Cusual analysis of data using 2-layerd Spherical Self-Organizing Map

Gen Niina, Kazuhiro Muramatsu, and Hiroshi Dozono

Faculty of Science and Engineering, Saga University,
1-Honjyo Saga 840-8502 JAPAN

hiro@dna.ec.saga-u.ac.jp

Abstract— **Today, we are able to easily obtain data such as stock prices and market information through media such as the Internet. However, it is not so easy to come by useful information from this data. The reason for this is that there is a mix of many different kinds of information. Such a situation leads to difficulty in grasping trends in the data through just one rule, and so one must find a rule for each relevant factor. Therefore, the need for finding relevant factors is inevitable. For this reason, in our research we developed an algorithm that enables one to extract factors that have causal relationships with one another, and indicated its usefulness through experiments**.

Keywords: Self-organizing-map, Spherical SOM, Casual learning, neural network

I. INTRODUCTION

Various items can be stored as computerized data, which analysts then attempt to use to create new value. For example, individual businesses use the large-scale data that they maintain, such as customer and purchaser data, in marketing activities such as purchase estimates and new product development. However, data analysis can often be skewed by the noise from inconsequential, unnecessary components included in the original data. On the other hand, when analysts adhere too slavishly to past experience and general knowledge when attempting to identify causal relationships and thereby limit their sources of information too severely, they tend to limit the possibility of gleaning new knowledge from their data[1]. Therefore, in data analysis, one must first single out the elements that involve causal relationships. When choosing a model for the object of analysis, one should study the differences between the elements that belong to the model of the object and those that belong to a model not of the object. However, if one's goal is to study the entire field of data and discover new characteristics without having decided on a model for the object of analysis, the elements are too disparate spatially, and it is extremely difficult to select characteristics between elements. Therefore, in this study, we developed an algorithm that can extract elements with causal relationships using a self-organizing map.

II. SELF-ORGANIZING MAPS

Considering-Causation-Spherical Self-Organizing-map

To avoid heterogeneous learning of two-dimensional self-organizing maps (Plane-SOMs; hereafter PSOMs), we converted the Plane-SOMs to spherical self-organizing maps (Spherical-SOMs; hereafter SSOMs—Figure 1) for the purposes of this study.

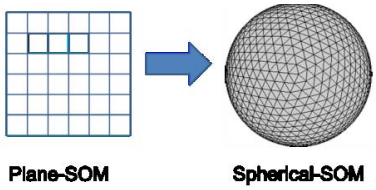

Fig. 1: PSOM vs. SSOM

The Considering-Causation-SOM developed in this study is a Spherical-SOM[2] [3]virtually constructed of more than 2 layers (n≧2)(Fig.2), the uppermost layer being the first layer, the next deepest layer being the second layer, and so on down to the nth layer.
Data whose element characteristics are to be analyzed is used as input for a Considering-Causation-SOM.

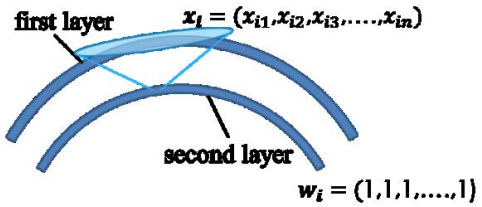

Fig. 2: structure of layer

Also, a reception field for the second layer is present in the first layer, with a width of a solid angle of 30 degrees. This is established as exactly five times the size of the final neighboring area. If the field is too large, noise will negatively impact the study; if it is too small, it will be insufficient to study the causal relationships between elements.

CSCI-ISCI: LATE BREAKING PAPER

978-1-4673-9795-7/15 $31.00 © 2015 IEEE
DOI 10.1109/CSCI.2015.106

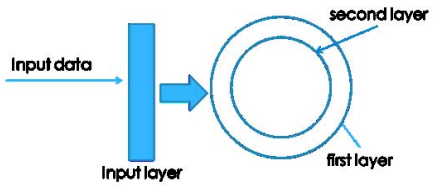

Fig. 3: The learning model structure

The inputted data is not directly learned, but mediated through the input layer. The input layer nodes are filtered with input data elements, and controlled so that unnecessary elements are not inputted.

The learning algorithm is below.

Considering-Causation-SOM's Learning Algorithm

1 Initialization

The elements of the vectors $\boldsymbol{x_i}$ of the first layer are assigned random values on initiation, and the mask vectors $\boldsymbol{w_i}$ of the second layer are all set at 1. In input layer, all elements of the casual-vector $\boldsymbol{h_a}$ are also set at value of 1.

2 Finding the winner nodes

In Formula (1), the node i with the smallest value is set as the Winner node for input vector $\boldsymbol{v_a} = (v_{a,1}, v_{a,2}, \dots, v_{a,j}, \dots, v_{a,n})$.

$$\underset{i}{argmin}\frac{1}{L}\sqrt{\sum_j\{w_{i,j} * (h_{a,j} * v_{a,j} - w_{i,j} * x_{i,j})\}^2} \qquad (1)$$

Here L is calculated with following Formula (2).

$$L = \sqrt{\sum_i {w_{i,j}}^2} \qquad (2)$$

3 Calculations

If there are any other Winner nodes in the extent of the reception field for the node on the second layer that is directly below the Winner node (vector $\boldsymbol{w_i}$), the average value of the distance between the initial Winner node and the other Winner nodes is stored in vector $\boldsymbol{\mu_a}$.

All elements $\boldsymbol{\mu_{ai}}$ of vector $\boldsymbol{\mu_a}$ are then reordered from smallest to largest and represented in vector $\boldsymbol{\omega}$.

$$\boldsymbol{\mu_a} = (\mu_{a1}, \mu_{a2}, \dots, \mu_{ai}, \dots, \mu_{an})$$
$$\boldsymbol{\omega} = (\omega_1, \omega_2, \dots \omega_m, \dots, \omega_n)$$

In order to find a minimum value m to satisfy Formula (3), updates are conducted using Formula (4) updated for elements of $\boldsymbol{\mu_a}$ that correspond to each element of $\boldsymbol{\omega}$ from 0 to m. Formula (5) is followed for the other elements.

$$\underset{m}{argmin}\left\{\frac{1}{m}\sum_{k=2}^{m}\left(\frac{\omega_k}{\omega_m}\right)^2\right\} \le \varphi \qquad (3)$$

where φ is represents the learning range.

when $f(k) \le m$,

$$w_{i,k}^{new} = w_{i,k}^{old} + \gamma\left(1 - w_{i,k}^{old}\right) \qquad (4)$$

when $m < f(k)$,

$$w_{i,k}^{new} = w_{i,k}^{old} + \gamma\left(0 - w_{i,k}^{old}\right) \qquad (5)$$

Here, γ is the learning rate and takes the range 0<γ<1.

If all elements of vectors $\boldsymbol{w_i}$ on the second layer exceed the threshold:

The elements of vectors $\boldsymbol{w_i}$ shall all be set to either 1 or 0 according to their thresholds and transferred to the casual-vector of the winner node.

The study results of the Considering-Causation-SOM are stored in the casual-vector of the winner node, and other elements in that casual-vector that are 1 display some manner of causal relationship.

4 Updating neighboring nodes

The parameters of Winner nodes are updated with Formulas (6).

$$\boldsymbol{x_i^{new}} = \boldsymbol{x_i^{old}} + \alpha\left(\boldsymbol{v_a} - \boldsymbol{x_i^{old}}\right) \qquad (6)$$

Here, α is the learning rate and takes the range 0<α<1.

5 Updating neighboring nodes

Based on the SOM learning algorithm, the parameters of the neighboring nodes on first layer approach the parameters of the Winner nodes $\boldsymbol{x_i}$. On the second layer, the neighboring nodes are adjusted toward to Winner nodes $\boldsymbol{w_i}$.

6 Updating the SOM parameters

As the learning parameters, such as the learning rate and the neighborhood range, are updated in the SOM, steps 2-4 are repeated for the number of times of training. The neighborhood range is designated by its solid angle with the Winner nodes and is made to become smaller with an increase in training (Fig. 4). The learning rate will also become smaller in value as training increases [4][5][6].

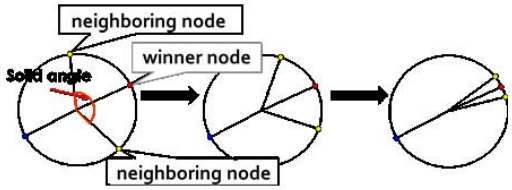

Fig. 4: Neighborhood range

III. Experiments with Artificial Data

In order to make the results easy to visualize, 100 to 150 pieces of data with 3-dimensional elements of x, y, and z were generated for use in the study. An experiment was then conducted that could study the existence of causal relationships between each element using a Considering-Causation-SOM.

3 types of models, detailed below, were prepared to generate the data.

Model 1

A model outputting data with *x* & *y* values with a positive correlation and a random *z* value(Fig. 5).

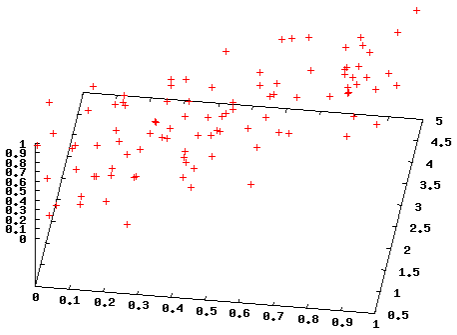

Fig. 5: Model 1

Model 2

A mixed model with a model outputting data with *x* & *y* values with a positive correlation and a model outputting data with *x* & *y* values with a negative correlation(Fig. 6).

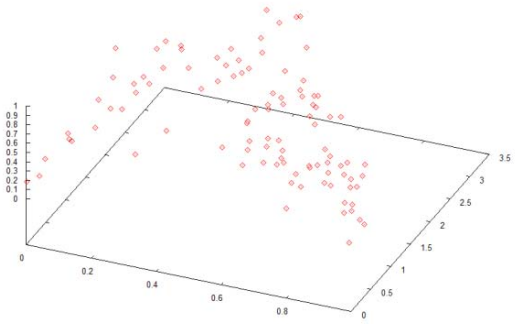

Fig. 6: Model 2

Model 3

A model outputting data with *x*, *y*, and *z* values with a positive correlation.

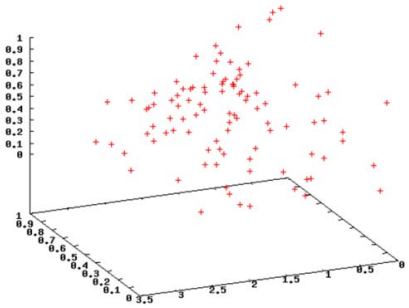

Fig. 7: Model 3

IV. Experimental Results

Model 1

Figure 8 describes the results of model 1, based on existing algorithms for finding winner nodes. The experiment found that most of the data from model 1 (the red points) had no causal relationship for the x, y, and z components. The other points were classified as having some causal relationship of components, but it is believed that this represents the effects of noise on the experimental data.

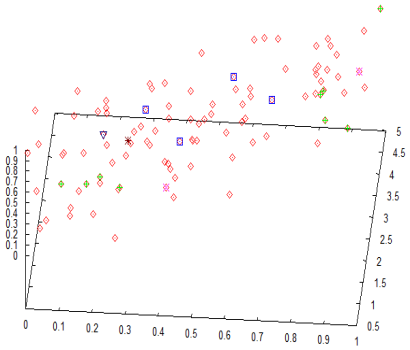

Fig. 8: Results of the Model 1 using conventional updating method

While on proposed method, accuracy in model 1 was confirmed to be 100% (green color represent the causal relationship between x and y (Fig. 9)). It can be stated that the uncorrelated z element was excluded from the mask vectors during the study, permitting an analysis of the relevant

elements. The actual correlation coefficient was about 0.819. The causal relationship between x and y was also able to be studied appropriately with this learning algorithm. The study accuracy of the Considering-Causation-SOM was found to be high.

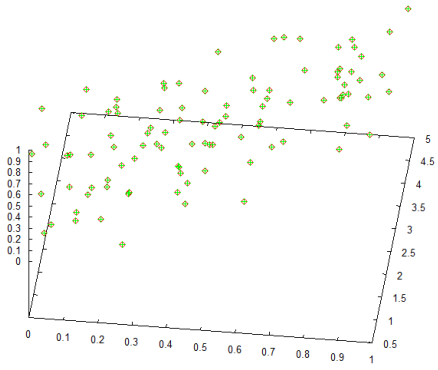

Fig. 9: Results of the Model 1 using the Considering-Causation-SOM's updating method

Model 2

As with model 1, Figure 10 showed that existing algorithms did not derive any causal relationship between the data points.

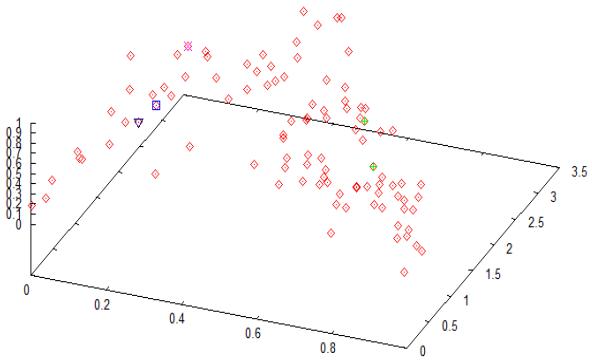

Fig. 10: Results of the Model 2 using the conventional updating method

On the proposed method, accuracy for model 2 was confirmed to be about 95% (Fig. 11). Blue color represents the causal relationship between x ,y and z. Green color represents existence of the causal relationship between x and y.

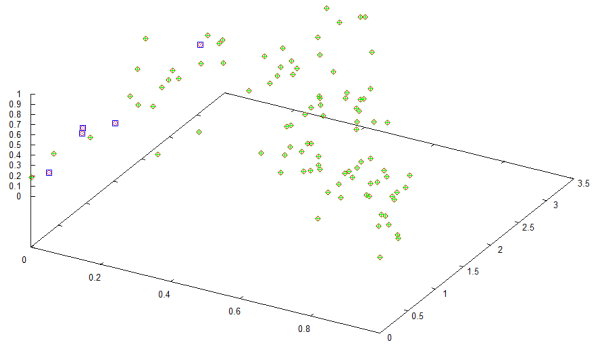

Fig. 11: Results of the Model 2 using the Considering-Causation-SOM's updating method

Model 3

Figure 12 showed the result using conventional updating algorithm. The data could not be classified in terms of casual relationships for the x, y, and z components, with noise affecting the experimental data.

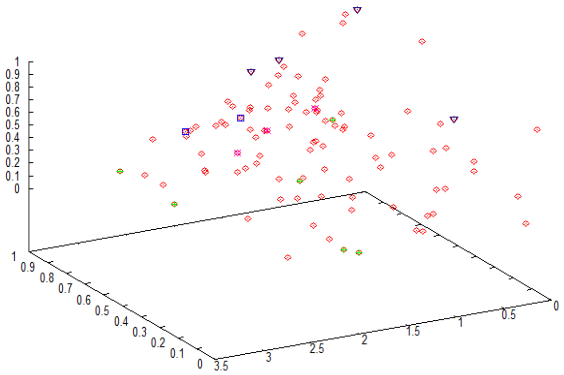

Fig. 12: Results of the Model 2 using conventional updating method

On the Considering-Causation-SOM's updating method,

From examining the study results (Fig. 13, Fig. 14), with data where x & y should have had a correlation, data points where y & z demonstrated a correlation were also included. A clean division of the data into two groups—a group with an x value of above 0.5 and a group with an x value of below 0.5—would perhaps have been the ideal result.

However, that did not occur in the results of the experiment.

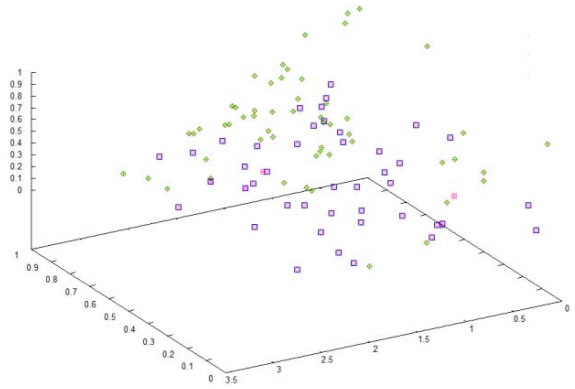

Fig. 13: Results of the Model 3 Considering-Causation-SOM's updating method

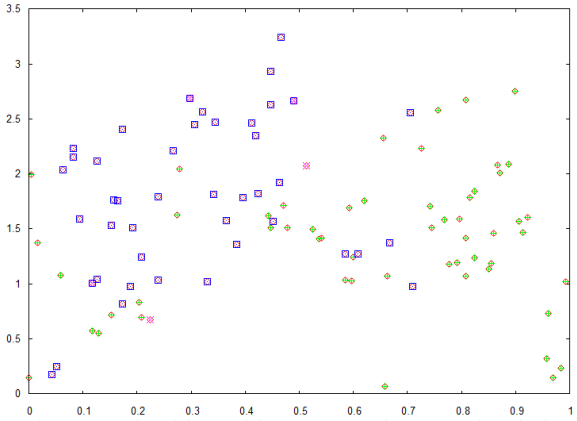

Fig. 14: Results of the Model 3 (x-y) using the proposed method

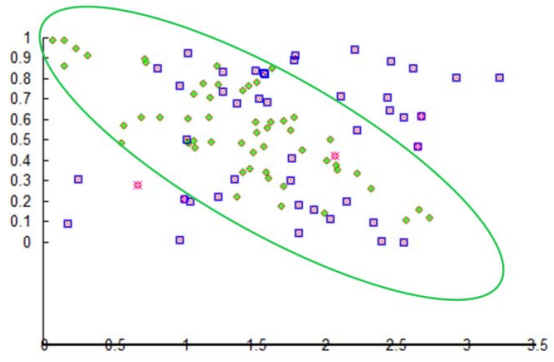

Fig. 15: Results of the Model 3 (y-z) using the proposed method

On Figure 14, the horizontal axis indicates the x-axis, while the vertical axis indicates the y-axis.
The x values of 0.5 and below (left of center) are gatherings of data that correlate both to x and y, with some portion of data mistakenly interpreted to correlate to y and z having been mixed in, but as there is an 78% level of accuracy in learning, it can be said that the overall distinct characteristic distribution of data has been grasped.

Figure 15 indicates these classification results on a y/z plane.
The aforementioned misinterpreted data of y and z correlation can be seen embedded here as the cluster of data (inside the green oval) in this chart.
It can be inferred from these results that, through excessive learning during the process of learning the data, the data that has lost the x-component information could not maintain its positional relationship on the map, and was affected by the learning of other data.

V. Conclusions

Learning was properly conducted in regards to data without model intermingling, but in cases where there were model intermingling, the other data learning were affected, resulting in a slight lowering of the accuracy of learning.
The positional relationship of the data has been saved on the first layer of the map, and it is thought that by conducting learning while appropriately referring to the classification results of the first layer, the accuracy of learning can be improved. Existing methodologies failed to provide means of classifying data. Experiments using a new modality are currently underway, but it is not accurate in terms of its ability to classify data. Going forward, setting forbidden terms in order to increase fidelity and thereby assess data such that values are not affected by noise during the classifying process will be considered.

REFERENCES

1. Chie Morita and Hiroshi Tsukimoto: Knowledge discovery from numerical data, Knowledge-based Systems, Vol.10, No.7, pp. 413-419, 1998.

2. N. Yamaguchi: Self Organizing Hidden Markov Models, Neural Information Processing,
Models and Application, Proc. of 17th International Conference, ICONIP 2010, LNCS6444, Springer (2010)

3. Nakatsuka, D., and Oyabu, M.: Application of Spherical SOM in Clustering. Proc.Workshop on Self-Organizing Maps (WSOM '03), pp.203-207 (2003)

4. Gen Niina, Hiroshi Dozono: The Spherical Hidden Markov Self Organizing Map for Learning Time Series Data. ICANN (1) 2012, pp.563-570

5. H.Dozono, T.Kabashima,et.al,: Visualization of the Packet Flows using Self Organizing Maps, WSEAS TRANSACTIONS on INFORMATION SCIENCE \& APPLICATIONS, Issue.1 Volume.7, 2010, pp.132-141

6. Jaziri,R., Lebbah, M. and et.al. : SOS-HMM Self Organizing Structure of Hidden Markov Model, Artificial Neural Networks and Machine Learning - ICANN 2011.
LNCS6792, Springer (2011)