A Two-Step Method for Clustering Mixed Categroical and Numeric Data

Ming-Yi Shih*, Jar-Wen Jheng and Lien-Fu Lai

Department of Computer Science and Information Engineering, National Changhua University of Education, Changhua, Taiwan 500, R.O.C.

Abstract

Various clustering algorithms have been developed to group data into clusters in diverse domains. However, these clustering algorithms work effectively either on pure numeric data or on pure categorical data, most of them perform poorly on mixed categorical and numeric data types. In this paper, a new two-step clustering method is presented to find clusters on this kind of data. In this approach the items in categorical attributes are processed to construct the similarity or relationships among them based on the ideas of co-occurrence; then all categorical attributes can be converted into numeric attributes based on these constructed relationships. Finally, since all categorical data are converted into numeric, the existing clustering algorithms can be applied to the dataset without pain. Nevertheless, the existing clustering algorithms suffer from some disadvantages or weakness, the proposed two-step method integrates hierarchical and partitioning clustering algorithm with adding attributes to cluster objects. This method defines the relationships among items, and improves the weaknesses of applying single clustering algorithm. Experimental evidences show that robust results can be achieved by applying this method to cluster mixed numeric and categorical data.

***Key Words*: Data Mining, Clustering, Mixed Attributes, Co-Occurrence**

1. Introduction

With the amazing progress of both computer hardware and software, a vast amount of data is generated and collected daily. There is no doubt that data are meaningful only when one can extract the hidden information inside them. However, "the major barrier for obtaining high quality knowledge from data is due to the limitations of the data itself" [1]. These major barriers of collected data come from their growing size and versatile domains. Thus, data mining that is to discover interesting patterns from large amounts of data within limited sources (i.e., computer memory and execution time) has become popular in recent years.

Clustering is considered an important tool for data mining. The goal of data clustering is aimed at dividing the data set into several groups such that objects have a high degree of similarity to each other in the same group and have a high degree of dissimilarity to the ones in different groups [2]. Each formed group is called a cluster. Useful patterns may be extracted by analyzing each cluster. For example, grouping customers with similar characteristics based on their purchasing behaviors in transaction data may find their previously unknown patterns. The extracted information is helpful for decision making in marketing.

Various clustering applications [3–12] have emerged in diverse domains. However, most of the traditional clustering algorithms are designed to focus either on numeric data or on categorical data. The collected data in real world often contain both numeric and categorical attributes. It is difficult for applying traditional clustering algorithm directly into these kinds of data. Typically, when people need to apply traditional distance-based clustering

*Corresponding author. E-mail: myshih@cc.ncue.edu.tw algorithms (ex., k-means [3]) to group these types of data, a numeric value will be assigned to each category in this attributes. Some categorical values, for example “low”, “medium” and “high”, can easily be transferred into numeric values. But if categorical attributes contain the values like “red”, “white” and “blue” … etc., it cannot be ordered naturally. How to assign numeric value to these kinds of categorical attributes will be a challenge work.

In this paper, a method based on the ideas to explore the relationship among categorical attributes’ values is presented. This method defines the similarity among items of categorical attributes based on the idea of co-occurrence. All categorical values will be converted to numeric according to the similarity to make all attributes contain only numeric value. Since all attributes has become homogeneous type of value, existing clustering algorithms can be applied to group these mixed types of data without pain. Nevertheless, most of the existing clustering algorithm may have some limitations or weakness in some way. For example, the returned results from k-means may depend largely on the initial selection of centroid of clusters. Moreover, k-means is sensitive to outliers. In this paper, a two-step method is applied to avoid above weakness. At the first step, HAC (hierarchical agglomerative clustering) [3] algorithm is adopted to cluster the original dataset into some subsets. The formed subsets in this step with adding additional features will be chosen to be the objects to be input to k-means in next step. Since every subset may contain several data points, applying chosen subsets as initial set of clusters in k-means clustering algorithm will be a better solution than selecting individual data. Another benefit of applying this strategy is to reduce the influences of outlier, since the outlier will be smoothed by these added features. The results show that this proposed method is a feasible solution for clustering mixed numeric and categorical data.

The rest of this paper is organized as follows. Next section shows the background and related works. Section 3 describes the proposed method for clustering on mixed categorical and numeric data, and the experimental results will be presented on section 4. Section 5 concludes our work.

2. Background

In most clustering algorithms, an object is usually viewed as a point in a multidimensional space. It can be represented as a vector $(x_1 ... x_d)$, a collection of values of selected attributes with d dimensions; and x_i is the value of i-th selected attribute. The value of x_i may be numerical or categorical.

Most pioneers of solving mixed numeric and categorical value for clustering problem is to redefine the distance measure and apply it to existing clustering algorithms. K-prototype [13] is one of the most famous methods. K-prototype inherits the ideas of k-means, it applies Euclidean distance to numeric attributes and a distance function is defined to be added into the measure of the closeness between two objects. Object pairs with different categorical values will enlarge the distance between them. The main shortcomings of k-prototype may fall into followings:

(1) Binary distance is employed for categorical value. If object pairs with the same categorical value, the distance between them is zero; otherwise it will be one. However, it will not properly show the real situation, since categorical values may have some degree of difference. For example, the difference between “high” and “low” shall not equal to the one between “high” and “medium”.

(2) Only one attribute value is chosen to represent whole attribute in cluster center. Therefore, the categorical value with less appearance seldom gets the chance to be shown in cluster center, though these items may play an important role during clustering process. Additionally, since k-prototype inherits the ideas of k-means, it will retain the same weakness of k-means.

Chiu et al. [14] presented a probabilistic model that applies the decrease in log-likelihood function as a result of merging for distance measure. This method improves k-prototype by solving the binary distance problem. Additionally, this algorithm constructs CF-tree [5] to find dense regions to form subsets, and applies hierarchical clustering algorithms on these subsets. Li et al. [15] represents a similarity measure that when two objects have a same categorical value with less appearance in whole data set, greater weight will be assigned to this match. The basic idea is based on Goodall’s similarity measure [16] that the values appearing in uncommon attributes will make greater contributions to the overall similarity among objects. Instead of choosing only one item to represent whole categorical attributes in cluster center, Yin et al. [17] and Ahmad et al. [18] list all items to represent cluster center. The similarity of categorical attributes is calculated based on the proportion of items' appearance. He et al. [19] calculates the distance between clusters and objects based on all numeric and categorical value's distribution. The distance is used to decide which cluster an object will belong to.

The major problem of existing clustering algorithms is that most of them treat every attribute as a single entity, and ignore the relationships among them. However, there may be some relationships among attributes. For example, the person with high incomes may always live in a costly residence, drive luxurious cars, and buy valuable jewelries… and so on. Therefore, in this paper we represent TMCM (a Two-step Method for Clustering Mixed numeric and categorical data) algorithm to solve above problems. The contributions of this proposed algorithm can be summarized as followings:

1. A new idea is presented to convert items in categorical attributes into numeric value based on co-occurrence theory. This method explores the relationships among items to define the similarity between pairs of objects. A reasonable numeric values can be given to categorical items according to the relationship among items.
2. A two-step k-means clustering method with adding features is proposed. K-means's shortcomings can be improved by applying this proposed method.

In the next session, the TMCM algorithm will be introduced.

3. TMCM Algorithm

In order to explore the relationships among categorical items, the idea of co-occurrence is applied. The basic assumption of co-occurrence is that if two items always show up in one object together, there will be a strong similarity between them. When a pair of categorical items has a higher similarity, they shall be assigned closer numeric values. For example in the instance of Table 1, when temperature is "hot", the humidity is always "high"; but when temperature is "cool", the humidity is "medium" or "low". It indicates that the similarity between "hot" and "high" is higher than the one between "cool" and "high". Therefore, "hot" and "high" shall be assigned a more similar numeric value than "cool" and "high".

The TMCM algorithm is based on above observation to produce pure numeric attributes. The algorithm is shown on Figure 1. Table 2 lists a sample data set, and this data set will be used to illustrate the proposed ideas.

The first step in the proposed approach is to read the input data and normalize the numeric attributes' value into the range of zero and one. The goal of this process is to avoid certain attributes with a large range of values will dominate the results of clustering. Additionally, a categorical attribute *A* with most number of items is selected to be the base attribute, and the items appearing in base attribute are defined as base items. This strategy is to ensure that a non-base item can map to multiple base items. If an attribute with fewer items is selected as the base attribute, the probability of mapping several non-based items to the same based items will be higher. In such a case, it may make different categorical items get the same numeric value.

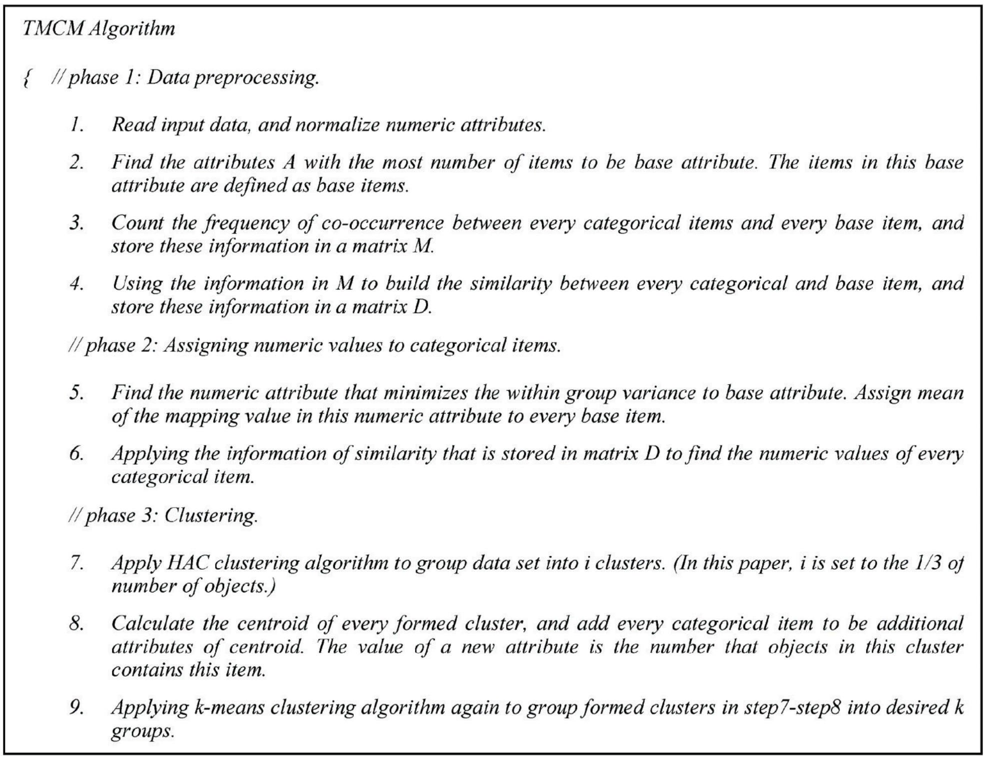
In step 2 of TMCM algorithm, Attribute X will be selected as the base attribute because it contains the most number of items. Item C, D, and E are defined as base items.

After the based attribute is defined, counting the frequency of co-occurrence among categorical items will be operated in this step. A matrix *M* with n columns and n rows is used to store this information,
Where n is the number of categorical items; m_{ij} represents the co-occurrence between item *i* and item *j* in *M*; m_{ii} represents the appearance of item i.

For example, if a matrix *M* is constructed for the data in Table 2, the value of n will be 5 because there are five categorical items. The value of m_{11} is 4 since item A appears in four transactions in Table 2; and the value of m_{13} is 2 because there are 2 transactions in Table 2 containing both item A and item C. Therefore, the matrix *M* will be:

Table 1. An example of co-occurrence

Temperature	Humidity	Windy
hot	high	false
hot	high	false
cool	low	true
cool	normal	true



```
TMCM Algorithm
{ //phase 1: Data preprocessing.
    1. Read input data, and normalize numeric attributes.
    2. Find the attributes A with the most number of items to be base attribute. The items in this base
       attribute are defined as base items.
    3. Count the frequency of co-occurrence between every categorical items and every base item, and
       store these information in a matrix M.
    4. Using the information in M to build the similarity between every categorical and base item, and
       store these information in a matrix D.
  //phase 2: Assigning numeric values to categorical items.
    5. Find the numeric attribute that minimizes the within group variance to base attribute. Assign mean
       of the mapping value in this numeric attribute to every base item.
    6. Applying the information of similarity that is stored in matrix D to find the numeric values of every
       categorical item.
  //phase 3: Clustering.
    7. Apply HAC clustering algorithm to group data set into i clusters. (In this paper, i is set to the 1/3 of
       number of objects.)
    8. Calculate the centroid of every formed cluster, and add every categorical item to be additional
       attributes of centroid. The value of a new attribute is the number that objects in this cluster
       contains this item.
    9. Applying k-means clustering algorithm again to group formed clusters in step7-step8 into desired k
       groups.
```

Figure 1. The TMCM algorithm.

Table 2. A sample data set

Attribute W	Attribute X	Attribute Y	Attribute Z
A	C	0.1	0.1
A	C	0.3	0.9
A	D	0.8	0.8
B	D	0.9	0.2
B	C	0.2	0.8
B	E	0.6	0.9
A	D	0.7	0.1

$$M = \begin{pmatrix} 4 & 0 & 2 & 2 & 0 \\ 0 & 3 & 1 & 1 & 1 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Since the frequencies of co-occurrence between base items and other categorical items is available by retrieving the elements in matrix *M*, the similarity between them can be calculated by adopting following equation:

$$D_{xy} = \frac{|m(X,Y)|}{|m(X)| + |m(Y)| - |m(X,Y)|} \qquad (1)$$

where X represents the event that item x appears in the set of objects; Y represents the event that item y appears in the set of objects; m(X) is the set of objects containing item x; m(X, Y) is the set of objects containing both item x and y.

In equation (1), when two items always show up together in objects, the similarity between them will be one. If two items never appear together, it will get zero for the similarity measure. The higher value of D_{xy} means the more similar between item x and item y. However, only the values of D_{xy} larger than a threshold will be recorded, or zero will be assigned. Now the similarity between every categorical item and every base item is available. For example, the value of |m(A)| is 4 which can be obtained from m_{11} in matrix *M*. Similarly, The value of |m(A, C)| is 2 because m_{13} in matrix *M* is 2. Therefore,

$D_{AC} = 2 / (4 + 3 - 2) = 0.4$,
$D_{AD} = 2 / (4 + 3 - 2) = 0.4$,
$D_{AE} = 0/ (4 + 1 - 0) = 0$.

The first process in phase 2 is to find the numeric attribute that minimizes the within group variance to base attribute. The equation for within group variance will be

$$SS_w = \sum_j \sum_i (X_{ij} - \overline{X_j})^2 \tag{2}$$

where $\overline{X_j}$ is the mean of mapping numeric attribute of *j-th* base item.

X_{ij} is the value of *i-th* value in mapping numeric attribute of *j-th* base item.

Attributes Y in Table 2 is identified to meet this requirement. Then, every base item can be quantified by assigning mean of the mapping value in the selected numeric attribute. For example, the value of item C in Table 2 is (0.1 + 0.3 + 0.2) / 3 = 0.2, item D is 0.7 and item E is 0.6.

Since every base item has been given a numeric value, all other categorical items can be quantified by applying the following function.

$$F(x) = \sum_{i=1}^{d} a_i * v_i \tag{3}$$

where d is the number of base item; a_i is the similarity between item x and i-th base item; v_i is the quantified value of *i-th* base item.

Therefore, item A in Table 2 will be assigned the following value: F(A) = 0.44 * 0.2 + 0.44 * 0.7 + 0 * 0.6 = 0.396.

All attributes in data set will contain only numeric value at this moment, the existing distance based clustering algorithms can by applied without pain. HAC (Hierarchical Agglomerative Clustering) is a widely used hierarchical clustering algorithm. Several HAC algorithms have appeared in the research community. The major difference is the applied similarity criteria. The HAC algorithm takes numeric data as the input and generates the hierarchical partitions as the output. Therefore it is applied in first clustering step to group data into subsets. In HAC, initially each object is considered as a cluster. Then by merging the closest clusters iteratively until the termination condition is reached, or the whole hierarchy is generated. It generates different levels of clusters bottom-up. The algorithm of HAC is presented in Figure 2.

The k-means algorithm takes numeric data as input, and generates crispy partitions (i.e., every object only belongs to one cluster) as the output. It is one of the most popularly used clustering algorithms in the research community. It has been shown to be a robust clustering method in practice. Therefore, the k-means algorithm is applied in second clustering step to cluster data sets. K-means starts by randomly selecting or by specifically picking k objects as the centroids of *k* clusters. Then k-means iteratively assigns the objects to the closest centroid based on the distance measure, and updates the mean of objects in this cluster as the new centroid until reaching a stopping criterion. This stopping criterion could be either non-changing clusters or a predefined number of iterations. The algorithm of HAC is presented in Figure 3.

Because k-means suffers from its shortcomings mentioned in previous section, a two-step method is introduced to improve it. Initially, this proposed method applies HAC to group data set into *i* subsets, where *i* is set to the one-third of number of objects in this paper. Based on the observations of clustering results, these settings of *i* yield satisfied solutions. Each formed subsets will be treated as an input object for applying k-means in next step. The centroid of each subset is calculated to represent whole subset. Moreover, every categorical item will be added to be additional attributes of centroid. The value of a new attribute is the number that objects in this cluster contains this item. For example, there are four ob-

```
1. Calculate the distance between every two objects.
2. View each object as an individual cluster.
3. Merge the closest two clusters
4. Update the distance between clusters.
5. Repeat 3-4 until reaching a stopping criterion or
   generating the whole hierarchy.
```

Figure 2. HAC algorithm.

```
1. Select first k objects randomly as the centroid of each
   cluster.
2. Assign each object to the closest cluster based on
   Euclidean distance or cosine similarity.
3. Update the centroid of each cluster.
4. Repeat steps 2-3 until stopping criterion is reached.
```

Figure 3. K-means algorithm.

jects in one cluster in Table 3; by applying this proposed idea, the seven attributes and their associative values in Table 4 will be added to be additional features of centroid of this cluster in Table 3.

The major benefits of applying two-step clustering method can be summarized as followings:

(1) In first step clustering, several similar objects are grouped into subsets, and these subsets are treated as objects to be input into second step clustering. Thus noise or outlier can be smoothed in k-means clustering process.

(2) The added attributes not only offer useful information for clustering, but also reduce the influence of noise and outlier.

(3) In second clustering step, the initial selections of centroids will be groups of similar objects. It is believed that this strategy will be a better solution than a random selection used in most applications.

After the result of clustering is produced, the entropy will be employed to evaluate the quality. The smaller value of entropy indicates the algorithm has formed clusters where there is a more dominant class for each cluster. This more dominant class can be used to represent a category for the cluster. On the other hand, the larger value of entropy shows the algorithm produces clusters consisting of objects from every class averagely. Therefore, the quality of clustering is considered is worse. Entropy is defined as followings:

$$Entropy = -\sum_{j=1}^{m}((n_j / n) * \sum_{i=1}^{l} P_{ij} * \log(P_{ij})) \quad (4)$$

Table 3. An example in a cluster

temperature	humidity	windy
hot	high	FALSE
hot	high	FALSE
cool	normal	TRUE
mild	normal	TRUE

where m is the number of clusters; l is the number of classes; n_j is the number of data points in cluster j; n is the number of all data points; P_{ij} is the probability that a member of cluster j belongs to class i.

4. Experiment Results

In this session, we present the results of applying TMCM algorithm on three data sets taken from UCI repository (http://archive.ics.uci.edu/ml/datasets.html). The objects in these three data sets have been pre-labeled for the class. Consequently, the quality of clustering can be achieved by applying entropy measure. K-prototype is a well-known and wide used method for clustering mixed categorical and numeric data. Moreover, SPSS Clementine is a popular commercial data mining tool, it adopts the algorithm in [14]. It is significant to compare the quality of clustering produced by our algorithm to theirs.

(1) Contraceptive method choice data set: This data set is collected from 1987 National Indonesia Contraceptive Prevalence Survey. There are 1473 instances in this data set. Every instance contains 2 numeric attributes and 7 categorical attributes. This data set is pre-labeled into 3 classes: no use, long-term methods, or short-term methods.

Table 5 presents the value of entropy of clustering by applying TMCM, k-prototype and Clementine algorithms under several cluster number settings. The first column is the settings of desired cluster number. The second, third and fourth columns show the results of applying TMCM, k-prototype and Clementine algorithms respectively. Table 7 and Table 9 will use the same format of Table 5. Because this data set is pre-labeled into 3 classes,

Table 4. The information about added attributes

hot	cool	Mild	high	normal	FALSE	TRUE
2	1	1	2	2	2	2

Table 5. The value of entropy for clustering on Contraceptive method choice data set

number of clusters	TMCM (A)	K-prototype (B)	Clementine (C)
2	0.649	0.967	0.667
3	0.697	0.955	0.853
4	0.713	0.951	0.824
8	0.806	0.939	0.871

Table 5 will show the result for setting the number of clusters to 3 which will not show on the other two tables.

Table 6 show the improving ratio for TMCM over other two algorithms. The first column is the settings of desired cluster number. The second and third columns show the improving ratio for TMCM over k-prototype and Clementine algorithms respectively. Table 8 and Table 10 will use the same format of Table 6.

(2) Statlog (heart) disease data set: There are 1473 instances in this data set. Every instance contains 5 numeric attributes and 8 categorical attributes. This data set is pre-labeled into 2 classes.

Table 7 presents the value of entropy of clustering by applying TMCM, k-prototype and Clementine algorithms under several cluster number settings. Table 8 shows the improving ratio for TMCM over other two algorithms.

(3) Credit approval data set: This data set is collected from credit card applications. All attributes are encoded to ensure confidentiality of the data. Although the attributes are transferred into meaningless symbols, the proposed method still works well on this case. There are 690 instances in this data set, and these instances are pre-labeled into 2 classes.

Table 9 presents the value of entropy of clustering by applying TMCM, k-prototype and Clementine algorithms under several cluster number settings. Table 10 shows the improving ratio for TMCM over other two algorithms.

From the observations of Table 6, Table 8 and Table

Table 6. The improving ratio for TMCM over k-prototype and Clementine

Number of clusters	The improving ratio for TMCM over k-prototype (B-A)/A	The improving ratio for TMCM over Clementine (C-A)/A
2	49.00%	2.77%
3	37.02%	22.38%
4	33.38%	15.57%
8	16.50%	8.06%

Table 7. The value of entropy for clustering on Statlog (heart) disease data set

number of clusters	TMCM (A)	K-prototype (B)	Clementine (C)
2	0.422	0.443	0.467
4	0.444	0.673	0.562
8	0.574	0.608	0.574

Table 9. The value of entropy for clustering on Credit approval data set

number of clusters	TMCM (A)	K-prototype (B)	Clementine (C)
2	0.649	0.958	0.764
4	0.564	0.904	0.598
8	0.576	0.845	0.871

Table 8. The improving ratio for TMCM over k-prototype and Clementine

Number of clusters	The improving ratio for TMCM over k-prototype (B-A)/A	The improving ratio for TMCM over Clementine (C-A)/A
2	4.97%	10.66%
4	51.58%	26.58%
8	5.92%	0%

Table 10. The improving ratio for TMCM over k-prototype and Clementine

Number of clusters	The improving ratio for TMCM over k-prototype (B-A)/A	The improving ratio for TMCM over Clementine (C-A)/A
2	47.61%	17.72%
4	60.28%	6.03%
8	35.58%	51.22%

10, the proposed TMCM outperform k-prototype by 38.18% averagely; and outperform SPSS Clementine by 16.10% averagely. TMCM algorithm almost outperforms the other two algorithms in all cases except in one. The experimental results show robust evidence that the proposed approach is a feasible solution for clustering mix categorical and numeric data.

5. Conclusion

Clustering has been widely applied to various domains to explore the hidden and useful patterns inside data. Because the most collected data in real world contain both categorical and numeric attributes, the traditional clustering algorithm cannot handle this kind of data effectively. Therefore, in this paper we propose a new approach to explore the relationships among categorical items and convert them into numeric values. Then, the existing distance based clustering algorithms can be employed to group mix types of data. Moreover, in order to overcome the weaknesses of k-means clustering algorithm, a two-step method integrating hierarchical and partitioning clustering algorithms is introduced. The experimental results show that the proposed approach can achieve a high quality of clustering results.

In this paper, the TMCM algorithm integrates HAC and k-means clustering algorithms to cluster mixed type of data. Applying other algorithms or sophisticated similarity measures into TMCM may yield better results. Furthermore, the number of subset *i* is set to one-third of number of objects in this paper. Although experimental results show that it is feasible, how to set this parameter precisely is worth more study in the future.

References

[1] Wiederhold, G., Foreword. In: Fayyad U., Shapiro G. P., Smyth P., Uthurusamy R., editors, Advances in Knowledge Discovery in Databases. California: AAAI/MIT Press, 1996;2.

[2] Han, J. and Kamber, K., Data mining: Concept and Techniques. San Francisco: Morgan Kaufman Publisher (2001).

[3] Jain, A. K. and Dubes, R. C., *Algorithms for Clustering Data*, New Jersey: Printice Hall (1988).

[4] Kaufman, L. and Rousseeuw, P. J., *Finding Groups in Data: An Introduction to Cluster Analysis*, New York: John Wiley & Sons (1990).

[5] Ng, R. and Han, J., *Efficient and Effective Clustering Method for Spatial Data Mining*, Proc. of the 20th VLDB Conf. 1994 September. Santiago, Chile (1994).

[6] Zhang, T., Ramakrishman, R. and Livny, M., *BIRCH: an Efficient Data Clustering Method for Very Large Databases*, Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data, 1996 June. Montreal, Canada (1996).

[7] Guha, S., Rastogi, R. and Shim, K., *Cure: An Efficient Clustering Algorithm for Large Databases*, Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data. 1998 june. Seattle, WA (1998).

[8] Ester, M., Kriegel, H. P., Sander, J. and Xu, X., *A Density-Based Algorithm for Discovering Clusters in Large spatial databases*, Proc. of the Second International Conference on Data Mining (KDD-96), 1996 August. Portland, Oregon (1996).

[9] Hinneburg, A. and Keim, D., *An Efficient Approach to Clustering in Large Multimedia Databases with Noise*, Proc. 1998 Int. Conf. on Data Mining and Knowledge Discovery (KDD'98). 1998 August. New York (1998).

[10] Wang, W., Yang, J. and Muntz, R., *Sting: A Statistical Information Grid Approach to Spatial Data Mining*, Proc. 23rd VLDB. 1997 August. Athens, Greece (1997).

[11] Kaufman, L. and Rousseeuw, P. J., *Finding Groups in Data: An Introduction to Cluster Analysis*, New York: John Wiley & Sons (1990).

[12] Hinneburg and Keim, D., *An Efficient Approach to Clustering in Large Multimedia Databases with Noise*, Proc. 1998 Int. Conf. on Data Mining and Knowledge Discovery (KDD'98). 1998 August. New York (1998).

[13] Huang, Z., "Extensions to the K-Means Algorithm for Clustering Large Data Sets with Categorical Values," *Data Mining and Knowledge Discovery*, Vol. 2, pp. 283–304 (1998).

[14] Chiu, T., Fang, D., Chen, J. and Wang, Y., *A Robust and Scalable Clustering Algorithm for Mixed Type Attributes in Large Database Environment*, Proc. 2001 Int. Conf. On knowledge Discovery and Data Mining. 2001 Auguest. San Fransico (2001).

[15] Li, C. and Biswas, G., "Unsupervised Learning with Mixed Numeric and Nominal Data," *IEEE Transactions on Knowledge and Data Engineering*, Vol . 14, p. 4 (2002).

[16] Goodall, D. W., “A New Similarity Index Based on Probability,” *Biometric*, Vol. 22, pp. 882–907 (1966).

[17] Yin, J., Tan, Z. F., Ren, J. T. and Chen, Y. Q., *An Efficient Clustering Algorithm for Mixed Type Attributes in Large Dataset*, Proc. of the Fourth International Conference on Machine Learning and Cybernetics, 2005 August. Guangzhou China (2005).

[18] Ahmad, L. and Dey, A., “K-Mean Clustering Algorithm for Mixed Numeric and Categorical Data,” *Data & Knowledge Engineering*, Vol. 63, pp. 503–527 (2007).

[19] He, Z., Xu, X. and Deng, S., “Scalable Algorithms for Clustering Mixed Type Attributes in Large Datasets,” *Interbational Journal of Intelligent Systems*, Vol. 20, pp. 1077–1089 (2005).

Manuscript Received: Jan. 8, 2010
Accepted: Mar. 3, 2010