A Growing Self-Organizing Map for Visualization of Mixed-Type Data

Wei-Shen Tai
Department of Information Management,
National Yunlin University of Science and Technology

Chung-Chian Hsu
Department of Information Management,
National Yunlin University of Science and Technology

Abstract

Large amount of high-dimensional mixed-type data including numeric as well as categorical attributes are commonly seen in corporate databases nowadays. Being able to analyze those data is important for supporting decision making. Visualization is essential in data mining, especially, at the initial stage of data analysis. Self-Organization Map (SOM) provides users an efficient data visualization interface to analyze the characteristics of high-dimensional data on a low-dimensional map. However, most SOMs need to predetermine the size of the map prior to training. Consequently, the resultant map must be constrained in a static, fixed-size map and could not extend with extra neurons in accordance with the nature of the data. Although growing SOM (GSOM) was proposed to tackle the foregoing problem via more flexible structures, GSOM lacks the ability to handle mixed-type data which include numeric as well as categorical attributes. In this study, we propose Growing Mixed SOM (GMixSOM) intending to handle high-dimensional mixed-type data in a map with flexible structure. Experimental results indicate that the proposed model not only can present the topological relationship between mixed-type data but also demonstrate better performances of data clustering compared to the conventional GSOM.

Key words : data mining; data visualization; Self-Organization Map (SOM); mixed-type data;

成長式自組映射圖視覺化混合型資料

戴偉勝
國立雲林科技大學資訊管理學系

許中川
國立雲林科技大學資訊管理學系

摘要

現今企業資料庫中，隨處可見大量包含數值型與類別型屬性的高維度混合型資料。這些資料中常隱含有用資訊，因此如何能有效地分析這些資料從而支援決策，儼然是企業經營管理上的一項重要課題。在探勘資料時，視覺化一直是資料分析初始階段中相當重要的一環。自組映射圖能夠提供一個高效率的資料視覺化介面，讓使用者能於低維度映射圖上分析高維度資料的特徵。然而，對大部分自組映射圖演算法而言，使用者必須在訓練之前先行決定映射圖大小，也因此最終的映射結果會被此預設固定大小的圖形所限制，無法依據資料的本質擴充所需的神經元。雖然已有學者提出具備更彈性化結構的成長式自組映射圖克服前述問題，然而成長式自組映射圖仍然無法有效處理包含數值型與類別型屬性的混合型資料。本研究提出一個成長式混合型自組映射圖架構及訓練演算法，以更彈性的結構圖處理高維度混合型資料。經由實驗結果證實，本研究所提出的方法不但可表現混合型資料的拓撲關係，更可於資料分群上表現出較傳統自組映射圖更好的績效。

關鍵字：資料探勘、資料視覺化、自組映射圖、混合型資料

1. INTRODUCTION

Nowadays, data mining has been regarded as an efficient tool for business analysts to explore and analyze large amount of data from corporate databases. In business, the knowledge mined from the data has been demonstrated that they can be beneficial for applications ranging from market analysis, fraud detection, and customer retention, to production control and science exploration (Dunham, 2003; Fayyad et al. 1996; Han et al. 2006). Nevertheless, most transactional data consist of a variety of numeric and categorical attributes in reality. It is difficult for human to explore and extract valuable information from mass high-dimensional mixed-type data. In order to help analysts process those data and extract knowledge to support decision making efficiently. It has become an important issue in data mining research to provide analysts an appropriate data visualization solution to handle high-dimensional mixed-type data (Cesario et al. 2007; Chakrabarti et al. 2009; Kaban et al. 2001; Kosara et al. 2006; Nabney et al. 2005).

Self-Organizing Map (SOM), proposed by Kohonen (Kohonen 1990; Kohonen 1995), is regarded as an effective data visualization technique in data mining applications. It can map high-dimensional data into low-dimensional representation space and preserve the original topological relationship between input data via the data projection concept. Nevertheless, these conventional SOMs must predefine the map structure prior to training. Once the map size is not enough large, it will be unable to represent the original topology of input data set appropriately. On the contrary, too large map size may cause similar data be dispersed to excess number of clusters.

Growing SOM (GSOM) provides a feasible visualization solution for improving the foregoing problem of conventional SOM methods (Alahakoon et al. 2000; Haiying et al. 2004). It applies a dynamic structure that is generated during the training processes. The training starts with a small size of initial map and the map grows along with increasing number of the training data so as to reflect on the map the distribution of the training data. The resultant visualization map can appropriately develop different shapes depending on the clusters presented in the data. Nevertheless, the SOM and GSOM can handle only numeric data. Neither one can manipulate mixed-type data, containing both numeric and categorical attributes, in data clustering problems. To address this deficiency, Hsu (2006) proposed Generalized SOM (GenSOM) which handles mixed-type data. GenSOM utilizes a novel data structure, *distance hierarchy*, for uniformly representing numeric and categorical data so that both can be processed in a united manner. The distance between two mixed-type data is calculated by first mapping the data to points in distance hierarchies and then measuring the distance between the mapping points in the

hierarchies. Although alleviating the problem with mixed-type data, GenSOM adopted a fixed structure of the map and lacked the flexibility of adapting its structure according to the nature of the training data. In other words, it is unable to effectively preserve the topological order because of applying the conventional scheme, requiring a pre-specified map size prior to training.

In this paper, to tackle the problems seen in the previous SOM models, we propose Growing Mixed SOM (GMixSOM), intending to process mixed-type data with a dynamic map structure so as to offer data analyst a convenient, visualized tool for analyzing nowadays business data. The proposed model retains the advantages of the previous models by integrating the data structure of distance hierarchy and the concept of dynamic structure such that the model can manipulate mixed-type data and improve the visualized results in the low-dimensional projection map. This paper is structured as follows. Section 2, dynamic structure schemes for data projection methods and distance hierarchy for categorical data are reviewed and discussed briefly. Section 3, the processes of GMixSOM, cross insert and several performance indices are elaborated. Section 4, several experiments were conducted to verify the performance of GMixSOM for mixed-type data. Finally, some conclusions are stated at the end of this paper.

2. LITERATURE REVIEW

Several dynamic structure schemes were proposed to improve the constraints of fixed-size map occurring in data projection methods. Distance hierarchy can manipulate the distance measurement of categorical data in the training of SOM. In this section, they are briefly reviewed and discussed.

2.1 Dynamic structure schemes for data projection method

A fixed-size map possesses two major limitations: (1) both *a priori* the map size (number of neurons) and the topology (dimension and links structure) must be determined prior to training and (2) the problem with cluster boundary decision due to predetermined and fixed topological boundary (Forti et al. 2006). Therefore, a dynamic structure scheme is an appealing resolution for data projection methods to overcome these limitations of fixed-size map.

Blackmore and Miikkulainen (1993) proposed Incremental Grid Growing (IGG) to resolve the drawback that fixed grid map cannot properly reflect the structure of clusters in the input space. In this model, nodes and connections are added or deleted from the map according to the input distribution and their Euclidean distance satisfying a given threshold. Growing Cell Structure (GCS), proposed by Fritzke (1999; 1993; 1994) is another solution for providing map

a dynamic structure. The network of GCS has variable number of elements and k-dimensional topology whereby k is an arbitrarily positive integer chosen in advance. Only the winner node and its direct, topological neighbors are adapted for each input. Always after a constant number of training iterations, a new node will be inserted by splitting the longest edge emanating from the node possessing the maximum accumulated error. Two hierarchical architectures of GCS were proposed to extend the GCS and improve the massive upheaval due to the fact that node deletion is removed (Burzevski et al. 1996; Hodge et al. 2001).

Growing Grid can be regarded as a variant of self-organizing feature map (Fritzke 1995). There are two major phases, a growth phase and a fine-tuning phase, to build a growing rectangular network in this model. In the growth phase, a rectangular network is initialized with a minimum size and grows by means of inserting complete rows and columns until the desired size is reached or a performance criterion is met. Once the growing terminates, the reference vectors are further tuned to find the best value by a decaying learning rate in the fine-tuning phase.

Growing Hierarchical SOM (GHSOM) presents a growing hierarchical architecture to resolve fixed network architecture and reflect hierarchical structure of the input data in the map (Rauber et al. 2002). This model has a hierarchical structure, where SOM-like neural networks with adaptive architecture forms the various layers of the hierarchy. The size of these SOM-like neural networks as well as the depth of the hierarchy of the GHSOM is determined during its unsupervised training process according to the structure of the data. Growing Hierarchical Tree SOM (GHTSOM) consists of two main processes: training and clustering. A tree of identical SOMs is constructed in the training while the clustering process considers each level of the tree and uses self-organization to group neurons in classes (Forti et al. 2006).

GSOM applies another dynamic structure in SOM (Alahakoon et al. 2000), consisting of three phases: initial, growing and smoothing. In the initial phase, four neurons are randomly initialized and a growth threshold (GT) is calculated according to the spread-out factor (SF). In the growing phase, either several new neurons may be added around the best matching unit (BMU) or BMU' s error is rippled outward to its immediate neighbors. In the smoothing phase, inputs are projected to their BMUs and existing quantization errors are smoothed out without inserting new neurons. Furthermore, a semi-supervised learning method for the GSOM (Hsu et al. 2008), a hybrid method combining GSOM with SOM (Wang et al. 2002), a hexagonal GSOM map structure applying fixed training time to insert new neurons (Chan et al. 2008) were proposed.

In summary, most dynamic structure models of data projection methods effectively overcome the deficiency of fixed-size maps. Nevertheless, they can handle only numeric attributes but cannot directly process categorical or mixed-type data. Moreover, inserting new

neurons around BMU is an essential process in a dynamic structure model. Those new neurons were inserted around BMU when a specified time or the growth threshold was met. However, they were determined by the available locations but not the most suitable location with respect to the BMU. Those redundant new neurons will increase unnecessary computational effort during training. Furthermore, both the learning rate and the neighborhood size are decreased as the training iteration increases in SOMs. Therefore, the update force for neurons will be shrunk as these foregoing parameters decrease simultaneously. The order of training data will influence the update force. Those earlier training inputs obtain stronger update force than latter ones.

2.2 Distance hierarchy for categorical data

A distance hierarchy, structured by concept nodes and links, represents the ontological relationship between concepts (Hsu 2006). In this hierarchical structure, the upper nodes represent more general concepts; on the contrary, the lower nodes represent more specific concepts. For example, the nodes of Coke and Pepsi are belonged to carbonated drinks as shown in Fig. 1a. Juice, coffee and carbonated drinks all belong to "Any".

To illustrate the difference between distance hierarchy and other methods, the distances between Coke, Pepsi and Mocca are measured through distance hierarchy, simple matching and binary encoding, respectively, as shown in Table 1. With simple matching, two distinct values result in a distance of one and identical values have a distance of zero. With binary encoding, a categorical value is encoded by a vector of binary values. One of the binary values, corresponding to the categorical value, has a value of one and the others are set to zero. The distance between two categorical values is then measured by the Euclidean distance of the two vectors. As for the distance hierarchy scheme, assume the weight of each link is set to one to represent the distance between a node and its parent node. The distance between two categorical values is measured as the path length between the values in the distance hierarchy. As shown in Fig. 1a, the path length is 4 between Coke and Mocca and 2 between Coke and Pepsi. As the result shown in Table 1, neither simple matching nor binary encoding can distinguish the difference between the three drinks. In other words, the three drinks have the same distance (or similarity) based on the foregoing two methods. By contrast, distance hierarchy can intuitively and appropriately represent that Coke is more similar to Pepsi than to Mocca. Consequently, distance hierarchy is a feasible mechanism to represent the distance between categorical data.

Specifically, a point X in a distance hierarchy can be denoted by an anchor and its positive offset as $X = (N_X, d_X)$, where N_X is one of the leaf nodes and d_X is the distance from the root to X. The distance between point X and another one Y can be calculated as follows.

$$\delta(X,Y) = d_X + d_Y - 2d_{LCP(X,Y)}$$

$$\text{where} \quad d_{LCP(X,Y)} = \min\{d_X, d_Y, d_{LCA(N_X,N_Y)}\} \tag{1}$$

d_X and d_Y are the distance from the root to point X and Y, respectively. $d_{LCP(X,Y)}$ is the distance from the root to the lowest common point *(LCP)* of X and Y. N_X and N_Y are the anchor nodes of X and Y, respectively. $d_{LCA(N_x,N_y)}$ is the distance from the root to the lowest common ancestor (*LCA*) of anchor nodes of X and Y. Lowest common ancestor of two points is the lowest tree node which is an ancestor of the two nodes. For the example in Fig. 1a, assume X = (Coke, 2) and Y = (Mocca, 1.6), $d_{LCA(Coke,Mocca)} = d_{Any} = 0$, $d_{LCP(X,Y)} = 0$, and $\delta(X,Y) = 2+1.6-0 = 3.6$.

Numeric distance hierarchy can be regarded as a special type for a numeric attribute in distance hierarchy. It is a distance hierarchy which consists of only two leaf nodes and two links as shown in Fig. 1b. The root and two leaf nodes are labeled by 0, – and +, respectively. A point X in a numeric distance hierarchy has the value (N_X, d_X) where the anchor N_X is either '+' or '– ' depending the sign of the value which the point represents and the offset d_X is the distance from X to the root 0.

In GenSOM, each attribute of the training data as well as its corresponding component of the prototype of a map neuron is associated with a distance hierarchy. In the context of identifying the BMU for an input during training a GenSOM, attribute values of the input and the components of the neuron's prototype are mapped to points in their associated distance hierarchies. Then, the distance between the input instance and the prototype of the neuron is measured via the mapping points by the aggregate distance between the points in the distance hierarchies (Hsu 2006).

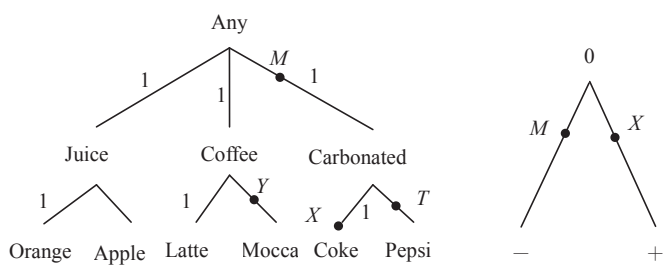


Figure 1: (a) Distance hierarchy with link weight 1 for a categorical attribute "Drink". (b) A distance hierarchy for a numeric attribute.

Table 1: Distance comparison between methods.

Node \ Method		Distance hierarchy	Simple matching	Binary encoding
Coke	Mocca	4	1	1.414
Coke	Pepsi	2	1	1.414
Mocca	Pepsi	4	1	1.414

2.3 Construction of distance hierarchy

In data mining, accredited domain concept hierarchy is the first choice for distance hierarchy such as International Classification for Diseases (ICD 2010) in medicine, ACM's Computing Classification System (CCS 2010) in computer science, and Global Product Classification (GPC 2010) in business. However, many domains still lack regular and accredited concept hierarchies for defining the relationship between different items. In such domains, the distance hierarchies can be constructed manually by domain experts. People may argue over the qualification of distance hierarchies based on experts' subjective viewpoints. Nevertheless, domain problems should be defined and resolved by the experts in the domain in general. The abovementioned existing hierarchies were defined by domain experts after all.

In addition to the aforementioned two sources for the distance hierarchies, an automatic-constructing distance hierarchy is another alternative. For example, a distance hierarchy can be built by hierarchically clustering the dissimilarity between categorical values. The dissimilarity between categorical values can be measured by the co-occurrence proportion of categorical values to an external probe set (Das et al. 1998; Palmer et al. 2003). Especially, this method is suitable for encrypted categorical values due to confidentiality or privacy requirement. Furthermore, a two-level distance hierarchy in which all values are located at the same level below the root can be used in case that the hierarchy cannot be built due to the lack of a proper external probe set. In a two-level hierarchy, the dissimilarity between two distinct categorical values is one and other zero between two identical values. A disadvantage of the hierarchy is that the different extent of similarity among different values cannot be discriminated. In other words, a two-level distance hierarchy fails to reflect the similarity degree between categorical data.

For data projection methods, the variety of distance hierarchies may yield different mappings in accordance with the similarity relationship of categorical items, respectively. Accredited domain concept hierarchies can be accepted by most people but they are only supported in a few domains. Consequently, expert-built concept hierarchy becomes a reasonable alternative for categorical data analysis in practice. Automatic-constructing concept hierarchies can be built objectively provided that a proper external probe set can be found. The set of labels in the class attribute is usually a good choice. In the four sources for the hierarchies, the approach of the two-level hierarchies is the simplest one but may fail to preserve correct topological order of the data on the map due to the equal distance between categorical values.

3. VISUALIZE MIXED-TYPE DATA BY GMIXSOM

In this paper, we propose a mixed-type SOM with dynamic structure, called growing mixed SOM (GMixSOM), which integrates the distance hierarchy and the dynamic structure

scheme to manipulate both numeric and categorical data in a flexible growing map. By means of GMixSOM, the resultant map could better present the topological relationship between mixed-type data in high-dimensional space than that of GSOM. The training algorithm of the model is presented below and elaborated in the subsequent sections.

Input: an *n*-dimensional training dataset *DS*, a set of *n* distance hierarchies, the number of training epoch E, and spreading factor SF
Output: a trained GMixSOM

Initial phase

Create five neurons and randomly assign their initial weights;

Determine growing threshold GT by spreading factor SF;

Growing phase

For each training epoch

Reset error of each neuron;

Initialize learning rate and neighborhood size;

For each input *x* in *DS*

- Determine the best matching unit (BMU) of *x;*
- Update BMU and its neighbors;
- Increase the error of BMU;
- Once the error of BMU is larger than GT, a new neuron will be added by cross insert or the error of BMU will be rippled outward;

Repeat till all inputs have been presented

Repeat till the epoch time equals to *E*

Figure 2: The proposed GMixSOM training algorithm.

3.1 Process of GMixSOM

The training of a GMixSOM can be divided into two phases: initialization and growing, as shown in Fig. 2. In the initialization, a small-size map with random weights is created and growing threshold is determined. In the growing, training data are presented one by one to adjust the weight distribution of map neurons and meanwhile training errors are accumulated in neurons. When the accumulated error in a neuron exceeds the growing threshold, a new neuron is inserted or the weight of the neuron is rippled out to neighbors. In GMixSOM, the identification of the BMU and the adaptation of map neurons during training are conducted via distance hierarchy so as to take the similarity information embedded in categorical data into consideration.

3.1.1 Initialization phase

(1) Five neurons arranged in a cross shape are initialized and their weights are assigned randomly. These initial neurons are ready for cross insertion in the growing phase.

(2) Growing threshold GT is calculated by a predetermined spreading factor SF and data

dimensionality D as follow.

$$GT = -D \times \ln(SF) \quad (2)$$

3.1.2 Growing phase

(1) Training epoch

In GMixSOM, we apply the epoch training process similar to that of Incremental Grid Growing (IGG)(Blackmore 1995) and the first two phases of GSOM (Alahakoon et al. 2000). Training proceeds in several epochs without the need of a smoothing phase; the smoothing effect can still be achieved via the below processes.

- The accumulated error of each neuron will be reset before the starting of each training epoch.
- Learning rate (LR) and neighborhood size (NS) are decreased by epoch rather than by each input data.
- All inputs will be presented to the map in each epoch.

Therefore, those existing neurons in the last epoch can be smoothed out since all inputs will be projected to the map in a new epoch. It makes those training epochs achieve smoothing effect similar to GSOM's.

(2) Initialize learning rate and neighborhood size

To prevent the influence of presentation order of the inputs, both LR and NS of GMixSOM are decreased by epoch. In other words, each update of BMU and its neighbors will use the same LR and NS in the same epoch. The learning rate function is stated as

$$\alpha(e+1) = \rho \times \psi(n) \times \alpha(e) \quad (3)$$

where ρ, $0 < \rho \leq 1$, is the reduction rate, $\alpha(e)$ is the learning rate at the eth epoch. $\phi(n)$ is a function that gradually takes higher values as the map grows and the number n of neurons becomes larger along with training time. One simple formula that can be used for $\phi(n)$ is $(1 - R / n_e)$ where n_e is the number of current neurons in the map. In this paper, R is set to 4.8 since the number of starting neurons is five.

The neighborhood size gradually decreases along with the increasing epoch as follow.

$$\sigma(e) = 1 + (\sigma(0) - 1) \times (1 - \frac{e}{E}) \quad (4)$$

where $\sigma(0)$ is the initial neighborhood size, E is the number of total training epochs specified by the user, e is the current training epoch, and $\sigma(e)$ is the neighborhood size at the eth epoch.

(3) Determine the best matching unit of each input

The BMU is identified by finding the closest neuron of which prototype m has the shortest distance with the n-dimensional input x. In essence, each categorical attribute value x_i of x is mapped to a point at the leaf labeled by the same value in the hierarchy, like the point X in Fig. 1a. The structure of GMixSOM's prototype m is the same with that of GenSOM's (Hsu 2006). Each component m_i of prototype m consists of two parts, a symbol and a positive real value: (N, d) which can be mapped to a point in its associated hierarchy in the way that the point has an anchor of N and a distance of d to the root. Unlike that of an input attribute, the mapping point of m_i can be at any position in the hierarchy, like the point M in Fig. 1a.

(4) Update BMU and its neighbors

The BMU and its neighbors are updated with the learning rate and the neighborhood function as follows.

$$w_i(t+1) = w_i(t) + \alpha(e) \times \eta_{i,x}(t) \times (w_x(t) - w_i(t)),\ i \in N_{BMU} \quad (5)$$

where $w_i(t)$ and $w_i(t+1)$ are the prototypes of neuron i before and after adaptation at the tth iteration. $\eta_{i,x}(t)$ is the neighborhood function and a Gaussian function is used for $\eta_{i,x}(t)$ in this paper. $w_x(t)$ is the weight vector of the input. N_{BMU} is the neighborhood of the BMU.

(5) Increase the error of BMU

The error of a BMU is accumulated by the error value, which is the difference between weight vectors of the BMU and the input x. The accumulated error of neuron is calculated as

$$Err_i(t+1) = Err_i(t) + \sqrt{\sum_{d=1}^{D} (w_{x,d}(t) - w_{i,d}(t))^2} \quad (6)$$

where, $Err_i(t)$ and $Err_i(t+1)$ are the error of neuron i before and after recomputed at the tth iteration. $x(t)$ and $w(t)$ are the prototypes of input and neuron i. D is the number of total vector dimension.

(6) The error of BMU is larger than GT

Once the accumulated error of the BMU is larger than GT, insertion of a new neuron or reduction of the error takes place. If the BMU is a boundary neuron, a new neuron will be inserted into the map by cross insert and its weight vector will be initialized to match its neighborhood. If the BMU is a non-boundary neuron, the error of BMU will ripple outward to its immediate neighbors, instead of growing. The new weights are assigned as follows (Hsu et al. 2008).

$$Err_{BMU}(t+1) = \frac{Err_{BMU}(t)}{2} \quad (7)$$

$$Err_{nbrs}(t+1) = Err_{nbrs}(t) + \frac{1}{n_{nbrs}} \times \frac{Err_{BMU}(t)}{2} \quad (8)$$

where, $Err_{\mathrm{BMU}}(t)$ and $Err_{\mathrm{BMU}}(t+1)$ are the error of BMU before and after recomputed at the tth iteration. $Err_{\mathrm{nbrs}}(t)$ and $Err_{\mathrm{nbrs}}(t+1)$ are the error of BMU's neighbors before and after recomputed. n_{nbrs} is the number of BMU's neighbors.

(7) Repeat steps 3 - 6 till all inputs are presented.

Add one to the epoch number after all inputs are presented to the training.

(8) Start new epoch training till $e = E$

If the epoch number is less than the number of total training epochs, then return to step 2 to start a new training epoch.

3.2 Cross insertion

The number of starting neurons is four in the initialization phase of GSOM (shown in Fig. 4a). When the error of BMU is larger than GT, one to four neurons will be inserted around a boundary BMU in GSOM (illustrated in Fig. 3). It is an essential process since the BMU is insufficient to represent its Voronoi region. However, redundant and unsuitable neurons will be inserted via this process and cost unnecessary effort to deal with in the growing phase.

To avoid redundancy and save computation effort, we propose a new method for neuron insertion, called "cross insert", to determine the most suitable location for the new neuron. In this study, the number of initial neurons is five as shown in Fig. 4b. The location of new neuron $z_{\mathrm{new}}(t)$ is determined by

$$new = \begin{cases} argmin_i\{\Delta_{v_3,z_i(t)} | z_i(t) \in N_{v_1}\}, \Delta_{v_2,z_i(t)} = \Delta_{v_2,z_j(t),i\neq j} \\ argmin_i\{\Delta_{v_2,z_i(t)} | z_i(t) \in N_{v_1}\}, \quad otherwise \end{cases} \quad (9)$$

where v_1, v_2 and v_3 are the first, second and third BMU of input $x(t)$, $z_i(t)$ is the eligible location for the new neuron, N_{v_1} is the neighbors of v_1 in the map, $\Delta_{v_2,z_i(t)}$ and $\Delta_{v_3,z_i(t)}$ are the Euclidean distance from $z_i(t)$ to v_2 and v_3 in the map space, respectively.

For the example shown in Fig. 4c, z_1, z_2 and z_3 are three eligible free locations around v_1. If v_1, v_2 and v_3 are the first, second and third BMU of input $x(t)$, z_2 will be the chosen location for the new neuron, because it is the closest eligible location near v_3. In another case, v_1, v'_2 and v'_3 are the first, second and third BMU of input $x(t)$, z_3 will be the chosen location for the new neuron via the cross insert.

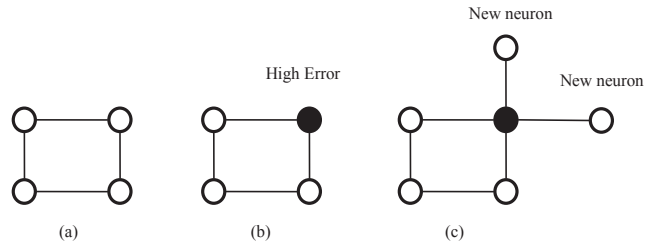


Figure 3: Generation of new neuron from boundary of the network

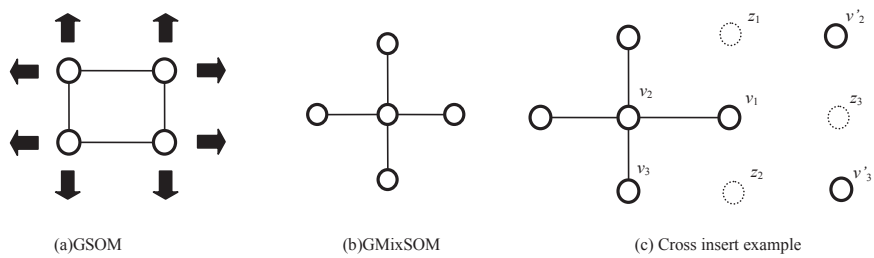


Figure 4: Initial neurons of GSOM and GMixSOM.

3.3 Clustering validity and mapping quality

The SOM is an effective visualization tool for presenting topological relationship between high-dimensional data on a low-dimensional map. Similar data instances will be projected into the same neuron. That is, the instances projected in a neuron can be regarded as a cluster. Therefore, we apply clustering validity and mapping quality to verify the projection quality of the various SOM models mentioned in this paper.

3.3.1 Mean squared error

Generally, the competitive learning methods are usually based on minimizing the mean squared error (MSE) (Du 2010; Tsypkin 1973). Additionally, most clustering methods apply MSE as the index of clustering performance as well. Thus, we apply MSE to measure cluster validity as

$$MSE = \frac{1}{N}\sum_{i=1}^{N} E_i \text{ where } E_i = \sum_{k=1}^{K} \mu_{ki}\|x_i - c_k\| \quad (10)$$

where N is the number of total inputs, and μ_{ki} is connection weight assigned to prototype c_k with respect to x_i, denoting the membership of the ith input into the kth neuron. When k is the winning neuron to x_i, $\mu_{ki} = 1$ and $\mu_{ki} = 0$ otherwise.

3.3.2 Within-between ratio

A good clustering method should generate clusters with small intra-cluster deviation (cluster compactness) and large inter-cluster separation (cluster separation). Another clustering validity index, considering both cluster compactness and cluster separation, was thus proposed to measure the performance of clustering as follows (Du et al. 2006).

$$E_{WBR} = \frac{1}{K}\sum_{k=1}^{K} \max_{l \neq k} \left\{ \frac{d_{WCS}(c_k) + d_{WCS}(c_l)}{d_{BCS}(c_k, c_l)} \right\} \quad (11)$$

where the within-cluster scatter for cluster k, denoted by $d_{\mathrm{WCS}}(c_{\mathrm{k}})$ and the between-cluster separation for clusters k and l, denoted by $d_{\mathrm{BCS}}(c_{\mathrm{k}}, c_{\mathrm{l}})$, are calculated by Eqs. (16) and (17).

$$d_{WCS}(c_k) = \frac{\sum_i \|x_i - c_k\|}{n_k} \quad (12)$$

$$d_{BCS}(c_k, c_l) = \|c_k - c_l\| \quad (13)$$

where n_{k} is the number of data points in cluster k. c_{k} and c_{l} is the prototype of cluster k and l, respectively. The clustering objective is to minimize E_{WBR}.

3.3.3 Aggregation entropy

For the data with class label, aggregation entropy can used to evaluate the extent of separation of different class labels to distinct clusters. A good clustering shall assign the data with the same class label to the same cluster rather than to different ones. The aggregation entropy is defined as follows.

$$Entropy = \sum_i \frac{n_i}{n} \times Entropy_i \quad (14)$$

$$Entropy_i = -\sum_{k=1}^{K} p(class_k)\, log_2\, p(class_k) \quad (15)$$

where n_{i} is the number of inputs projected in the ith neuron, n is the number of total inputs, $Entropy_{\mathrm{i}}$ is the entropy of the ith neuron. $p(class_{\mathrm{k}})$ is the probability mass function of outcome class k. The smaller the entropy the better the clustering is.

3.3.4 Mapping quality

Mapping quality can be analyzed by the degree of distance preservation (Vathy-Fogarassy et al. 2009). In this paper, Sammon stress (Sammon 1969) is applied to measure the mapping quality of each method. The metric Sammon stress is stated as follows.

$$E_{SM} = \frac{1}{\sum_{i<j}^{n} d_{ij}} \sum_{i>j}^{n} \frac{(d_{ij} - D_{ij})^2}{d_{ij}} \quad (16)$$

where $d_{i,j}$ and $D_{i,j}$ are the Euclidean distance between neuron i and j in data space and map space, respectively. The distance $D_{i,j}$ between inputs i and j shall be close to the distance $d_{i,j}$ between the weights of neurons to which inputs i and j are projected. The smaller E_{SM} the better quality the mapping has. When the applied mapping method utilizes geodesic distances instead of the Euclidean ones, the stress function is also evaluated using the geodesic distance calculation.

4. EXPERIMENTS

A testing platform in Java was built to demonstrate the performance of GMixSOM. One synthetic and one real dataset were used for the experiments.

4.1 Synthetic dataset

Synthetic dataset "student" consists of three attributes, *Department* and *Drink* for categorical data and *Amount* for numeric data. According to categorical attributes, those students can be divided into nine groups as shown in Table 2. The value of attribute "Amount" is assigned by a normal distribution with pre-specified mean and standard deviation. The distance hierarchies for categorical data were shown in Fig. 5.

A 10×10 map was used for SOM and GenSOM. According to the suggestion in software package SOM_PAK (Kohonen et al. 1996), the initial LR, the initial NS, and the number of total training time were set to 0.5, 5, and 1000, respectively. Binary encoding was applied to convert categorical data into binary values prior to training the SOM. In the GSOM, the initial LR, initial NS, and SF were set to 0.95, 3 and 0.3, respectively. Binary encoding was applied to convert categorical data into binary value. In the GMixSOM, the number of total training epochs, initial LR, initial NS, and SF were set to 5, 0.95, 3 and 0.3 respectively.

Table 2: Synthetic mixed-type dataset "Student"

Group	Dept.	Drink	Amount (μ , σ)	Count
1	MIS	Coke	(500, 25)	60
2	MBA	Pepsi	(400, 20)	30
3	MBA	Pepsi	(300, 15)	30
4	EE	Latte	(500, 25)	60
5	CE	Mocca	(400, 20)	30
6	CE	Mocca	(300, 15)	30
7	SD	Apple	(500, 25)	60
8	VC	Orange	(400, 20)	30
9	VC	Orange	(300, 15)	30

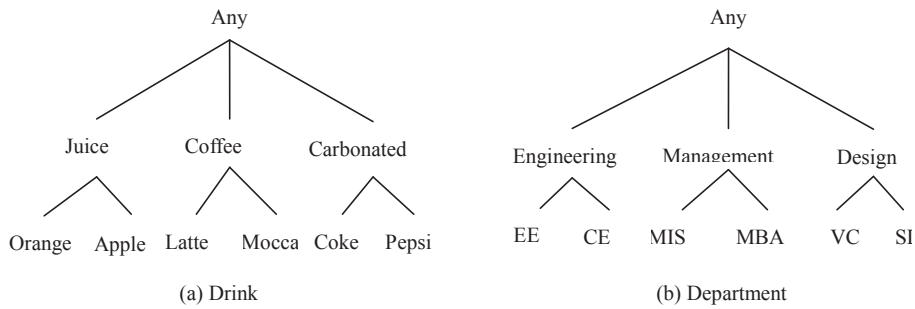
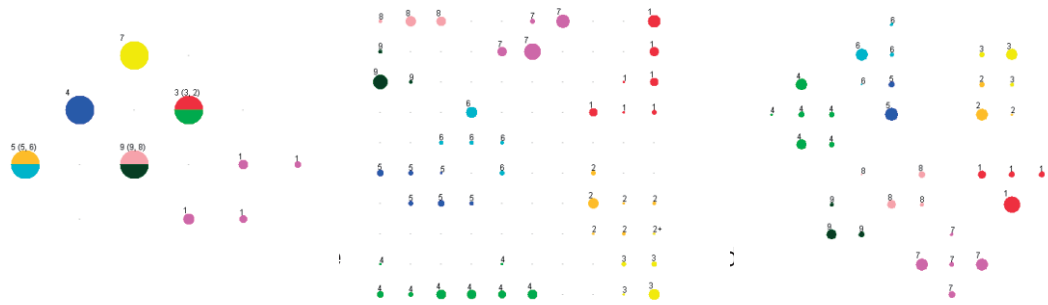


Figure 5: Concept hierarchies for “Student”

As shown in Fig. 6a, the visualization map of GSOM obviously fails to represent the topological relationship between different groups in the mixed-type dataset. Several problems can be observed from the visualized results. I) Those inputs from different groups are projected to the same neuron. For example, inputs from group 5 and 6 are projected to the same neuron. II) Neurons holding the data from similar groups are not closer to each other than those holding the data from different groups. For example, neurons for group 8 and 9 are closer to that for group 1 than that for group 7. In fact, group 7, 8 and 9 should be closer to each other in accordance with the dataset. The reasons for the problems are that the binary encoding scheme which GSOM utilized cannot reflect the distance between categorical data appropriately. A more appropriate topological relationship between mixed-type data was revealed by the GenSOM in the resultant maps shown in Fig. 6b, those neurons holding similar data groups are closer than those holding different groups: groups 1, 2, and 3 are next to each other and so are groups 4, 5 and 6, as well as groups 7, 8 and 9. The result demonstrates that similarity relation embedded in categorical values can be preserved by means of the distance hierarchy mechanism which is used by the GenSOM and the GMixSOM. Nevertheless, the predetermined fixed map may cause the neurons close to the map border unavoidably take more inputs, some of which may significantly vary. In other words, border effect is resulted from the phenomenal that some inputs cannot find an appropriate BMU and have to select one from those neurons near the border of the map.

GMixSOM can reflect *correct* topological relationship among the groups in the visualization map (shown in Fig. 6c) contrast to GSOM and GenSOM. Inputs from different groups can be projected to appropriate neurons in the sense that those similar groups can be closer than different groups. Groups 1, 2 and 3 are projected to one region, and so are groups 4, 5 and 6 as well as groups 7, 8 and 9. The experiment demonstrated that GMixSOM can yield a more feasible and appropriate visualization map than GenSOM and GSOM with respect to mixed-type data.



(a) GSOM. (b) GenSOM. (c) GMixSOM.

Figure 6: Visualized results for "Student".

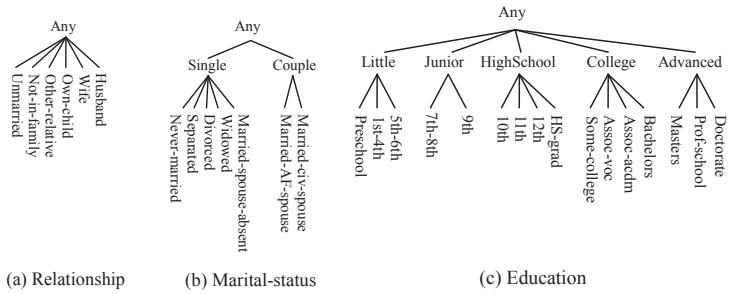
4.2 Real dataset

"Adult" is a real-world mixed-type dataset which comes from the UCI repository (Asuncion et al. 2007). The dataset contains 48,842 instances with 15 attributes including eight categorical and six numeric attributes. Each data instance has a class label, indicating whether the household' s salary is over 50K. About 76% of the data are of less than 50K, only 24% are of over 50K. A subset including 10,000 instances were sampled randomly from the dataset. Seven relevant attributes according to (Hsu 2006), including three categorical attributes *Marital_status, Relationship*, and *Education*; and four numeric attributes, *Capital_gain, Capital_loss, Age*, and *Hours_per_week,* were used for the experiment. The manually-constructed concept hierarchies for categorical attributes were shown in Fig. 7. GSOM, GenSOM and GMixSOM retained the same parameters setting as the last experiment.

The resultant visualization maps by the three methods are shown in Fig. 8 and Fig. 9. The size of the pies reflects the number of data instances projected in the neuron. The color of the pie slice represents the class of the data. The yellow color indicates salary >50K while the red indicates <50K. The resultant visualization maps of GenSOM is shown in Fig. 8a. Since the GenSOM used a predetermined fixed-size map, the border neurons cannot but take many inputs of which some may be projected to other extra neurons shall the map size have been larger. Consequently, border neurons may contain more inputs than expected and result in a high error value than those neurons in the other areas due to border effect. Contrast to the GenSOM, the border neurons will not take irrelevant inputs in the GMixSOM with an extendable structure. Every GMixSOM neuron can share the error to its neighbors via neuron insertion and ripple outward when its accumulated error exceeds the threshold during training. In addition, GMixSOM can obtain a lower MSE than GenSOM under the same map size as shown in Fig. 8b.

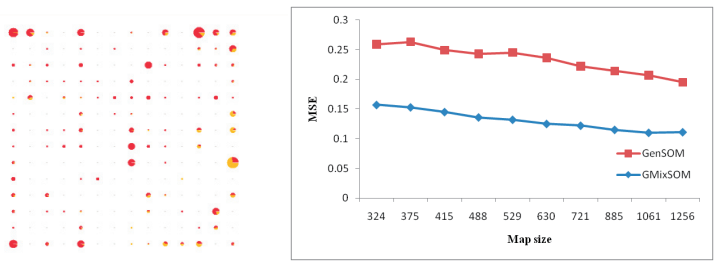
As shown in Fig. 9, The GSOM generated only 16 neurons and the projection result failed to generate a sufficient gap between different groups to distinguish those boundaries of

potential clusters on the map since those neurons were not enough for distributing different inputs to appropriate neurons in accordance with their diversity. Contrary to the GSOM, under the same parameters GMixSOM yielded sufficient neurons. The resultant map spread out and data instances were projected to appropriate neurons according to their characteristics. Closely inspecting the map, we discovered that the instance with salary >50K mainly located at the upper-left region. In particular, some of those neurons are in yellow without any slice in red, indicating all of the instances in the neurons are of salary >50K. The instances projected in the neurons in the lower-right region are mainly with salary <50K, that is, the slice in red occupies the major portion of the pie.



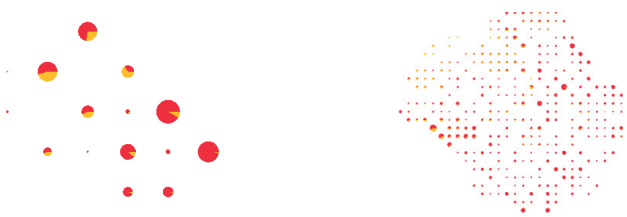
(a) Relationship (b) Marital-status (c) Education

Figure 7: Concept hierarchies for “Adult”.



(a) GenSOM. (b) Comparison between GenSOM and GMixSOM.

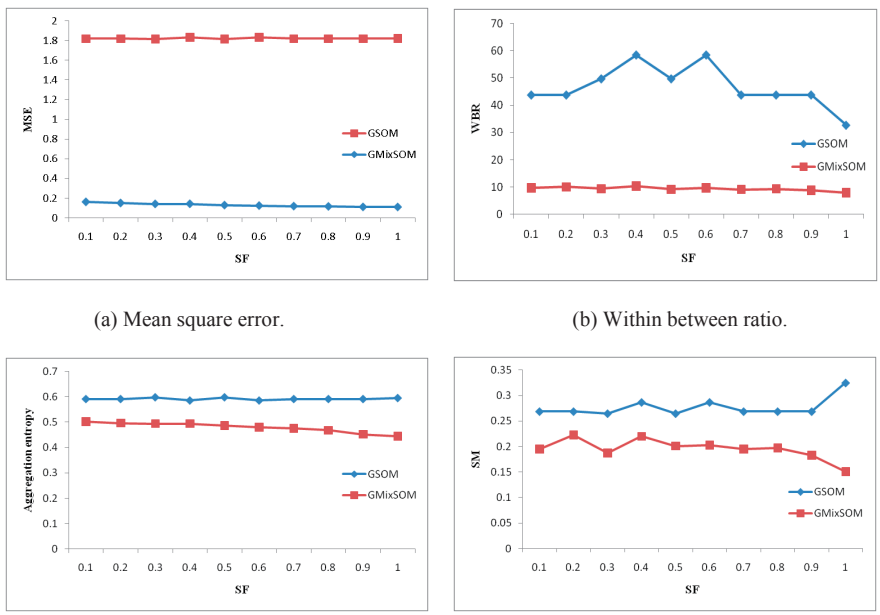
Figure 8: Visualized result and MSE of GenSOM for “Adult”.



(a) GSOM. (b) GMixSOM.

Figure 9: Visualized results of GSOM and GMixSOM for “Adult”.

Furthermore, we applied a variety of spreading factors SF to verify the effectiveness of GMixSOM. As the results shown in Fig. 10a, the MSE of GMixSOM is obviously lower than GSOM because GMixSOM could provide enough neurons to avoid under-representation of Voronoi region in the growing phase. Additionally, the Within-Between Ratio, as shown in Fig. 10b, demonstrates that GMixSOM could also consider both cluster compactness and cluster separation. For the aggregation entropy of both methods as shown in Fig. 10c, GMixSOM can decrease linearly but GSOM increases and slightly fluctuates as SF increases. As the Sammon stress shown in Fig. 10d, GMixSOM provides better mapping quality than GSOM and mapping quality improves as the SF increases. The result indicates that the map quality can be controlled as user's specified SF in GMixSOM but GSOM. In summary, GMixSOM can provide a better spread-out control for different SFs according to user's need.



(a) Mean square error.

(b) Within between ratio.

(c) Aggregation entropy.

(d) Sammon stress function.

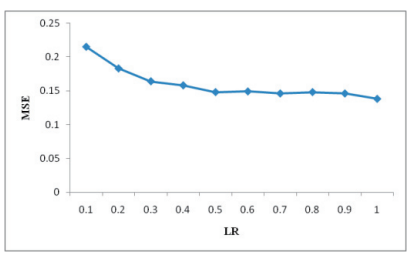
Figure 10: Performance of two methods in different SF.

4.3 Sensitivity of parameters

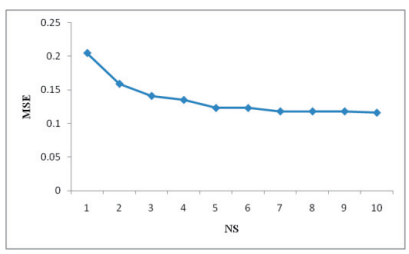
Parameter setting plays a crucial role for influencing the performance of final results in most algorithms. Therefore, sensitivities of four major parameters, including SF, initial LR, initial NS and total epoch time, were verified via several experiments. In these experiments, all the foregoing parameters setting remained the same except for the specified parameter which was under testing. As shown in Fig. 10a, the change of MSE is quite slight even the map size grows as SF increases. MSE did not fluctuate as the map size increase. The boundary of clusters may be identified more clearly as SF increases. Additionally, MSE shrunk as initial LR and NS increase (as shown in Fig. 11a and b). Generally speaking, the update ratio and

range were determined by the initial LR and NS, respectively. Higher initial LR makes neurons move toward inputs in a short time. The update range of BMU's neighbors will be enlarged via a higher initial NS. As a result, more neurons can be updated under a larger NS. The results indicate that higher initial LR and NS make better resultant maps, which is consistent with the suggestion of parameters setting seen in the literature of the SOMs.

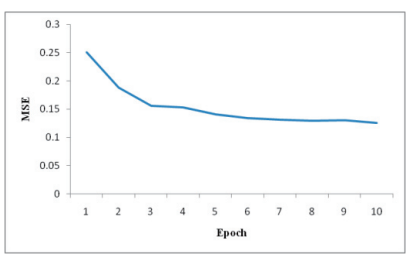
Finally, the influence of total epoch time (E) was verified in two different intervals. In the first interval, the total epoch time was set from 1 to 10 with a step of 1 to obtain a variety of resultant maps. As the results shown in Fig. 11c, MSE swiftly decreased from E = 1 to E = 5 and became slightly decreasing after E = 5. The outcome demonstrated quantization error of existing neurons can be smoothed out via the next epoch in the training. In the second interval, the total epoch time was set from 5 to 100 with a step of 5. According to the trend of MSE changing (shown in Fig. 11d), MSE swiftly decreased from E = 5 to E = 20 and became slightly decreasing from E = 20 to E = 55. Finally, GMixSOM reached convergent status since MSE retained a stable value after E = 60. The result showed that GMixSOM can reach convergence as the epoch increases. However, it seems unnecessary to spend tremendous effort for a little gain. According to the MSE trend in this case, GMixSOM becomes relative stable when E = 5.



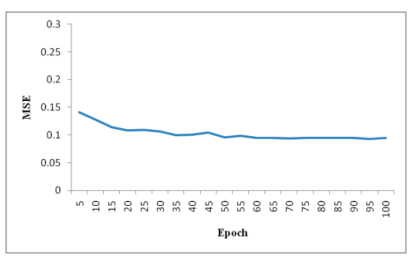
(a) MSE in different initial LR.



(b) MSE in different initial NS.



(c) MSE in the first interval.



(d) MSE in the second interval.

Figure 11: MSE for GMixSOM in different parameter setting

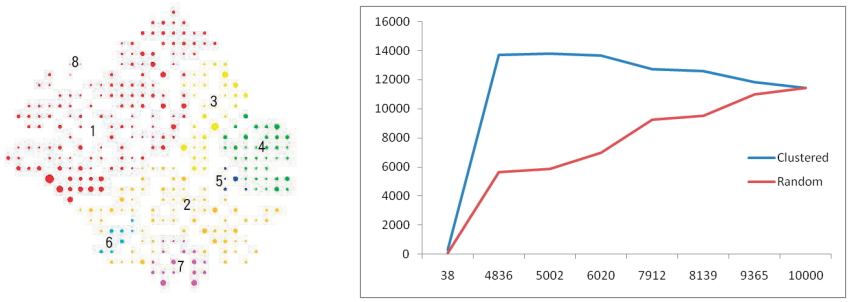
4.4 Application to catalogue marketing

To demonstrate practical value of analysis result, we apply the result to catalog marketing. We first cluster the data according to the projection of GMixSOM and then analyze the sales profit of catalogue marketing based on the clustered data compared with that based on random sampling.

The clustering was performed automatically upon the projected GMixSOM map by using DBSCAN clustering algorithm (Ester et al. 1996) These projected neurons were clustered to eight clusters without any noise (neuron) under automatic cluster searching. In this case, the expected number of cluster and the minPts (minimum number of points) were set to 8 and 2, respectively. Then, the radius parameter Eps started from 0.5 and was iteratively adjusted till the number of cluster was equal to the expected value or reached a stable value without change on the number of clusters. At the final, eight clusters were obtained with Eps = 1.414 as shown in Fig. 12a. The cluster numbers were superimposed to indicate the cluster's location.

Table 3 shows the distribution of Salary >50 K in individual clusters sorted by descending percentage of >50K. Obviously, Cluster 8 has the highest >50K ratio (100%) which is far exceeding the overall distribution (23.82%). It has 32% of Education value Prof-school, 100% of Marital_status value Married-civ-spouse (MCS), 97% of Relationship value Husband, an average of 46 years old in age, about 50 in Hours_per_week, 99,999 in capital_gain, and an average of zero in capital_loss. In fact, the result shows that the percentage of >50K in individual clusters are all significantly different from the overall distribution, demonstrating that the projection and the clustering are effective regarding separating the data according to their characteristics.

For catalogue marketing, the richer groups, which have a larger portion of >50 K, shall have higher priority of receiving promotional catalogues. We assume that the cost of mailing a catalogue is NT$2 and an average profit of NT$10 per person can be collected if the person's salary is over 50K, and otherwise an average profit of NT$1. Fig. 12b shows the expected profits of the catalogue marketing under this setting. The profits shown on the upper line are calculated based on the clusters of the dataset segmented by our method, in which the richer groups have higher priority of receiving the catalogues. We get the maximum expected profit of NT$13,808 upon mailing to the first three groups, i.e., 8, 1 and 5, representing a collection of 5,002 customers. In contrast, the profits shown on the lower line were calculated according to random selection, in which customers for receiving the catalogues were randomly drawn. The expected profit reaches the maximum of NT$11,438 when all the customers are included in the mailing list. This simulation shows that GMixSOM can segment customer data to further help decision maker identify better candidates for catalogue marketing and raise promotion effectiveness.



(a) Clustered projected map (b) The expected profit

Figure 12: Clustered projected GMixSOM map via DBSCAN and the expected profit of catalogue marketing

Table 3: The distribution of Salary attribute in each cluster

C	Education (%)	Marital status (%)	Relationship (%)	Age	Hrs	Gain	Loss	>50K (%)	Count
8	Prof-school(32)	MCS(100)	Husband(97)	46	50	99,999	0	100	38
1	HS-grade(31)	MCS(96)	Husband(85)	43	40	911	121	42	4,798
5	Bachelors(61)	Divorced(59)	Not-in-family(78)	37	42	1,197	96	19	166
4	Some-college(59)	Never-married(66)	Not-in-family(100)	38	40	935	46	10	1,018
2	HS-grade(52)	Never-married(64)	Own-child(37)	36	43	549	57	6	1,892
6	HS-grade(87)	Divorced(77)	Unmarried(77)	44	40	618	19	5	227
3	Some-college(62)	Never-married(70)	Own-child(45)	30	36	283	65	4	1,226
7	HS-grade(91)	Never-married(53)	Not-in-family(100)	35	42	445	37	4	635
All	**HS-grade(33)**	**MCS(47)**	**Husband(41)**	**39**	**41**	**1,113**	**86**	**24**	**10,000**

5. CONCLUSIONS

To facilitate analysis of complex business data for decision making, a growing mixed SOM, integrating distance hierarchy and dynamic structure concept, is proposed to manipulate mixed-type data and improve the visualization result of GSOM. The model can provide a dynamic structure to generate a flexible map according to nature of the data and faithfully present the topological relationship between mixed-type data via distance hierarchy.

In addition, we applied only the first two phases of the original GSOM training algorithm excluding its smoothing phase but achieved similar smoothing effect by using epoch training. By means of epoch training, the resultant map in fact obtains better smoothing out effect than that of GSOM. Furthermore, cross insert is proposed to determine the most suitable location for adding new neurons without expensive computation. The scheme avoids redundant neurons

being freely added to neighboring positions of BMU and saves the effort for processing redundant neurons. Furthermore, GMixSOM could reach convergent status as the epoch time increases and the performance of clustering and mapping could be improved as SF increases.

According to the results of experiments, we demonstrate that GMixSOM generates more appealing visualization result for high-dimensional mixed-type data than GSOM. Moreover, both clustering validity and mapping quality of GMixSOM are superior to those of GSOM as well. Therefore, the proposed GMixSOM can offer a feasible solution for an effective visualization means in high-dimensional mixed-type data analysis. An application of clustering results of a real-world mixed-type data to catalogue marketing was also presented to demonstrate the practical value of the proposed model as a data analysis tool.

6. ACKNOWLEDGMENT

The work is supported by National Science Council, Taiwan under grant NSC 98-2410-H-224-010-MY2.

REFERENCE

1. Alahakoon, D., Halgamuge, S.K., and Srinivasan, B. "Dynamic self-organizing maps with controlled growth for knowledge discovery," *Neural Networks, IEEE Transactions on* (11:3), 2000, pp. 601-614.
2. Asuncion, A., and Newman, D.J. "UCI Machine Learning Repository," Irvine, CA: University of California, School of Information and Computer Science, 2007.
3. Blackmore, J., and Miikkulainen, R. "Incremental grid growing: encoding high-dimensional structure into a two-dimensional feature map," Neural Networks, 1993., IEEE International Conference on, 1993, pp. 450-455.
4. Blackmore, J.M. "Visualizing high-dimensional structure with the incremental grid growing neural network," University of Texas at Austin, 1995.
5. Burzevski, V., and Mohan, C.K. "Hierarchical growing cell structures," Neural Networks, 1996., IEEE International Conference on, 1996, pp. 1658-1663.
6. Computing Classification System (available online at http://www.acm.org/about/class/).
7. Cesario, E., Manco, G., and Ortale, R. "Top-Down Parameter-Free Clustering of High-Dimensional Categorical Data," *Knowledge and Data Engineering, IEEE Transactions on* (19:12), 2007, pp. 1607-1624.
8. Chakrabarti, S., Cox, E., Frank, E., Guting, R.H., and Han, J. *Data Mining: Know It All*, Morgan Kaufmann, Burlington, 2009.

9. Chan, C.-K.K., Hsu, A.L., Tang, S.-L., and Halgamuge, S.K. "Using Growing Self-Organising Maps to Improve the Binning Process in Environmental Whole-Genome Shotgun Sequencing," *Journal of Biomedicine and Biotechnology* (2008), 2008, pp. 1-10.
10. Das, G., Mannila, H., and Ronkainen, P. "Similarity of attributes by external probes," Knowledge Discovery and Data Mining, AAAI Press, New York, 1998, pp. 23-29.
11. Du, K.L. "Clustering: A neural network approach," *Neural Networks* (23:1), 2010, pp. 89-107.
12. Du, K.L., and Swamy, M.N.S. *Neural networks in a softcomputing framework*, Springer, London, 2006.
13. Dunham, M.H. *Data Mining: Introductory and Advanced Topics,* Prentice Hall, Upper Saddle River, NJ, 2003.
14. Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. "A density-based algorithm for discovering clusters in large spatial databases with noise," Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), 1996, pp. 226-231.
15. Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. "From Data Mining to Knowledge Discovery in Databases," *AI Magazine* (17:3), 1996, pp. 37-54.
16. Forti, A., and Foresti, G.L. "Growing Hierarchical Tree SOM: An unsupervised neural network with dynamic topology," *Neural Networks* (19:10), 2006, pp. 1568-1580.
17. Fritzke, B. "Unsupervised clustering with growing cell structures," Neural Networks, IJCNN-91-Seattle International Joint Conference on, 1991, pp. 531-536.
18. Fritzke, B. "Kohonen Feature Maps and Growing Cell Structures - a Performance Comparison," *Advances in neural information processing system* (5), 1993, pp. 1-8.
19. Fritzke, B. "Growing cell structures - A self-organizing network for unsupervised and supervised learning," *Neural Networks* (7:9), 1994, pp. 1441-1460.
20. Fritzke, B. "Growing Grid - A self-organizing network with constant neighborhood range and adaption strength " *Neural Processing Letters* (2:5), 1995, pp. 1-5.
21. Global Product Classification (available online at http://www.gs1.org/gdsn/gpc).
22. Haiying, W., Azuaje, F., and Black, N. "An integrative and interactive framework for improving biomedical pattern discovery and visualization," *Information Technology in Biomedicine, IEEE Transactions on* (8:1), 2004, pp. 16-27.
23. Han, J., and Kamber, M. *Data Mining: Concepts and Techniques*, (2nd ed.), Morgan Kaufmann, San Francisco, 2006.
24. Hodge, V.J., and Austin, J. "Hierarchical growing cell structures: TreeGCS," *Knowledge and Data Engineering, IEEE Transactions on* (13:2), 2001, pp. 207-218.
25. Hsu, A., and Halgamuge, S.K. "Class structure visualization with semi-supervised growing self-organizing maps," *Neurocomputing* (71:16-18), 2008, pp. 3124-3130.
26. Hsu, C.-C. "Generalizing self-organizing map for categorical data," *Neural Networks, IEEE*

Transactions on (17:2), 2006, pp. 294-304.

27. International Classification of Diseases (available online at http://www.cdc.gov/nchs/icd.htm).
28. Kaban, A., and Girolami, M. "A combined latent class and trait model for the analysis and visualization of discrete data," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (23:8), 2001, pp. 859-872.
29. Kohonen, T. "The self-organizing map," *Proceedings of the IEEE* (78:9), 1990, pp. 1464-1480.
30. Kohonen, T. *Self-organizing maps*, Springer-Verlag, Berlin, 1995.
31. Kohonen, T., Hynninen, J., Kangas, J., and Laaksonen, J. "SOM_PAK: The Self-Organizing Map Program Package," Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996.
32. Kosara, R., Bendix, F., and Hauser, H. "Parallel Sets: interactive exploration and visual analysis of categorical data," *Visualization and Computer Graphics, IEEE Transactions on* (12:4), 2006, pp. 558-568.
33. Nabney, I.T., Sun, Y., Tino, P., and Kaban, A. "Semisupervised learning of hierarchical latent trait models for data visualization," *Knowledge and Data Engineering, IEEE Transactions on* (17:3), 2005, pp. 384-400.
34. Nurnberger, A., and Detyniecki, M. "Externally growing self-organizing maps and its application to e-mail database visualization and exploration," *Applied Soft Computing* (6:4), 2006, pp. 357-371.
35. Palmer, C.R., and Faloutsos, C. "Electricity Based External Similarity of Categorical Attributes," in: *Advances in Knowledge Discovery and Data Mining*, 2003, pp. 565-565.
36. Rauber, A., Merkl, D., and Dittenbach, M. "The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data," *Neural Networks, IEEE Transactions on* (13:6), 2002, pp. 1331-1341.
37. Sammon, J.W., Jr. "A Nonlinear Mapping for Data Structure Analysis," *Computers, IEEE Transactions on* (C-18:5), 1969, pp. 401-409.
38. Tsypkin, Y.Z. *Foundations of the theory of learning systems*, Academic Press, New York, 1973.
39. Vathy-Fogarassy, A., and Abonyi, J. "Local and global mappings of topology representing networks," *Information Sciences* (179:21), 2009, pp. 3791-3803.
40. Wang, H., Azuaje, F., and Black, N. "Improving biomolecular pattern discovery and visualization with hybrid self-adaptive networks," *NanoBioscience, IEEE Transactions on* (1:4), 2002, pp. 146-166.