TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.4, April 2014, pp. 2924 ~ 2935
DOI: http://dx.doi.org/10.11591/telkomnika.v12i4.4805

Hybrid K-means Algorithm and Genetic Algorithm for Cluster Analysis

Dianhu Cheng*[1], Xiangqian Ding[1], Jianxin Zeng[2], Ning Yang[3]
[1]Ocean University of China, Shandong Qingdao, China
[2]China tobacco Yunnan Industrial Co., Ltd, Yunnan, Kunming, China
[3]Newstar Computer Engineering Center of Qingdao Ocean University, Shandong Qingdao, China
*Corresponding author, email: 35113479@qq.com

Abstract

Cluster analysis is a fundamental technique for various filed such as pattern recognition, machine learning and so forth. However, the cluster number is predefined by users in K-means algorithm, which is unpractical to implement. Since the number of clusters is a NP-complete problem, Genetic Algorithm is employed to solve it. In addition, due to the large time consuming in conventional method, an improved fitness function is proposed. According to the simulation results, the proposed approach is feasible and effective.

Keywords: *cluster analysis, K-means algorithm, genetic algorithm, cluster number, time consuming*

1. Introduction

Cluster analysis is to group a set of objectives in the way that the objectives in the same cluster have the similar characterizes each other, while they have differing characterizes among clusters. In recent research, it has been a useful tool for statistical data analysis and implemented in data fusion, machine learning, information retrieval and signal processing. As a general problem, cluster analysis could be solved by various kinds of algorithms according to the kind of the clusters' notion and algorithms' efficiency. In common, the notion of the clusters includes groups with small distances among the cluster individuals, diversity of the data space, the distribution of the particular intervals and etc. For this reason, clustering could be considered as a multi objective optimization problem. The type of individual data set and expected target decide the employment of clustering algorithm and could be helpful to parameter tuning. In dealing with cluster analysis as a multi-objective optimization problem, there exist trials and failures in the interactive process of knowledge. Therefore it is necessary to tune parameters and modify the optimization models until the desired results are achieved.

The algorithms for cluster analysis can be categorized based on the cluster model. Up to now, there are more than hundreds of algorithms proposed. None of them is overwhelmingly better than others. However, it is possible to choose a proper algorithm for a particular problem by experimental or empirical results, unless one cluster model is better than others by proof in a mathematical way. An example is given that the K-means method cannot find non-convex clusters [1]. The prominent clustering methods are connectivity based clustering (such as hierarchical clustering), Centroid-based clustering (such as k-means clustering), distribution-based clustering and density-based clustering.

In this decade, a lot of attention has been paid to cluster analysis and the performance has been improved [2-4]. With the development of information science, the processing of huge data set such as a big data problem has been a pressing need. The willingness to trade semantic and image meaning of the generated clusters for performance has been dramatically increasing. It results in the development of pre-clustering methods such as canopy clustering, which can process huge data sets efficiently, but the resulting "clusters" are merely a rough pre-partitioning of the data set to then analyze the partitions with existing slower methods such as k-means clustering. Various other approaches to clustering have been tried such as seed based clustering [5].

Received September 5, 2013; Revised November 9, 2013; Accepted November 23, 2013

For the data set with high-dimension, many of the existing approaches will fall down to clustering due to the curse of dimensionality, which presents that the particular distance functions is difficult to solve the data clustering in high-dimensional spaces. This results in the generation of novel clustering analysis methods for the data set with high-dimension which mainly focus on the subspace clustering and correlation clustering that also searches arbitrary rotated subspace clusters that can be modeled by designing a correlation of their characteristics. Famous ones of this kind of clustering algorithms include CLIQUE [6] and SUBCLU [7]. In addition, the ideas inspired from density-based clustering methods (in particular the DBSCAN / OPTICS family of algorithms) have been employed to subspace clustering such as HiSC [8] and DiSH [9] and correlation clustering such as HiCO, [10]. The "correlation connectivity" is proposed and implemented in 4C [11] and ERiC [12] which explored hierarchical density based correlation clusters).

Based on the concept of mutual information, several different clustering systems are proposed. The examples are given as Marina Meilă's variation of information metric [13] and hierarchical clustering [14, 15]. In addition, a recently proposed method message passing algorithms, has led to the creation of new types of clustering algorithms [16].

In this paper, we investigated a K-means algorithm, which the cluster number is predefined by users in K-means algorithm. In practical problem, it is impossible to decide the number cluster in advance. Since the number of clusters is a NP-complete problem, Genetic Algorithm is employed to solve it. In addition, due to the large time consuming in conventional method, an improved fitness function is proposed. According to the simulation results, the proposed approach is feasible and effective. We proposed a hybrid K-means algorithm and genetic algorithm for cluster analysis.

The rest of this paper is organized as follows. Section II briefly introduces the concept of cluster analysis in mathematic way. Section III introduces the framework of Genetic Algorithm for clustering. In Section IV, an improved version to enhance the algorithms performance is proposed. The simulation and results analysis are conducted in Section V. This paper is ended with conclusions and future work is proposed in Section VI.

2. Preliminary of Cluster Analysis

In computer science, clustering analysis is a classic problem and relevant technologies have been a key task in the process of acquiring knowledge [17, 18]. It has been implemented in many applications such signal processing, data mining, machine learning [19-21]. The target of clustering is to partition a given data set into several subsets termed clusters, which maximizes the homogeneity of the data intra one cluster and the heterogeneity inter clusters. To find optimal clustering is a challenge task in current research.

Up to now, evaluation the similarities among data is based on the measure of the distance of two data, which the Euclidean distance is most used. In clustering, considering that in a data set $A=\{a_1,a_2,...,a_N\}$, there are N data, where each $a_i \in R^\rho$ is an attribute vector consisting of ρ real valued measurements. The goal of clustering is to partition the data into several groups $C=\{C_1,C_2,...,C_M\}$, where M is the number of clusters. The mathematical description can be given as follows [22, 23]:

$$\bigcup_{i=1}^{M} C_i = A \tag{1}$$

Where:

$$C_i \neq \varnothing \quad \text{for } i=1,...,M \tag{2}$$

And,

$$C_i \cap C_j = \varnothing \text{ for } i=1,...,M, \quad i \neq j \tag{3}$$

The Euclidean distance can be defined as f and given as follows:

$$f = \arg\min_{f} E_{VQ}\left(\vec{c_1},....,\vec{c_M}\right) = \arg\min_{f} \sum_{i=1}^{M} \left\|\vec{x_i} - \vec{c}_{f(\vec{x_i})}\right\|^2 \quad (4)$$

Where,

$$\vec{c}_m = \frac{1}{|C_m|} \sum_{\vec{x}_i \in \vec{C}_m} \vec{x}_i, \quad m = 1,...,M \quad (5)$$

Hence, searching for the centers of clusters can instead the function f and can rewrite f as follows:

$$f\left(\vec{x}\right) = \arg\min_{i} \left\|\vec{x} - \vec{c}_i\right\|^2 \quad (6)$$

Which means the distance can be measured from the data point to the center of a cluster.

In most previous work, the number of clusters is calculated based on designer's requirement or experience. However, a small number of clustering cannot partition the data into suitable groups, while the large number of clusters will make the clustering no sense. The formula that calculated the number of clusters is shown as follows:

$$NW\left(N,M\right) = \frac{1}{M!}\sum_{i=0}^{M}(-1)^i \binom{M}{i}(M-i)^N \quad (7)$$

According to (7), it is difficult to obtain the best clustering even that the cluster number M is known, let along unknown in practice. The conventional method is empirical, which is to employ a suitable value of m based on the results analysis after conducting simulations several times. Due to the limitation of the domain knowledge and searching for the best solution only in a small scale, the performance of the methods is not satisfied. Other than the predefined criterion, a feasible method to optimize the value of M is based on the numeric criteria. In this case, the number of cluster ways is given as follows [23]:

$$\sum_{i=1}^{n} NW\left(N,M\right) \quad (8)$$

With the assume that the value of M is unknown, which is more reasonable, the problem of finding an optimal value for M to partition N data could considered as a NP-complete problem and the complexity of the problem can be calculated approximately as $\frac{M^N}{M!}$. Therefore, attempting to obtain an optimum solution by conventional methods is not computationally feasible [22] and it is necessary to appeal to an efficient approximation algorithms.

Heuristic algorithms are well known for solving NP-complete problems and Genetic Algorithm is one of the famous heuristic algorithms [24, 25]. It can obtain good enough solutions in reasonable time. Thus in this paper, Genetic Algorithm is employed for solving the adaptive number of clustering problem.

3. Framework of Genetic Algorithm for Clustering

3.1. Genetic Algorihtm

As a branch in the field of artificial intelligence, Genetic Algorithms plays an important role in optimization and searching problems. It belongs to the larger class of evolutionary algorithms (EA) which is inspired by nature evolution. Due to the advantages of efficiency, accuracy and easy implement, it has found applications in computer science, engineering, chemistry, manufacturing, bioinformatics and other fields.

Like other heuristic algorithm, Genetic Algorithm is population-based, which contains a certain number of chromosomes where consisting of genes [26]. Each chromosome can be considered as a solution that can be mutated and altered. The encoding of Genetic Algorithm can be conducted in binary space, but other encodings are also feasible. The main operators in Genetic Algorithm are crossover and mutation. Crossover is used to reassemble the chromosomes from one generation to the next. Mutation is used to enrich the diversity of chromosome in each generation so that Genetic Algorithm can obtain better solutions. It occurs during the whole evolutionary process according to a predefined mutation probability which should be set as a small value. Otherwise, the mutation probability is so large that the evolution process will be a primitive random search. The schedule of Genetic Algorithm is shown as follows, and the flowchart is given in Figure 1.

Step 1. Initialize a population of a certain number of random chromosomes;
Step 2. Evaluate each chromosome according to the fitness function.
Step 3. Select chromosome according to the proportional selection probability.
Step 4. Normalize the chromosomes so that the chromosomes can be compared.
Step 5. Conduct crossover and mutation operators.
Step 6: Select top ranking chromosomes as the population in the next generation.
Step 7: If the stopping criteria is satisfied, end the algorithm. Otherwise, go to Step 2.

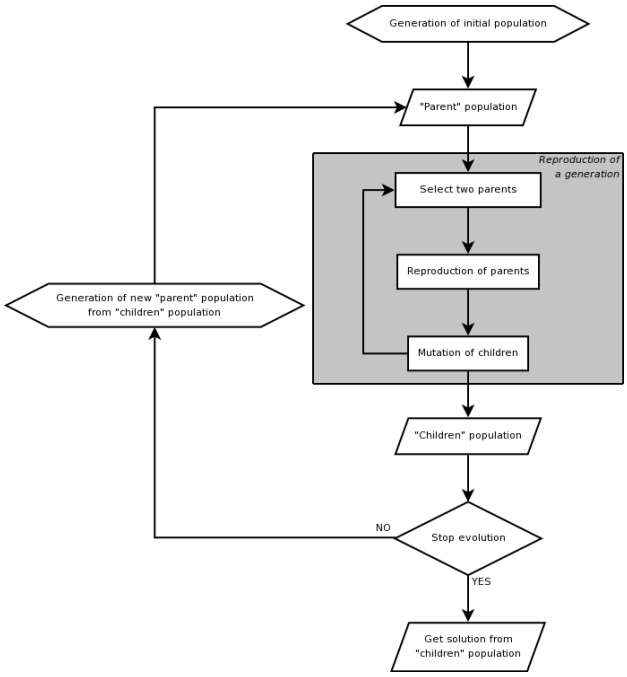

Figure 1. The Flowchart of Genetic Algorithm

3.2. Encoding

The Genetic Algorithms for clustering is based on a simple scheme. Considering that a data set is consisting by N data, there are $N+1$ ways to partition the set. To illustrate the strategy clearly, an example is given by assuming one chromosome in Genetic algorithm is encoded as follows:

Chromosome 1: 3 23221

The chromosome includes two parts: the first number “3” means the number of clusters and the rest numbers present the group index. In Chromosome 1, the data at the location index {1} is 3, which means there are 3 groups in the clustering. The data at the location index {3, 4, 5} belong to the Group 2, and the data at the location index {3} and location index {6} belongs to the Group 3 and Group 1 respectively. Considering Chromosome 1, there are many different coding ways to express the same case. For example:

Chromosome 2 : 3 31332

Chromosome 2 has the same meaning as Chromosome 1. Actually, there are 6 different ways to express the same meaning. Therefore, the size of the searching space of genetic algorithm is much larger than the virtual space, which will lead to a poor efficiency of algorithm. Other than the poor efficiency, it also affects the crossover operation in Genetic Algorithm since the redundant coding could make the offspring no any improvement. For example, if Chromosome 1 is chosen to cross over with Chromosome 2, there exists the probability that the offspring has the same meaning with parents.

In addition, the connection with genes in one chromosome is not taken into account. To solve the problem, a hybrid K-means and Genetic Algorithms are proposed. Actually, the inter connections among gene values constitute the genuine optimization goal in clustering problems. Based on the analysis, the development of genetic operators specially designed to clustering problems has been investigated.

3.3. Crossover Operation

To solve the problems mentioned above, there are three kinds of crossover are employed in this paper, one point crossover, two points crossover and combining crossover [27, 28].

One-point crossover shown in Fig.1 is similar with the binary one point crossover. The point on both parents' chromosomes is selected. All data beyond that point in either chromosome is swapped between the two parent organisms. The resulting organisms are the offspring [33, 35, 38].

Two-point crossover shown in Figure 2 calls for two points to be selected on the parent organism strings. Everything between the two points is swapped between the parent organisms, rendering two offspring organisms:

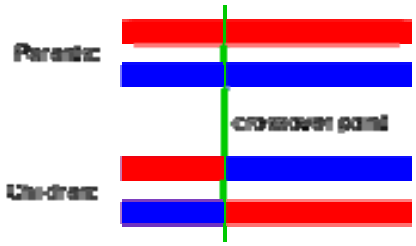

Figure 1. The Sketch Map of One Point Crossover

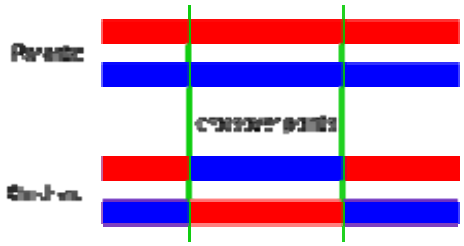

Figure 2. The Sketch Map of Two Point Crossover

The combining crossover combines the two solutions. It builds the new offspring center by center. For each center from the parent chromosome it finds the nearest centers from the second parent and generates two new centers randomly on the line joining the two parent centers.

3.4. Mutation Operation

Mutation operation is a key operator in Genetic Algorithm which can explore the searching space and break away local minima. In cluster analysis, two chromosomes are employed to conduct the mutation, one is adopt where is closer to a centroid of a cluster and the other one is adopt by the data where is closer to the farthest data from the centroid. Then the two genes are mutated by mutation operator. [29, 36, 37]

3.5. Fitness Function

As described above, clustering could be considered as an optimization problem. To evaluate each solution, the fitness function is defined as (4). Thus, without loss of generality, we define a solution as $S = \{\vec{c}_1, ..., \vec{c}_M\}$, then it can be evaluated by the fitness function,

$$f(S) = -E_{VQ}\left(\vec{c}_1, ..., \vec{c}_M\right) \tag{9}$$

In some other paper, different methods are proposed, such as silhouette which defined an average distance between $\vec{x}$ and others.

$$a\left(\vec{x}\right) = \frac{1}{|A|}\sum_{\vec{y}\in A}\left\|\vec{x} - \vec{y}\right\|^2 \tag{10}$$

Besides, the distances between one data and a cluster can be calculated as follows:

$$b\left(\vec{x}\right) = \min_{C\neq A} d\left(\vec{x}, C\right) \tag{11}$$

Hence, the silhouette of $\vec{x}$ is given as follows:

$$s\left(\vec{x}\right) = \frac{b\left(\vec{x}\right) - a\left(\vec{x}\right)}{\max\left\{a\left(\vec{x}\right), b\left(\vec{x}\right)\right\}} \tag{12}$$

Therefore the fitness function is given by:

$$f(S) = \sum_{i=1}^{N} s\left(\vec{x}_i\right) \tag{13}$$

3.6. Normalization

In standard Genetic Algorithm for clustering, the normalization could help enhance the convergence performance of algorithm, which is described as follows:

1. The clusters are initialized to produce $C_1, ..., C_M$.
2. For all $\vec{x}_i \in S$, put $\vec{x}_i$ into a cluster C_l , where:

$$l = \arg\min_m \left\|\vec{x}_i - \vec{c}_m\right\|^2 \tag{14}$$

3. Sort the centers of each cluster $C_1, ..., C_M$ and update each data as follows:

$$\vec{c}_i = \frac{1}{|C_i|}\sum_{x\in C_i}\vec{x} \tag{15}$$

Where $i = 1, ..., M$.

4. Design for the Improved Genetic Algorithms Clustering

4.1. Hybrid K-means clustering and Genetic Algorithm

The idea of K-means clustering was proposed by Steinhaus in 1957 [39] and first used by MAcQueen in 1967 [40]. Now, it has been s a popular approach in cluster analysis. The schedule can be summarized as follows: Given as initial set of M means $m_1^{(1)}, m_2^{(1)}, ..., m_M^{(1)}$, the algorithm deals with the objectives by alternating between the following two steps:

Assignment: Assign each objective to the cluster whose mean is closest to it.

$$C_i^{(t)} = \left\{x_p : \left\|x_p - m_i^{(t)}\right\| \leq \left\|x_p - m_j^{(t)}\right\|, \forall 1 \leq j \leq M\right\} \tag{16}$$

Where each x_p is assigned to a certain cluster $C^{(t)}$, no matter whether it could be assigned to other clusters.

Update: Update the means to ensure the means are located at the center of each new corresponding cluster.

$$m_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{x_i \in S_i^{(t)}} x_j \tag{17}$$

The algorithm ends if the assignments no longer change. Although in standard K-means algorithm the number of clusters should be predefined, it could be combined with Genetic Algorithm to blend their advantages to enhance the clustering performance.

In addition, according to the design of (13), it is easy to find that the time consuming of the evaluation for solutions is very large both in time complexity and space complexity since a huge matrix should be storage and the matrix calculation is time expensive. To solve this problem, it is necessary to improve the fitness function. According to (12), to maximize the value of $b(S)$ and minimize the value of $a(S)$ could maximize the fitness function. To solve the problem, a novel fitness function is proposed as follows:

$$s(S) = \frac{b(S)}{a(S) + \delta} \tag{18}$$

Where $a(S)$ and $b(S)$ have the same meaning with (12), δ is a small value to guarantee the denominator not be zero. We use function (18) instead, which have the same characters and reduce both of the time complexity and space complexity.

The schedule of the hybrid algorithm is described as follows,

Step 1: Initialize chromosomes to compose a population.

Step 2: Use k-means algorithm to the chromosome.

Step 3: Based on (18) calculate the fitness function and the chromosomes are evaluated.

Step 4: Conduct the crossover and mutation operator.

Step 5: If the stopping criteria is meet, end the algorithm. Otherwise, go to Step 2.

5. Simulation Results and Analysis

In this section, several simulations are conducted to test the performance of proposed method. We term the improved Clustering Genetic Algorithm as ICGA and give the simulation results as follows. It is obvious that our proposed method achieves the fastest convergence and obtains a better accuracy. Ruspini Dataset, Vowels and mushroom problems are employed.

5.1. Ruspini Dataset

In Ruspini dataset problem there are 80 objects which are includes two features {x,y}. The main goal of this benchmark is to compare different genetic operators and investigate the effects on the algorithm performance.

The simulation environment is set as follows. VC++ 6.0 is employed. The hardware is 2.7*2 GHz CPU and 2*1G RAM. In each simulation, a chromosome consisted with 4 genes. To measure the distance among objective, Euclidean norm is employed and all the data are normalized in the interval [0, 1].

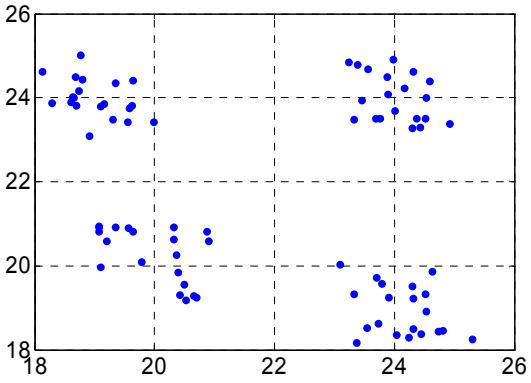

Figure 3. Ruspini Dataset

The algorithm stops once one of the following two conditions is met,

1. The maximum iteration 100 is reached.
2. The error is less than or equal to 0.001.

We run each algorithm 50 times to obtain an average performance.

First, the domain of the clustering number belongs to the integer set {2,…,40}, which means there are at least 2 clusters and at most 40 clusters in the analysis. The reason to set the maximum number as 40 is that although there exists 80 objectives in this problem, it makes no sense if we set 80 clusters in this problem although 80 is the maximum number. To set the clusters belongs to the set {2,…,N/2} is reasonable to apply in practice. The simulation results are given in Table 1. Time consuming represents the average time cost for the 50 simulations and the first reach iteration means the iteration index which the results are not improved afterward during one simulation. The smaller first reach iteration means a better convergence ability of an algorithm.

Table 1. Results of Randomly Generated Initial Population

Algorithm	Time consuming	First reach iteration
K-means	0.25	62.7
CGA	0.22	50.2
ICGA	0.21	31.3

Table 1 indicated that the performance of CGA is better than K-means both in time consuming and first reach iteration, while ICGA is better than both of them. According to Table 1, the conclusion can be drawn that the first reach iteration is correlated positive proportional with the time consuming. However, it should be noted that for different benchmarks and simulations, the same iterations may cause a huge time consuming due to the probabilistic characterized in heuristic algorithms.

The set of $k \in \{2,...,40\}$ is very helpful to the simulation results for initial populations generated randomly. Next, the two cases, 2 clusters and 40 clusters respectively, based on the number of clusters are taken into account so that additional diversity represented by initialization can be ignored. This consideration is not practical but useful to evaluate the proposed algorithms and the simulation results are given in Table 2.

Table 2. Results of the Fixed Clusters Number (2 clusters and 40 clusters)

	2 Clusters	40 Clusters
Algorithm	Time consuming	First reach iteration
K-means	0.197	87.4
CGA	0.053	61.9
ICGA	0.026	47.1

According to Table 2, the average time consuming of ICGA is less than those for both CGA and K-means. Besides, the performance of CGA is better than K-means. For the first reach iteration, ICGA is also overcome others.

Figure 4-1 shows a dataset which is used to test the CGA and K-means algorithm [32]. In Figure 4-2, it shows that ICGA recognize all the clusters and the centers are located well, while Figure 4-3 indicates that CGA falls down to find out all the clusters. In Figure 4-2, each set has a central point which is marked by a black point. Hence the performance of ICGA could find the center of data set with a good performance. However, in Figure 4-3, some centers are far away to the cluster and some of them are overlapped. For the (Row 1, Line 3), (Row 4, Line 1) and (Row 5, Line 5), there are overlapped points in the data set. For the (Row 3, Line 3), (Row 4, Line 5), (Row 5, Line 2), (Row 5, Line 3), there are no central points in the data set. In Figure 4-3, it is obvious that some center points are far away from data sets. Hence it could be concluded that the proposed algorithm ICGA has a huge ability in cluster analysis.

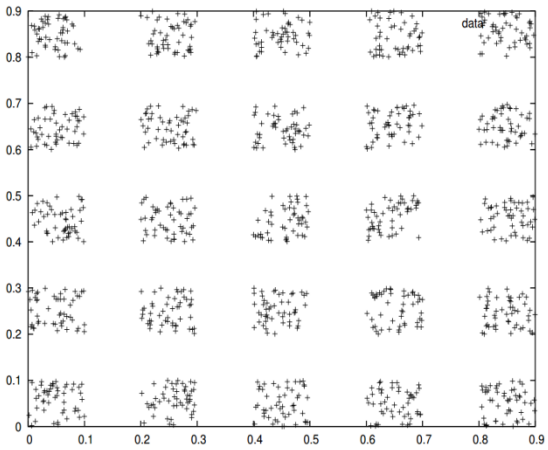

Figure 4-1. Data Set

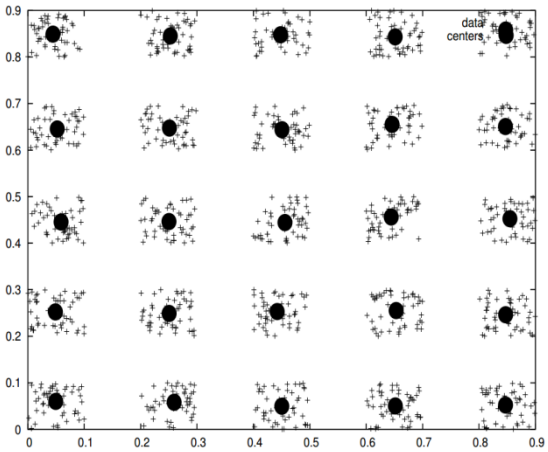

Figure 4-2. Results Obtained by ICGA

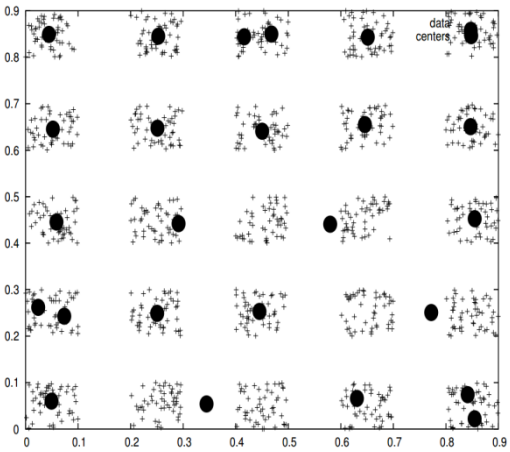

Figure 4-3. Results Obtained by CGA

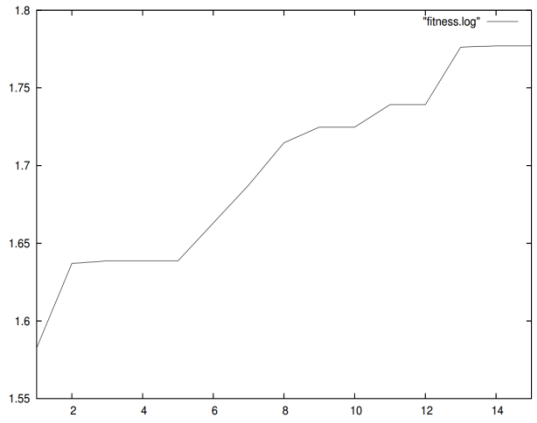

Figure 5-1. Fitness Evaluated by the Silhouette Function

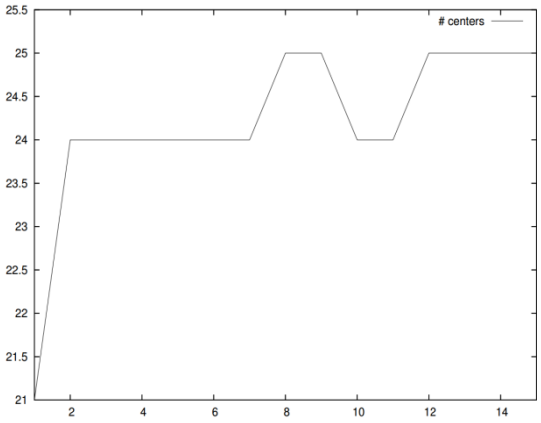

Figure 5-2. Numbers of Clusters in Best Solution in Iteration

Figure 5 shows the fitness function to determine the optimal number of clusters, which is to verify the feasibility of ICGA with the silhouette fitness function. According to Figure 5, with the increment of fitness function, the number of clusters is approximately invariant, which shows the proposed algorithm has the ability to find out the best cluster number.

5.2. Ruspini Dataset

In this subsection, the simulations are to compare different crossover and mutation operators in Genetic Algorithm to test the effects to the performance of algorithms [33, 34, 35]. First the employed dataset are illustrated as follows,

SODAR Data Set 1: This data described a 2 dimensions space which has 3 clusters. The number of all the objectives is 90.

SODAR Data Set 2: This dataset is similar with SODAR Data Set 1, which has 3 clusters in a 2 dimensions space and has 3 clusters. The total number of observations is 90. In Table 1, it has been referred to as Sodar2.

Wisconsin Breast Cancer Database Original: This dataset is also maintained in the UC Irvine Machine learning repository and the data obtained from the University of Wisconsin Hospitals. It is 9 dimensions space and has 2 clusters. The number of the total objectives is 699.

Second, the one point crossover is run with different kinds of mutation. The simulation results are shown in Table 3. In Table 3, the K-means mutation is inferior to the one point mutation for 25 clusters problem but superior to one point mutation for mushroom and vowels problems.

Table 3. Comparison of Different Mutation Operators

	Cancer	SODAR 1	SODAR 2
One point mutation	0.22	1368.5	930.1
K-means mutation	0.27	1362.4	927.9

Table 4. Comparison of Different Mutation Operators

	Cancer	SODAR 1	SODAR 2
1 point crossover	0.22	1367.5	927.7
Combine crossover	0.27	1519.9	927.5

Different crossover operators are compared in Table 4 which shows that for the 25 clusters task, simpler operator (one point crossover) obtains the best results were achieved. However, for the Mushroom and Vowel problem, the best performance is achieved by K-means mutation.

The accuracies of CGA and ICGA are compared in Table 5, where Iris dataset is employed and which shows that the proposed algorithm has a better performance in accuracy. It outperforms both K-means and CGA.

Table 4. Comparison of Accuracy

Method	K-means	CGA	ICGA
Accuracy	95.9%	97.6%	98.7%

6. Conclusions and Future Works

In this paper, the Genetic Algorithm Clustering is hybridized with K-means algorithm to merge their advantages to propose an adaptive number of clusters. The number of cluster could be optimized by Genetic Algorithm which is more reasonable in practice. In addition, since the conventional silhouette function has the lots of matrix storage and calculation, we improved the fitness function which reduces the time complexity and space complexity. The classic benchmarks are investigated, and the results show that the method is feasible and effective to conduct cluster analysis.

Acknowledge

This work is supported by National Twelfth Five-Year Science and Technology Support Program (2012BAH12F01)

References

[1] Estivill-Castro V. Why so many clustering algorithms. *ACM SIGKDD Explorations Newsletter*. 2002; 4: 65.
[2] Z Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*. 1998 20022:283–304.
[3] R Ng, J Han. *Efficient and effective clustering method for spatial data mining*. Proceedings of the 20th VLDB Conference. Santiago, Chile. 1994: 144-155.
[4] Tian Zhang, Raghu Ramakrishnan, Miron Livny. *An Efficient Data Clustering Method for Very Large Databases.* Proc. Int'l Conf. on Management of Data, ACM SIGMOD. 103–114.
[5] Can F Ozkarahan EA. Concepts and effectiveness of the cover-coefficient-based clustering methodology for text databases. *ACM Transactions on Database Systems.* 1990; 15(4): 483.
[6] Agrawal R, Gehrke J, Gunopulos D, Raghavan P. Automatic Subspace Clustering of High Dimensional Data. *Data Mining and Knowledge Discovery*. 2005; 11: 5.
[7] Karin Kailing, Hans-Peter Kriegel, Peer Kröger. Density-Connected Subspace Clustering for High-Dimensional Data. In:Proc. SIAM Int. Conf. on Data Mining (SDM'04). 2004; 246-257.
[8] Achtert E, Böhm C, Kriegel HP, Kröger P, Müller-Gorman I, Zimek A. Finding Hierarchies of Subspace Clusters. LNCS: Knowledge Discovery in Databases: PKDD 2006. *Computer Science*. 2006; 4213: 446–453.
[9] Achtert E, Böhm C, Kriegel HP, Kröger P, Müller-Gorman I, Zimek A. Detection and Visualization of Subspace Cluster Hierarchies. LNCS: Advances in Databases: Concepts, Systems and Applications. *Computer Science.* 2007; 4443: 152–163.
[10] Achtert E, Böhm C, Kröger P, Zimek A. *Mining Hierarchies of Correlation Clusters.* Proc. 18th International Conference on Scientific and Statistical Database Management (SSDBM). 2006; 119–128.
[11] Böhm C, Kailing K, Kröger P, Zimek A. *Computing Clusters of Correlation Connected objects.* Proceedings of the 2004 ACM SIGMOD international conference on Management of data – SIGMOD. 2004; 455.
[12] Achtert E, Bohm C, Kriegel HP, Kröger P, Zimek A. On Exploring Complex Relationships of Correlation Clusters. 19th International Conference on Scientific and Statistical Database Management (SSDBM). 2007: 7.
[13] Meilă Marina. Comparing Clusterings by the Variation of Information. Learning Theory and Kernel Machines. *Computer Science.* 2003; 2777: 173–187.
[14] Alexander Kraskov, Harald Stögbauer, Ralph G Andrzejak, Peter Grassberger. Hierarchical Clustering Based on Mutual Information. 2003; ArXiv q-bio/0311039
[15] Auffarth B. Clustering by a Genetic Algorithm with Biased Mutation Operator. WCCI CEC. *IEEE.* 2010.
[16] BJ Frey, D Dueck. Clustering by Passing Messages between Data Points. Science. 2007; 315(5814): 972–976.
[17] H He, J Sun, Y Wang, Q Liu, J Yuan. Cluster Analysis Based Switching-off Scheme of Base Stations for Energy Saving. *Journal of Networks*. 2012; 7(3): 494-501.

[18] P Arabie, LJ Hubert. An Overview of Combinatorial Data Analysis. (Chapter 1). Clustering and Classification, ed. P Arabie, LJ Hubert, G DeSoete. *World Scientific*. 1999.
[19] H Park, D Baik. A study for control of client value using cluster analysis. *Journal of Network and Computer Applications*. 2006; 29(4): 262-276.
[20] H She, Z Lu, A Jsantsch, D Zhou, L Zheng. Performance analysis of flow-based traffic splitting strategy on cluster-mesh sensor networks. *International Journal of Distributed Sensor Networks*. 2012; 2012.
[21] L Yuan, X Wang, J Gan, Y Zhao. A data gathering algorithm based on mobile agent and emergent event-driven in cluster-based WSN. *Journal of Networks*. 2010; 5(10): 1160-1168.
[22] L Kaufman, PJ Rousseeuw. Finding Groups in Data – An Introduction to Cluster Analysis. *Wiley Series in Probability and Mathematical Statistics*. 1990.
[23] G Liu. Introduction to combinatorial mathematics. McGraw Hill. 1968.
[24] L Guo, S Zhao, S Shen, C Jiang. Task scheduling optimization in cloud computing based on heuristic Algorithm. *Journal of Networks.* 2012; 7(3); 547-553.
[25] J Sun, S Gao, W Yang, Z Jiang. Heuristic replica placement algorithms in content distribution networks. *Journal of Networks*. 2011; 6(3): 416-423.
[26] RJ Streifel, MJ Robert, R Reed, JJ Choi, M Healy. Dynamic fuzzy control of genetic algorithm parameter coding. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*. 1999; 29(3): 426-433.
[27] S Jiang, Q Zhou, Y Zhang. Analysis on parameters in an improved quantum genetic algorithm. *International Journal of Digital Content Technology and its Applications*. 2012; 6(18): 176-184.
[28] J Guo, L Sun, R Wang, Z Yu. An improved quantum genetic algorithm. *International Conference on Genetic and Evolutionary Computing.* 3rd International Conference on Genetic and Evolutionary Computing, WGEC. 2009; 14-18.
[29] U Maulik, S Bandyopadhyay. Genetic algorithm-based clustering technique. *Pattern Recognition*. 2009; 33: 1455–1465.
[30] M Keijzer, JJ Merelo, G Romero, M Schoenauer. Evolving objects: A general purpose evolutionary computation library. *Artificial Evolution*. 2001; 231–244.
[31] S Bao, Z Deng, Z Chen. Stochastic validation of structural FE-models based on hierarchical cluster analysis and advanced Monte Carlo simulation. *Finite Elements in Analysis and Design*. 2013; 67: 22-33.
[32] U Reuter. A fuzzy approach for modelling non-stochastic heterogeneous data in engineering based on cluster analysis. *Integrated Computer-Aided Engineering*. 2011; 18(3): 281-289.
[33] U Ghose, CS Rai, Y Singh. Socio economic characterization of student's data using ICA and cluster analysis. *IEEE International Conference on Industrial Informatics (INDIN).* 2010; 714-718.
[34] L Niw, M Yan. *Application of hierarchical cluster analysis in classification of tunnel rock masses.* International Conference on Remote Sensing, Environment and Transportation Engineering, RSETE. Proceedings. 2011; 2944-2947.
[35] J Castellanos. A visual analytics framework for cluster analysis of DNA microarray data. *Expert Systems with Applications*. 2013; 40(2): 758-774.
[36] N Wang, Y Yang. An iterative fuzzy identification method hybridising modified objective cluster analysis with genetic algorithm. *International Journal of Modelling, Identification and Control.* 2010; 10(1-2): 44-49.
[37] L Lee, Y Qu, K Lee. Mining qualitative patterns in spatial cluster analysis. *Expert Systems with Applications*. 2012; 39(2): 1753-1762.
[38] Q Yang. *Application of fuzzy cluster analysis to tax planning for location of foreign direct investment*. Proceedings 3rd International Conference on Information Management, Innovation Management and Industrial Engineering, ICIII. 2010; 1: 553-556.
[39] Steinhaus H. Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci.* (in French). 1957; 4(12): 801–804.
[40] MacQueen JB. *Some Methods for classification and Analysis of Multivariate Observations*. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1. University of California Press. 1967; 281–297.