

An immune-based clustering algorithm for large data sets with categorical values *

Zhendong Pan, Jiafu Tang[†], Lifeng Mu

Dept of Systems Engineering, College of Information Sciences and Engineering
Northeastern University (NEU), Shenyang 110004, China

(Received August 5 2005, Accepted December 11 2005)

Abstract. In practical applications of data mining, data sets with categorical values are common everywhere. However, most known clustering algorithms are designed for numerical values only, because of their similarity measures. Few algorithms, such as k -modes, are able to deal with this kind of clustering problems using their own similarity measures for categorical values. In this paper, a new similarity measure is proposed, and a cost function is given based on this new similarity measure. To optimize the cost function, an immune-based algorithm for large data sets with categorical values, short for ICCV, is designed. Experimental results show that ICCV algorithm can obtain a higher accuracy than other algorithms such as k -modes, etc.

Keywords: cluster, categorical values, immune-based algorithms, data mining

1 Introduction

In the real world, there exist volumes of categorical data. How to deal with these categorical data efficiently is an active research area in the field of data mining^[1,8]. Clustering is an important problem to be addressed in data mining. Cluster analysis is a technique that divides data sets into a number of groups (clusters) for the purposes of summarization or improved understanding. It has been used in a number of tasks in data mining, such as unsupervised classification, aggregation and outlier dissection.

As a branch of statistic, cluster analysis has received many attentions since its appearance^[1], and lots of efficient algorithms are proposed. Existing cluster algorithms can be classified as^[8,16]: partition-based algorithms, hierarchical algorithms, density-based algorithms, grid-based algorithms and model-based algorithms. The partition-based algorithms divides data sets into given number of clusters initially and adjust the clusters afterwards. Hierarchical algorithms view every data object in data set as a cluster firstly, and then combine the similar cluster until a satisfactory result is obtained. Density-based algorithms cluster data sets according to data object's density. Grid-based algorithms divide the vector space into consecutive subspace, and treat each subspace as a special data object. Model-based algorithms imitate the distribution of data sets with some mathematical stochastic distribution.

There exist two kinds of data in real life world; one is numerical data and the other is categorical data. The domain of numerical data is continuous, and can either be finite or infinite. Conversely, categorical data is a kind of data that can only take one of the discrete values in a finite domain. It can be found everywhere in daily life. For example, the description of employee in a company may have the attributes like sex, position, address, which are all categorical data. In other words, if categorical data is just simply neglected in cluster analysis, the cluster result will be impractical in real-life world. Nevertheless, most of the current clustering

* The research is supported by the Natural Science Foundation of China (NSFC 70471028, 70431003) together with the Key Project (104064) of Research and Scientific of Ministry of Education (MOE) in China, the program of New Century Excellent Talents in University (NCET-04-280) of MOE of China and Liaoning Provincial Natural Science Foundation (20022019).

[†] E-mail address: jftang@mail.neu.edu.cn.

algorithms are designed and applied to numerical data, while algorithms for categorical data received little attention.

The definition of “similarity” is pivotal for a successful clustering algorithm. Most previous work in cluster analysis focused on data with numerical values whose similarity can be measured by inherent algebraic properties as Euclidian distance, etc. These definitions of distance describe the similarity of numerical data values perfect well, but cannot be directly applied to categorical values. Some researchers proposed many methods to convert categorical values to numerical values, so the above distance definitions can be fit for similarity measurement of categorical values; however, the converted numerical values fail to reflect their corresponding categorical values, thus inevitably lead to meaningless clustering results.

In the past few years, researchers designed a few clustering algorithms that can deal with categorical values, such as, STIRR^[7], ROCK^[5], CACTUS^[6] and Squeezer^[9]. Gibson et al.^[7] introduce an iterative algorithm based on non-linear dynamical systems, short for STIRR. They represent each attribute value as a weighted vertex in a graph. (Edges between vertices-derived from tuples in the dataset are not explicitly maintained.). Multiple copies $b_1, b_2 \dots b_m$, called basins, of this set of weighted vertices are maintained, where b_1 is called the principal basin and $b_2 \dots b_m$ are called non-principal basins. The weights on any given vertex may differ across basins. Starting with a set of weights on all vertices (in all basins), the system is “iterated” until a fixed point is reached. Guha et al.^[5] introduce ROCK, an adaptation of an agglomerative hierarchical clustering algorithm, to heuristically optimize a criterion function defined in terms of the number of “links” between tuples. Informally, the number of links between two tuples is the number of common “neighbors” they have in the dataset. Given a similarity function, two tuples in the dataset are said to be neighbors if the similarity between them is greater than a certain threshold. Starting with each tuple in its own cluster, they repeatedly merge the two “closest” clusters till the pre-specified number (e.g. K) of clusters remains. The closeness between two clusters is defined to be the sum of the number of links between all pairs of tuples-one in each cluster. Since the complexity of this hierarchical algorithm is quadratic in the number of tuples in the dataset, they cluster a sample randomly drawn from the dataset, and then partition the entire dataset based on the clusters from the sample. Beyond that the set of all “clusters” together may optimize a criterion function. Ganti et al.^[6] develop CACTUS to provide a fresh view of cluster for data sets with categorical values. With CACTUS, a cluster is not composed of data objects from the data sets, but a vector space. Whether a data object belongs to a certain cluster is determined by mapping it to the vector space that the cluster represents. He et al.^[9] invented an algorithm called Squeezer. The Squeezer algorithm reads each tuple t in sequence, and then either assign t to an existing cluster (initially none), or to a new cluster, depending the similarities between t and clusters. Due to its characteristics, the proposed algorithm is extremely suitable for clustering data streams. Given a sequence of points, the objective is to maintain consistently good clustering of the sequence so far, using a small amount of memory and time. Outliers can also be handled efficiently and directly in Squeezer.

The aforementioned algorithms can get quite decent clustering results; even though their efficiency is lower than most of clustering algorithms for numerical values. This deficiency is acceptable by many practitioners and researchers, because of the complexity of categorical values similarity measurement.

The algorithm k -means^[8] is both outstanding on efficiency and effectiveness when it applies in many practical scenes. It uses an iteration strategy to improve current cluster centers until a satisfactory result is obtained. However, the k -means algorithm is also based on numerical data only. Huang^[12,13] present an algorithm called k -modes extending k -means to categorical domain. The algorithm k -modes replaces ‘means’ with ‘modes’ to describe the similarity of categorical values. Compared with ROCK, CACTUS, algorithm k -modes shows great advantage on efficiency. FK-modes^[10], as an improved version of k -modes, give more sounding similarity definition. Algorithms k -modes and FK-modes adopt analogous heuristics to solve this problem, of which cost functions are different from this one here. Firstly, k initial cluster centers are chosen, and then the algorithms begin the process of iteration. During each iteration, every data object from the data sets is partitioned into one of these clusters according to a distance function. After that according to these data objects in a cluster, a new cluster center can be generated using different methods of k -means and k -modes. The iterations go on, until a termination condition is reached. Obviously, this search process belongs to local search strategies. Hence the clustering result of k -modes and FK-modes can easily be plunged into a local optimal solution.

Both of the k -modes and FK-modes are heuristics essentially using local search strategies and could thus easily plunge into local optimal solution. In order to address this problem, an immune-based cluster algorithm for data sets with categorical values, short for ICCV, is proposed in this paper. The algorithm ICCV is also an adaptation of k -modes. Distinguished from its forebears, it uses a global search strategy with immune model to optimize the cost function that is also an improved version of k -modes to increase the similarity between data from the same cluster.

The rest of the paper is organized as follows. Some mathematical definitions and notations are given in the next section. In section 3, algorithm ICCV is introduced and the overall procedure is presented in detail. The result of experiments for both artificial data set and real world data sets are given and analyzed in section 4. Conclusion and future work are presented in section 5 finally.

2 Notations and problem formulation

Before introduction of ICCV, some important definitions and notations are given in this section.

Definition 1. Cluster is a group of data objects satisfying^[8]:

- (1) Among the same cluster, data object are similar to each other.
- (2) Data objects form different clusters are different from each other too.

Definition 2. Categorical Domain and Attributes^[13]:

Let A_1, A_2, \dots, A_m be m attributes describing a space, and $DOM(A_1), DOM(A_2), \dots, DOM(A_m)$ the domains of the attributes. A domain $DOM(A_j)$ is defined as categorical if it is finite and unordered, e.g., for any a, b of $DOM(A_j)$, either $a = b$ or $a \neq b$. A_j is called a categorical attribute. Ω is a categorical space if all A_1, A_2, \dots, A_m are categorical.

Let $X = \{x_1, x_2, \dots, x_n\}$ denote a set of n objects and $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,m}]^T$ be an object represented by m values of attributes. Let k be a positive integer. The objective of clustering X is to find a partition that divides objects in X into k disjoint clusters.

Definition 3. Distance of Categorical values:

Let x_1, x_2 denote two data objects and x_2 belongs to one of the clusters C_q . The distance denoted by $d(x_1, x_2)$ from x_1 to x_2 is defined as below:

$$d(x_1, x_2) = \sum_{i=1}^m \phi(x_{1,i}, x_{2,i}), \quad (1)$$

where

$$\phi(x_{1,i}, x_{2,i}) = \begin{cases} 1 - f_r(x_{j,i} = x_{1,i} | C_q), & (x_{1,i} = x_{2,i}) \\ 1, & (x_{1,i} \neq x_{2,i}) \end{cases}, \quad (2)$$

and $f_r(x_{j,i} = x_{1,i} | C_q)$ represents the frequency of the data objects, of which i th attribute equals to $x_{1,i}$ in C_q .

It is a common sense that the further two data objects are, the less similar they seem. Therefore, the similarity between two data objects can be defined as follows:

Definition 4. Similarity of Categorical values

$$s(x_1, x_2) = \frac{1}{d(x_1, x_2)}. \quad (3)$$

For a given number of objects n , the number of possible partitions of the data set is definite but may be innumerable. It is impractical to investigate every partition in order to find a better one for a classification problem. A common solution is to choose a clustering criterion to guide the search for a partition. Cost function is one of commonly used clustering criteria to measure the quality of a clustering of a data set.

Definition 5. Cost function is defined to be a kind of criteria that measures quality of a clustering and it represents the sum of distances from each data object to a cluster center.

According to its definition, the cost function can be formulated as follow:

$$C(W, Q) = \sum_{l=1}^k \sum_{i=1}^n \sum_{j=1}^m w_{i,l} \phi(x_{i,j}, q_{l,j}). \quad (4)$$

With the formulation (4), the clustering problem is equivalently transformed into an optimization problem (COP) as follows:

$$\min C(W, Q) = \sum_{l=1}^k \sum_{i=1}^n \sum_{j=1}^m w_{i,l} \phi(x_{i,j}, q_{l,j}) \quad (5)$$

$$s.t. \quad \sum_{l=1}^k w_{i,l} = 1, \quad 1 \leq i \leq n \quad (6)$$

$$w_{i,l} \in \{0, 1\}, \quad 1 \leq i \leq n, \quad 1 \leq l \leq k. \quad (7)$$

In the COP model, W is an order partition matrix of $n \times k$, of which the element indicates whether data object x_i belongs to cluster C_l , with cluster center being q_l . The symbol Q is the set of cluster centers, denoted by $Q = \{q_1, q_2, \dots, q_k\}$. The constraints (6) and (7) imply that each data object belongs to one and only one cluster. Clustering results can be achieved by minimizing the objective function (5) that represents the sum of distance from each data object to a cluster center.

The decision variables are W and Q in this optimization model, which is a mixed integer program model. Because of the huge size of W , traditional mathematical program methods are not practical in this problem. It's interesting to observe that W and Q is dependent with each other. When a Q matrix is determined, the W matrix can be deduced by calculating the distance between each data object and each cluster centers in Q matrix.

It's easy to extend this cost function to a fuzzy version by letting $w_{i,l} \in [0, 1]$, which means a data object can partly belongs to more than one cluster.

3 Immune-based clustering algorithm for data sets with categorical values

As explained above, k -means and k -modes are heuristics in essential. If the initial cluster centers are not chosen properly, these algorithms will easily get stuck in the local optimal solution. It is this defect that restricts their use in more applications in practice. In order to solve this problem, researchers recently try to use other search strategies, e.g. evolutionary-based algorithms. Genetic algorithm (GA), a kind of algorithm that imitates the process of evolution of life-form, is a global search strategy and fits for implementing in parallel. In recent years, some researchers have successfully exploited some GAs to optimize the cost function derived by k -means and k -modes [4,11,14,17]. Still, GA shows some problems, as degradation which increase the generation of GA to an unbearable level, the cluster centers obtained by GA are not points from the data sets, etc. In order to obtain a global optimal solution of cost function (5), a new search strategy is used here-Immune System Model.

3.1 Immune system

For years, researchers from different areas try to get inspiration from the activities of life-form. The immune system, which is made up of special cells, proteins, tissues, and organs, defends people against germs and microorganisms^[3]. Through a series of steps called the immune response, the immune system attacks organisms and substances that invade our systems and cause diseases.

The immune system usually produces a group of B-cells to secrete antibodies. These antibodies can recognize and bind antigens and finally kill them. The affinity between an antigen and an antibody describes

the strength of the antigen antibody interaction, also referred as the affinity between the antigen and the B-cell. The larger the affinity between an antibody and an antigen, the tighter the antibody can bind the antigen. The body employs a group of immune mechanisms that can facilitate the B-cell generation to bind the antigens. The immune system randomly generates many B-cells. The B-cells with higher affinity to antigens are cloned. These cloned cells can easily recognize and bind antigens, and are thus called memory cells. This cloning process of generating memory cells is called clone selection. Memory cells have a longer life than normal B-cells and are thus useful when a similar infection occurs at a future time. The B-cells that have low affinity to antigens are either directly eliminated or mutated.

So far, many researchers have already exploited the immune knowledge to solve a lot of data mining problems^[3,15], and also several new models and methods have been proposed and applied to some practical projects^[2]. Among these models, a model named AiNet, proposed by De Castro and Fernando J have been proved efficient in many applications^[2]. A reduced version of AiNet model is suggested in this paper.

The immune network theory indicates that the immune system involves not only the interaction of antibodies and antigens but also the interaction of antibodies with other antibodies. Here, we only consider the interaction between antibodies and antigens. Antibodies can be improved through this interaction, which can be formulated as:

$$C = C - \alpha(C - X), \quad (8)$$

where C , X stand for the antibody and antigen respectively; and α , called evolution rate, represents the level that an antibody learns from the antigen.

In the ICCV algorithm, each data point is treated as an antigen. The algorithm evolves a population of antibodies based on formula (8). These antibodies form a network, which can represent the antigens in a compressed way.

3.2 Encoding

It can be observed from the cost function (5) that two encoding schemes, encoding the W matrix or encoding Q matrix can be selected alternatively. Because our clustering algorithm deals with large data sets, the W matrix must be of large size, which will lead to huge search space. Hereby, the second encoding scheme is adopted.

3.3 Initial antibodies

The easiest way to get initial antibodies is random generation. Of course, the initial antibodies can be also generated according to the background knowledge of the problem, or by calculating the centers of clusters that are randomly partitioned. In ICCV, the latter method is adopted.

3.4 Evolution rate

As indicated in formula (8), evolution rate α is an essential parameter in the immune system, as well in the ICCV algorithm. The ability that the whole immune system improves the affinity between the antigens and the antibodies is controlled by α . The actual value of α must be selected according to the problems encountered. A larger α can help the algorithm jump out of the local optimal solution, while reduces the efficiency of ICCV. Conversely, a smaller value of α may make the algorithm plunged into the local optimal solution.

3.5 Overall procedure of ICCV

The overall procedure of ICCV is presented in detail as follows:

Step 1. Input the data set X of size m . Regard each data object as an antigen.

Step 2. Generate the K initial antibodies.

Step 3. Identification of antigens. According to the affinity between antibodies and antigens, group each antigen to an antibody. The affinity can be calculated from formula (1). In the same time, update the appearance frequency of each attribute value in its corresponding cluster.

Step 4. Evolve antibodies. For each cluster:

Step 4.1. For every antigen in this cluster, formula (5) is applied to get a new antibody.

Step 4.2. For every newly-born antibody, the sum of affinity between it and every antigen is calculated.

Step 4.3. Choose the antibody with the largest affinity sum as the best antibody.

Step 4.4. Keep the best antibodies in each cluster; eliminate every other antibody.

Step 5. Check the stop condition. If the difference of cost function is smaller than a value preassigned within certain iterations, it can be thought that the cost function cannot be improved anymore. Contrarily, go to Step 3.

4 Experiments and simulation analysis

The experiments are mainly focused on validation and accuracy of the algorithm ICCV. Two kinds of data sets are selected for this purpose in this section. One is artificial data set and the other is benchmark data sets included in UCI Repository of Machine Learning Databases^[18].

In each test, an attribute of data set is considered to be “major” attribute, which are used to classify the data set. Of course, the length of “major” attribute’s domain is the input parameter K . The accuracy of the algorithm ICCV is obtained by comparing the cluster label of each data object with the label of its “major” attribute. A value named miscluster rate is calculated to quantify the accuracy of the ICCV. It is formulated as follows:

$$miscluster\ rate = \frac{N_m}{N}, \quad (9)$$

where, N_m denotes the number of data objects whose cluster labels is different from their labels of “major” attribute, and N denotes the size of the data set.

4.1 Experiments on artificial data set

The experimental data sets are randomly generated. The data sets have 10 attributes, each of which has 5 different values. The other attributes can be distributed stochastically in their ranges according to the corresponding cluster label attribute. Ten tests with sizes of 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500 and 5000 of data set are conducted.

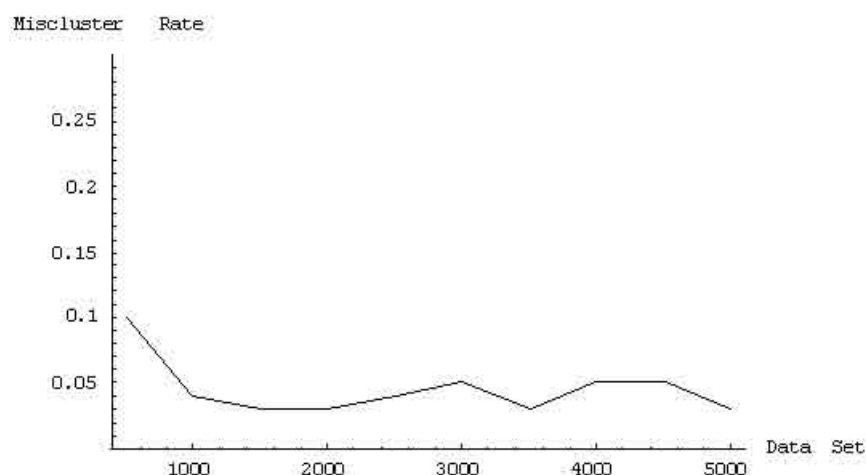


Fig. 1. Miscluster rate of artificial data sets variation with data set size

The simulation results of the algorithm ICCV on different sizes of artificial data sets are shown in Fig. 1, of which the horizontal and vertical axis represents the size of data set and the miscluster rate respectively.

It can be observed from Fig. 1 that, the miscluster rates of the algorithm for the tests with sizes of 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500 and 5000 are 11%, 4.8%, 3.5%, 3.6%, 4.4%, 5.3%, 3.5%, 5.7%, 5.8% and 3.3% respectively. In average, the miscluster rate of the algorithm is around 6% no matter how many data points to be clustered. As the size of data set increases, the fluctuation of miscluster rate decreases. This is to say, ICCV is computational scalable for large-scale data sets. So the algorithm ICCV can be applied for large data sets. The reason why there are errors about 6% of total clustering result is because the data sets are generated randomly, so maybe a data with label “1” have other attributes more similar to data with label “2”.

4.2 Experiments on real-life data sets

Two benchmark data sets are selected from the UCI Machine Learning Repository^[18] for tests, one is mushroom data set and another is congressional votes data set. Tests are given as follows:

Mushroom Data Set: It has 22 attributes and 8124 records. Each record represents physical characteristics of a single mushroom. A classification label of poisonous or edible is provided with each record. The numbers of edible and poisonous mushrooms in the dataset are 4208 and 3916, respectively. Also, ten tests of data size of 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500 and 5000 are conducted.

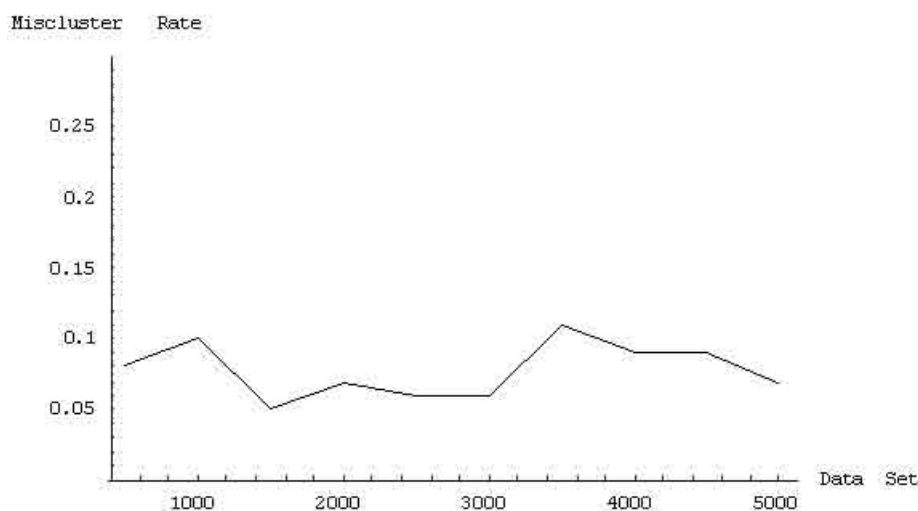


Fig. 2. Miscluster rate of mushroom data set of different scale

The simulation result of the algorithm ICCV on mushroom data set is shown in Fig. 2. It's easy to observe that the average miscluster rate of these tests is around 8.5% and thus one can conclude that ICCV work quite well for mushroom data set. So we can see ICCV can be used in real world application.

Congressional Votes Data Set: It is the United States Congressional Voting Records in 1984. Each record corresponds to one Congress-man's votes on 16 issues (e.g., education spending, crime). All attributes are Boolean with Yes and No values, and very few contain missing values. A classification label of Republican or Democrat is provided with each data records. The data set contains records for 168 Republicans and 267 Democrats. At last, according to the results calculated by ICCV, 31% of the congress-men are Republicans, while the actually number is 39%, as shown in Table 1:

5 Conclusions

In this paper, we present an algorithm-ICCV for clustering categorical data. Distinguished from past algorithms such as k-means, k-modes, ICCV adopted an immune system model to prevent the cluster solution to plunge into local optimal solutions. The satisfactory results have demonstrated the effectiveness of the

Table 1. Simulation result for congressional votes data set

Congressman \ Proportion	actual value	cluster result
Republicans	39%	31%
Democrats	61%	69%

algorithms for large complex data sets in terms of both the number of records and the number of attributes. As ICCV solves the problem of clustering data sets with categorical values, it's possible to remedy the definition of similarity, and extend this work to the job of clustering data sets with mixed numerical and categorical values.

References

- [1] M. R. Anderberg. *Cluster Analysis for Applications*. Academic Press, 1973.
- [2] D. Castro, J. Fernando. An evolutionary immune network for data clustering. **in:** *Proc. Of the IEEE SBRN*, 2000, 84–89
- [3] D. Dasgupta. *Artificial Immune Systems and Their Applications*. Springer Verlag Inc., 1999.
- [4] D. E. Goldberg. *Genetic Algorithms in Search Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [5] S. Guha, R. Rastogi, K. Shim. ROCK: A robust clustering algorithm for categorical attributes. **in:** *Proc.1999 Int. Conf. Data Engineering*, 1999, 512–521.
- [6] V. Ganti, J. Gehrke, R. Ramakrishnan. CACTUS-Clustering categorical data using summaries. **in:** *Proc.1999 Int. Conf. Knowledge Discovery and Data Mining*, 1999, 73–83.
- [7] D. Gibson, K. Jon, R. Prabhakar. Clustering categorical data: An approach based on dynamical systems, **in:** *Proc. of the 24th International Conference on Very Large Databases*, 1998, 311–323.
- [8] J. Han, M. Kamber. *Data mining : concepts and techniques*, Morgan Kaufmann Publishers, San Francisco, 2001.
- [9] Z. He, X. Xu, S. Deng. Squeezer: An efficient algorithm for clustering categorical data, *Journal of Computer Science and Technology*, 2002, (5): 611–624.
- [10] Z. He, X. Xu, S. Deng. FK-modes: Improving k-modes algorithm considering frequencies of attribute values in mode, *working paper*.
- [11] Z. Huang. Clustering large data sets with mixed numeric and categorical values. **in:** *Proceedings of the First Pacific-Asia Conference on Knowledge Discovery and Data Mining*, World Scientific, Singapore.
- [12] Z. Huang, A fast clustering algorithm to cluster very large categorical data sets in data mining. **in:** *Proc. SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery. Tech. Report 97-07*, UBC, Dept. of CS., 1997.
- [13] Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 1998, (2): 283–304.
- [14] J. Li, X. Gao, L. Jiao. A GA-based clustering algorithm for large data sets with mixed numeric and categorical values, **in:** *IEEE Proceedings for the Fifth International Conference on Computational Intelligence and Multimedia Applications (ICCI'03)*, 2003.
- [15] T. Liu, Y. Wang, Z. Wang. A new clustering method based on artificial immune system. *Computer Engineering and Application*, 2004, 182–184.
- [16] J.B. MacQueen. Some methods for classification and analysis of multivariate observations. **in:** *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, 281–297.
- [17] U. Maulik, S. Bandyopadhyay. Genetic algorithm-based clustering technique. *Pattern Recognition Society*. 2000, 1455–1465.
- [18] C. J. Merz, P. Merphy. *UCI Repository of Machine Learning Databases*, 1996.