International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 4, Issue 2 (March-April, 2016), PP. 142-147

A METHOD FOR CLUSTERING MIXED CATEGROICAL AND NUMERIC DATA BY TWO-STEPS

Sameera Nerella[1], Lalitha Bhavani.B[2]
[1]Asst.ProfessorDepartment of Information Technology,
Sir C R Reddy College of Engg, Eluru, A.P.India.
sameeracrrit@gmail.com

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.

Index terms- **Data Mining, Clustering, Mixed Attributes, Co-Occurrence,**

I. INTRODUCTION

With the amazing progress of both computer hard-ware and software, a vast amount of data is generated and collected daily. There is no doubt that data are mean-ingful 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 limita-tions of the data itself" [1]. These major barriers of col-lected data come from their growing size and versatile domains. Thus, data mining that is to discover interest-ing 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 dif- ferent groups [2]. Each formed group is called a cluster. Useful patterns may be extracted by analyzing each clus- ter. For example, grouping customers with similar cha- racteristics based on their purchasing behaviors in tran- saction 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 clus- tering 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 attri- butes. It is difficult for applying traditional clustering al- gorithm directly into these kinds of data. Typically, when people need to apply traditional distance-based clustering

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 nu- meric values. But if categorical attributes contain the va- lues 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 be- come homogeneous type of value, existing clustering al- gorithms can be applied to group these mixed types of data without pain. Nevertheless, most of the existing clus- tering 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 cen- troid of clusters. Moreover, k-means is sensitive to out-liers. In this paper, a two-step method is applied to avoid above weakness. At the first step, HAC (hierarchical ag-glomerative 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 cluster- ing algorithm will be a better solution than selecting indi- vidual 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. Sec- tion 3 describes the proposed method for clustering on mixed categorical and numeric data, and the experimen-tal results will be presented on section 4. Section 5 con-cludes our work.

II. BACKGROUND

In most clustering algorithms, an object is usually

viewed as a point in a multidimensional space. It can be represented as a vector (x1...xd), a collection of values of selected attributes with d dimensions; and xi is the

value of i-th selected attribute. The value of xi may be numeri-cal or categorical.

Most pioneers of solving mixed numeric and cate- gorical value for clustering problem is to redefine the distance measure and apply it to existing clustering algo-rithms. K-prototype [13] is one of the most famous met-hods. K-prototype inherits the ideas of k- means, it ap-plies Euclidean distance to numeric attributes and a dis-tance function is defined to be added into the measure of the closeness between two objects. Object pairs with dif-ferent categorical values will enlarge the distance be-tween them. The main shortcomings of k-prototype may fall into followings:

Binary distance is employed for categorical value. If object pairs with the same categorical value, the dis-tance 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".

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. Ad- ditionally, 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. Ad- ditionally, 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] re- presents 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 exam- ple, 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 repre- sent 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:

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 categori- cal items according to the relationship among items.

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 in-troduced.

III. TMCM ALGORITHM

In order to explore the relationships among categori- cal 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 nu- meric 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 "me- dium" 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 as-signed a more similar numeric value than "cool" and

"high".

The TMCM algorithm is based on above observa-tion 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 se- lected 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 se-lected 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 fre-quency 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; mij represents the co-occurrence between item i and item j in M;

mii 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 m11 is 4 since item A appears in four transactions in Table 2; and the

value of m13 is 2 because there are 2 transactions in Table 2 containing both item A and item C. Therefore, the ma-trix 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 retrie- ving the elements in matrix *M*, the similarity between them can be calculated by adopting following equation:

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

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 contain-ing 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 to-gether 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 Dxy means the more similar between item x and item y.

However, only the values of Dxy larger than a threshold will be re-corded, or zero will be assigned. Now the similarity be-tween every categorical item and every base item is available. For example, the value of |m(A)|

is 4 which can be obtained from m11 in matrix *M*. Similarly, The value of |m(A, C)| is 2 because m13 in matrix *M* is 2. Therefore,

DAC = 2 / (4 + 3   2) = 0.4,

DAD = 2 / (4 + 3   2) = 0.4,

DAE = 0/ (4 + 1   0) = 0.

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

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

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

(2) *Xij* is the value of *i-th* value in mapping numeric at-tribute of *j-th* base item.

Attributes Y in Table 2 is identified to meet this re-quirement. Then, every base item can be quantified by assigning mean of the mapping value in the selected nu-meric 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 va- lue, all other categorical items can be quantified by ap- plying the following function.

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

(1) where *d* is the number of base item; *ai* is the similarity between item *x* and i-th base item; *vi* is the quantified value of *i-th* base item.

Therefore, item A in Table 2 will be assigned the fol-lowing 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 cluster- ing algorithms can by applied without pain. HAC (Hier- archical Agglomerative Clustering) is a widely used hi- erarchical clustering algorithm. Several HAC algorithms have appeared in the research community. The major dif- ference is the applied similarity criteria. The HAC algo- rithm takes numeric data as the input and generates the hierarchical partitions as the output. Therefore it is ap- plied 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 termi- nation condition is reached, or the whole hierarchy is generated. It generates different levels of clusters bot-tom- 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 com- munity. It has been shown to be a robust clustering met-hod in practice. Therefore, the k-means algorithm is ap-plied 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 cen-troid 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 men- tioned in previous section, a two-step method is intro- duced to improve it. Initially, this proposed method ap- plies 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 repre- sent whole subset. Moreover, every categorical item will be added to be additional attributes of centroid. The va-lue 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

(3) 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 cen- troid 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 cluster-ing process.
(2) The added attributes not only offer useful informa- tion 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 be- lieved 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 va-lue of entropy indicates the algorithm has formed clus-ters where there is a more dominant class for each clus-ter. This more dominant class can be used to represent a category for the cluster. On the other hand, the larger va-lue of entropy shows the algorithm produces clusters consisting of objects from every class averagely. There-fore, the quality of clustering is considered is worse. En-tropy is defined as followings:

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

(4) where *m* is the number of clusters; *l* is the number of classes; *nj* is the number of data points in cluster *j*; *n* is

the number of all data points; *Pij* is the probability that a member of cluster *j* belongs to class *i*.

IV. EXPERIMENT RESULTS

In this session, we present the results of applying TMCM algorithm on three data sets taken from UCI re- pository (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 Clemen- tine 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.

(5) Contraceptive method choice data set: This data set is collected from 1987 National Indonesia Contra-

ceptive Prevalence Survey. There are 1473 instances in this data set. Every instance contains 2 numeric at-tributes and 7 categorical attributes. This data set is pre-labeled into 3 classes: no use, long-term met-hods, 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 respec-tively. Table 7 and Table 9 will use the same format of Table 5. Because this data set is pre-labeled into 3 classes,

Table 3. An example in a cluster

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

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.

(1) 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 Ta-ble 10 will use the same format of Table 6.

(2) Statlog (heart) disease data set: There are 1473 in-stances in this data set. Every instance contains 5 nu-meric 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 en- coded to ensure confidentiality of the data. Although the attributes are transferred into meaningless sym- bols, 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 pro- posed approach is a feasible solution for clustering mix categorical and numeric data.

V. CONCLUSION

Clustering has been widely applied to various do- mains to explore the hidden and useful patterns inside data. Because the most collected data in real world con- tain both categorical and numeric attributes, the tradi- tional clustering algorithm cannot handle this kind of data effectively. Therefore, in this paper we propose a new approach to explore the relationships among cate-gorical 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 cluster-ing algorithm, a two-step method integrating hierarchi-cal and partitioning clustering algorithms is introduced. The experimental results show that the proposed appro-ach 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 si- milarity 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

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 Pub- lisher (2001).

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

[4] Kaufman, L. and Rousseeuw, P. J., Finding Groups in

[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 At-tributes 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 Transac- tions 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 Effi-cient 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 Algo-rithm 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).