ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

Genetic Algorithm with Novel Crossover, Selection and Health Check for Clustering

A. H. Beg and Md Zahidul Islam

School of Computing and Mathematics
Charles Sturt University, Panorama Avenue
Bathurst 2795, Australia

Abstract. We propose a genetic algorithm for clustering records, where the algorithm contains new approaches for various genetic operations including crossover and selection. We also propose a health check operation that finds sick chromosomes of a population and probabilistically replaces them with healthy chromosomes found in the previous generations. The proposed approaches improve the chromosome quality within a population, which then contribute in achieving good clustering solution. We use fifteen datasets to compare our technique with five existing techniques in terms of two cluster evaluation criteria. The experimental results indicate a clear superiority of the proposed technique over the existing techniques.

1. Introduction

Clustering is a well-known data mining technique. It aims to group similar records in one cluster and dissimilar records in different clusters. It has a wide range of applications including business, machine learning and social network analysis [1-6]. There are many existing clustering algorithms. K-means is one of the most popular techniques for its simplicity and light weight i.e. low complexity. However, a drawback of K-means is its requirement of a user defined number of clusters (k). In reality it can be difficult for a user to estimate the appropriate number of clusters in advance [4, 5, 7]. Therefore, clustering techniques that are capable of finding the number of cluster automatically are highly desirable.

Moreover, another drawback of K-means is that it has tendency to getting stuck at local optima resulting in poor quality clustering results [4, 5]. In order to overcome these limitations in recent years many GA based [2-6] approaches for clustering were proposed that achieved encouraging results. Genetic algorithms (GA) are heuristic search and optimization techniques based on the concepts of natural activities of genes, individual selection and evolution process [2-6].

In GA a chromosome is considered to be a clustering solution and a gene of a chromosome is considered to be the center of a cluster. However, there are some limitations of the existing GA based clustering techniques. Generally, the number of genes of a chromosome are generated randomly in the initial population. The genes are also selected randomly from a dataset instead of carefully choosing them. Careful selection of genes increases the possibility of getting high quality chromosomes in the initial population. Having high quality chromosomes in the initial population is more likely to produce a good quality clustering solution [4, 5].

Recently, an existing technique called GenClust [4] produces high quality chromosomes in the initial population and thereby obtain good clustering result. However, the complexity for generting the initial population is high $O(N^2)$, where N is the number of records in a dataset. Moreover, GenClust also requires a user to define different radius values for generating the initial population. It can be very difficult for a user to define the radius values for each individual dataset separately.

Moreover, the gradual health improvement is also crucial for a GA to finally find a good quality chromosome. In each generation, GA goes through some genetic operations such as crossover and mutation. The crossover and mutation operation can improve the fitness/heath of a chromosome, but they can also decrease the health of some chromosomes. Therefore, it is important to check the chromosome health at the end of each generation.

In this paper, we propose a genetic algorithm based clustering technique called "**G**enetic Algorithm with Novel **C**rossover, **S**election and Health Check for Clustering (GCS)" that solves the above mentioned issues. We now introduce the main contributions of this study as follows. Following the approach of an existing technique [5] GCS produces high quality chromosomes in the initial population through two phases: a deterministic phase and a random phase **.** It does not require any user input on the number of cluster (k) and keeps the complexity low, $O(N)$. GCS uses two phases of selection operation in order to increase the quality of chromsomes in a population. It also modifies the process which selects a pair of chromosomes in a crossover operation through two phases in order to increase the possibility of getting better quality offspring chromosomes.

Moreover, the presence of healthy chromosomes (chromosomes with high fitness values) in a population increases the possibility of getting good quality of the final clustering result. Therefore, GCS uses a Health Check operation in order to find sick chromosomes and replaces them with healthy chromosomes. It also uses the elitist operation after each genetic operation within a generation, in order to keeps track of the best solution obtained so far. It ensures that the best chromosome is not lost.

We evaluate the proposed technique by comparing its performance with the performance of five high quality existing techniques namely AGCUK [2], GAGR [3], GenClust [4], K-means [7] and K-means ++ [8]. Two evaluation criteria called Silhouette coefficient [1] and DB index [2, 5] are used. Detail experimental results on fifteen (15) datasets indicate clear superiority of the proposed technique over five existing techniques. Therefore, the main contribution of the proposed technique can be summarized as follows.

- Two phases of selection operation.
- Two phases of crossover operation.
- Health check operation.
- Elitist operation after each genetic operation.

The rest of the paper is organized as follows: We present our novel technique in Section 2. In Section 3 we discuss experimental results and in Section 4 we give the concluding remarks.

2. Our Technique

We now briefly explain the main steps of GCS and their advantages as follows.

Step 1- Normalization: GCS takes a dataset *D* as input. It first normalizes the dataset *D* in order to weigh each attribute equally regardless of their domain sizes. The normalization brings the domain range of each numerical attribute of the dataset between 0 and 1 [4, 5].

Step 2- Population Initialization: We prepare an initial population (***P***) of 2× |r| chromosomes, |r| chromosomes from the deterministic phase and |r| chromosomes from the random phase. The value of *r* is set to10, in this study. In the deterministic phase, GCS uses a set of predefined numbers of genes in order to produce a set of chromosomes. The default set of predefined *k* is {2, 3, 4……10} where the size of the predefined set is nine. GCS uses each element of the set as the predefined number of cluster (*k*) in K-means and thus produce a clustering solution. It then runs K-means five times for each element and thus produce five chromosomes. Hence, it produces altogether 5 * 9 = 45 chromosomes in the deterministic phase. GCS then selects top |r| chromosomes (according to their fitness values) from these 45 chromosomes. Typically the value of *k* of a dataset varies between 2 to 10 which is supported by the empirical analysis of DeRanClust [5]. Therefore, GCS uses the set of *k* {2, 3, 4….10} in the deterministic phase.

However, in many dataset the actual *k* values are more than 10. In order to handle such situation GCS uses the random phase. For each chromosome it generates the *k* value randomly between 2 to $\sqrt{N}$ (N is the number of records in a data set) and then selects *k* number of records randomly to from *k* genes of a chromosome. GCS produces |r| chromosomes in the random phase. Thus, GCS produces 2× |r| chromosomes from two phases. It then find the best chromosome P_b from 2× |r| chromosomes and stores it for the elitist operation. The fitness of each chromosome is calculated using DB Index [2, 5]. A small value of DB Index indicate a good clustering result, therefore the fitness is computed by 1/*DB*.

Step 3- Two Phases of Selection Operation: Starting from generation 2, GCS applies the two phases of selection operation in order to get a new population for the next genetic operations such as crossover and mutation. In Phase 1, GCS selects the *top* |r| chromosomes (according to the fitness values) from 2× |r| chromosomes of the current population. In Phase 2, it selects |r| chromosomes probabilistically from a set of 3× |r| chromosomes, which is made of the remaining *bottom* |r| chromosomes of the current population and 2× |r| chromosomes from the last population of the immediate previous generation.

Step 4- Crossover Operation: GCS performs a crossover operation on a pair of chromosomes, where each chromosome is first divided into two segments and then the chromosomes swap segments (like any single point crossover [6]) in order to generate a pair of offspring chromosomes. We propose two phases of crossover operation in GCS. In Phase 1, GCS selects 2× |r| -1 pair of chromosomes, where in each pair the 1st chromosome is always the best chromosome of the population. All other chromosomes are chosen one by one to be the 2nd chromosome of a pair. All pairs have different 2nd chromosome.

In order to facilitate extensive exploration**,** GCS applies this crossover operation five times on each pair and thereby generates 5 different pair**s** of offsprings. That is it produces altogether 5 × (2× |r| -1) × 2 chromosomes**,** from which it then selects the top |r| chromosomes. This phase increases the possibility of getting high quality offspring chromosomes. In order to maintain the random exploration ability, GCS in Phase 2 also uses the traditional roulette wheel [4, 5] approach for selecting pairs of chromosomes for crossover. In Phase 2, it selects |r| pairs of chromosomes and applies the traditional single point crossover. GCS then produces |r| offspring chromosomes from Phase 2. Thus, from the two phases it finally produces 2× |r| chromosomes.

Step 5- Mutation Operation: Following the approaches of existing techniques [2, 5] GCS applies division and absorption mutation operation. For the best chromosome GCS applies both the division and absorption. For all remaining chromosomes it randomly applies either division or absorption operation. For the division operation GCS find the sparsest cluster and then divides the cluster into two clusters by applying K-means where the value of k is set to 2. In the absorption operation GCS finds two closest clusters and merges those into one cluster. The clusters that has minimum seed to seed distance is considered as the closest one.

Step 6- Health Check Operation: GCS applies the proposed health check operation after I generations. In this study we use I = 20. GCS prepares a set of chromosomes S, where it stores the best chromosomes of each generation for the first I generations. It then calculates the average fitness F_d of the chromosomes in S. If the fitness of a chromosome in the current population is less than F_d, the chromosome is selected as sick. GCS then probabilistically selects a chromosome from S to replace the sick chromosome.

***Step 7-Elitist Operation**:* Generally GA [2-5] applis the the elitist operation at the end of each generation. However, GCS applies the elitist operation at the end of each genetic operations within a generation. If the fitness of the worst chromosome P_w^i of the i-th population (i.e. the current population) is less than the fitness of the best chromosome P_b^{All} (from all previous generation) then P_w^i is replaced with P_b^{All}. Moreover, if the fitness of the best chromosome P_b^i of the i-th population has the higher fitness than P_b^{All} then P_b^{All} is replaced by P_b^i.

GCS continues step 3 to step 7 for the total number (user defined) of iterations. At the end of all iterations GCS selects P_b^{All} as the best chromosome. The genes of the best chromosome represent the cluster centers and records are allocated to their closest seeds to form the final clusters.

3. Experimental Results and Discussion

We empirically compare the performance of our technique with five existing techniques called AGCUK [2], GAGR [3], GenClust [4], K-Means [7] and K-means ++ [8] on 15 natural datasets (shown in Table 1) that are available in the UCI machine learning repository [9]. The existing techniques are shown in the literature [2-4] are recent, of high quality and to be better than many other techniques. For the experimentation of AGCUK, GAGR, GenClust and GCS we consider the population size to be 20. The number of generations/iterations for all techniques set to be 50 for a fair comparison. The cluster number in GGAR and AGCUK are generated randomly in a range 2 to $\sqrt{N}$ (N is the number of records in a data set). We run each technique 20 times on each dataset and we take the average result. Two evaluation criteria: Silhouette Coefficient and DB Index are used. The higher value of Silhouette Coefficient represents the better clustering result and the lower value of DB index indicate the better clustering result.

Fig. 1, Fig. 2, Fig. 3, and Fig. 4 shows that GCS performs better than all other techniques in all 15 datasets based on Silhouette Coefficient and DB Index. Moreover, in 14 out of 15 datasets the standard deviation of GCS do not overlap the standard deviation of GenClust based on Silhouette Coefficient. The standard deviation of GCS do not overlap the standard deviation of AGCUK in 12 out 15 datasets based on Silhouette Coefficient. Note that the cases where the standard deviation of GCS overlap with the standard deviations of other techniques are indicated with an arrow.

Dataset	Records
Glass Identification (GI)	214
Vertebral Column (VC)	310
Ecoli (EC)	336
Leaf (LF)	340
Liver Disorder (LD)	345
Dermatology (DT)	366
Blood Transfusion (BT)	748
Pima Indian Diabetes (PID)	768
Statlog Vehicle Silhouettes (SV)	846
Bank Note Authentication (BN)	1372
Yeast (YT)	1484
Image Segmentation (IS)	2310
Wine Quality (WQ)	4898
Page Blocks Classification (PBC)	5473
MAGIC Gamma Telescope (MGT)	19020

Table 1: Dataset at a glance

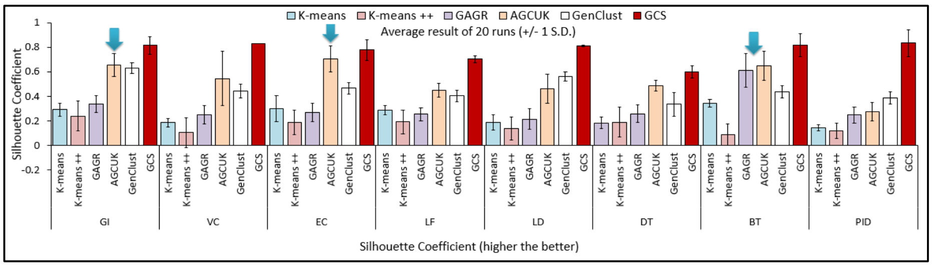

Fig. 1: Silhouette Coefficient of the techniques on eight datasets

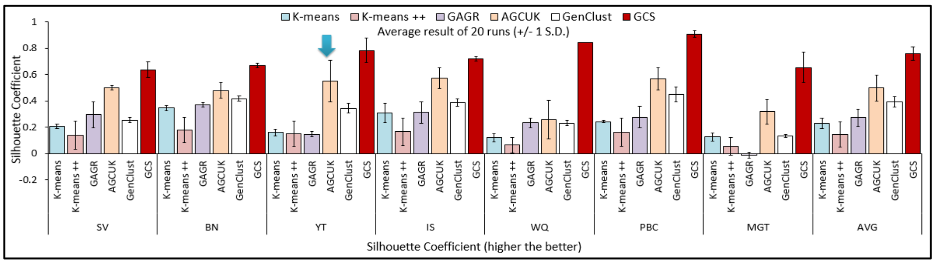

Fig. 2: Silhouette Coefficient of the techniques on seven datasets

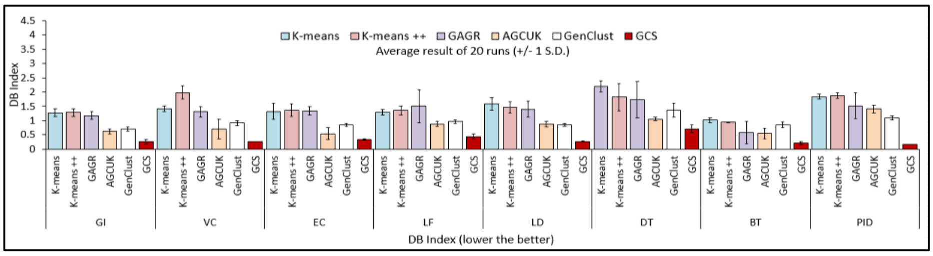

Fig. 3: DB Index of the techniques on eight datasets

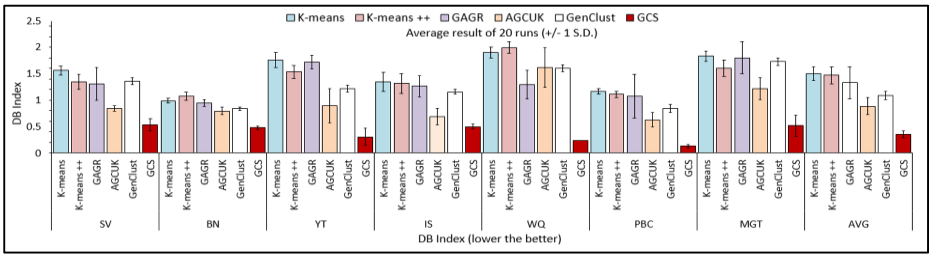

Fig. 4: DB Index of the techniques on seven datasets

The right most column in Fig. 2 and Fig.4 show the average Silhouette Coefficient and DB Index of all techniques on all datasets. GCS achieves clearly better results on an average than all other techniques without any overlapping of the standard deviations. We believe that this is a very strong result in order to demonstrate the superiority of the proposed technique over a number of recent and high quality clustering techniques.

Conclusion

GCS uses proposed crossover and selection approaches in order to increase the quality of chromosomes in a population. It also ensures gradual health improvement of the chromosomes through the proposed health check operation. The experimental results based on two commonly used cluster evaluation criteria clearly indicate superiority of the proposed technique over five existing techniques. Our future research plan includes the proposal of new and effective genetic operations in order to achieve better clustering results.

References

[1] P. N. Tan, M. Steinbach, V. Kumar, Introduction to data mining, 1st ed., Pearson Addison Wesley, 2006.

[2] Y. Liu, X. Wu, and Y. Shen, Automatic clustering using genetic algorithms, *Applied Mathematics and Computation*, 218:1267-1279, Elsevier, 2011.

[3] D. Chang, X. Zhang, and C. Zheng, A genetic algorithm with gene rearrangement for K-means clustering, *Pattern Recognition*, 42:1210-1222, Elsevier, 2009.

[4] M.A. Rahman, and M.Z. Islam, A hybrid clustering technique combining a novel genetic algorithm with K-Means, *Knowledge-Based Systems*, 71:345-365, Elsevier, 2014.

[5] A. H. Beg, and M.Z. Islam, Clustering by Genetic Algorithm- High Quality Chromosome Selection for Initial Population, *IEEE 10th Conf. on Industrial Electronics and Applications*, pp. 129-134, 2015.

[6] P. Peng, et al., Reporting and analyzing alternative clustering solutions by employing multi-objective genetic algorithm and conducting experiments on cancer data, Knowl-Based Syst. Elsevier, 56:108-122, 2014.

[7] S.P, Lloyd, Least squares quantization in PCM. IEEE Transactions on Information Theory, 28 (2): 129-13, 1982.

[8] D. Arthur, S. Vassilvitskii, k-means++: The Advantages of Careful Seeding, SODA '07 Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pp.1027-1035, 2007.

[9] UCI Machine Learning Repository, Retrieved from http://archive.ics.uci.edu/ml/ (June 22, 2013).