Journal of mathematics and computer science 11 (2014), 277-290

Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: www.tjmcs.com

ANR: An algorithm to recommend initial cluster centers for k-means algorithm

Arash Ghorbannia Delavar, Gholam Hasan Mohebpour

Department of Computer Science, Payame Noor University, PO BOX 19395-3697,
Tehran, Iran
a_ghorbannia@pnu.ac.ir, *mohebpoor_gh_h@yahoo.com*

Article history:
Received May 2014
Accepted June 2014
Available online July 2014

Abstract

Clustering is one of the widely used knowledge discovery techniques to detect structure of datasets and can be extremely useful to the analyst. In center based clustering algorithms such as k-means, choosing initial cluster centers is really important as it has an important impact on the clustering result. It is desirable to select initial centers which are well separated. In this paper, we have proposed an algorithm to find initial cluster centers based on choosing two attributes that can describe the data space better and using the number of neighbors in a specific radius in data space. The proposed Attribute and Neighborhood Radius based (ANR) initial cluster center computing algorithm is applied to several well-known datasets .experimental results shows that it prevents form choosing noise data points as cluster center and tries to choose data points from dense areas in data space.

Keywords: Data mining, clustering, k-means, Initial cluster centers

1. Introduction

K-means algorithm has been proposed in 1967 by MacQueen and is a typical clustering algorithm which is widely used for clustering datasets. It is a simple, non-supervised algorithm which doesn't need any prior knowledge about the data distribution. K-means algorithm is basically an iterative algorithm and tries to divide the given dataset into k groups to decrease intra cluster distance and increase inter cluster distance. The algorithm consists of two phases. At the first phase it selects k data object as cluster centers randomly, where the value of k is fixed and user defined. At the second phase it assigns each data object to the cluster with the nearest center. K-means generally uses Euclidean distance to determine the distance between data objects and the cluster centers. After all the data objects are included in one of clusters, the early grouping is done and recalculating of cluster centers will be continued until the convergence criterion become the minimum. For each cluster we introduce the mean of instances as new cluster center. Each time when the algorithm computes new cluster centers, these centers are not member of dataset.

$$c_j = \frac{\sum_{i=1}^{n_j} . \mathrm{xi}}{n_j} \tag{1}$$

Objective function of k-means is defined as follow:

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} |x - c_i|^2 \tag{2}$$

E is the sum of the squared error of all instances in dataset. This Objective function tries to produce k clusters so that the objects in the same cluster are as compact as possible while the objects in different clusters are as separated as possible.

Although k-means algorithm process has been demonstrated to be very useful for extracting spherical clusters based on iterative optimization, it often suffers from Sensitivity to initial cluster centers problem such that bad initializations lead to convergence to the wrong local optima and also sometimes it introduces empty clusters .The correct choose of initial cluster centers can help the algorithm to converge faster. Another key limitation of k-means is its cluster model which is based on spherical clusters which the correct choose of initial cluster centers cannot help the algorithm. In this paper we will introduce an algorithm which computes initial cluster centers to improve k-means algorithm efficiency.

2. Related work

Several methods proposed to solve the cluster initialization for k-means algorithm. Bradley and Fayyad [1] proposed an algorithm which builds a set of small random sub-samples of the dataset and then clusters data points in each of these sub-samples by k-means. Centers of all sub-samples are then clustered together by k-means using the centers of each sub-sample as initial centers. The centers of the final clusters detected by k-means which have minimum clustering error will be used as the initial centers.

Likas et al. [2] proposed the global k-means algorithm which incrementally adds one cluster center at any iteration through a deterministic global search procedure consisting of executing k-means algorithm for N times where N is the size of dataset.

Khan and Ahmad [3] have proposed an algorithm to find initial cluster centers based on two observations, which similar data points form the core of clusters and their cluster membership remain the same and also individual attribute provides information in computing initial cluster centers. This algorithm assumes that the distribution of data for each attribute is normal so for K clusters the normal curve is divided into K equal partitions and the midpoint of each interval is computed using mean and standard deviation of the attribute. The process is repeated for all the attributes and finds K cluster centers.

Deelers and Auwatanamongkol [4] proposed an algorithm which separates the dataset in to some cells and tries to reduce the sum squared errors of the two cells as much as possible until the number of cells equals to the predefined number of clusters .Finally the centers of the k cells become as initial cluster centers for k-means algorithm.

Arai and Barakbah [5] proposed a hierarchical method to find initial cluster centers by using k-means several times and obtain a set of centers from each different run .These centers will form a dataset which will be used in a hierarchal clustering algorithm to find the best centers.

F. Cao et al. [6] have presented a method for initializing k-means Based on Rough set Concepts [7] and neighborhood of data points. This method defines Cohesion Degree of the neighborhood of any data point and the coupling degree between neighborhoods of all data points and selects the data point with the highest cohesion degree as the first center. After that, it will repeatedly continue until finding all initial cluster centers.

Alaa H. Ahmed ,Wesam Ashour [8] have presented an initialization method for k-means clustering based on reverse nearest neighbor (RNN) and Coupling Degree . For each data point RNN (x_i) is a set of data points whose nearest neighbor is xi and the Coupling Degree measures the similarity between two data points. This algorithm first initialize a candidate set (CS) with all data points and a Representative set (RS) as an empty set then it computes the RNNs of each data point in CS, and ranks points according to the number of their RNNs in a descending order. Then it selects the data point in the head of this ranked list as a center candidate, and deletes the selected point and its RNNs from the RNN list and (CS) and adds that data point to (RS). While list is not empty, the process of selection, deletion and recalculating of RNN of data points in (CS) will be continued. At the rest of the algorithm it computes number of neighbors in a specific radius and computes Coupling Degree of data points and adds mean of each neighborhood to centriod set.

Mohammad F. Eltibi, Wesam M. Ashour [9] proposed an algorithm which uses statistical information from dataset. It uses Central Limit Theory (CLT) which asserts that for large datasets, the distribution of dataset is approximately normal with mean μ and variance $\frac{1}{n}\sigma^2$ .This algorithm also uses Maximum Likelihood Estimator (MLE) and Gaussian distribution. The algorithm computes a radius and k-1 angular coordinates and chooses k data points with these coordinates and radius as initial cluster centers.

Celebi et al.[10] have compared computational efficiency of eight commonly used linear time complexity initialization methods. Their statistical analyses revealed that popular initialization methods such as **forgy**,**macqueen**, and **maximin** often perform poorly .

Liang Bai et al.[11] have applied local geometrical information to describe approximately the set of objects represented by a candidate cluster center then propose an acceleration mechanism for the production of new cluster centers for fast global k-means (FGKM) clustering algorithm.

At the rest of this paper we will introduce a refinement based on Erisoglu et al. [12] proposed algorithm to find initial cluster centers and apply the algorithm to artificial datasets and real world datasets.

3. Proposed algorithm

In this section, the proposed algorithm for computing initial cluster centers for optimizing k-means algorithm is explained. The main idea is taken from Erisoglu et al. article .Their algorithm is based on choosing two of the p variables that describes the change in the dataset better and make a data space of two axes. Firstly absolute value of the variation coefficient in equation (3) is considered for determination of the main axis,

$$Cv_j = \left|\frac{sd(x_j)}{\bar{x}_j}\right| \quad j = 1,2,..,p \tag{3}$$

Where $sd(x_j)$ is the standard deviation and $\bar{x}_j$ is the mean of the j th variable respectively. The main axis is selected as the variable which has maximum value of the coefficient of variation. When coefficient of variation is used to determine the main axis, this is eliminating the problems of the size of terms and differences in measurement units. After determining the main axis, the correlation coefficient is used to determine the second axis. The correlation coefficient between selected variable for main axis and the other variables are compute using equation (4).

$$r_{jj'} = \frac{\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)(x_{ij'}-\bar{x}_{j'})}{\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)^2 \sum_{i=1}^{n}(x_{ij'}-\bar{x}_{j'})^2} \quad j,j' = 1,2,..,p \tag{4}$$

Second axis is determined by the minimum absolute value of the correlations among the main axis variable and the other variables we just examine numeric attributes. Using equation (3) chooses an attribute that has the greatest spread in the data and equation (4) chooses the attribute that has minimum correlation with the main axis. Here, the purpose of election these two axes, spread of the data should be provided the best explain in this tow dimension feature space. Selection of two axes is not a problem for the large data in terms of number of features and number of patterns. However, more than two explanatory variables may need to be selected when the number of clusters is increased. Erisoglu et al. have used this tow dimension feature space for finding all initial cluster centers but we just use it to the find first and second centers. After determining two axes for the proposed algorithm, the mean of data points is calculated as the center of the data set according to selected axis

$$m=[\bar{x}_I \ \bar{x}_{II}] \tag{5}$$

Where $\bar{x}_I$ is mean according to variable of selected main axis and $\bar{x}_{II}$ is defined similarly. After finding the mean of data points according to selected axis, we calculate Normalized Euclidean Distances of each data point to all other data points to be able to find a radius ε and calculate the number of neighbors of each data point in the radius. Finding a suitable radius for neighborhood computing is an important challenge in most of data mining algorithms .Because when this radius is too big or too small it will be useless and also this radius depends on density of clusters data points and may be different for each cluster. In our algorithm we need the number of neighbors to prevent choose of noises and data points in external borders of cluster to become chosen as cluster centers.

We compute Normalized Euclidean Distance by equation (6) which divides subtraction of each pair of numeric attributers of tow data point with the maximum value of that attribute in whole dataset to decrease the effect of attributes which has big values. At the beginning of our algorithm we compute distance between all data points and use it in the rest of algorithm.

$$d(x,y) = \sqrt[2]{\sum_{i=1}^{p} \left(\frac{[\![x_i - y_i]\!]}{\max \quad x_i}\right)^2} \tag{6}$$

After computing the normalized distance between data points, we calculate ε as the median of distance of each data point and its nearest neighbor with equation (7).

$$\varepsilon = median\left(\left\{foreach\ x_i \in S\ \middle|\ \min\{\mathrm{d}(\mathrm{x_i}, \mathrm{y_j}), y_j \in S\}\right\}\right) i,j = 1,2,..,p\ , S = DataSet \tag{7}$$

To preventing from candidate of noise data points, we have used a threshold value which is summation of standard deviation of nearest neighbor's distance of data points $sd(NNDis)$ and average of nearest neighbor's distance $Avg(NNdis)$. Those data points that have a distance more than this threshold value won't be selected as cluster centers. $IsCandidate(x_i)$ Threshold function determines x_i is able to be candidate as cluster center or not.

$$NNDis(x_i) = \{foreach\ x_i \in S\ |\min\ \{\mathrm{d}(\mathrm{x_i}, \mathrm{y_j}), y_j \in S\}\}\ \mathrm{j{=}1,2,\ \ldots,n,}\ S\mathrm{{=}DataSet} \tag{8}$$

$$IsCandidate(x_i) = \begin{cases} 0 & if\ NNDis(x_i) > sd(NNDis) + Avg(NNdis) \\ 1 & if\ NNDis(x_i) \leq sd(NNDis) + Avg(NNdis) \end{cases} \tag{9}$$

Now we calculate the number of neighbors of each data point $|N(x_i)|$ according to ε with equation (10).

$$N(x_i) = \{y_j | y_j \in S\ , d(x_i, y_j) \leq \varepsilon\} \tag{10}$$

The radius is suitable just to give an overview of data space and give more chance to data points in dense areas to be chosen and less chance to noise data and data points in external borders of clusters. If you compare Fig.1, Fig.2 and Fig.3 they show advantage of use of number of neighbors$|N(x_i)|$.

Erisoglu et al. algorithm does not use number of neighbors for determination of cluster centers so as you can see in Fig.2 and Fig.3 it causes the choice of noise data and data points in borders.

To choose the first initial cluster center, we calculate the Euclidean distances of data points and mean point of data space by equation (11).

$$d(x_i , m)=\sqrt{(x_{iI}-\bar{x}_I)^2+(x_{iII}-\bar{x}_{II})^2} \quad i=1,2, \dots ,n \tag{11}$$

And choose the farthest data point with bigger number of neighbors as $c_1$whit equation (12).

$$C_1=\max \left\{|N(x_i)| * \sqrt{(x_{iI}-\bar{x}_I)^2+(x_{iII}-\bar{x}_{II})^2}\right\} \quad i=1,2, \dots ,n \tag{12}$$

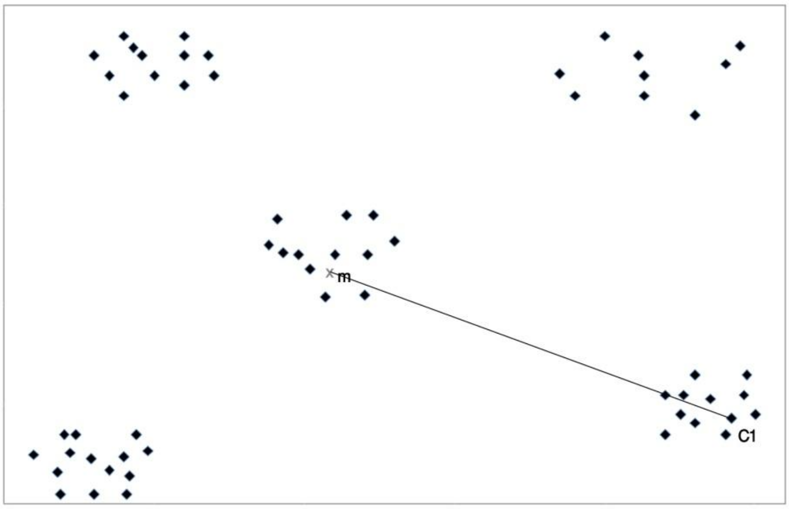

Fig.1.Proposed algorithm tries to choose initial cluster centers from dense areas of data space

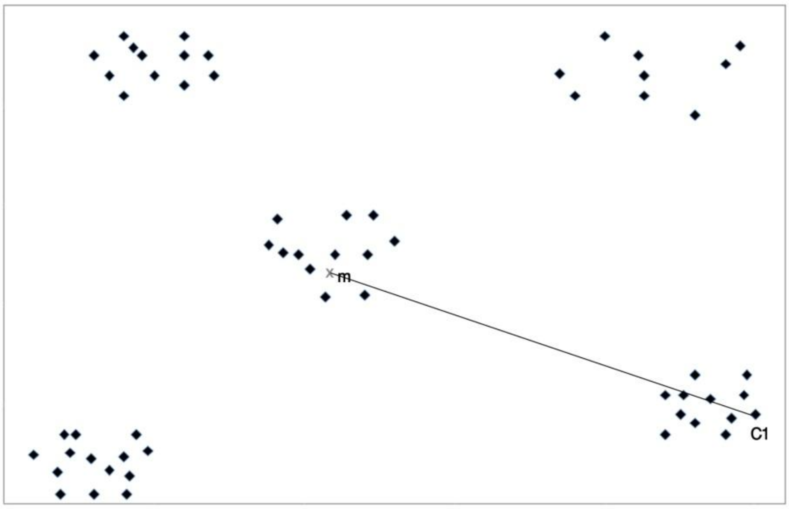

Fig.2.Erisoglu et al. algorithm chooses initial cluster centers from borders

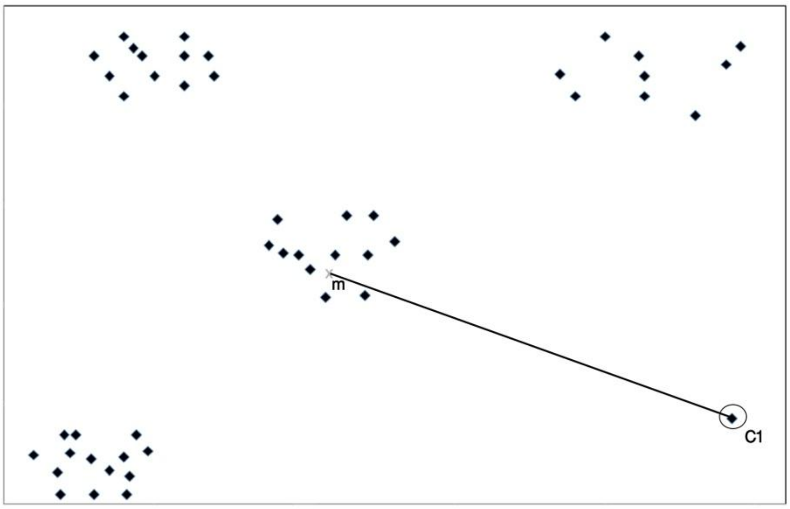

Fig.3.Erisoglu et al. algorithm chooses noise data points as initial cluster centers

Fig.1 shows that $|N(x_i)|$ Helps our algorithm to prevent from choosing noises and data points in borders of clusters. Although it won't choose a point in center of a cluster as initial cluster center, at least it will choose an initial cluster center from a dense area of data space.

To select a candidate for the second initial cluster center, the same mechanism is applied using d(x_i,c_1) instead of d(x_i,m). The data point with maximum distance of d(x_i ,c_1) and bigger $|N(x_i)|$ will be selected as the second initial cluster center. Equation (13) is used to choose c_2.

$$C_2=\max \left\{|N(x_i)| * \sqrt{(x_{iI}-C_{1I})^2+(x_{iII}-c_{1II})^2}\right\} \quad i=1,2,\ldots,n \tag{13}$$

Fig.4 shows how our algorithm choose second cluster center.

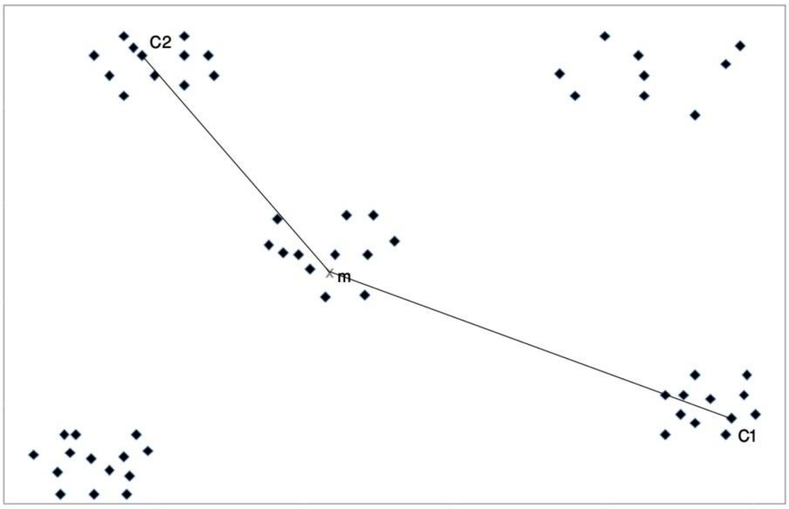

Fig.4.choosing second cluster center

To select c_3 to c_k candidates of initial cluster centers, we use a loop r=3 to r=k (where r is the current iteration step) and use Normalized Euclidean Distances between each data point and C_{r-1} .The Mdis(x_i) is used as multiplication of distances between x_i and previous initial cluster centers and helps us to choose a data point which is far enough from other cluster centers. At the beginning of our loop we initialize Mdis(x_i) as bellow

$$\mathrm{Mdis}(x_i)_0 = \begin{cases} 0 & \text{if } x_i \in C \\ |Nx(i)| & \text{if } x_i \notin C \end{cases} \tag{14}$$

This initialization helps our algorithm not to choose previous initial cluster centers a gain and also choose data point which is far enough from other centers and also belongs to a dense area of data space. (Note that we will change $\mathrm{Mdis}(\ c_r\ )$=0 in each iteration).

As you remember we calculated Normalized Euclidean distance between all data points, so we calculate average of distance of each data point and other data points as $DisAvg(x_i)$ which help our algorithm to give less chance to noise data points to be chosen.

$$DisAvg(x_i) = \sum_{j=1}^{n} \sqrt[2]{\sum_{k=1}^{p} \left(\frac{[\![x_{ik} - x_{jk}]\!]}{\max x_k} \right)^2} \Big/ (n-1) \tag{15}$$

We recomputed the $\mathrm{Mdis}(\ x_i\ )$ as bellow

$$\mathrm{Mdis}(\ x_i\ )_r = \mathrm{Mdis}(\ x_i\ )_{r-1} * d(x_i, c_{r-1}) / DisAvg(x_i) \tag{16}$$

And choose data point with maximum of $\mathrm{Mdis}(\ x_i\ )$ as next initial cluster center

$$c_r = \max\ \ \{\mathrm{Mdis}(\ x_i\ )_r\}\ \ i = 1,2,..,n \tag{17}$$

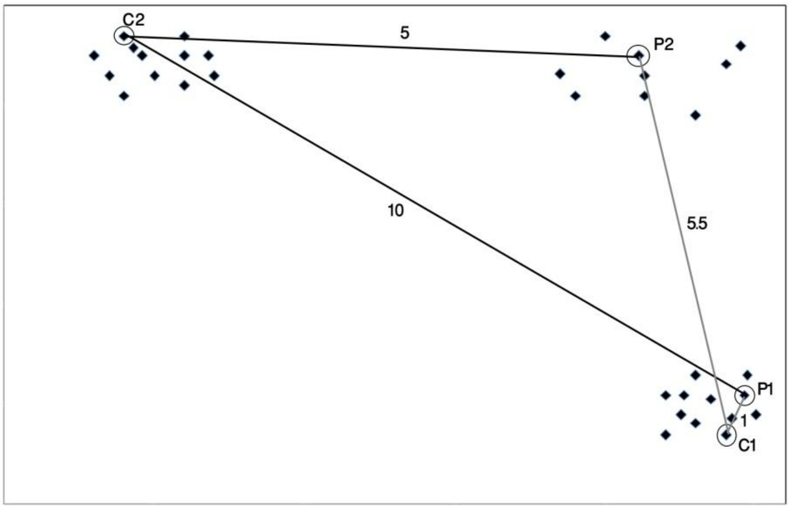

Fig.5.Choosing third initial cluster center

The process will be repeated until the number of initial cluster centers equals to the predefined number of clusters. Then, cluster membership of each point is determined according to candidate initial cluster centers.

Erisoglu et al. have used an accumulator for each data point which sums the distance of it with previous initial cluster centers and choose the data point which has the maximum value as the next cluster center in any iteration. We have used multiplication instead of it.Fig.5 shows an example which we want to choose third initial cluster center Erisoglu et al. algorithm sums distance of p_1 and p_2 with c_1 and c_2 which this sum for p_1 is 10+1=11 and for p_2 is 5.5+5=9.5 .So Erisoglu et al. algorithm choose p_1 as the third center but if we use multiplication (10*1=10, 5.5*5=27.5) so p_2 will be selected which is better than p_1. Table 1 and Fig.6 show Pseudo code and Flowchart of proposed algorithm.

Table 1.Pseudo code of proposed algorithm

```
Input: dataset, number of clusters (k).
Output: initial cluster centers.

step1.   For each attribute Compute $Cv_j = \left|\frac{sd(x_j)}{\overline{x_j}}\right|$
step2.   Set max $\left\{\left|\frac{sd(x_j)}{\overline{x_j}}\right|\right\}$ as main axis
step3.   Compute $r_{jj'} = \frac{\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)(x_{ij'}-\bar{x}_{j'})}{\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)^2 \sum_{i=1}^{n}(x_{ij'}-\bar{x}_{j'})^2}$ between main axis and the other attributes
step4.   Set min $\left\{\frac{\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)(x_{ij'}-\bar{x}_{j'})}{\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)^2 \sum_{i=1}^{n}(x_{ij'}-\bar{x}_{j'})^2}\right\}$ as second axis.
step5.   Compute the mean of data points
step6.   For each data point Compute $d(x,y) = \sqrt[2]{\sum_{i=1}^{p}\left(\frac{[\![x_i-y_i]\!]}{\max \quad x_i}\right)^2}$
step7.   Compute $\varepsilon = median(\{foreach\ \ x_i \in S \mid \min\ \{\mathrm{d}(\mathrm{x_i}, \mathrm{y_j}), y_j \in S\}\})$
step8.   For each data point Compute $N(x_i) = \{y_j \mid y_j \in S, d(x_i, y_j) \leq \varepsilon\}$
step9.   For each data point Compute d(x$_i$ , m)=$\sqrt{(x_{iI}-\bar{x}_I)^2+(x_{iII}-\bar{x}_{II})^2}$
step10.  C$_1$=max $\left\{|N(x_i)| * \sqrt{(x_{iI}-\bar{x}_I)^2+(x_{iII}-\bar{x}_{II})^2}\right\}$
step11.  C$_2$=max $\left\{|N(x_i)| * \sqrt{(x_{iI}-C_{1I})^2+(x_{iII}-c_{1II})^2}\right\}$
step12.  For each data point  Set Mdis( x$_i$ )$_0$ $= \begin{cases} 0 & \text{if } \mathrm{x_i} \in \mathrm{C} \\ |Nx(i)| & \text{if } \mathrm{x_i} \notin \mathrm{C} \end{cases}$
step13.  For each data point Compute $DisAvg(x_i) = \sum_{j=1}^{n} \sqrt[2]{\sum_{k=1}^{m}\left(\frac{[\![x_{ik}-x_{jk}]\!]}{\max \quad x_k}\right)^2} \Big/ (n-1)$
step14.  For r=3 To k do
             ○ Mdis( x$_i$ )$_r$=Mdis( x$_i$ )$_{r-1}$ $* d(x_i, c_{r-1})/DisAvg(x_i)$
             ○ C$_r$=max  {Mdis( x$_i$ )$_r$}
             ○ Mdis( C$_r$)=0
step15.  Finish.
```

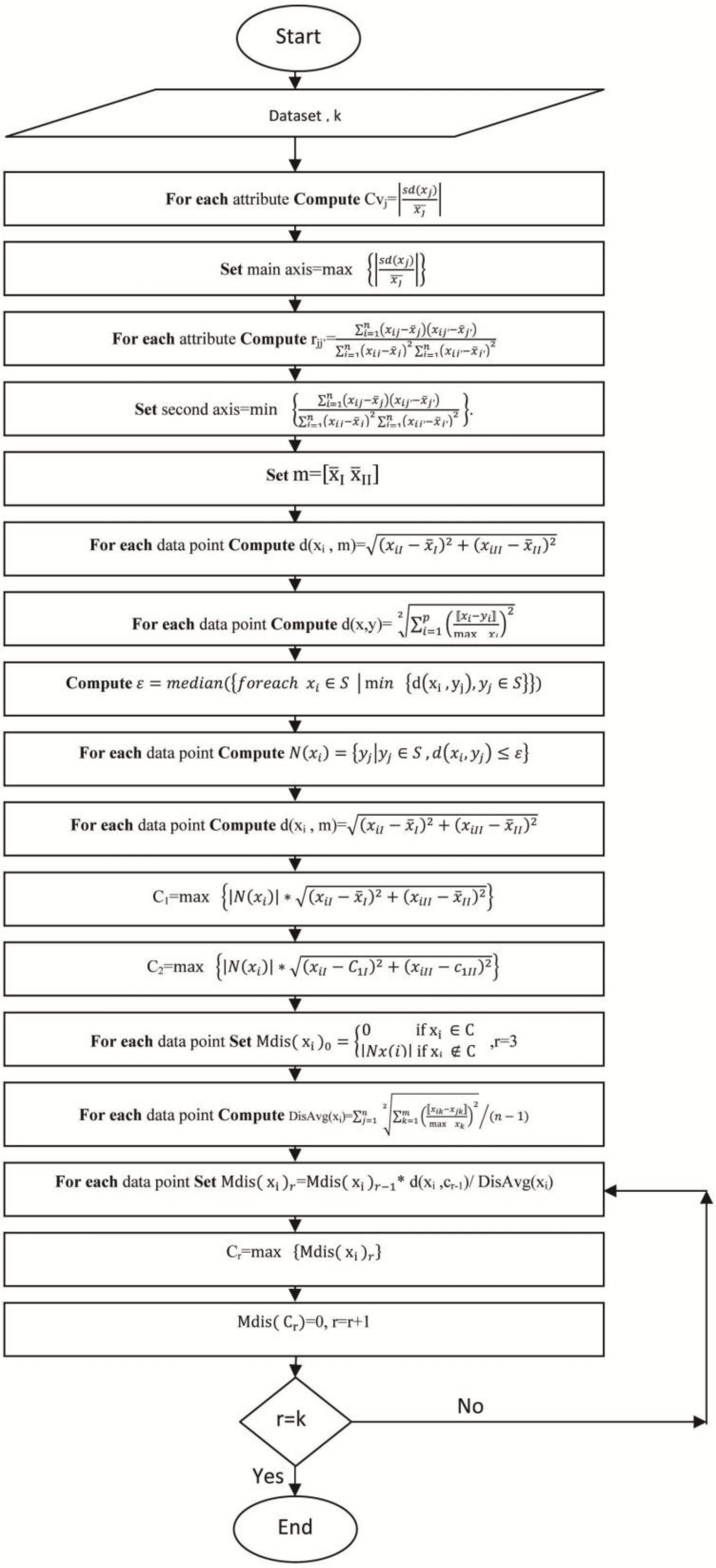
Start
Dataset , k
For each attribute **Compute** $Cv_j=\left|\frac{sd(x_j)}{\overline{x_J}}\right|$
Set main axis=max $\left\{\left|\frac{sd(x_j)}{\overline{x_J}}\right|\right\}$
For each attribute **Compute** $r_{jj'}=\frac{\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)(x_{ij'}-\bar{x}_{j'})}{\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)^2\sum_{i=1}^{n}(x_{ij'}-\bar{x}_{j'})^2}$
Set second axis=min $\left\{\frac{\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)(x_{ij'}-\bar{x}_{j'})}{\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)^2\sum_{i=1}^{n}(x_{ij'}-\bar{x}_{j'})^2}\right\}$.
Set $m=[\bar{x}_I\ \bar{x}_{II}]$
For each data point **Compute** $d(x_i, m)=\sqrt{(x_{iI}-\bar{x}_I)^2+(x_{iII}-\bar{x}_{II})^2}$
For each data point **Compute** $d(x,y)=\sqrt[2]{\sum_{i=1}^{p}\left(\frac{[\![x_i-y_i]\!]}{\max\ x_i}\right)^2}$
Compute $\varepsilon = median(\{foreach\ x_i \in S\ |\min\ \{d(x_i, y_j), y_j \in S\}\})$
For each data point **Compute** $N(x_i)=\{y_j|y_j \in S, d(x_i,y_j) \le \varepsilon\}$
For each data point **Compute** $d(x_i, m)=\sqrt{(x_{iI}-\bar{x}_I)^2+(x_{iII}-\bar{x}_{II})^2}$
$C_1=\max\ \left\{|N(x_i)| * \sqrt{(x_{iI}-\bar{x}_I)^2+(x_{iII}-\bar{x}_{II})^2}\right\}$
$C_2=\max\ \left\{|N(x_i)| * \sqrt{(x_{iI}-C_{1I})^2+(x_{iII}-c_{1II})^2}\right\}$
For each data point **Set** $Mdis(x_i)_0=\begin{cases}0 & \text{if } x_i \in C\\ |Nx(i)| & \text{if } x_i \notin C\end{cases}$,r=3
For each data point **Compute** $DisAvg(x_i)=\sum_{j=1}^{n}\sqrt[2]{\sum_{k=1}^{m}\left(\frac{[\![x_{ik}-x_{jk}]\!]}{\max\ x_k}\right)^2}/(n-1)$
For each data point **Set** $Mdis(x_i)_r=Mdis(x_i)_{r-1} * d(x_i, c_{r-1})/DisAvg(x_i)$
$C_r=\max\ \{Mdis(x_i)_r\}$
$Mdis(C_r)=0$, r=r+1
r=k
No
Yes
End

Fig. 6.Flowchart of proposed algorithm

4. Experimental results

We implemented Erisoglu et al. and the proposed algorithms with java language in Netbeans IDE 7.1 environment and used weka.jar library of weka 3.6.6 software which contains necessary classes for implementing a new clusterer .After we implemented and tested the algorithm, we used code of simple k-means of weka 3.6.6 and changed it to use initial cluster centers of our algorithm instead of random initialization of cluster centers for it. So except choosing initial cluster centers all other steps will be done according to standard k-means algorithm.

Our experimental result consists of two steps. At first we examine algorithm with artificial data sets. We made a data set with 60 instances which has five spherical clusters with two numeric attributes and used it to find initial cluster centers.Fig.7 shows the initial cluster centers found with Erisoglu et al. algorithm and Fig.8 shows the initial cluster centers found with proposed algorithm. As you can see Erisoglu et al. algorithm introduces two instances from same cluster as initial cluster centers.

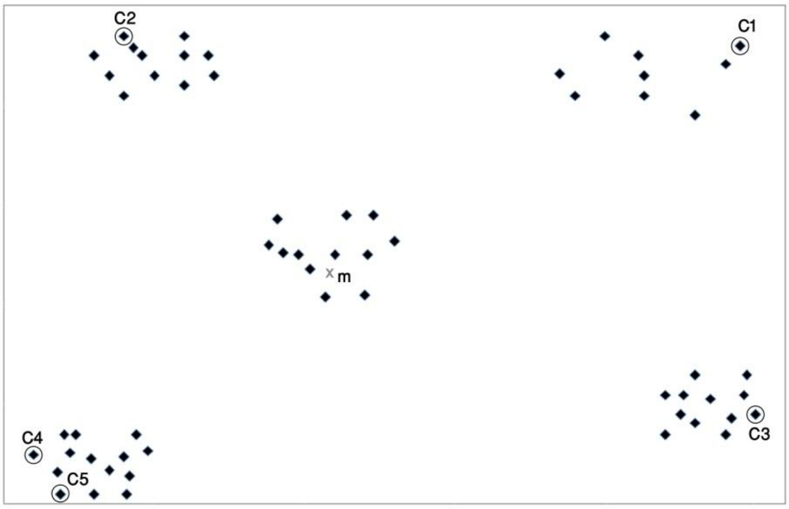

Fig.7.Computed initial cluster centers with Erisoglu et al. algorithm for an artificial dataset with 5 clusters

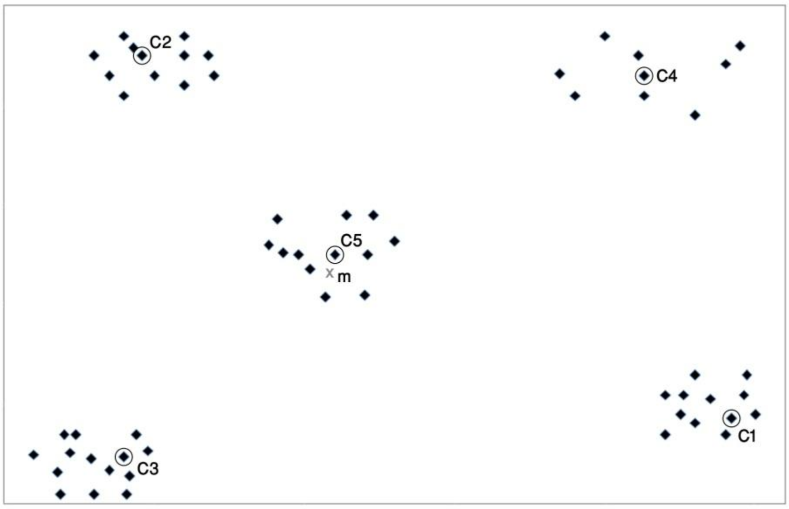

Fig.8.Computed initial cluster centers with proposed algorithm for an artificial dataset with 5 clusters

As you can see Erisoglu et al. algorithm introduces two instances from same cluster as initial cluster centers.

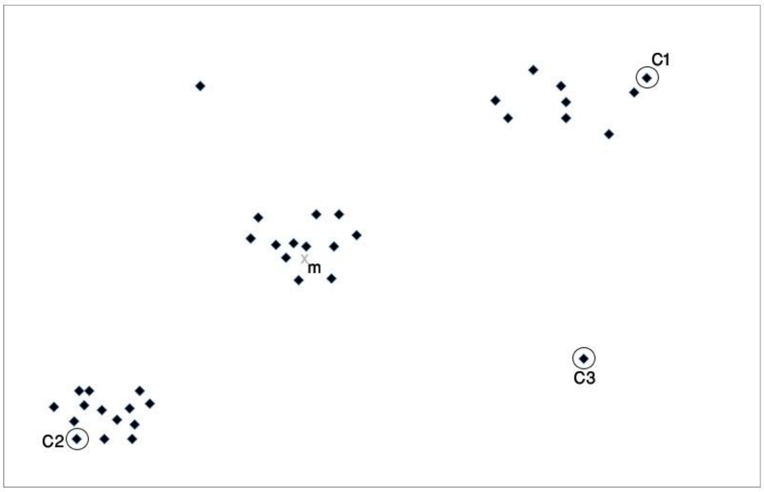

Fig.9.Computed initial cluster centers with Erisoglu et al. algorithm for an artificial dataset with noise

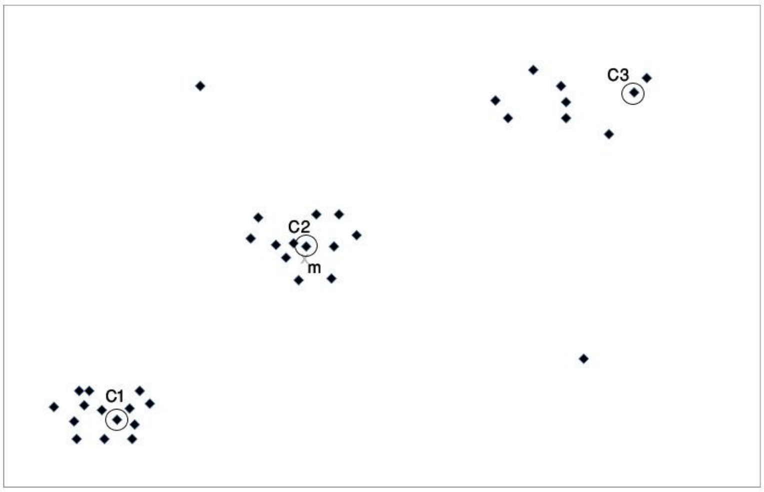

Fig.10.Computed initial cluster centers with proposed algorithm for an artificial dataset with noise

The other dataset has 3 clusters with tow noise instances. Fig.9 shows that Erisoglu et al. algorithm choose one of noises as initial cluster center and Fig.10 shows changes we have made in Erisoglu et al. algorithm prevent from choosing these kinds of noises.

At the second step we have tested and compared standard k-means clustering error percentage using some famous data sets [13] and initial cluster centers of Erisoglu et al. algorithm, proposed algorithm and random initialization.

We used initial cluster centers gained with both algorithms as initial cluster centers of simple k-means and clustered datasets without any iteration and computed error percentage. Table 2 and Fig.11 show the results.

Table 2.Error percentage of k-means using initial cluster centers of two algorithms without any iterations

Data set	Initialization with Erisoglu et al. algorithm	Initialization with proposed algorithm
Iris	26.00%	8.00%
Wine	33.15%	7.86%
Hepatitis	37.42%	29.68%
Glass	64.95%	49.00%
Vehicle	63.36%	55.55%
Autos	62.43%	58.54%

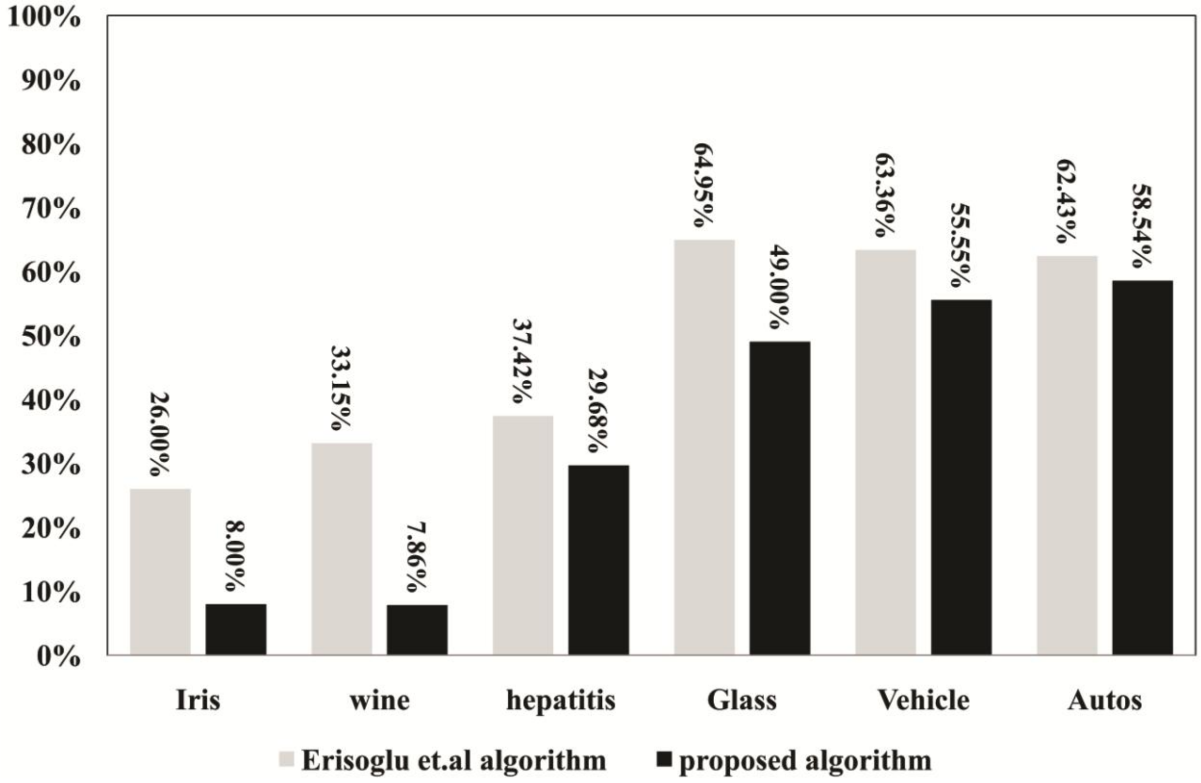

Fig.11.Error percentage of k-means using initial cluster centers of two algorithms without any iterations

We also did the above experience with an iteration of simple k-means algorithm and computed the error percentage to compare both algorithms with random initialization of k-means. Table 3 and Fig.12 show the results.

Table 3.Error percentage of k-means using initial cluster centers of two algorithms and random initialization with an iteration.

Data set	Initialization with Erisoglu et.al algorithm	Initialization with proposed algorithm	Initialization with simple k-means
Iris	11.37%	4.00%	16.67%
wine	33.39%	5.62%	32.00%
hepatitis	38.70%	29.68%	43.22%
Glass	64.95%	47.66%	58.58%
Vehicle	63.36%	58.75%	63.36%
Autos	62.43%	60.00%	60.00%

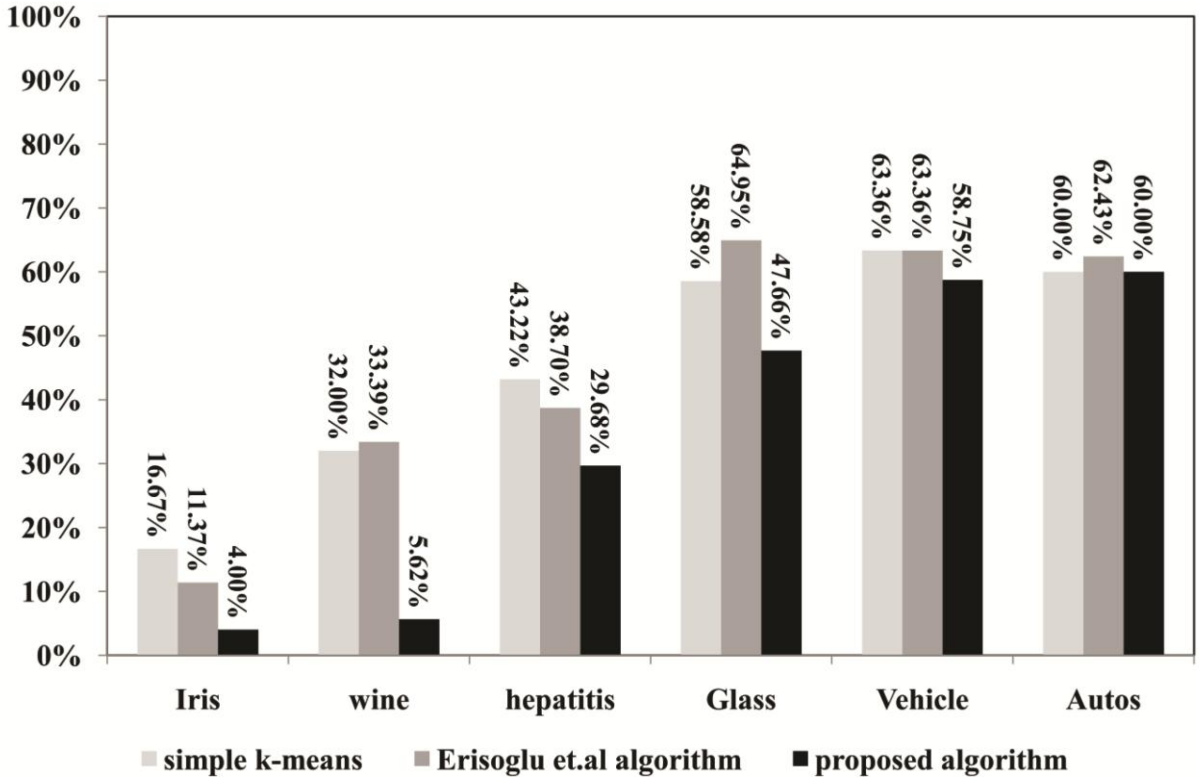

Fig.12.Error percentage of k-means using initial cluster centers of two algorithms and random initialization with an iteration

5. Conclusions

Our experiments show that proposed algorithm is effective and computes suitable initial cluster centers and can help k-means algorithm to find better solutions and converges in less iterations in data sets that made up of spherical clusters without overlap and also when we have different attributes with different ranges of values which attributes with bigger value will have more effect on the result, the normalizing of attributes or using Normalized Euclidean Distance can help to improve the result. As you know, k-means can just find spherical shape clusters, so we cannot use k-means algorithm to examine efficiency of initial cluster centers algorithms on data sets containing arbitrary shape clusters and sometimes due to weakness of k-means algorithm in finding new cluster centers and its objective function, better initial cluster centers cannot reduce error percentage of clustering and will increase number of iterations .For instance if we use simple k-means algorithm of weka 3.6.6 software for autos dataset we have 60% error in one iteration execution and after 8 iterations it increases error to 61.46%.

References

[1] Bradley, P. S., and Fayyad, U. M.Refining initial points for k-means clustering. In Proceedings of the Fifteenth International Conference on Machine learning (ICML'98) (1998), pp. 91–99.

[2] Likas, A., Vlassis, N. A., and Verbeek, J. J.The global k-means clustering algorithm. Pattern Recognition 36, 2 (2003), 451–461.

[3] Khan, S. S., and Ahmad, A. Cluster center initialization algorithm for k-means clustering. Pattern Recognition Letters 25, 11 (2004), 1293–1302.

[4] Deelers, S., and Auwatanamongkol, S. k-means algorithm with initial cluster centers derived from data partitioning along the data axis with the highest variance. International Journal of Computer Science 2 (2007), 247–252.

[5] Arai, K., and Barakbah, A. R. Hierarchical k-means: an algorithm for centroids initialization for k-means.Reports of the Faculty of Science and Engineering Saga University 36, 1 (2007), 25–31.

[6] Cao, F., Liang, J., and Jiang, G. An initialization method for the k-means algorithm using neighborhood model. Computers & Mathematics with Applications 58, 3 (2009), 474–483.

[7] Pawlak, Z. Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[8] Ahmed, A. H., and Ashour, W. An initialization method for the k-means algorithm using rnn and coupling degree. International Journal of Computer Applications 25, 1 (July 2011), 1–6. Published by Foundation of Computer Science, New York, USA.

[9] Eltibi, M. F., and Ashour, W. M. Initializing k-means clustering algorithm using statistical information. International Journal of Computer Applications 29, 7 (September 2011), 51–55. Published by Foundation of Computer Science,New York, USA.

[10] Celebi, M. E., Kingravi, H. A., and Vela, P. A. A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Systems with Applications 40, 1 (2013), 200 – 210.

[11] Liang Bai , Jiye Liang , Chao Sui , Chuangyin Dang ,Fast global k-means clustering based on local geometrical information . Information Sciences 245,(2013), 168 – 180.

[12] Erisoglu, M., Calis, N., and Sakallioglu, S. A new algorithm for initial cluster centers in k-means algorithm. Pattern Recogn. Lett. 32, 14 (oct, 2011), 1701–1705.

[13] http://repository.seasr.org/Datasets/UCI/arff/

[14] R. Maghsoudi , A. Ghorbannia Delavar, S. Hoseyny, R. Asgari, Y. Heidari, Representing the New Model for Improving K-Means Clustering Algorithm based on Genetic Algorithm, The Journal of Mathematics and Computer Science Vol .2 No.2 (2011) 329-336.