A Study of optimized clustering method based on SOM for CRM

Jong T. Rhee[a] and Joon. Lee[b]

[a] *Industrial Engineering Department, Dongguk University*
3-26, Pil-dong, Chung-gu, Seou, 100-715, Korea
Tel: +82-2-2260-3378, Fax: +82-2-2260-3378, E-mail:jtrhee@dongguk.edu

[b] *Industrial Engineering Department, Dongguk University*
3-26, Pil-dong, Chung-gu, Seou, 100-715, Korea
Tel: +82-2-2260-3378, Fax: +82-2-2260-3378, E-mail:capjoon@dongguk.edu

Abstract

CRM(Customer Relationship Management : CRM) is an advanced marketing supporting system which analyze customers' transaction data and classify or target customer groups to effectively increase market share and profit. Many engines were developed to implements the function and those for classification and clustering are considered core ones.

In this study, an improved clustering method based on SOM(Self-Organizing Maps : SOM) is proposed. The proposed clustering method finds the optimal number of clusters so that the effectiveness of clustering is increased. It considers all the data types existing in CRM data warehouses. In particular, an adaptive algorithm where the concepts of degeneration and fusion are applied to find optimal number of clusters. The feasibility and efficiency of the proposed method are demonstrated through simulation with simplified data of customers.

Keywords:

CRM(Customer Relationship Management), Neural Network, Data Mining, Clustering, SOM(Self-Organizing Maps)

1. Introduction

CRM is an effective Marketing Supporting System which enables formulating effective marketing strategy on the basis of the customer's information of dealings. To implement CRM, it is required to go through a series of processes including customer's data collection, data analysis of total dimension to obtain useful information, strategy development and verification from analysis result, survey and evaluation of strategic effect from marketing practice. The engine to implement these functions is called data mining(DM) and various methodologies has been developed to enhance the capability of the engine. Some of the methodologies that are frequently used for the data mining have the capabilities of examining the correlation, typical patterns, and trends existing in massive data via scoring, associating, classifying, clustering and constructing decision trees.

Data mining techniques are divided to those for directed data mining and those for undirected data mining[1]. The directed data mining techniques have clear objects to control such as the classification of customers by the measure of profitability, the estimation of target variables, and the prediction of stock price. The undirected data mining techniques, on the other hand, does not have clear objects for the analyses and the results of analyses make it possible to draw meaningful target variables and useful strategies. They have the functions of clustering entities by similarity, rule extracting from data, explaining and visualizing relations among attributes[1]. The data mining makes use of a advanced statistics technique, artificial intelligence, and neural networks models, fuzzy theory, and genetics algorithm. The clustering is one of the most popular function of data mining engine. It is done by finding multiple groups of entities such that the similarities of entities within a group are very high and the similarities of entities of different groups are low[2,6]. Clustering methods based on the distance concept to examine the similarity are divided into agglomerative methods and partitioning methods[3,6]. With agglomerative methods, each cluster starts with a singleton of entity and grows by agglomerating similar groups. On the other hand, with the partitioning methods, a big cluster of including the whole entities is divided into small ones such that the similarity of partial clusters has higher level of similarities. For the measure of similarity, the euclidean distance is used more often than others[4,6].

The clustering method suggested in this paper is based on the structure of Self-Organizing Maps(SOM)[5]. Self-Organizing Maps is one of neural network models invented by T. Kohonen, in 1984, has the capability of finding prototypes of groups adaptively from entities whose distribution is unknown. Self-Organizing Map constitutes of two layers, the input layer and the competition layer. Nodes of the competition layer responses to the input vector given to the nodes of input layer by "Winner Take it all" concept. The only node which wins the competition is called the winner node. The winner node and neighbor nodes experience the learning, that is, the connecting weights between the node and input nodes are modified. In each of iteration of learning, an input vector is given to cause the learning, and the process goes until connecting weights does not change significantly.

The clustering process constitutes of two steps, the first step is just the basic clustering process and the second step is to optimize the number of clusters. In the second step, the policies of degeneration and fusion are applied in the process of competitive learning.

2. Clustering Analysis

The clustering analysis is the technique to find separate groups from multi-variate data according to the similarity of attributes. That is, the entities are combined into similar data cluster where the center of clusters are unknown but the measure of similarity is given before the learning[7].

The difference of clustering and classifying is that with the classifying, the number and prototypes of clusters are given and each entity is just assigned to one of the clusters, but with the clustering, the number or prototypes of clusters are not given but obtained from the process of learning[9]. The criteria of clustering is the similarity or proximity of attributes of entities. One of difficulties in clustering analysis is that the types of group can be different depending on the entity distribution as in the following figure.

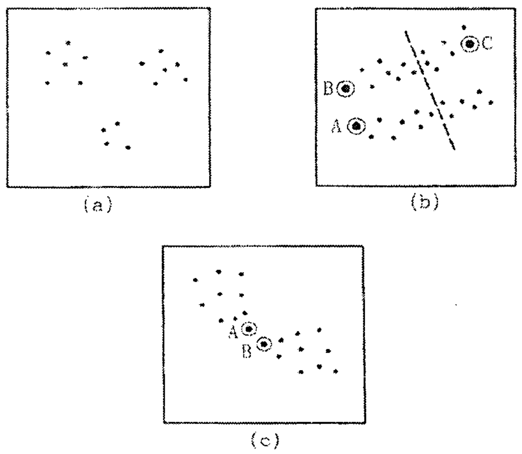

Figure 1 - Cluster from different types of entities distribution

3. Self-Organizing Maps(SOM)

SOM is one of neural network models that combines entities into separate groups, finds the prototypes and saves them into the connection weights between competition layer and input layer. That is each weight vector between one node of competition layer and nodes of input nodes reserves a prototype of a cluster.

By the functioning mechanism of SOM, in each iteration, the attribute vector of an entity is selected randomly and given to the input nodes and the node in the competition layer, whose weight vector from input nodes is the most similar with the input vector by the euclidean distance concept, responds to the input vector. The node is called the winner node and the weight vectors of the winner node and the neighbor nodes are modified such that each weight vector of them moves to the direction of input vector, where the changing occurs most significantly with the winner node.

Here, the neighbor relation is predetermined before learning and never changed. The neighbor relation, as shown in Figure 2, is determined by the relative location of nodes in competition layer.

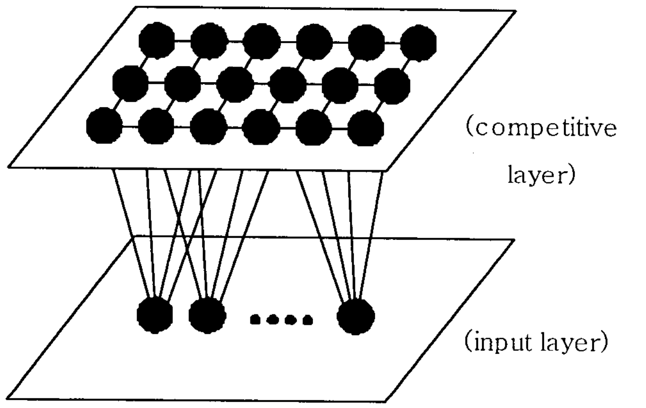

Figure 2 - SOM structure

The learning algorithm of SOM with N attributes of entities, M nodes in competition layer is described below.

- **Step 1. Initialize weight vectors of competition layer**
  - Initialize M weight vectors of competition layer nodes from N input nodes with small values selected randomly.
  - The neighbor range starts with a large size enough to cover all the competition layer nodes from any winner nodes and gradually decreases as the learning proceeds.
- **Step 2. Select an entity randomly and present the attribute values as the input $X_i(i = 1, 2, ..., N)$ to the input layer nodes.**
- **Step 3. Determine the winner node.**
  - For each node j of competition layer, compute the distance between the weight

vector of the node and the input vector by the following equation.

dj = {(X1 - W1j)2 + (X2 - W2j)2 + ··· (XN - WNj)2}½ where Wij(j = 1, 2, ..., N) is weight vector of node j of competition layer and node i of input layer.

- Find the winner node j^* corresponding to the minimum distance.

- **Step 4. Modify weight vectors**
  - Modify the weight vectors of winner node and nodes within the neighbor range as follows:

  $W_{ij} = W_{ij} + \alpha\ M(j^*, j)(X_j - W_{ij})$,

  - where j indexes one of such nodes, α is a learning coefficient which takes a value between 0 and 1, and $M(j^*, j)$ is called a mexican hat function whose value depends on the neighbor distance between node j and the winner node j^*, as shown in Figure 3. If node j is the winner node, this value is 1. Namely,

  $M(j^*, j^*) = 1.$

  The learning rate α decreases as the learning proceeds.

- **Step 5. Exit or continue**
  - If every weight vector changes by only tiny amount, stop the iteration(Learning is completed). Otherwise, go on to step 2.

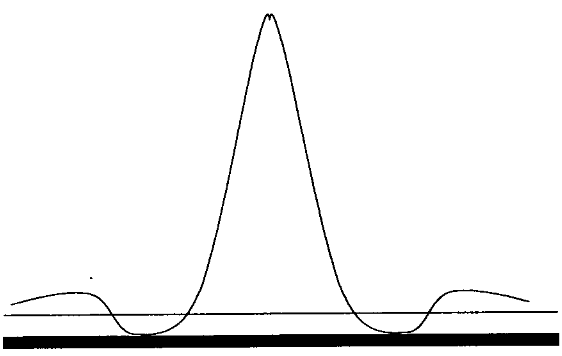

Figure 3 - The Mexican Hat function with the maximum value of 1.

$$\psi(x) = \left(\frac{2}{\sqrt{3}}\pi^{-1/4}\right)\left(1 - x^2\right)e^{-x^2/2} \qquad (1)$$

(The x axis represents the distance between the neighbor node and the winner node)

The Mexican Hat function shown in Figure 3 determines the moving direction of each weight vector of node of competition layer from nodes of input layer so that the weight vectors of nodes of competition layer become located according to the distribution of entities in the attributes space. Each weight vector of competition layer's node represents the prototype of a cluster after the learning is completed.

4. Comparison of SOM and simple competitive learning

In each the iteration of the learning process, described above, not only the winner node's weight vector but also the neighbor nodes' weight vectors are changed. If the process of changing the neighbor nodes' weight vectors is skipped, the learning procedure makes just the simple competitive learning.

With the simple competitive learning, entities can be grouped. But in this case, the sequential implication of clusters disappears and the effectiveness of learning deceased. That is, under the SOM structure, the neighbor nodes of competition layer become to have similar weight vectors after learning, and by this feature, weight vectors of sequential nodes can get a sequential interpretation[8].

For example, if one node represents a group of low profitability, then the closest neighbor node might represent a group of higher profitability and the next closest neighbor node might represent a group of very high level of profitability. Such effect helps the interpretation of the characteristics of clusters obtained by the result of learning and is may not be expected with the simple competitive learning.

Another advantage of SOM is that during learning, most of the nodes of competition layer becomes to participates the process of learning but with the simple competitive learning, more often that not, some nodes can never participates and dropped out of learning process. This happens in the case that the starting position of a node's weight vector is in the "wilderness area" where not an entity's attribute vector is located nearby and the attribute vectors of entities are distributed more closely to other nodes' weight vectors, so that the node can never have the chance of becoming a winner node ever.

However with SOM, such a node's weight vector is also changed even when the node is not the winner node. And the node can have the chance of becoming the winner node, later. Such an advantage of SOM guarantees the higher effectiveness and appropriateness of clustering.

5. Proposed algorithm for optimizing the number of cluster

From the result of SOM learning process, entities are clustered and the number of clusters have determined by the number of nodes in competition layer.

With classic SOM structure, the number of competition layer nodes is predetermined so that the number of clusters is predetermined, too. The number of clusters, however, should be determined on a reasonable basis to be applied for practical purpose. In this research, the process of optimizing the number of clusters is included in the learning algorithm as follows:

Preparation stage) the competition layer nodes are set up by a large number.

Stage 1) the weight vectors of competition layer nodes are modified by the SOM learning process.

Stage 2) As the learning process converges, some nodes of competition layer are dropped off by the following criteria:

A(Degeneration) : Nodes that have never been winner nodes for long iterations are dropped out.

B(Fusion) : In case multiple nodes of competition layer are very similar, all but one of them are dropped out.

In State 2, the learning process stops when the number of competition layer becomes the predetermined minimum number or there occurs no more nodes being dropped out of the competition layer. The first criterion of Stage 2 is adopted because a node that can never become a winner node cannot represent a cluster, which reflects the biological observation of degeneration of neurons. The second criterion of the Stage 2 is adopted to prevent making twin clusters whose prototypes are almost same and the concept of fusion fits this situation because the twins is merged by this process.

The degeneration function is implemented by redesigning the SOM structure such that there are self connecting weight for each competition layer node.

The self-connecting weight decreases as the learning proceeds and increases only when the node becomes a winner node. The node is dropped out when the weight decrease below a predefined limit. The fusion function is implemented in the learning process such that in an iteration, when a winner node is found, the second winner node whose weight vector is the second most similar to the input vector is also found compared with the winner node. That is, if two nodes have almost same weight vectors, the second winner node is dropped out. This method is more efficient that comparing ever pair of competition layer nodes to check the weight vector similarity. From simulation results, it was observed that the two functions were working together without contradictions, in Stage 2.

6. Simulation result

The suggested learning procedure has been applied to clustering customers with simplified attribute vector consisting of the element of customer's age and the internet-using hour of a day. It is assumes that the minimum number of cluster is limited by 2. From the simulation several prototype of clusters were obtained and each customer could be assigned one of cluster with the most similar prototype vector. In the figures below, the nodes' weight vectors are represented with small squares. During learning, the location of the weight vectors are changed in the attribute space and some nodes are dropped out as shown in the figures.

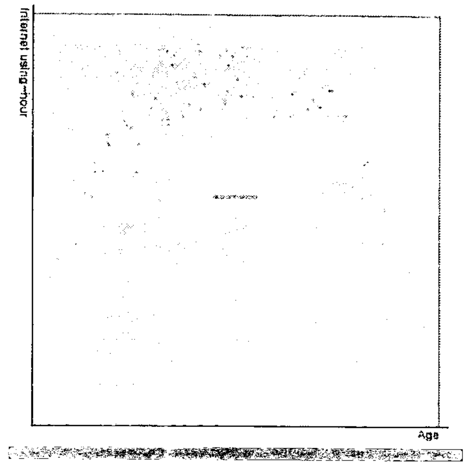

Figure4 - dimension=1, iteration=0

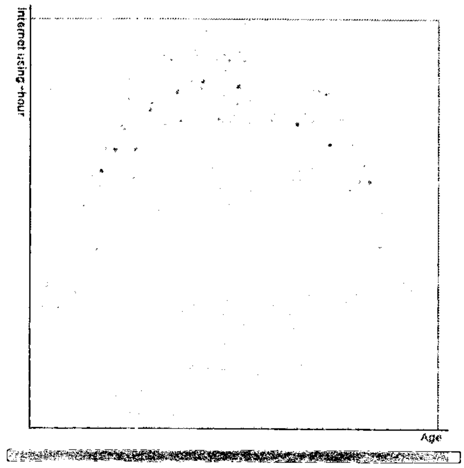

Figure 5 - dimension=1, iteration=2000

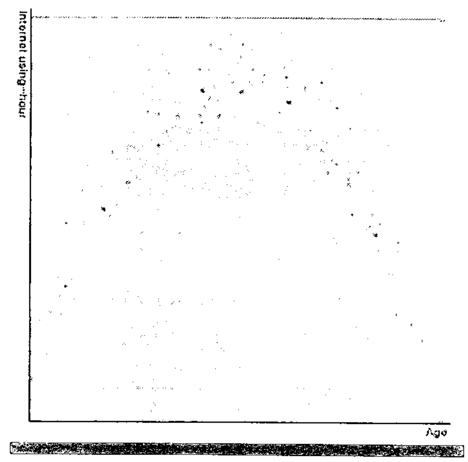

Figure 6. dimension=1, iteration=4000

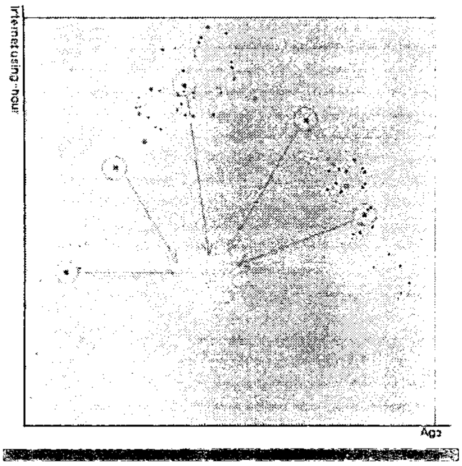

Figure 7. dimension=1, iteration=8000

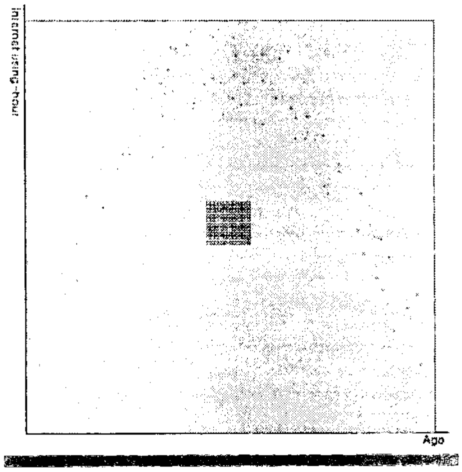

Figure 8. dimension=2, iteration=0

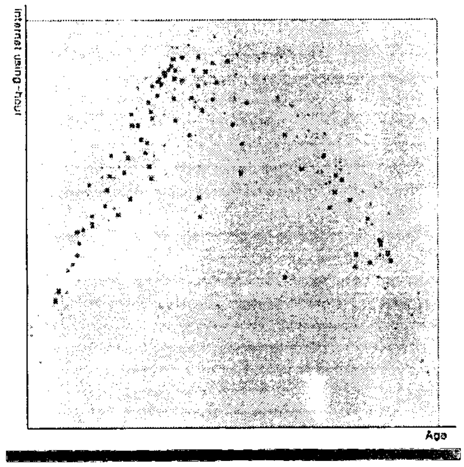

Figure 9. dimension=2, iteration=2000

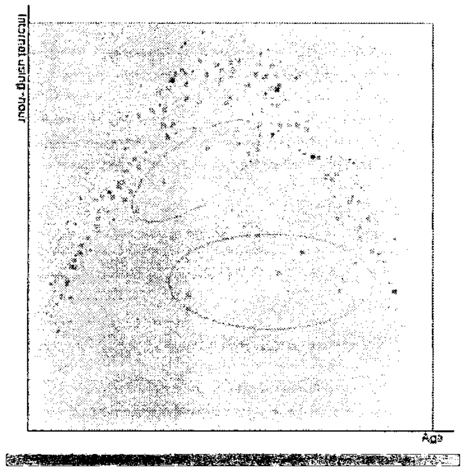

Figure 10. dimension=2, iteration=4000

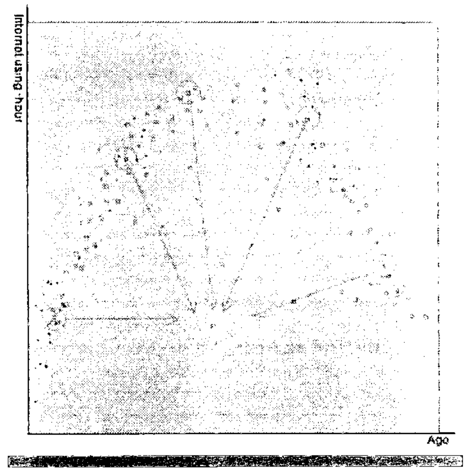

Figure 11. dimension=2, iteration=8000

Most simulations with various values of α yielded a satisfactory result of clustering. The above simulation assumes two attributes of customers but if more attributes are considered, the learning can be very slow. However, it has been found that such a case can be overcome by restricting the number of neighbor nodes participating the learning process in each iteration.

References

[1] Michael J. A. Berry, Gordon S. Linoff, "Mastering Data Mining", John Wiley & Sons. Inc, 1999

[2] G. F. Luger, W. A. Stubblefield, "Artificial Intelligence" chap.12, The Benjamin Cumming Pub. Co., Redwood City 1993

[3] K. C. Gowda, E. Diday, "Symbolic Clustering using a similarity measure", IEEE Trans on system, Man, and Cybernetics, Vol.22, No.2, 1992

[4] K. C. Gowda, T. V. Ravi, "Divisive clustering of symbolic objects using the concepts of both similarity and dissimilarity", Pattern Recognition, Vol.28, No.8,

1995

[5] T. Kohonen, "Self-Organizing Maps", Berlin/Heidelberg, Germany : Springer, Vol.30, 1995

[6] Lee kunmyung, "Fuzzy Clustering of Fuzzy Data using a Dissimilarity Measure", Korea Information Science Society, Vol .26, No.9, 1999

[7] G. J. McLahlan and K. E. Basford, "Mixture Models : Inference and Applications to clustering", New York: Marcel Dekker, 1987

[8] T. Kohonen, "Self-organizing maps : Optimization approaches in Artificial Neural Networks", T. Kohonen, K. Makisara, O. Simula and J. Kangas, Eds, Amsterdam, The Netherlands : Elsevier, 1991

[9] Juha Vesanto and Esa Alhoniemi, "Clustering of the Self-Organizing Map", IEEE Transaction on Neural Networks, Vol.11, No.3, 2000