ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 11, November 2013

Statistically Refining the Initial Points for K-Means Clustering Algorithm

Kamaljit Kaur[1], Dr. Dalvinder Singh Dhaliwal[2], Dr. Ravinder Kumar Vohra[3]

Abstract— **In today's world, an organization generates more information in a week than most people can read in a lifetime. The amount of raw data stored in databases is exploding. Cluster analysis is one of the major data mining methods and the k-means clustering algorithm is widely used for many applications. K-means algorithm is computationally expensive and the quality of the resulting clusters depends on the choice of initial centroids. This paper proposes an improvement on the classic k-means algorithm to produce more accurate clusters. The proposed algorithm comprises of method, based on sorting and partitioning the input data, for finding the initial centroids in accordance with the data distribution. Experimental results show that the proposed algorithm produces better clusters in less computation time.**

Index Terms—**Clustering, Data Mining, Initial Centroid, K-Mean, Median.**

I. Introduction

In modern times, the amount of data been stored in the databases is measured in gigabytes and terabytes. It is not possible to manually interpret and find useful information from all the data. Data Mining is a knowledge discovery process that helps to unearth patterns and trends in the raw data [3]. Data from various sources is integrated into a single data store which is pre-processed and transformed to a standard format. The data is processed and interpreted to useful knowledge or information.

Data mining centers on the automated discovery of new facts and relationships in data. Data mining tools predict behaviors and future trends, allowing businesses to make knowledge-driven decisions. Data Mining is used to discover patterns and relationships in data in order to help make better business decisions [4]. Data Mining can help to predict sale trends, develop marketing strategies, aids in Market Segmentation to identify common customer characteristics, Customer Churn, Fraud Detection, Market Basket Analysis that helps in identifying products purchased together and Trend Analysis [7].

II. CLUSTERING

Clustering is fundamental task in Data Mining. Clustering is the process of partitioning or grouping a given set of data into disjoint clusters. This is done such that data in the same cluster are similar and patterns belonging to two different clusters are different. The k-means method has been shown to be effective in producing good clustering results for many practical applications. This algorithm partitions the data into K clusters (C_1; C_2; ; C_K), represented by their centers is calculated as the mean of all the instances belonging to that cluster [3]. However this method requires time proportional to the product of number of patterns and number of clusters per iteration. This is computationally very expensive for large datasets. Also, the clustering results for same data set can be different with different initial centroids chosen randomly.

III. K-Mean CLUSTERING ALGORITHM

The K-means algorithm randomly selects K initial centroids where K is a user defined number of desired clusters. Each point is then assigned to a closest centroid and the collection of points close to a centroid form a cluster. The centroid gets updated according to the points in the cluster and this process continues until the points stop changing their clusters.

```
Initialization (Initialize K centroids)
do
Assignment (Assign each data point to its closest centroid)
Re-Calculation (Recompute centroid)
while (centroid does not change)
```

Figure1: Representation of the K-means algorithm

Kamaljit Kaur, *Research Scholar, Punjab Technical University, Kapurthala., Punjab*

Dr. Dalvinder Singh Dhaliwal, Director, Bharat Institute of Engineering & Technology, Sardulgarh, Punjab,

Dr.Ravinder Kumar Vohra, Professor, Bhai Gurdas Institute of Engineering & Technology, Sangrur, Punjab,

The working of K-Mean Algorithm is described as follows:

Initialization Step (Choosing Initial Centroid)

When initial centroids are chosen randomly the different runs of the algorithm can produce poor results. The choice of the initial centroid has a huge effect on the final result. Clusters produced vary from one run to another. There are different possible solutions to the problem of random selection. The proposed work focuses on the first step of Initializing K centroids.

Assignment Step (Assigning Data to Closest Centroid)

In order to determine which centroid is closest to a particular data point a proximity measure is used. There are several proximity measures and is chosen based on the data type to be clustered. Manhattan, Euclidean, Cosine and Bregman Divergence are used to determine which cluster a point should be assigned to. The Euclidean is commonly used.

Re-estimation Step (Re-Compute Centroids)

Re-calculating the centroid is based on the goal of the clustering algorithm and the proximity measure that is being used. The objective function is defined and the centroid is calculated mathematically. When the objective is to minimize the sum of the squared distance of an object to its cluster centroid the sum of the squared error (SSE) is used. The SSE calculates for each point the distance from that point to the nearest cluster and then squares it and adds up the sum for all points in the dataset.

$$\text{SSE}=\sum_{i=1}^{K}\sum_{x\in C_i} dist^2\ (m_i, x)$$

where x is a data point in cluster C_i and m_i is the point for cluster C_i The K-means algorithm tries to minimize this value for each set of centroids that is given for that iteration.

IV EXISTING METHODS

The earliest method to initialize K-means was proposed by Forgy in 1965 that involves choosing initial centroids randomly from the database. M.E. Celebi et al. revealed that cluster centroid initialization methods such as Forgy, Macqueen, and max-min perform poorly and there are other methods with same computational requirements which can give better results [5]. KKZ method choose point x as the first seed, and then finds a point furthest from x and this point will be the second seed. Then the method calculates the distance of all points in the dataset to the nearest of first and second seed. The process of choosing the furthest point from its nearest seed is repeated until K seeds are chosen [7]. Bradley and Fayyad suggested a new method where data is broken down randomly into 10 subsets. In the second step K-means algorithm is applied on each of the 10 subsets, the initial centroids for these are chosen using Forgy's method. Tou and Gonzales suggested Simple Cluster-Seeking (SCS) method that initializes the first seed with the first value and then calculates the distance between the chosen seed and the next point in the database, if this distance is greater than some threshold then this point is chosen as the second seed, otherwise it will move to the next instance in the database and repeat the process. This process is repeated until K seeds are chosen [6]. Madhu Yedla et al. proposed a simpler algorithm for choosing the initial clusters. The proposed algorithm first checks whether the given data set contain the negative value attributes or not. If the data set contains the negative value attributes then all the data points are transformed to the positive space by subtracting the each data point attribute with the minimum attribute value in the given data set. In the next step, for each data point the distance from the origin is calculated. Then, the original data points are sorted accordance with the sorted distances. After sorting partition the sorted data points into k equal sets. In each set take the middle points as the initial centroids.

V NEW PROPOSED METHOD OF FINDING INTIAL CENTROIDS: *Medkmeans*

The K-Mean chooses the initial centroids randomly which results in different clusters produced from one run to another. As the choice of the initial centroid has a huge effect on the clustering result; a new method of finding initial centroids is proposed below:

Algorithm Medkmeans: Finding the Initial Cluster Centroids

Input: Data items which needs to be clustered and K is user defined variable of number of desired clusters

Output: A set of K initial centroids are calculated; which otherwise in standard K-Mean are selected randomly.
Steps:
1. The data items are sorted and the median of whole data is calculated, this will be the first initial cluster centre.

2. Data is then divided into two parts; with values less than the median value as the first subset of data and with values greater than the median value as the remaining subset of data

3. Median of these two parts is calculated which becomes the second and third cluster centres respectively.

4. This process is repeated until user defined K cluster centres are found.

EXAMPLE:
Tree Hierarchy Adopted By Medkmeans while Finding Initial Centroids is illustrated as follows:

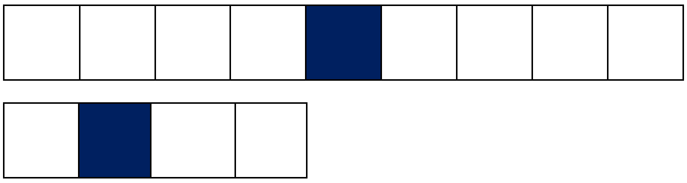


Figure 2: Selection of Initial Centroids (when K=2)

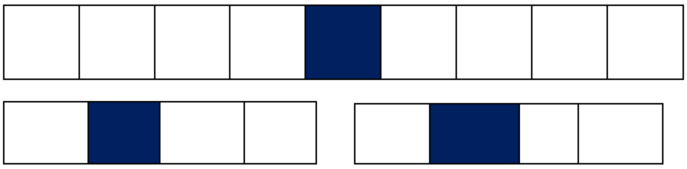


Figure 3: Selection of Initial Centroids (when K=3)

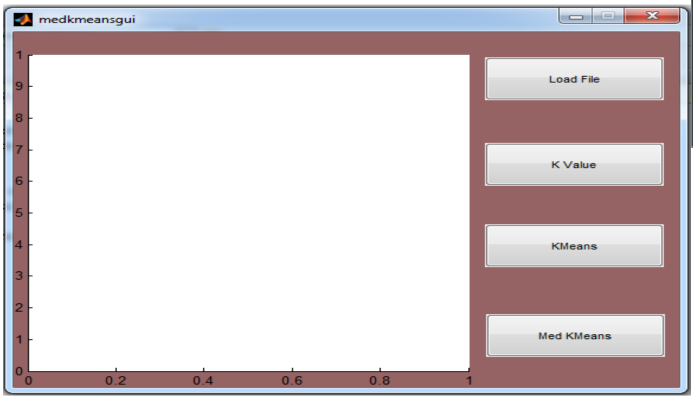
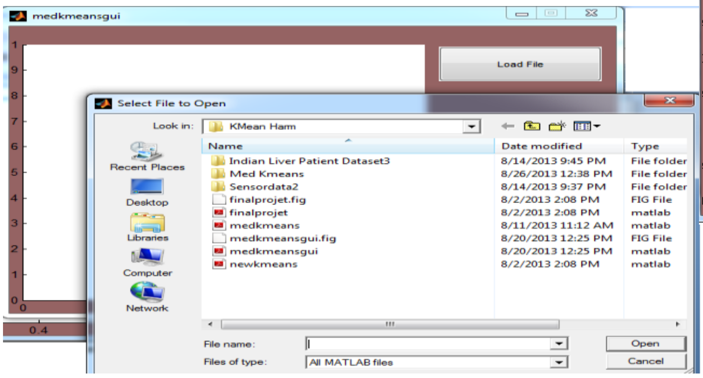


Figure 4: User Interface: Medkmeans

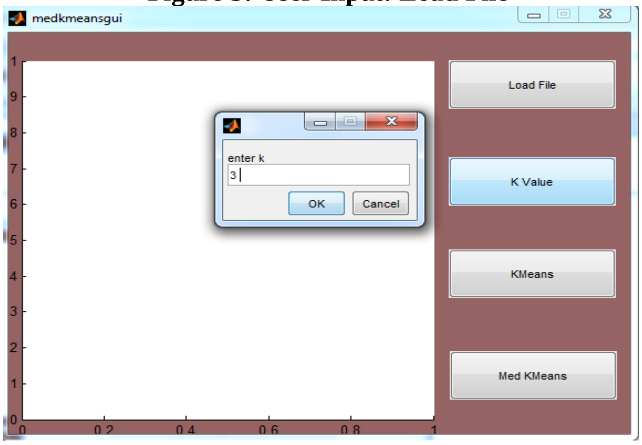


Look in: KMean Ham

Name	Date modified	Type
Indian Liver Patient Dataset3	8/14/2013 9:45 PM	File folder
Med Kmeans	8/26/2013 12:38 PM	File folder
Sensordata2	8/14/2013 9:37 PM	File folder
finalprojet.fig	8/2/2013 2:08 PM	FIG File
finalprojet	8/2/2013 2:08 PM	matlab
medkmeans	8/11/2013 11:12 AM	matlab
medkmeansgui.fig	8/20/2013 12:25 PM	FIG File
medkmeansgui	8/20/2013 12:25 PM	matlab
newkmeans	8/2/2013 2:08 PM	matlab

File name:
Files of type: All MATLAB files

Figure 5: User Input: Load File



enter k
3

Figure 6: User Input: K (Number of Clusters to form)

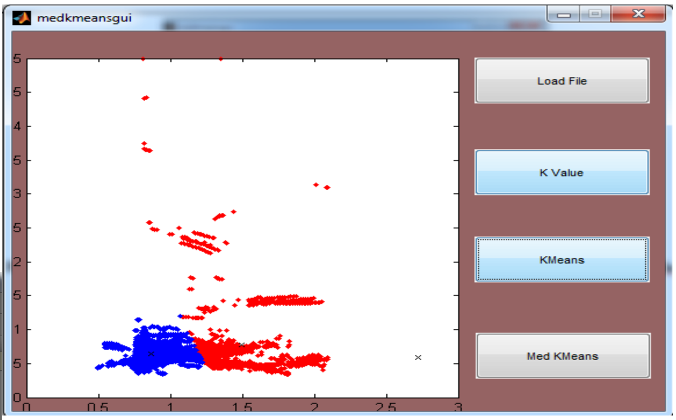


Figure 7: With the given user input (Input File and K-Value): Output using Kmeans

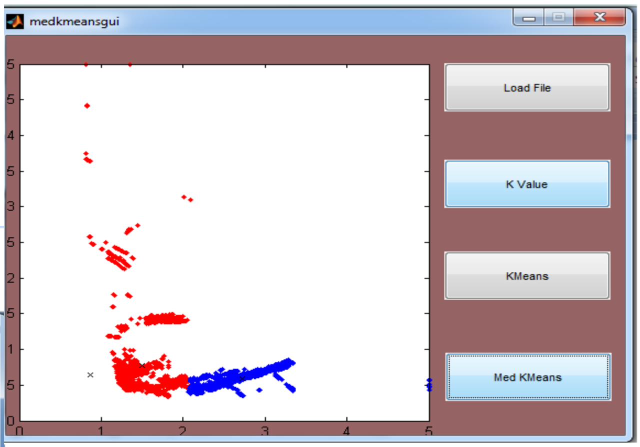
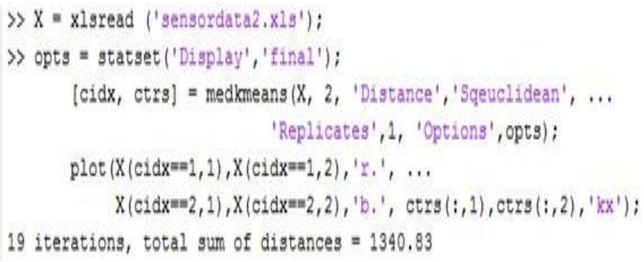


Figure 8: With the given user input (Input File and K-Value): Output using Med KMeans

V RESULTS AND DISCUSSION

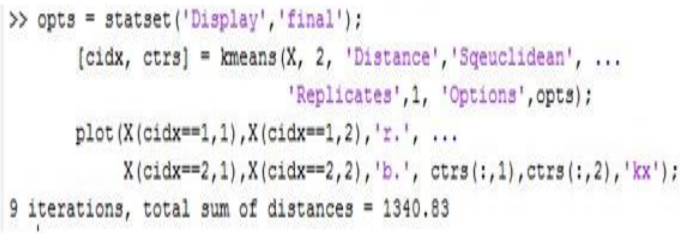
DATA SET: sensordata2 (contains 5456 rows)

COMPARISON of K-Means with New Proposed Median-Based K-Mean (medkmeans) Based on Iterations and Total Sum of Distances



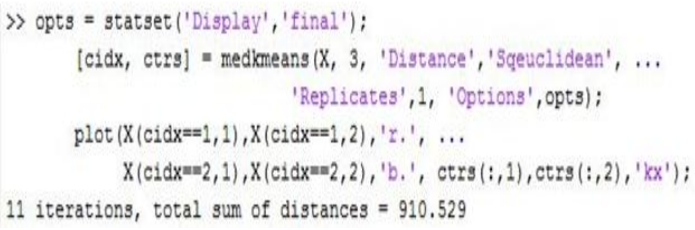
```
>> X = xlsread ('sensordata2.xls');
>> opts = statset('Display','final');
        [cidx, ctrs] = medkmeans(X, 2, 'Distance','Sqeuclidean', ...
                              'Replicates',1, 'Options',opts);
        plot(X(cidx==1,1),X(cidx==1,2),'r.', ...
             X(cidx==2,1),X(cidx==2,2),'b.', ctrs(:,1),ctrs(:,2),'kx');
19 iterations, total sum of distances = 1340.83
```

Snapshot: Medkmeans for K=2



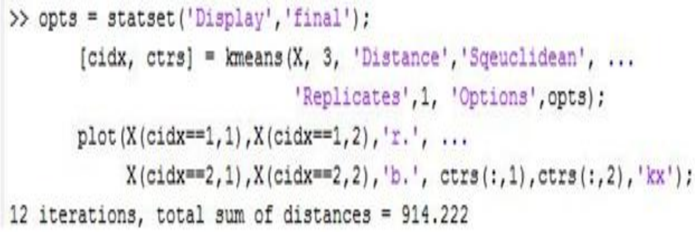
```
>> opts = statset('Display','final');
        [cidx, ctrs] = kmeans(X, 2, 'Distance','Sqeuclidean', ...
                              'Replicates',1, 'Options',opts);
        plot(X(cidx==1,1),X(cidx==1,2),'r.', ...
             X(cidx==2,1),X(cidx==2,2),'b.', ctrs(:,1),ctrs(:,2),'kx');
9 iterations, total sum of distances = 1340.83
```

Snapshot: kmeans for K=2



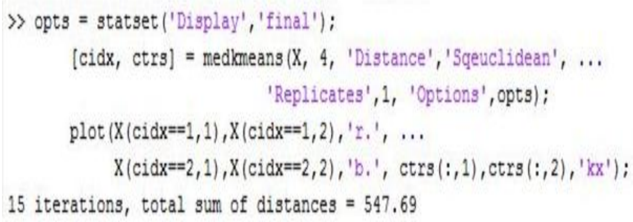
```
>> opts = statset('Display','final');
      [cidx, ctrs] = medkmeans(X, 3, 'Distance','Sqeuclidean', ...
                           'Replicates',1, 'Options',opts);
      plot(X(cidx==1,1),X(cidx==1,2),'r.', ...
           X(cidx==2,1),X(cidx==2,2),'b.', ctrs(:,1),ctrs(:,2),'kx');
11 iterations, total sum of distances = 910.529
```

Snapshot: Medkmeans for K=3



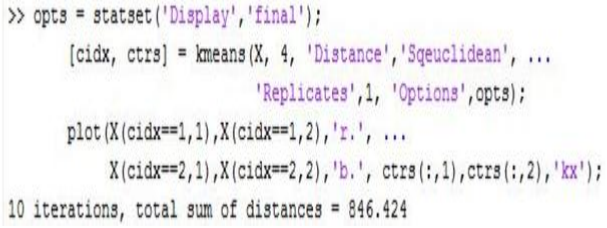
```
>> opts = statset('Display','final');
      [cidx, ctrs] = kmeans(X, 3, 'Distance','Sqeuclidean', ...
                           'Replicates',1, 'Options',opts);
      plot(X(cidx==1,1),X(cidx==1,2),'r.', ...
           X(cidx==2,1),X(cidx==2,2),'b.', ctrs(:,1),ctrs(:,2),'kx');
12 iterations, total sum of distances = 914.222
```

Snapshot: kmeans for K=3



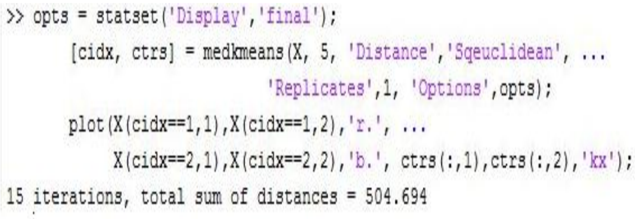
```
>> opts = statset('Display','final');
      [cidx, ctrs] = medkmeans(X, 4, 'Distance','Sqeuclidean', ...
                           'Replicates',1, 'Options',opts);
      plot(X(cidx==1,1),X(cidx==1,2),'r.', ...
           X(cidx==2,1),X(cidx==2,2),'b.', ctrs(:,1),ctrs(:,2),'kx');
15 iterations, total sum of distances = 547.69
```

Snapshot: Medkmeans for K=4



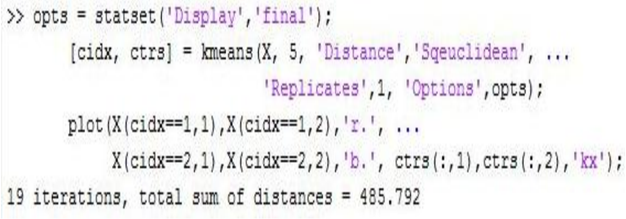
```
>> opts = statset('Display','final');
      [cidx, ctrs] = kmeans(X, 4, 'Distance','Sqeuclidean', ...
                           'Replicates',1, 'Options',opts);
      plot(X(cidx==1,1),X(cidx==1,2),'r.', ...
           X(cidx==2,1),X(cidx==2,2),'b.', ctrs(:,1),ctrs(:,2),'kx');
10 iterations, total sum of distances = 846.424
```

Snapshot: kmeans for K=4



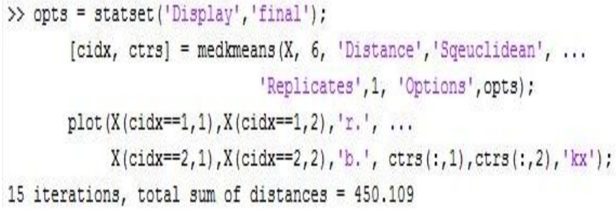
```
>> opts = statset('Display','final');
      [cidx, ctrs] = medkmeans(X, 5, 'Distance','Sqeuclidean', ...
                           'Replicates',1, 'Options',opts);
      plot(X(cidx==1,1),X(cidx==1,2),'r.', ...
           X(cidx==2,1),X(cidx==2,2),'b.', ctrs(:,1),ctrs(:,2),'kx');
15 iterations, total sum of distances = 504.694
```

Snapshot: Medkmeans for K=5



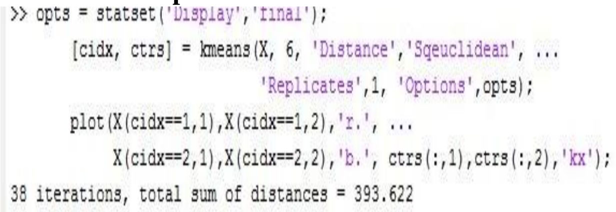
```
>> opts = statset('Display','final');
      [cidx, ctrs] = kmeans(X, 5, 'Distance','Sqeuclidean', ...
                           'Replicates',1, 'Options',opts);
      plot(X(cidx==1,1),X(cidx==1,2),'r.', ...
           X(cidx==2,1),X(cidx==2,2),'b.', ctrs(:,1),ctrs(:,2),'kx');
19 iterations, total sum of distances = 485.792
```

Snapshot: kmeans for K=5



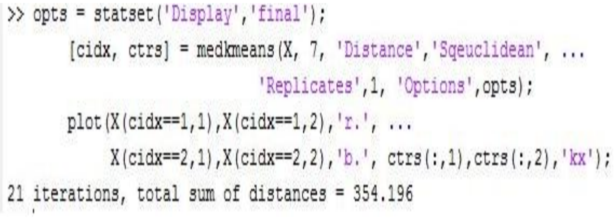
```
>> opts = statset('Display','final');
      [cidx, ctrs] = medkmeans(X, 6, 'Distance','Sqeuclidean', ...
                           'Replicates',1, 'Options',opts);
      plot(X(cidx==1,1),X(cidx==1,2),'r.', ...
           X(cidx==2,1),X(cidx==2,2),'b.', ctrs(:,1),ctrs(:,2),'kx');
15 iterations, total sum of distances = 450.109
```

Snapshot: Medkmeans for K=6



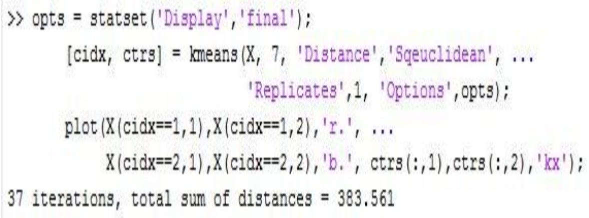
```
>> opts = statset('Display','final');
      [cidx, ctrs] = kmeans(X, 6, 'Distance','Sqeuclidean', ...
                           'Replicates',1, 'Options',opts);
      plot(X(cidx==1,1),X(cidx==1,2),'r.', ...
           X(cidx==2,1),X(cidx==2,2),'b.', ctrs(:,1),ctrs(:,2),'kx');
38 iterations, total sum of distances = 393.622
```

Snapshot: kmeans for K=6



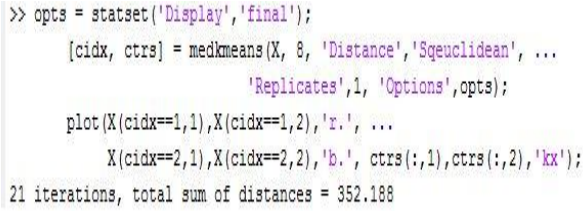
```
>> opts = statset('Display','final');
      [cidx, ctrs] = medkmeans(X, 7, 'Distance','Sqeuclidean', ...
                           'Replicates',1, 'Options',opts);
      plot(X(cidx==1,1),X(cidx==1,2),'r.', ...
           X(cidx==2,1),X(cidx==2,2),'b.', ctrs(:,1),ctrs(:,2),'kx');
21 iterations, total sum of distances = 354.196
```

Snapshot: Medkmeans for K=7



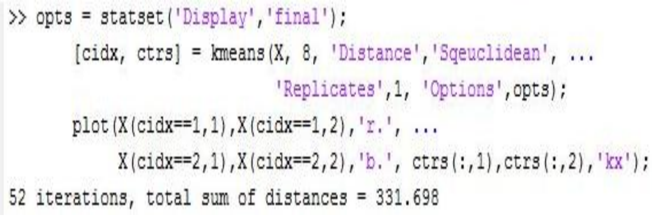
```
>> opts = statset('Display','final');
      [cidx, ctrs] = kmeans(X, 7, 'Distance','Sqeuclidean', ...
                           'Replicates',1, 'Options',opts);
      plot(X(cidx==1,1),X(cidx==1,2),'r.', ...
           X(cidx==2,1),X(cidx==2,2),'b.', ctrs(:,1),ctrs(:,2),'kx');
37 iterations, total sum of distances = 383.561
```

Snapshot: kmeans for K=7



```
>> opts = statset('Display','final');
      [cidx, ctrs] = medkmeans(X, 8, 'Distance','Sqeuclidean', ...
                           'Replicates',1, 'Options',opts);
      plot(X(cidx==1,1),X(cidx==1,2),'r.', ...
           X(cidx==2,1),X(cidx==2,2),'b.', ctrs(:,1),ctrs(:,2),'kx');
21 iterations, total sum of distances = 352.188
```

Snapshot: Medkmeans for K=8



```
>> opts = statset('Display','final');
      [cidx, ctrs] = kmeans(X, 8, 'Distance','Sqeuclidean', ...
                           'Replicates',1, 'Options',opts);
      plot(X(cidx==1,1),X(cidx==1,2),'r.', ...
           X(cidx==2,1),X(cidx==2,2),'b.', ctrs(:,1),ctrs(:,2),'kx');
52 iterations, total sum of distances = 331.698
```

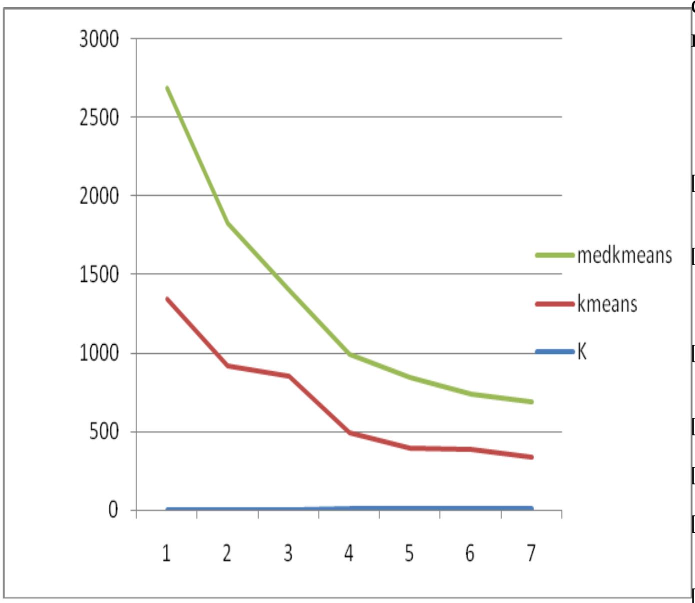
Snapshot: kmeans for K=8

Table: Comparison of Kmeans and Medkmeans Based on Iterations and Total Sum of Distances

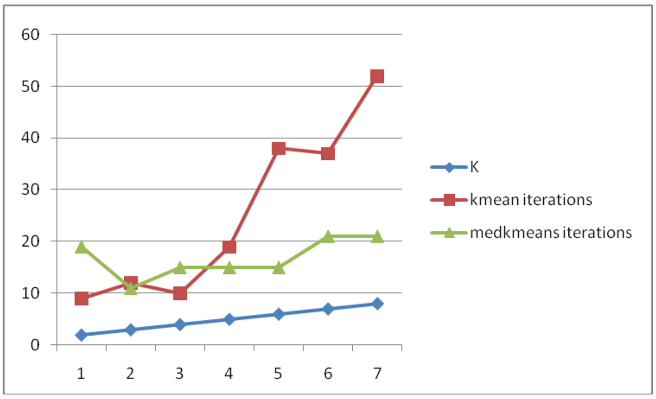
K-Means		
K	**Iterations**	**Total Sum Of Distances**
2	9	1340.83
3	12	914.222
4	10	846.424
5	19	485.792
6	38	393.622
7	37	383.561
8	52	331.698
	Total	4696.149

Medkmeans		
K	**Iterations**	**Total Sum Of Distances**
2	19	1340.83
3	11	910.529
4	15	547.69
5	15	504.694
6	15	450.109
7	21	354.196
8	21	352.188
	Total	4460.236

GRAPH COMPARING kmeans and medkmean BASED ON TOTAL SUM OF DISTANCES



GRAPH COMPARING kmeans and medkmean BASED ON NUMBER OF ITERATIONS



IV. Conclusion

The K-Means Algorithm suffers from the two major limitations of being computationally very expensive as it involves several distance calculations of each data point from all the centroids in each iteration and secondly the final cluster results heavily depends on the selection of initial centroids which causes it to converge at local optimum. The proposed method tends to improve k-Means clustering algorithm in MATLAB with datasets from UCI machine learning repository. This method may not work for some datasets. The new proposed method is used for selection of initial centroids instead of selecting initial centroids randomly. By using new approach we obtained good clustering results. The new method of selection of initial centroid is better than selecting the initial centroids randomly.

REFERENCES

[1] Huang, “Extensions to the k-Means Algorithms for Clustering Large Data Sets with Categorical Values,” Data Ming and Knowledge Discovery, vol. 2, no. 3, pp. 283-304, 1998.

[2] Joshua Zhexue Huang , Michael K. Ng , Hongqiang Rong , Zichen Li,“Automated Variable Weighting in k-Means Type Clustering,” IEEE Transactions on Pattern Analysis and Machine Intelligence, v.27 n.5, p.657-668, May 2005.

[3] Jiawei Han and Micheline Kamber, Data Mining:Concepts and Techniques (Second Edition. Jim Gray, Series Editor, Morgan Kaufmann Publishers, March 2006).

[4] Jiawei Han, Data mining: concepts and techniques (Morgan Kaufman Publishers, 2006).

[5] Margaret H Dunham, Data mining: introductory and advanced concepts (Pearson Education, 2006).

[6] Pena, J. M. , Lozano, J. A. , Larranaga, P, An empirical comparison of four initialization methods for the K-Means algorithm, Pattern Recognition Letters 20 (1999) pp. 1027-1040.

[7] Anderberg, M, Cluster analysis for applications (Academic Press, New York 1973).

[8] M. E. Celebi, H. Kingravi, P. A. Vela, A Comparative Study of Efficient Initialization Methods for the K-Means Clustering Algorithm, Expert Systems with Applications, 40(1), 2013, pp. 200-210.

[9] Tou, J. , Gonzales, Pattern Recognition Principles (Addison-Wesley, Reading, MA, 1974).

[10] Katsavounidis, I. , Kuo, C. , Zhang, Z. , A new initialization technique for generalized lloyd iteration, IEEE Signal Processing Letters 1 (10), 1994, pp. 144-146.

[11] Takashi Onoda, Miho Sakai, Seiji Yamada, Careful Seeding Method based on Independent Components Analysis for k-means Clustering, Journal Of Emerging Technologies In Web Intelligence, vol. 4, No. 1, February 2012.

[12] Stephen J. Redmond, Conor Heneghan, A method for initialising the K-means clustering algorithm using kd-trees, Pattern Recognition Letters 28(8), 2007, pp. 965-973.

[13] \Bradley, P. S. , Fayyad, Refining initial points for K-Means clustering: Proc. 15th International Conf. on Machine Learning, San Francisco, CA, 1998, pp. 91-99.

[14] Fernando Bacao, Victor Lobo, Marco Painho, Self-organizing maps as substitutes for K-means clustering, Computers and Geosciences, vol. 31, Elsevier, 2005, pp. 155-163.

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

[16] Shehroz S. Khan, Shri Kant, Computation of initial modes for k-modes clustering algorithm using evidence accumulation, 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, 2007, pp. 2784-2789.

[17] Koheri Arai and Ali Ridho Barakbah, Hierarchical k-means: an algorithm for centroids initialization for k-means, Reports of The Faculty of Science and Engineering Saga University, vol. 36, No. 1, 2007.

[18] S. A. Majeed, H. Husain, S. A. Samad, A. Hussain, Hierarchical k-means algorithm applied on isolated Malay digit speech recognition, International Conference on System Engineering and Modeling, vol. 34, Singapore, 2012.

[19] Samarjeet Borah, M. K. Ghose, Performance Analysis of AIM-K-means & K- means in Quality Cluster Generation, Journal of Computing, vol. 1, Issue 1, December 2009.

[20] K. A. Abdul Nazeer and M. P. Sebastian, Improving the accuracy and efficiency of the k-means clustering algorithm, Proceedings of the World Congress on Engineering, London, UK, vol. 1, 2009.

[21] Madhu Yedla, S. R. Pathakota, T. M. Srinivasa, Enhancing K-means Clustering Algorithm with Improved Initial Centre, International Journal of Computer Science and Information Technologies, 1 (2) , 2010, pp. 121-125.

Kamaljit Kaur is pursuing her PhD (Computer Science and Engineering) degree from Punjab Technical University, Kapurthala, Punjab. Her areas of interest are data mining, and clustering

Dr. Dalvinder Singh Dhaliwal received his PhD (Computer Science and Engineering) degree from Punjab Technical University, Kapurthala, Punjab. Presently, he is working as Director-Principal at Bharat Institute of Engineering and Technology, Sardulgarh, Mansa, Punjab. His areas of interest are data mining.

Dr. Ravinder Kumar Vohra is working as Professor at Bhai Gurdas Institute of Engineering and Technology, Sangrur, Punjab.