Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I
IMECS 2008, 19-21 March, 2008, Hong Kong

Genetic Algorithm and Simulated Annealing based Approaches to Categorical Data Clustering

Indrajit Saha * and Anirban Mukhopadhyay †

Abstract—**Recently, categorical data clustering has been gaining significant attention from researchers, because most of the real life data sets are categorical in nature. In contrast to numerical domain, no natural ordering can be found among the elements of a categorical domain. Hence no inherent distance measure, like the Euclidean distance, would work to compute the distance between two categorical objects. In this article, genetic algorithm and simulated annealing based categorical data clustering algorithms have been proposed. The performances of the proposed algorithms have been compared with that of different well known categorical data clustering algorithms and demonstrated for a variety of artificial and real life categorical data sets.** ***Keywords: Genetic Algorithm based Clustering (GAC), Simulated Annealing based Clustering (SAC), K-medoids Algorithm, Minkowski score.***

1 Introduction

Genetic algorithms (GA) [1, 2, 3] are randomized search and optimization techniques guided by the principles of evolution and natural genetics, having a large amount of implicit parallelism. GAs perform search in complex, large and multimodal landscapes, and provide near-optimal solutions for objective or fitness function of an optimization problem. The algorithm starts by initializing a population of potential solutions encoded into strings called chromosomes. Each solution has some fitness value based on which the fittest parents that would be used for reproduction are found (survival of the fittest). The new generation is created by applying genetic operators like crossover (exchange of information among parents) and mutation (sudden small change in a parent) on selected parents. Thus the quality of population is improved as the number of generations increases. The process continues until some specific criterion is met or the solution converges to some optimized value. Simulated Annealing (SA) [4] is a popular search algorithm, utilizes the principles of statistical mechanics regarding the behaviour of a large number of atom at low temperature, for finding minimal cost solutions to large optimization problems by minimizing the associated energy. In statistical mechanics, investigating the ground states or low-energy states of matter is of fundamental importance. These states are achieved at very low temperatures. However, it is not sufficient to lower the temperature alone since this results in unstable states. In the annealing process, the temperature is first raised, then decreased gradually to a very low value (T_{min}), while ensuring that one spends sufficient time at each temperature value. This process yields stable low-energy states. Geman and Geman [5] provided a proof that SA, if annealed sufficiently slow, converges to the global optimum. Being based on strong theory, SA has been applied in diverse areas by optimizing a single criterion. Clustering [6, 7, 8, 9] is a useful unsupervised data mining technique which partitions the input space into K regions depending on some similarity/dissimilarity metric where the value of K may or may not be known a priori. K-means [6] is a traditional partitional clustering algorithm which starts with K random cluster centroids and the centroids are updated in successive iterations by computing the numerical averages of the feature vectors in each cluster. The objective of the K-means algorithm is to maximize the global compactness of the clusters. K-means clustering algorithm cannot be applied for clustering categorical data sets, where there is no natural ordering among the elements of an attribute domain. Thus no inherent distance measures, such as Euclidean distance, can be used to compute the distance between two feature vectors [10, 11, 12, 13, 14]. Hence it is not feasible to compute the numerical average of a set of feature vectors. To handle such categorical data sets, a variation of K-means algorithm, namely K-medoids clustering has been proposed in [15]. In K-medoids algorithm, instead of computing the mean of feature vectors, a representative feature vector (cluster medoid) is selected for each cluster. A cluster medoid is defined as the most centrally located element in that cluster, i.e., it is the point from which the distance of the other points of the cluster is the minimum. K-medoids algorithm is also known as Partitioning Around Medoids (PAM) [15]. A major disadvantage of K-means, K-medoids clustering algorithms is that these algorithms often tend to converge to local optimum solutions. They optimize a single objective function. Motivated by this

*Mallabhum Institute of Technology, Bishnupur, Bankura, West Bengal. Department of Information Technology. *Email* : *indra_raju@yahoo.co.in*

†University of Kalyani, Nadia, West Bengal. Department of Computer Science and Engineering. *Email* : *anirbanbuba@yahoo.com*

fact, here we are using a global optimization tool like genetic algorithm and simulated annealing and try to optimize the objective function, the K-medoids error function by reducing the fact. The superiority of the proposed method over hierarchical clustering [16] and K-medoids clustering algorithm has been demonstrated on different synthetic and real life data sets. The rest of the article is organized as follows: Section 2 describes some well known algorithms for categorical data clustering. In Section 3, the distance metric used in this article to compute the distance between two categorical objects has been described. In section 4 and 5, a brief introduction to genetic algorithm and simulated annealing have been presented along with proposed genetic algorithm based clustering and simulated annealing based clustering methods for categorical attributes. Section 6 presents the experimental results conducted on several synthetic and real life data sets. Finally, Section 7 concludes the article.

2 Categorical Data Clustering Algorithms

This section describes some Hierarchical and Partitional clustering algorithms used for categorical data.

2.1 Complete-Linkage Clustering

The complete-linkage (CL) Algorithm is also called the maximum method or the farthest neighbor method [16]. It is obtained by defining the distance between two clusters to be the largest distance between a sample in one cluster and a sample in the other cluster. If C_i and C_j are clusters, we define

$$D_{CL}(C_i, C_j) = \max_{a \in C_i, b \in C_j} d(a, b) \tag{1}$$

2.2 Average-Linkage Clustering

The average-linkage (AL) clustering algorithm, also known as the unweighted pair-group method using arithmetic averages (UPGMA) [16], is one of the most widely used hierarchical clustering algorithms. The average-linkage algorithm is obtained by defining the distance between two cluster to be the average distance between a pont in one cluster and a point in the other cluster. Formally, if C_i is a cluster with n_i members and C_j is a cluster with n_j members, the distance between the clusters is

$$D_{AL}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{a \in C_i, b \in C_j} d(a, b). \tag{2}$$

2.3 K-medoids Clustering

Partitioning around medoids (PAM), also called K-medoids clustering [15], is a variation of K-means with the objective to minimize the within cluster variance W(K).

$$W(K) = \sum_{i=1}^{K} \sum_{x \in C_i} D(x, m_i) \tag{3}$$

Here m_i is the medoid of cluster C_i and $D(x, m_i)$ denotes the distance between the point x and m_i. K denotes the number of clusters. The resulting clustering of the data set X is usually only a local minimum of $W(K)$. The idea of PAM is to select K representative points, or medoids, in X and assign the rest of the data points to the cluster identified by the nearest medoid. Initial set of K medoids are selected randomly. Subsequently, all the points in X are assigned to the nearest medoid. In each iteration, a new medoid is determined for each cluster by finding the data point with minimum total distance to all other points of the cluster. After that, all the points in X are reassigned to their clusters in accordance with the new set of medoids. The algorithm iterates until $W(K)$ does not change any more.

3 Distance Metric

As discussed earlier, absence of any natural ordering among the elements of a categorical attribute domain prevents us to apply any inherent distance measure like Euclidean distance, to compute the distance between two categorical objects [17]. In this article following distance measure has been adopted for all the algorithms considered. Let $x_i = [x_{i1}, x_{i2}, ..., x_{ip}]$, and $x_j = [x_{j1}, x_{j2}, ..., x_{jp}]$ 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 number of mismatches of the corresponding attribute categories of the two objects. Formally,

$$D(x_i, x_j) = \sum_{k=1}^{p} \delta(x_{ik}, x_{jk}) \tag{4}$$

where

$$\delta(x_{ik}, x_{jk}) = \begin{cases} 0 & \text{if } x_{ik} = x_{jk} \\ 1 & \text{if } x_{ik} \neq x_{jk} \end{cases} \tag{5}$$

Note that $D(x_i, x_j)$ gives equal importance to all the categories of an attribute. However, in most of the categorical data sets, the distance between two data vectors depends on the nature of the data sets. Thus, if a distance matrix is precomputed for a given data set, the algorithms can adopt this for computing the distances.

4 Genetic Algorithm based Clustering: GAC

4.1 Basic Principle

The searching capability of GAs has been used in this article for the purpose of appropriately determining a

fixed number K of cluster centers in $\Re^n$; thereby suitably clustering the set of n unlabelled points. The clustering metric that has been adopted is the sum of the distances of the points from their respective cluster centers. Mathematically, the clustering metric ζ for the K clusters $C_1, C_2, ..., C_K$ is given by

$$\zeta(C_1, C_2, ..., C_K) = \sum_{i=1}^{K} \sum_{x_j \in C_i} D(x_j, z_i), \quad (6)$$

where D is the distance metric. The task of the GA is to search for the appropriate cluster centers $z_1, z_2, ..., z_K$such that the clustering metric ζ is minimized. The basic steps of GAs, which are also followed in the GA-clustering (GAC) algorithm. These are now described in detail.

4.2 Chromosome representation

Each chromosome has K genes and each gene of the chromosome has an allele value chosen randomly from the set $\{1, 2, \ldots, n\}$, where K is the number of clusters and n is the number of points. Hence a chromosome is represented as a vector of indices of the points in the data set. Each point index in a chromosome implies that the corresponding point is a cluster medoid.

Example 1. Let K = 6, i.e., Then the chromosome

51 72 18 15 29 32

represents the indices of six points qualified for cluster medoids. A chromosome is valid if no point index occurs more than once in the chromosome.

4.3 Population initialization

The K cluster medoids encoded in each chromosome are initialized to K randomly chosen points from the data set. This process is repeated for each of the P chromosomes in the population, where P is the size of the population.

4.4 Fitness computation

The fitness computation process consists of two phases. In the first phase, the clusters are formed according to the centers encoded in the chromosome under consideration. This is done by assigning each point $x_i, i = 1, 2, ..., n$ to one of the clusters C_j with center z_j such that

$$D(x_i, z_j) < D(x_i, z_p), \quad p = 1, 2, .., K, \quad and \quad p \neq j, \quad (7)$$

where D is the distance metric. All ties are resolved arbitrarily. After the clustering is done, the cluster medoids encoded in the chromosome are replaced by the points having minimum total distance to the points of the respective clusters. In other words, for cluster C_i, the new medoid is point x_t where,

$$x_t = \arg \min_{x_j \in C_i} \sum_{x_k \in C_i} D(x_j, x_k). \quad (8)$$

Hence the ith gene in the chromosome is replaced by t. Subsequently, the clustering metric ζ computed as follows:

$$\zeta = \sum_{i=1}^{K} \zeta_i, \quad (9)$$

where,

$$\zeta_i = \sum_{x_j \in C_i} D(x_j, z_i). \quad (10)$$

The fitness function is defined as $f = \zeta$, so that minimization of the fitness function leads to minimization of ζ.

4.5 Selection

The selection process selects chromosomes from the mating pool directed by the survival of the fittest concept of natural genetic systems. In the proportional selection strategy adopted in this article, a chromosome is assigned a number of copies, which is proportional to its fitness in the population, that go into the mating pool for further genetic operations. Tournament selection is one common technique that implements the proportional selection strategy.

4.6 Crossover

Crossover is a probabilistic process that exchanges information between two parent chromosomes for generating two child chromosomes. In this article single point crossover with a fixed crossover probability of μ_c is used. For chromosomes of length l, a random integer, called the crossover point, is generated in the range [1, l-1]. The portions of the chromosomes lying to the right of the crossover point are exchanged to produce two offspring chromosomes.

4.7 Mutation

Each chromosome undergoes mutation with a fixed probability μ_m. The mutation operation has been defined as following: From the string to be mutated, a random element is chosen and it is replaced by a different index of point in the range $\{1, \ldots, n\}$ such that no element is duplicated in the string.

4.8 Termination criterion

In this article the processes of fitness computation, selection, crossover, and mutation are executed for a fixed number of iterations. The best string seen upto the last generation provides the solution to the clustering problem. We have implemented elitism at each generation by preserving the best string seen upto that generation in

a location outside the population. Thus on termination, this location contains the centers of the final clusters.

4.9 GAC Algorithm.

```
1) Encoding
Generation=100
while( i<Generation)
2) Initial population creation of size 20.
3) Fitness value calculation.
4) Selection.
5) Crossover with probability = 0.8.
6) Mutation with probability = 0.1.
i=i+1;
End while
```

5 Simulated Annealing based Clustering: SAC

Simulated annealing (SA) [4] is an optimization tool which has successful applications in a wide range of combinatorial optimization problems. This fact has motivated researchers to use SA in simulation optimization. However SA still needs to evaluate the objective function values accurately, and there have been few theoretical studies for the SA algorithm when the objective function is estimated through simulation. There are some applications of SA in clustering [18, 19]. In this article, we have used SA for designing a categorical data clustering method. The algorithm is named as simulated annealing clustering (SAC). This algorithm is described below.

5.1 String representation

In this article, a configuration (string) is represented in similar way a chromosome is represented in GAC, i.e., the string has length K and Each element of the string is chosen randomly from the set $\{1, 2, \ldots, n\}$, where K is the number of clusters and n is the number of points. Hence a string is represented as a vector of indices of the points in the data set. Each point index in a string indicates that the corresponding point is a cluster medoid. A string is valid if no point index occurs more than once in it.

5.2 Fitness computation

The fitness of a string is computed similarly as in GAC, i.e., first the encoded medoids are used for cluster assignments and the string is updated using new medoids. Thereafter the fitness (K-medoid error function) is computed as per Eqn. 6.

5.3 Perturbation

The current string undergoes perturbation as follows: the position of perturbation is chosen randomly and the value of that position is replaced by some other value chosen randomly from the set $\{1, 2, \ldots, n\}$. This way, perturbation of a string yields a new string.

5.4 SAC Algorithm

```
1) Generation of initial String Randomly = q.
T = T_{max}.
2) E(q,T) = Fitness of q.
while( T >= T_{min} )
for i = 1 to k
3) s = Perturb ( q ).
4) E(s,T) = Fitness of s.
if (E(s,T) - E(q,T) < 0 )
5) Set q = s and E(q,T) = E(s,T) .
else
6) Set q = s and E(q,T) = E(s,T) with
probability exp-( E(s,T) - E(q,T) )/T
End for
T= T*r. /* 0 < r < 1 */
End while
```

6 Experimental Results

The performance of the proposed algorithm has been evaluated on two synthetic data sets (Cat01 and Cat02) and three real life data sets (Tic-tac-toe, Zoo and Soybean). The proposed SAC scheme has been compared with different algorithms, viz., Complete-linkage, Average-linkage and K-medoids. Each algorithm has been run for 20 times. The average of Minkowski score (described later) has been reported.

6.1 Synthetic Data Sets

Cat01: The *'Cat01'* is a synthetic data set which consists of 20 instances with 5 features. The data set has 2 clusters. ***Cat02:*** The *'Cat02'* data is also a synthetic data set which consists of 132 instances with 5 features. This data set has 3 clusters. The synthetic data sets are generated using a web based data generation tool[1].

6.2 Real Life Data Sets

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 the presence of hair, feathers, eggs, milk, backbone, fins, tail; and whether airborne, aquatic, predator, toothed, breathes, venomous, domestic and catsize. 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.This is a categorical pattern. ***Soybean:*** The Soybean data set contains 47 data points on diseases in soybeans. Each data point has 35 categorical attributes and is classified as one of the four diseases, i.e.,

[1] http://www.datgen.com

number of clusters in the data set is 4.This is a categorical pattern. ***Tic-tac-toe:*** The Tic-tac-toe data consists of 958 instances of legal tic-tac-toe endgame boards with 10 features where each corresponding to one tic-tac-toe square. Out of this 10 features last one is a class identifier. Others are corresponding to the top-left-square, top-middle-square, top-right-square, middle-left-square, middle-middle-square, middle-right-square, bottom-left-square, bottom-middle-square and bottom-right-square. Those squares are identified by x = player x has taken or o = player o has taken or b = blank. The real life data sets mentioned above were obtained from the UCI Machine Learning Repository[2].

6.3 Input Parameters

The GAC algorithm is run for 100 generations with population size=20. The crossover and mutation probabilities are taken to be 0.8 and 0.1, respectively. The parameters of the SAC algorithm are as follows: T_{max}=100, T_{min}=0.01, r=0.9 and k=100. The K-medoids algorithm is run for 100 iterations unless it converges before that.

6.4 Performance Metric

Here, the performances of the clustering algorithms are evaluated in terms of the *Minkowski score*(MS). A clustering solution for a set of n elements can be represented by an $n \times n$ matrix C, where $C_{i,j} = 1$ if point i and j are in the same cluster according to the solution, and $C_{i,j} = 0$ otherwise. The Minkowski score of a clustering result C with reference to T , the matrix corresponding to the true clustering, is defined as

$$MS(T, C) = \frac{\| T - C \|}{\| T \|} \tag{11}$$

where

$$\| T \| = \sqrt{\sum_i \sum_j T_{i,j}}$$

The Minkowski score is the normalized distance between the two matrices. Lower Minkowski score implies better clustering solution, and a perfect solution will have a score zero.

6.5 Results

The Tables 1 and 2 report the average values for Minkowski scores obtained by different algorithms over 20 runs on synthetic and real life data sets, respectively. It is evident from the tables that both the GAC and SAC clustering methods consistently outperform the hierarchical clustering and K-medoids algorithms. The performances of GAC and SAC are comparable to each other.

[2]http://www.ics.uci.edu/∼ mlearn/MLRepository.html

Table 1: Average minkowski score for synthetic data sets.

Algorithm	Cat01	Cat02
CL	0.9953	1.3520
AL	0.9492	1.1384
K-medoids	0.6617	0.9425
GAC	0.0000	0.4522
SAC	0.0000	0.4922

Table 2: Average minkowski score for real life data sets.

Algorithm	Zoo	Soybean	Tic-tac-toe
CL	1.1642	1.3954	1.4503
AL	1.1262	1.2432	1.0520
K-medoids	0.6920	0.7851	0.6372
GAC	0.4832	0.2522	0.4512
SAC	0.5340	0.2982	0.5001

7 Conclusions

In this article, genetic algorithm based clustering and simulated annealing based clustering algorithms for clustering categorical data around medoids, have been proposed. The proposed algorithms effectively optimizes the K-medoids error function globally. the performance of the proposed algorithms have been demonstrated for different synthetic and real life data sets and also compared with that of other well-known clustering algorithms used for categorical data clustering. The results indicate the proposed genetic algorithm and simulated annealing based algorithms can be efficiently used for clustering different categorical data sets.

References

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

[2] L. Davis, ed., *Handbook of Genetic Algorithms.* New York: Van Nostrand Reinhold, 1991.

[3] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles.* Reading: Addison-Wesley, 1974.

[4] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 22, pp. 671–680, 1983.

[5] S. Geman and D. Geman, "Stochastic relaxation, Gribbs distributions and the Bayesian restoration of image.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

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

[7] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and valid-

ity indices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1650–1654, 2002.

[8] S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay, "Multiobjective genetic clustering for pixel classification in remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 5, pp. 1506–1511, 2007.

[9] U. Maulik and S. Bandyopadhyay, "Genetic algorithm based clustering technique," *Pattern Recognition*, vol. 33, pp. 1455–1465, 2000.

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

[11] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining Knowledge Discovery*, vol. 2, no. 3, pp. 283–304, 1998.

[12] Z. Huang and M. K. Ng, "A fuzzy k-modes algorithm for clustering categorical data," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 4, 1999.

[13] P. Zhang, X. Wang, and P. X. Song, "Clustering categorical data based on distance vectors," *The Journal of the American Statistical Association*, vol. 101, no. 473, pp. 355–367, 2006.

[14] V. Ganti, J. Gehrke, and R. Ramakrishnan, "CACTUS-clustering categorical data using summeries," in *Proc. ACM SIGKDD*, 1999.

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

[16] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. London: Prentice-Hall, 1982.

[17] E.-G. T. L. Vermeulen-Jourdan, C. Dhaenens, "Clustering nominal and numerical data: a new distance concept for an hybrid genetic algorithm," in *Proc. 4th European Conf. Evolutionary Computation on Combinatorial Optimization (LNCS 3004)*, (Coimbra, Portugal), pp. 220–229, April 2004.

[18] S. Bandyopadhyay, "Simulated annealing using a reversible jump markov chain monte carlo algorithm for fuzzy clustering," *IEEE Transactions on Knowledge and data Engineering*, vol. 17, no. 4, pp. 479–490, 2005.

[19] S. Bandyopadhyay, U. maulik, and M. Pakhira, "Clustering using simulated annealing with probabilistic redistribution," *Int. J. Patt. Rec. Art. Int*, vol. 15, no. 2, pp. 269–285, 2001.