Enhancement in the Performance of K-means Algorithm

Daljit Kaur[1], Kiran Jyoti[2]

[1]M.Tech Scholar, Department of Computer Science
[2]Assistant Professor, Department of Information Technology
[1,2]Guru Nanak Dev Engineering College, Ludhiana (Punjab), India
[1]er.daljitkaur1985@gmail.com, [2]kiranjyotibains@yahoo.com

Abstract— **Clustering is a division of data into groups of similar objects. Each group consists of objects that are similar among them and dissimilar compared to objects of other groups. K-means algorithm is widely used for clustering data. But this algorithm is computationally expensive and quality of final results depends on the selection of initial centroids. This paper proposes a method to make the algorithm more efficient and effective. The proposed method decreases the complexity and effort of numerical calculation but it maintains the easiness of implementing k-means algorithm. It also solves the problem of dead unit.**

Keywords— **Data clustering, K-means algorithm, Objective function.**

I. Introduction

Data clustering is an important unsupervised learning method. In clustering, a set of objects are classified into groups such that members of one group are similar to one another [1]. Clustering is a main task of explorative data mining and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval and bioinformatics[9]. There are many clustering algorithms. Two goals of clustering algorithms are determining good clusters and doing it efficiently.

K-means algorithm is the simplest and most commonly used algorithm. K-means algorithm is successful in producing clusters for many practical applications. But the computational complexity of k-means is very high, especially for large data sets [6]. It does not give the same result with each run, since the resulting clusters depend on the initial center points which are selected randomly [3]. It can contain dead unit problem. This paper presents an improved k-means algorithm which uses a systematic method to finding initial center points and an efficient way for assigning data items to appropriate clusters in less time. The proposed algorithm enhances the speed of clustering thus improves the time complexity of algorithm. It also solves the problem of dead unit. By comparing the experimental results of the standard k-means and proposed k-means, it shows that the proposed method outperforms in terms of efficiency and accuracy.

The paper includes six sections. Section 2 details the k-means algorithm and shows the shortcomings of k-means algorithm. Section 3 describes the related work. Section 4 presents the proposed k-means algorithm. Section 5 shows experimental results. Finally section 6 concludes the paper.

II. K-means Clustering Algorithm

The K-means algorithm is the one of the simplest unsupervised learning algorithm that solves the clustering problem. The algorithm follows an easy way to classify a given data set into a certain number of clusters. The algorithm consists of two separate steps. The first step is to select k initial centroid randomly, one for each cluster. The next step is to take each point from a given data set and assign it to the nearest centroid. When all points are assigned to some clusters, the first step is completed and an early grouping is done. At this point we need to find the new centroids by calculating the mean values of clusters. A loop has been generated. As a result of this loop, k centroids change their location step by step until no more changes are done. This signifies the convergence criterion for clustering [7]. Finally, this algorithm aims at minimizing an *objective function*, in this case a squared error function. The objective function

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

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen 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. K-means algorithm is given below (Algorithm 1).

Algorithm 1. K-means Clustering Algorithm

Input:
D = {d1, d2, d3,…,dn} //set of n data items
K // number of desired clusters

Output:
A set of k clusters

Steps:
1. Choose k data items from D randomly as initial centroids;
2. Repeat
   Assign each item d_i to the cluster which has closest centroid;
   Calculate new mean for each cluster;

Until convergence criteria is met.

Advantages of K-means Clustering Algorithm:

- This algorithm is scalable and efficient in processing large data sets.
- This algorithm is easy and results are easily understandable.

Shortcomings of K-means Clustering Algorithm:

- The results depend on the initial cluster centroid.
- The number of clusters k must be given in advance.
- It is sensitive to noise and outlier data points.
- It can contain the dead unit problem.
- Results depend on the ordering of data.

III. Related Work

Several attempts were made by researchers for improving the performance of K-means algorithm. As shown in Algorithm 1, original k-means algorithm has two steps: one for determining initial centroids and the other for allocating the data points to respective clusters.

Singh et al.[4] proposed a modified algorithm based on the improvement of the sensitivity of initial center of the clusters. This algorithm partitions the whole space into k*k segments and calculates the frequency of data point in each segment. It decreased the complexity and the effort of numerical calculation, maintaining the easiness of implementing the k-means algorithm.

Shi Na et al.[5] shows a problem of K-means clustering algorithm that this algorithm has to calculate the distance between each data object and all cluster centroids in each iteration, which makes the efficiency of the clustering not very high. As a solution, an improved K-means clustering algorithm is shown as an improvement in avoiding the computing distance for every data object from the centroid and also used for saving the time.

Napoleon et al.[6] discussed that how uniform distribution of the data points approach reduce the time complexity of the K-means clustering algorithm. This method calculates the distance between each data points. Then it selects pair which show less distance and delete it from data set. Repeat this process until threshold value achieved. By using this approach the elapsed time is reduced and the cluster is of better quality.

Nazeer et al.[7] proposed an improvement in K-means clustering. In this paper, in the first phase of K-means clustering algorithm, the initial centroids are determined systematically so as to produce clusters with better accuracy. The second phase makes use of an efficient way for assigning data points to clusters. This method ensures the entire process of clustering in O (n2) time without sacrificing the accuracy of clusters. So this algorithm improves the accuracy and efficiency and also reduced complexity.

IV. Proposed K-means Algorithm

The proposed k- means algorithm solve the problem of dead unit and optimizes the selection of initial centroids of clusters by using most populated area as a centroid of cluster. It ensures the minimum execution time during the allocation of data points to respective clusters. It is scalable and efficient in processing large data sets. Final results never depend on the ordering of data. It can be used when large number of clusters is required. Proposed algorithm overcomes the shortcomings of k-means algorithm. We divide our algorithm into two phases:

Phase1: Selection of Initial Centroids

The number of clusters (K) and data set is provided by the user. With the help of value K, value of x is calculated. Whole data space is divided into x*x blocks (x horizontally and x vertically). Frequency of data items in each block is calculated. Then K blocks are selected with the highest frequency. To find out the initial centroid, the mean of each selected block is calculated.

Phase 2: Allocation of Data Points to Respective Clusters

To assign the data items to the appropriate clusters, we have to find the distance matrix by calculating the distance between each cluster's centroid. For each centroid, take the minimum distance from the remaining centroid and make it half and denote it by T_i (threshold value). Then take data items one by one and find the difference from i^{th} centroid. If this difference is less than or equal to the threshold value T_i, then the data item will be assigned to the i^{th} cluster, otherwise, find out the distance from the other centroid. If data item is not assigned to any of the cluster, then assign data item to the cluster which has closest centroid. Repeat this process for each data item. Calculate the mean of the clusters and check the termination condition. If the condition is not satisfied then update the centroid of clusters and repeat the Phase 2.The proposed algorithm is given below (Algorithm 2).

Algorithm 2. Proposed K-means Clustering Algorithm

Input:

D = {d1, d2, d3,…,dn} //set of n data items
K // number of desired clusters

Output:

A set of K clusters

Steps:

1. Input the value of K and data set D.
2. Calculate the value of x as integer
   x= $(K*2.5)^{1/2}$ +1
   // divide the data space into x*x means x horizontally and x vertically
3. For each dimension
   Find out the minimum and maximum value of data items.
   Calculate range of group (G) using equation
   G= (max-min)/x
   Divide the data space in x group with width G.
4. Calculate the frequency of data items in each block.
5. Select K groups having highest frequency.
6. Find out the value of initial centroid of clusters by calculating mean of selected group.
7. Find out the distance matrix for clusters by calculating distance between centroids.

$|C_i, Cj|= \{d (m_i,m_j)\}$// m_i,m_j denotes the means of i,j clusters respectively.

8. Take the minimum distance for each cluster and make it half
   $Ti=½(\min \{|C_i,C_j|\})$ $1<=i<=K, 1<=j<=K$
9. For each data item p=1 to N
   For each cluster q=1 to K
   Calculate distance between data item and centroid (d_p,m_q)
   If distance $(d_p,m_q)<=T_q$ then
   Assign data itemd_p to cluster C_q.
   Break;
10. For each data item p=1 to N
    If data item d_p does not assigned to any cluster then
    Assign data item d_p to the cluster which has closest centroid.
11. Take mean of each cluster separately and check the termination condition of algorithm.
12. If satisfied then exit.
    Otherwise update the centroid of cluster.
    Go to Step 7.

V. Experimental Results

In this section, we evaluate the performance of the proposed algorithm. We have implemented proposed algorithm in Visual Basic.NET. Our experiments were run on 2.30 GHz Intel® core™ i3-2350 machine with 2 GB of RAM using Visual Studio 2008 on Windows 7.

The real life data set Flame (shape) is taken for experiments. It is two dimensional data having 2 clusters. This data set has 240 data points. This data set is used for testing the efficiency and accuracy of the proposed algorithm [8].

We have examined the execution time needed by K-means algorithm and proposed K-means algorithm. The sum of squared error is also calculated. The number of clusters is chosen by the user. The data points in each cluster are displayed by different colours and the execution time is calculated in milliseconds. Both of the algorithms are executed for the different number of clusters for five times and the average is taken. Execution time, sum of squared error and number of iterations for clustering are shown in Table 1.From the Table 1, it is seen that execution time and sum of squared error required by proposed K-means is less as compared to that of original k-means algorithm.

For the purpose of comparison of clustering quality, accuracy of clustering is calculated. The accuracy of clustering is determined by comparing the clusters obtained by the experiments with the two clusters already available with data set [8]. The percentage accuracy for k-means algorithm and proposed k-means algorithm is 83.75 and 85 respectively.

It can be seen that proposed k-means algorithm outperforms the k-means algorithm in terms of efficiency and accuracy. It is also observed that in k-means algorithm, the result depends on the selection of initial centroid and the order of data points. But in proposed k- means algorithm, result remains same for different order of data points. Every time algorithm runs, results remain same in terms of efficiency and accuracy. Problem of dead unit is also solved.

Figure 1 and Figure 2 depicts the performance of K-means algorithm and proposed k-means algorithm respectively in terms of accuracy, execution time and sum of squared error.

TABLE 1
COMPARISON OF K-MEANS ALGORITHM AND PROPOSED K-MEANS ALGORITHM

No. of clust-ers	K-means algorithm			Proposed K-means algorithm		
	SSE	Exec. time in ms	No. of iterati-ons	SSE	Exec. time in ms	No. of iterati-ons
2	5574	4	8	5206	1.6	4
4	2089	11.4	14	2087	3	7
5	1820	9.4	9	1778	2.4	5
7	1295	19.2	13	1217	7.2	12
10	898	28.8	14	879	8.4	10
12	749	30	13	748	6.2	6
17	573	46.4	14	533	13.4	10

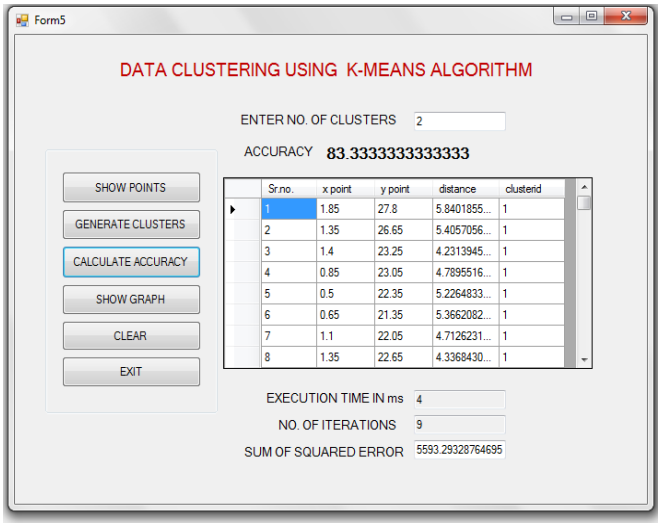

Figure 1. Performance of K-means algorithm

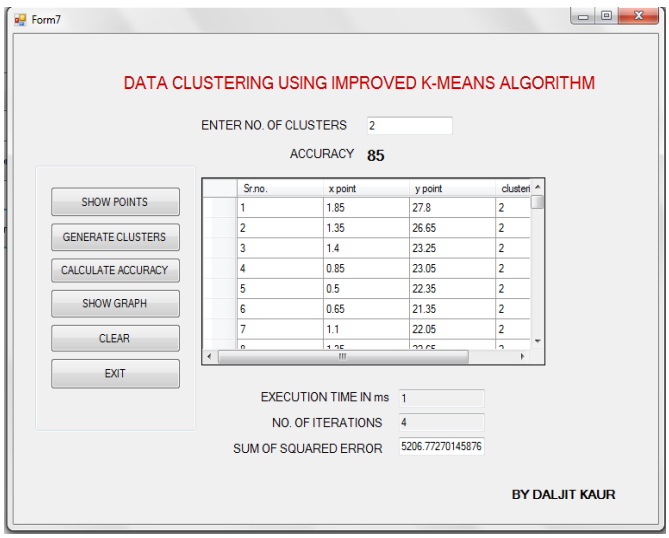

Figure 2. Performance of proposed K-means algorithm

The resulting clusters and their centroids for k=2 for K-means algorithm are presented in Figure 3. The resulting clusters and their centroid for k=2 for proposed K-means algorithm are presented in Figure 4.

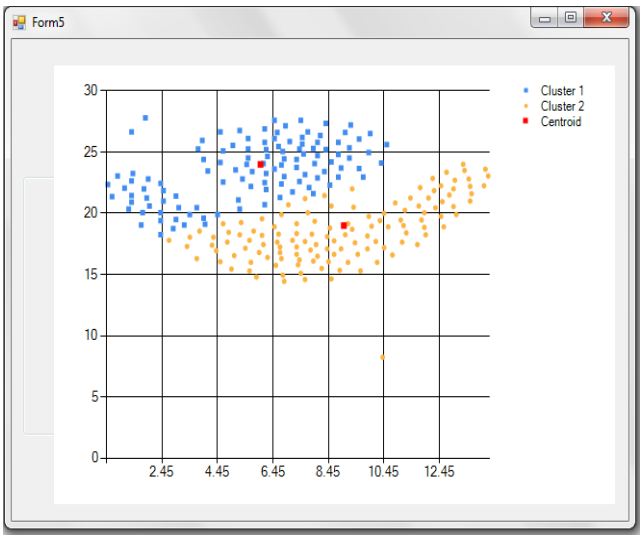

Figure 3. Resulting clusters and their centroids for k=2 for K-means algorithm

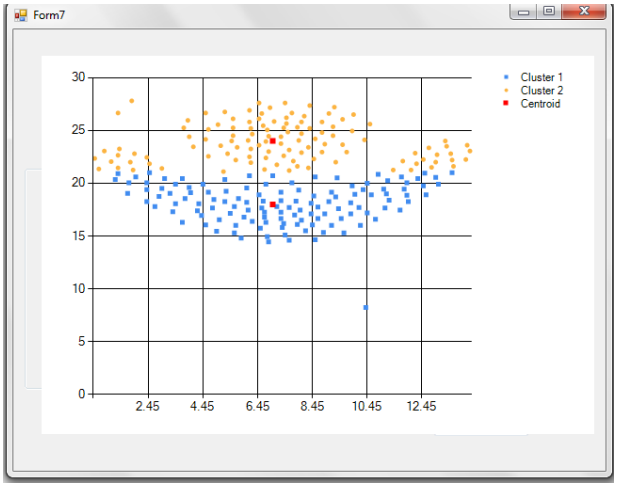

Figure 4. Resulting clusters and their centroids for k=2 for proposed K-means algorithm

VI. Conclusion

The k-means algorithm is widely used for clustering data. But this algorithm does not always give good results, because the accuracy and efficiency of the resulting clusters depend on the selection of initial centroids. 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 does not have dead unit problem. This algorithm ensures the clustering of data in less time without sacrificing 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 k-means algorithm.

Acknowledgements

I would like to express my very great appreciation to Mr. Sukhmandeep Singh for his valuable and constructive suggestions during the planning and development of this research work. His willingness to give his time so generously has been very much appreciated. I wish to thank my parents for their support and encouragement throughout my study.

References

[1] Malay K. Pakhira“ Clustering Large Databases in Distributed Environment” IEEE International Advance Computing Conference (IACC 2009) held at Patiala, India from 6-7 March 2009.

[2] Madhuri A. Dalal, Naresh Kumar D.Harale, Umesh L. Kulkarni “An Iterative Improved K-means Clustering” ACEEE Proc. Of Int. Conf. on Advances in Computer Engineering 2011, DOI:02.ACE.2011.02.183, 2011, Pg.25-28.

[3] JirongGu, Jieming Zhou, Xianwei Chen “An Enhancement of K-means Clustering Algorithm” IEEE International conference on Business Intelligence and Financial Engineering DOI:10.1109/BIF.2009.204 Pg.237-240

[4] Ran Vijay Singh, M.P.S. Bhatia “Data Clustering with Modified K-means Algorithm” IEEE-Int. Conf. on Recent Trends in Information Technology, ICRTIT 2011, MIT held at Anna University, Chennai from 3-5 June, 2011.

[5] Shi Na, Liu Xumin, Guan Yong “Research on k-means Clustering Algorithm” IEEE Third International Symposium on Intelligent Information Technology and Security Informatics, DOI: 10.1109/IITSI.2010.74, 2010 Pg. 63-67.

[6] D. Napoleon, P. Ganga Lakshmi “An Efficient K-Means clustering algorithm for reducing time complexity using uniform distribution data points” IEEE, 2010.

[7] K. A. Abdul Nazeer, M.P. Sebastian “Improving the Accuracy and Efficiency of the k-means Clustering Algorithm” Proc. Of the World Congress on Engineering 2009 Vol 1, WCE2009 held at London, U.K. from 1-3 July 2009 ISBN: 978-988-10712-5-1.

[8] Flame data set http://cs.joensuu.fi/sipu/datasets/flame.txt

[9] http://en.wikipedia.org/wiki/Cluster_analysis