniques, discussing their use, variety and potential in a number of both common as well as upcoming bioinformatics applications.

Keywords

Bioinformatics; Feature Selection; Text Mining; Wrapper; Genotype analysis.

Introduction

Now a day's interest for using Feature Selection (FS) techniques in bioinformatics becoming compulsion for model building from being just example. The modeling tasks going to spectral analysis and text mining from sequence analysis over microarray analysis in bioinformatics. FS helps to acquire better understanding about the data's important features and their relationship type and can be applied to supervised (classification, prediction) and unsupervised (Clustering) learning [1]. The original representation of the variables does not vary in FS techniques but dimensionality reduction techniques such as projection and compression can vary the original representation of the variables. From an informatics perspective, the process of selecting differentially expressed genes is readily achieved via data-mining techniques known as Feature Selection. It is an important step in the data-mining process aims to find representative optimal feature subsets that meet desired criteria. The key consideration in this review is FS techniques application and the idea is to bring awareness of the requirements and benefits of using FS techniques. This article also will give an idea about few useful data mining and bioinformatics software packages used for FS.

In microarray data analysis, one criterion for a desired feature subset would be a set of genes whose expression patterns vary significantly when compared across different sample groups. The resulting subset can then be used to further analysis such as building a diagnostic classifier. Problem of selecting some subset of a learning algorithm's input variables upon which it should focus attention, while ignoring the rest (Dimensionality Reduction). Several pattern recognition techniques alone do not handle with large amounts of irrelevant features. Pattern recognition techniques and FS techniques jointly work effectively in many applications [2].A large number of features enhances the model's flexibility, but makes it prone to over fitting. The FS objectives are

(a) To increase the speed of learning algorithm's
(b) To improve the accuracy of classifier on new data
(c) To remove redundant features from dataset.

In classification context approaches for FS techniques tasks are: filters, wrappers and embedded methods [3].

Filter Approach

FS is based on an evaluation criterion for quantifying how well feature (subsets) discriminate the two classes in Filter techniques. Filters assess the relevance of features. Relevance score calculated and low scored are removed and then this subset is input to classification algorithm. Only once FS needs to be performed and then different classifiers evaluated [4]. Improved scalability, simple and fast is the advantages of filter techniques. Disadvantages of filter techniques are classifiers performance may be non-optimal features [5]. To prevail over the problem of overlooking feature dependencies, numbers of multivariate filter techniques were introduced.

Wrapper Approach

Wrapper techniques are iterative approach, many feature subsets are scored based on classification performance. Running a model on the subset wrappers use a search algorithm to search through the space of possible features and evaluate each subset. Wrappers have higher over fitting risk and can be computationally expensive. These search methods assess subsets of variables according to their usefulness to a given classifier [6]. Based on search method the wrapper methods divided into two kinds a)randomize [7,8] b)Greedy [9]. Advantages of Wrapper techniques are improving the performance of given

classifier. Disadvantages of Wrapper techniques are computationally intensive, high cost and poor scalability.

Embedded Approach

Embedded techniques are specific to a model. These methods use all the variables to generate and analyze the model to recognize the importance of the variables [10]. FS is part of classifier's training procedure (e.g. decision trees). Consequently, they directly link variable importance to the learner used to model the relationship. Attempt to jointly or simultaneously train both a classifier and a feature subset. Often optimize an objective function that jointly rewards accuracy of classification and penalizes use of more features. Advantages of Embedded technique are less computationally intensive. Disadvantage of embedded technique is classifier dependent classifier.

Literature Mining

Automated methods for knowledge retrieval from the text are known as literature mining. Most knowledge is stored in terms of texts, both in industry and in academia. In biology promising area for data mining is literature mining [11]. Word based system Bag-of-Words (BOW) representation is changing set of words linearly structured into unstructured which may lead to very high dimensional datasets and the need for feature selection techniques [12]. BOW based models use statistical weights based on term frequency, document frequency, passage frequency, and term density. BOW disregards grammatical structure, layout free representation and context dependent. Literature mining developed for document clustering, classification and researcher's practical use.

Sequence Analysis

Sequence analysis is the modern operation in computational biology. This operation find out which part of the biological sequences is alike and which part differs during medical analysis and genome mapping processes. The sequence analysis implies subjecting a DNA to sequence databases, sequence alignment, repeated sequence searches, or other methods in bioinformatics [13]. New sequencing methodologies, fully automated instrumentation, and improvements in sequencing-related computational resources greatly contributed for genome-size sequencing projects. Multistage process contains the purpose of sequence (protein), its fragmentation, analysis and resulting sequence information. This information reveals similarities of homologous genes and its regulation and function of the gene, leads to a better understanding of disease states related to gene variation [14].

Microarray Analysis

Human genome contains approximately 30,000 genes [15]. Each of our cells has some combination of these genes active at any given instant and others inactive. Computation in the microarray data is great challenge because of large dimensionality and small sample size. Multivariate is unsupervised Clustering, Principle component analysis, Classification (statistical learning, discriminant analysis, supervised clustering). According to Jafari considerable and valuable effort has been done to contribute adapt FS, since microarray claims to be infancy [15]. Univariate features ranking techniques has been developed such as parametric and non-parametric (model free). Parametric method assumes given distribution from which samples have been generated. Two samples t-test and ANOVA are mostly used in microarray analysis even though usage not advisable [16]. ANOVA is for measuring the statistical significance of set of independent variables. ANOVA produces the p-value for the features set. ANOVA procedure recommended only for balanced data. Other types of parametric techniques such as regression modelling, Gamma distribution model. Since uncertainty is high in parametric techniques, the model free (non-parametric) techniques proposed. Metrics are from statistical categories(BSS/WSS) [17]. Using random permutation reference distribution of statistics were estimated in model free techniques. Multivariate regressions are Correlation features selection (CFS), minimum redundancy maximum relevance(MRMR). Proposed the use of methods under ROC curve or optimization of LASSO model. ROC gives interesting evaluation measure. Three broad problems in microarray analysis: a) class discovery (unsupervised classification), b) class comparison (differential gene expression), c) class prediction (supervised classification).

Genotype analysis

In the genome wide association study (GWAS) a large number of data have been generated for SNP analysis, its range from 100 to 1000 SNP. These SNP analysis is import to look the relation between phonotypic with genotypic data to relate the different disease condition. Different approaches were used based on data mining and genetic algorithm [18]. A weighted decision tree, a correlation-based heuristic are used for selecting significant genes. The goal of feature selection for SNPs can be achieved with supervised and unsupervised methods such as clustering, neighborhood analysis, applying classification algorithm and eliminating the lowest weight features can pruned DNA gene expression data sets by eliminating insignificant features [19]. The significant gene/SNP set in cross-validation accuracy was increased by 10% over the baseline measurements and the specificity increased by 3.2% over baseline measurements. Block free approach for tagging SNPs. the selection of tagging SNPs can be partitioned into the three following steps:

a. Determining neighborhoods of linkage disequilibrium: Find out which sets of SNPs can be meaningfully used to infer each other.

b. Tagging quality assessment: Define a quality measure that describes how well a set of tagging SNPs captures the variance observed.

c. Optimization: Minimize the number of tagging SNPs.

The disadvantage of block free approach is not always straightforward definition of blocks and no consensus on how blocks must be formed. It is based only on the local correlations. To avoid computational complexity, did not look for subsets of SNPs but discard redundant markers using FS technique. It can give better performance on

large data set using exhaustive search to short chromosomal regions but this does not guarantee optimal solutions. In the genotyping the huge data generated and related between the SNP and LD (Linkage disequilibrium) was used by the block based approach [20].

Mass Spectroscopy Methodology

Mass Spectroscopy analysis is for protein-based biomarker profiling and disease diagnosis. Two different types of mass spectroscopy methodology used for analysis. The common method of sorting ions is the Time-Of-Flight (TOF) analyzer. In TOF analyzer ions are collected to an ion trap, and then accelerated with one push into an empty chamber with an electrical field in it. An instrument MALDI-TOF (Matrix-Assisted Laser Desorption and Ionization Time-Of-Flight) low resolution can contain upto 15,500 data points in spectrum and number of points can even grows for higher resolution instruments. MALDI-TOF is the most popular techniques presently employed for detecting quantitative or qualitative changes of proteins [21]. Mass spectrometry measures two properties of ion mixtures in the gas phase under a vacuum environment: the mass/charge ratio (m/z) of ionized proteins in the mixture and the number of ions present at different m/z values. Thus the mass spectrometry for a sample is a function of the molecules and used to test for presence or absence of one or more. The general pipeline is show in Figure 1, which includes three steps.

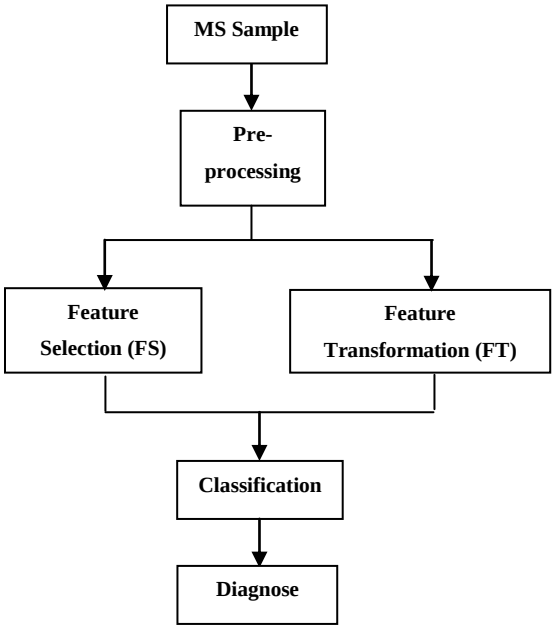


Figure 1. Pipeline of pattern analysis of MS data

Firstly, the MS data is pre-processed. Then, two kinds of dimension reduction methods are accepted. One is called feature transformation (FT). FT methods construct new features as functions that express relationships between the initial features. The other kind of methods is called feature selection (FS). The FS methods output a subset of the original input features without transforming them, such as t-test, sequential forward selection (SFS), boosting approaches, etc. The last step is classification, which gives the results of diagnosis, such as SVM, KNN, decision tree, etc.

As Somorjai et al. explained the data analysis steps is constrained by both high dimension input spaces and their inherent sparseness [20]. Several studies employ the simplest approach of considering every measured value as predictive features, so applying FS technique over 15000 variables upto around 100000 variables [22].

Ensemble feature selection

An ensemble system is composed of a set of multiple classifiers and performs classification be selecting from the predictions made by each of the classifiers. Ensemble FS derived from decision tree and used to assess relevance of each features. Since wide research has shown that ensemble systems are often more accurate than any of the individual classifiers of the system alone and it is only natural that ensemble systems and feature selection would be combined at some point. Composed of set of multiple classifiers and performs classification be selecting from predictions made by each of the classifiers. Frequently a single FS technique is not optimal and redundant subset of feature data [23]. Therefore, Ensemble FS have been incorporated to improve the methods strength and methods stability [24]. Additional computational resources are required to use ensemble FS and if additional resources are affordable, ensemble FS offer framework to deal with small sample.

Conclusion and Future perspective

In this review we assess feature selection techniques in bioinformatics applications. Table1 shows software's packages, their main reference and website shown. These software packages are free for academic use. We found issues and problems of small sample size and large dimensionality in data mining. Feature Selection techniques designed to deal with these problems. Productive effort has been performed in the proposal of univariate filter FS techniques. Future research is the development of ensemble Feature Selection approaches to enhance the robustness of selected feature subset and literature mining. Interesting opportunities towards genotype analysis is needed.

References

[1] Varshavsky,R., et al. (2006) Novel unsupervised feature filtering of biological data. Bioinformatics,22,e507–e513.

[2] Guyon,I. and Elisseeff,A. (2003) An introduction to variable and feature selection. J. Mach Learn Res., 3, 1157–1182.

[3] Y. Saeys et al. (2007) A review of feature selection techniques in bioinformatics. Bioinformatics/btm344, Vol. 23 no. 19, pages 2507-2517, June 2007.

[4] Yu,L. and Liu,H. (2004) Efficient feature selection via analysis of relevance and redundancy. J. Mach. Learn. Res., 5, 1205–1224.

[5] Ben-Bassat,M. (1982) Pattern recognition and reduction of dimensionality. In Krishnaiah,P. and Kanal,L., (eds.) Handbook of Statistics II, Vol. 1. North-Holland, Amsterdam. pp. 773–791.

[6] Inza,I., et al. (2000) Feature subset selection by Bayesian networks based optimization. Artif.Intell., 123, 157–184.
[7] X. Wang, J. Yang, X. Teng, W. Xia, J. Richard, Feature selection based on rough sets and particle swarm optimization, Pattern Recognition Letters 28 (2007) 459–471.
[8] M. Ronen, Z. Jacob, Using simulated annealing to optimize feature selection problem in marketing applications, European Journal of Operational Research 171 (2006) 842–858.
[9] S.F. Cotter, K. Kreutz-Delgado, B.D. Rao, Backward sequential elimination for sparse vector selection, Signal Processing 81 (2001) 1849–1864.
[10] Eugene Tuv et al (2009) Feature Selection with Ensembles, Artificial Variables, and Redundancy Elimination.Journal of Machine Learning Research 10, Pages 1341-1366, July 2009.
[11] M. Grobelnik et al."Text Mining as Integration of Several Related Research Areas: Report on KDD'2000 Workshop on Text Mining," 2000.
[12] Tellex, S., B. Katz, J. Lin, A. Fernandes, and G. Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pages 41–47.
[13] Pawel Smialowski et al., *Data and text mining* Pitfalls of supervised feature selection, Vol. *26 no. 3 2010, pages 440–443 doi:10.1093/bioinformatics/btp621*
[14] Chikina MD, Troyanskaya OG (2011) Accurate Quantification of Functional Analogy among Close Homologs. PLoS Comput Biol 7(2): e1001074. doi:10.1371/journal.pcbi.1001074
[15] Somorjai,R., et al. (2003) Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. Bioinformatics, 19, 1484–1491.
[16] Jafari,P. and Azuaje,F. (2006) An assessment of recently published gene expression data analyses: reporting experimental design and statistical factors. BMC Med. Inform. Decis. Mak., 6, 27.
[17] Sima,C., et al. (2005) Superior feature-set ranking for small samples using bolstered error estimation. Bioinformatics, 21, 1046–1054.
[18] Shital C. Shah, Andrew Kusiak, Data mining and genetic algorithm based gene/SNP selection, Artificial Intelligence in Medicine (2004) 31, 183—196
[19] Raychaudhuri S, Sutphin PD, Chang JT, Altman RB. Basicmicroarray analysis: grouping and feature reduction. Trends Biotechnol 2001;19(5):189—93.
[20]Tu Minh Phuong, Zhen Lin Russ B. Altman, Choosing SNPs Using Feature Selection, Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference (CSB'05)
[21] R. Aebersold and M. Mann, "Mass spectrometry-based proteomics," *Nature*, vol. 422, pp. 198-207, 2003
[22] Li,L. et al. (2004) Applications of the GA/KNN method to SELDI proteomics data. Bioinformatics, 20, 1638–1640.
[23] Yeung,K. and Bumgarner,R. (2003) Multiclass classification of microarray data with repeated measurements: application to cancer. Genome Biol., 4, R83.
[24] Ben-Dor,A., et al. (2000) Tissue classification with gene expression profiles. J. Comput. Biol., 7, 559–584
[25]Alyssa J Porter et al. (2009), ProMerge - A ToolKit for Data Capture and Integration in Differential Proteomics, 3rd Annual Conference In Quantitative Genomics, November 11-13, 2009, Joseph B Martin Conference Center, Boston MA, US
[26] Li, L., Weinberg, C.R., Darden, T.A. and Pedersen, L.G. (2001) Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics*, **17**, 1131-1142.
[27] Aharoni A. and Vorst O. 2002. DNA microarrays for functional plant genomics. Plant Mol. Biol. 48(1):99-118.
[28] Saeed AI, Sharov V, White J, Li J, Liang W, Bhagabati N, Braisted J, Klapa M, Currier T, Thiagarajan M, Sturn A, Snuffin M, Rezantsev A, Popov D, Ryltsov A, Kostukovich E, Borisovsky I, Liu Z, Vinsavich A, Trush V, Quackenbush J. TM4: a free, open-source system for microarray data management and analysis. *Biotechniques*. 2003 Feb;34(2):374-8.
[29] Bradley Efron and Tibshirani, 2007, On testing the signi_cance of sets of genes, Annals of Applied Statistics vol 1.
[30] Trevino &Falciani (2006), "GALGO: an R package for multivariate variable selection using genetic algorithms." Bioinformatics 22(9): 1154-6
[31] Nema Dean (2006), The Normal Uniform Di_erential Gene Expression (nudge) detection package.
[32] Yang et.al (2011), Bioconductor's DEDS package
[33] Rodrigo Alvarez-Gonzalez, et al. (2011), Discriminant Fuzzy Pattern to Filter Di_erentially Expressed Genes.
[34] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.
[35] Kohavi et al. (1996), Data Mining with MLC++. A broad view with a large comparison of many algorithms in MLC++ on the large UC Irvine datasets.Received the IEEE Tools with Artificial Intelligence Best Paper Award, 1996.
[36] Lei Yu, Yue Han, and Michael E Berens(2011). "Stable Gene Selection from Microarray Data via Sample Weighting".*IEEE/ACM Transactions on Computational Biology and Bioinformatics*, in press, 2011

Table 1.Softwares for Feature Selection

Software Names [References]	**Mass Spectra analysis FS Software**
ProMerge[25]	http://www.hsph.harvard.edu/research/bioinfocore/resources/software/index.html
GA/KNN[26]	http://www.niehs.nih.gov/research/resources/software/gaknn/index.cfm
Microarray analysis FS Software	
GA/KNN[26]	http://www.niehs.nih.gov/research/resources/software/gaknn/index.cfm
GeneMaths XT [27]	http://www.applied-maths.com/download/software.htm
TM4[28]	http://www.tm4.org/
SAM[29]	http://www-stat.stanford.edu/~tibs/SAM/
GALGO[30]	http://biptemp.bham.ac.uk/vivo/galgo/AppNotesPaper.htm
Nudge[31]	http://www.bioconductor.org/packages/release/bioc/html/nudge.html
DEDS[32]	http://www.bioconductor.org/packages/release/bioc/html/DEDS.html
DFP[33]	http://www.bioconductor.org/packages/release/bioc/html/DFP.html
General Purpose FS Software	
WEKA[34]	http://www.cs.waikato.ac.nz/ml/weka/
MLC++ [35]	http://www.sgi.com/tech/mlc/utils.html
FCBF[36]	http://www.public.asu.edu/~huanliu/FCBF/FCBFsoftware.html
Genomic Analysis	
SLAM[13]	http://bio.math.berkeley.edu/slam/
Multiz[13]	http://www.bx.psu.edu/miller_lab/

Modularity and Spectral Co-Clustering for Categorical Data

Lazhar Labiod and Mohamed Nadif

***Abstract*— To tackle the co-clustering problem on categorical data, we consider a spectral approach. We first define a generalized modularity measure for the co-clustering task. Then, we reformulate its maximization as a trace maximization problem. Finally we develop a spectral based co-clustering algorithm performing this maximization. The proposed algorithm is then capable to cluster rows and colunms simultaneously. Experimental results on synthetic and real data sets confirm the good performance of our algorithm.**

I. Introduction

Clustering is a method of unsupervised learning allowing the assignment of a set of observations into groups. Data clustering is a data analysis technique and has been considered as a primary data mining method for knowledge discovery. The central objective in clustering is the following: given a set of points and a pairwise distance measure, partition the set into clusters such that points that are close to each other according to the distance measure occur together in a cluster and points that are far away from each other occur in different clusters. The problem of clustering becomes more challenging when the data is categorical, due the difficulty of the choice of the distance to use. Ralambondrainy [36] has used the $k$means algorithm on binary data obtained after conversion of multiple category attributes into binary attributes. In contrast, Huang [13], has directly worked on the categorical data and proposed a simple criterion based on a measure giving the total mismatches of the corresponding attribute categories of the two instances. He proposed the $k$modes algorithm which is a version of $k$means adapted to categorical data. Jollois and Nadif [31] have considered the clustering problem under the classification maximum likelihood approach. In this setting, with a mixture model, they have defined a generalization of the $k$modes criterion. They proposed better fit criteria and developed the Classification Expectation-Maximization (CEM) algorithm. In addition, they have proved that $k$modes is just a particular version of CEM and the $k$modes criterion is associated to a multinomial mixture model with too restrictive constraints. Other approaches were also developed in data mining context such as [24] [26] and [1].

In several applications, the data itself has a lot of structure, which may be hard to capture using a traditional clustering objective. Consider the example of a Boolean matrix, whose rows correspond to objects (observation) and the columns correspond to attributes (variables), and an entry is one if and only if the attribute has owned by the object. The goal is to cluster both the objects and the attributes. One way to accomplish this would be to independently cluster the objects and attributes using the standard notion of clustering-cluster similar objects and cluster similar attributes. However, such a process might fail to elicit subtle structures that might exist between the two sets. This is precisely the advantage of co-clustering, also called *block clustering* or two mode analysis in a general context. These methods consider the two sets (the set of objects and the set of attributes) simultaneously and organize the data into homogenous blocks. The basic idea of these methods consist in making permutations of objects and attributes in order to draw a correspondence structure between this two sets. these last years co-clustering has received a significant amount of attention as an important problem with many application in data mining context.

The earliest co-clustering formulation called Direct clustering has been introduced by Hartigan [30], who proposes a greedy algorithm for hierarchical co-clustering. Dhillon [21] developed a spectral co-clustering algorithm on word-document data, the largest several left and right singular vectors of the normalized word-document matrix are computed and then a final clustering step using $k$means is applied to the data projected to the topmost singular vectors. In [20], the authors proposed an information-theoretic co-clustering algorithm that presents a non-negative matrix as an empirical joint probability distribution of two discrete random variables and set co-clustering problem under an optimization problem in information theory. Probabilistic model-based clustering techniques have also shown promising results in several co-clustering situations, the co-clustering of binary and contingency data has been treated by using latent block Bernoulli and Poisson models [28], [25]. The co-clustering implicitly performs an adaptive dimensionality reduction at each iteration, leading to better data clustering accuracy compared to one side clustering methods [20]. Co-clustering is also preferred when there is an association relationship between the data and the features (i.e., the columns and the rows) [19].

Even if the co-clustering problem is not the main objective of nonnegative factorization matrix (NMF), this approach has attracted many authors for data co-clustering and particularly for document clustering. Then, different algorithms based on nonnegative tri-factorization matrix are proposed. Given a nonnegative matrix A, they consist in seeking a 3-factor decomposition USV^t with all factor matrices restricted to be nonnegative (the superscript t denotes matrix transposition). The matrices U and V play the roles of row and column memberships. Each value of both matrices U and V corresponds to the degree in which a row or column belongs to a cluster. The matrix S makes it possible to absorb the scales of U, V and A. All proposed algorithms are iterative, which

Lazhar Labiod and Mohamed Nadif are with LIPADE, University of Paris Descartes, 45 rue des Saints Pères 75006 Paris, France; email: {firstname.secondname}@parisdescartes.fr.

can be differentiated by the update rules of the three matrices due to the chosen optimization method or the supplementary constraints imposed on the three matrices.

The modularity measure has been used recently for graph clustering [15] [6]. In this paper we show how Newman's modularity measure can be generalized to categorical data co-clustering and can be related to the broader family of spectral clustering methods. Specifically:

- We propose a new generalized modularity measure for categorical data co-clustering.
- We show how the problem of maximizing the generalized modularity measure $\tilde{Q}$ can be reformulated as an eigenvector problem. In this manner we link work on categorical data co-clustering using the generalized modularity measure to relevant work on spectral co-clustering [21].
- We develop an efficient spectral based procedure to find the optimal simultaneous objects and attributes partitions maximizing the normalized modularity criterion.

The rest of the paper is organized as follows: Section 2 introduces some notations and definitions. Section 3 provides the proposed generalized modularity measure. Some discussions on the spectral connection and optimization procedure are given in Section 4. Section 5 shows our experimental results and finally, Section 6 presents the conclusions and some future works.

II. Definitions and Notation

Let D be a dataset with a set I of N objects $(O_1, O_2, \ldots, O_N)$ described by the set V of M categorical attributes (or variables) $V^1, V^2, \ldots, V^m, \ldots, V^M$ each one having $p_1, \ldots, p_m, \ldots, p_M$ categories respectively and let $P = \sum_{m=1}^{M} p_m$ denote the full number of categories of all variables. Each categorical variable can be decomposed into a collection of indicator variables. For each variable V^m, let the p_m values naturally correspond to the numbers from 1 to p_m and let $V_1^m, V_2^m, \ldots, V_{p_m}^m$ be the binary variables such that for each j, $1 \leq j \leq p_m$, $V_j^m = 1$ if and only if the V^m takes the j-th value. Then the dataset can be expressed as a collection of $N \times p_m$ matrices K^m, $(m = 1, \ldots, M)$ of the general term k_{ij}^m such as: $k_{ij}^m = 1$ if the object i takes the attribute j of V^m and 0 otherwise.Then we obtain $N \times P$ binary disjunctive matrix

$$K = (K^1|K^2|\ldots|K^m|\ldots|K^M).$$

The table 1 shows different coding forms for a qualitative dataset containing 5 objects measured on a qualitative variable with 3 modalities. We will consider in the rest of this paper, the division of the sets I of objects and the set J of attributes into g non overlapping clusters, where g may be greater or equal to 2 and . Let us define an $N \times g$ index matrix R and an $M \times g$ index matrix C with one column for each cluster; $R = (R_1|R_2|\ldots|R_g)$ and $C = (C_1|C_2|\ldots|C_g)$. Each column is an index vector row of $(0, 1)$ elements such that $r_{ik} =$ (1 if object i belongs to cluster R_k, 0 otherwise), and $c_{jk} =$ (1 if attribute j belongs to cluster C_k, 0 otherwise).

TABLE I
LINEAR CODING - DISJUNCTIVE CODING

	V_1
o_1	1
o_2	2
o_3	1
o_4	2
o_5	3

$\mapsto$

	V_1^1	V_1^2	V_1^3
o_1	1	0	0
o_2	0	1	0
o_3	1	0	0
o_4	0	1	0
o_5	0	0	1

A. Weighted bipartite graph

An interesting connection between data matrices and graph theory can be established,. A data matrix can be viewed as a weighted bipartite graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, is said to be bipartite if its vertices can be partitioned into two sets I and J such that every edge in E has exactly one end in I and the other in J : $V = I \cup J$. The data matrix A can be viewed as a weighted bipartite graph where each node i in I corresponds to a row and each node j in J corresponds to a column. The edge between i and j has a weight a_{ij}, denoting the element of the matrix in the intersection between row i and column j.

III. Generalized Modularity Measure

This section shows how to adapt the Modularity measure for categorical data co-clustering. Hereafter, we review the modularity in graph clustering task.

A. Modularity and Graphs

Modularity is a recently quality measure for graph clustering, it has immediately received a considerable attention in several disciplines [15] [6]. Maximizing the modularity measure can be expressed in the form of an integer linear programming. Given the graph $G = (V, E)$, let A be a binary, symmetric matrix with (i, j) as entry; and $a_{ij} = 1$ if there is an edge between the nodes i and j. If there is no edge between nodes i and j, a_{ij} is equal to zero. We note that in our problem, A is an input having all information on the given graph G and is often called an adjacency matrix. Finding a partition of the set of nodes V into homogeneous subsets leads to the resolution of the following integer linear program: $\max_X Q(A, X)$ where $Q(A, X)$ is the modularity measure

$$Q(A, X) = \frac{1}{2|E|} \sum_{i,i'=1}^{n} (a_{ii'} - \frac{a_{i.}a_{i'.}}{2|E|}) \sum_{k=1}^{g} r_{ik} r_{i'k}.$$

Taking $x_{ij} = \sum_{k=1}^{g} r_{ik} r_{i'k}$, the expression of Q becomes

$$Q(A, X) = \frac{1}{2|E|} \sum_{i,i'=1}^{n} (a_{ii'} - \frac{a_{i.}a_{i'.}}{2|E|}) x_{ii'}, \qquad (1)$$

where $2|E| = \sum_{i,i'} a_{ii'} = a_{..}$ is the total number of edges and $a_{i.} = \sum_{i'} a_{ii'}$ the degree of i. Le $\delta = (\delta_{ij})$ be the $(n \times n)$ data matrix defined by $\forall i, i' \quad \delta_{ii'} = \frac{a_{i.}a_{i'.}}{2|E|}$, the expression (1) becomes

$$Q(A, X) = \frac{1}{2|E|} Tr[(A - \delta)X]. \qquad (2)$$

The researched binary matrix X is defined by RR^t which models a partition in a relational space and therefore must check the properties of an equivalence relation:

$$\begin{cases} x_{ii} = 1, \ \forall i & \text{reflexivity} \\ x_{ii'} - x_{i'i} = 0, \forall (i,i') & \text{symmetry} \\ x_{ii'} + x_{i'i''} - x_{ii''} \leq 1, \forall (i,i',i'') & \text{transitivity.} \end{cases}$$

B. Modularity measure for categorical data

The basic idea consists in modelling the simultaneous row and column partitions using a block seriation relation Z defined on $I \times J$. Noting that $Z = RC^t$ and the general term can be expressed as follows: $z_{ij} = 1$ if object i is in the same block as attribute j and $z_{ij} = 0$ otherwise. Then

$$z_{ij} = \sum_{k=1}^{g} r_{ik} c_{jk}.$$

Now, given a disjunctive matrix K, to tackle the co-clustering for categorical data, we propose the following generalized modularity measure $Q_1(K,Z)$:

$$Q_1(K,Z) = \frac{1}{2|E|} \sum_{i,j=1}^{n} (k_{ij} - \frac{k_{i.}k_{.j}}{2|E|}) z_{ij}. \quad (3)$$

where $2|E| = \sum_{i,j} k_{ij} = k_{..}$ is the total weight of edges and $k_{i.} = \sum_j k_{ij}$ - the degree of i and $k_{.j} = \sum_i k_{ij}$- the degree of j. This Modularity measure takes the following form

$$Q_1(K,Z) = \frac{1}{2|E|} Tr[(K-\delta)^t Z]. \quad (4)$$

The matrix Z represents a block seriation relation (see Marcotorchino [33] [32] for further details), then it must respect the following properties.

- **Binarity.**

$$z_{ij} \in \{0,1\}, \forall (i,j) \in I \times J. \quad (5)$$

- **Assignment constraints.** These constraints ensures the bijective correspondence between classes of two partitions, meaning that each class of the partition of I has one and one corresponding class of the partition J, and conversely, these constraints are expressed linearly as follows:

$$\begin{cases} \sum_{j \in J} z_{ij} \geq 1 & \forall i \in I \\ \sum_{i \in I} z_{ij} \geq 1 & \forall j \in J. \end{cases} \quad (6)$$

- **Triad impossible.** The role of these constraints is to ensure the blocks disjoint structure which is expressed by the following system inequality:

$$\begin{cases} z_{ij} + z_{ij'} + z_{i'j'} - z_{i'j} - 1 \leq 1 \\ z_{i'j'} + z_{i'j} + z_{ij} - z_{ij'} - 1 \leq 1 \\ z_{i'j} + z_{ij} + z_{ij'} - z_{i'j'} - 1 \leq 1 \\ z_{ij'} + z_{i'j'} + z_{i'j} - z_{ij} - 1 \leq 1. \end{cases} \quad (7)$$

Furthermore, noting that these constraints generalizes the transitivity for non symmetric data. In the case where I is equivalent to J ($I \equiv J$), it is easy to show that the block seriation relation Z becomes an equivalence relation i.e. $Z \equiv X$.

As the objective function (4) is linear with respect to Z and as the constraints that Z must respect are linear equations, theoretically we can solve the problem using an integer linear programming solver. However, this problem is NP hard, as result in practice we use heuristics for dealing with large data set.

IV. Maximization of the Normalized Generalized Modularity with Spectral Algorithm

A. Normalized generalized modularity

The expression 4 is not balanced by the row and column cluster size, meaning that a cluster might become small when affected by outliers. Thus we propose a new measure which we call normalized generalized modularity whose objective function is given as follows:

$$\tilde{\mathcal{Q}}_1(K,Z) = Tr[(K-\delta)^t G^{\frac{-1}{2}} Z F^{\frac{1}{2}}]. \quad (8)$$

where $G = diag(Z\mathbb{1})$ is a N by N diagonal matrix, each diagonal element g_{ii} corresponds to the number of attributes in the same block with the object i and $F = diag(Z^t\mathbb{1})$, each diagonal element f_{jj} gives the number of objects in the same block with the attribute j. Finally, $\mathbb{1}$ is the vector of the appropriate dimension which all its values are 1.

On the other hand, note that the expression (8) can be written as

$$\tilde{\mathcal{Q}}_1(K,Z) = Tr[(K-\delta)^t \tilde{Z}]. \quad (9)$$

where $\tilde{Z} = \tilde{R}\tilde{C}^t$ with $\tilde{R} = RG^{\frac{-1}{2}}$ and $\tilde{C} = CF^{\frac{-1}{2}}$. It is easy to verify that $\tilde{R}$ and $\tilde{C}$ satisfy the orthogonality constraint i.e.

$$\tilde{R}^t\tilde{R} = I_g \text{ and } \tilde{C}^t\tilde{C} = I_g,$$

then the maximization of the normalized generalized modularity is equivalent to the following trace optimization problem

$$\max_{\tilde{R}^t\tilde{R}=I_g, \tilde{C}^t\tilde{C}=I_g} Tr[\tilde{R}(K-\delta)\tilde{C}^t]. \quad (10)$$

This optimization problem can be performed by Lagrange multipliers into eigenvalue problem.

B. Spectral connection

In the following, the number of clusters g on I and J is assumed fixed. We use the following strategy to address the problem of finding a simultaneous partitioning that maximizes $\tilde{Q}_1(K,Z)$ as follows:

1) Approximate the resulting assignment problem by relaxing it to a continuous one which can be solved analytically using eigen-decomposition techniques.

2) Compute the first $(g-1)$ left and right eigenvectors of this solution to form a $(g-1)$-dimensional embedding of data into a Euclidean space. Then we use a hard-assignment thanks to $k$means on this new space to obtain a simultaneous clustering R and C.

Proposition. Let K be a disjunctive matrix, taking $D_r = diag(K\mathbb{1})$ and $D_c = diag(K^t\mathbb{1})$, the modularity matrix $(K-\delta)$ can be approximated by the $(g-1)$th largest eigenvectors of the scaled matrix

$$\tilde{K} = D_r^{\frac{-1}{2}} K D_c^{\frac{-1}{2}}$$

minus the trivial vectors (corresponding to the largest eigenvalue).

Proof: Note that we can rewrite K as

$$K = D_r^{\frac{1}{2}}(D_r^{\frac{-1}{2}} K D_c^{\frac{1}{2}}) D_c^{\frac{1}{2}}.$$

It is well known that the largest eigenvalue of

$$\tilde{K} = D_r^{\frac{-1}{2}} K D_c^{\frac{-1}{2}}$$

is equal to $\lambda_0 = 1$ and and the associated left and right eigenvectors are respectively [16], [17],

$$U_0 = \frac{D_r^{\frac{1}{2}}\mathbb{1}}{\sqrt{k_{..}}} \text{ and } V_0 = \frac{D_c^{\frac{1}{2}}\mathbb{1}}{\sqrt{k_{..}}}.$$

Applying the spectral decomposition of the scaled matrix $\tilde{K}$ instead on K directly, leading to

$$K = D_r^{\frac{1}{2}} \sum_{k\geq 0} U_k \lambda_k V_k^t D_c^{\frac{1}{2}}. \quad (11)$$

Subtract the trivial eigenvectors corresponding to the largest eigenvalue $\lambda_0 = 1$ give

$$K = \frac{D_r \mathbb{1}\mathbb{1}^t D_c}{k_{..}} + D_r^{\frac{1}{2}} \sum_{k\geq 1} U_k \lambda_k V_k^t D_c^{\frac{1}{2}}. \quad (12)$$

Keeping the $(g-1)$th first eigenvectors, we obtain the following approximation

$$K - \frac{D_r \mathbb{1}\mathbb{1}^t D_c}{k_{..}} \approx \sum_{k=1}^{g-1} \tilde{U}_k \lambda_k \tilde{V}_k^t \quad (13)$$

where

$$\tilde{U}_k = D_r^{\frac{-1}{2}} U_k \text{ and } \tilde{V}_k = D_c^{\frac{-1}{2}} V_k.$$

Then taking $\delta = \frac{D_r \mathbb{1}\mathbb{1}^t D_c}{k_{..}}$, we can approximate $(K-\delta)$ by

$$\sum_{k=1}^{g-1} \tilde{U}_k \lambda_k \tilde{V}_k^t.$$

Furthemore, note that the general term of δ is defined by $\delta_{ij} = \frac{k_{i.} k_{.j}}{k_{..}}$, that is its expresion in (3). ■

The modularity matrix $(K-\delta)$ used in (8) is expressed in terms of $(g-1)$th first eigenvectors of the scaled matrix $\tilde{K}$. Then we have a $(N \times (g-1))$ matrix

$$U = [U_1, ..., U_{g-1}]$$

formed by the $(g-1)$ left eigenvectors and a $(M \times (g-1))$ matrix

$$V = [V_1, ..., V_{g-1}]$$

formed by the $(g-1)$ right eigenvectors. We then normalize U into $\tilde{U}$ in which

$$\tilde{U}_k = \frac{D_r^{\frac{1}{2}} U_k}{||D_r^{\frac{1}{2}} U_k||},$$

and V into $\tilde{V}$ in which

$$\tilde{V}_k = \frac{D_c^{\frac{1}{2}} V_k}{||D_c^{\frac{1}{2}} V_k||}.$$

C. Spectral Co-clustering algorithm

The eigenmatrices $\tilde{U}$ and $\tilde{V}$ can be an input of the $k$means or other clustering algoritms via the following new matrix

$$Q = \begin{pmatrix} \tilde{U} \\ \tilde{V} \end{pmatrix} \quad (14)$$

The proposed algorithm called *SpecCo* begins by computing the first $(g-1)$ eigenvectors ignoring the trivial ones. This algorithm is similar in spirit to the one developed by Dhillon [21]. The algorithm embed the input data into the Euclidean space by eigen-decomposing a suitable affinity matrix and then cluster Q using a geometric clustering algorithm. Hereafter, the pseudo code of the proposed algorithm.

Algorithm 1 SpecCo

Input: data K, number of clusters g
Output: partition matrices R and C
1. Form the affinity matrix K
2. Define D_r and D_c to be the diagonal matrices

$$D_r = diag(K\mathbb{1}) \text{ and } D_c = diag(K^t\mathbb{1})$$

3. Find U,V the $(g-1)$ left-right largest eigenvectors of

$$\tilde{K} = D_r^{-\frac{1}{2}} K D_c^{-\frac{1}{2}}$$

4. From U and V, form the matrices $\tilde{U}$, $\tilde{V}$ and

$$Q = \begin{pmatrix} \tilde{U} \\ \tilde{V} \end{pmatrix}$$

5. Cluster the rows of Q into g clusters by using $k$means
6. Assign object i to cluster R_k if and only if the corresponding row Q_i of the matrix Q was assigned to cluster R_k and assign attribute j to cluster C_k if and only if the corresponding row Q_j of the matrix Q was assigned to cluster C_k.

The *SpecCo* algorithm contains two majors components: Computing the eigenvectors and executing $k$means to partition the rows and columns data. We run $k$means on Q each row is a $(g-1)$ vector. Standard $k$means with Euclidean distance metric has time complexity $O((N+M)dkt)$, where $(N+M)$ is the number of data points plus the number of attributes, and t is the number of iterations required for $k$means to converge. In addition, for the *SpecCo* algorithm there is the additional complexity for computing the matrix eigenvectors Q; for computing the largest eigenvectors using

the power method or Lanczos method [29], the running time is $O(N^2M)$ per iteration. Similar to other spectral graph clustering method, the time complexity of *SpecCo* can be significantly reduced if the affinity matrix K is sparse.

V. Experimental Results

A performance study has been conducted to evaluate our method *SepcCo*. To test its clustering performance against other algorithms, we ran our algorithm on real-life data set obtained from the UCI Machine Learning Repository. The description of the used data sets is given in Table II and the competitive retained algorithms are $k$modes [13] *Spec* proposed in [35], NMF and ONMTF developed in [19]. The update rules of NMF are defined with the row and column coefficients matrices $\mathbf{U}$ and $\mathbf{V}$ corresponding to the following approximation

$$K \approx \mathbf{U}\mathbf{V}^t,$$

and for ONMTF with the row, column coefficients matrices $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{S}$ ($\mathbf{S}$ consists to absorb the scales of U, V and K) corresponding to the following approximation

$$K \approx \mathbf{U}S\mathbf{V}^t.$$

These update rules of the three factors are reported in Table III where $\odot$ represents the Hadamard product.

TABLE II
DESCRIPTION OF THE DATA SETS

Data sets	# of Objects	# of Attributes	Classes
Soybean small	47	21	4
Zoo	101	16	7
Soybean large	307	35	19
Congressional votes	435	16	2

TABLE III
ALGORITHMS AND UPDATE RULES

Factors	NMF	ONMTF
$U =$	$U \odot \frac{AV}{UV^TV}$	$U \odot (\frac{AVS^T}{UU^TAVS^T})^{-\frac{1}{2}}$
$V =$	$V \odot \frac{A^TU}{VU^TU}$	$V \odot (\frac{A^TUS}{VV^TA^TUS})^{-\frac{1}{2}}$
$S =$	-	$S \odot (\frac{U^TAV}{U^TUSV^TV})^{-\frac{1}{2}}$

Validating clustering results is a non-trivial task. In the presence of true labels, as in the case of data sets we used, the clustering accuracy and normalized mutual information are employed to measure the quality of clustering.

A. Performance evaluation

In the first step, we focus on the quality of row clusters. Clustering Accuracy noted Acc discovers the one-to-one relationship between obtained clusters and true classes. It measures the extent to which each cluster contained data points from the corresponding class; it is defined as follows:

$$Acc = \frac{1}{N} max[\sum_{\mathcal{C}_k, \mathcal{L}_m} T(\mathcal{C}_k, \mathcal{L}_m)],$$

where $\mathcal{C}_k$ is the kth cluster in the final results, and $\mathcal{L}_m$ is the true mth class. $T(\mathcal{C}_k, \mathcal{L}_m)$ is the number of entities which belong to class m and are assigned to cluster k. Accuracy computes the maximum sum of $T(\mathcal{C}_k, \mathcal{L}_m)$ for all pairs of clusters and classes, and these pairs have no overlaps. The greater clustering accuracy means, the better clustering performance. The second measure employed is the normalized mutual information (NMI); it is estimated by

$$NMI = \frac{\sum_{k,\ell} N_{k,\ell} \log \frac{N_{k,\ell}}{N_k \hat{N}_\ell}}{\sqrt{(\sum_k N_k \log \frac{N_k}{N})(\sum_\ell \hat{N}_k \log \frac{\hat{N}_\ell}{N})}},$$

where N_k denotes the number of data contained in the cluster $\mathcal{C}_k (1 \leq k \leq g)$, $\hat{N}_\ell$ is the number of data belonging to the class $\mathcal{L}_\ell (1 \leq \ell \leq g)$, and $N_{k,\ell}$ denotes the number of data that are in the intersection between the cluster $\mathcal{C}_k$ and the class $\mathcal{L}_\ell$. The larger the NMI, the better the quality of clustering.

B. Results analysis

The results arising from experiments in Acc and NMI terms are reported in Tables IV and V. Note that, the proposed method *SpecCo* outperforms or is at least equivalent to the other algorithms. Even if here the objective is just the clustering, note that the co-clustering is beneficial.

TABLE IV
CLUSTERING ACCURACY (%)

Data sets	kmodes	NMF	ONMTF	Spec	SpecCo
Soybean small	97	100	100	100	100
Zoo	89	88	90	89	90
Soybean large	53	58	65	49	67
Congressional votes	86	81	87	87	87

TABLE V
NORMALIZED MUTUAL INFORMATION (%)

Data sets	kmodes	NMF	ONMTF	Spec	SpecCo
Soybean small	94	100	100	100	100
Zoo	89	87	80	83	92
Soybean large	67	71	77	60	78
Congressional votes	45	48	48	48	47

C. Co-clustering and visualization

To illustrate the obtained result in clustering task, we visualize, for instance, the data set Zoo in the figure 1 and tree synthetic 3×3 data sets generated according to a latent block Bernoulli mixtures model [27]: data1 (500×300), data2 (500×300) and data3 (1000×500) with 3 different patterns illustrated in the left of figures 2, 3 and 4. The co-clustering task is to recover groups of rows and columns. After the learning, the clusters indicators are given by the matrices R and C. We then reorganize the rows and columns separately or simultaneously according the obtained clusters

in the figures 1, 2, 3 and 4. From the simulated data, it can be seen that our method reconstructs effectively all co-clusters for balanced and unbalanced data sets.

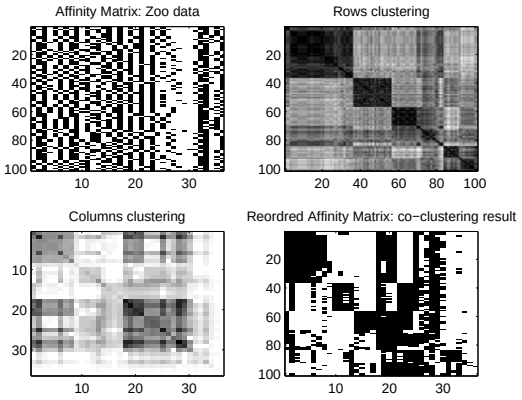


Fig. 1. Zoo data set.

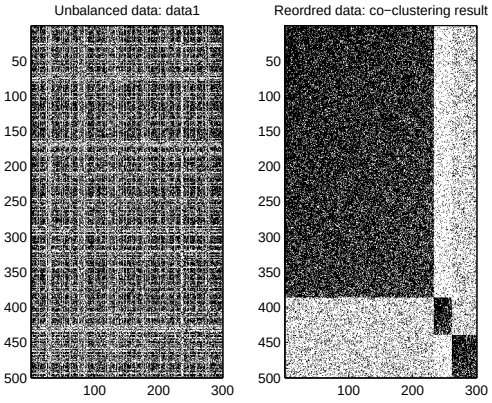


Fig. 2. Unbalanced data. Left: original data. Right reorganized data.

VI. Conclusion

In this paper, we propose a normalized generalized modularity criterion for categorical data in the aim of co-clustering. We have studied its maximization. An efficient spectral procedure for optimization is presented, the experimental results obtained using different simulated and real world data sets show that our method works effectively for categorical data. We obtain simultaneously row and columns clusters where each row cluster is characterized by a column cluster.

Our method can be easily extended to more general spectral framework for combining multiples heterogenous data sets for co-clustering. Thus, an interesting future work is to apply the approach on a variety of heterogenous data sets; numerical data, categorical data and graph data.

VII. Acknowledgment

This research was supported by the CLasSel ANR project ANR-08-EMER-002

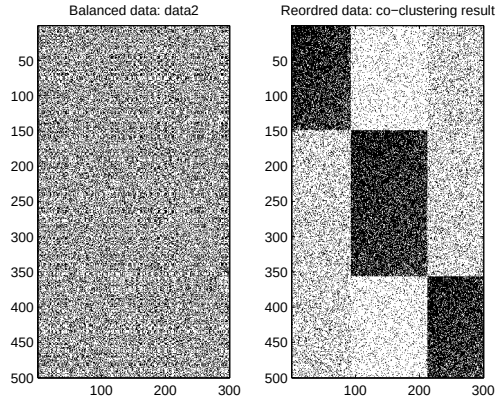


Fig. 3. Balanced data. Left: original data. Right reorganized data.

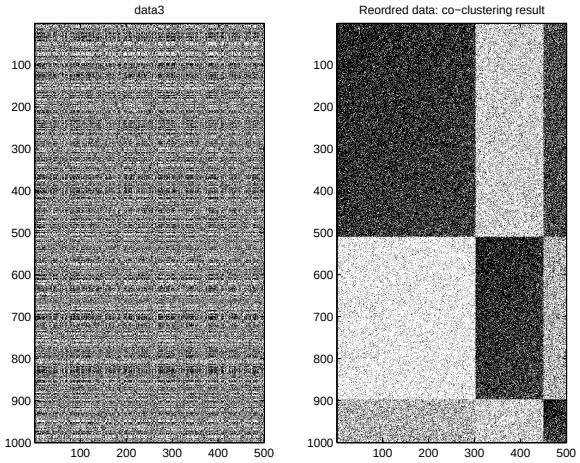


Fig. 4. data3-1000x500. Left: original data. Right reorganized data.

References

[1] V. Ganti, J. Gehrke and R. Ramakrishnan, "CACTUS - clustering categorical data using summaries," Proceedings of the Fifth ACM SIGKDD Conference, pp. 73- 83. 1999

[2] J. F. Marcotorchino, *Relational analysis theory as a general approach to data analysis and data fusion.* In Cognitive Systems with interactive sensors, 2006.

[3] J. F. Marcotorchino and P. Michaud, *Optimisation en analyse ordinale des données* In Masson, 1978.

[4] O. M. San, V. N. Huynh and Y. Nakamori, "An Alternative Extension of The kMeans algorithm For Clustering Categorical Data", J. Appl. Math. Comput. Sci, Vol. 14, No. 2, pp.241-247, 2004.

[5] R. S. Wallace, "Finding natural clusters through entropy minimization," (Technical Report CMU-CS-89- 183). Carnegie Mellon University, 1989.

[6] S. White and P. Smyth, "A spectral clustering approach to finding communities in graphs," In SDM, pp. 76-84, 2005.

[7] J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888-905, August 2000.

[8] A. Y. Ng, M. Jordan and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in Proc. of NIPS-14, 2001.

[9] P. Chan, M. Schlag and J. Zien, "Spectral k-way ratio cut partitioning," IEEE Trans. CAD-Integrated Circuits and Systems, vol. 13, pp. 1088-1096, 1994.

[10] S. Aranganayagi and K. Thangavel, "Improved K-Modes for Categorical Clustering Using Weighted Dissimilarity Measure", International Journal of Engineering and Mathematical Sciences. vol5-2-19, 2009.

[11] U. Von Luxburg, "A Tutorial on Spectral Clustering," Technical Report at MPI Tuebingen, 2006.
[12] H. Bock, "Probabilistic aspects in cluster analysis,". In O. Opitz (Ed.), Conceptual and numerical analysis of data, pp. 12-44. Berlin: Springer-verlag, 1989.
[13] Z. Huang, "Extensions to the k-means algorithm for proposition clustering large data sets with categorical values,". Data Mining and Knowledge Discovery, 2, pp. 283-304, 1998.
[14] T. Li, S. Ma and M. Ogihara, "Entropy-based criterion in categorical clustering," ICML'04, pp. 536-543, 2004.
[15] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," Physical Review E., 69, 026113. 2004.
[16] C. Ding, H. Xiaofeng, Z. Hongyuan and S. Horst, "Self-aggregation in scaled principal component space,". Technical Report LBNL–49048. Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley, CA, USA, 2001.
[17] F. R. Bach and M. I. Jordan, "Learning spectral clustering, with application to speech separation," Journal of Machine Learning Research, 2005
[18] B. Long, Z. Zhang, Ph. S. Yu, "Co-clustering by value decomposition," *KDD'05*, pp. 635-640, 2005.
[19] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," *in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, Philadelphia, PA, 2006.
[20] I. Dhillon, S. Mallela and D. S. Modha, "Information-Theoretic Co-clustering," *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 89-98, 2003.
[21] I. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," *ACM SIGKDD International Conference*, San Francisco, USA, pp. 269-274, 2001.
[22] J. Yoo and S. Choi, "Orthogonal nonnegative matrix tri-factorization for co-clustering: Multiplicative updates on Stiefel manifolds," *Information Processing & Management* Volume 46, Issue 5, pp. 559-570, September 2010.
[23] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature* 401, pp. 788-791, 1999.
[24] D. Gibson, J. Kleinberg and P. Raghavan, "Clustering categorical data: An approach based on dynamical systems," VLDB'98, pp. 311-323, 1998.
[25] G. Govaert and M. Nadif, "Block clustering with Bernoulli mixture models: Comparison of different approaches," *Computational Statistics and Data Analysis*, 52, pp. 233-3245, 2008.
[26] S. Guha, R. Rastogi and K. Shim, "Rock: A robust clustering algorithm for categorical attributes," *IEEE International Conference on Data Engineering*, Sydney, 1999.
[27] G. Govaert and M. Nadif, "Clustering with block mixture models," *Pattern Recognition*, pp. 463-473, 2003.
[28] G. Govaert and M. Nadif, "Latent Block Model for contingency table," Communications in Statistics, Theory and Methods, 39, pp 416-425, 2010.
[29] G. H. Golub and C. F. Van Loan, "Matrix Computations," John Hopkins Press, 1999.
[30] J. A. Hartigan. "Direct clustering of a data matrix," *Journal of the American Statistical Association* 67 (337), pp. 123-129, 1972.
[31] F.X. Jollois and M. Nadif, "Clustering Large Categorical Data," PAKDD'02: pp. 257-263, 2002.
[32] F. Marcotorchino, "Seriation problems : An overview," *Applied Stochastic Models And Data Analysis* vol 7 , pp. 199-202, Wiley, 1991.
[33] F. Marcotorchino, "Block seriation problems : A unified approch," *Applied Stochastic Models and Data Analysis* vol 3, pp.7391, Wiley, 1987.
[34] F. Marcotorchino, "Une approche unifiée des méthodes de sériation par blocs," *Data anlysis and Informatics. E.Diday. Elsevier Science Publishers B.V. North Holland*, 1988.
[35] L. Labiod and Younès Bennani. "A Spectral Based Clustering Algorithm for Categorical Data with Maximum Modularity," in Proc. of the ESANN'11, European Symposium on Artificial Neural Networks. Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011.
[36] H. Ralambondrainy, "A Conceptual Version of the k-means Algorithm," *Pattern Recognition Letters*, 16, pp.1147-1157, 1995.

Statistical Procedure For Simultaneous Testing of Many Hypothesis

Gurpreet Singh Bawa

Chandigarh-India

cell no. + 91 9049658009, +91 9540131280

Gurpreetsinghbawa@yahoo.co.in

Abstract

Suppose we have k points (where, $k > 2$) in our parametric space and on the basis of given information, we want to test which of these k points is true, for testing such situation our suggested framework provides a systematic mathematical approach to find best criterion. The Neyman-Pearson (NP) approach specifies the most powerful test of size α when parametric space is partitioned into two disjoint sets (or it contains only two points), under the assumption that the distributions for each hypothesis are known or (in some cases) the likelihood ratio is monotonic in the unknown parameter. Here we extend the NP theory to situations in which one has to investigate about the population parameter, when parametric space is partitioned into k (where, $k > 2$) disjoint parts or contains k points from real line. NP theory turns out to be a special case of what is given here with $k = 2$.

Key words : *Neyman-Pearson lemma, class of tests, Hypothesis testing, Most powerful test, Power of a test.*

1 Introduction

A statistical test is a method of making statistical decisions using experimental data. Usually, the parametric space is partitioned into two mutually exclusive and exhaustive parts or is a set that consists of exactly two points. Based on a sample from the population, decisions are made regarding the correctness of the two complementary hypotheses. Satisfactory theory of statistical inference (test) based on the frequency of errors resulting from the use of an inference procedure had been advanced by NP theory in 1933. There they developed a theory for binary hypothesis testing ($k = 2$) with two possible errors (false acceptance and false rejection) and searched for a best test (minimizing the frequency of errors) from a class of unbiased tests of fixed size. The NP approach to hypothesis testing is useful only in situations where we have two points in the parametric space or the parametric space can be partitioned into two disjoint sets. The problem occurs when one considers a set, or a family of statistical inferences simultaneously. Here, parametric space is partitioned into k mutually exclusive and exhaustive parts or it contains k points from real line. We are interested in extending the NP (binary hypothesis test) paradigm to the situations where one has to identify the correct value from k (> 2) points in the parametric space.

To motivate the need of extending the NP approach, consider the problem of classifying cancer ($k > 2$) patients using genes expression. Firstly, identify several patients whose status for a particular type of cancer is known. Next, collect cell samples from the appropriate tissue in each patient. Then, assess the relative abundance of various genes in each of the subjects. Repeat the above steps for all types of cancer patients and finally use these training data to build a classifier (test), that in principle be used to diagnose future patients (based on their gene expression profiles) by knowing the type of cancer they have, then sent for further treatment. NP approach is not applicable for identifying the type of cancer the patient is suffering from, as there are more than 2 types of cancers. It can only test whether a patient is suffering from particular type of cancer or not, but using our approach, we are able to identify the type of cancer the patient is suffering from.

To the best of our knowledge the present work is a new attempt, which can be considered as a generalization of NP-lemma for identifying the correct value of the parameter

when there are k (> 2) points in the parametric space. We provide general definitions of unbiased, identical and biased tests and follow a systematic approach to find the best test from the class of unbiased tests, for testing which of the k points in the parametric space is true, on the basis of a given sample from the population. The same can also be extended further to identify (test) the value of parameter, if the parametric space is an interval (and contains infinite number of points). The main focus of our work is on deriving the best test (from the class of unbiased tests). Interestingly, when $k = 2$ our results coincides with the results of NP approach.

In Section 2 we have proposed the theory of statistical tests and the general format of the hypothesis. This includes different types of errors resulting from framed inference procedure and their relationships. In Section 3 we have mentioned the chain of reasoning and different classes of tests and their properties. This is followed by introduction of a lemma, which provides a systematic mathematical approach for finding the best test from the class of unbiased tests with maximum efficiency. The last section discuss few examples.

2 Proposed theory of statistical tests

2.1 General Formulation

A hypothesis consists either of a suggested explanation for an observable phenomenon or of a reasoned proposal predicting a possible causal correlation among multiple phenomena. Let Θ denotes a parametric space of population parameter θ, where $\Theta = \{\theta_1, \theta_2, \theta_3, \cdots, \theta_{k-1}, \theta_k\}$. In all instances, the parametric space is a set consist of k points from real line, where $k > 2$. The general format of hypothesis, which we would like to test is

$$H_1 : \theta = \theta_1;\ H_2 : \theta = \theta_2;\ H_3 : \theta = \theta_3, \cdots, H_k : \theta = \theta_k.$$

The acceptance of anyone of them rejects all the others.

We restrict all our discussion to the case when $k = 3$. The results can be easily extended to any arbitrary integer k (> 3).

Here the aim of hypothesis testing is to formally examine three opposing conjectures.

$$H_1 : \theta = \theta_1;\ H_2 : \theta = \theta_2;\ H_3 : \theta = \theta_3,$$

therefore, the parametric space Θ is a set of 3 elements viz., θ_1, θ_2, and θ_3. We view this problem as a problem in which one of three actions will take place from the set of an action space. Let E_{n1}, E_{n2}, E_{n3} form a partition of the sample space such that $E_{ni} \cap E_{nj} = \phi, i \neq j$ and $\bigcup_{i=1}^{3} E_{ni}$ = sample space. If the observed $\underline{x}$ lies in E_{ni}, we accept the hypothesis $H_i, i = 1, 2, 3$.

2.2 Errors resulting from framed inference procedure

The goal is to determine the correct identification of the hypothesis. The natural procedure is to suggest a test for this problem. Any suggested test for this hypothesis can have six different types of errors. They are

Type I error : Accept H_1 when H_2 is true

Type II error : Accept H_1 when H_3 is true

Type III error : Accept H_2 when H_1 is true

Type IV error : Accept H_2 when H_3 is true

Type V error : Accept H_3 when H_1 is true

Type VI error : Accept H_3 when H_2 is true.

In order to reduce the frequency of mentioned error, we switch ourselves to the **probability of false acceptance**. Let

$$\begin{array}{rcl} P_{\theta 2}(E_{n1}) & = & \text{Probability of type I error} \\ P_{\theta 3}(E_{n1}) & = & \text{Probability of type II error} \\ P_{\theta 1}(E_{n2}) & = & \text{Probability of type III error} \\ P_{\theta 3}(E_{n2}) & = & \text{Probability of type IV error} \\ P_{\theta 1}(E_{n3}) & = & \text{Probability of type V error} \\ P_{\theta 2}(E_{n3}) & = & \text{Probability of type VI error} \end{array} \tag{2.1}$$

The following tabular representation will make this formulation clear.

Power I	*P(Type I Error)*	*P(Type II Error)*
P(Type III Error)	*Power II*	*P(Type IV Error)*
P(Type V Error)	*P(Type VI Error)*	*Power III*

We have to reduce all these probabilities simultaneously in order to reduce the frequency of false acceptance. Minimization of $P_{\theta 2}(E_{n1})$ and $P_{\theta 2}(E_{n3})$ is same as the minimization of $P_{\theta 2}(E_{n1} \cup E_{n3})$, which is same as minimization of $P_{\theta 2}(E'_{n2})$. Similarly, minimization of $P_{\theta 1}(E_{n2})$ and $P_{\theta 1}(E_{n3})$ is same as the minimization of $P_{\theta 1}(E'_{n1})$ and minimization of $P_{\theta 3}(E_{n2})$ and $P_{\theta 3}(E_{n1})$ is same as the minimization of $P_{\theta 3}(E'_{n3})$. Therefore, we need to minimize the probabilities of three events simultaneously; they are $P_{\theta 1}(E'_{n1}), P_{\theta 2}(E'_{n2})$ and $P_{\theta 3}(E'_{n3})$. Therefore, minimization of (2.1) is the same as the minimization of these three probabilities.

2.3 Relation between different errors

We have seen that in order to minimize the errors associated with our decision about the population parameter, we have to minimize $P_{\theta 1}(E'_{n1}), P_{\theta 2}(E'_{n2})$ and $P_{\theta 3}(E'_{n3})$. Typically, in the process of reducing $P_{\theta 1}(E'_{n1})$, we have to increase the span of the region E_{n1}, which definitely increase the span of E'_{n2} and/or E'_{n3} and the probability associated with them (see Figure 2.1)

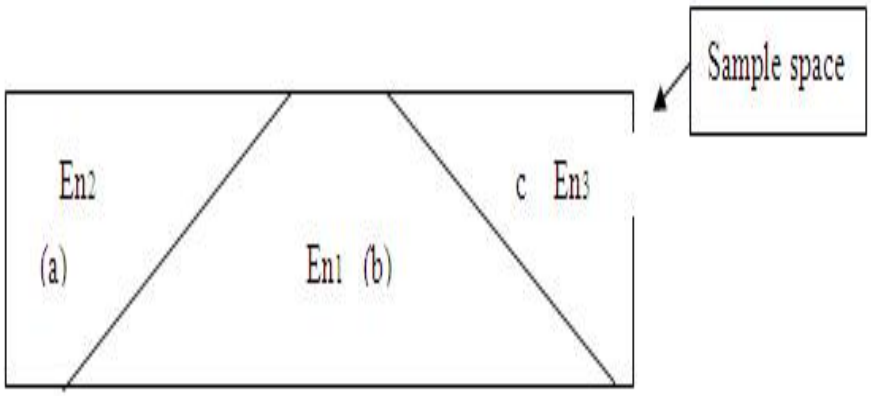


Figure 2.1 : Sample space and the corresponding partition

Similarly, reducing $P_{\theta 2}(E'_{n2})$ is essentially same as increasing $P_{\theta 1}(E'_{n1})$ and/or $P_{\theta 3}(E'_{n3})$ and reducing $P_{\theta 3}(E'_{n3})$ is increasing $P_{\theta 1}(E'_{n1})$ and/or $P_{\theta 2}(E'_{n2})$. Thus, minimization of any

$P_{\theta j}(E'_{nj}), j = 1, 2, 3$ will naturally increase the $P_{\theta i}(E'_{ni})$ for $i = 1, 2, 3$ for $i \neq j$.

$$P_{\theta 1}(E'_{n1}),\ P_{\theta 2}(E'_{n2}) \ \text{ and } \ P_{\theta 3}(E'_{n3}) \tag{2.2}$$

3 Classes of tests

We have seen that in the process of reducing any of the probability in the expression (2.2), the other probabilities automatically get increased. Therefore, in order to overcome this issue, a compromise should be reached to limit the probability of more serious type of errors and then to minimize the other probabilities. So, we prefix the probability of E'_{n1} under θ_1 to a maximum of our limit (say α), while making decision about the population parameter. By doing so, we are with the class of tests $\mathcal{K}$ having upper bound for the probability of E'_{n1} under θ_1 as α. Thus, if a test $(E^*_{n1}, E^*_{n2}, E^*_{n3})$ belongs to class of test $\mathcal{K}$ then $P_{\theta 1}(E^{*'}_{n1}) \leq \alpha$ (prefixed).

As we have planed the maximum tolerance of $P_{\theta 1}(E'_{n1})$, we are now much more concerned about the remaining two probabilities $P_{\theta 2}(E'_{n2})$ and $P_{\theta 3}(E'_{n3})$. In order to reduce $P_{\theta 2}(E'_{n2})$ and $P_{\theta 3}(E'_{n3})$, we have to shrink the region b (see Figure 2.1). This is possible if $P_{\theta 1}(E_{n1})$ move towards its smallest lower bound $(1 - \alpha)$ or we can say that $P_{\theta 1}(E'_{n1})$ attain its greatest upper bound, which is α. By doing this we are with class of tests $\mathcal{A}$ - the subset of class of tests $\mathcal{K}$ such that if a test $(E^*_{n1}, E^*_{n2}, E^*_{n3})$ belongs to class of tests $\mathcal{A}$ then $P_{\theta 1}(E^{*'}_{n1}) = \alpha$.

As region b is fixed, to make its probability under θ_1 as $1-\alpha$, we expand the region a in order to reduce $P_{\theta 2}(E'_{n2})$, which in result increase $P_{\theta 3}(E'_{n3})$. In order to reduce $P_{\theta 3}(E'_{n3})$, we have to expand the region c, which result in increase of $P_{\theta 2}(E'_{n2})$. Therefore, we are not in a position to reduce $P_{\theta 2}(E'_{n2})$ and $P_{\theta 3}(E'_{n3})$ simultaneously. Hence we prefix the probability of E'_{n2} under θ_2 to a value (say β), while making decision about the population parameter.

Thus, we are with the class of tests $(\mathcal{B})$ having an upper bound of probability of E'_{n2} under θ_2 as β and probability of E'_{n1} under θ_1 as α. If a test $(E^*_{n1}, E^*_{n2}, E^*_{n3})$ belongs to class of tests $\mathcal{B}$, then $P_{\theta 1}(E^{*'}_{n1}) = \alpha$ (prefixed) and $P_{\theta 2}(E^{*'}_{n2}) \leq \beta$. As we have fixed a maximum tolerance for $P_{\theta 1}(E'_{n1})$ and $P_{\theta 2}(E'_{n2})$, next we consider the probability $P_{\theta 3}(E'_{n3})$. Note that

$P_{\theta 3}(E'_{n3}) = 1 - P_{\theta 3}(E_{n3})$, where $P_{\theta 3}(E_{n3})$ is probability of accepting the H_3 when it is true (the power). Thus, we have to maximize the probability of E_{n3} under θ_3. This is same as reducing $P_{\theta 3}(E'_{n3})$. In order to reduce $P_{\theta 3}(E'_{n3})$, we have to increase the region c (see Figure 2.1). Maximization of the region c is possible only when $P_{\theta 2}(E_{n2})$ move towards its smallest lower bound $(1 - \beta)$ or we can say that $P_{\theta 2}(E'_{n2})$ attain its greatest upper bound β. This would give us the class of tests $(\mathcal{D})$ such that if a test (E_{n1}, E_{n2}, E_{n3}) belongs to class $\mathcal{D}$ then $P_{\theta 1}(E'_{n1}) = \alpha, P_{\theta 2}(E'_{n2}) = \beta$ and $P_{\theta 3}(E_{n3})$ is arbitrary.

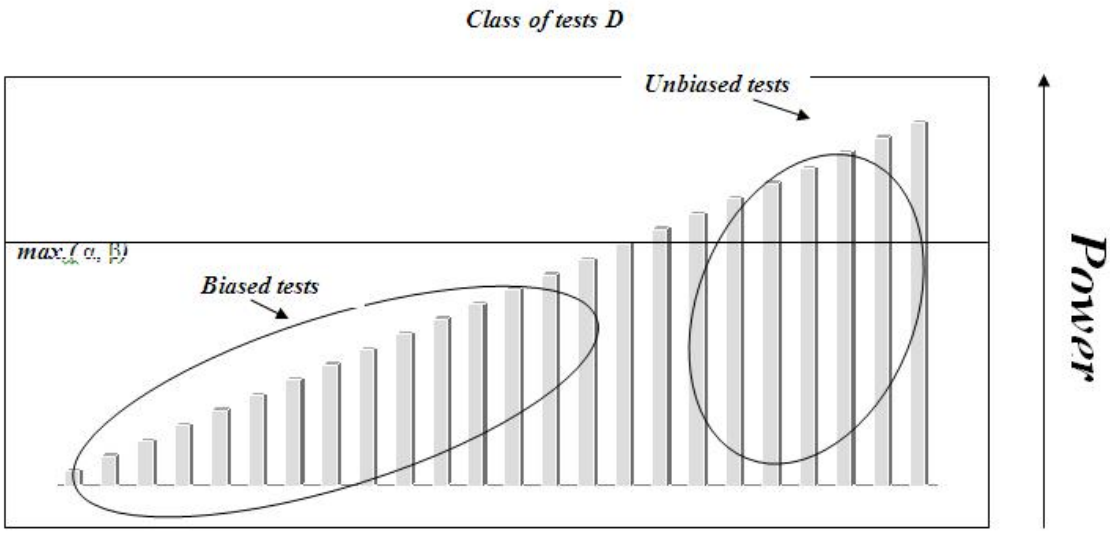


Figure 3.1: Class of tests D

3.1 Tests and their properties

If a test (E_{n1}, E_{n2}, E_{n3}) belongs to class $\mathcal{D}$, then $P_{\theta 1}(E'_{n1}) = \alpha$ and $P_{\theta 2}(E'_{n2}) = \beta$. Then either one of the three can happen, that is $\alpha = \beta, \alpha > \beta$ and $\alpha < \beta$. We restrict all our discussion to the case when $\alpha = \beta$. The result can easily be extended to all other cases. Thus, $\mathcal{D}$ is defined a class of tests such that if a test (E_{n1}, E_{n2}, E_{n3}) belongs to class $\mathcal{D}$ then $P_{\theta 1}(E'_{n1}) = \alpha$ and $P_{\theta 2}(E'_{n2}) = \alpha$.

Best Test

Test (E_{n1}, E_{n2}, E_{n3}) is said to be best if $P_{\theta 1}(E'_{n1}) = \alpha, P_{\theta 2}(E'_{n2}) = \beta$ and $P_{\theta 3}(E_{n3})$ is

maximum among all members of the class. When $\alpha = \beta$, the test is best if $P_{\theta 1}(E'_{n1}) = \alpha, P_{\theta 2}(E'_{n2}) = \alpha$ and $P_{\theta 3}(E_{n3})$ is maximum among all members of the class.

We have described various classes of hypothesis tests. Some of these classes control the probability of a $P_{\theta 1}(E'_{n1})$ and $P_{\theta 2}(E'_{n2})$. A good test in such a class would also have a small $P_{\theta 3}(E'_{n3})$ than all other tests in the class, it would certainly be a stronger contender for the best test in the class.

Unbiased Test

Test (E_{n1}, E_{n2}, E_{n3}) is said to be unbiased if $P_{\theta 1}(E'_{n1}) = \alpha, P_{\theta 2}(E'_{n2}) = \beta$ and $P_{\theta 3}(E_{n3})$ is not less than maximum of (α, β).

Biased Test

Test (E_{n1}, E_{n2}, E_{n3}) is said to be biased if $P_{\theta 1}(E'_{n1}) = \alpha, P_{\theta 2}(E'_{n2}) = \beta$ and $P_{\theta 3}(E_{n3})$ is less than maximum of (α, β).

Identical Test

Test (E_{n1}, E_{n2}, E_{n3}) is said to be identical if $P_{\theta 1}(E'_{n1}) = \alpha, P_{\theta 2}(E'_{n2}) = \beta$ and $P_{\theta 3}(E_{n3})$ is maximum of (α, β).

3.2 Test function

A function $\phi(\underline{x})$ defined on the sample space to $\{\gamma_1, \gamma_2, \gamma_3\}$, such that its value is γ_1 if observed value $\underline{x}$ is in E_{n1}, γ_2 if $\underline{x}$ is in E_{n2} and γ_3 if $\underline{x}$ is in E_{n3} is known as a test function. The constants $\gamma_1, \gamma_2, \gamma_3$ are evaluated using the expressions $E_{\theta 1}(\phi(\underline{X})) = P_{\theta 1}(E'_{n1}), E_{\theta 2}(\phi(\underline{X})) = P_{\theta 2}(E'_{n2})$ and $E_{\theta 3}(\phi(\underline{X})) = P_{\theta 3}(E_{n3})$.

Lemma 3.1 : The best test for testing the hypothesis $H_1 : \theta = \theta_1; H_2 : \theta = \theta_2; H_3 : \theta = \theta_3$ is an unbiased test.

Proof : Let $\phi(\underline{x})$ be the best test for testing $H_1 : \theta = \theta_1; H_2 : \theta = \theta_2; H_3 : \theta = \theta_3$. This implies that $E_{\theta 1}(\phi(\underline{X})) = \alpha = E_{\theta 2}(\phi(\underline{X}))$ and $E_{\theta 3}(\phi(\underline{X}))$ is maximum among all tests in class $\mathcal{D}$. Let $\phi^*(\underline{x})$ be another test for testing the $H_1 : \theta = \theta_1; H_2 : \theta = \theta_2; H_3 : \theta = \theta_3$ defined as $\phi^*(\underline{x}) = \alpha, \quad \forall\ \underline{x}$. Then, $E_{\theta 1}(\phi^*(\underline{X})) = \alpha = E_{\theta 2}(\phi^*(\underline{X}))$ which implies that $\phi^*(\underline{x})$ belongs to class of tests $\mathcal{D}$. By definition of best test $E_{\theta 3}(\phi(\underline{X})) \geq E_{\theta 3}(\phi^*(\underline{X}))$, which implies that $E_{\theta 3}(\phi(\underline{X})) \geq \alpha$ (as $E_{\theta 3}(\phi^*(\underline{X})) = \alpha$). Thus the best test is unbiased.

Lemma 3.2 : Let $X_1, X_2, \cdots, X_n$, be a random sample from $f(x, \theta)$. Then the likelihood

function of $X_1, X_2, \cdots, X_n$ is given by $L(\theta; \underline{x}) = \prod_{i=1}^{n} f(x_i, \theta)$. Let θ_1, θ_2 and θ_3 be the three distinct values of θ, and k_1, k_2 be two real numbers. Let E_{n1}, E_{n2}, E_{n3} be the partition of sample space such that:

$$\text{(i)} \qquad L(\theta_1; \underline{x})/L(\theta_3; \underline{x}) \leq k_1, \quad \text{for} \quad \underline{x} \in E_{n3} \tag{3.1}$$

$$> k_1, \quad \text{for} \quad \underline{x} \in E_{n1} \cup E_{n2} \tag{3.2}$$

Now in span of (3.2) we have to look for,

$$\text{(ii)} \quad \begin{array}{lll} L(\theta_1; \underline{x})/L(\theta_2; \underline{x}) & \leq & k_2, \quad \text{for} \quad \underline{x} \in E_{n2} \\ & > & k_2, \qquad\quad \underline{x} \in E_{n1} \end{array} \tag{3.3}$$

(iii) $\alpha = P_{\theta 1}(E'_{n1}) = P_{\theta 2}(E'_{n2})$.

Then

(*Sufficiency*) Any test that satisfies (i), (ii) and (iii) is best of given size for testing $H_1 : \theta = \theta_1; H_2 : \theta = \theta_2; H_3 : \theta = \theta_3$.

(*Necessity*) If there exists a test of given size for testing $H_1 : \theta = \theta_1; H_2 : \theta = \theta_2; H_3 : \theta = \theta_3$ and is as powerful as test by (i), (ii) and (iii) then they would essentially have the same region; that is, they could differ only by a set having probability zero.

Proof : If (E_{n1}, E_{n2}, E_{n3}) is the only member of $\mathcal{D}$, then there is no proof required. Let $(E^*_{n1}, E^*_{n2}, E^*_{n3})$ be another member of $\mathcal{D}$. Then we need to establish that

$$\int_{E_{n3}} L(\theta_3; \underline{x}) d\underline{x} \geq \int_{E^*_{n3}} L(\theta_3; \underline{x}) d\underline{x}.$$

Now, $E_{n3} = (E_{n3} \cap E^*_{n3}) \cup (E_{n3} \cap E^{*'}_{n3})$. Similarly $E^*_{n3} = (E_{n3} \cap E^*_{n3}) \cup (E'_{n3} \cap E^*_{n3})$.

Therefore,

$$\begin{aligned}
&\int_{E_{n3}} L(\theta_3;\underline{x})d\underline{x} - \int_{E^*_{n3}} L(\theta_3;\underline{x})d\underline{x} = \int_{E_{n3}\cap E^*_{n3}} L(\theta_3;\underline{x}d\underline{x} + \int_{E_{n3}\cap E^{*'}_{n3}} L(\theta_3;\underline{x})d\underline{x} \\
&- \int_{E_{n3}\cap E^*_{n3}} L(\theta_3;\underline{x})d\underline{x} - \int_{E'_{n3}\cap E^*_{n3}} L(\theta_3;\underline{x})d\underline{x} \\
&= \int_{E_{n3}\cap E^{*'}_{n3}} L(\theta_3;\underline{x})d\underline{x} - \int_{E'_{n3}\cap E^*_{n3}} L(\theta_3;\underline{x})d\underline{x} \qquad (3.4) \\
&\geq \frac{1}{k_1}\left\{\int_{E_{n3}\cap E^*_{n3}} L(\theta_1;\underline{x})d\underline{x} - \int_{E'_{n3}\cap E^*_{n3}} L(\theta_1;\underline{x})d\underline{x}\right\} \\
&= \frac{1}{k_1}\left\{\begin{array}{cccc} \int_{E_{n3}\cap E^*_{n3}} L(\theta_1;\underline{x})d\underline{x} & - & \int_{E'_{n3}\cap E^*_{n3}} L(\theta_1;\underline{x})d\underline{x} \\ +\int_{E_{n3}\cap E^*_{n3}} L(\theta_3;\underline{x})d\underline{x} & - & \int_{E'_{n3}\cap E_{n3}} L(\theta_1;\underline{x})d\underline{x} \end{array}\right\} \\
&= \frac{1}{k_1}\left\{\int_{E_{n3}} L(\theta_1;\underline{x})d\underline{x} - \int_{E^*_{n3}} L(\theta_1;\underline{x})d\underline{x}\right\}
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{k_1}\left\{1 - \int_{E_{n1}\cup E_{n2}} L(\theta_1;\underline{x})d\underline{x} - 1 + \int_{E^*_{n1}\cup E^*_{n2}} L(\theta_1;\underline{x})d\underline{x}\right\} \\
&= 0.
\end{aligned}$$

$$\begin{aligned}
&\Rightarrow \int_{E_{n3}} L(\theta_3;\underline{x})d\underline{x} - \int_{E^*_{n3}} L(\theta_3;\underline{x})d\underline{x} \geq 0 \\
&\Rightarrow \int_{E_{n3}} L(\theta_3;\underline{x})d\underline{x} \geq \int_{En3^*} L(\theta_3;\underline{x})d\underline{x}.
\end{aligned}$$

As stated in theorem, conditions (i), (ii) and (iii) are sufficient ones for the region (E_{n1}, E_{n2}, E_{n3}) to be a best region of given size. Suppose there is a region $(E^*_{n1}, E^*_{n2}, E^*_{n3})$ of given size that doesn't satisfies (i), (ii) and is powerful as (E_{n1}, E_{n2}, E_{n3}). Then the expression (3.4) would be zero, since power of (E_{n1}, E_{n2}, E_{n3}) is the same as that of $(E^*_{n1}, E^*_{n2}, E^*_{n3})$. As a matter of fact, (E_{n1}, E_{n2}, E_{n3}) and $(E^*_{n1}, E^*_{n2}, E^*_{n3})$ would essentially be the same region, that is, they could differ only by a set having probability zero, each would necessarily enjoy the conditions (i), (ii) and (iii) to be a best test of given size.

It can be seen that the test stated in lemma is **unique** (except on a null set), if there exist a test which is most powerful from an unbiased class it must be of the form (i), (ii) and (iii) except on a set whose probability under θ_i is surely zero. $\mathcal{D}$ is also a non

empty set as identical test which is unbiased always exists for every case. So, atleast one unbiased test is there in $\mathcal{D}$ which act as a best test when no other unbiased test exists. Hence, **existence** is always guaranteed.

Remark 1 : The results can easily be extended to any arbitrary integer $k(> 3)$. Let $E_{n1}, E_{n2}, \cdots, E_{nk}$ form a partition of the sample space such that $E_{ni} \cap E_{nj} = \phi$ where, $i \neq j$ and $\bigcup_{i=1}^{k} E_{ni}$ = sample space. ¸is defined as a class of tests having $P_{\theta i}(E'_{ni}) = \alpha \ \forall \ i = 1, 2, \cdots, k-1$ and the necessary and sufficient conditions of lemma 3.2 reduces to

$$L(\theta_1; \underline{x})/L(\theta_k; \underline{x}) \begin{array}{lll} \leq & k_1, & \text{for} \quad \underline{x} \in E_{nk} \\ > & k_1, & \text{for} \quad \underline{x} \in \bigcup_{i=1}^{k-1} E_{ni} \end{array}$$

$$L(\theta_1; \underline{x})/L(\theta_{k-1}; \underline{x}) \begin{array}{lll} \leq & k_2, & \text{for} \quad \underline{x} \in E_{nk-1} \\ > & k_2, & \text{for} \quad \underline{x} \in \bigcup_{i=1}^{k-2} E_{ni} \end{array}$$

$\cdots$

$$L(\theta_1; \underline{x})/L(\theta_2; \underline{x}) \begin{array}{lll} \leq & k_{k-1}, & \text{for} \quad \underline{x} \in E_{n2} \\ > & k_{k-1}, & \text{for} \quad \underline{x} \in E_{n1} \end{array}$$

and $P_{\theta i}(E'_{ni}) = \alpha \ \ \forall \ \ i = 1, 2, \cdots, k-1$, where $k_1, k_2, \cdots, k_{k-1}$ are real numbers.

Remark 2 : For $k = 2, (E_{n1}, E_{n2})$ form a partition of the sample space such that $E_{n1} \cap E_{n2} = \phi$ and $E_{n1} \cup E_{n2}$ = sample space. If observed value $\underline{x}$ lies in E_{n1}, we accept the hypothesis H_1 (or reject the alternate hypothesis H_2, **acceptance region**) and if $\underline{x}$ lies in E_{n2}, we accept the hypothesis H_2 (or reject the null hypothesis H_1, **critical region**). The necessary and sufficient conditions of lemma 3.2 reduces to

$$L(\theta_1; \underline{x})/L(\theta_2; \underline{x}) \begin{array}{lll} \leq & k, & \text{for} \quad \underline{x} \in E_{n2} \\ > & k, & \text{for} \quad \underline{x} \in E_{n1} \end{array}$$

and $P_{\theta 1}(E'_{n1}) = \alpha$ which turns out to be same as that by NP lemma . This shows that the present work is generalization of NP theory for identifying the correct value of parameter.

4 Some examples

Example 4.1 : Let X be distributed as normal with mean θ and variance 4. Suppose we want to test the hypotheses $H_1 : \theta = -1; H_2 : \theta = 3; H_3 : \theta = 10$ from a sample of size $n = 4$. We consider the case $\alpha = \beta = 0.05$.

Using the procedure discussed above, the test turned out to be

Accept H_1 if $\overline{x} \leq .645$,

Accept H_2 if $0.645 \leq x \leq 4.75$

and Accept H_3 if $4.75 \leq x$.

Example 4.2 : Let X follows Cauchy distribution with location parameter θ and scale parameter one. The Probability density function is given by $f(x,\theta) = \frac{1}{\prod(1+(x-\theta)^2)}$, $-\infty < X < +\infty$. We would like to test the hypotheses $H_1 : \theta = -0.025; H_2 : \theta = 0.060; H_3 : \theta = 0.000$ from a single observation. We consider the case $\alpha = \beta = .05$.

The test turns out to be

Accept H_1 if $-\infty < x < 0$

Accept H_2 if $.037 < x$

and Accept H_3 if $0 \leq x \leq 0.037$.

Example 4.3 : Let X be distributed as normal with mean θ and variance 4. Suppose we want to test the hypotheses $H_1 : \theta = -1; H_2 : \theta = 3; H_3 : \theta = 10$ from a sample of size $n = 4$. We consider the case $\alpha = 0.05$ and $\beta = .06$.

The test turned out to be

Accept H_1 if $\overline{x} \leq .645$

Accept H_2 if $0.645 \leq x \leq 4.604$

and Accept H_3 if $4.6045 \leq x$.

Example 4.4 : Let X follows an exponential distribution with mean $1/\theta$. The probability density function is given by $f(x,\theta) = \theta e^{-(\theta x)}, x > 0, \theta > 0$. Suppose we want to test the hypotheses $H_1 : \theta = 1; H_2 : \theta = 2; H_3 : \theta = 3$ from a single observation we consider the case $\alpha = \beta = 0.05$.

The test turned out to be

Accept H_1 if $X \geq 0.051$

Accept H_2 if $.02 \leq X \leq 0.051$

and Accept H_3 if $X \leq 0.02$.

References

[1] Bechhofer, R.E. (1954). A single sample multiple decision procedure. *Annals of Mathematical Statistics*, 25, 16-39.

[2] Benjamini, Y., and Hochberg Y. (1995). On controlling the false discovery rate: A practical and powerful approach to multiple testing, *Journal of the Royal Statistical Society, Series B (Methodological)*, 57, 125-133.

[3] Duncan, D. B. (1955). On multiple range tests. *Biometrics*, 11, 1-42.

[4] Ferguson, T.S. (1967). *Mathematical Statistics: A Decision Theoretic Approach*. New York: Academic Press.

[5] Gupta, S. S. (1965). On some multiple decision rules. *Technometrics*, 7, 225-245.

[6] Gupta, S.S. and Panchapakesan, S.(1979). *Multiple Decision Procedures : Theory and Methodology of selection*. New York : Wiley.

[7] Hochberg Y. (1988). A sharper Bonferonni procedure for multiple tests of significance. *Biometrika*, 75, 800-803.

[8] Holm, S. (1979). Simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6, 65-70.

[9] Jaccard, J., Becker, M. A. and Wood, G. (1984). Pairwise multiple comparison procedures: A review. *Psychological Bulletin*, 96, 589-596.

[10] Kotz,S., Johnson, N. L. and Read, C.B.(1982). *Encyclopedia of Statistical Sciences* (Nine Volumes). New York : Wiley.

[11] Lehmann, E. L. (1986). T*esting Statistical Hypotheses*, 2nd edition, New York: Wiley.

[12] Lindley, D. V. (1957). A Statistical Paradox. *Biometrika*, 44, 187-192.

[13] Marcus R., Peritz E., Gabriel, K. R. (1976). On closed testing procedures with special reference to ordered analysis of variance, *Biometrika*, 63, 655-660

[14] Ramsey, P. H. (1993). Multiple comparisons of independent means. In L. K. Edwards (Ed.), *Applied analysis*, Vol. 137 (pp. 25-62). Inc, New York, NY: Marcel Dekker.

[15] Shaffer, J. P. (1995). Multiple Hypothesis Testing. *Ann. Rev. Psych.*, 46, 561-584.

[16] Snedecor, G. W. and Cochran, W. G. (1989). *Statistical Methods*, 8th edition. Ames, Iowa: Iowa state University Press.

[17] Wald, A. (1950). *Statistical Decision Functions*. New York: Wiley.