Modified K-Means Algorithm for Genetic Clustering

Mohammad Babrdel Bonab

Islamic Azad University Bonab Branch, Iran

Summary

The K-Means Clustering Approach is one of main algorithms in the literature of Pattern recognition and Machine Learning. Yet, due to the random selection of cluster centers and the adherence of results to initial cluster centers, the risk of trapping into local optimality ever exists. In this paper, inspired by a genetic algorithm which is based on the K-means method , a new approach is developed, in which cluster centers are selected and computed appropriately. Examining the suggested approach by using standard data sets and comparing it with alternative methods in the literature reveals out that the proposed algorithm outperforms the K-means algorithm and other candidate algorithms in the pool.

Key words:

Clustering, the K-Means Algorithm, the Genetic Algorithm

1. Introduction

Data mining literally revolves around extracting knowledge from huge amounts of stored data in databases or other information sources in an automated manner. Data mining emerging at the late 1980's accelerated its paces during the 90's and it is even expected to advance faster in the current century. Clustering is an importing technique in data mining. Clustering is a method of grouping similar samples. Each formed group is then called a cluster.[1]

Generally, clustering algorithms and techniques are introduced in different ways which can be classified generally into five categories: Partitioning, Hierarchical, Concentration-Based, Grid-Based and Model-Based methods. In a partitioning method first k data partitions are built up with at least one data item for each partition and k <= n. If the petitioning is of Hard-type a data item can contribute in one cluster while in the Fuzzy-type model of partitioning one data item can participate in several clusters albeit with different membership grades. Two famous exploratory algorithms for Hard partitioning are K-means and K-methods algorithms. The Fuzzy counterparts of these algorithms are Fuzzy K-means and Fuzzy K-methods algorithms respectively.[1],[2]

The K-means algorithm is a major clustering method. Despite its simplicity it is a basic method for many other clustering methods. In its simple version, first some points as many as required clusters are selected randomly. Then based on the ongoing similarities and trends each data item is allocated to one relevant cluster which in turn creates new clusters as well. Accordingly, by repeating the same procedure and by averaging data items new cluster centers are calculated and then data items are reassigned to new clusters. This procedure continues as long as there is no observable change among data items. Although in this algorithm convergence is guaranteed but due to its dependence on the initial cluster centers the ultimate solution is neither unique nor necessarily optimum.[5]

Many different approaches have been suggested to overcome this shortcoming of K-means algorithm such as Particle Swarm Optimization (PSO), The Colony of Ants ,Taboo Search,[5] The Metal Annealing method and some other combined solutions[6]. In some approaches it is tried to overcome this challenge by selecting a suitable set of initial cluster centers.[8]

The rest of this paper comes as follow: In section (2) and (3) the concepts of clustering and the K-Means algorithm are reviewed. In section (4) the suggested approach is introduced. In section (5) experimental results of implementing the suggested approach are reported and finally the section (6) incorporates concluding remarks.

2. Clustering

Clustering is defined as grouping similar objects either physical or abstract .Each created group is named a cluster. The objects inside one cluster have most similarity with each other and maximum diversity with other groups.[1],[2].

Definition: Suppose the set of $X = \{x_1, x_2, \dots, x_k\}$ containing n objects. The purpose of clustering is to group objects in k clusters as $C = \{c_1, c_2, \dots, c_k\}$ while each cluster satisfies the following conditions:

1) $C1 \cup C2 \cup \dots \cup Ck = X$
2) $C_i \neq \emptyset \qquad i = 1 \dots k$
3) $Ci \cap Cj \neq \emptyset$

According to the above definition possible states for clustering n objects in k clusters are:

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

In most approaches, the number of clusters i.e. k is specified by user. Relation (1) implies that even with a

Manuscript received September 5, 2011
Manuscript revised September 20, 2011

given k, finding out the optimum solution for clustering is not so simple. Also the number of possible solutions for clustering n objects in k clusters increases by the order of $\frac{k^n}{k!}$. So finding the best state for clustering n objects in k clusters is an intricate NP-Complete problem which needs to be solved by optimization techniques.[4]

3. The K-Means Algorithm

There have been suggested many algorithms for addressing the clustering problems among them the $K-Means$ Algorithm is a top famous one. The main stages of the $K-Means$ Algorithm for clustering data items come as below:

Stage I: choose k data items randomly from $X = \{x1, x2, \dots, xn\}$ as cluster centers (m1,m2,…,mk)

Stage II: Based on the relation (2) add every data item to a relevant cluster. I.E. if the following relation (2) holds, the object xi from the set of $X = \{x1, x2, \dots, xn\}$ is added to the cluster Cj

$$\|x_i - m_j\| < \|x_i - m_p\| \quad 1 \le p \le k, j \ne p \quad (2)$$

Stage III: Now based on the clustering of stage II new cluster centers (m1*,m2*,…,mk*) are re-estimated by using the relation(3) as below .(ni is the number of objects in the cluster i)

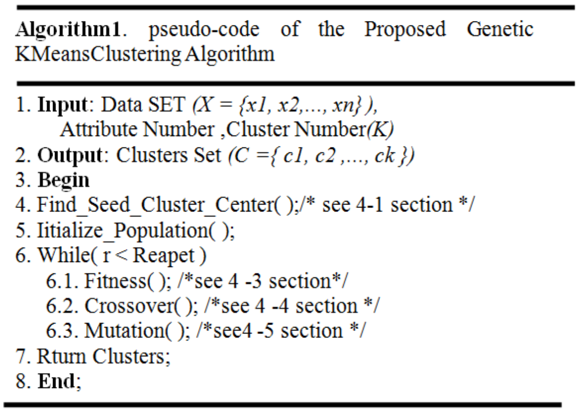
$$m_j^* = \frac{1}{n_j} \sum_{x_j \in C_i}^{n} x_j \qquad 1 \le i \le k \qquad (3)$$

Stage IV: If the cluster centers are changed, repeat the algorithm from Stage II, else do clustering based on the resulted centers.

As observable, initial cluster centers are selected stochastically in the proposed algorithm. Stochastic selection of initial cluster centers makes this algorithm yield out different results for different runs over the same data sets which are considered as one of potential weak points of this algorithm.[5]

4. Conclusion The suggested algorithm

In the majority of genetic clustering algorithms, chromosomes keep cluster centers [5]. Cluster centers in a chromosome initially are selected from data items. The main idea rests on the fact that at first data items are clustered in the Stage I of The *K-Means* Algorithm according to their individual attributes. The number of generated clusters at this stage is more than (or equal to) the primary clusters of input data . Now data centers inside every chromosome is selected among these initial cluster centers in a non-iterative mode. Algorithm 1 (below) outlines the suggested algorithm generally.



Algorithm1. pseudo-code of the Proposed Genetic KMeansClustering Algorithm

```
1. Input: Data SET (X = {x1, x2,..., xn} ),
      Attribute Number ,Cluster Number(K)
2. Output: Clusters Set (C ={ c1, c2 ,..., ck })
3. Begin
4. Find_Seed_Cluster_Center( );/* see 4-1 section */
5. Iitialize_Population( );
6. While( r < Reapet )
    6.1. Fitness( ); /*see 4 -3 section*/
    6.2. Crossover( ); /*see 4 -4 section */
    6.3. Mutation( ); /*see4 -5 section */
7. Rturn Clusters;
8. End;
```

4.1 An algorithm for finding seed points of clusters

In this algorithm, the proposed approach for finding initial cluster centers comes as follows: First by using The K-Means Algorithm all data items are clustered due to their individual attributes. Now regarding the generated clusters for every attribute in every stage, one single pattern is formed for every data item. Hence the obtained clusters at this stage outnumber the initial clusters. The main idea in this paper is borrowed from Ref [8]. In this Reference clustering is rendered in two stages. The first stage is conducted as mentioned recently in the paper. But in the second stage similar clusters are merged until a pre-specified number of clusters are achieved. Algorithm 2 shows the proposed approach for initial clustering of data items. The newly achieved cluster centers are called cluster seed points.

As easily seen from the Algorithm 2 for every attribute of data items, it is generated a cluster label for each data object and this label is added to the data object. Objects with the same patterns are encapsulated in the same cluster. For generating each data label for every attribute first the mean and standard deviation of values of that attribute are calculated for all data items. Then based on the mean and standard deviation the interval of attribute values is divided into k identical intervals. Now all initial centers of data items are clustered by the K-Means method.

4.2 the chromosome structure

Chromosomes are presented in string formats. If k is the number of clusters and q represents the number of attributes for data items the chromosome's length will be $q * k$ and is defined as:

$$M = \lfloor m_{11}, m_{12}, \dots, m_{1q}, m_{21}, m_{22}, \dots, m_{2q}, \dots, m_{k1}, m_{k2}, \dots, m_{kq} \rfloor$$

According to the above structure $M_i = \lfloor m_{i1}, m_{i2}, \dots, m_{iq} \rfloor$ is called the center of cluster i.

At the beginning of the algorithm a predefined number of chromosomes are generated with the above-mentioned structure. The cluster centers inside chromosomes are selected randomly among seed points generated by applying the recently described algorithm in the *subsection 4-1* in a non-iterative manner.

Algorithm2. pseudo-code of the Proposed Find_Seed_Cluster_Center Algorithm

```
1. Input: Data SET ($X = \{x1, x2, \dots, xn\}$)
,Attribute Set ($A = \{A1, A2, \dots, Aq\}$),Cluster
Number(K),
2. Output: Clusters Seed Set
($SC = \{sc1, sc2, \dots, scH\}$)
3. Begin
4. while ($\forall Aj \in A$)
        4.1. Compute Mean ( $\mu_j$ ) and Standard
             Deviation ( $\sigma_j$ )
        4.2. Compute Cluster Center( $e$= 1,2,..., $k$ )
                 $X_e = Z_e * \sigma_j + \mu_j$    $Z_e = \frac{2*e-1}{2*k}$
        4.3. Execute K-means on this attribute
        4.4. Allocate cluster labels obtained from step 6.3 to
             every data pattern
5. Find unique patterns ($H \geq k$) and clustering
   each data whit obtained patterns.
6. Return SC
7. End;
```

4.3 Testing the chromosome's fitness

For computing the fitness value of each chromosome from the scratch by the help of the *Relation (2)* and based on cluster centers the data objects are assigned to the clusters. Now following the clustering new cluster centers are re-estimated by using *Relation (3)* and are replaced in chromosome formats. Based on new cluster centers the fitness value of each chromosome is calculated according to:

$$Fitness(C) = \sum_{i=1}^{k} \sum_{x_j \in C_i} \| x_j - m_i \| \qquad (4)$$

4.4 The Crossover Operator

A single point Crossover technique method is used for combining chromosomes with a fix probability of P_c. First a random point namely z is generated in the *[1,k]* interval. Now the two selected chromosomes are cut down from the cutting point of *cut=z*q* (*q* equals the number of attributes) and the right-hand sides of chromosomes are interchanged.

4-5 the Mutation Operator

A Mutation operator is used for searching in the unattended solutions pool with a fix probability of P_M. First a random gene such as m_{ir}, $1 \le i \le k, 1 \le r \le q$.

Is selected from a random chromosome. To compute the new gene value m_{ir} first a random δ value between *[-R,R]* is generated for which the amount of *R* is computed as:

$$R = \begin{cases} \dfrac{M - M_{min}}{M_{Max} - M_{min}}, & M_{Max} > M \\ 1 \quad , & M_{Max} = M_{min} \end{cases} \qquad (5)$$

Where *M* is the fitness value of the selected chromosome, and M_{max} ,M_{min} are respectively the maximum and minimum values among all existing chromosomes. The value of m_{ir} gene is calculated by the Relation (6) as below:

$$m_{ir} = \begin{cases} m_{ir} + \delta \times (m_{max}^{ir} - m_{ir}), & \delta > 0 \\ m_{ir} + \delta \times (m_{ir} - m_{min}^{ir}), & \delta < 0 \end{cases} \qquad (6)$$

Where in Relation (6) m_{ir} is the value of the ir^th gene in the selected chromosome and m_{max}^{ir} , m_{min}^{ir} are respectively the available maximum and minimum values for the ir^th gene among all present chromosomes.

5. Experimental Results

The suggested algorithm is coded in C#.NET programming language and ran in a *Pentium 4* computer with *3.08 GHz* microprocessor speed and *512 MB* main memory.

For measuring the efficiency of the proposed algorithm the standard data items of Table (1) are used. The execution results of the proposed algorithm over the selected data sets as well as the comparison figures relative to reported K-Means , *PSO* , *K-NM-PSO* results in *Ref [6]* are tabulated in the Table (2) .As easily seen in the Table, the suggested algorithm provides superior results relative to *K-Means* and *PSO* algorithms.

Table 1. The characteristics of the used data sets

Name of data set	No. Of attribute	No. of Cluster	Size of data set
Iris	4	3	150(50,50,50)
Wine	13	3	178(59,71,48)
CMC	9	3	1473 (629, 334, 510)
Glass	9	6	214 (70, 17, 76, 13, 9, 29)

For better survey and analysis of the proposed approach, the execution results of the proposed approach along with, *SA, PSO, PSO-ACO, PSO-ACO-k, PSO-AS,ACO-AS, HBMO, GA, ACO,TS,K-Means* results which are reported in *Ref [7]* are tabulated in the *Tables 3 to 6*. It is worth mentioning to note that the investigated algorithms of Ref [7] are implemented in Matlab7.1 by engaging a Pentium IV system of *2.8 GHz* CPU speed and 512 MB main memory.

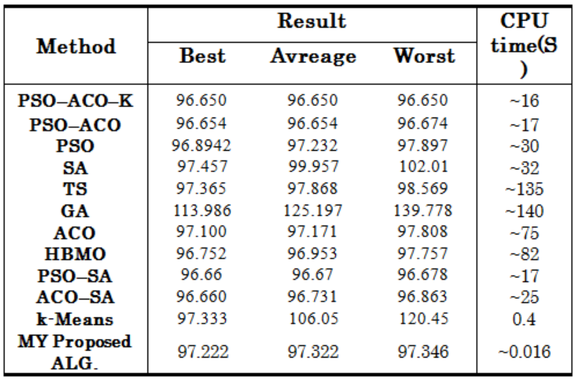
In *Tables (3) to (6)* best, worst and average results are reported for 100 runs respectively. The resulted figures represent the distance of every data item (object) from the center of the cluster to which belongs and is computed by using the *Relation (4)*. As simply observed in the tables, regarding the due execution time the proposed algorithm generates acceptable solutions.

Table 2. The obtained results from implementing the suggested algorithm over selected data sets

Data Set	PSO[Yi-Tung et al. 2008]	K-NM-PSO[Yi-Tung et al. 2008]	Proposed Alg. Result	CPU time(S)
Iris	96.66	96.66	97.222	0.016
Wine	16294.00	16292.00	16530.53	0.015
CMC	5538.50	5532.40	5541.64	0.109
Glass	271.29	199.68	226.28	0.031

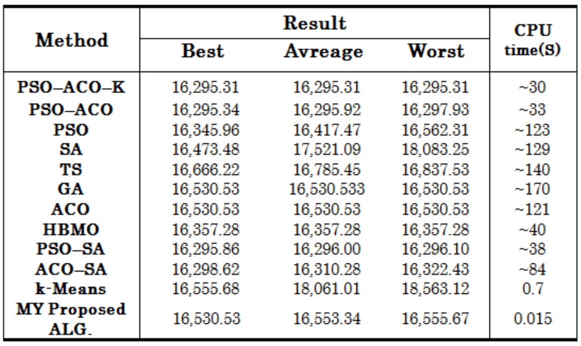
According to the reported results in *Tables* (2) to (6) the suggested algorithm over the CMC data sample (with *1473 data* items) provides the best solution in comparison with the other alternative algorithms. (*Refer to the Table 1*). according to Tables 2 and 5 as well as *Tables 3,4 and 6* the *PSO-ACO-K* algorithm provides the best solution for the data sample in-question among all alternative algorithms while according to the *Table (5)* the best result belongs to our suggested algorithm. The reason for this behavior is justified by the fact that as data items increase in number the efficiency of the under-study algorithms decrease while the deficiency of the suggested algorithm highlights more.

Table 3- The results of implementing the algorithms over Iris test data for 100 runs:



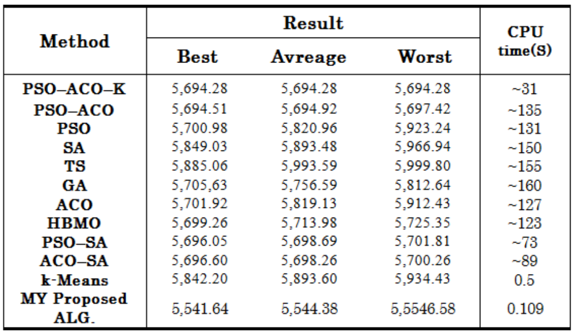
Method	Result			CPU time(S)
	Best	Avreage	Worst	
PSO–ACO–K	96.650	96.650	96.650	~16
PSO–ACO	96.654	96.654	96.674	~17
PSO	96.8942	97.232	97.897	~30
SA	97.457	99.957	102.01	~32
TS	97.365	97.868	98.569	~135
GA	113.986	125.197	139.778	~140
ACO	97.100	97.171	97.808	~75
HBMO	96.752	96.953	97.757	~82
PSO–SA	96.66	96.67	96.678	~17
ACO–SA	96.660	96.731	96.863	~25
k-Means	97.333	106.05	120.45	0.4
MY Proposed ALG.	97.222	97.322	97.346	~0.016

Table 4- The results of implementing the algorithms ver Wine test data for 100 runs:



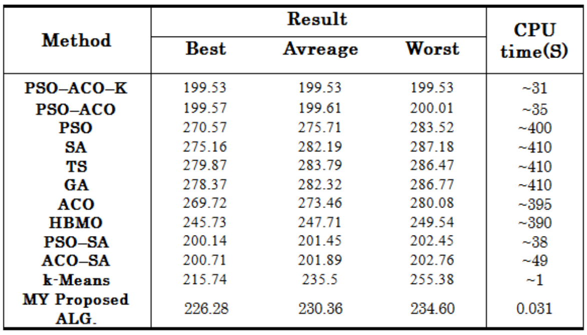
Method	Result			CPU time(S)
	Best	Avreage	Worst	
PSO–ACO–K	16,295.31	16,295.31	16,295.31	~30
PSO–ACO	16,295.34	16,295.92	16,297.93	~33
PSO	16,345.96	16,417.47	16,562.31	~123
SA	16,473.48	17,521.09	18,083.25	~129
TS	16,666.22	16,785.45	16,837.53	~140
GA	16,530.53	16,530.533	16,530.53	~170
ACO	16,530.53	16,530.53	16,530.53	~121
HBMO	16,357.28	16,357.28	16,357.28	~40
PSO–SA	16,295.86	16,296.00	16,296.10	~38
ACO–SA	16,298.62	16,310.28	16,322.43	~84
k-Means	16,555.68	18,061.01	18,563.12	0.7
MY Proposed ALG.	16,530.53	16,553.34	16,555.67	0.015

Table 5- The results of implementing the algorithms over CMC test data for 100 runs



Method	Result			CPU time(S)
	Best	Avreage	Worst	
PSO–ACO–K	5,694.28	5,694.28	5,694.28	~31
PSO–ACO	5,694.51	5,694.92	5,697.42	~135
PSO	5,700.98	5,820.96	5,923.24	~131
SA	5,849.03	5,893.48	5,966.94	~150
TS	5,885.06	5,993.59	5,999.80	~155
GA	5,705,63	5,756.59	5,812.64	~160
ACO	5,701.92	5,819.13	5,912.43	~127
HBMO	5,699.26	5,713.98	5,725.35	~123
PSO–SA	5,696.05	5,698.69	5,701.81	~73
ACO–SA	5,696.60	5,698.26	5,700.26	~89
k-Means	5,842.20	5,893.60	5,934.43	0.5
MY Proposed ALG.	5,541.64	5,544.38	5,5546.58	0.109

Table 6- The results of implementing the algorithms over Glass test data for 100 runs



Method	Result			CPU time(S)
	Best	Avreage	Worst	
PSO–ACO–K	199.53	199.53	199.53	~31
PSO–ACO	199.57	199.61	200.01	~35
PSO	270.57	275.71	283.52	~400
SA	275.16	282.19	287.18	~410
TS	279.87	283.79	286.47	~410
GA	278.37	282.32	286.77	~410
ACO	269.72	273.46	280.08	~395
HBMO	245.73	247.71	249.54	~390
PSO–SA	200.14	201.45	202.45	~38
ACO–SA	200.71	201.89	202.76	~49
k-Means	215.74	235.5	255.38	~1
MY Proposed ALG.	226.28	230.36	234.60	0.031

6. Concluding Remarks

In this paper, based on the K-Means algorithm, a new genetic algorithm is proposed for clustering data items. We can mitigate the main problem of the K-Means algorithm partially by using seed points. The results obtained from implementation of the suggested approach over test data sets accentuate on the efficiency superiority of it over the K-Means algorithm. According to the reported results about the implementation of the proposed algorithm over the selected data sample we found that when the data set is highly massive the proposed approach outperforms the alternative algorithms surveyed in this paper. Although the random selection of cluster centers in chromosome formats in the proposed approach remains still as a problem, but their random selection is rendered among seed points which are substantially less than data items. Furthermore, worthy to say that similar (close) data items are merged with each other to form a single seed point for clusters. In future works it can be specified a special benchmark which allows us to choose seed points selectively in order to improve the suggested algorithm further.

Acknowledgment

The authors would like to express their cordial thanks to Ms. Roohollah Etemadi for his valuable advice.

References

[1] J. Han, and M. Kamber, "Data Mining: Concepts and Techniques", San Francisco: Morgan Kaufmann, 2001.

[2] F. Keller, "Clustering", Computer University Saarlandes, Tutorial Slides.

[3] G.L. Liu," Introduction to Combinatorial Mathematics", McGraw-Hill, 1968.

[4] E.R. Hruschka, N.F.F. Ebecken,"A genetic algorithm for cluster analysis", Intelligent Data Analysis 7(1) 15–25, 2003.

[5] Sanghamitra Bandyopadhyay, Ujjwal Maulik," An evolutionary technique based on K-Means algorithm for optimal clustering in RN ", Information Sciences 146 (2002) 221–237.

[6] Yi-Tung Kao, Erwie Zahara, I-Wei Kao," A hybridized approach to data clustering", Expert Systems with Applications 34 (2008) 1754–1762.

[7] Taher Niknam, Babak Amiri," An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis",Applied Soft Computing 10 (2010) 183–197.

[8] Shehroz S. Khan, Amir Ahmad," Cluster center initialization algorithm for K-means clustering", Pattern Recognition Letters 25 (2004) 1293–1302.

Mohammad Babrdel Bonab received the B.Sc. and M.S. degrees, from Qazvin Azad Univ and Islamic Azad University North Tehran branch. in 2007 and 2010, respectively. After working as a Academic staff and IT Manager in the Islamic Azad University Bonab Branch. His research interest includes Network sensor and Genetic Algorithm.