Multiobjective Genetic Fuzzy Clustering of Categorical Attributes

Anirban Mukhopadhyay
Department of Computer Science and Engineering, University of Kalyani
Kalyani - 741235, India, anirbanbuba@yahoo.com

Ujjwal Maulik
Department of Computer Science and Engineering, Jadavpur University
Kolkata - 700032, India, drumaulik@jdvu.ac.in

Sanghamitra Bandyopadhyay
Machine Intelligence Unit, Indian Statistical Institute
Kolkata-700108, India, sanghami@isical.ac.in

Abstract

Most of the algorithms designed for categorical data clustering optimize a single measure of the clustering goodness. Such a single measure may not be appropriate for different kinds of data sets. Therefore, consideration of multiple, often conflicting, objectives appears to be natural for this problem. In this article a multiobjective genetic algorithm based approach for fuzzy clustering of categorical data is proposed. The performance of the proposed technique has been compared with that of the other well known categorical data clustering algorithms. For this purpose, various synthetic and real life categorical data sets have been considered. Statistical significance test has been conducted to establish the significant superiority of the proposed multiobjective approach.

1 Introduction

Clustering [9, 10, 12] is a popular unsupervised pattern classification approach in which a given data set is partitioned into a number of distinct groups based on some similarity/dissimilarity measures. Traditional data clustering algorithms are designed for such data sets where the dissimilarity between any two points of the data set is well defined. However, many real life data sets are categorical in nature, where no natural ordering can be found among the elements in the attribute domain. In such situations, the traditional algorithms, such as K-means [9], fuzzy C-means [2], etc. cannot be applied, because the concept of *mean* does not work for categorical attribute domains. Some variations of K-means type algorithms, namely Partitioning Around Medoids (PAM) or K-medoids [10], K-modes [7] and fuzzy K-modes [8] have been developed to cope with categorical data. However, all these algorithms rely on optimizing a single objective to obtain the partitioning. A single objective function may not work uniformly well for different kinds of categorical data. Hence it is natural to consider multiple objectives that need to be optimized simultaneously.

Genetic algorithms [3, 5] have been previously used in data clustering problems [11, 13]. However, most of them use a single objective to be optimized, which is hardly equally applicable to all kinds of data sets. In this article, the problem of fuzzy partitioning of categorical data set is modeled as multiobjective optimization (MOO) problem [3, 1], where search is performed over a number of, often conflicting, objective functions. Unlike single objective optimization, which yields a single best solution, in MOO the final solution set contains a number of Pareto-optimal solutions, none of which can be further improved on any one objective without degrading another [3]. Multiobjective Genetic Algorithms (MOGAs) are used in this regard in order to determine the appropriate cluster centers (modes) and the corresponding partition matrix. NSGA-II [4], a popular elitist MOGA, is used as the underlying optimization strategy. The two objective functions, the global fuzzy compactness of the clusters and fuzzy separation, are optimized simultaneously.

Another recent work on multiobjective clustering around (MOCK-AM) has been proposed in [6], where the authors had used strings of length equal to the number of data points, hence the search space is too large requiring more computational power. Also this method does not work well if there are overlaps in the data set.

Experiments have been carried out for a number of synthetic and real life categorical data sets. The performance of the proposed multiobjective technique has been compared with several well known algorithms and the single objective ones. Also statistical significance tests are conducted in order to confirm that the superior performance of the proposed technique is significant and does not occur by chance.

2 Multiobjective Optimization

The multiobjective optimization can be formally stated as [3]: Find the vector $\overline{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ of decision variables which will satisfy the m inequality constraints :

$$g_i(\overline{x}) \geq 0, \quad i = 1, 2, \ldots, m, \tag{1}$$

the p equality constraints

$$h_i(\overline{x}) = 0, \quad i = 1, 2, \ldots, p, \tag{2}$$

and optimizes the vector function

$$\overline{f}(\overline{x}) = [f_1(\overline{x}), f_2(\overline{x}), \ldots, f_k(\overline{x})]^T. \tag{3}$$

The constraints given in Eqns. 1 and 2 define the feasible region $\mathcal{F}$ which contains all the admissible solutions. Any solution outside this region is inadmissible since it violates one or more constraints. The vector $\overline{x}^*$ denotes an optimal solution in $\mathcal{F}$. In the context of multiobjective optimization, the difficulty lies in the definition of optimality, since it is only rare that we will find a situation where a single vector $\overline{x}^*$ represents the optimum solution to all the objective functions.

The concept of *Pareto optimality* comes handy in the domain of multiobjective optimization. A formal definition of Pareto optimality from the viewpoint of minimization problem may be given as follows: A decision vector $\overline{x}^*$ is called Pareto optimal if and only if there is no $\overline{x}$ that dominates $\overline{x}^*$, i.e., there is no $\overline{x}$ such that

$$\forall i \in \{1, 2, \ldots, k\}, f_i(\overline{x}) \leq f_i(\overline{x}^*), \text{ and}$$

$$\exists i \in \{1, 2, \ldots, k\}, f_i(\overline{x}) < f_i(\overline{x}^*).$$

In other words, $\overline{x}^*$ is Pareto optimal if there exists no feasible vector $\overline{x}$ which causes a reduction on some criterion without a simultaneous increase in at least one other. In this context, two other notions, *weakly non-dominated* and *strongly non-dominated* solutions are defined [3]. A point $\overline{x}^*$ is a weakly non-dominated solution if there exists no $\overline{x}$ such that $f_i(\overline{x}) < f_i(\overline{x}^*)$, for $i = 1, 2, \ldots, k$. A point $\overline{x}^*$ is a strongly non-dominated solution if there exists no $\overline{x}$ such that $f_i(\overline{x}) \leq f_i(\overline{x}^*)$, for $i = 1, 2, \ldots, k$, and for atleast one i, $f_i(\overline{x}) < f_i(\overline{x}^*)$. In general, pareto optimum usually admits a set of solutions called *non-dominated* solutions.

There are several modern techniques for multiobjective optimization. Among them, the GA based techniques such as NSGA-II [4], SPEA [17] and SPEA2 [16] are very popular. The present article uses NSGA-II as underlying multiobjective framework for developing the proposed fuzzy clustering algorithm.

3 Some Algorithms for Clustering Categorical Data

In this section, some popular clustering techniques used for clustering categorical data sets have been described.

3.1 Fuzzy K-modes

The fuzzy K-modes [8] algorithm is the extension of well-known fuzzy C-means [2] algorithm in categorical domain. Assume that $X = \{x_1, x_2, \ldots, x_n\}$ be a set of n objects having categorical attribute domains. Each object x_i, $i = 1, 2, \ldots, n$, is described by a set of p attributes $A_1, A_2, \ldots, A_p$. Let $DOM(A_j)$, $1 \leq j \leq p$, denotes the domain of j^{th} attribute and it consists of different q_j categories such as $DOM(A_j) = \{a_j^1, a_j^2, \ldots, a_j^{q_j}\}$. Hence the i^{th} categorical object is defined as $x_i = [x_{i1}, x_{i2}, \ldots, x_{ip}]$ where $x_{ij} \in DOM(A_j)$, $1 \leq j \leq p$.

The cluster centers in fuzzy C-means are replaced by cluster modes in fuzzy k-modes clustering. A mode is defined as follows: Let $Y = \{y_1, y_2, \ldots, y_v\}$ be a set of categorical objects with attributes $A_1, A_2, \ldots, A_p$. A mode of Y is a vector $O = [o_1, o_2, \ldots, o_p]$, $o_j \in DOM(A_j)$, $1 \leq j \leq p$, such that the following criterion is minimized.

$$\mathcal{D}(O, Y) = \sum_{i=1}^{v} D(O, y_i). \tag{4}$$

Here $D(O, y_i)$ denotes the dissimilarity measure between O and y_i. Note that O is not necessarily an element of set Y.

The aim of fuzzy K-modes algorithm is to cluster the data set X into K partitions so that the following criterion is minimized.

$$J_m(U, Z : X) = \sum_{k=1}^{n} \sum_{i=1}^{K} u_{ik}^m D(z_i, x_k), \tag{5}$$

where m is the fuzzy exponent. $U = [u_{ik}]$ denotes the $K \times n$ fuzzy partition matrix and u_{ik} (between 0 and 1) denotes the membership degree of k^{th} categorical object to the i^{th} cluster. $Z = \{z_1, z_2, \ldots, z_K\}$ represents the set of cluster centers (modes).

The main disadvantages of fuzzy K-modes clustering algorithms are (1) it depends much on the initial choice of the modes and (2) it often gets trapped into some local optimum.

3.2 K-medoids

Partitioning around medoids (PAM), also called K-medoids clustering [10], is a variation of K-means with the objective to minimize the within cluster variance W(K).

$$W(K) = \sum_{i=1}^{K} \sum_{p \in C_i} D(p, m_i) \tag{6}$$

Here m_i is the medoid of cluster C_i and *D(.)* denotes a dissimilarity measure. Medoid means the most centrally located point within a cluster, i.e., the point from which the summation of distances to other points of that cluster is minimum. The resulting clustering of the data set X is usually only a local minimum of $W(K)$.

3.3 K-modes

K-modes clustering [7] is the crisp version of the fuzzy K-modes algorithm. K-modes algorithm works similarly to K-medoids with the only difference that here, instead of medoids, modes are used to represent a cluster. K-modes algorithm minimizes the following objective function.

$$TC(K) = \sum_{i=1}^{K} \sum_{p \in C_i} D(p, v_i) \tag{7}$$

Here v_i denotes the mode of the cluster C_i.

3.4 Clustering Categorical Data based on Distance Vectors

Clustering Categorical Data based on Distance Vectors (CCDV) [15] is a recently proposed clustering algorithm for categorical attributes. CCDV sequentially extracts clusters from a given data set based on Hamming Distance (HD) vectors, with automatic evolution of number of clusters. The output of the algorithm does not depend on the order of the input data points.

Except these, hierarchical agglomerative clustering algorithms (single, average and complete linkage) are also used for clustering categorical data.

4 Proposed Multiobjective Fuzzy Clustering Technique

In this section, the distance measure between two feature vectors that is adopted in this article is described. Subsequently, the method of using NSGA-II for evolving a set of near-Pareto-optimal non-degenerate fuzzy partition matrices is described.

4.1 Distance Measure

This article adopts the following dissimilarity measure for all the algorithms considered. Let $x_i = [x_{i1}, x_{i2}, \ldots, x_{ip}]$, and $x_j = [x_{j1}, x_{j2}, \ldots, x_{jp}]$ be two categorical objects described by p categorical attributes. The distance measure between x_i and x_j, $D(x_i, x_j)$, can be defined by the total mismatches of the corresponding attribute categories of the two objects. Formally,

$$D(x_i, x_j) = \sum_{k=1}^{p} \delta(x_{ik}, x_{jk}), \tag{8}$$

where

$$\delta(x_{ik}, x_{jk}) = \begin{cases} 0 & \text{if } x_{ik} = x_{jk} \\ 1 & \text{if } x_{ik} \neq x_{jk}. \end{cases} \tag{9}$$

Note that $D(x_i, x_j)$ gives equal importance to all the categories of an attribute.

4.2 Chromosome Encoding and Population Initialization

Each chromosome is a sequence of attribute values representing the K cluster modes. If each categorical object has p attributes $\{A_1, A_2, \ldots, A_p\}$, the length of a chromosome will be $p \times K$, where the first p positions (or, genes) represent the p-dimensions of the first cluster mode, the next p positions represent those of the second cluster mode, and so on. Each chromosome of the initial population consists of the points randomly chosen from the data set.

4.3 Computation of Fitness functions

The fitness vector is composed of two fitness functions: global compactness π [14] of the clusters and fuzzy separation *Sep* [14] have been considered as the two objectives that need to be optimized simultaneously. For computing the measures, the modes encoded in a *chromosome* are first extracted. Let these be denoted as $z_1, z_2, \ldots, z_K$. The membership values u_{ik}, $i = 1, 2, \ldots, K$ and $k = 1, 2, \ldots, n$ are computed as follows [8]:

$$u_{ik} = \frac{1}{\sum_{j=1}^{K} \left(\frac{D(z_i, x_k)}{D(z_j, x_k)}\right)^{\frac{1}{m-1}}}, \text{ for } 1 \leq i \leq K;\ 1 \leq k \leq n, \tag{10}$$

where $D(z_i, x_k)$ and $D(z_j, x_k)$ are as described earlier. m is the weighting coefficient. (Note that while computing u_{ik} using Eqn. 10, if $D(z_j, x_k)$ is equal to zero for some j, then u_{ik} is set to zero for all $i = 1, \ldots, K$, $i \neq j$, while u_{jk} is set equal to one.) Subsequently, each mode encoded in a chromosome is updated to $z_i = [z_{i1}, z_{i2}, \ldots, z_{ip}]$ where

$z_{ij} = a_j^r \in DOM(A_j)$ using the following inequation [8]:

$$\sum_{k, x_{kj}=a_j^r} u_{ik}^m \;\geq \sum_{k, x_{kj}=a_j^t} u_{ik}^m, \quad 1 \leq t \leq q_j, \quad r \neq t. \tag{11}$$

and the cluster membership values are recomputed.

Thereafter, the variation σ_i and fuzzy cardinality n_i of the i^{th} cluster, $i = 1, 2, \ldots, K$, are calculated using the following equations [14]:

$$\sigma_i = \sum_{k=1}^{n} u_{ik}^m D(z_i, x_k), \quad 1 \leq i \leq K, \tag{12}$$

and

$$n_i = \sum_{k=1}^{n} u_{ik}, \quad 1 \leq i \leq K. \tag{13}$$

Now the global compactness π of the solution represented by the chromosome is computed as [14]:

$$\pi = \sum_{i=1}^{K} \frac{\sigma_i}{n_i} = \sum_{i=1}^{K} \frac{\sum_{k=1}^{n} u_{ik}^m D(z_i, x_k)}{\sum_{k=1}^{n} u_{ik}}. \tag{14}$$

To compute the other fitness function fuzzy separation Sep, the mode z_i of the i^{th} cluster is assumed to be the center of a fuzzy set $\{z_j | 1 \leq j \leq K, j \neq i\}$. The membership degree of each z_j to z_i, $j \neq i$, is computed as [14]:

$$\mu_{ij} = \frac{1}{\sum_{l=1, l\neq j}^{K} \left(\frac{D(z_j, z_i)}{D(z_j, z_l)}\right)^{\frac{1}{m-1}}}, \quad i \neq j. \tag{15}$$

The fuzzy separation is defined as [14]:

$$Sep = \sum_{i=1}^{K} \sum_{j=1, j\neq i}^{K} \mu_{ij}^m D(z_i, z_j). \tag{16}$$

Note that in order to obtain compact clusters, the measure π should be minimized. On the other hand, to get well separated clusters, the fuzzy separation Sep should be maximized. Hence the objective is to minimize π and to maximize Sep. These objectives are chosen as they are contradictory in nature, because minimizing π means increasing the compactness of the clusters, whereas maximizing Sep means increasing the inter-cluster separation. Thus they balance each other critically producing good solutions.

4.4 Genetic Operators

The *selection* operation used here is the crowded binary tournament selection used in NSGA-II. We have used conventional uniform *crossover* with random mask for generating the new offspring solutions from the chromosomes selected in the mating pool. For performing *mutation*, if a chromosome is selected to be mutated, the gene position that will undergo mutation is selected randomly. After that, the categorical value of that position is replaced by another random value chosen from the corresponding categorical domain. The most characteristic part of NSGA-II is its elitism operation, where the non-dominated solutions among the parent and child populations are propagated to the next generation. For details on the different genetic processes, the reader may refer to [4]. The near-Pareto-optimal strings of the last generation provide the different solutions to the clustering problem.

4.5 Obtaining Final Solution

As the multiobjective method produces a set of non-dominated solutions in the final generations, it is needed to obtain a solution from this set. In this article, following method is used in this purpose. For each non-dominated solution, first the clustering label vector is extracted from the solution by assigning each point to the cluster to which it has the highest membership. Thereafter the label vectors are reordered so that they correspond to each other. Subsequently the points which are assigned to the same cluster by atleast 50% of the clustering solutions are obtained. Taking this points as the training set, the remaining points are assigned a class label using k-nearest neighbour (k-nn) classification, where k is taken as 5. This way the final solution is generated.

5 Experimental Results

The performance of the proposed algorithm has been evaluated on two synthetic data sets (Data1 and Data2) and two real life data sets (Congressional Votes and Zoo), and compared with different algorithms, viz., fuzzy K-modes, K-modes, K-medoids, hierarchical average linkage clustering, single objective GA based clustering that minimizes π (SGA(π)), single objective GA based clustering that maximizes Sep (SGA(Sep)), and CCDV. Each algorithm has been run for ten times and mean value of %CP scores (described later) obtained over ten runs of the algorithms have been reported.

5.1 Performance Measure

The performance of the algorithms have been measured with respect to a term %CP which is defined as the percentage of pairs of points that have been correctly clustered together (i.e., which actually belong to the same cluster, and have been identified as such by the algorithm).

5.2 Input Parameters

The GA based algorithms are run for 100 generations with population size 50. The crossover and mutation probabilities are fixed at 0.8 and 0.1, respectively. The fuzzy K-modes, K-modes and K-medoids algorithms have been run for 100 iterations unless they converge before that.

5.3 Synthetic Data Sets

Data1: This synthetic dataset has a one-layer clustering structure with 15 attributes, 250 points and 5 clusters (50 points in each cluster). Each cluster has random categorical values selected from $\{1, 2, 3, 4, 5\}$ in a distinct continuous set of 12 attributes, while the rest attributes are set to 0.

Data2: This is a synthetic data set with 100 points, 10 attributes and 4 clusters (25 points in each cluster). For each cluster, 2 random attributes of the points of that cluster are zero valued and the remaining attributes have values in the range $\{1, 2, \ldots, 5\}$.

5.4 Real Life Data Sets

Congressional Votes: This data set is the United States Congressional voting records in 1984. Total number of records is 435. Each row corresponds to one Congress man's votes on 16 different issues (e.g., education spending, crime etc.). All attributes are boolean with Yes (that is, 1) and No (that is, 0) values. The data set contains records for 168 Republicans and 267 Democrats.

Zoo: The Zoo data consists of 101 instances of animals in a zoo with 17 features. The name of the animal constitutes the first attribute. This attribute is neglected. There are 15 boolean attributes corresponding to different characteristics of animals. The character attribute corresponds to the number of legs lying in the set $\{0, 2, 4, 5, 6, 8\}$. The data set consists of 7 different classes of animals.

The real life data sets are obtained from the UCI Machine Learning Repository (www.ics.uci.edu/~mlearn/MLRepository.html).

5.5 Results

Clustering results in terms of %CP scores on synthetic and real life data sets using different algorithms are reported in Table 1. From the table it can be observed that the proposed multiobjective genetic clustering algorithm gives the best %CP scores for all the data sets. For Data1, Data2, Votes and Zoo data sets, the proposed method provides average %CP scores of 100%, 91.54%, 82.98% and 97.05%, respectively. It is evident from the tables that for all the data sets, the proposed multiobjective method consistently outperforms all other algorithms in terms of %CP scores.

Table 1. Average %CP scores for different data sets

Algorithm	Data1	Data2	Votes	Zoo
Fuzzy C-modes	95.31	76.27	77.82	89.78
K-modes	88.95	73.91	75.38	88.24
K-medoids	93.57	77.36	74.92	88.47
Average linkage	100	84.48	81.77	93.48
CCDV	100	69.92	80.36	95.05
SGA (π)	96.11	79.49	79.61	92.17
SGA (Sep)	91.17	77.09	80.39	91.80
MOGA (π, Sep)	100	91.54	82.98	97.05

Table 2. Objective function values and %CP scores for Data1

Algorithm	π	Sep	ARI
SGA (π)	11.29	13.44	96.11
SGA (Sep)	11.57	16.39	91.17
MOGA (π,Sep)	11.34	15.38	100

Table 2 reports another interesting observation. Here the %CP scores for single objective and multi-objective GA based algorithms have been shown for Data1 data set. The final objective function values are also reported. As expected, the single objective GA based algorithm that minimizes π only, produces the minimum π value (11.29), whereas, the single objective GA based method that maximizes Sep only, gives the maximum Sep value (16.39). The proposed multi-objective GA based technique provides slightly poorer π (11.34) and Sep (15.38) values, however, in terms of the $\%CP$ scores, the proposed technique provides the best result (100). This signifies the importance of optimizing both π and Sep simultaneously instead of optimizing them separately to get good clustering solution.

5.6 Statistical Significance Test

In this article, we have used one way ANOVA (ANalysis Of VAriance) at the 5% significance level, to compare the mean %CP values produced by different algorithms in order to test the statistical significance of clustering solutions. Eight groups have been created for each data set corresponding to eight algorithms considered here. Each group consists of %CP values obtained by ten consecutive runs of the corresponding algorithm. The null hypothesis is that there are no significant differences among the mean %CP values produced by all the algorithms, and the alternative hypothesis is that there are significant differences in mean %CP values for at least two methods.

Table 3 reports the ANOVA results for the four data sets

Table 3. ANOVA result for different data sets

Data set	SS			MS		F-Stat	P-value	F-critical
	Btn Gr	Wthn Gr.	Total	Btn Gr	Wthn Gr.			
Data1	1264.91	189.03	1453.94	180.70	2.63	68.83	2.38E-29	2.14
Data2	3092.99	306.06	3399.05	441.86	4.25	103.94	4.80E-35	2.14
Votes	585.51	537.51	1122.99	83.64	7.47	11.20	1.66E-09	2.14
Zoo	685.12	209.62	894.74	97.87	2.91	33.62	2.82E-20	2.14

considered here. The sum of squares (SS) and the mean square (MS) of variances of both between groups and within groups are reported. The statistic to check the null hypothesis (F statistic) is the ratio of the mean squares between groups to the mean squares within groups. A high value for the F statistic (greater than F-critical) allows us to reject the null hypothesis. For example, ANOVA result for Data1 data set shows that the value of the F statistic (68.83) is greater than its critical value (F-critical = 2.14) and probability (P-value) that this result occurred by chance is very less (2.38E-29). This is extremely strong evidence against the null hypothesis, indicating that the better mean %CP value obtained by the multiobjective fuzzy clustering technique is only due to goodness of the algorithm and not by chance. This is true for all other data sets also.

6 Conclusions

In this article a multiobjective genetic algorithm based fuzzy clustering algorithm that optimizes the fuzzy compactness and fuzzy separation of the clusters simultaneously, has been proposed for clustering categorical data. The algorithm is designed on the framework of NSGA-II, a popular multiobjective GA. The performance of the proposed method has been compared with other well known clustering techniques on several synthetic and real life categorical data sets to establish its superiority. Also the use of multiple objectives rather than single objective has been justified. Statistical significance test has been carried out in order to judge the statistical significance of the clustering solutions. As a scope for future research, use of multiobjective algorithms other than NSGA-II is to be studied.

References

[1] S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay. Multiobjective genetic clustering for pixel classification in remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 45(5):1506–1511, 2007.

[2] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum, New York, 1981.

[3] K. Deb. *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons, Ltd, England, 2001.

[4] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.

[5] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, New York, 1989.

[6] J. Handl and J. Knowles. Multiobjective clustering around medoids. In *Proc. IEEE Cong. Evolutionary Computation*, volume 1, pages 632–639, 2005.

[7] Z. Huang. Clustering large data sets with mixed numeric and categorical values. In *First Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Singapur, 1997. World Scientific.

[8] Z. Huang and M. K. Ng. A fuzzy k-modes algorithm for clustering categorical data. *IEEE Transactions on Fuzzy Systems*, 7(4), 1999.

[9] A. K. Jain and R. C. Dubes. Data clustering: A review. *ACM Computing Surveys*, 31, 1999.

[10] L. Kaufman and P. Roussenw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, NY, US, 1990.

[11] U. Maulik and S. Bandyopadhyay. Genetic algorithm based clustering technique. *Pattern Recognition*, 33:1455–1465, 2000.

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

[13] U. Maulik and S. Bandyopadhyay. Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification. *IEEE Transactions on Geoscience and Remote Sensing*, 41(5):1075– 1081, 2003.

[14] G. E. Tsekouras, D. Papageorgiou, S. Kotsiantis, C. Kalloniatis, and P. Pintelas. Fuzzy clustering of categorical attributes and its use in analyzing cultural data. *Transactions on Engineering, Computing and Technology*, pages 202–206, 2004.

[15] P. Zhang, X. Wang, and P. X. Song. Clustering categorical data based on distance vectors. *The Journal of the American Statistical Association*, 101(473):355–367, 2006.

[16] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001.

[17] E. Zitzler and L. Thiele. An evolutionary algorithm for multiobjective optimization: The Strength Pareto approach. Technical Report 43, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 1998.