VOL. 11, NO. 2, JANUARY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

INITIALIZATION OF OPTIMIZED K-MEANS CENTROIDS USING DIVIDE-AND-CONQUER METHOD

J. James Manoharan and S. Hari Ganesh
Department of Computer Applications, Bishop Heber College, Tiruchirappalli, India
E-Mail: james_7676@yahoo.com

ABSTRACT

K-means clustering algorithm is one of the most popular unsupervised learning algorithm that is broadly used to clustering the given data items. The k-means algorithm is one of the commonly used clustering methods in data mining. A number of algorithms have been developed for clustering the data items using K-Means due to its simplicity and efficiency. The final clustering result of the K-Means clustering algorithm highly depends upon the initial centroids, which are selected at random by the user. The difficulty of determining "the right number of clusters" in traditional K-Means clustering has attracted significant importance especially in the recent years. There are many improvement were already developed to get better performance of the k-means, but most of these methods needed other inputs like threshold values for the number of data points in a data set. In this work, the proposed algorithm can solve the problems of finding initial centroids and assigning data items to proper clusters using divide-and-conquer method. So in proposed method, the initial cluster centers have obtained using divide-and-conquer property after that K-Means algorithm is applied to gain optimal cluster centers in dataset. The proposed algorithm can improve the execution speed of clustering the data items using little number of iterations. With the help of mathematical calculations the proposed algorithm decreases the complexity which we face in k-means clustering algorithm.

Keywords: K-means clustering, centroids, divide-and-conquer.

INTRODUCTION

Due to the enlarged availability of computer hardware and software and the fast computerization of business, huge amount of data has been composed and stored in databases. Researchers have expected that amount of information in the world doubles for every 20 months. However the raw data cannot be used directly. Its actual value is predicted by extracting information useful for assessment support. In most areas, data analysis was conventionally a manual procedure. When the size of data manipulation and exploration goes beyond human capabilities, people look for computing technologies to computerize the process. Data mining is one of the youngest research actions in the field of computing science and is defined as extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amount of data. Data mining is applied to gain some useful information out of bulk data. There are number of tools and techniques provided by researchers in data mining to obtain the pattern out of data.

Clustering is the method of organizing data objects into a set of disjoint classes called clusters. Large amount of data is being collected every day in many business and science areas [7]. This data needs to be analyzed in order to find interesting information from it, and one of the most important analyzing methods is data clustering. The simple K-means clustering algorithm is a popular data clustering algorithm. It is simple to implement and it is fast and sensitive [10]. However the K-Means algorithm has some drawbacks such as selection of initial centroids, number of iterations needed to find the clusters, and creation of empty clusters [4]. To overcome the drawbacks of traditional K-Means clustering algorithm a lot of works have been done by various researchers. In real life clustering problems it is quite difficult to choose the number of clusters present in final result [2]. A large numbers of procedures have been developed to determine the number of clusters present in the dataset. The appropriate number of clusters can be predicted for a given data set is generally a trial-and-error process made more difficult by the subjective nature of deciding what constitute perfect clustering. In this paper, a novel method is proposed to enhance the initialization problem of K-Means algorithm because the convergence result of K-Means algorithm is highly dependent on the initial centroids [8]. If the initial centroids are not chosen appropriately then the local optimum problem will be exist in traditional K-means clustering [5]. The good convergence result is directly proportional to the superior centroids. So the proposed method addresses the initialization as well as local optimum issues of traditional K-means clustering [1].

TRADITIONAL K-MEANS CLUSTERING ALGORITHM

The K-Means clustering algorithm is a partition-based cluster analysis technique. In this algorithm first we can randomly select k objects as initial centroids, then calculate the distance between each data object with each cluster centre and assign the data object to the nearest cluster and then calculate the new centroids, repeat this procedure until the criterion function converged. Finally, this algorithm aims at minimizing an objective function know as squared error function given by

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 ,$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a selected distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the *n* data points from their respective cluster centres**.**

Steps for k-means clustering Algorithm given below:
Algorithm: 1 Traditional K-Means
Let D = {x_1, x2, x3… xn} be the set of data objects.
1) Randomly select *'c'* centroids.
2) Compute the distance between each data objects and centroids using Euclidean distance.
3) Assign each data object to the nearest centroid whose distance from the centroid is minimum of all the other cluster centers.
4) New cluster center can be calculated using:

$$v_i = (1 / c_i) \sum_{j=1}^{c_i} x_i$$

where 'c_i' represents the number of data objects in i^{th} cluster.
5) The distance between each data object and newly obtained
centroids can be recalculated.
6) Stop the process if there is no data objects was reassigned,
Otherwise repeat from step (3).

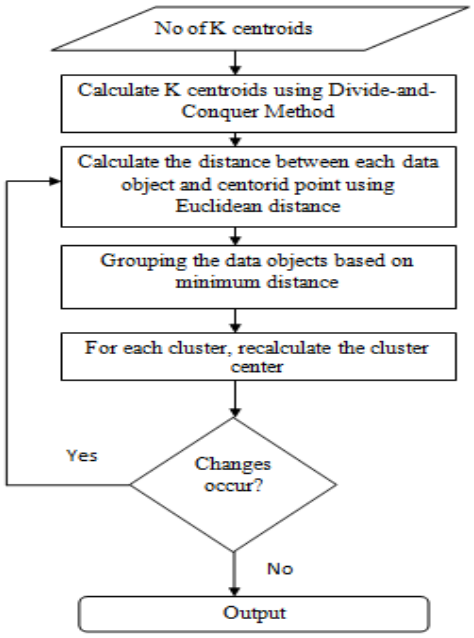

Figure-1. Flow chart of modified K-means.

The k-means algorithm randomly selects *k* initial cluster centers from the given dataset. After several iterations this algorithm will converge to the actual centroids. In K-means algorithm, choosing the correct set of initial cluster centers are mainly significant. However it is very difficult to select a good set of initial cluster centers randomly.

RELATED WORK

The simple k-means clustering algorithm is very sensitive to the initial starting points [10]. So, it is somewhat crucial for k-means to have refined initial centroids. A number of methods have been proposed in the literature for predicting the better initial centroids and some methods were proposed to improve both the accuracy and efficiency of the traditional k-means algorithm. In this paper, some of the more recent proposals are reviewed.

Yugal Kumar and G. Sahoo *et al.* [1], proposed a method to deal with the initial cluster centers problem in K-Means algorithm based on binary search procedure. Binary search technique is one of the popular searching methods that are used to find an item in given list of data items. In the proposed method, the initial centroids have obtained using binary search property and after that K-Means algorithm is applied to gain optimal cluster centers in dataset.

Arash Ghorbannia Delavar *et al.* [2], projected an algorithm to find the initial cluster centers based on selecting two attributes that can describe the data space better and using the number of neighbors in a specific radius in data space. 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. The improved algorithm is effective and computes suitable initial cluster centers and can help to find better solutions and converges in less iteration in data sets. Normalizing of attributes or using Normalized Euclidean Distance can help to improve the result.

Jiangang Qiao *et al.* [3], proposed an algorithm for improving the initialization of the cluster centers by reducing dimensions followed by moving cluster centers towards high density regions. The proposed algorithm can consists of three part. First, in order to speed up the process of choosing the initial centroids, a two dimensional subspace is selected from the feature space, in other words, two main variables which are most representative for the original data are selected for initializing the centroids. In second part, they find the determination of radius(R). Finally they can select the candidate cluster centers. The absolute clustering results can be enhanced by shifting the candidate centers to high density area.

Kalpana D. Joshi *et al.*[4], proposed an algorithm that has additional steps for selecting better cluster centers. This algorithm can compute Min and Max distance for every g and find high density objects for selection of better k. First, the user has to specify numeric value of k and

then randomly select k objects from data as initial centroids. The proposed algorithm can take k=2 as default value and randomly select two objects from data as first initial centers. Depending on data distribution the value of k can be incremented by splitting previously selected centers and for splitting they can apply some conditions. The proposed method can choose better value of k by splitting and select high dense object as cluster centers.

G Komarasamy *et al.* [5], proposed an algorithm, the k means combined with Bat Algorithm (KMBA) that utilizes the echolocation behavior of bats. This algorithm does not require the user is given in advance the number of centroid. However this KMBA does not guarantee unique clustering because we get different results with randomly chosen initial clusters. The final cluster centroids may not be the optimal ones as the algorithm can converge into local optimal solutions. Hence the blended k-means algorithm uses modified hill-climbing search to attempt to find the global optimal solution of the objective function. Hill-climbing algorithms are iterative algorithms which make modifications that increase the value of their objective function at each and every step. It is more effective in terms of reduced number of iterations with equal cluster density in all clusters reducing running time complexity of the deployment.

Er. Nikhil Chaturved *et al.* [6], proposed an algorithm that has two phases. In the first phase initial cluster centers can be determined systematically. In this phase the distance between each data items can be computed and then find out the closest pair of data. The set A1 consisting of these two data items and then delete these data items from the given data set D , after that calculate the data which is closest to the set A1, add the data item to A1 and then delete the data item from the given data set. This process repeated until the entire element in the set A1 is fulfilled. Next go back to the second step and then form another data set A2. Finally the initial cluster centers are obtained by averaging all the data in each given data set D. The Euclidean distance is used for finding the close of each data to the centroids. In the second phase, the initial centroids are used as input and assigning data to appropriate clusters.

PROPOSED ALGORITHM

In this section an enhanced cluster initialization method is proposed for K-Means algorithm. The proposed method is used to generate the initial cluster centers using divide-and-conquer technique. The method and algorithmic steps of proposed algorithm are given.

Clustering Process: In the enhanced algorithm, the given input remains in the same order in which data items are entered. The entire process is divided into two phases.

Phase-I: How to choose initial cluster centers?
Clusters (K) and data set is provided by the user. First calculate the difference between minimum and maximum value from the given data as follows:

#
$$diff = Max - Min \quad (1)$$

Then divide the Diff with K as follows:

$$diff = \frac{\text{Max}-\text{Min}}{\text{K}} \quad (2)$$

where K-is the number of clusters.

Divide the given data objects into Gj($1 \leq j \leq K$) groups. Each group can contain items and then store each group into two dimensional array GR in row-wise order. Calculate the distance between data item d_i ($1 \leq i \leq n$) in each group Gj($1 \leq j \leq K$) and the value calculated from equation (2) and then the result will be stored into array GD as follows:

$$GD[i,j] = Abs(GR[i,j] - diff) \quad (3)$$

Finally calculate mean for each row in the array GD. This value will be taken as initial centroids that can be stored into an array CENT as follows:

$$\text{CENT[i]} = 1/(\text{n/k})\left(\sum_{j=1}^{n/k} \text{GD[i,j]}\right) \quad (4)$$

where $1 \leq i \leq k$ and $1 \leq j \leq n/k$.

Phase-II

In this phase our algorithm [11] is to set two simple data structures to retain the labels of cluster and the distance of all the date objects to the nearest cluster center during the each iteration, that value might be used in next iteration. Then compute the distance between the current data object and the new centroid, suppose the computed distance is smaller than or equal to the distance to the old centroid then the data object stays in same cluster. Therefore, there is no need to calculate the distance from this data object to the other k-1 cluster centers, saving the calibrated time to the k-1 clusters. Otherwise, it will calculate the distance from the current data object to all k cluster centers, find the nearest cluster center and assign this point to the nearest cluster center, then individually records the label of nearest cluster center and the calculated distance of its center. Because in each iteration some data points still remain in the existing cluster that means some parts of the data points need not be calibrated and saving the total time of calculating the distance, thereby increasing the efficiency of the algorithm.

Algorithm 2: The enhanced K-means method
Require: *D = {d1, d2, d3... dn}* // Set of n data points.
k // Number of desired clusters.
Ensure: A set of *k* clusters.

Steps:

1) Compute the difference between the maximum and minimum value of given data set.
2) Divide the difference value obtained in step 1 with K.
3) Partition the given data objects into G groups, each group can contain n/k data objects.

4) Calculate distance between data item in each group with the
difference value obtained in step 2 and then the result will
be stored into an array GD [i,j], where $1 \leq i \leq k$ and $1 \leq j \leq n/k$.
5) Calculate the mean for each row in GD then the mean value
will be taken as initial centroids.
6) The calculated centroid values can be stored into an array
CENT[i] ($1 \leq i \leq k$).

7) Calibrate the distance between every data object di ($1 <= i <= n$) and all k cluster centers cj ($1 <= j <= k$) as
Euclidean distance d (d_i, c_j) and assign data object d_i to the
nearest cluster.
8) For each data object d_i, find the nearest center c_j and assign
data object d_i to cluster center c_j
9) Detect the name of cluster center and the distance of data
object d_i to the closest cluster. Then this information is
stored in list Clu[ ] and the Dis[ ] separately.
Set Clu[i]=j, j is the name of nearby cluster center.
Set Dis[i]=d(d_i, c_j), d(d_i, c_j) is the Euclidean distance to the nearest center.
*10) R*ecalculate the cluster center;
11) Repeat
12) For each data object di ($1 <= i <= N$) Compute its distance to
the center of the current closest cluster;
a) If this distance <= Dist[i], then assign the data object d_i in the initial cluster;
b) Else
For every cluster center c_j, calculate the distance
d (d_i, c_j) of each data object to all the centroid and
then allocate the data object of data set d_i to the
nearest neighboring cluster center c_j.
Set Clu[i]=j;
Set Dis[i]= d(d_i, c_j);
*13) R*ecalculate the centers for each cluster center j($1 <= j <= k$),
*14) Repeat the process u*ntil the convergence criteria is achieved.
15) Showing the clustering results.

EXPERIMENTAL RESULT

In this experiment section, here we estimate the performance of the proposed algorithm. We tested original K-means and proposed K-means algorithm for the data sets with known clustering, Iris, New Thyroid, Height-Weight and Diggle. Both the algorithms need number of clusters as an input. In additional, for the traditional k-means clustering algorithm the set of initial centroids are randomly selected by user but in the proposed method the initial cluster centers can be selected systematically. The proposed method needs only the data values and number of clusters as inputs and it does not take any additional inputs like threshold values. The simple K-means algorithm is executed several times for different sets of values of the initial centroids. In each experiment the accuracy and time was computed and taken the average accuracy and time of all experiments. Table-1 depicts the performance comparison of the traditional k-means and proposed k-means algorithms.

Table-1. Performance comparison of simple K-means and proposed K-means algorithms.

Data Set	Running Time in ms		No. Of Iterations		Accuracy (%)	
	Standard K-means	Enhanced K-means	Standard K-means	Enhanced K-means	Standard K-means	Enhanced K-means
Fisher's Iris	0.096	0.082	8	4	86.3	89.7
Height-Weight	0.081	0.072	14	8	78.9	86.2
Echocardiogram	0.097	0.082	9	6	91.3	97.65
Diggle	0.092	0.072	13	6	79.3	89.65

Figure-2, 3, and 4 can depict the results with the help of bar charts. The results obtained illustrate that the proposed algorithm is producing better distinctive clustering results compared to the traditional k-means algorithm in less amount of computational time.

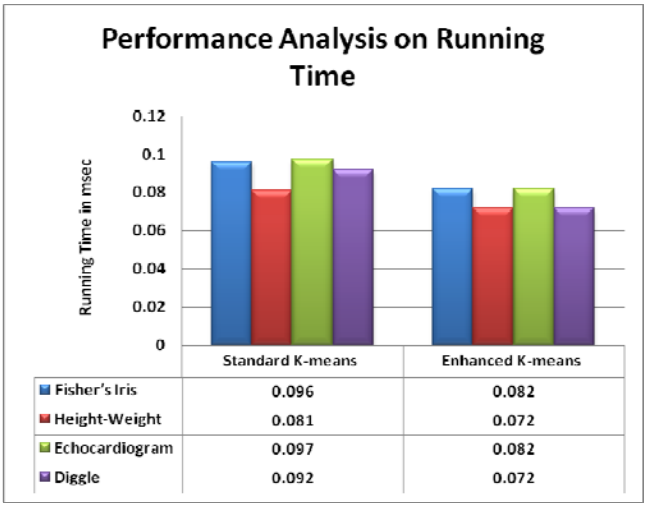

Figure-2. Comparison of traditional K-means clustering algorithm and proposed K-means clustering algorithm on the basis of running time.

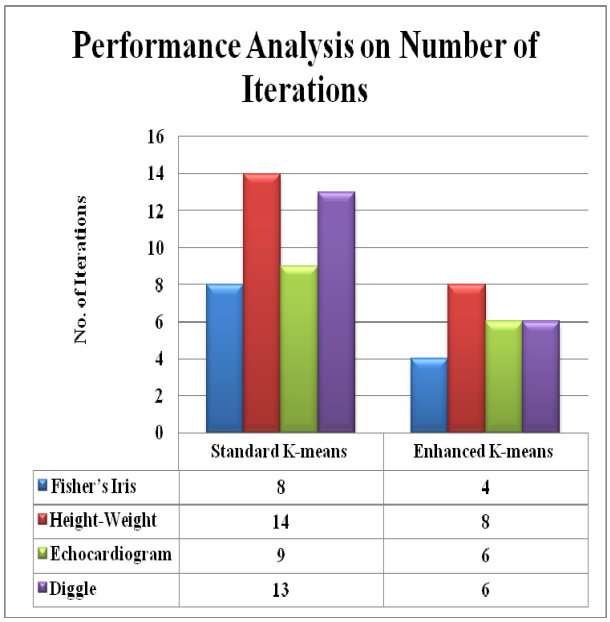

Figure-3. Comparison of traditional K-means clustering algorithm and proposed K-means clustering algorithm on the basis of number of iterations.

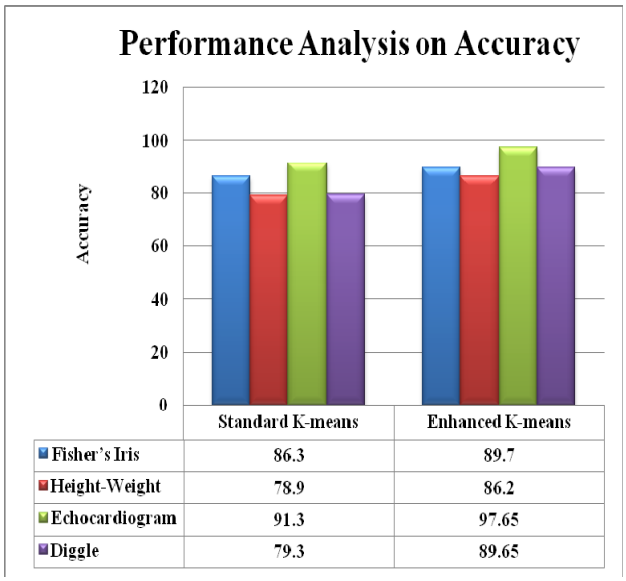

Figure-4. Comparison of traditional K-means clustering algorithm and proposed K-means clustering algorithm on the basis of accuracy.

CONCLUSIONS

The problem of initialization in simple K-Means clustering algorithm is formulated by two ways; first, the number of clusters required for clustering task and second, how to initialize initial centroids for K-Means clustering algorithm.

The second issue of the initialization problem of cluster centers can be resolved in this paper using divide-and-conquer method. In proposed algorithm, initial cluster centers are obtained with the help of divide-and-conquer method and after that K-Means algorithm is applied. This paper presents an improved k-means algorithm which uses a systematic method to finding initial centroids and an efficient way for assigning data items to appropriate clusters. This algorithm ensures the clustering of data in less time without affecting the accuracy of clusters. The results do not depend on the ordering of data and computational efforts are minimized by using the threshold value. Our experimental results show that the proposed algorithm produces better results than that of traditional k-means algorithm.

In future new approach could be applied to avoid the empty clusters.

REFERENCES

[1] Yugal Kumar and G. Sahoo *et al.*," A New Initialization Method to Originate Initial Cluster Centers for K-Means Algorithm", International Journal of Advanced Science and Technology ,Vol.62(2014), pp.43 -54 http://dx.doi.org/10.14257/ijast.2014 .62 .04.

[2] Arash Ghorbannia Delavar and Gholam Hasan Mohebpour, "ANR: An algorithm to recommend initial cluster centers for k-means algorithm", Journal of mathematics and computer science 11 (2014), 277-290.

[3] Jiangang Qiao and Yonggang Lu, " A new algorithm for choosing initial cluster centers for k-means" , Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013), Published by Atlantis Press, Paris, France.

[4] Kalpana D. Joshi *et al.*, "Modified K-Means for Better Initial Cluster Centres", International Journal of Computer Science and Mobile Computing , IJCSMC, Vol. 2, Issue. 7, July 2013, pg.219– 223.

[5] G Komarasamy *et al.*, "A New Algorithm For Selection Of Better K value using Modified Hill Climbing in K-Means Algorithm", Journal of Theoretical and Applied Information Technology, ISSN: 1992-8645, 30th September 2013. Vol. 55 No.3.

[6] Er. Nikhil Chaturvedi *et al.*, "Improvement in K-mean Clustering Algorithm Using Better Time and Accuracy", International Journal of Programming Languages and Applications (IJPLA) Vol.3, No.4, October 2013.

[7] M. S. V. K. Pang-NingTan, "Data mining," in Introduction to data mining, Pearson International Edition, 2006, pp. 487-496.

[8] Middle-East Journal of Scientific Research 12 (7): 959-963, 2012 ISSN 1990-9233 © IDOSI Publications, 2012 DOI: 10.5829/idosi.mejsr.2012.12.7.1845.

[9] W. Barbakh, and C. Fyfe, 2008. Local vs. Global Interactions in Clustering Algorithms: Advances over K-means, International Journal of knowledge-based and Intelligent Engineering Systems, vol. 12. http://iospress.metapress.com/content/6443723640674366.

[10] K. A. Abdul Nazeer and M. P. Sebastian, "Improving the accuracy and efficiency of the k-means clustering algorithm," in International Conference on Data Mining and Knowledge Engineering (ICDMKE),Proceedings of the World Congress on Engineering (WCE-2009), Vol 1, July 2009, London, UK.

[11] J. James Manoharan and S. Hari Ganesh, "Improved K-means clustering Algorithm using Linear Data Structure List to Enhance the Efficiency", International Journal of Applied Engineering Research, ISSN 0973-4562 Vol. 10 No.20 (2015).