http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 11 (10): 1409-1417, 2012
ISSN 1812-5638 / DOI: 10.3923/itj.2012.1409.1417

A Fast Evolutionary Algorithm for Automatic Evolution of Clusters

[1]Singh Vijendra, [2]K. Ashiwini and [3]Sahoo Laxman
[1]Department of Computer Science and Engineering, India
[2]Faculty of Engineering and Technology, Mody Institute of Technology and Science,
Lakshmangarh, Rajasthan, India
[3]School of Computer Engineering, KIIT University, Bhubaneswar, India

Abstract: This paper proposed an evolutionary clustering algorithm which can automatically determine the number of clusters present in a data set. The chromosomes are represented as strings of real numbers, encode the centers of a fixed number of clusters. The searching capability of evolutionary clustering is exploited in order to search for appropriate cluster centers in the feature space such that a similarity metric of the resulting clusters is optimized. The proposed clustering approach called Fast Automatic Clustering Evolution (FACE) in data set. To obtain a speedup over linear search in high dimensional data a randomized *k*-d trees based nearest neighbor search is used. The chromosomes are able to exchange their gene values according to nearest cluster centers and relation among genes in crossover operator. Mutation operator replaced the mutation gene value with respect to nearest neighbor cluster. Adaptive probabilities of crossover and mutation are employed to prevent the convergence of the GA (Genetic Algorithm) to a local optimum. The Adjusted-Rand Index is used as a measure of the validity of the clusters. Effectiveness of the proposed algorithm is demonstrated for both artificial and real-life data sets. The experimental result demonstrates that the proposed clustering algorithm (FACE) has high performance, effectiveness and flexibility. The proposed evolutionary algorithm is able for clustering low to high dimensional data set.

Key words: Clustering, evolutionary computation, randomized *k*-d trees, genetic algorithm, adaptive probabilities, high dimensional data

INTRODUCTION

Clustering is a common data analysis task that aims to partition objects into homogeneous groups (classes). Clustering detects new classes of data without any a priori knowledge (Han and Kamber, 2004). Therefore, clustering is often also called unsupervised learning in contrast to classification where the classes are predefined and which is often also called supervised learning. The task of clustering has been effectively applied in statistics (Jain and Dubes, 1988), machine learning (Jain *et al*., 1999), scientific disciplines (Everitt *et al*., 2001), quality control (Zarandi and Alaeddini, 2010) and databases. Clustering techniques are categorized into several different approaches: partitioning, hierarchical, density based, subspace based, grid based and soft computing. Partitioning clustering algorithms compute a flat partition of the data set. Such Partitioning clustering are *k*-means, EM (Mclachlan and Krishnan, 1997), AntClust (Ouadfel and Batouche, 2007) and parallel and distributed *k*-mean (Hemalatha and Vivekanandan, 2008). Garg and Jain (2006) performed a study on clustering algorithms based on partition and variation of *k*-means algorithm. Velmurugan and Santhanam (2011) explored the behavior of some of the partition based clustering algorithms in their survey. Hierarchical clustering (Guha *et al*., 1999; Karypis *et al*., 1999; Ranjan and Khalil, 2007; Arora *et al*., 2009) builds a cluster hierarchy or, in other words, a tree of clusters. Hierarchical clustering can be further categorized into agglomerative (Abghari *et al*., 2009) and divisive methods based on how it is constructed. Subspace clustering technique (Agrawal *et al*., 1998; Friedman and Meulman, 2004; Zhou *et al*., 2007; Chu *et al*., 2009; Chu *et al*., 2010; Vijendra *et al*., 2010) searches interesting subsets of objects and their associated subsets of attributes. These techniques can be divided in two categories: partition based and density based approaches. Density-based clustering methods (Ester *et al*., 1996; Hinneburg and Keim, 1998; Sun *et al*., 2008; Yousria *et al*., 2009; Deng *et al*., 2010;

Corresponding Author: Singh Vijendra, Department of Computer Science and Engineering, Faculty of Engineering and Technology, Mody Institute of Technology and Science, Lakshmangarh, Rajasthan, India

Vijendra, 2011) group neighboring objects into clusters based on local density conditions rather than proximity between objects. In Grid-based clustering (Wang *et al*., 1997; Nagesh *et al*., 2001; Milenova and Campos, 2002; Pilevar and Sukumar, 2005) the objectspace rather than the data is divided into a grid. Genetic Algorithms (GAs) Goldberg (1989) are randomized search and optimization techniques guided by the principles of evolution and natural genetics. Murthy and Chowdhury (1996) have considered a partition to be encoded as a string of length *n*. A comparison of the performance of their algorithm with that of the *k*-means algorithm is also performed. Bandyopadhyay and Maulik, (2002) proposed a genetic algorithm-based efficient clustering technique that utilizes the principles of *k*-Means algorithm. It avoids major limitation of getting stuck at locally optimal values. Gautam and Chaudhur (2004) proposed a Genetic Clustering Algorithm with two-phase process. At the first phase the original data set is decomposed into a number of fragmented clusters. At the second phase hierarchical cluster merging algorithm is used for final clustering. Mitra (2004) proposed an evolutionary rough c-means clustering algorithm that model clusters as c rough sets, expressed in terms of upper and lower approximations. Lin *et al*. (2005) proposed a genetic clustering algorithm that selects the cluster centers directly from the data set. Ioannis *et al*. (2005) proposed a rule based Novel Evolutionary Algorithm (NOCEA) that evolves individuals of variable-length consisting of disjoint and axis-aligned hyper-rectangular rules with homogeneous data distribution. A new point symmetry distance based genetic clustering algorithm is proposed by (Bandyopadhyay and Saha, 2007), which incorporates both the Euclidean distance as well as a measure of symmetry with respect to a point in its computation. Nguyen and Cios (2008) introduced a novel clustering algorithm that combines the best characteristics of the *k*-means and EM algorithms but avoids their weaknesses. Lai and Chang (2009) proposed a clustering based approach using a Hierarchical Evolutionary Algorithm (HEA) for medical image segmentation. Vijendra *et al*. (2010) presented a Euclidean distance-based Genetic Clustering Algorithm (GCA) that finds a globally optimal partition of a given data sets into a specified number of clusters. Singh *et al*. (2011) also proposed a genetic algorithm with chromosome reorganize that removes the degeneracy of chromosome which makes the evolution process converge fast. In this paper, an evolutionary clustering algorithm is introduced to enhance the performance of clustering. Proposed evolutionary algorithm uses multiple randomized *k*-d trees to improve the performance for nearest neighbor in high dimensional data.

CLUSTERING

Data clustering is an NP complete problem of finding groups in data by minimizing some measure of dissimilarity. A cluster is a set of points that are similar and points from different clusters are not similar. Let the set of n points $\{x_1, x_2 \ldots, x_n\}$ be represented by the set S and the K clusters be represented by $C_1, C_2 \ldots C_k$ Then:

$$C_i \neq \phi \text{ for } i=1,\ldots,K, C_i \cap C_j = \phi \text{ for } i=1,\ldots,K, j=1,\ldots,K \text{ and } i \neq j, \bigcup_{i=1}^{K} C_i = S$$

For hard clustering, given N points to k clusters, the number of all possible clustering is (Liu, 1968):

$$NW(N,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^i \binom{k}{i}(k-i)^N \qquad (1)$$

For example there are N (25, 5) = 2,436,684,974,110,751 ways of sorting 25 objects into five groups. If the number of clusters is unknown the objects can be sorted:

$$\sum_{i=1}^{n} N(n,k)$$

ways. For our 25 objects this is over 4×10^8 clustering. It is impractical for an algorithm to exhaustively search the solution space to find the optimal solution. Traditional clustering algorithms search a relatively small subset of the solution space. The probability of success of these methods is small. Clearly, we need an algorithm with the potential to search large solution spaces effectively. Genetic algorithms have been widely employed for optimization problems in several domains. Their success lies in their ability to span a large subset of the search space.

MULTIPLE RANDOMIZED KD TREES BASED NEAREST NEIGHBOR SEARCH

The most widely used algorithm for nearest-neighbor search is the *k*-d tree (Freidman *et al*., 1977). *k*-d tree is a space-partitioning data structure for organizing points in a *k*-dimensional space which iteratively bisects the search space into two regions containing half the points of the parent region. Queries are performed via traversal of the tree from the root to a leaf by evaluating the query point at each split. *k*-d tree does not work well for exact nearest neighbor search in high dimensional data because with high dimensional data, a *k*-d tree usually takes a lot of time to backtrack through the tree to find the optimal solution. The approximate nearest-neighbor approach can be used for improving the search speed over linear search. Proposed evolutionary algorithm nearest neighbor search is based on multiple randomized *k*-d trees (Silpa-Anan and Hartley, 2008). The *k*-d tree's search performance have increased by using Principal Component Analysis that align the principal axes of the data with the coordinate axes.

A fast evolutionary algorithm for automatic evolution of clusters: In this section, the evolutionary clustering algorithm for automatic clustering is proposed. This algorithm uses a variable-length encoded chromosome to automatically clustering the data set. This evolutionary clustering approach is called Fast Automatic Clustering Evolution (FACE). The steps for proposed algorithm is shown in Fig. 2. Its different steps are discussed in the following.

Chromosome representation: Each individual in the population is represented by a chromosome of length n, where n is the number of objects in the data set. For instance, if the data set having 6 (x_1, x_2, x_3, x_4, x_5, x_6) objects is to be clustered according to number of clusters value k = 3. The objects will align from 1 to 6 and each of them will get a cluster number k = {1,2,3}. One example of chromosome representation is shown in Fig. 1.

The use of simple encoding scheme causes two problems. First problem occurs if one clustering solution can be coded by several different chromosomes. Second problem, the clustering invalidity occurs if the recombination operator reproduces new clustering solutions, whose number of clusters is smaller than the given number of clusters. These problems has solved by using the chromosome reorganization method (Singh *et al*., 2011).

Population initialization: For individual i, the number of clusters, denoted by K_i is randomly generated in the range [K_{min}, K_{max}].Here K_{min} is chosen as 2 and K_{max} is chosen to be $\sqrt{N}$ (Lin *et al*., 2005). Where, N denotes number of objects. Each data points are assigned to the cluster with closed cluster center.

Fitness function: The fitness f of an individual is computed using the inverse of SSE (sum of squared error):

$$f = \frac{1}{SSE} \tag{2}$$

The SSE is defined as:

$$SSE = \sum_{C_i}\sum_{X\in Ci}\left(x - c_j\right)^T\left(x - c_j\right) = \sum_{Ci}\sum_{x\in Ci}\left\|x - c_j\right\|^2, j=1, 2,..., K \tag{3}$$

where $x\in C_i$ a data point assigned to that cluster C_i and cj is the center of the jth cluster.

Gene	X_1	X_2	X_3	X_4	X_5	X_6
Allele	1	2	2	3	3	1

Fig. 1: Chromosome representation

```
begin
Gemerate the initial population P randomly/* Popsize = |P|*/
for i = 1 to propsize call CR_method for P (I)
Evaluate the fitness for each chromosome P(i)
Initialize the generator counter t = 1
While (the termination criterion is not satisfied) do
Randomly generate k = rndint $\lfloor 2, \sqrt{N} \rfloor$
Adopt the k-means clustering to create K centroids for each chromosome
Select chromosomes from the current population for crossover operation with probability $p_c$
Apply mutation operator to the selected chromosome with mutation probability $p_m$
Evaluate fitness of new chromosomes
t = t+1
end while
end for
end
```

Fig. 2: Basic steps of FACE algorithm

Evolutionary operators

Selection: In proposed clustering algorithm (FACE), chromosomes are selected by using roulette wheel selection method (Goldberg, 1989). Here, each slot on the roulette wheel represents an individual. The area of the slot is directly proportional to the objective function value (fitness) of the individual. Each individual is selected by a spin of the roulette wheel. The probability of a chromosome being selected in any trial is proportional to its scaled fitness value within its subpopulation.

Crossover: Crossover operator combines the features of two parent chromosomes for generating two child chromosomes. The two problems, clustering invalidity and context insensitivity exists in the classical crossover operator. The FACE algorithm combines clustering solutions of different chromosomes. Randomly two chromosomes are selected form population. FACE randomly chooses j∈ {1, 2, ..., k} clusters from first chromosome and copied them in corresponding genes of second chromosome. Allele values of remaining genes are allocated to nearest clusters according to their centroids and relation between selected clusters of first chromosome and clusters of corresponding genes of second chromosome. In this way, FACE algorithm generates new offspring1 (child chromosome 3). The same procedure is applied for generating new offspring 2 (child chromosome 4) but now considering that the changed clusters of second chromosome are copied into first chromosome. Crossover probability is selected adaptively as in Srinivas and Patnaik (1994).

Let f_{max} be the maximum fitness value of the current population, $\bar{f}$ be the average fitness value of the population and f' be the larger of the fitness values of the solutions to be crossed. Then the probability of crossover, p_c, is calculated as:

$$p_c = k_1 \times \frac{(f_{max} - f')}{(f_{max} - \bar{f})} \text{if } f' > \bar{f} \tag{4}$$

$$p_c = k_3, , \text{if } f' > \bar{f}$$

Here, the values of k_1 and k_3 are equal to 1.0 (Srinivas and Patnaik, 1994). Clearly, when $f_{max} = \bar{f}$, then $f' = f_{max}$ and p_c will be equal to k_3.The value of p_c increases when the chromosome is quite poor. In contrast if p_c is low it means chromosome is good.

Mutation: Mutation is applied on each chromosome with fixed probability. During the mutation gene value a_i is replaced with a_i', with respect to nearest neighbor clusters (according to centroids), for I = 1,…,N. a_i, is a cluster number randomly selected from {1,2,…., k}, with the probability p_j.

$$p_j = \frac{e^{-d(x_i - c_k)}}{\sum_{j=1}^{k} e^{-d(x_i - c_j)}} \tag{5}$$

where, j∈[1, ..., k] and d (x_i, c_k) is Euclidean distance between object x_i and the center of the cluster k. The mutation probability p_m is selected adaptively for each chromosome as in Srinivas and Patnaik (1994). The expression is given below:

$$p_m = k_2 \times \frac{(f_{max} - f)}{(f_{max} - \bar{f})} \text{ if } f > \bar{f}, \text{ pm} = \text{k4 if } f \le \bar{f} \tag{6}$$

where, k_2 and k_4 are equal to 0.5. When f_{max}-$\bar{f}$ value decreases then p_c and p_m both will be increased. As a result GA will come out of local optimum. The default mutation rate (of 0.01) is kept for every solution in the FACE algorithm to overcome from problem of getting stuck at a local optimum.

Termination criterion: The FACE algorithm has been executed for a fixed number of generations. The fixed number is supplied by the user for terminating the algorithm. After termination, the algorithm gives the best string of the last generation that provides the solution to the clustering problem.

EXPERIMENTAL RESULTS

To conduct experiments on the FACE algorithm we have considered various types of the data sets. For this task, several artificial data sets and real data sets that are used to measure the performance of proposed algorithm. One performance measures Adjusted Rand index is used for this purpose. All experiments were run on a PC with a 2.0 GHz processor and 2 GB ram. We compared the performance of FACE algorithm with other algorithms in this area.

Parameter settings: The parameter settings used for FACE in our experimental study are given in Table 1.

Table 1: Parameter setting for FACE algorithm

Parameter	Setting
Number of generations	1000
Population size	50
No. of clusters	K_{min} to K_{max} [2 to 20]
Crossover	0.8
Mutation	0.001

Apart from the maximum number of clusters; these parameters are kept constant over the entire range of data sets in our comparison study.

Evaluation of clustering quality: Clustering quality is evaluated using an external measure of clustering quality which can be considered an objective evaluation. Especially we choose the Adjusted Rand Index (Rand, 1971) which is a generalization of the Rand Index. The Rand indices take two partitioning as the input (one of which, in our case, is the known correct solution) and count the number of pair-wise co-assignments of data items between the two partitioning. The Adjusted Rand Index additionally introduces a statistically induced normalization in order to yield values close to 0 for random partitions. Using a representation based on contingency tables, the Adjusted Rand Index A (Hubert, 1985) is given as:

$$\text{AdjustedIndex} = \frac{\text{Index} - \text{ExpectedIndex}}{\text{MaxIndex} - \text{ExpectedIndex}}$$

$$R(U,V) = \frac{\sum_k \binom{n_{lk}}{2} - [\sum_l \binom{n_l}{2} \cdot \sum_k \binom{n_{\cdot k}}{2}] / \binom{n}{2}}{\frac{1}{2} [\sum_l \binom{n_l}{2} + \sum_k \binom{n_{\cdot k}}{2}] - [\sum_l \binom{n_l}{2} \cdot \sum_k \binom{n_{\cdot k}}{2}] / \binom{n}{2}} \quad (7)$$

where, n_{lk} denotes the number of data items that have been assigned to both cluster l and cluster k. The Adjusted Rand Index return values in the interval (0, 1) and is to be maximized.

Artificial data sets: We applied the proposed algorithm (FACE) to several artificial data sets. The dimensionalities of these data sets are varying between two to fifty. The details features of these data sets are given below:

- **Data set 1:** This data set contains 200 points distributed on two crossed ellipsoidal shells
- **Data set 2:** The Swiss roll dataset contains 1600 data points. This data set to be clustered in 4 clusters
- **Data set 3:** This data set is combination of ring shaped, compact and linear clusters. The total number of points in it is 300. The dimension of this data set is two
- **Data set 4:** This data set is consists of 250 data points distributed over five spherically shaped clusters. The K means and proposed algorithm performs well this data set
- **Data set 5:** This is a 10 dimensional data set and it is consists of three classes of 838 data points. This dataset is to be clustered into 4 clusters
- **Data set 6:** This is a 10 dimensional data set consists of eight clusters of 3050 data points. This dataset is to be clustered into 10 clusters
- **Data set 7:** This is a 50 dimensional data set consists of 351 data points. This dataset is to be clustered into 4 clusters
- **Data set 8:** This is a 50 dimensional data set consists of 2328 data points. This dataset is to be clustered into 10 clusters

Real data sets: The proposed algorithm was tested for clustering with real data sets in multi-dimensional feature space obtained from UCI Machine Learning Repository. The performance of the FACE algorithm, CGA algorithm and *k*-means algorithm are compared through the experiments on the following three real data sets.

Wisconsin breast cancer: It contains two classes namely, malignant and benign with 683 unique data and 9 attributes corresponding to clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitoses.

Wine: This data set contains 178 samples of a chemical analysis of wines grown in the same region of Italy but brewed by three different brewers. The analysis has determined the quantities of 13 constituents found in each type of wines.

Diabetes: This is the diabetes data set consisting of 768 instances having 8 attributes.

The final clustering results obtained by *k*-mean, GCA and FACE are given in Fig. 3a-c, of data set1, respectively. We find that the *k*-means and GCA algorithm cannot work well for this data set but FACE algorithm accurately clustered the points of above data set. Figure 4 (a-c) show the results for *k*-means, GCA and FACE, respectively of data set 2. The clustering result of data set3 is show in Fig. 5a-c for *k*-means, GCA and FACE, respectively. The *k*-means is fail to found correct clusters, the proposed algorithm works well for this data set. Fig 5 (b) shows that GCA also can't get good result for this data set. The performance results reported in Table 2, clearly demonstrate the clustering accuracy of *k*-means, GCA and FACE for real data sets. The high value of Adjusted Rand Index indicates the good quality of clustering result. Table 3 indicates the quality of best

Table 2: Adjusted Rand Index values obtained by *k*-means, GCA and FACE for real data sets

Data sets	N	D	K	K means	GCA	FACE
Wine	178	13	3	0.9265	0.9415	0.9572
Cancer	683	09	2	0.9328	0.9360	0.9522
Diabetes	768	08	2	0.9450	0.9550	0.9860

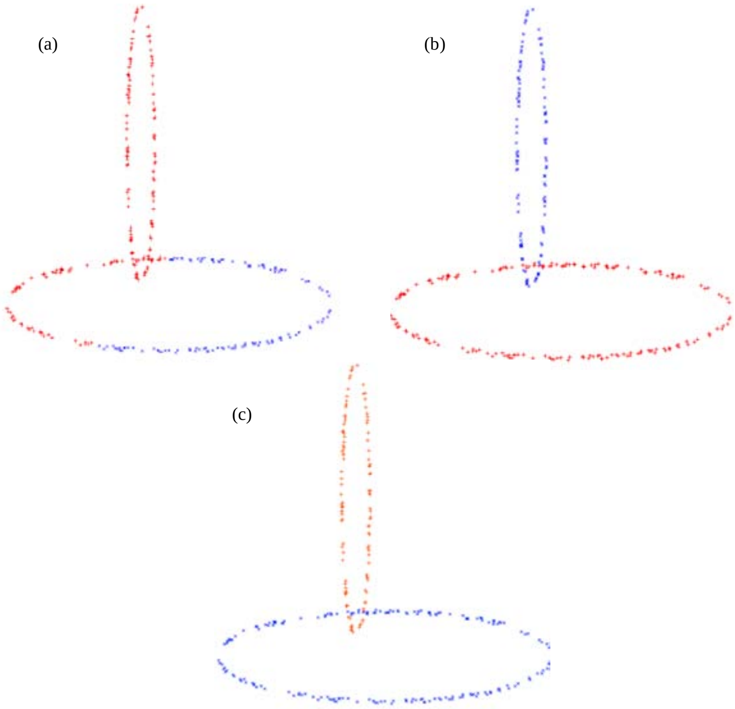

Fig. 3(a-c): Clustering of deta set 1 by (a) K-means, (b) GCA and (c) FACE

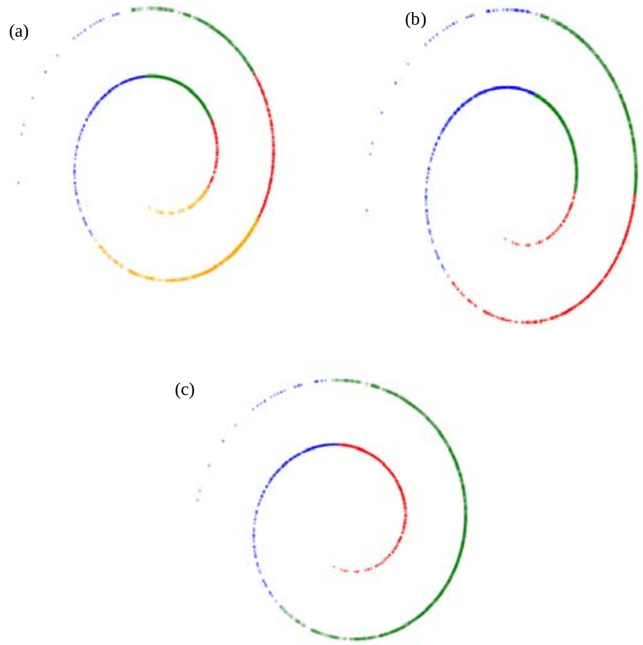

Fig. 4(a-c): Clustering of data set 2 by (a) K-means, (b) GCA and (c) FACE

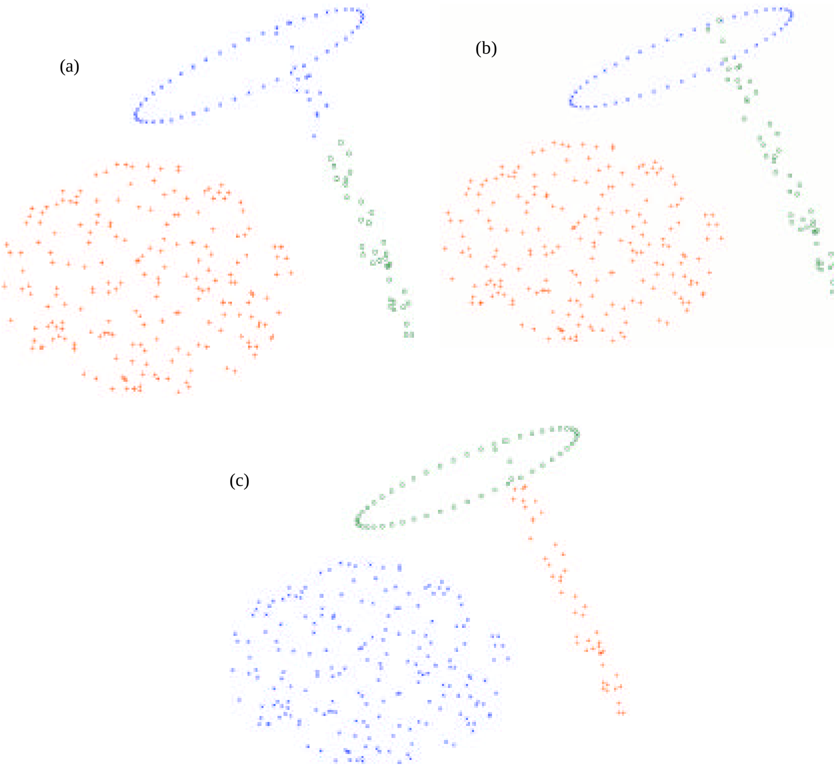

Fig. 5(a-c): Clustering of data set 3 by (a) *k*-means, (b) GCA and (c) FACE

Table 3: Adjusted Rand Index values obtained by *k*-means, GCA and FACE for eight synthetic data sets

Data set	N	D	K	K means	GCA	FACE
Data set 1	300	2	3	0.5680	0.5850	0.9850
Data set 2	1600	3	4	0.1575	0.6850	0.9580
Data set 3	400	2	3	0.8870	0.9075	0.9505
Data set 4	250	2	5	0.8560	0.8950	0.9760
Data set 5	838	10	4	0.9703	0.9780	0.9850
Data set 6	3050	10	10	0.9250	0.9350	0.9560
Data set 7	351	50	4	0.5616	0.8500	0.9680
Data set 8	2328	50	10	0.5115	0.8680	0.9400

clustering results in term of Adjusted Rand Index generated by *k*-means, GCA and FACE for eight artificial data sets. In Table 3, this is reflected in low values of the adjusted Rand Index as dimensionality and number of cluster increased.

CONCLUSION

In this study we have presented an evolutionary algorithm which can automatically determine the number of clusters and the proper partition from a given data set. Adaptive probabilities of crossover and mutation are adapted to prevent the FACE algorithm from getting stuck at a local optimal solution. In mutation operation, the gene cluster value is randomly selected with a probability based on Euclidean distance. To improve efficiency for nearest neighbor search this algorithm used multiple randomized *k*-d trees. This method create m different *k*-d trees each with a different structure in a such way that searches in the different trees will be independent. We use m = 5 for creating *k*-d trees. By using multiple *k*-d trees our algorithm is able to detect clusters in high dimensional data sets. The effectiveness of the FACE algorithm has been experimentally tested for eight artificial and three real data sets and the results are compared with those obtained by another clustering algorithm *k*-means and GCA. All experiments have demonstrated that FACE algorithm generally outperforms the other clustering algorithm *k*-means and GCA in efficiency and solution quality. Adjusted Rand Index is optimized as cluster validity measure. The FACE algorithm is based only on one objective function. So much further research needs to be carried out to use multi objective fitness functions with evolutionary algorithm.

ACKNOWLEDGMENT

The authors gratefully acknowledge Management, Faculty of Engineering and Technology, MITS, Rajasthan, India for his kind support in this research work.

REFERENCES

Abghari, H., M. Mahdavi, A. Fakherifard and A. Salajegheh, 2009. Cluster analysis of rainfall-runoff training patterns to flow modeling using hybrid RBF networks. Asian J. Applied Sci., 2: 150-159.

Agrawal, R., J. Gehrke, D. Gunopulos and P. Raghavan, 1998. Automatic subspace clustering of high dimensional data for data mining applications. ACM SIGMOD Rec., 27: 94-105.

Arora, A., S. Upadhyaya and R. Jain, 2009. Integrated approach of reduct and clustering for mining patterns from clusters. Inform. Technol. J., 8: 173-180.

Bandyopadhyay, S. and S. Saha, 2007. GAPS: A clustering method using a new point symmetry-based distance measure Technique. Pattern Recogn., 40: 3430-3451.

Bandyopadhyay, S. and U. Maulik, 2002. An evolutionary technique based on *k*-means algorithm for optimal clustering in RN. Inform. Sci., 146: 221-237.

Chu, Y.H., Y.J. Chen, D.H. Yang and M.S. Chen, 2009. Reducing redundancy in subspace clustering. IEEE Trans. Knowledge Data Eng., 21: 1432-1446.

Chu, Y.H., J.W. Huang, K.T. Chuang, D.N. Yang and M.S. Chen, 2010. Density conscious subspace clustering for high-dimensional data. IEEE Trans. Knowledge Data Eng., 22: 16-30.

Deng, Z., K.S. Choi, F.L. Chung and S. Wang, 2010. Enhanced soft subspace clustering integrating with in-cluster and between-cluster information. Pattern Recognit., 43: 767-781.

Ester, M., H.P. Kriegel, J. Sander and X. Xu, 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (ICKDDM'96), Portland, pp: 226-231.

Everitt, B., S. Landau and M. Leese, 2001. Cluster Analysis. Arnold, London.

Freidman, J.H., J.L. Bentley and R.A. Finkel, 1977. An algorithm for finding best matches in logarithmic expected time. ACM Trans. Math. Software, 3: 209-226.

Friedman, J.H. and J.J. Meulman, 2004. Clustering objects on subsets of attributes. J. R. Stat. Soc. Ser. B, 66: 815-849.

Garg, S. and R.C. Jain, 2006. Variations of *k*-mean algorithm: A study for high-dimensional large data sets. Inform. Technol. J., 5: 1132-1135.

Gautam, G. and B.B. Chaudhuri, 2004. A novel genetic algorithm for automatic clustering. Patt. Recogn. Lett., 25: 173-187.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Pub. Co, MA.

Guha, S., R. Rastogi and K. Shim, 1999. ROCK: A robust clustering algorithm for categorical attributes. Proceedings of the 15th International Conference on Data Engineering, March 23-26, 1999, Sydney, Australia, pp: 512-521.

Han and M. Kamber, 2004. Data Mining: Concepts and Techniques. Morgan Kaufmann USA.

Hemalatha, M. and K. Vivekanandan, 2008. A semaphore based multiprocessing *k*-mean algorithm for massive biological data. Asian J. Sci. Res., 1: 444-450.

Hinneburg, A. and D.A. Keim, 1998. An efficient approach to clustering in large multimedia databases with noise. Proceedings of 4rth International Conference on Knowledge Discovery and Data Mining, Aug. 27-31, New York, pp: 58-65.

Hubert, A., 1985. Comparing partitions. J. Classification, 2: 193-198.

Ioannis, A., A. Sarafis, P.W. Trinder and A.M.S. Zalzala, 2005. NOCEA: A rule-based evolutionary algorithm for efficient and effective clustering of massive high-dimensional databases. Appl. Soft Comput., 7: 668-710.

Jain, A.K. and R.C. Dubes, 1998. Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs, New Jersey, ISBN: 013022278X.

Jain, A.K., M.N. Murty and P.J. Flynn, 1999. Data clustering: A review. ACM Comput. Surveys, 31: 264-323.

Karypis, G., E.H. Han and V. Kumar, 1999. Chameleon: Hierarchical clustering using dynamic modeling. IEEE Comput., 32: 68-75.

Lai, C.C. and C.Y. Chang, 2009. A hierarchical evolutionary algorithm for automatic medical image segmentation. Exp. Syst. Appl., 36: 248-259.

Lin, H.J., F.W. Yang and Y.T. Kao, 2005. An efficient GA-based clustering technique, tamkang. J. Sci. Eng., 8: 113-122.

Liu, G.L., 1968. Introduction to Combinatorial Mathematics. Mc Graw Hill, New York.

Mclachlan, G.J. and T. Krishnan, 1997. The EM algorithm and extensions. John Wiley and Sons, Inc., New York.

Milenova, B.L. and M.M. Campos, 2002. O-Cluster: Scalable clustering of large high dimensional data sets. Proceedings of the IEEE International Conference on Data Mining, December 2002, Burlington, MA USA., pp: 290-297.

Mitra, S., 2004. An evolutionary rough portative clustering. Pattern Recognition Lett., 25: 1439-1449.

Murthy, C.A. and N. Chowdhury, 1996. In search of optimal clusters using genetic algorithms. Pattern Recog. lett., 17: 825-832.

Nagesh, H., S. Goil and A. Choudhary, 2001. Adaptive grids for clustering massive data sets. Proceedings of the 1st SIAM International Conference on Data Mining (SDM`01), New York, USA., pp: 1-17.

Nguyen, C.D. and K.J. Cios, 2008. GAKREM: A novel hybrid clustering algorithm. Inf. Sci., 178: 4205-4227.

Ouadfel, S. and M. Batouche, 2007. AntClust: An ant algorithm for swarm-based image clustering. Inform. Technol. J., 6: 196-201.

Pilevar, A.H. and M. Sukumar, 2005. GCHL: A grid-clustering algorithm for high-dimensional very large spatial data bases. Pattern Recogn. Lett., 26: 999-1010.

Rand, W., 1971. Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc., 66: 846-850.

Ranjan, J. and S. Khalil, 2007. Clustering methods for statistical analysis of genome databases. Inform. Technol. J., 6: 1217-1223.

Silpa-Anan, C. and R. Hartley, 2008. Optimised KD-trees for fast image descriptor matching. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2008, Anchorage, AK USA., pp: 1-8.

Singh, V., L. Sahoo and A. Kelkar, 2010. Mining subspace clusters in high dimensional data. Int. J. Recent Trends Eng. Technol., 3: 118-122.

Singh, V., L. Sahoo and A. Kelkar, 2011. Mining clusters in data sets of data mining: An effective algorithm. Int. J. Comput. Ther. Eng., 3: 171-177.

Srinivas, M. and L.M. Patnaik, 1994. Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Trans. Syst. Man Cybernet., 24: 656-667.

Sun, J.G., J. Liu and L.Y. Zhao, 2008. Clustering algorithms research. J. Software, 19: 48-61.

Velmurugan, T. and T. Santhanam, 2011. A survey of partition based clustering algorithms in data mining: An experimental approach. Inform. Technol. J., 10: 478-484.

Vijendra, S., L. Sahoo and K. Ashwini, 2010. An effective clustering algorithm for data mining. Proceedings of the International Conference on Data Storage and Data Engineering, February 9-10, 2010, Bangalore, India, pp: 250-253.

Vijendra, S., 2011. Efficient clustering for high dimensional data: Subspace based clustering and density based clustering. Inform. Technol. J., 10: 1092-1105.

Wang, W., J. Yang and R. Muntz, 1997. STING: A statistical information grid approach to spatial data mining. Proceedings of the International Conference on Very Large Data Bases, Aug. 25-29, Athens, Greece, pp: 86-195.

Yousria, N.A., M.S. Kamel and M.A. Ismail, 2009. A distance-relatedness dynamic model for clustering high dimensional data of arbitrary shapes and densities. Pattern Recognit., 42: 1193-1209.

Zarandi, M.H.F. and A. Alaeddini, 2010. A general fuzzy-statistical clustering approach for estimating the time of change in variable sampling control charts. Inform. Sci. Int. J., 180: 3033-3044.

Zhou, H., B. Feng, L. Lv and Y. Hui, 2007. A robust algorithm for subspace clustering of high-dimensional data*. Inform. Technol. J., 6: 255-258.