Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 35 (2008) 1177–1185
Expert Systems with Applications
www.elsevier.com/locate/eswa

Incremental clustering of mixed data based on distance hierarchy

Chung-Chian Hsu [a], Yan-Ping Huang [a,b,*]

[a] Department of Information Management, National Yunlin University of Science and Technology, Taiwan
[b] Department of Information Management, Chin Min Institute of Technology, Taiwan

Abstract

Clustering is an important function in data mining. Its typical application includes the analysis of consumer's materials. Adaptive resonance theory network (ART) is very popular in the unsupervised neural network. Type I adaptive resonance theory network (ART1) deals with the binary numerical data, whereas type II adaptive resonance theory network (ART2) deals with the general numerical data. Several information systems collect the mixing type attitudes, which included numeric attributes and categorical attributes. However, ART1 and ART2 do not deal with mixed data. If the categorical data attributes are transferred to the binary data format, the binary data do not reflect the similar degree. It influences the clustering quality. Therefore, this paper proposes a modified adaptive resonance theory network (M-ART) and the conceptual hierarchy tree to solve similar degrees of mixed data. This paper utilizes artificial simulation materials and collects a piece of actual data about the family income to do experiments. The results show that the M-ART algorithm can process the mixed data and has a great effect on clustering.

Keywords: Adaptive resonance theory network; Conceptual hierarchy; Clustering algorithm; Unsupervised neural network; Data mining

1. Introduction

Clustering is the unsupervised classification of patterns into groups. It is an important data analyzing technique, which organizes a collection of patterns into clusters based on similarity (Hsu, 2006; Hsu & Wang, 2005; Jain & Dubes, 1988). Clustering is useful in several exploratory pattern-analysis, grouping, decision-making, and machine-learning situations. This includes data mining, document retrieval, image segmentation, and pattern classification. Clustering methods have been successfully applied in many fields including pattern recognition (Anderberg, 1973), biology, psychiatry, psychology, archaeology, geology, geography, marketing, image processing (Jain & Dubes, 1988) and information retrieval (Rasmussen, 1992; Salton & Buckley, 1991). Intuitively, patterns with a valid cluster are more similar to each other than they are to a pattern belonging to a different cluster.

Data clustering has been considered as a primary data mining method for knowledge discovery. There have been many clustering algorithms in the literature. In general, major clustering methods can be classified into the hierarchical or the partition category. A hierarchical method creates a hierarchical decomposition of the given set of data patterns. A partition approach produces k partitions of the patterns, where each partition represents a cluster. Further classification in each of the categories is possible (Jain & Dubes, 1988). In addition, Jian (1999) discussed some cross-cutting issues that might affect all of the different approaches regardless of their placement in the categories (Jain, Murty, & Flynn, 1999). Being non-incremental or incremental is one of the issues (Hsu, 2006; Hsu & Wang, 2005). Non-incremental clustering methods process all the data patterns at a time. These algorithms usually require the entire datasets being loaded into memory and therefore have high requirement in memory space.

The major advantage with the incremental clustering algorithms is that it is not necessary to store the entire

* Corresponding author. Address: Department of Information Management, National Yunlin University of Science and Technology, Taiwan. Tel.: +886 37627153; fax: +886 37605684.
E-mail addresses: hsucc@yuntech.edu.tw (C.-C. Hsu), sunny@ms.chinmin.edu.tw (Y.-P. Huang).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.08.049

pattern matrix in the memory. So, the space requirements of incremental algorithms are very small. Incremental clustering considers input patterns one at a time and assigns them to the existing clusters (Jain & Dubes, 1988). Here, a new input pattern is assigned to a cluster without affecting the existing clusters significantly. Moreover, a major advantage of the incremental clustering algorithms is their limited space requirement since the entire dataset is not necessary to store in the memory. Therefore, these algorithms are well suited for a dynamic environment and for very large datasets. They have already been applied along these directions (Can, 1993; Ester, Kriegel, Sander, Wimmer, & Xu, 1998; Somlo & Adele, 2001).

Most of clustering algorithms consider either categorical data or numeric data. However, many mixed datasets including categorical and numeric values existed nowadays. A common practice to clustering mixed dataset is to transform categorical values into numeric values and then proceed to use a numeric clustering algorithm. Another approach is to compare the categorical values directly, in which two distinct values result in distance 1 while identical values result in distance 0. Nevertheless, these two methods do not take into account the similarity information embedded between categorical values. Consequently, the clustering results do not faithfully reveal the similarity structure of the dataset (Hsu, 2006; Hsu & Wang, 2005).

This article is based on distance hierarchy (Hsu, 2006; Hsu & Wang, 2005) to propose a new incremental clustering algorithm for mixed datasets, in which the similarity information embedded between categorical attribute is considered during clustering. In our setting, each attribute of the data is associated with a distance hierarchy, which is an extension of the concept hierarchy (Somlo & Adele, 2001) with link weights representing the distance between concepts. The distance between two mixed data patterns is then calculated according to distance hierarchies.

It is worth mentioning that the representation scheme of distance hierarchy can generalize some conventional distance computation schemes including the simple matching and the binary encoding for categorical values, and the subtraction method for continuous values and ordinal values.

The rest of this article is organized as follows. Section 2 reviews clustering algorithms and discusses the shortcomings of the conventional approaches to clustering mixed data. Section 3 presents distance hierarchy for categorical data and proposes the incremental clustering algorithm based on distance hierarchies. In Section 4, experimental results on synthetic and real datasets are presented. Conclusions are given in Section 5.

2. Literature review

Adaptive resonance theory neural networks model real-time prediction, search, learning, and recognition. ART networks function as models of human cognitive information processing (Carpenter, 1997; Carpenter & Grossberg, 1993; Grossberg, 1980, 1999, 2003). A central feature of all ART systems is a pattern matching process that compares an external input with the internal memory of an active code. ART1 deals with the binary numerical data and ART2 deals with the general numerical data (Grossberg, 1999). However, these two methods do not deal with mixed data attributes.

About clustering mixed data attributes, there are two approaches for mixed data. One is resorted to a pre-process, which transferred the data to the same type, either all numeric or all categorical. For transferring continuous data to categorical data, some metric function is employed. The function is based on simple matching in which two distinct values result in distance 1, with identical values of distance 0 (Guha, Rastogi, & Shim, 1999). The other is to use a metric function, which can handle mixed data (Wilson & Martinez, 1997). Overlap metric is for nominal attributes and normalized Euclidean distance is for continuous attributes.

Among problems with simple matching and binary encoding, a common approach for handling categorical data is *simple matching*, in which comparing two identical categorical values result in distance 0, while two distinct values result in distance 1 (Ester et al., 1998; Wilson & Martinez, 1997). In this case, the distance between patterns of Gary and John in the previous example becomes $d(\text{Gary}, \text{John}) = 1$, which is the same as $d(\text{John}, \text{Tom}) = d(\text{Gary}, \text{Tom}) = 1$. Obviously, the simple matching approach disregards the similarity information embedded in categorical values.

Another typical approach to handle categorical attributes is to employ binary encoding that transforms each categorical attribute to a set of binary attributes and a categorical value is then encoded to a set of binary values. As a result, the new relation contains all numeric data, and the clustering is therefore conducted on the new dataset. For example, as the domain of the categorical attribute: Favorite_Drink. The set of it is {Coke, Pepsi, Mocca}. Favorite_Drink is transformed to three binary attributes: Coke, Pepsi and Mocca in the new relation. The value Coke of Favorite_Drink in a pattern is transformed to a set of three binary values in the new relation, i.e. $\{\text{Coke} = 1, \text{Pepsi} = 0, \text{Mocca} = 0\}$. The Manhattan distance of patterns Gary and John is $d_{\text{M}}(\text{Gary}, \text{John}) = 2$, which is the same as $d_{\text{M}}(\text{Gary}, \text{Tom})$ and $d_{\text{M}}(\text{John}, \text{Tom})$, according to the new relation. Traditional clustering algorithm transfers Favorite_Drink categorical attributes into a binary numerical attribute type as shown in Fig. 1.

After transformation, each original categorical attribute handled by the binary encoding approach contributes as twice as that by the simple matching approach, as shown in the above example of distance (Gary, John). Consequently, when the binary encoding approach is adopted by a clustering algorithm, categorical attributes have larger influence on clustering data than those adopting the simple matching approach.

The ART network is a popular incremental clustering algorithm (Jain & Dubes, 1988). It has several variants

ID	Favorite Drink	Amt.
Gary	Coke	70
John	Pesi	70
Tom	Coffee	70

ID	Coke	Pepsi	Coffee	Amt.
Gary	1	0	0	70
John	0	1	0	70
Tom	0	0	1	70

Fig. 1. Traditional clustering algorithm transfers Favorite_Drink categorical attributes into binary numerical attribute type.

(Carpenter & Grossberg, 1987; Carpenter, Grossberg, & Rosen, 1991), in which ART1 handles only the binary data and ART2 can handle only the arbitrary continuous data. K-prototype (Huang, 1998) is a recent clustering algorithm for mixed data. It transfers categorical data attributes to the binary data format, however, the binary data do not reflect the similar degree. It influences the clustering quality. Therefore, this paper proposes a modified adaptive resonance theory network algorithm and the conceptual hierarchy tree to solve the similar degree of mixed data.

3. Clustering hybrid data based on distance hierarchy

This paper proposes the distance hierarchy tree structure to overcome the expression for similar degree. This distance hierarchy tree algorithm combines the adaptive resonance theory network algorithm and it can be effective with mixed data in data clustering. This section presents distance hierarchy for categorical data and it proposes the incremental clustering algorithm based on distance hierarchies.

3.1. Distance hierarchy

The *distance hierarchy* tree is a concept hierarchy structure. It is also a better mechanism to facilitate the representation and computation of the distance between categorical values. A concept hierarchy consists of two parts: a node set and a link set (Dash, Choi, Scheuermann, & Liu, 2002; Hsu, 2006; Hsu & Wang, 2005; Maulik & Bandyopadhyay, 2002). According to binary encoding approach, it does not reflect the similar degree. However, it influences the clustering quality. Maintenance was difficult when the domain of a categorical attribute changes, because the transformed relation schema also needs to be changed. The transformed binary attributes cannot preserve the semantics of the original attribute. Because of the drawbacks resulting from the binary-encoding approach, this paper uses distance hierarchy to solve the similar degree of mixed data. A concept hierarchy extends with distance weights as shown in Fig. 2.

This paper extends the distance hierarchy structure with link weights. Each link has a weight representing a distance. Link weights are assigned by domain experts. There are several assignment alternatives. The simplest way is to assign all links as a uniform constant weight. Another alternative is to assign heavier weights to the links closer to the root and lighter weights to the links away from the root. For simplicity, unless stated explicitly, each link weight is set to 1 in this article. The distance of two concepts at the leaf nodes is the total weight between those two nodes.

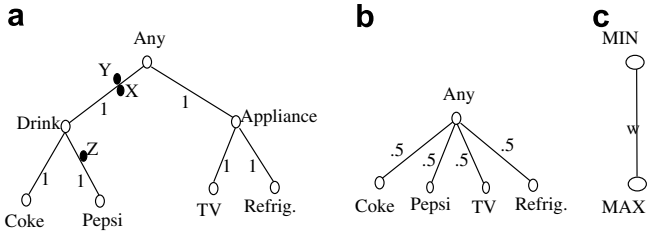

Fig. 2. (a) A distance hierarchy with weight 1, (b) two-level distance hierarchy for simple matching approach, and (c) degenerated distance hierarchy with $w = (\max - \min)$ for a numeric attribute.

A point X in a distance hierarchy consists of two parts, an *anchor* and a positive real-value *offset*, denoted as $X(N, d)$, that is, $\text{anchor}(X) = N$ and $\text{offset}(X) = d$. The anchor is a leaf node and the offset represents the distance from the root of the hierarchy to the point. A point X is an *ancestor* of Y if X is in the path from Y to the root of the hierarchy. If neither one of the two points is an ancestor of the other point, then the *least common ancestor*, denoted as $\text{LCA}(X, Y)$, is the deepest node that is an ancestor of X as well as Y.

Let $X(N_X, d_X)$ and $Y(N_Y, d_Y)$ be two points, the distance between X and Y can be defined as

$$|X - Y| = d_X + d_Y - 2d_{\text{LCP}(X,Y)} \tag{1}$$

where $\text{LCP}(X, Y)$ is the *least common point* of X and Y in the distance hierarchy. $d_{\text{LCP}(X,Y)}$ is the distance between the least common point and the root. The distance between two equivalent points is 0, i.e. $|X - Y| = 0$. For example, $W = (\text{Coke}, 1.4)$, $X = (\text{Coke}, 0.3)$, $Y = (\text{Pepsi}, 0.3)$ and $Z = (\text{Pepsi}, 1.4)$. Both X and Y are ancestors of W and Z. X is equivalent to Y, and their distance is 0. Both X and Y are the least common points of these two points. They are the least common points of Y and W, as well as those of Y and Z. The distance between nodes Any and Drink is 1. The distance between Y and Z is $|1.4 + 0.3 - 2 * 0.3| = 1.1$. The least common point, $\text{LCP}(W, Z)$, of W and Z is the node Drink, which is also the least common ancestor of these two points. Therefore, $d_{\text{LCP}(W,Z)} = 1$. Furthermore, the distance between W and Z is $|1.4 + 1.4 - 2 * 1| = 0.8$.

A special distance hierarchy calls *numeric distance hierarchy* for a numeric attribute, say $\mathbf{x}_i$, is a degenerate one, which consists of only two nodes, a root MIN and a leaf MAX (e.g. Fig. 2c), and has the link weight w being the domain range of $\mathbf{x}_i$, i.e. $w = (\max_i - \min_i)$. A point p in such a distance hierarchy has the value (MAX, d_p) where the anchor is always the MAX and the offset d_p is the distance from the point to the root MIN.

About measuring distance, the distance between two data points can be measured as follows: Let $x = [x_1 x_2 \ldots x_n]$ and $y = [y_1 y_2 \ldots y_n]$. The distance between a

training pattern **x** and an M-ART neuron **y** is measured as the square root of the sum of the square differences between each-paired components of **x** and **y**. Specifically, **x** and**y** represent a training data and a map neuron, respectively, with *n*-dimension, and *C* is a set of n distance hierarchies, then the distance between **x** and **y** can be expressed as

$$d(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\| = \left(\sum_{i=1,n} w_i (x_i - y_i)^2\right)^{1/2} = \left(\sum_{i=1,n} w_i (h(x_i) - h(y_i))^2\right)^{1/2} \quad (2)$$

where $h(x_i)$ and $h(y_i)$ are the mapping of x_i and y_i to their associated distance hierarchy h_i and w_i, the attribute weight, is a user specified parameter allowing the domain expert to give different weights. For a numeric attribute I, $h(x_i) - h(y_i)$ is equal to $x_i - y_i$, since $h(x_i) - h(y_i) = (\mathrm{MIN}, d_{h(xi)}) - (\mathrm{MIN}, d_{h(yi)}) = (\mathrm{MIN}, x_i - \mathrm{min}_i) - (\mathrm{MIN}, y_i - \mathrm{min}_i) = (x_i - y_i)$.

The attribute weight w_i can be used to remedy the *mixed depth effect* of distance hierarchies, especially when all the attribute domains are normalized to a small range, such as [0 1]. For example, the difference between any two distinct values of an attribute with a two-level distance hierarchy, like Fig. 2b, is always 1 while the difference between two distinct values of a three-level distance hierarchy can be the 0.5 (minimum) or 1 (maximum). Therefore, in a multidimensional data set with attributes associated with various depths of distance hierarchies, attributes with shallow distance hierarchies tend to dominate the distance computation. Decreasing the weight of attributes with a shallow hierarchy or in contrast, increasing the weight of an attribute with a deep hierarchy can alleviate the mixed depth effect.

3.2. Incremental clustering of hybrid data

The unsupervised clustering algorithm has two layers, the input layer and output layer. There is one distance tree. As shown in Fig. 3, each layer is explained as follows.

The input layer is used for receiving the data or the vector. In the input layer, each neuron corresponds to the input vector. An input vector can be a numerical attribute or a categorical attribute. The output layer presents the clustering result. Each neuron represents a clustering result. Each related prototype vector is the representative of a clustering vector.

Particularly, supposes $D = \{1,2,\ldots,n\}$ is the training dataset. Input neurons $\langle x_1, x_2, \ldots, x_n\rangle$ receive the input data vector. A neuron x_k represents an input vector, an attribute or a variable. The sets of distance tree are $\{\mathrm{dt}_1, \mathrm{dt}_2, \ldots, \mathrm{dt}_n\}$, each dt_i expresses the training dataset's attribute x_i. It represents the concept distance in the special and general relation.

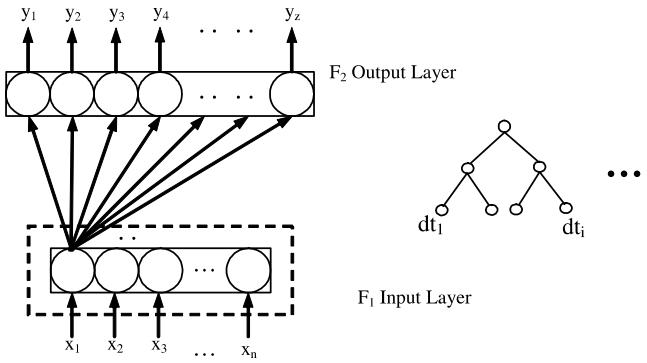

Fig. 3. Modified adaptive neural network architecture.

A neuron $\mathbf{y}_i$ of the output layer is a vector. $\mathbf{y}_i = [y_{1i}, y_{2i}, \ldots, y_{ni}]$. The neuron $y_{ki} = (N, d)$ is composed of two parts: N is a symbol and d is a real number. When y_{ki} corresponds to a categorical attribute, N corresponds to a categorical attribute value. When y_{ki} corresponds to a numerical attribute, N corresponds to a numerical symbol.

As the traditional self-organizing maps and adaptive neural network, the vector has two functions. One is the vector in the output layer and the other is mapped each other in the training dataset. Each dt_i in the distance tree relates to the attribute x_i in the training data and the neuron attribute y_{ik} in the output layer. The algorithm has the following steps:

Input: N records in the training datasets, which include distance hierarchy sets, warning value and stopping parameter value.
Output: Z neurons (the prototype of clustering).
Step 1. Read the first record and map it into the first neuron's vector.
Step 2. Read the next record until the record is empty. Find out the most similar output neuron. If similar degrees exceed the warning value, then join this record to the group and adjust the vector else set up a new neuron. And set the vector as a new neuron's vector.
Step 3. Meet the stopping conditions and then stop else repeat Step 2 until the record is empty.

3.3. Evaluating clustering results

About time complexity, the main decisive factors of the complexity of time in the training algorithm are the training data record N, data dimension P, output neuron number Z and train round C. For each round in the training data, the vectors compare all the attributes with the output neurons. So the time complexity of the whole process is $O(C * N * Z * P)$.

The other method is the significance test on external variables. This technique compares the clusters on variables, but it does not generate. One way of doing this is to compute the expected entropy of the clusters using a variable, say a class attribute C that does not participate in the

clustering. The expected entropy of an attribute C in a set of clusters can be computed as follows. First, it calculates the entropy of an attribute C in each cluster. Then it summates all the entropies weighted by its cluster size. The relationship is:

$$\overline{E}(\check{C}) = -\sum_k \left(\frac{|C_k|}{|D|} \sum_j P(C = V_j) \log P(C = V_j) \right) \quad (3)$$

where V_j denotes one of the possible values that the attribute C can take, $|C_k|$ is the size of the cluster k, and |D| is the size of the data set.

The categorical utility function (Gluck & Corter, 1985) attempts to maximize the probability that the two objects in the same cluster have attribute values in common and the probability that the objects from different clusters have different attributes. The categorical utility of a set of clusters can be calculated as

$$\mathrm{CU} = \sum_k \left(\frac{|C_k|}{|D|} \sum_i \sum_j [P(A_i = V_{ij}|C_k)^2 - P(A_i = V_{ij})^2] \right) \quad (4)$$

Here, $P(A_i = V_{ij}|C_k)$ is the conditional probability that the attribute i has the values V_{ij} given the cluster C_k, and $P(A_i = V_{ij})$ is the overall probability of the attribute i having the values V_{ij} in the entire set. The function aims to measure if the clustering improves the likelihood of similar values falling in the same cluster. Obviously, the higher the CU values, the better the clustering result (Barbara, Couto, & Li, 2002).

For numeric attributes, the standard deviation represents the dispersion of values. Variance (σ^2) can be used for evaluating the quality of clustering numeric data. Here, it can sum up the respective variance of every numeric attribute in all the clusters to evaluate the quality of clustering. The method of calculating the variance is shown in Eq. (5), where $V^k_{i,\mathrm{avg}}$ and $V^k_{i,j}$ are the average and the jth record value of attribute i in cluster k, respectively. Incidentally, the attribute values have been normalized before clustering. Apparently, the lower the variance values, the better the clustering results:

$$\sigma^2 = \sum_k \frac{1}{|C_k|} \sum_i \sum_j (V^k_{i,j} - V^k_{i,\mathrm{avg}})^2 \quad (5)$$

Several cluster validity indices, such as Davies–Bouldin (DB) Index and Calinski Harabasz (CH) Index (Halkidi, Batistakis, & Vazirgiannis, 2001; Maulik & Bandyopadhyay, 2002), have been published; however, they are only suitable for the numeric data. Hence, in order to evaluate the effectiveness of clustering mixed data, this paper uses CV index (Hsu & Chen, 2007), which combined the category utility (CU) function with variance. The CV is defined as in Eq. (6), where the CU and variance are the validity index for categorical and numeric data, respectively. The higher the CV values, the better the clustering result:

$$\mathrm{CV} = \frac{\mathrm{CU}}{1 + \mathrm{Variance}} \quad (6)$$

4. Experiments and discussion

This paper develops a prototype system with Borland C++ Builder 6. A series of experiments have been performed in order to verify the method. A mixed synthetic dataset and a UCI dataset have also been designed to show the capability of the M-ART in reasonably expressing and faithfully preserving the distance between the categorical data. It also reports the experimental results of artificial and actual data.

4.1. Synthetic data sets

The synthetic dataset consisted of three categorical attributes: Sex, Department and Product. The two numeric attributes are Age and Amt. One class level attribute is College. The total records are 600. Table 1 shows all distributions of synthetic categorical dataset. Fig. 4 shows the concept hierarchies for the synthetic categorical dataset.

The M-ART parameters are established as follows: The initial warning value is 0.5 and it increases progressively by 0.05 until 0.7. The initial learning rate is 0.4–0.6. The stop condition is occurs when the momentum of the output layer is lower than 0.000015. The input sequence influences the performance of M-ART and ART2 algorithms. Therefore M-ART uses five different introductory orders. The experimental results are shown in Table 2. In the K-prototype method, the initial central points are influenced by clustering. There are five experiments by randomly choosing one central point. Each parameter in ART2 algorithm is established $a = 0.1$, $b = 0.1$, $c = 0.1$, $d = 0.9$, theta = 0.1–0.2, rho = 0.755–0.76, bottom to top weight = 2.

The experimental result is shown in Table 2. The entropy values can be used for evaluating the quality of

Table 1
Synthetic dataset

Sex	Age	Amt	Department	Product	College	Count
F(50%) M(50%)	$N(20, 2)$	70–90	EE	Coke	Engineering	100
			ME	Pepsi		100
			ID	Bread	Design	100
			VC	Rice		100
			MBA	Apple	Management	100
			MIS	Orange		100

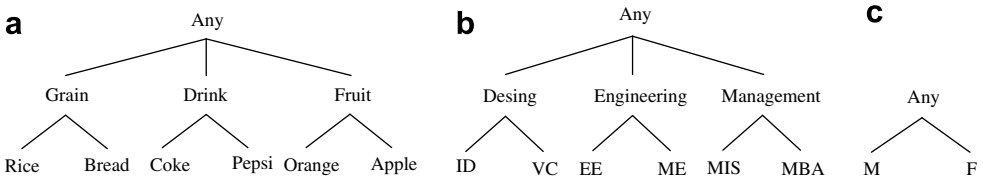

Fig. 4. The concept hierarchies for the synthetic datasets.

Table 2
Entropy values for Student dataset with clusters by ART2, K-prototype and M-ART

Order	ART2	K-prototype	M-ART
1	0.47	0.44	0
2	0.42	0.38	0
3	0.43	0.38	0
4	0.42	0.37	0
5	0.42	0.39	0
Mean	0.43	0.39	0

clustering. The value is small and the quality of clustering is good. In these five experiments, entropy values are zero in M-ART. That means that the classification value in each group is the same. That is to say, the datasets are divided into three groups such as Engineering, Design, and Management. The experimental results show that, in the same input order, the quality of clustering M-ART method is the highest, K-prototype method is second, and traditional ART2 is the lowest. ART2 and K-prototype are unable to divide the datasets into three groups.

4.2. UCI adult data

These experiments use a real Adult dataset from the UCI repository [Murphy 1992] with 48,842 records of 15 attributes, including eight categorical attributes, six numerical attributes, and one class attribute.

This experiment uses seven attributes, which include three categorical attributes, such as Relationship, Marital_status, and Education; and four numeric attributes, Capital_gain, Capital_loss, Age, and Hours_per_week. The concept hierarchies are constructed in Fig. 5. The M-ART parameters are established as follows: the initial warning value is 0.55 and it increases progressively 0.05 until 0.75. The initial learning rate is 0.9. The stop condition t occurs when the momentum of the output layer is lower than 0.000015.

4.3. Experiments and discussion

This paper collects the dataset with different methods. These methods divide adult datasets into 5, 6, 7 and 8 groups. Concerning the CU values for categorical attributes, the higher the CU values, the better the clustering result. The CU value of clustering M-ART method is the highest, K-prototype method is second, and traditional ART2 is the lowest. The symbol "***" means that it does not find the suitable parameter to divide into group with the datasets. The parameter of ART2 reaches 7, it is unable to divide seven groups all the time. The problem occurs because there are too many parameters in ART2. Table 3 shows the CU values of the clustering results by M-ART, ART2 and K-prototypes of each categorical attribute on level 1 and the leaf level in individual concept hierarchies with cluster numbers 5, 6, 7 and 8. The parameter of M-ART is established as follows: the initial warning value is 0.53–0.58, and the initial learning rate is 0.7.

This paper normalizes the variance between 0 and 1 for numeric results. The normalized variance is useful in CV index. Table 4 shows the CV values of the clustering results by M-ART, ART2 and k-prototypes on level 1 and the leaf level in individual concept hierarchies with cluster numbers 5, 6, 7 and 8. The higher the value of CV values, the better the clustering result. The CV value in M-ART method is

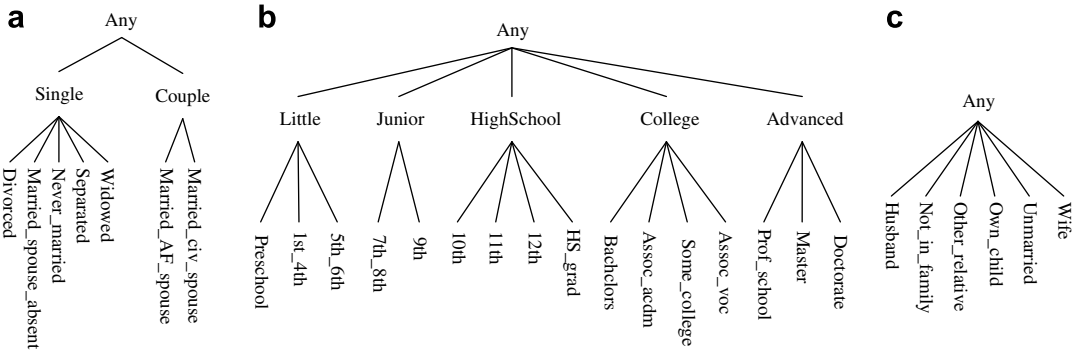

Fig. 5. Concept hierarchies for (a) Marital-status, (b) Education and (c) Relationship attributes of the Adult dataset.

Table 3
The CU values for Adult dataset with 5, 6, 7 and 8 clusters by M-ART, ART2 and K-prototypes

Clusters	Leaf_Level	Level 1	Increased (%)
M-ART			
5	1.069	1.16	8.5
6	1.113	1.20	7.8
7	1.115	1.21	8.5
8	1.177	1.31	11.3
K-prototype			
5	0.859	0.834	−2.9
6	1.002	0.977	−2.5
7	1.039	0.919	−11.6
8	1.088	1.087	−0.3
ART2			
5	0.001	0.00081	−19
6	0.0043	0.0057	3.3
7	***	***	***
8	0.0073	0.00757	3.7

the highest, K-prototype method is second, and traditional ART2 is the lowest.

After M-ART clustering, each clustering has a prototype vector to represent the characteristic of the group. Table 5 shows the result. There is a binary set in a prototype vector. The binary set includes the anchor and Offset. The anchor shows the mode of this field in this group. The Offset is the distance vector to the root. The offset means that a rough proportion of response is accounted for all number values. The higher the value of Offset values, the bigger the mode proportion of the anchor result. For example the fourth group of anchors is HS-grad. The Offset value is 0.84 (Offset value regularized between 0 and 1). The proportion of HS-grad is 73%. Further, the anchor of 2, 3, 6 and 8 groups is all HS-grad. The Offset value approaches zero. It expresses the HS-grad node near the root. Table 6 shows the result. As an example in fifth group with Marital-status and Relationship, the Offset value is quite big. The populations of Married-civ-spouse and Husband of this group are 99% and 89% separately. The result shows each prototype vector can reflect characteristics of each group.

For example, the fourth group of anchors is HS-grad. The Offset value is 0.84 (Offset value is regularized between 0 and 1). The proportion of HS-grad is 73%. Further, the anchor of 2, 3, 6 and 8 groups is all HS-grad. The Offset value approaches zero. It expresses the HS-grad node near the root. Table 6 shows the result. As an example in the fifth group with Marital-status and Relationship, the Offset value is quite big. The populations of Married-civ-spouse and Husband of this group are 99% and 89%, respectively. The result shows that each prototype vector can reflect the characteristics of each group.

The prototype vector reflects the characteristic in this group. A comparison with these prototype vectors and Salary attribute of each group can obtain some characteristics about Salary attribute. Table 5 shows that the salary was over 50k in an order. In the fifth group, all values in the Salary attribute is over 50k. In the first group, the population is 1%. From these prototype vectors, the characteristics in the high-income group can be understood. The salary is over 50k; the proportion is big in 2, 3 and 5 groups. The characteristics are steady marriage states and family relationships, the age is generally relatively longer, and the working hours are relatively longer. The value of Gain and Loss are relatively bigger. Contrarily, the proportion is small in the 1, 4, 6 and 8 groups. The characteristic is not married or had divorced, the age is younger, and working hours is relatively lower. The value of Gain and Loss are relatively small. It can find out the obvious differences in groups 1 and 5.

The comparison with K-prototype and M-ART, K-prototype can present the prototype vector of each group. In the categorical attribute, K-prototype is a representation of the mode as the prototype vector but unable to express the weight of this group. The M-ART algorithm uses the Offset value to understand rough population.

Table 4
The CV values for Adult dataset with 5, 6, 7, 8 clusters by M-ART, ART2 and K-prototypes

Clustering number	CU			Variance			CV		Increased (%)
	Leaf_Level	Level 1	Age	Capital gain	Capital_loss	Hours_per_week	Leaf_Level	Level 1	
M-ART									
5	1.069	1.16	0.127	0.022	0.038	0.071	0.85	0.59	31.08
6	1.113	1.2	0.163	0.023	0.044	0.085	0.85	0.60	29.05
7	1.115	1.21	0.182	0.011	0.044	0.1	0.83	0.61	27.05
8	1.177	1.31	0.218	0.014	0.052	0.115	0.84	0.65	23.00
K-prototype									
5	0.859	0.834	0.114	0.033	0.046	0.073	0.68	0.46	32.85
6	1.002	0.977	0.424	0.72	1.454	2.937	0.15	0.16	1.72
7	1.039	0.919	0.515	0.893	1.801	3.635	0.13	0.12	7.01
8	1.088	1.087	0.61	1.068	2.154	4.342	0.12	0.13	5.56
ART2									
5	0.001	0.00081	0.176	0.027	0.042	0.079	0.001	0.001	6.64
6	0.0043	0.0057	0.211	0.033	0.051	0.095	0.003	0.005	55.87
8	0.0073	0.00757	0.281	0.044	0.068	0.128	0.005	0.006	26.71

Table 5
The categorical attribute prototype for Adult dataset with 1–8 clusters by M-ART

C_no	>50k	Education	Marital-status	Relationship	Age	Hours	Gain	Loss
5	100	(Prof-school, 0.34)	(Married-civ-spouse, 0.99)	(Husband, 0.89)	48	51	99,999	0
2	46	(HS-grad, 0.03)	(Married-civ-spouse, 0.99)	(Wife, 0.98)	40	37	1000	131
3	44	(HS-grad, 0)	(Married-civ-spouse, 1)	(Husband, 1)	44	44	1000	131
7	13	(Some-college, 0.5)	(Never-married, 0.61)	(Not-in-family, 1)	38	41	1000	87
4	7	(HS-grad, 0.84)	(Never-married, 0.5)	(Not-in-family, 1)	40	40	0	87
6	6	(HS-grad, 0)	(Divorced, 0.5)	(Unmarried, 1)	40	39	0	44
8	3	(HS-grad, 0.05)	(Never-married, 0.5)	(Other-relative, 1)	33	37	0	44
1	1	(Some-college, 0)	(Never-married, 0.38)	(Own-child, 1)	25	33	0	44

Table 6
The distribution of Education attribute in each cluster

Education	C5	C2	C3	C7	C4	C6	C8	C1
Preschool	0	0	0	0	0	0	0	0
1st–4th	0	0	1	0	1	1	2	0
5th–6th	1	1	1	1	1	1	4	0
7th–8th	0	1	3	1	2	2	3	1
9th	0	1	2	1	2	2	3	1
10th	1	2	2	0	6	4	4	4
11th	0	2	2	0	6	4	6	9
12th	0	1	1	0	2	1	3	3
HS-grad	12	31	33	0	73	38	40	30
Some-college	6	20	19	37	0	22	20	34
Assoc-voc	2	5	5	7	0	5	3	3
Assoc-acdm	0	5	3	7	0	4	2	3
Bachelors	24	20	18	34	0	10	9	10
Masters	13	8	7	2	5	4	0	1
Prof-school	33	2	3	8	1	1	1	0
Doctorate	8	1	2	1	1	1	0	0

In the C7 prototype vector, the value of Education field is (HS-grad, 0.84): Anchor is HS-grad, and the Offset is 0.84. It expresses that the leaf node in the hierarchy tree is near the root node and the distance is 0.84. The distance between leaf node and HS-grad node is 0.16. It shows that HS-grad has taken heavy proportion in this group. In Table 6, the heavy proportion is 73%. Otherwise, the Offset value of Relationship field is about 1. It means that the mode proportions in each group are between 96% and 100%.

5. Conclusions

Most traditional clustering algorithms can only handle either categorical or numeric value. Although some research results have been published for handling mixed data, they still cannot reasonably express the similarities among categorical data. The paper presents a MART algorithm, which can handle mixed dataset directly. The experimental results on synthetic data sets show that the proposed approach can better reveal the similarity structure among data, particularly when categorical attributes are involved and have different degrees of similarity, in which the traditional clustering approaches do not perform well. The experimental results on the real dataset have better performances than other algorithms.

The future work will try to use this method in finding out the pattern rules from a large time series database.

References

Anderberg, M. (1973). *Cluster analysis for applications*. New York: Academic Press.

Barbara, D., Couto, J., & Li, Y. (2002). COOLCAT: An entropy-based algorithm for categorical clustering. In *Proceedings of the 11th international conference on information and knowledge management* (pp. 582–589).

Can, F. (1993). Incremental clustering for dynamic information processing. *ACM Transaction for Information Systems, 11*, 143–164.

Carpenter, G.-A. (1997). Distributed learning, recognition, and prediction by ART and ARTMAP neural networks. *Neural Networks, 10*(8), 1473–1494.

Carpenter, G.-A., & Grossberg, S. (1987). ART2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics: Special Issue on Neural Networks, 26*, 4919–4930.

Carpenter, G.-A., & Grossberg, S. (1993). Normal and amnesic learning, recognition, and memory by a neural model of cortico-hippocampal interactions. *Trends in Neuroscience, 16*(4), 131–137.

Carpenter, G., Grossberg, A.-S., & Rosen, D.-B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks, 4*, 759–771.

Dash, M., and Choi, K., Scheuermann, P., and Liu, H., (2002). Feature selection for clustering – a filter solution. In *IEEE International Conference on Data Mining*, pp. 115–122.

Ester, M., Kriegel, H.-P., Sander, J., Wimmer, M., & Xu, X. (1998). Incremental clustering for mining in a data warehousing environment. In *Proceedings of the 24th international conference on very large data bases (VLDB)* (pp. 323–333).

Gluck, M.-A., & Corter, J.-E. (1985). Information, uncertainty, and the utility of categories. In *Proceedings of the seventh annual conference of the cognitive science society*.

Grossberg, S. (1980). How does a brain build a cognitive code? *Psychological Review, 87*, 1–51.

Grossberg, S. (1999). The link between brain, learning, attention, and consciousness. *Consciousness and Cognition, 8*, 1–44.

Grossberg, S. (2003). How does the cerebral cortex work? Development, learning, attention, and 3D vision by laminar circuits of visual cortex. *Behavioral and Cognitive Neuroscience Reviews, 2*(1), 47–76.

Guha, S., Rastogi, R., & Shim, K. (1999). ROCK: A robust clustering algorithm for categorical attributes. In *Proceedings of the IEEE conference on data engineering* (pp. 512–521).

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems, 17*, 107–145.

Hsu, C.-C., & Chen, Y.-C. (2007). Mining of mixed data with application to catalog marketing. *Expert Systems with Applications, 32*(1), 12–23.

Hsu, C.-C. (2006). Generalizing self-organizing map for categorical data. *IEEE Transactions on Neural Networks, 17*(2), 294–304.

Hsu, C.-C., & Wang, S.-H. (2005). An integrated framework for visualized and exploratory pattern discovery in mixed data. *IEEE Transactions on Knowledge and Data Engineering, 18*(2), 161–173.

Huang, Z. (1998). Extensions to the *k*-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery, 2*(3), 283–304.

Jain, A., & Dubes, R. (1988). *Algorithms for clustering Data*. Englewood Cliffs, NJ: Prentice-Hall.

Jain, A.-K., Murty, M.-N., & Flynn, P.-J. (1999). Data clustering: A review. *ACM Computing Surveys, 31*(3), 264–323.

Maulik, U., & Bandyopadhyay, S. (2002). Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(12), 1650–1654.

Rasmussen, E. (1992). Clustering algorithms. In William B. Frakes & Ricardo Baeza-Yates (Eds.), *Information retrieval: Data structures & algorithms*. Prentice Hall.

Salton, G. & Buckley, C. (1991). Automatic text structuring and retrieval-experiments in automatic encyclopedia searching. In *Proceedings of the 14th international ACM SIGIR conference on research and development in information retrieval* (pp. 21–30).

Somlo, G.-L. & Adele, E.-H. (2001). Incremental clustering for profile maintenance in information gathering web agents. In *Proceedings of the fifth international conference on autonomous agents* (pp. 262–269).

Wilson, D.-R., & Martinez, T.-R. (1997). Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research, 6*, 1–34.