RESEARCH ARTICLE OPEN ACCESS

Image Clustering Using Evolutionary Computation Techniques

I. Ravi Kumar[1], V. Durga Prasad Rao[2]
M-Tech Research Scholar[1], Associate Professor[2]
Department of Computer Science and Engineering
Kaushik College of Engineering
Gambheeram, Vishakhapatnam
Andhra Pradesh, India

ABSTRACT

In the cluster analysis most of the existing clustering techniques for clustering, accept the numbers of clusters K as an input and determine that many number of cluster for a given data set. The projecting technique will try to discover true number of cluster centers automatically on the run. It will not only determines the true number of the cluster centers but also extracts real cluster centers and make a good classification. The goal of feature selection for unsupervised learning is to find the smallest feature subset that best uncovers "interesting natural" groupings (clusters) from data according to the chosen criterion. There may exist multiple superfluous feature subset solutions. We are satisfied in finding any one of these solutions. Unlike supervised learning, which has class labels to guide the feature search, in clustering (unsupervised learning) we need to define what "interesting" and "natural" mean. These are usually represented in the form of criterion functions.

Keywords:- Differential evolution (DE), Evolutionary Computational Techniques (ECT), K-Means Algorithm (KA), particle swarm optimization (PSO), Partitional Clustering (PC)

I. INTRODUCTION

CLUSTERING is the act of partitioning an unlabeled data set into groups of similar objects. Each group is called a "cluster", which consists of objects those are same among themselves and disparate from objects of other groups. In the past few decades, cluster analysis has played a central role in a variety of fields, ranging from engineering to social science and economics. Although an through list is impracticable it is worthwhile to mention that clustering has found applications in machine knowledge, artificial intelligence, pattern recognition, mechanical engineering and electrical engineering , web mining, spatial database exploration, textual document collection and image segmentation, genetics, biology, microbiology, paleontology, psychiatry and pathology, geography, geology and remote sensing, sociology, psychology, archeology, education, advertising and business[1]-[8]. In the cluster analysis most of the existing clustering techniques accept the number of clusters K, as an input instead of determining the same on the run. Also, if the data set is described by high-dimensional feature vectors, it may be virtually impossible to visualize the data for tracking its number of clusters. Chiefly in image pixel clustering knowing cluster number beforehand is a challenging task. A recent paper [9] has presented a new Differential Evolution (DE) based policy called ACDE (Automatic Clustering Using an Improved Differential Evolution) which is an evolutionary working out algorithm for crisp clustering of real–world data sets. The important feature of this technique is that it is able to robotically find the optimal number of clusters (i.e. the number of clusters does not have to be known in advance) even for very prominent dimensional data sets, where tracking of the number of clusters may be difficult.

There are various evolutionary computation techniques like genetic algorithm, Particle swarm Optimization techniques, Evolutionary Strategy etc can be very well implemented to address the problem of automatic clustering. In our proposed work we have envision to realize few of these techniques and develop some interesting hybridization of these approaches for effective image pixel clustering.

II. BRIEF REVIEW OF EXISTING WORK

Data clustering algorithms can be hierarchical or partitioned [10], [11]. Within each of the types, there exist a large number of subtypes and different algorithms for finding the clusters. In hierarchical clustering, the output is a tree showing a sequence of clustering, with each cluster being a partition of the data set [11]. Hierarchical algorithms can be agglomerative (bottom-up) or divisive (top-down). Agglomerative algorithms begin with each element as a separate cluster and merge them in successively larger clusters. Disruptive algorithms begin with the whole set and proceed to divide it into one after another smaller clusters. Hierarchical algorithms have two basic advantages [10]. First, the number of classes need not be specified a priori, and second, they are sovereign of the initial conditions. However the main drawback of hierarchical clustering techniques is that they are static; that is, data points assigned to a cluster cannot move to another cluster. In addition to that, they fail to

separate overlapping cluster due to lack of information about the global shape or size of clusters [12]. On the other hand partitional clustering algorithms challenge to crumble the data set directly into a set of disjoint clusters. They try to optimize certain criteria e.g., square-error function. The criterion function may accentuate the local structure of the data, such as by assigning clusters to peaks in the probability density function, or the global structure. Typically, the global criteria involve minimizing some measure of dissimilarity in the samples within each cluster while maximizing the dissimilarity of different clusters. The advantages of hierarchical algorithms are the disadvantages of the partition algorithms and vice versa.

Clustering can also be performed in two different modes: 1) crisp and 2) fuzzy. In crisp clustering, the clusters are displace and non-overlapping in nature. Any pattern may
belong to one and only one class in this case. In fuzzy clustering, a pattern may belong to all the classes with a certain fuzzy association grade [12].

The problem of partition clustering has been approached from assorted fields of knowledge, such as statistics (multivariate analysis)[13], graph theory [14], expectation-maximization algorithms [15], artificial neural networks [16]-[18], evolutionary computing [19],[20], and so on. Researchers all over the world are coming up with new algorithms, on a regular basis, to meet the increasing complexity of vast real-world data sets. In the evolutionary approach, clustering of a data set is viewed as an optimization problem and solved by using an evolutionary search heuristic such as genetic algorithm [21], which is inspired by Darwinian evolution and genetics. The key idea is to create a population of candidate solutions to an optimization problem, which is iteratively refined by variation and selection of good solutions for the next iteration. Candidate solutions are selected according to a fitness function, which evaluates their quality with respect to the optimization problem. In the case of genetic algorithms, the adjustment consists of mutation to explore solutions in the local neighborhood of existing solutions and crossover to recombine information between different candidate solutions. An important advantage of these algorithms is their ability to cope with local optima by maintaining, recombining and comparing several candidate solutions at the same time. In contrast, local search heuristics, such as the simulated annealing algorithm [22-23], only refine a single candidate solution and are notoriously weak in coping with local optima. Deterministic local search, which is used in algorithms like the K-means always converges to the nearest local optimum from the starting position of the search.

Enormous research effort has gone in the past few years to evolve the clusters in complex data sets through evolutionary computing techniques. However, not much research work has been reported to determine the optimal number of clusters at the same time. Most of the existing clustering techniques, based on evolutionary algorithm, accept the number of classes K as an input instead of determining the same on the run. Nevertheless, in many practical situations, the appropriate number of groups in previously unhandled data set may be unknown or impossible to determine even approximately. For example, while clustering a set of documents arising from query to a search engine, the number of class K changes for each set of documents that result from an communication with the search engine. Also, if the data set is described by high – dimensional features vectors, it may be practically impossible to visualize the data for tracking its numbers of clusters.

III. OBJECTIVES OF THE PROPOSED SYSTEM

The following objectives are to be worked out in the proposed research work

1) **Automatic determination of the optimal number of clusters in any unlabeled data set.**

In the cluster analysis most of the existing clustering techniques for clustering, accept the numbers of clusters K as an input and resolve that many number of cluster for a given data set. The proposed technique will try to settle on true number of cluster centers automatically on the run. It will not only determines the true number of the cluster centers but also extracts real cluster centers and make a good classification.

2) **Automatic research of the clusters with the choice of the most relevant features.**

The goal of feature selection for unsupervised learning is to find the smallest feature subset that best uncovers "interesting natural" groupings (clusters) from data according to the chosen criterion. There may exist multiple redundant feature subset solutions. We are satisfied in pronouncement any one of these solutions. Unlike supervised learning, which has class labels to guide the feature search, in clustering (unsupervised learning) we need to define what "interesting" and "natural" mean. These are usually represented in the form of decisive factor functions.

Data sets to be used for testing

The following real-life data sets will be used in this research work. Here, n is the number of data points, d is the number of features, and K is the number of clusters.

1**) Iris plants database** (n = 150, d = 4, K = 3): This is a well-known database with 4 inputs, 3 classes, and 150 data vectors. The data set consists of three different species of iris flower: *Iris setosa*, *Iris virginica*, and *Iris versicolour*. For each species, 50 samples with four features each (sepal length, sepal width, petal length, and petal width) were collected. The number of objects that belong to each cluster is 50.

2) **Glass** (n = 214, d = 9, K = 6): The data were sampled from six different types of glass: (i) building windows float processed (70 objects); (ii) building windows no float processed (76 objects); (iii) vehicle windows float processed (17 objects); (iv) containers (13 objects); (v) tableware (9 objects); and (vi) headlamps (29 objects). Each type has nine

features: (i) refractive index; (ii) sodium; (iii) magnesium; (iv) aluminum; (v) silicon; (vi) potassium; (vii) calcium; (viii) barium; and (ix) iron.

3) **Wisconsin breast cancer data set** (n = 683, d=9, K=2): The Wisconsin breast cancer database contains nine relevant features: (i) clump thickness; (ii) cell size uniformity; (iii) cell shape uniformity; (iv) marginal adhesion; (v) single epithelial cell size; (vi) bare nuclei; (vii) bland chromatin; (viii) normal nucleoli; and (ix) mitoses. The data set has two classes. The objective is to classify each data vector into benign (239 objects) or malignant tumors (444 objects).

4) **Wine** (n = 178, d = 13, K = 3): This is a classification problem with "well-behaved" class structures. There are 13 features, three classes, and 178 data vectors.

5) **Vowel data set** (n = 871, d = 3, K = 6): This data set consists of 871 Indian Telugu vowel sounds. The data set has three features, namely $F1$, $F2$, and $F3$, corresponding to the first, second and, third vowel frequencies, and six overlapping classes {d (72 objects), a (89 objects), i (172 objects), u (151 objects), e (207 objects), o (180 objects)}.

6) **Images** like Mandrill, Lena, Brain MRI, Cameraman etc

IV. EXPECTED RESULTS

The algorithm is expected to

(1) Automatically project the data to a low dimensional feature subspace,

(2) Determine the number of clusters, Find out the appropriate cluster centers with the most relevant features as a faster pace.

V. SCIENTIFIC BACKGROUND

A. Problem Definition

A *pattern* is a physical or abstract structure of objects. It is distinguished from others by a collective set of attributes called *features*, which together represent a pattern [27]. Let $P = \{P1, P2, \ldots, Pn\}$ be a set of n patterns or data points, each having d features. These patterns can also be represented by a profile data matrix $\mathbf{X}n\times d$ with n d-dimensional row vectors. The ith row vector $\_Xi$ characterizes the ith object from the set P, and each element Xi,j in $\_Xi$ corresponds to the$j$th real-value feature (j = 1, 2, . . . , d) of the ith pattern (i =1, 2, . . . , n). Given such an $\mathbf{X}n\times d$ matrix, a partitional clustering algorithm tries to find a partition $C = \{C1, C2, \ldots, CK\}$ of K classes, such that the similarity of the patterns in the same cluster is maximum and patterns from different clusters differ as far as possible. The partitions should maintain three properties.

1) Each cluster should have at least one pattern assigned, i.e., $Ci \_= \Phi \forall i \in \{1, 2, \ldots, K\}$.

2) Two different clusters should have no pattern in common, i.e., $Ci \cap Cj = \Phi \forall i \_= j$ and $i, j \in \{1, 2, \ldots, K\}$.

3) _Each pattern should definitely be attached to a cluster i.e., $U^K{}_{i=1} C_k = P$.

Since the given data set can be partitioned in a number of ways, maintaining all of the aforementioned properties, a fitness function (some measure of the adequacy of the partitioning) must be defined. The problem then turns out to be one of finding a partition $\mathbf{C}*$ of optimal or near-optimal adequacy, as compared to all other feasible solutions $\mathbf{C} = \{C^1, C^2, \ldots, C^{N(n,K)}\}$, where

$$N(n, K) = \frac{1}{K!}\sum_{i=1}^{K}(-1)^i \binom{K}{i}^i (K-i)^i \qquad (1)$$

is the number of feasible partitions. This is the same as

$$\text{Optimize}_{C} \; f(Xn\times d, C) \qquad (2)$$

where C is a single partition from the set $\mathbf{C}$, and f is a Statistical–mathematical function that quantifies the goodness of a partition on the basis of the distance measure of the patterns(please see Section II-C). It has been shown in [28] that the clustering problem is NP-hard when the number of clusters exceeds 3.

B. Similarity Measures

As previously mentioned, clustering is the process of recognizing natural groupings or clusters in multidimensional data based on some similarity measures. Hence, defining an appropriate similarity measure plays a fundamental role in clustering [11]. The most popular way to evaluate similarity between two patterns amounts to the use of a *distance measure*. The most widely used distance measure is the Euclidean distance, which between any two d-dimensional patterns $\vec{x}_i$ and $\vec{x}_j$ is given by

$$d\left(\vec{x}_i, \vec{x}_j\right) = \sqrt{\sum_{p=1}^{d}(X_{i,p} - X_{j,p})^2} = \left\|\vec{x}_i - \vec{x}_j\right\|. \qquad (3)$$

The Euclidean distance measure is a special case (when α = 2) of the Minowsky metric [11], which is defined as

$$d^{\alpha}\left(\vec{x}_i, \vec{x}_j\right) = \left(\sum_{p=1}^{d}(X_{i,p} - X_{j,p})^{\alpha}\right)^{1/\alpha} = \left\|\vec{x}_i - \vec{x}_j\right\|^{\alpha} \qquad (4)$$

When α = 1, the measure is known as the Manhattan distance [28]. The Minowsky metric is usually not efficient for clustering data of high dimensionality, as the distance between the patterns increases with the growth of dimensionality. Hence, the concepts of *near* and *far* become weaker [29]. Furthermore, according to Jain *et al.* [11], for the Minowsky metric, the large scale features tend to dominate over the other features. This can be solved by normalizing the features over a common range. One way to do the same is by using the cosine distance (or vector dot product), which is defined as

$$\left\langle \vec{x}_i, \vec{x}_j \right\rangle = \frac{\sum_{p=1}^{d}(X_{i,p} - X_{j,p})}{\left\|\vec{x}_i\right\|\left\|\vec{x}_j\right\|} \qquad (5)$$

The cosine distance measures the angular difference of the two data vectors (patterns) and not the difference of their magnitudes.

C. Clustering Validity Indexes

Cluster validity indexes correspond to the statistical–Mathematical functions used to evaluate the results of a clustering algorithm on a quantitative basis. Generally, a cluster validity index serves two purposes. First, it can be used to determine the number of clusters, and second, it finds out the corresponding best partition. One traditional approach for determining the optimum number of classes is to repeatedly run the algorithm with a different number of classes as input and then to select the partitioning of the data resulting in the best validity measure [30]. Ideally, a validity index should take care of the two aspects of partitioning.

1) ***Cohesion***: The patterns in one cluster should be as similar to each other as possible. The fitness variance of the patterns in a cluster is an indication of the cluster's cohesion or compactness.
2) ***Separation***: Clusters should be well separated. The distance among the cluster centers (may be their Euclidean distance) gives an indication of cluster separation.

For crisp clustering, some of the well-known indexes available in the literature are the Dunn's index (DI) [31], the Calinski–Harabasz index [32], the DB index [33], the Pakhira Bandyopadhyay Maulik (PBM) index [34], and the CS measure [35]. All these indexes are optimizing in nature, i.e., the maximum or minimum values of these indexes indicate the appropriate partitions. Because of their optimizing character, the cluster validity indexes are best used in association with any optimization algorithm such as GA, PSO, etc. In what follows, we will discuss only two validity measures in detail, which have been employed in the study of our automatic clustering algorithm. *1) DB Index:* This measure is a function of the ratio of the sum of within-cluster scatter to between-cluster separation, and it uses both the clusters and their sample means. First, we define the *within i*th *cluster scatter* and the *between* i^{th} and j^{th} *cluster distance*, respectively, i.e.,

$$S_{i,q}=\left[\frac{1}{N}\sum_{\vec{x}\in C_i}\left\|\vec{x}-\vec{m}_i\right\|_2^q\right]^{1/q} \quad (7)$$

$$d_{ij,t}=\left\{\sum_{p=1}^{d}\left|m_{i,p}-m_{j,p}\right|^t\right\}^{1/t}=\left\|\vec{m}_i-\vec{m}_j\right\|_t \quad (8)$$

where *_mi* is the *i*th cluster center, q, $t \geq 1$, q is an integer, and q and t can be independently selected. *Ni* is the number of elements in the *i*th cluster *Ci*. Next, *Ri,qt* is defined as

$$R_{i,qt}=\max_{j\in K, j\neq i}\left\{\frac{S_{i,q}+S_{j,q}}{d_{ij,t}}\right\} \quad (9)$$

Finally, we define the DB measure as

$$\text{DB}(K)=\frac{1}{K}\sum_{i=1}^{K}R_{i,qt} \quad (10)$$

The smallest DB(K) index indicates a valid optimal partition.

VI. CONCLUSION

Works on automatic clustering have been done since many years in which clustering of datasets, which was time consuming, burden-some and unorganized due to a large number of given specifications has improved significantly. In the present proposed investigation, it is expected that automatic clustering can take place with a limited number of most relevant features with ECTs which will further improve the data clustering technique. In addition, reducing the number of features increases comprehensibility and ameliorates the problem for high dimensional data for which some clustering algorithms break down.

REFERENCES

[1] I. E. Evangelou, D. G. Hadjimitsis, A. A. Lazakidou, and C. Clayton, "Data mining and knowledge discovery in complex image data using artificial neural networks," in Proc. Workshop Complex Reason. Geogr Data, Paphos, Cyprus, 2001.

[2] T. Lillesand and R. Keifer, Remote Sensing and Image Interpretation. Hoboken, NJ: Wiley, 1994.

[3] H. C. Andrews, Introduction to Mathematical Techniquesin Pattern Recognition. New York: Wiley, 1972.

[4] M. R. Rao, "Cluster analysis and mathematical programming,"J. Amer Stat. Assoc., vol. 66, no. 335, pp. 622–626, Sep.1971.

[5] R. O. Duda and P. E. Hart, PatternClassification andScene Analysis. Hoboken, NJ: Wiley, 1973.

[6] K. Fukunaga, Introduction to Statistical PatternRecognition.NewYork: Academic, 1990.

[7] B. S. Everitt, Cluster Analysis, 3rd ed. New York:Halsted,1993.

[8] J. A. Hartigan, Clustering Algorithms. New York: Wiley, 1975.

[9] H. Frigui and R. Krishnapuram, "A robust competitive clustering algorithm with applications in computervision," IEEE Trans. Pattern Anal. Mach. Intell., vol. 21,no. 5, pp.450–465, May 1999.

[10] Y. Leung, J. Zhang, and Z. Xu, "Clustering by scale-Space filtering," IEEE Trans. Pattern Anal. Mach. Intell., vol.22, no. 12, pp. 1396–1410, Dec. 2000.

[11] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: Areview," ACM Comput. Surv., vol. 31, no. 3, pp. 264–323,Sep. 1999.

[12] E. W. Forgy, "Cluster analysis of multivariate data: Efficiency versus interpretability of classification," Biometrics, vol. 21, no. 3, pp. 768–769, 1965.

[13] C. T. Zahn, "Graph-theoretical methods for detecting And describing gestalt clusters," IEEE Trans. Comput., vol. C- 20, no. 1, pp. 68–86, Jan. 1971.

[14] T. Mitchell, Machine Learning. New York: McGraw-Hill, 1997.

[15] J. Mao and A. K. Jain, "Artificial neural networks forfeature extraction and multivariate data projection,"

IEEE Trans. Neural Netw., vol. 6, no. 2, pp. 296–317, Mar. 1995.

[16] N. R. Pal, J. C. Bezdek, and E. C.-K. Tsao, "Generalizedclustering networks and Kohonen's self-organizing scheme," IEEE Trans. Neural Netw., vol. 4, no. 4, pp.549–557, Jul. 1993.

[17] T. Kohonen, Self-Organizing Maps, vol. 30. Berlin, Germany: Springer- Verlag, 1995.

[18] E. Falkenauer, Genetic Algorithms and Grouping Problems. Chichester U.K.: Wiley, 1998.236 IEEETRANSACTIONS ON SYSTEMS, MAN, ANDCYBERNETICS— PARTA:SYSTEMSANDHUMANS, VOL. 38,NO. 1, JANUARY 2008

[19] S. Paterlini and T. Minerva, "Evolutionary approaches Forcluster analysis," in Soft Computing Applications, Bonarini, F. Masulli, and G. Pasi, Eds. Berlin, Germany:Springer-Verlag, 2003, pp. 167–178.

[20] J. H. Holland, Adaptation in Natural and ArtificialSystems. Ann Arbor, MI: Univ. Michigan Press, 1975.

[21] S. Z. Selim and K. Alsultan, "A simulated annealingalgorithm for the clustering problem," Pattern Recognit., vol. 24, no. 10, pp. 1003–1008, 1991.

[22] J. MacQueen, "Some methods for classification andanalysis of multivariate observations," in Proc. 5th Berkeley Symp.Math. Stat. Probability, 1967, pp. 281–297.

[23] R. Storn and K. Price, "Differential evolution—A simple And efficient heuristic for global optimization over continuous spaces," J. Glob. Optim., vol. 11, no. 4, pp. 341–359, Dec.1997.

[24] S. Bandyopadhyay and U. Maulik, "Genetic clusteringfor automatic evolution of clusters and application to image classification," Pattern Recognit., vol. 35, no. 6,pp. 1197– 1208, Jun. 2002.

[25] M. Omran, A. Salman, and A. Engelbrecht, "Dynamic clustering using particle swarm optimization with application in unsupervised image classification," inProc5th World Enformatika Conf. (ICCI), Prague,CzechRepublic, 2005.