International Journal of Advanced Computer and Mathematical Sciences
ISSN 2230-9624. Vol 3, Issue 2, 2012, pp 257-265
http://bipublication.com

COMPARATIVE ANALYSIS OF K-MEANS AND GENETIC ALGORITHM BASED DATA CLUSTERING

***Rajashree Dash and Rasmita Dash**
School of Computer Science and Engineering, ITER, Siksha O Anusandhan University,Bhubaneswar, India
*Corresponding author: Email: rajashree_dash@yahoo.co.in

[Received-17/04/2012, Accepted-28/05/2012]

ABSTRACT:

Clustering is a useful unsupervised data mining task that subdivides an input data set into a desired number of subgroups so that members of the same subgroup will have high similarity and the members of different groups have large differences. K-means is a commonly used partitioning based clustering technique that tries to find a user specified number of clusters (k), which are represented by their centroids, by minimizing the square error function. Although K-means is simple and can be used for a wide variety of data types, it is quite sensitive to initial positions of cluster centers. There are two simple approaches to cluster center initialization i.e. either to select the initial values randomly, or to choose the first k samples of the data points. Both approaches cause the algorithm to converge to sub optimal solutions. On the other hand Genetic algorithm one of the commonly used evolutionary algorithms performs global search to find the solution to a clustering problem. The algorithm typically starts with a set of randomly generated individuals called the population and creates successive, new generations of the population by genetic operations such as natural selection, crossover, and mutation. Each chromosome of the population represents K no. of centroids. Steps of genetic algorithm are repeatedly applied for a no. of generations to search for appropriate cluster centers in the feature space such that a similarity metric of the resulting clusters is optimized. K-means and genetic algorithm based data clustering have been compared in this paper on the basis of their working principle, advantage and disadvantage with proper example.

Keywords*:* Data clustering, K-means, Genetic algorithm, Fitness function,SSD

[I] INTRODUCTION

The tremendous growth of scientific databases put a lot of challenges before the researches to extract useful information from them using traditional data base techniques. Hence effective mining methods are essential to discover the implicit information from huge databases.

Cluster analysis is one of the major data mining techniques, widely used for many practical applications in various emerging areas like Bioinformatics. Clustering is an unsupervised method that subdivides an input data set into a desired number of subgroups so that the objects of the same subgroup will be similar (or related) to one another and different from (or unrelated to) the objects in other groups [1]. A good clustering method will produce high quality clusters with high intra-cluster similarity and low inter-cluster similarity [10]. The quality of a clustering result depends on both the similarity measure used by the method and its implementation and also by its ability to discover some or all of the hidden patterns.

K-means is a commonly used partitioning based clustering technique that tries to find a user specified number of clusters (k), which are represented by their centroids, by minimizing the square error function. The clustering method aims at optimizing the cost function to minimize the dissimilarity of the objects within each cluster, while maximizing the dissimilarity of different clusters. Being an iterative and hill-climbing method, it is quite sensitive to initial positions of cluster centers.. Furthermore, since the associated cost functions are nonlinear and multimodal, usually these algorithms converge to a local minimum i.e. it produces different clusters for different sets of values of the initial centroids [2].

On the other hand Genetic Algorithm (GA) parallel search technique, that searches for a global approximate solution to the clustering problems through application of the principles of evolutionary biology [4]. The algorithm typically starts with a set of randomly chosen solutions called the population and creates successive, new generations of the population by genetic operations such as natural selection, crossover, and mutation [5]. Natural selection is performed based on the fitness of an individual. For an individual, the better its fitness, the more chances it has to survive in the next generation. Crossover is performed by certain crossover rule and mutation aims at changing an individual by a user-specified mutation probability. The intuition underlying the approach is that each new population will be better than the previous one. Traditionally, solutions are represented using fixed length strings, especially binary strings, but alternative encodings have been developed. The greatest advantage of genetic algorithms is that the fitness function can be altered to change the behavior of the algorithm.

The rest of the paper is organized as follows. The next section provides a detailed over view of K-means based data clustering with proper example. The section 3 contains the details of GA based data clustering with proper example. A comparative analysis has specified in section 4 followed by conclusion.

[II] K-MEANS CLUSTERING

The K-means algorithm is one of the partitioning based, nonhierarchical clustering methods. Given a set of numeric objects ***X*** and an integer number *k*, the K-means algorithm searches for a partition of ***X*** into *k* clusters that minimizes the within groups sum of squared errors i.e. SSD. The K-means algorithm starts by initializing the *k* cluster centers. The input data points are then allocated to one of the existing clusters according to the square of the Euclidean distance from the clusters, choosing the closest. The mean of each cluster is then computed so as to update the cluster center. This update occurs as a result of the change in the membership of each cluster. The processes of re-assigning the input vectors and the update of the cluster centers is repeated until no more change in the value of any of the cluster centers.

Although K-means is simple and can be used for a wide variety of data types, it is quite sensitive to initial positions of cluster centers. There are two simple approaches to cluster center initialization i.e. either to select the initial values randomly, or to choose the first k samples of the data points. As an alternative, different sets of initial values are chosen (out of the data points) and the set, which is closest to optimal, is chosen [6]. Again the k-means algorithm is computationally expensive and requires time proportional to the product of the number of data items, number of clusters and the number of iterations.

2.1. Steps of K-Means Clustering

The steps of the K-means algorithm are written below:

1. Initialization: Randomly *K* data points are chosen to initialize the cluster centers.

2. Nearest-neighbor search: Each data point is assigned to the cluster center that is closest to it. The distance from the data vector to the centroid is calculated using the following equation.

$$d(z_p, a_j) = \sqrt{\sum_{k=1}^{d} (z_{pk} - a_{jk})^2}$$

Where d is the dimension of the data vector, z_p is the centroid of cluster p and a_j is the data vector.

3. Mean update: New cluster centers are calculated finding the mean of the input vectors assigned to a particular cluster.

4. Stopping rule: repeat steps 2 and 3 until no more change in the value of the means.

2.2. Example of K-Means Clustering

A data set with 6 data objects and 2 variables has been taken for implementing the K-Means algorithm. Initially with k =2, two centroids (10, 10) and (8, 20) are chosen randomly. Then the Euclidean distances from each data object to the centroids are obtained. For each data object the closest centroid is selected and accordingly the objects are assigned to the clusters. Then the new centroid value of each cluster is updated to (12, 11.3) and (18, 22.6) taking the mean of the data objects assigned to each cluster. The process of comparing the distances from data object to the centroids, reassigning the objects to cluster and updating the centroid value has done for thee iterations because after that there is no changes in the new centroid value. The detail of implementation of k-means has shown in table 1. The sum of squared distance obtained with this clustering is 248.92. Again the k-means clustering has implemented on the same data object with k=2 and two randomly chosen initial centroids (10, 10) and (20, 12), whose details has shown in table2. By just changing the initial centroid value the final output of clustering has also changed and the sum of squared distance obtained i.e. 239.99 is also less than the SSD value of first clustering. Hence from the solved example, it can be clearly seen that output of k-means clustering is quite dependent on the initial position of centroid.

[Table-1].

Data Objects	**X Value**	**Y Value**	**Iteration 1**		**Iteration 2**		**Iteration 3**	
			Euclidean Distance from data object to Centroid 1 (10,10)	**Euclidean Distance from data object to Centroid 2 (8,20)**	**Euclidean Distance from data object to Centroid 1 (12,11.3)**	**Euclidean Distance from data object to Centroid 2 (18,22.6)**	**Euclidean Distance from data object to Centroid 1 (11,13.5)**	**Euclidean Distance from data object to Centroid 2 (23,24)**
1	10	10	**0**	10.198	**2.3854**	14.925	**3.6401**	19.105
2	20	12	**10.198**	14.422	**8.0306**	10.787	**9.1241**	12.3693
3	24	30	24.413	**18.868**	22.219	**9.5268**	21.006	**6.0828**
4	8	20	10.198	**0**	**9.5755**	10.3325	**7.159**	15.5242
5	6	12	**4.4721**	8.2462	**6.0407**	16.0112	**5.22**	20.8087
6	22	18	14.4222	**14.1421**	12.037	**6.0959**	11.8849	**6.0828**
Output of each iteration			**Cluster 1 : (1,2,5) Cluster 2 : (3,4, 6) New Cent 1: (12,11.3) New Cent 2:(18,22.6)**		**Cluster 1 : (1,2,4,5) Cluster 2 : (3, 6) New Cent 1: (11,13.5) New Cent 2:(23 24)**		**Cluster 1 : (1,2,4,5) Cluster 2 : (3, 6) New Cent 1: (11, 13.5) New Cent 2:(23,24) SSD = 248.92**	

Table: 1.O/p of K-means with k=2 and randomly chosen centroids (10, 10) and (8,20)

[Table-2].

Data Objects	**X value**	**Y value**	**Iteration 1**		**Iteration 2**	
			Euclidean Distance from data object to Centroid 1 (10,10)	**Euclidean Distance from data object to Centroid 2 (20,12)**	**Euclidean Distance from data object to Centroid 1 (8,14)**	**Euclidean Distance from data object to Centroid 2 (22,20)**
1	10	10	**0**	10.198	**4.4721**	15.6205
2	20	12	10.198	**0**	12.1655	**8.2462**
3	24	30	24.413	**18.439**	22.6274	**10.198**
4	8	20	**10.198**	14.422	**6**	14
5	6	12	**4.4721**	14	**2.8284**	17.885
6	22	18	14.4222	**6.3246**	14.5602	**2**
Output of each iteration			**Cluster 1 : (1,4,5)** **Cluster 2 : (2,3, 6)** **New Cent 1: (8,14)** **New Cent 2:(22,20)**		**Cluster 1 : (1,4,5)** **Cluster 2 : (2,3, 6)** **New Cent 1: (8,14)** **New Cent 2:(22,20)** **SSD: 239.99**	

Table: 2. O/p of K-means with k=2 and randomly chosen centroids (10, 10) and (20, 12)

2.2. Drawbacks of K-Means Clustering

- It is data dependent.
- It is based on greedy approach. So the final clusters may not represent a global optimization result but only the local one, and complete different final clusters can arise from difference in the initial randomly chosen cluster centers.
- Need to specify k, the number of clusters, in advance.
- Unable to handle noisy data and outliers.
- Not suitable to discover clusters with non-convex shapes or clusters of very different size.

[III] GA BASED CLUSTERING

Genetic algorithms are randomized search and optimization techniques guided by the principles of evolution and natural genetics, having a large amount of implicit parallelism [3] [7]. GAs perform search in complex, large and multimodal landscapes, and provide near-optimal solutions for objective or fitness function of an optimization problem. In GAs, the parameters of the search space are encoded in the form of strings (called chromosomes). A collection of such strings is called a population. Initially, a random population is created, which represents different points in the search space. An objective and fitness function is associated with each string that represents the degree of goodness of the string. Based on the principle of survival of the fittest, a few of the strings are selected and each is assigned a number of copies that go into the mating pool. Biologically inspired operators like *cross*-over and *mutation* are applied on these strings to yield a new generation of strings. The process of selection, crossover and mutation continues for a fixed number of generations or till a termination condition is satisfied [8].

The basic steps of genetic algorithm for data clustering include individual representation and population initialization, fitness computation, selection, crossover and mutation. Each individual represents one feature subspace. Its fitness represents the clustering result with respect to the feature space that the individual represents. Larger the fitness, denser the data in such feature subspace and better the clustering results [9]. The details are described below.

3.1. Individual Representation and Population Initialization

Individual representation or Encoding transforms one possible solution from solution space to search space which can be handled by GAs. The individuals are vectors of the solution space in the form of strings. One individual represents one possible solution to the problem. GAs can find the optimal solution or approximate optimal solution of the problem after applying a certain number of genetic operators on those individuals.

There are two commonly used encoding methods: binary encoding and floating point encoding. Comparing with floating point encoding, the searching space of binary encoding is larger, moreover, the crossover and mutation implemented on it is more convenient.

The initial population has set up at random. At first, the K cluster centers encoded in each chromosome are initialized to K different randomly chosen features from the original feature space. Then, this process is repeated for each chromosome in the population. For k clusters and data objects with n dimension length of each chromosome is equivalent to n*k.

3.2. 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. After the clustering is done, the cluster centers encoded in the chromosome are replaced by the mean points of the respective clusters. The fitness function is defined as $f'=1/\mu$, so that maximization of the fitttness function leads to minimization of μ, where

$$\mu = \sum_{i=1}^{k} \mu_i$$

$$\mu_i = \sum_{x_j \in C_i} \left\| x_j - z_i \right\|$$

3.3. Selection

Selection process is used to get the optimum solution by preferring individuals with high fitness. It is usually biased towards fitter chromosomes. Selection methods are used as mechanisms to focus the search on apparently more profitable regions in the search space. Various types of selection procedures are as follows:

Roulette Wheel Selection

Parent chromosomes are probabilistically selected based on their fitness. In this process each chromosome occupies a slot in the roulette wheel with slot size proportional to the chromosome’s fitness. When the wheel is randomly spun, the chromosome corresponding to the slot where the wheel stopped is selected as the first parent. This process is repeated for finding the other parents. Roulette wheel selection suffers from the problem that highly fit individuals may dominate in the selection process, which may cause diversity to decrease rapidly resulting in premature convergence.

Rank Selection

Rank selection sorts the chromosomes according to their fitness and the chromosomes are selected based on their rank, not on the absolute fitness value. With this process all chromosomes will have a good chance to be selected. However this approach may have a slower convergence rate than roulette wheel approach.

Tournament selection

A set of chromosomes are randomly chosen. The fittest chromosome from the set is then placed in a mating pool. This process is repeated until the mating pool contains a sufficient number of chromosomes to start the mating process.

Elitism

In this approach the fittest chromosome or a user specified number of best chromosomes is copied into the new population. The remaining chromosomes are then chosen using any selection operator. Since the best solution is never lost the performance of GA can significantly be improved.

3.4. Crossover

Crossover is a probabilistic process that exchanges information between two parent chromosomes for generating two child chromosomes. It greatly accelerates search early in evolution of a population. Crossover occurs with a user specified probability, called the crossover probability p_c. Various types of crossover operators are:

Single Point Crossover

In this approach a position is randomly selected at which the parents are divided into two parts.

The parts of the two parents are then swapped to generate two new offspring.

Parent		Offspring
A: 11001010	⟹	11001**011**
B: **01110011**		**01110**010

Two Point Crossover

In this approach two positions are randomly selected. The middle parts of the two parents are then swapped to generate two new offspring.

Parent		Offspring
A: 11001010		11**1100**10
B: **01110011**	⟹	**01**0010**11**

Uniform Crossover

In this approach alleles are copied from either the first parent or the second parent with some probability usually set to 0.5. A mask is specified. According to the bit in mask the new value is selected either from the parent A or Parent B.

Mask: 11001010

Parent		Offspring
A: 11001010		11**11**1**0**1**1**
B: **01110011**	⟹	**01**00**0**0**1**0

3.5. Mutation

Mutation process normally changes the structure of chromosomes, by negating a randomly chosen bit. It randomly modifies the gene values at selected locations. Mutation increases genetic diversity, by forcing the algorithm to search areas other than the current focus. For binary representation of chromosomes, a bit position is mutated by simply flipping its value. Otherwise a number δ in the range [0, 1] is generated with uniform distribution. If the value at a gene position is v, after mutation it becomes

$$v = v \pm 2 * \delta * v, \qquad v \neq 0$$
$$v = v \pm 2 * \delta, \qquad v = 0$$

3.6. Steps of Clustering using GA

Input:

k:the no of clusters

P: Population size

D:the data set containing n objects

tmax : Maximum no. of iterations

Output: A set of K clusters

```
1)   Initialize each chromosome to contain k randomly chosen centroids from the data set.
2)   For t=1 to tmax
       a) for each chromosome i
         i) Assign the data object to the cluster with the closest centroid.
         ii) Recalculate k cluster centroids of chromosome i as the mean of their data objects.
         iii) Calculate the fitness of chromosome i.
       b) Create the new generation of chromosomes using selection, crossover and mutation.
```

3.6. Example of Clustering using GA

The same data set with 6 data objects and 2 variables has been taken for implementing the GA based data clustering, with population size 4. Initially with k =2, 4 chromosomes are chosen randomly from the data set. With each individual chromosome, clustering is done and new centroid is obtained. Then the fitness value is calculated by adding the distances from each object to their corresponding centroids, which is represented in table 3. The new chromosomes obtained by applying the Rank selection, one point crossover at the middle and the mutation on the new centroids, have shown in the table 4. Now the new chromosomes are applied as input for the second generation and its fitness value is calculated. Then again the selection, crossover and mutation are applied to get a set of chromosomes for the third generation. These steps are repeatedly applied till a termination condition reached. In this example a maximum no. of iteration i.e. 3 has taken as termination condition.

[Table-3].

Initial chromosomes				**New centroids obtained using clustering**				**Total distance from each object to their corresponding cluster centroid.**
Xcen 1	**Ycen 1**	**Xcen 2**	**Ycen 2**	**Xcen1**	**Ycen1**	**Xcen2**	**Ycen2**	**Fitness value**
10	10	20	12	8	14	22	20	33.74481
24	30	8	20	15.33333	16	14.66667	18	53.4512
6	12	22	18	8	14	22	20	33.74481
10	10	8	20	12	11.33333	18	22.66667	41.63504

Table: 3. Initial chromosomes and their fitness value

[Table-4].

O/P Of Rank Selection				**O/P Of One Point Crossover**				**O/P Of Mutation**			
Xcen1	**Ycen1**	**Xcen2**	**Ycen2**	**Xcen1**	**Ycen1**	**Xcen2**	**Ycen2**	**Xcen1**	**Ycen1**	**Xcen2**	**Ycen2**
8	14	22	20	8	14	**22**	**20**	8	14	22	**11.163**
8	14	22	20	8	14	**22**	**20**	8	14	22	**12.935**
12	11.333	18	22.666	12	11.333	**14.666**	**18**	12	11.333	14.666	**15.235**
15.333	16	14.666	18	15.333	16	**18**	**22.666**	15.333	16	18	**7.3521**

Table: 4. New chromosomes obtained using selection crossover and mutation

[Table-5].

Input chromosomes				**New centroids obtained using clustering**				**Total distance from each object to their corresponding cluster centroid.**
Xcen1	**Ycen1**	**Xcen2**	**Ycen2**	**Xcen1**	**Ycen1**	**Xcen2**	**Ycen2**	**Fitness value**
8	**14**	**22**	11.1634	8	14	22	20	**33.74481**
8	14	22	12.935	8	14	22	20	33.74481
12	11.33333	14.66667	15.2351	5.333333	7.333333	24.66667	26.66667	50.8914
15.33333	16	18	7.3521	23.33333	30	6.666667	4	59.23666

Table: 5. Input chromosomes and their fitness value for second generation

[Table-6].

Input chromosomes				**New centroids obtained using clustering**				**Total distance from each object to their corresponding cluster centroid.**
Xcen1	**Ycen1**	**Xcen2**	**Ycen2**	**Xcen1**	**Ycen1**	**Xcen2**	**Ycen2**	**Fitness value**
8	**14**	**22**	36.8774	22	24	8	10	39.31851
8	14	22	23.4780	8	14	22	20	**33.74481**
5.333333	7.333333	6.666667	4.6832	24.6667	26.6667	5.3	7.3	50.89
23.33333	30	24.66667	53.1812	15.33	16	14.67	18	53.4512

Table: 6. Input chromosomes and their fitness value for third generation

After the termination condition reached, the chromosome with least fitness value is taken and the clustering is done with respect to that. For this example, finally the two clusters c1(1,4,5) and c2(2,3,4) with Centroid1=(8, 14) and Centroid 2=(22, 20) having SSD=239.99 is obtained as output. Normally the output of GA based data clustering depends on a no. of parameters like no. of iterations, no. of Chromosomes, selection technique, crossover probability, mutation probability.

[IV] COMPARATIVE STUDY OF K-MEANS AND GA BASED DATA CLUSTERING

K-MEANS BASED DATA CLUSTERING	GA BASED DATA CLUSTERING
Partitioning Based Method	Evolutionary Based Method
Input: k, dataset, randomly chosen k centroids	Input: k, dataset, P, randomly chosen P chromosomes, tmax,
Objective: Minimizing sum of squared distance	Objective: Minimizing the sum of distances from each data point to its cluster centroid
Termination condition: No changes in new cluster centroids.	Termination condition: Maximum no. of iterations reached.
Final clustering may converge to local optima.	GA is based on global search approaches with implicit parallelism.

Time complexity: O(n*k*d*i)
Where
n= no. of data points
k= no. of clusters
d= dimension of data
i= no. of iterations
	Time complexity: O(tmax*p*n*k*d)
	Where
	n=no. of data points
	k= no. of clusters
	d= dimension of data
	tmax= maximum no. of iterations
	p= population size

[V] CONCLUSION

Clustering is an important unsupervised classification technique where a set of data objects taken in a multi-dimensional space, are grouped into clusters in such a way that data objects in the same cluster are similar in some sense and objects in different clusters are dissimilar in the same sense. K-Means is an intuitively simple and effective clustering technique, but it may get stuck at suboptimal solutions, depending on the choice of the initial cluster centers. Whereas GA is a randomized search and optimization technique guided by the principles of evolution and natural genetics, and having a large amount of implicit parallelism. So it provides near optimal solutions for objective or fitness function of an optimization problem. Under limiting conditions, a GA-based clustering technique is expected to provide an optimal clustering, more superior to that of K-Means algorithm, but with little more time complexity.

REFERENCES

[1] Dash R, Mishra D, Rath A. K, Acharya Milu. [2010] A hybridized K-means clustering approach for high dimensional dataset *International Journal of Engineering, Science and Technology* 2(2): 59-66

[2] Fahim A. M, Salem A. M, Torkey F. A, Saake G. and Ramadan M. A. [2009] An efficient k-means with good initial starting points *Georgian Electronic Scientific Journal: Computer Science and Telecommunications* 2(19): 47-57.

[3] Kala R, Shukla A. and Tiwary R. [2010] A Novel Approach to Clustering using Genetic Algorithm *International Journal of Engineering Research and Industrial Applications* 3(1): 81-88.

[4] Lin Hwei Jen, Yang Fu-Wen and Kao Yang-Ta. [2005] An Efficient GA-based Clustering Technique *Tamkang Journal of science and Engineering* 8(2): 113-122.

[5] Maulik U, Bandyopadhyay S. [2000] Genetic algorithm-based clustering

technique *Pattern Recognition* 33: 1455-1465.

[6] Nazeer K, A. Abdul and Sebastian M.P, [2009] Improving the accuracy and efficiency of the k-means clustering algorithm *Proceedings of the World Congress on Engineering* 1: 308-312.

[7] Sun Hao-jun, Xiong Lang-huan. [2009] Genetic Algorithm based High-Dimensional Data Clustering Technique *International Conference on Fuzzy Systems and Knowledge Discovery* 1: 485-489.

[8] Tiwari A. K, Sharma L.K, G. Rama Krishna. [2010] Entropy Weighting Genetic k-Means Algorithm for Subspace Clustering *International Journal of Computer Applications* 7(7): 27-30

[9] Wei Jian-Xiang, Liu Huai, Sun Yue-hong and Su Xin-Ning. [2009] Application of Genetic Algorithm in Document Clustering *Information Technology and Computer Science* 1: 145 -148.

[10] Xu R. and Wunsch D. [2005] Survey of clustering algorithms *IEEE Trans. Neural Networks*, 16(3): 645-678.