Multidimensional Analysis Using Knowledge Layer for Effective Decision Making

Ayaz Mahmood, Dr. Sohail Asghar
Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST), Islamabad, Pakistan.
Department of Computer Sciences
COMSATS Institute of Information Technology, Islamabad, Pakistan.
ayaztg@gmail.com sohail.asg@gmail.com

Abstract

Business intelligence is one of the most focusing areas for researchers in the last few years for providing various solutions using the concept of Knowledge Discovery. Various researchers have proposed their own models to utilize knowledge discovery for efficient use of information to enable enhanced and effective decision making process. The paper is supplemented with validation of the proposed framework through a case study. Further, the proposed model is visualized through a flowchart for better understanding of the process model. In addition, implementation details are also discussed. Finally, a dashboard is built on the top for the proof of concept.

Keywords: Business Intelligence, Knowledge Discovery, Decision Support System, Data Mining.

1. Introduction

Business Intelligence (BI) has become widely used technology due to rapid and massively increased volumes of data. Organizations are tending to utilize BI as a core source of information for decision making process. Modern BI tools operate on the data layer and decision making is done by different analysis reports. When a new query arises, finding associations between data, for instance, conventional BI tools may not be able to answer such queries. Using massive amount of data in a traditional BI system makes is harder to find hidden patterns in the data.

In order to improve effective decision making using BI systems, we transform data into valuable information and knowledge management practices are executed to handle such information to support decision making. In this paper we introduce the concept of Knowledge Layer to conventional BI system to provide the ability of knowledge-driven decision making. We have implemented Apriori algorithm at Knowledge Layer for generating association rules from database transactions. These rules reveal the hidden patterns and relationship among data sets of database transactions.

In data mining, Apriori algorithm is used for finding relationships between data sets of a transaction. These relationships are known as Association Rules. Support and Confidence are two key concepts in algorithm for searching relationships. Support is describes as the percentage of transactions in a data set which contains the item set. Whereas, Confidence is defined as $\text{conf}(X \rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)}$.

With the assistance of Apriori algorithm implementation and association rules generation, ability of effective decision making of BI systems can be improved while answering new questions.

Apriori algorithm is devised to work on databases which contain data in the form of transactions. The aim of Apriori algorithm is to search relationships and associations between data sets of a database. Usually it is known as “Market Basket Analysis”. Each data set contains item set and is known as a transaction.

The algorithm produces sets of rules which shows how often items exists in a set of data which has gone through under algorithm. Following table illustrates the
concept in a better way. Each line refers to set of items and is called transaction.

<table>
<thead>
<tr>
<th>Bread</th>
<th>Milk</th>
<th>Butter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bread</td>
<td>Milk</td>
<td>Cheese</td>
</tr>
<tr>
<td>Bread</td>
<td>Milk</td>
<td>Egg</td>
</tr>
<tr>
<td>Bread</td>
<td>Milk</td>
<td>Cheese</td>
</tr>
</tbody>
</table>

1. 100% of sets with Bread also contain Milk
2. 25% of sets with Bread, Milk also have Butter
3. 50% of sets with Bread, Milk also have Cheese

Apriori algorithm utilizes breadth-first search and a Hash tree to calculate item sets from a transaction. Algorithm Pseudo code of Apriori for a database transaction $T$, and support $\varepsilon$ is given as follows. Where $C_k$ is the set of candidates for level $k$. $count[c]$ approaches a member of the data structure that corresponds to candidate set $c$, which is in the beginning taken as zero [3]. Diagram below shows the Pseudo code of Apriori algorithm:

```
apriori($T, \varepsilon$)
L_1 \leftarrow \{\text{large 1-itemsets}\}
k \leftarrow 2
\text{while } L_{k-1} \neq \emptyset
C_k \leftarrow \{c|c = a \cup \{b\} \land a \in L_{k-1} \land b \in \bigcup L_{k-1} \land b \notin a\}
\text{for transactions } t \in T
C_t \leftarrow \{c|c \in C_k \land c \subseteq t\}
\text{for candidates } c \in C_t
\text{count}[c] \leftarrow \text{count}[c] + 1
L_k \leftarrow \{c|c \in C_k \land \text{count}[c] \geq \varepsilon\}
k \leftarrow k + 1
\text{return} \bigcup_{k} L_k
```

![Figure1. Pseudo code of Apriori algorithm [13]](image)

In data mining, Association rule is a famous technique to discover hidden relationships between the data sets of transactions of a database or data warehouse. For instance, the rule $\{\text{Bread, Milk}\} \rightarrow \{\text{Egg}\}$ found in the sales data of a shop would show that if a customer buys Bread and Milk together, he or she is likely to buy Eggs as well [3]. This type of information can be precious for the decision making authorities for taking decisions about marketing strategies, e.g. product promotional pricing or placement of products on the shelf of stores.

If we formalize the problem (of finding associations between data sets of a transaction) then:

- **Database Transaction** $T$: set of target transactions, $T = \{t_1, t_2, t_3... t_n\}$
- **Each transaction owns set of items I (item set)**
- **An item set is a combination or collection of items, I = \{i_1, i_2, i_3... i_m\**

$X \rightarrow Y$, is the form of putting relationships as an association rule. Following example makes the concept more clearly about association rules and related concepts.

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>Bread, jelly, peanut-butter</td>
</tr>
<tr>
<td>T2</td>
<td>Bread, peanut-butter</td>
</tr>
<tr>
<td>T3</td>
<td>Bread, milk, peanut-butter</td>
</tr>
<tr>
<td>T4</td>
<td>Cheese, bread</td>
</tr>
<tr>
<td>T5</td>
<td>Cheese, milk</td>
</tr>
</tbody>
</table>

Bread $\rightarrow$ Peanut-Butter
Cheese $\rightarrow$ Bread

Following are the item sets that appear frequently together.

$I = \{\text{Bread, peanut-butter}\}$
$I = \{\text{Cheese, bread}\}$

**Support Count ($\sigma$)** is the Frequency of occurrence if an item set [14].

$\sigma(\{\text{Bread, peanut-butter}\}) = 3$
$\sigma(\{\text{Cheese, bread}\}) = 1$
Support is the ratio of database transactions that owns item sets.

S (\{Bread, peanut-butter\}) = \frac{3}{5}
S (\{Cheese, bread\}) = \frac{1}{5}

The two most used concepts in association rules are:

- **Support (S)** is the occurring frequency of rule i.e. num. of transactions containing both X and Y [14].
  \[ S = \frac{\sigma (X U Y)}{\# of transactions} \]

- **Confidence (C)** is the strength of association which is the measure of Y’s occurrence in a transaction that contains X [14].
  \[ C = \frac{\sigma (X U Y)}{\sigma (X)} \]

Hence the support and confidence of the transactions mentioned in the above table are depicted in the following table.

<table>
<thead>
<tr>
<th>TID</th>
<th>S</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bread (\rightarrow) peanut-butter</td>
<td>0.60</td>
<td>0.75</td>
</tr>
<tr>
<td>peanut-butter (\rightarrow) Bread</td>
<td>0.60</td>
<td>1.00</td>
</tr>
<tr>
<td>Cheese (\rightarrow) bread</td>
<td>0.20</td>
<td>0.50</td>
</tr>
<tr>
<td>peanut-butter (\rightarrow) jelly</td>
<td>0.20</td>
<td>0.33</td>
</tr>
<tr>
<td>jelly (\rightarrow) peanut-butter</td>
<td>0.20</td>
<td>1.00</td>
</tr>
<tr>
<td>jelly (\rightarrow) milk</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

The paper is organized as follows: This section contains introduction. Section II provides an overview of existing Knowledge management techniques, methods and frameworks. Proposed Knowledge Layer based model is provided in section III and section IV presents validation of propose framework through a case study. Implementation and Results interpretation is provided in Section V followed by conclusion and possible dimensions of future research work.

2. **Literature Review**

A number of different techniques and models have been proposed in the past to improve effective and efficient decision making via business intelligence tools.

Yong Feng [1] introduced multi-agent technology in BI systems for efficient decision making. The approach also proposed a low-cost BI framework for the analysis of core components’ function and operation mechanism of the BI system. Proposed framework is based upon the layered architecture which consists of three layers mainly. Data resource layer is the backend layer, which handles all the tasks related to data ranging from data availability to data monitoring. Middle layer is known as core function layer, which consists of agent programs to perform functions like handling requests from users, process it and send results back to user. User management agent and OLAP agent are the main programs which performs respective functions. Front end layer is known as UI-Layer, which deals with the data representation to the prospective users or clients. Proposed solution is effective for low-cost BI system; however, it lacks the systematic implementation of the proposed framework. Authors only discussed the framework in detail, no implementation has carried out to support the idea presented in the paper.

Qing-sheng [2] proposed a Rosetta Net framework for business intelligence systems. Framework helps in realization of optimize business process. It also provides a cost effective BI solution. Proposed architecture claims seamless integration BI systems based upon Rosetta Net Implementation framework (RNIF). RNIF mainly has been developed with respect to Network environment as it deals with network protocols like HTTP, SMTP, TCP/IP and other transfer protocols. RNIF’s core component is RosettaNet Business Message Component. The component mainly consist of two parts, one is Header, which contains metadata information attached to the actual message. Second is Service Content which contains the actual message. RosettaNet utilizes XML as a data source for the information exchange. For analytic purposes, RNIF utilizes Workflow messaging to interact with different business processes and workflows. The proposed approach is good to work with business workflows and processes, also with the integration of BI systems. Though, the proposed model needs to incorporate variety of data sources other than XML data source. This will enhance the flexibility and scalability of the proposed approach.

SHAN Wei [3] proposed knowledge mining model based upon Shannon Information Theory and Bayesian
estimation algorithm for generating association rules. The paper also discusses the knowledge evaluation system of the BI system. It helps in solving the data consistency and redundancy problem during the data acquisition stage in a BI system. Overall paper discussed knowledge mining in a BI system, solution of data quality problems were proposed by introducing the use of Bayesian estimation. Although, paper discusses the conceptual model and development, however, any concrete implementation on a real world scenario is missing to support the proposed architecture of knowledge mining technique in a BI system.

Tong Gang [4] introduced the idea of business process and knowledge management to the traditional BI system. The approach utilizes the implementation of case-based and rules-based reasoning technology. It also provides a knowledge management approach in BI systems.

Luan Ou [5] proposed a customized model of BI system for the retail industry. Proposed framework utilizes JAVA platform to develop custom applications based upon the different data sources. A custom software application for BI system for decision making has been developed for the proof of concept purpose. This system helps in profitability analysis, outside environment analysis, and KPI management, etc. The methodology is good for small scale applications but not feasible for big data enterprises.

Li Dalin [6] discusses design and development of a multi agent based BI system (MABBI). Proposed approach utilizes Database, Data Warehouse and Knowledge Discovery / Data Mining modules as main components of the model. MABBI uses agent technology to deal problems in the BI system.

Alexander Loebbert [7] discusses role of knowledge discovery in a web environment for BI system of retail business. It also proposed a model for web data mining based upon Euclidean distance, Mahalanobis distance and City block distance. Classification model technique is used in the approach to distribute data to different groups.

SHI Changqiong [8] proposed an integration framework for BI systems based on Ontology. In this research, a service-resource based ontology is proposed which includes Ontology mode and Data model patterns.

Xu Xi [9] proposed a BI system model based on Web-Services. The design of the overall framework is divided into three parts which are BI processing system (based on Web Services), commercial service distributors and BI services registration centre which are the end users.

A limitation observed during literature review is that these techniques proposed models which usually are based on traditional BI system architecture. These systems can answer the adhoc queries, manages KPI, provides information, etc but are unable to find the hidden patterns and relationships among data sets of the transactions. For instance, a user wants to see the relationship between items sold over the period of time. There is no such methodology proposed earlier for answering such queries.

3. Knowledge Layer Model

To handle aforementioned problem, we propose a multi-dimensional decision making model using knowledge layer. The proposed model is divided into four main layers as follow:

1. Data Layer
2. Knowledge Layer
3. Transformation Layer
4. Presentation Layer

Data Layer (DL): This layer directly deals with the data. This layer contains heterogeneous data sources such as flat files, xml data files, spreadsheets, enterprise resource planning system, or multiple operational databases such as MY SQL, Oracle, MS SQL Server. These components may serve as a standalone data source or they can be combination of hybrid data sources. Moreover, a single data ware house (DWH) could also serve as a data source. We assumed that in case of DWH as a data source, target data which needs to be analyzed, is already uploaded into DWH from operational data sources as mentioned earlier. The first step of our framework is data provisioning that is done by DL. DL supports and manages data extraction from the data sources and passes it to the upper layer.

Knowledge Layer (KL): Deals with the Knowledge management in terms of finding association and relationships between data sets provided by the DL. KL controls the execution of Apriori Algorithm and generates association rules. Target data which is extracted and provided by DL gets passed to KL. These transactions are then input into Apriori algorithm for
data mining. Apriori algorithm then generates association rules based upon the supplied transactions. Moreover, Apriori needs minimum ‘Support’ and ‘Confidence’ to be entered as Inputs for generation of association rules in the algorithm. Based upon these inputs Apriori produces association rules. Resultant rules contain the relationships between item sets of transactions. Later, these rules serve as input to the next layer.

**Transformation Layer (TL):** Supports and manages the data transformations. Since association rules in KL are in a specific format so they need to be transformed into required format for the analysis. For instance, \{C \rightarrow L = 100\%\} is an association rule. The output rule generated by Apriori algorithm is not clear enough to get the actual interpretation. Hence, to comprehend the rules, a utility is required to interpret association rules. We have implemented a utility program in JAVA which takes the association rules as input and interprets them accordingly based upon transformation logics. Therefore, the rule \{C \rightarrow L = 100\%\} gets transformed into \{Cotton \rightarrow Lawn = 100\%\} which is easier to understand.

**Presentation Layer (PL):** Displays required information to the users via OLAP tools such as adhoc analysis, dashboards, formatted reports etc. PL gets transformed rules from TL. Transformed rules are processed on this layer for presentation. This layer only
deals with displaying information to user. Later, user can perform multi-dimensional analysis upon the processed target data for effective decision making.

**Figure 3 Flow Chart of process for proposed model.**

Figure shows the process flow of the proposed framework. It starts with the extraction of data from DWH. Data extraction is performed on Extraction, Transformation and Loading (ETL) process. For the ease of understanding we name it target data. Once extracted, target data is passed to Apriori algorithm. Two input parameters i.e., Support and Confidence are also passed. Algorithms get executed and process the transactions from target data and generate association rules. The generated rules are in a semi formal format. These rules then undergo a transformation process for converting them into required format. After transformation, these rules get available for OLAP tools where user can perform multi-dimensional analysis which helps in effective decision making.

### 4. Case Study

We apply our proposed model on a case study based on SAP Business One (B1) ERP.

**Problem Statement:** A company Leisure Club (LC) is maintaining its daily transactional data in SAP BI—an ERP product by SAP. LC is a leading clothing brand in Pakistan. LC has implemented the SAP Business Warehouse (BW) module of SAP ERP for maintaining its DWH. They have implemented SAP Business Object (BO) to meet Business Intelligence (BI) demands. The top management of LC is interested to know the relationship between different cloth types during their sales of past two years, in order to develop better marketing and production strategies. Unfortunately, current OLAP analysis implementation is unable to answer such queries. We have implemented the proposed model in different tasks as follow:

**Task 1: Data Extraction.** Based upon the requirements of senior management we have extracted sales data for past two years that is from 2009–2011. Data gets extracted from the SAP BW using ETL process and finally loaded into a text file. Text file contains sales data transactions for the required period of time. Regarding data to be processed, we took the different cloth types such as Chiffon, Silk, Cotton, Malmal, Swiss Lawn, Twill and Organza etc.

**Task 2: Apriori Algorithm Implementation.** We have implemented Apriori algorithm using JAVA programming language. It takes the transactions from the text file and processes them for the rules generation. We pass on transactions to the interface of program and then enter two input parameters i.e. Support and Confidence. Apriori Implementation executes the algorithm and show results in form of association rules. Our implementation takes items of the transactions as alphabets, for instance, “C” represents Cotton, “c” shows Chiffon, “T” points at Twill, “L” for Lawn and so on.

**Figure 4 Prototype interface of Apriori algo.**

We took 400 transactions from the text file and input them in interface. Interface allows add, edit and delete transactions if entered by mistake. Similarly, items of the transactions can also be add or delete from interface. After inserting the items and transactions, support and confidence parameters need to be filled and then press solve for association rules generation.
Output interface shows Frequent Items and Strong Rules based upon the support and confidence entered previously. “Frequent Items” calculates and shows the support of items and their combination whereas, “Strong Rules” shows Association between items and their confidence. We got different association rules as a result such as \{T \rightarrow C = 50\%\}, \{T \rightarrow c = 50\%\}, \{T \rightarrow L = 75\%\} and so forth. These rules are generated on relationship between the items of each transaction. Ideally we are in search of the rules whose confidence is 100% but for the multi-dimensional analysis purposes we need all these rules to be displayed at OLAP analysis.

**Task 3: Rules Transformations.** Rules generated by Apriori program are in semi formal format. To convert them into required format for OLAP analysis tools, we have implemented a utility program in JAVA programming language. This program converts each association rule into the required format. For instance, \{T \rightarrow C = 50\%\} gets converted into \{Twill \rightarrow Cotton = 50\%\}, \{L \rightarrow C = 100\%\} becomes \{Lawn \rightarrow Cotton = 100\%\} and so on. This converted form is now easy to understand for analysis tools at presentation layer.

**Task 4: Multidimensional Analysis.** Finally, the transformed rules are sent to presentation layer for analysis purposes. User can perform analysis in many ways; such as, adhoc analysis, dashboards, and formatted reports. In our case study, we utilized the SAP BO platform for the analysis purpose. SAP BO is a combination of different tools for multidimensional analysis and reporting. Under the SAP BO platform, adhoc analysis is done by SAP Web Intelligence (Webi) tool, dashboards are designed in SAP Dashboard Designer tools and formatted reports are created in SAP Crystal Reports.

We have implemented the results in SAP dashboard designer tool by creating a dashboard on top of transformed rules. Dashboard allows users to select and evaluate different cloth types and analyze their confidence. For instance, user can select cotton and silk and see their association during the last two years. Based upon this analysis user can do effective decision making.

**Interpretation**

Dashboard is used to interpret and evaluate the results of association rules. For better understanding of analysis the rules needs to be interpreted. For Instance, \{Twill \rightarrow Cotton = 50\%\} shows that chances of selling twill with cotton were 50% during the last two years according to processed transactions. Similarly, \{L \rightarrow C = 100\%\} depicts that chances of lawn to be sold in combination with cotton were 100%. Now while looking at these two rules, management can decide that there is a need to produce more lawn and cotton (Production Policy) and place them on selling shelf consecutively (Marketing Strategy). Apparently these are just some processed rules; however, these rules can deduce new policies and strategies as explained above. In this way, senior management can do effective decision making from multidimensional analysis using knowledge layer.

**Implementation**

As mentioned earlier, KL is responsible for implementing Apriori algorithm and we have implemented it in JAVA. Apriori Implementation consists of:

**Class AprioriCalculation:** Generates Apriori item sets from transactions. Class consists of following methods:
aprioriProcess(): used to generate the Apriori item sets.

do
{
    //increase the itemset that is being looked at
    itemsetNumber++;

    //generate the candidates
    generateCandidates(itemsetNumber);

    //determine and display frequent itemsets
    calculateFrequentItemsets(itemsetNumber);
    if(candidates.size() == 0)
    {
        System.out.println("Frequent " + itemsetNumber + "-itemsets");
        System.out.println(candidates);
    }
    //if there are <=1 frequent items, then its the end.
    //This prevents reading through the database again. When there is
    //only one frequent itemset.
    while(candidates.size() > 1);
}

generateCandidates(): Generate all possible
candidates for the n-th item sets

private void generateCandidates(int n)
{
    Vector<String> tempCandidates = new Vector<String>();
    String str1, str2;
    String tokenizer = new Stringtokenizer;
    if(n==1)
    {
        for(int i=0; i<numItems; i++)
        {
            tempCandidates.add(Integer.toString(i));
        }
    }
    else if(n==2) //second itemset is just all combinations
    //of itemset 1
    {
        //add each itemset from the previous frequent
        //itemsets together
        for(int i=0; i<candidates.size(); i++)
        {
            str1 = new Stringtokenizer(candidates.elementAt(i));
            str1 = str1.nextToken();
            for(int j=i+1; j<candidates.size(); j++)
            {
                str2 = new Stringtokenizer(candidates.elementAt(j));
                str2 = str2.nextToken();
                tempCandidates.add(str1 + " + " + str2);
            }
        }
    }
    else
    {
        candidates.clear();
        candidates = new Vector<String>(tempCandidates);
        tempCandidates.clear();
    }

Transformation Utility

This program is used to transform association rules into
required format

    if (itemset.contains('c')){
        newItem = Replace(item).equalto("Chiffon");
    }

    elseif (itemset.contains('C')){
        newItem = Replace(item).equalto("Cotton");
    }

    elseif (itemset.contains('L')){
        newItem = Replace(item).equalto("Lawn");
    }

    elseif (itemset.contains('S')){
        newItem = Replace(item).equalto("Silk");
    }

Conclusion and Future Work

Our proposed model can be effective for
multidimensional analysis which supports improved
decision making to the decision authorities in an
organization. Utilizing knowledge layer with the
traditional BI architecture helps in discovering the
hidden but useful information that lies within the data
in a database or DWH. Thus, our proposed framework
is capable enough to deliver the real insights of data
which is difficult to capture with conventional
architecture of BI systems.

Our proposed model provides effective decision
making using knowledge layer. However, proposed
approach is limited only to find relationships i.e.
generating association rules from data. It needs to incorporate other data mining or knowledge discovery techniques as well.

In future, we intend to extend and enhance the proposed architecture to include Clustering and Classification techniques of data mining for better decision making and knowledge discovery.

REFERENCES


