Design of Efficient K-Means Clustering Algorithm with Improved Initial Centroids

Afzali Maedeh
Dept. of CSE, FET,
MRIU Faridabad, Haryana, India
maedeh.af@gmail.com

Kumar Suresh
Dept. of CSE, FET,
MRIU Faridabad, Haryana, India
suresh.fet@mriu.edu.in

Abstract: *Data mining is a process of analyzing data from different perspectives and summarizing it into useful information. Clustering is one of the existing data mining techniques. It is the process of grouping a set of items into classes of similar objects. A cluster is a group of data elements having similar characteristics within the same cluster and are dissimilar to the objects in other clusters. One of the most common clustering algorithms is k-means algorithm that groups data with similar characteristics or features together. The k-means algorithm is very expensive and because of arbitrary selection of initial centroids, it does not produce unique clustering results for the multiple runs of the same input. Several attempts were made in the literature to improve the efficiency of k-means algorithm. In this paper an efficient method for finding better initial centroids is proposed. The proposed algorithm produces more accurate and unique clustering results.*

Keywords: *Data minnig, Data analysis, Cluster analysis, Clustering algorithms, K-means clustering algorithm, Initial centroids.*

1. Introduction

Data Mining is the process of extracting meaningful information patterns using computational tools and techniques on large datasets. Currently, there are a lot of data mining techniques that have been used in data mining projects. Clustering is a data mining technique that is used to form meaningful or useful clusters of objects which have similar characteristics[9, 4]. The purpose of clustering is grouping of data objects into clusters such that the objects in same cluster are similar and objects in different clusters are disimilar.

K means is the well known algorithm. It has been presented by MacQueen in 1967. K-means algorithm is a prototype based partitioning clustering technique that is used to partition n number of objects into user specified number of k clusters. The important concept in k-means algorithm is the centroid. Each cluster has a centroid. The Centroid refers to the center of the cluster.

The k-means clustering algorithm is easy to use and can be applied in several fields, but it is very sensitive to initial selection of clusters centers. The initial cluster centers are selected randomly. So the quality of final clustering results highly depends on the selection of the initial cluster centers. Due to this reason k-means algorithm does not produce unique clustering results when applied on the same data inputs multiple times. The computational complexity of the original k- means algorithm is very high. The computational time complexity of k-means is O (nkl), where n is the total number of data points in data set, k is required number of clusters and l is the number of iterations [1, 14].

Several attempts have been made by researchers to improve the efficiency of k-means algorithm. In this paper an enhanced method for finding better initial cluster centers is proposed, to get better clustering results. The algorithm uses efficient way to assign data points to appropriate clusters. The proposed algorithm produces more accurate and unique clustering results.

2. K-means Clustering Algorithm

In this secction original k-means clustering algorithm is discussed. The purpose of k-means algorithm is to classify a given set of data into k number of disjoint clusters, where the value of k is fixed in advance and is required as an input. The algorithm accepts two inputs: n the number of data points and k

the number of clusters. The output is k clusters with input data partitioned among them. Because of simplicity of k-means clustering algorithm it is used in various fields. The K-mean algorithm very popular because it can classify huge data efficiently[4].

K-means algorithm randomly selects k number of data points as initial centroids. Each data point is assigned to the cluster with the closest centorid. Then the centorid of each cluster is recalculated based on the mean of data points of each cluster. In this step some points can move from one cluster to other cluster. Again the new centroids are updated and the data points are assigned to suitable cluster. The assignment and updating of centroids is repeated, until no data point changes its cluster, means no data point move from one cluster to the other [12]. Pseudocode for k-means clustering algorithm is described below.

Algorithm 1 : K-Means Clustering Algorithm

Input: D = $\{d_1, d_2, d_3, \ldots\ldots, d_n\}$ // Set of n data points.
K // The number of desired clusters.

Output: A set of k clusters

Steps:

1: Select k data points as initial cluster centers randomly from the data set D.

Repeat

2: Calculate the distance between each data point d_i and the cluster centers.

3: Assign each data point to the cluster which has the nearest cluster center. (distance of data point is minimum from the cluster center)

4: For each cluster recalculate the cluster center.

Until no change in the center of clusters (no data point move from one cluster to another cluster).

Although k-means algorithm is easy to implement and it can classify huge amount of data, but it has some limitations. The algorithm can be applied only on numerical data. The quality of the final clustering result of the k-means algorithm highly depends on the selection of the initial cluster centers. In the original k-means algorithm, the initial centroids are chosen randomly. Hence the algorithm may give different clusters for multiple runs of the same input data points. The k-means algorithm is computationally expensive and requires time proportional to the product of the number of data items, number of clusters and the number of iterations. So the computational time complexity of k-means is O(nkl), where n is the total number of data points in data set, k is the required number of clusters and l is the number of iteratons [1, 14].

3. Related Work

K-means clustering algorithm is effictive in producing good clustering results for many practical applications, but it has some drawbacks. The quality of final clustering result is highly depending on the arbitrarly selection of initial centroids. The k-means algorithm is computationally expensive [1, 14, 5]. Several methods have been proposed to solve the cluster center initialization for k-means algorithm. In this section some of the proposals are reviewed.

A. M. Fahim et al. [5] proposed a simple and efficient algorithm based on the k-means algorithm, named enhanced k-means algorithm. In the original k-means algorithm the distance is calculated between each element to all centroids in each iteration and the required computational time of this algorithm depends on the number of iterations. In this approach time complexity is reduced by assigning the data points to suitable clusters. The initial centroids are determined randomly, so this approach also may not produce the unique clustering results.

Madhu Yelda et al. [14] proposed an enhanced method for improving the performance of k-means clustering algorithm. In the proposed solution they have used two methods: one for finding the better initial cluster centers and the other method for assigning data points to appropriate clusters.

K.. A. Abdul Nazeer et. al. [1] discussed a method to make the k-means algorithm more effective and efficient. In their proposed solution they have also used two methods, one for finding the better initial centroids and another for assigning data points to appropriate clusters with reduced time complexity.

Youguo Li et. al. [15] proposed a clustering method based on K-means algorithm. They have combined the largest minimum distance algorithm and the traditional k-means algorithm to produce an improved k-means clustering algorithm. The improved K-means algorithm effectively solved both limitations of the traditional k-means algorithm.

Koheri Arai et al. [10] proposed an algorithm to omptimize initial centroids of k-means algorithm. In the propsed algorithm both k-means and heirarchical algorithm is used. The mentioned method utilizes all the clustering results of k-means in certain times. Then, the result is tranformed by combining with hierarchical

algorithm in order to find the better initial centroids for k-means algorithm.

Aristidis Likas et al. [11] presented a global k-means algorithm which is an increamental approach to clustering that dynamically adds one cluster center at a time through a deterministic global search procedure consisting of N execution of the k-means algorithm from suitable initial positions. The authors in this paper proposed modifications of the method to reduce the computational load without significantly affecting solution quality.

4. Proposed Algorithm

In the proposed method both parts of the original k-mean, finding the initial cluster centroids and assigning the data points to suitable clusters is modified. The proposed algorithm is outlined as Algorithm 2.

Algorithm 2: The Proposed Algorithm

Input: D={d_1, d_2, d_3,……, d_n} // Set of n data points
K // Number of desired clusters

Output: A set of K clusters

Steps: Stage 1: Finding the initial cluster centers by using algorithm 3.Stage 2: Assigning each data point to the suitable cluster by using algorithm 4.

In the first stage the initial centorids are found to produce clusters with better accuracy. In this stage we use the larget minimum distance to find k initial centroids and in the second stage the algorithm assign the data points to the suitable clusters based on their distance to the initial centroids. Both part of the proposed method is discussed as Algorithm 3 and Algorithm 4.

The Algorithm 3 describes a method to find the initial centroids. Initially for each data point it compute the distance from origin by using Euclidean distance measure. The distance between two data points X= (x_1, x_2, x_3, … , x_n) and Y=(y_1, y_2, y_3, ……, y_n) is obtained as:

$$d(X,Y) = \sqrt{|x_1 - y_1|^2 + |x_2 + y_2|^2 + \ldots .. + +x_n - y_n|^2}$$

Then the data points are sorted according to the obtained distance in ascending order. The first and last point is selected as first two initial centroids. For each data point it finds the distance from first two initial centroids by using Euclidean distance. Then it assigns data points to the cluster having the closest centroid.

Algorithm 3: Finding initial centroids

Input: D={d_1, d_2, d_3,……, d_n} // Set of n data points
K // Number of desired clusters

Output: A set of K clusters

Steps:

1: For each data point calculate the distance from origin

2: Sort the distance obtained in step 1. Sort data points according to the distance.

3: Choose the first and last point as first c_1 and second c_2 initial centroids.

Repeat

4: For each data point d_i calculate its distance to initial centroids.

5: Assign each data point d_i to the cluster with closest centroid..

6: For each data point d_i
- 6.1: Set ClusterId[i]= j
- 6.2: Set Nearest Dist[i]= d(d_i, c_j)

7: Take the point that has the maximum Nearest Dist as the next initial centroid.

Until c_j =k

Algorithm 4: Assigning data points to suitable clusters

Input: D ={d_1, d_2, ……, d_n} // Set of n data points
C = {c_1, c_2, c_3, …., c_n} // Initial K centroids

Output: A set of k clusters.

Steps:

1: Compute the distance between each data point d_i and the initial cluster centers c_j.

2: For each data point d_i, find the closest centroid c_j and assign d_i to cluster j.

3: Set ClusterId[i] = j;

4: Set NearestDist[i] = d(d_i, c_j);

5: For each cluster j recalculate the centroids;

Repeat

6: For each data point d_i
- 6.1: Compute its distance from the centroid of the present nearest cluster.
- 6.2: If this distance is less than or equal to the present nearest distance, the data point stays in the same cluster; Else
- 6.2.1: For every centroid c_j calculate the distance d (d_i,c_j)
- 6.2.2: Assign the data point d_i to the cluster with the nearest cluster center c_j
- 6.2.3: Set ClusterId[i]= j
- 6.2.4: Set Nearest Dist[i]= d(d_i, c_j) Endfor;

7: For each cluster j (1<= j<=k), recalculate the centroids;

Until the convergence criteria is met.

This result in grouping the data points into two parts. For each data point, sets Cluster Id and Nearest Dist. Cluster Id denotes the cluster to which it is assigned and Nearest Dist of data points denotes the present nearest distance from closest cluster center. In the next step it takes the point which has the maximum nearest distance as the third centroid. Repeat these steps until k number of cluster centers are obtained.

The obtained initial centroid from Algorithm 3 is given as input to second stage to assign each data point to suitable cluster. Algorithm 4 determines an efficient way to assign data points to suitable cluster based on their distance to closest cluster center.

First it computes the distance of each data point to the initial centroids obtained by algorithm 3. Then it set the ClusterId and NearestDist. Then it recalculates the cluster. Next step is an iterative process. At the beginning of the iteration, the distance of each data point from the new cluster center of its present nearest cluster is determined. If the obtained distance is less than or equal to the previous nearest distance, the data point stays in the same cluster itself and no need to compute the distance to other cluster centers. If the obtained distance is more than the previous nearest distance, then there is need to compute the distance to other cluster centers. Then based on the compute distance assign the data point to the appropriate cluster and set the new value for ClusterId and Neasrest Dist. Repeat this iteration until the convergence criteria is met. In next section the 30 random data points are taken. We run experiments on this data points on the proposed algorithm to get better clustering results.

5. Illustrative Example

We have taken data points as shown in Table 1. Then we run experiments on this data on the algorithms proposed by [1, 5, 14] and the algorithm proposed by us. The results are shown in figures 5, 6, 7 and 8.

From the results it is clear that the clusters obtained by our proposed algorithm are better than algorithms proposed by [1, 5, 14].

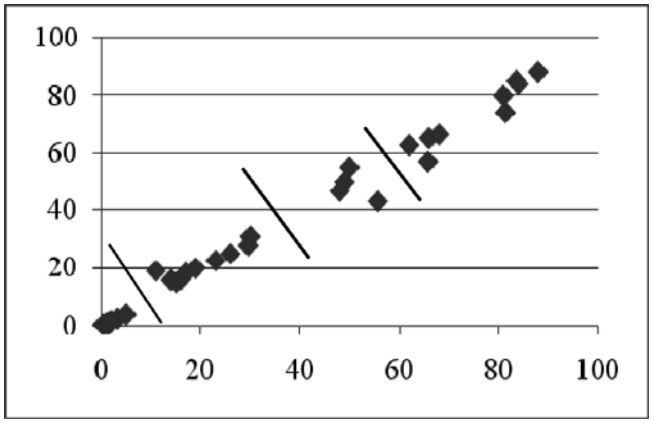

Fig 5: Proposed Algorithm

Table 1: Data Set D1

Pattern	abscissa	ordinate	Pattern	abscissa	ordinate	Pattern	abscissa	ordinate
d_1	2	1.7	d_{11}	66	65	d_{21}	11	19
d_2	14	15.8	d_{12}	26	25	d_{22}	16	16
d_3	1	1	d_{13}	0.3	0.6	d_{23}	84	84
d_4	15	15	d_{14}	1.5	1	d_{24}	88	88
d_5	30	31	d_{15}	1.67	1.5	d_{25}	50	55
d_6	29.6	28	d_{16}	62	62.63	d_{26}	49	50
d_7	23	22.6	d_{17}	68	66.3	d_{27}	48	46.67
d_8	3.2	2.5	d_{18}	81	80	d_{28}	19	20
d_9	5	4	d_{19}	81.4	74	d_{29}	55.7	43
d_{10}	65.8	57	d_{20}	83.6	85	d_{30}	17	18.23

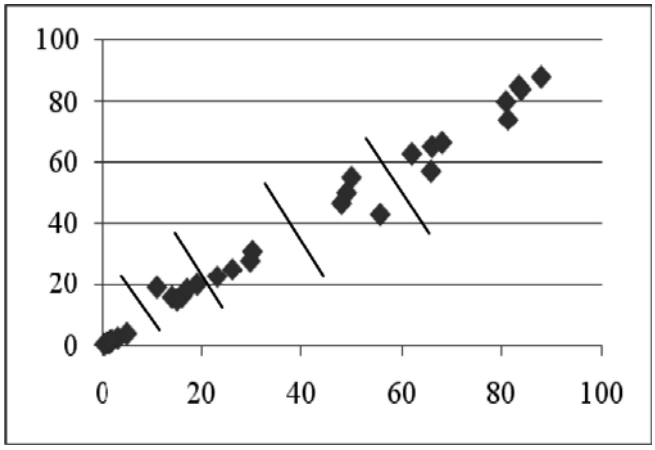

Fig 6: Proposed Algorithm by Fahim [5]

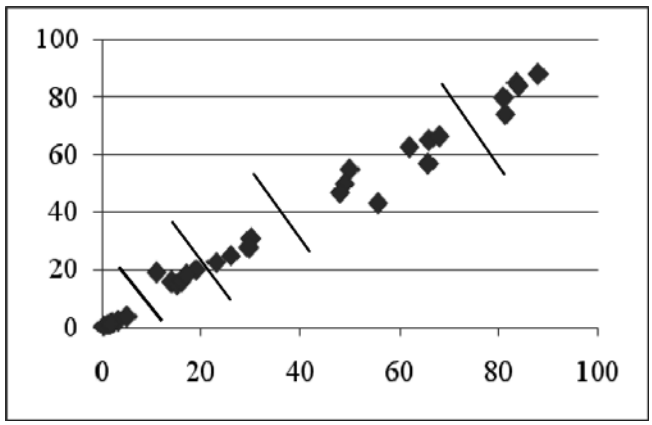

Fig 7: Proposed Algorithm by Nazeer [1]

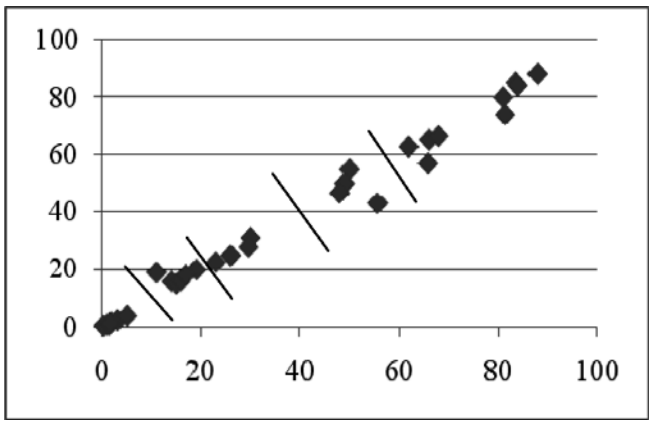

Fig 8: Proposed Algorithm by Madhu [14]

6. Conclusion

Several attempts were made by researchers to improve the efficiency of k-means algorithm. In This paper an enhanced method for finding better initial cluster centers, to get better clustering results and an efficient way to assign data points to appropriate clusters is presented. The proposed algorithm produces more accurate unique clustering results. We have taken data points as shown in Table 1 and then we run experiments on this data on the algorithm proposed by [1, 5, 14] and our proposed algorithm. The experimental results shows that the clusters obtained by proposed algorithm is better as compared to other algorithms. Like other algorithms the proposed algorithm also require value of k, the number of desired clusters as an input. We are working for finding some methods to compute the value of k, depending on the data distribution as a future work.

REFERENCES

1. Abdul Nazeer K. A. and Sebastian M. P. , " Improving the Accuracy and Efficiency of the K-means Clustering Algorithm", International Conference on Data Mining and Knowledge Engineering (ICDMKE), Vol. 1, London UK, 2009.

2. Chun Shemg Li, "Cluster Center Initialization Method for K-means Algorithm Over Data Sets with Two Clusters", International Conference on Advances in Engineering, Vol. 24, pp. 324-328, 2011.

3. Deelers S. and Auwatanamongkol S. , "Enhancing K-means Algorithm with Initial Clusters Centers Derived from Data Partitioning enter Initialization for k-mean Clustering", Pattern recognition Letters, Vol 25, Issue 11, pp. 1293-1302, 2004.

4. Erisoglu Murat, Calis Nazif, Sakallioglu, "A New Algorithm for Initial Cluster Centers in K-means Algorithm", Pattern Recognition Letters, Vol. 32, pp. 1701-1705, 2011.

5. Fahim A. M., Salem A. M. , Torkey F. A. and Ramadan M. A., "An Efficient Enhanced K-means Clustering Algorithm", Journal of Zhejiang University, Vol. 10, No. 7, pp. 1626-1633, 2006.

6. Fayyad Usama, Shapiro Gregory Piatetsky and Smyth Padhraic, "From Data Mining to Knowledge Discovery in Databases", AI Magazine, pp. 37-54, 1996.

7. Jiawei Han M. K., "Data Mining (Concepts and techniques)", Morgan Kufman Publishers, An Imprint of Elsevier, 2006.

8. Khan Shehroz. S., Ahmad Amir, "Cluster Center Initialization Algorithm for K-means Clustering", Pattern Recognition, Vol. 25, pp. 1293-1302, 2004.

9. Kanungo Tapas, Mount David M. , Netanyahu Nathan S., Piatko Christine D. , Ruth Silverman, and Angela Y. Wu, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 7, 2002.

10. Koheri Arai and Ali Ridho Barakbah, "Hierarchical K-means: An Algorithm for Centorids Initialization for k-means", Departement of Information Science and Electrical Engineering Politechnique in Surabaya, Faculty of Sicence and Engineering, Saga University, Vol. 36, No. 1, 2007.

11. Likas Aristidis, Vlassis Nikos and Verbeek Jakob J, "The global K-means clustering algorithm", Pattern Recognition 36, pp. 451-461, 2003.

12. Prasad R. N. , Archarya Seema, "Fundamentals of Business Analytics", First Edition, published by Wiley India, 2011.

13. Saha Sriparna, Bandyopadhyay Sanghamitra, "A Generalized Automic Clustering Algorithm in a Multiobjective Framework", Applied Soft Computing, vol. 13, pp.89-108, 2013.

14. Yedla Madhu, Srinivasa Rao Pathakota, T. M. Srinivasa, "Enhancing K-means Clustering Algorithm with Improved Initial Center", International Journal of Computer Science and Information Technologies (IJCSIT), Vol.1, No. 2, pp.121-125, 2010.

15. Youguo Li, Haiyan Wu, "A Clustering Method Based on K-Means Algorithm", International Conference on Solid State Devices and Materials Science", pp.1104-1109, 2012.

16. Yuan F., Meng Z. H. , Zhang H. X., Dong C. R., "A New Algorithm to Get the Initial Centroids", Proceedings of the third International Conference onMachine Learning and Cybernetics, pp. 26-29, 2004.