Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 36 (2009) 1615–1620
Expert Systems with Applications
www.elsevier.com/locate/eswa

A genetic fuzzy *k*-Modes algorithm for clustering categorical data

G. Gan, J. Wu, Z. Yang *

Department of Mathematics and Statistics, York University, Toronto, Ontario, Canada M3J 1P3

Abstract

The fuzzy *k*-Modes algorithm introduced by Huang and Ng [Huang, Z., & Ng, M. (1999). A fuzzy *k*-modes algorithm for clustering categorical data. *IEEE Transactions on Fuzzy Systems, 7*(4), 446–452] is very effective for identifying cluster structures from categorical data sets. However, the algorithm may stop at locally optimal solutions. In order to search for appropriate fuzzy membership matrices which can minimize the fuzzy objective function, we present a hybrid genetic fuzzy *k*-Modes algorithm in this paper. To circumvent the expensive crossover operator in genetic algorithms (GAs), we hybridize GA with the fuzzy *k*-Modes algorithm and define the crossover operator as a one-step fuzzy *k*-Modes algorithm. Experiments on two real data sets are carried out to illustrate the performance of the proposed algorithm.
© 2007 Elsevier Ltd. All rights reserved.

Keywords: Genetic algorithm; *k*-Modes; Fuzzy logic; Categorical data

1. Introduction

As a primary tool of data mining, cluster analysis (Cormack, 1971; Gordon, 1987; Jain, Murty, & Flynn, 1999; Murtagh, 1983), also called segmentation analysis or taxonomy analysis, is a way to create groups of objects, or clusters, in such a way that objects in one cluster are very similar and objects in different clusters are quite distinct. In general, a well-designed clustering algorithm usually involves the following four design phases: data representation, modeling, optimization and validation (Buhmann, 2003). The data representation phase predetermines what kind of cluster structures can be identified in the data. On the basis of data representation, the modeling phase defines the notion of clusters and the criteria that separates the desired group structures from unfavorable ones. In the modeling phase, a quality measure which can be either optimized or approximated during the search for hidden structures in the data is produced. Since the clustering process is an unsupervised process, the validation phase is necessary to validate the results produced by the clustering algorithm.

In general, clustering algorithms are classified into two categories (Everitt, Landau, & Leese, 2001; Jain & Dubes, 1988): hard clustering algorithms and fuzzy clustering algorithms. In the framework of hard clustering, each object belongs to one and only one cluster. On the contrary, in the framework of fuzzy clustering each object is allowed to have membership functions to all clusters rather than having a distinct membership to exactly one cluster. Mathematically, a fuzzy clustering problem can be represented as an optimization problem (Dunn, 1974):

$$\min_{W,Z} F(W,Z) = \sum_{l=1}^{k}\sum_{i=1}^{n} w_{li}^{\alpha} d(\mathbf{z}_l, \mathbf{x}_i)$$

such that

$$0 \le w_{li} \le 1, \quad 1 \le l \le k, \quad 1 \le i \le n, \tag{1a}$$

$$\sum_{l=1}^{k} w_{li} = 1, \quad 1 \le i \le n, \tag{1b}$$

$$0 < \sum_{i=1}^{n} w_{li} < n, \quad 1 \le l \le k, \tag{1c}$$

* Corresponding author. Tel.: +1 416 7362100x66098; fax: +1 416 7365287.
E-mail addresses: gjgan@mathstat.yorku.ca (G. Gan), wujh@mathstat.yorku.ca (J. Wu), zyang@mathstat.yorku.ca (Z. Yang).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.11.045

where n is the number of objects in the data set under consideration, k is the number of clusters, $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ is a set of n objects each of which is described by d attributes, $Z = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k\}$ is a set of k cluster centers, $W = (w_{li})$ is a $k \times n$ fuzzy membership matrix, $\alpha \in [1, \infty]$ is a weighting exponent, and $d(\mathbf{z}_l, \mathbf{x}_i)$ is a certain distance measure between cluster center $\mathbf{z}_l$ and the object $\mathbf{x}_i$.

A well-known fuzzy clustering algorithm is the fuzzy k-Means algorithm due to Bezdek (1974) and Ruspini (1969). The fuzzy k-Means algorithm starts with an initial value of W and then repeatedly iterates between estimating cluster centers Z given W and estimating the membership matrix W given Z until two successive values of W or Z are equal. Since the fuzzy k-Means algorithm works only on numeric values, a fuzzy k-Modes algorithm (Huang & Ng, 1999) has been developed for the purpose of clustering categorical data sets. A known problem associated with both the fuzzy k-Means algorithm and the fuzzy k-Modes algorithm is that they may only stop at local optima of the optimization problem, since the function $F(W, Z)$ is non-convex in general (Ng & Wong, 2002).

To find a global solution of the optimization problem, genetic algorithms (GAs) (Davis, 1991) and tabu search (TS) based techniques (Glover & Laguna, 1997) are applied. The genetic k-Means algorithm (Krishna & Narasimha, 1999), for example, integrates the k-Means algorithm and the genetic algorithm so as to find the globally optimal solution. In order to find the globally optimal solution for the fuzzy k-Modes algorithm, Ng and Wong introduced tabu search based fuzzy k-Modes algorithm (Ng & Wong, 2002).

The main aim of this paper is to develop a genetic fuzzy k-Modes algorithm, which integrates the genetic algorithm and the fuzzy k-Modes algorithm in order to find the globally optimal solution of the optimization problem. The outline of the paper is as follows. In Section 2, the fuzzy k-Modes algorithm is briefly reviewed. In Section 3, the new genetic fuzzy k-Modes algorithm is proposed. In Section 4, the experimental results are presented to illustrate the effectiveness of our new algorithm. Finally, some concluding remarks are given in Section 5.

2. Fuzzy k-Modes

To describe the fuzzy k-Modes algorithm (Huang & Ng, 1999), let us begin with some notations. Let $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a categorical data set with n objects each of which is described by d categorical attributes $A_1, A_2, \ldots, A_d$. Attribute $A_j (1 \le j \le d)$ has n_j categories, i.e., $\mathrm{DOM}(A_j) = \{a_{j1}, a_{j2}, \ldots, a_{jn_j}\}$. Let the cluster centers be represented by $\mathbf{z}_l = (z_{l1}, z_{l2}, \ldots, z_{ld})$ for $1 \le l \le k$, where k is the number of clusters. The simple matching distance measure between $\mathbf{x}$ and $\mathbf{y}$ in D is defined as

$$d_c(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{d} \delta(x_j, y_j), \tag{2}$$

where x_j and y_j are the jth components of $\mathbf{x}$ and $\mathbf{y}$, respectively, and

$$\delta(x_j, y_j) = \begin{cases} 0 & \text{if } x_j = y_j; \\ 1 & \text{if otherwise.} \end{cases}$$

Then the objective of the fuzzy k-Modes clustering is to find W and Z that minimize

$$F_c(W, Z) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{li}^{\alpha} d_c(\mathbf{x}_i, \mathbf{z}_l), \tag{3}$$

subject to 1a, 1b and 1c, where $\alpha > 1$ is the weighting component, $d_c(\cdot, \cdot)$ is defined in Eq. (2), $W = (w_{li})$ is the $k \times n$ fuzzy membership matrix, and $Z = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k\}$ is the set of cluster centers. Note that $\alpha = 1$ gives the hard k-Modes clustering, i.e., the k-Modes algorithm.

To update the cluster centers given the estimate of W, Huang and Ng (1999) proved the following theorem.

Theorem 1. *The quantity $F_c(W, Z)$ defined in Eq. 3 is minimized if and only if $z_{lj} = a_{jr} \in \mathrm{DOM}(A_j)$ where*

$$r = \arg \max_{1 \le t \le n_j} \sum_{i, x_{ij} = a_{jt}} w_{li}^{\alpha},$$

i.e.,

$$\sum_{i, x_{ij} = a_{jr}} w_{li}^{\alpha} \ge \sum_{i, x_{ij} = a_{jt}} w_{li}^{\alpha}, \quad 1 \le t \le n_j$$

for $1 \le j \le d$ and $1 \le l \le k$.

To update the fuzzy membership matrix W given the estimate of Z, Huang and Ng (1999) also presented the following theorem.

Theorem 2. *Let $Z = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k\}$ be fixed, then the fuzzy membership matrix W which minimizes the quantity $F_c(W, Z)$ defined in Eq. (3) subject to 1a, 1b and 1c is given by*

$$w_{li} = \begin{cases} 1 & \text{if } \mathbf{x}_i = \mathbf{z}_l; \\ 0 & \text{if } \mathbf{x}_i = \mathbf{z}_h,\ h \ne l; \\ \dfrac{1}{\sum_{h=1}^{k} \left[\frac{d(\mathbf{x}_i, \mathbf{z}_l)}{d(\mathbf{x}_i, \mathbf{z}_h)} \right]^{\frac{1}{\alpha - 1}}} & \text{if otherwise,} \end{cases} \quad 1 \le l \le k,\ 1 \le i \le n.$$

Based on the two theorems described above, the fuzzy k-Modes algorithm can be implemented recursively (see Algorithm 1).

Algorithm 1. Fuzzy k-Modes algorithm, r is the maximum number of iterations.

1: Choose initial point $Z_0 \in \mathbb{R}^{mk}$;
2: Determine W_0 such that the cost function $F(W_0, Z_0)$ is minimized;
3: **for** $t = 1$ to r **do**
4: Determine Z_1 such that the cost function $F(W_0, Z_1)$ is minimized;
5: **if** $F(W_0, Z_1) = F(W_0, Z_0)$ **then**
6: stop;

```
7:     else
8:       Determine W_1 such that the cost function
       F(W_1, Z_1) is minimized;
9:       if F(W_1, Z_1) = F(W_0, Z_1) then
10:        stop;
11:      else
12:        W_0 ⇐ W_1;
13:      end if
14:    end if
15:  end for
```

3. Genetic fuzzy k-Modes

In general, a GA consists of five basic elements: *coding* or *string representation*, *population initialization*, *selection*, *crossover* and *mutation*. In order to speed up the convergence process, we use a one-step fuzzy k-Modes algorithm in the place of the crossover process. In this section, we will introduce these five elements of the GA for fuzzy k-Modes clustering.

3.1. String representation

A natural coding approach is to represent the $n \times k$ fuzzy membership matrix W in a chromosome, where n is the number of objects in the data set and k is the number of clusters. That is, the length of a chromosome is $n \times k$, where the first k positions (or, genes) represent the k fuzzy membership of the first data point, the next k positions represent those of the second data point, and so on. For example, if $n = 4$ and $k = 3$, then the chromosome, $(a_1, a_2, \ldots, a_{12})$, represents the following 3×4 fuzzy membership matrix:

$$W = \begin{pmatrix} a_1 & a_4 & a_7 & a_{10} \\ a_2 & a_5 & a_8 & a_{11} \\ a_3 & a_6 & a_9 & a_{12} \end{pmatrix},$$

where W satisfies 1a, 1b and 1c. We call a chromosome representing a fuzzy membership matrix which satisfies 1a, 1b and 1c a legal chromosome.

3.2. Initialization process

In the initialization phase, a population of N legal chromosomes is generated, where N is the size of the population. To generate a chromosome $(a_1, a_2, \ldots, a_{n\cdot k})$, we employ the method introduced by Zhao, Tsujimura, and Gen (1996), which is described as follows:

```
for i = 1 to n do
  Generate k random numbers v_{i1}, v_{i2}, ..., v_{ik} from [0, 1]
  for the ith point of the chromosome;
  Calculate a_{(j-1)*n+i} = v_{ij} / Σ_{l=1}^{k} v_{il} for j = 1, 2, ..., k;
end for
```

If the produced chromosome satisfies 1a, 1b and 1c, then generate next chromosome, otherwise repeat the above process.

The process described above is repeated N times to generate an initial population.

3.3. Selection process

To describe the selection process, let us first introduce how to calculate the fitness of a chromosome. In our algorithm, we use the well-known rank-based evaluation function, i.e.,

$$F(s_i) = \beta(1-\beta)^{r_i - 1}, \tag{4}$$

where $s_i (1 \le i \le N)$ is the ith chromosome in the population, r_i is the rank of s_i, and $\beta \in [0, 1]$ is a parameter indicating the selective pressure of the algorithm. Note that the chromosome with rank 1 is the best one and the chromosome with rank N is the worst one.

In our algorithm, the selection process is based on spinning the roulette wheel (Zhao et al., 1996) N times and each time a chromosome is selected for the next population. Let $P_j (0 \le j \le N)$ be the cumulative probabilities defined as

$$P_j = \begin{cases} 0 & \text{for } j = 0; \\ \dfrac{\sum_{i=1}^{j} F(s_i)}{\sum_{i=1}^{N} F(s_i)} & \text{for } j = 1, 2, \ldots, N. \end{cases}$$

Then the new population is generated as follows:

```
for i = 1 to N do
  Generate a random real number v from [0, 1];
  if P_{j-1} < v < P_j then
    Select s_j;
  end if
end for
```

3.4. Crossover process

After the selection process, the population will go through a crossover process. Similar to genetic k-Means algorithm (Krishna & Narasimha, 1999), in our algorithm we employ a one-step fuzzy k-Modes algorithm as the crossover operator. Based on Theorems 1 and 2, we can update each chromosome in the population as follows:

```
for t = 1 to N do
  Let W_t be the fuzzy membership matrix represented
  by s_t;
  Obtain the new set of cluster centers Ẑ_t given W_t
  according to Theorem 1;
  Obtain the fuzzy membership matrix Ŵ_t given Ẑ_t
  according to Theorem 2;
  Replace s_t with the chromosome representing Ŵ_t.
end for
```

3.5. Mutation process

In the mutation process, each gene has a small probability P_m (say 0.01) of mutating, decided by generating a random number (the gene will mutate if the random number is less than 0.01, otherwise not). In our algorithm, a change of one gene of a chromosome will trigger a series of changes of genes in order to satisfy 1b. Thus in the mutation process, the fuzzy memberships of a point in a chromosome will be selected to mutate together with probability P_m. The mutation process is described as follows:

for $t = 1$ to N **do**
  Let $(a_1, a_2, \ldots, a_{n\cdot k})$ denote the chromosome s_t;
  for $i = 1$ to n **do**
    Generate a random real number $v \in [0, 1]$;
    if $v \leq P_m$ **then**
      Generate k random numbers $v_{i1}, v_{i2}, \ldots, v_{ik}$ from $[0, 1]$ for the ith point of the chromosome;
      Replace $a_{(j-1)*n+i}$ with $v_{ij}/\sum_{l=1}^{k} v_{il}$ for $j = 1, 2, \ldots, k$;
    end if
  end for
end for

3.6. Termination criterion

In our algorithm, the processes of selection, one-step fuzzy k-Modes, mutation are executed for, $G_{\max}$, a maximum number of iterations (or, generations). The best chromosome up to the last generation provides the solution to the clustering problem. We also have implemented the elitist strategy (Cowgill, Harvey, & Watson, 1999) at each generation by creating $N - 1$ children and retaining the best parent of the current generation for the next generation.

4. Experiments

The genetic fuzzy k-Modes algorithm is coded in C++ programming language. Two data sets from UCI machine learning repository (Blake & Merz, 1998) are used to test the feasibility and effectiveness of our new algorithm. Our experiments were conducted on a PC with 2.2 Hz CPU and 512 M RAM and a Sun Blade 1000 workstation.

4.1. Clustering quality measures

The clustering result of the genetic fuzzy k-Modes algorithm is a fuzzy membership matrix from which we obtain the cluster memberships as follows. The object $\mathbf{x}_i$ is assigned to the rth cluster if

$$r = \arg\max_{1\leq l\leq k} w_{li}, \quad \text{or} \quad w_{ri} = \max_{1\leq l\leq k} w_{li}.$$

In the case that the maximum is not unique, the object $\mathbf{x}_i$ is assigned to the cluster first archiving the maximum.

We used the corrected Rand index (Hubert & Arabie, 1985) to assess the recovery of the underlying cluster structure. Let $\mathscr{P} = \{C_1, C_2, \ldots, C_{k_1}\}$ and $\mathscr{P}' = \{C'_1, C'_2, \ldots, C'_{k_2}\}$ be two clusterings of D. Denote by n_{ij} the number of points simultaneously in C_i and C'_j, i.e. $n_{ij} = |\, C_i \cap C'_j \,|$, then the corrected Rand index is defined as

$$\gamma = \frac{\binom{n}{2}\sum_{i=1}^{k_1}\sum_{j=1}^{k_2}\binom{n_{ij}}{2} - \sum_{i=1}^{k_1}\binom{|C_i|}{2}\sum_{j=1}^{k_2}\binom{|C'_j|}{2}}{\frac{1}{2}\binom{n}{2}\left[\sum_{i=1}^{k_1}\binom{|C_i|}{2} + \sum_{j=1}^{k_2}\binom{|C'_j|}{2}\right] - \sum_{i=1}^{k_1}\binom{|C_i|}{2}\sum_{j=1}^{k_2}\binom{|C'_j|}{2}}.$$

The corrected Rand index γ ranges from 0 when the two clusterings have no similarities (i.e. when one consists of a single cluster containing the whole data set and the other only clusters containing single points), to 1 when the two clusterings are identical. Since we know the true clustering of the data set, the true clustering and the resulting clustering are used to calculate γ.

4.2. Data sets

Our first data set is the well-known soybean data set. The soybean data set has 47 records each of which is described by 35 attributes. Each record is labeled as one of the four diseases: diaporthe stem rot, charcoal rot, rhizoctonia root rot and phytophthora rot. Except for the phytophthora rot which has 17 instances, all other diseases have 10 instances each. Since there are 14 attributes that have only one category, we only selected other 21 attributes for the purpose of clustering.

Our second data set is the Congressional voting data set. The Congressional voting data set includes votes for each of the US House of Representatives Congressmen on the 16 key votes identified by the CQA. It has 435 objects (267 democrats, 168 republicans) each of which is described by 16 binary attributes. Some of the objects have missing values, we denote the missing value by "?" and treat it as an additional category for that attribute.

4.3. Results

For the soybean data set, we use the parameters with following values: $k = 4$, $N = 20$, $G_{\max} = 15$, $\alpha = 1.2$, $\beta = 0.1$ and $P_m = 0.01$. The results of the 100 runs are summarized in Table 1. The first column is the best case of the 100 runs in terms of the objective function defined in Eq. 3. The corresponding corrected Rand index γ of the best case is given in the third column of the table. The second column and the fourth column give the average objective function value $F_c(W, Z)$ and the average corrected Rand index γ of the 100 runs, respectively. The fifth column is the number of correct clusterings of the 100 runs.

The best case has a corrected Rand index of 1, which means that all objects are correctly clustered into the four

Table 1
The summary of 100 runs of the genetic fuzzy k-Modes algorithm on the soybean data set

Best	Average	γ for the best	Average γ	# of runs $\gamma = 1$
193.832904	209.081562	1.000000	0.771319	24

given clusters. The fuzzy membership of the best case is given in Table 2. One can see from Table 2 that most of the objects have a heavy membership for exact one cluster and light memberships for other clusters. The object $\mathbf{x}_{10}$ has a membership of 1 for one cluster and 0 for others, which implies that $\mathbf{x}_{10}$ is the center of the cluster it belongs to. The objects $\mathbf{x}_{41}$ and $\mathbf{x}_{47}$ have two equal memberships for two different clusters, so they can be assigned to either of the two clusters.

We also tested the genetic k-Modes algorithm on the soybean data set for different configurations of the parameters α, β and P_m. It seems that the parameter configuration, $\alpha = 1.2$, $\beta = 0.1$ and $P_m = 0.01$, gives better clustering results than those produced by other configurations.

We specify $k = 2$, $N = 20$, $G_{\max} = 15$, $\alpha = 1.2$, $\beta = 0.1$ and $P_m = 0.01$ and run the algorithm 100 times on the Congressional voting data sets. Table 3 summarizes the clustering results of the 100 runs. The meaning of each column in Table 3 is the same as that in Table 1. We see from Table 3

Table 2
The fuzzy membership of the best case of the 100 runs of the genetic fuzzy k-Modes algorithm on the soybean data set

Object	w_{1i}	w_{2i}	w_{3i}	w_{4i}	First choice	Second choice
$\mathbf{x}_1$	0.99949	0.00013	0.00032	0.00006	1	3
$\mathbf{x}_2$	0.87185	0.01826	0.06780	0.04210	1	3
$\mathbf{x}_3$	0.92638	0.00949	0.01940	0.04473	1	4
$\mathbf{x}_4$	0.99596	0.00065	0.00242	0.00097	1	3
$\mathbf{x}_5$	0.99376	0.00065	0.00409	0.00150	1	3
$\mathbf{x}_6$	0.96755	0.00814	0.01215	0.01215	1	4
$\mathbf{x}_7$	0.92278	0.01334	0.04455	0.01933	1	3
$\mathbf{x}_8$	0.98832	0.00133	0.00407	0.00628	1	4
$\mathbf{x}_9$	0.99543	0.00065	0.00242	0.00150	1	3
$\mathbf{x}_{10}$	1.00000	0.00000	0.00000	0.00000	1	4
$\mathbf{x}_{11}$	0.00065	0.99888	0.00023	0.00023	2	1
$\mathbf{x}_{12}$	0.05951	0.88110	0.01950	0.03988	2	1
$\mathbf{x}_{13}$	0.00020	0.99961	0.00013	0.00006	2	1
$\mathbf{x}_{14}$	0.00408	0.99051	0.00134	0.00408	2	4
$\mathbf{x}_{15}$	0.00065	0.99880	0.00032	0.00023	2	1
$\mathbf{x}_{16}$	0.04194	0.92658	0.01097	0.02051	2	1
$\mathbf{x}_{17}$	0.01009	0.98532	0.00188	0.00272	2	1
$\mathbf{x}_{18}$	0.00065	0.99886	0.00032	0.00017	2	1
$\mathbf{x}_{19}$	0.01880	0.96905	0.00399	0.00816	2	1
$\mathbf{x}_{20}$	0.00065	0.99844	0.00045	0.00045	2	1
$\mathbf{x}_{21}$	0.00627	0.00071	0.98674	0.00627	3	4
$\mathbf{x}_{22}$	0.00150	0.00032	0.99409	0.00409	3	4
$\mathbf{x}_{23}$	0.00242	0.00023	0.99637	0.00097	3	1
$\mathbf{x}_{24}$	0.00032	0.00003	0.99933	0.00032	3	4
$\mathbf{x}_{25}$	0.00150	0.00023	0.99585	0.00242	3	4
$\mathbf{x}_{26}$	0.15242	0.03233	0.74914	0.06611	3	1
$\mathbf{x}_{27}$	0.00150	0.00023	0.99585	0.00242	3	4
$\mathbf{x}_{28}$	0.00150	0.00023	0.99585	0.00242	3	4
$\mathbf{x}_{29}$	0.04068	0.01989	0.89874	0.04068	3	4
$\mathbf{x}_{30}$	0.00241	0.00017	0.99333	0.00409	3	4
$\mathbf{x}_{31}$	0.00402	0.00132	0.01695	0.97770	4	3
$\mathbf{x}_{32}$	0.03696	0.01603	0.18163	0.76539	4	3
$\mathbf{x}_{33}$	0.00065	0.00032	0.00242	0.99661	4	3
$\mathbf{x}_{34}$	0.09868	0.01300	0.30116	0.58716	4	3
$\mathbf{x}_{35}$	0.00096	0.00044	0.01423	0.98436	4	3
$\mathbf{x}_{36}$	0.00009	0.00003	0.00020	0.99968	4	3
$\mathbf{x}_{37}$	0.00557	0.00360	0.11530	0.87553	4	3
$\mathbf{x}_{38}$	0.01128	0.00522	0.08563	0.89788	4	3
$\mathbf{x}_{39}$	0.00188	0.00096	0.01011	0.98705	4	3
$\mathbf{x}_{40}$	0.00559	0.00287	0.03005	0.96150	4	3
$\mathbf{x}_{41}$	0.07594	0.02045	0.45181	0.45181	4	3
$\mathbf{x}_{42}$	0.01801	0.00881	0.11324	0.85994	4	3
$\mathbf{x}_{43}$	0.00006	0.00004	0.00032	0.99958	4	3
$\mathbf{x}_{44}$	0.02868	0.00940	0.04431	0.91762	4	3
$\mathbf{x}_{45}$	0.01398	0.00965	0.30884	0.66753	4	3
$\mathbf{x}_{46}$	0.00976	0.00707	0.02979	0.95337	4	3
$\mathbf{x}_{47}$	0.01532	0.00413	0.49028	0.49028	4	3

Table 3
The summary of 100 runs of the genetic fuzzy k-Modes algorithm on the Congressional voting data set

Best	Average	γ for the best	Average γ	# of runs $\gamma = 1$
1659.401378	1663.201251	0.529821	0.506913	0

Table 4
The misclassification matrix of the best case of the 100 runs of the genetic fuzzy k-Modes algorithm on the Congressional voting data set

	Cluster 1	Cluster 2
Republican	153	15
Democrat	44	223

Table 5
The misclassification matrix of the best case of the 100 runs of the genetic fuzzy k-Modes algorithm on the Congressional voting data set with $k = 4$

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Republican	90	71	1	6
Democrat	18	24	86	139

Table 6
The misclassification matrix of the best case of the 100 runs of the genetic fuzzy k-Modes algorithm on the Congressional voting data set with $k = 6$

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Republican	4	70	2	84	6	2
Democrat	41	9	65	5	24	123

that the difference between the best case and the average case is very small relative to the average, which indicates the algorithm is stable. However, there is no such run of the 100 runs that all the objects are correctly clustered into the two given clusters.

Table 4 gives the misclassification matrix of the best case of the 100 runs. We see from this table that only 59 out of 435 objects are misclassified. We also run the algorithm on the Congressional voting data set 100 times with $k = 4$ and other parameters kept the same. The misclassification matrix of the best case of the 100 runs is given in Table 5. We see from Table 5 that only $18 + 24 + 1 + 6 = 49$ out of 435 objects are misclassified. Table 6 gives the misclassification matrix of 100 runs of the algorithm on the Congressional data set with $k = 6$ and other parameters kept the same. In this case, only $4 + 9 + 2 + 5 + 6 + 2 = 28$ out of 435 objects are misclassified.

From these experimental results of the algorithm on the Congressional data set, we observed the following interesting fact: when we increase k in the algorithm, the number of objects that are clustered incorrectly decreases. Consider the Congressional data set, for example, we specified $k = 2, 4, 6$ and observed that the numbers of misclassified objects $59, 49, 28$, respectively.

5. Conclusions

In this paper we presented the genetic fuzzy k-Modes algorithm for clustering categorical data sets. We treated the fuzzy k-Modes clustering as an optimization problem and used GAs to solve the problem in order to obtain globally optimal solution. To speed up the convergence process of the algorithm, we used the one-step fuzzy k-Modes algorithm in the crossover process instead of the traditional crossover operator. We tested the algorithm using two real world data sets from UCI Machine Learning Repository (Blake & Merz, 1998) and the experimental results have shown that genetic fuzzy k-Modes is very effective in identifying the inherent cluster structures in categorical data set if such structures exist.

References

Bezdek, J. (1974). Fuzzy mathematics in pattern classification, Ph.D. thesis, Ithaca, NY: Cornell University (April).

Blake, C., & Merz, C. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/mlearn/MLRepository.html.

Buhmann, J. (2003). Data clustering and learning. In M. Arbib (Ed.), *The handbook of brain theory and neural networks* (pp. 308–312). Cambridge, Massachusetts: The MIT Press.

Cormack, R. (1971). A review of classification. *Journal of the Royal Statistical Society. Series A (General), 134*(3), 321–367.

Cowgill, M., Harvey, R., & Watson, L. (1999). A genetic algorithm approach to cluster analysis. *Computers and Mathematics with Applications, 37*(7), 99–108.

Davis, L. (1991). *Handbook of genetic algorithms*. New York, USA: van Nostrand Reinhold.

Dunn, J. (1974). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics, 3*(3), 32–57.

Everitt, B., Landau, S., & Leese, M. (2001). *Cluster analysis* (4th ed.). New York: Oxford University Press.

Glover, F., & Laguna, M. (1997). *Tabu search*. Boston: Kluwer Academic Publishers.

Gordon, A. (1987). A review of hierarchical classification. *Journal of the Royal Statistical Society. Series A (General), 150*(2), 119–137.

Huang, Z., & Ng, M. (1999). A fuzzy k-modes algorithm for clustering categorical data. *IEEE Transactions on Fuzzy Systems, 7*(4), 446–452.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification, 2*, 193–218.

Jain, A., & Dubes, R. (1988). *Algorithms for clustering data*. Englewood Cliffs, New Jersey: Prentice Hall.

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

Krishna, K., & Narasimha, M. (1999). Genetic k-means algorithm. *IEEE Transactions on Systems, Man and Cybernetics, Part B, 29*(3), 433–439.

Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal, 26*(4), 354–359.

Ng, M., & Wong, J. (2002). Clustering categorical data sets using tabu search techniques. *Pattern Recognition, 35*(12), 2783–2790.

Ruspini, E. (1969). A new approach to clustering. *Information and Control, 15*, 22–32.

Zhao, L., Tsujimura, Y., & Gen, M. (1996). Genetic algorithm for fuzzy clustering. In *Proceedings of IEEE International Conference on Evolutionary Computation, 1996* (pp. 716–719). Nagoya Japan: IEEE.