An Improved K-means Clustering Algorithm with refined initial centroids

Madhu Yedla, Sandeep Malle, Srinivasa T M
Department of Computer Science and Engineering
National Institute of Technology
Calicut, Kerala, India
yedlamadhu@gmail.com, sandeep.nitc@yahoo.com, srini_2007@nitc.ac.in

ABSTRACT: *A final Clustering result of the k-means clustering algorithm greatly depends upon the correctness of the initial centroids. Generally the initial centroids for the k-means clustering are chosen randomly so that the selected initial centroids may converges to numerous local minima, not the global optimum. In this paper a new initialization approach to find initial centroids for k-means clustering is proposed. According to our experimental results, the Improved k-means Clustering Algorithm has the more accuracy with less computational time comparatively Original k-means clustering algorithm.*

Keywords: Clustering, Data nining, Data partition-ning, Initial centroids, *K*-means algorithm. Cluster analysis, Data analysis

Received: 12 April 2009, Revised 9 May 2009, Accepted 14 May 2009

1. Introduction

Clustering is the process of organizing a given set of objects into a set of disjoint groups called clusters. It makes the objects within a cluster are more similar to each other than the objects in different clusters [4][3]. Clustering is an important area of research, which finds applications in many fields including bioinformatics, pattern recognition, image processing, marketing, data mining, economics, etc.

Cluster analysis is a one of the primary data analysis tool in the data mining. Clustering algorithms are mainly divided, into two categories: Hierarchical algorithms and Partitional algorithms. A hierarchical clustering algorithm divides the given data set into smaller subsets in hierarchical fashion. A partitional clustering algorithm partition the data set into desired number of sets in a single step [4].

This paper is organized as follows. Section 2 presents an overview of k-means algorithm and a short analysis of the existing clustering methods. Section 3 introduces proposed method. Section 4 describes about the time complexity of the proposed method. Section 5 experimentally demonstrates the performance of the proposed method. And the final Section 6 describes the conclusion.

2. Related Work

One of the most well known partitional clustering algorithms is the *k*-means algorithm. In k-means clustering algorithm we choose k points as initial centroids randomly, where k is a user specified parameter. Each point is then assigned to the cluster with the closest centroid [5][1][2]. Then the centroid of each cluster is updated by taking the mean of the data points of each cluster. We repeat the assignment and update the centroids, until no point changes clusters, or equivalently, until the centroids remain the same. In this algorithm mostly Euclidean distance is used to find distance between data points and centroids [3]. Pseudocode for the k-means clustering algorithm is summarized in Algorithm 1.

Algorithm 1: The k-means clustering algorithm [3]

Require: D = {d_1, d_2, d_3, ..., d_i, ..., d_n } // Set of n data points.

k // Number of desired clusters

Ensure: A set of *k* clusters.

Steps:

1. Arbitrarily choose *k* data points from *D* as initial centroids;
2. **Repeat**

   Assign each point d_i to the cluster which has the closest centroid;

   Calculate the new mean for each cluster;

   Until convergence criteria is met.

The quality of the final clusters of the k-means algorithm highly depends on the initial centroids. In the original k-means algorithm, the initial centroids are chosen randomly and hence we get different clusters for different runs for the same input data [5]. Moreover, the k-means algorithm is computationally

very expensive also. The computational time complexity of the k-means algorithm is (nkl), where n is the total number of data points in the dataset, k is the required number of clusters and l is the number of iterations [3]. Several methods have been proposed in the literature to improve the performance of the original k-means clustering algorithm.

Zhang Chen and Xia Shixiong [1] proposed the initial centroids algorithm based on k-means that have avoided alternative randomness of initial center.

Fang Yuan [2] proposed the initial centroids algorithm. The standard k-means algorithm selects k-objects randomly from the given data set as the initial centroids. If different initial values are given for the centroids, the accuracy output by the standard k-means algorithm can be affected. In Yuan's method the initial centroids are calculated systematically.

Koheri Arai et al. [3] proposed an algorithm for centroids initialization for k-means. In this algorithm both k-means and hierarchical algorithms are used. This method utilizes all the clustering results of k-means in certain times. Then, the result transformed by combining with Hierarchical algorithm in order to find the better initial cluster centers for k-means clustering algorithm.

3. Proposed Algorithm

In this section, we proposed an enhanced method for improving the performance of k-means clustering algorithm. In the first step in algorithm 2, for each data point we calculate the Euclidean distance from origin. For the different data points as showed in Figure 1, we will get the same Euclidean distance from the origin. This will result in incorrect selection of data points as the initial centroids. To overcome this problem all data point's distances will be multiplied by its weightages. The weightage of a data point denotes the sum of attributes of that data point. The resulted weighted distance will be unique for

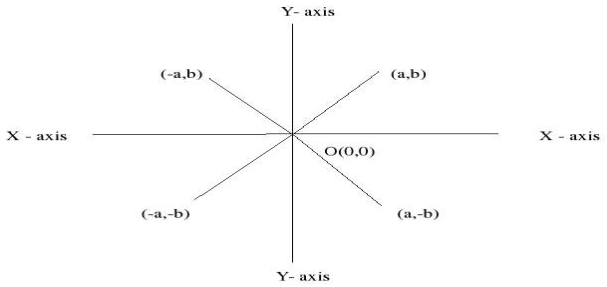

Figure 1. Data points in the two dimensional spaces

all the data points showed in Figure 1. In the next step, the weighted distances are sorted. Next, the original data points are sorted accordance with the sorted weighted distances.

After sorting the data points, select the k data points as initial centroids in systematic way with n/k difference position from one initial centroid to the next initial centroid in the sorted order. Next, we follow the original k-means procedure for assigning the data points to the appropriate clusters.

For each data point calculated the distance from all the initial centroids and the data point assigned to the cluster with the closest centroid. Then for each cluster centroids are recalculated by taking the mean of the data points of each cluster. After getting the new centroids, for each data point calculated the distance from all the centroids and the data point assigned to the cluster with the closest centroid. We repeat this process until no point changes clusters, or equivalently, until the centroids remain the same. The intitial centroids obtained by the proposed algorithm are lead to the better unique clustering results.

Algorithm 2: The enhanced method

Require: D = {d_1, d_2, d_3,..., d_i,..., d_n } // Set of n data points.

d_i = { x_1, x_2, x_3,..., x_l,..., x_m } // Set of attributes of one data point.

k // Number of desired clusters.

Ensure: A set of k clusters.

Steps:

1: For each data point calculate the Euclidean distance E_{di} from origin.

2: Calculate the weighted distance W_{di} for each data point d_i as follows:

$$W_{d_i} = \sum_{i=1}^{m} x_i * \mathrm{E}_{d_i}$$

3: Sort the weighted distances obtained in step 2.
Sort the data point's accordance with the weighted distances.

4: Select the data points, (n + k)/2k, (3n +k)/2k, …. ,
(n(2k -1) + k)/2k as the initial centroids with n/k difference position from one centroid to the next centroid in the sorted order.

5: Compute the distance between each data point d_i
(1 <= i <= n) to all the initial centroids cj (1 <= j <= k).

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

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

8: **Repeat**

9: Compute the distance between each data point d_i
(1 <= i <= n) to all the centroids cj (1 <=j <= k).

10: For each data point d_i, find the closest centroid cj
and assign di to cluster j.

11: For each cluster j (1 <= j <= k), recalculate the centroids.

Until the convergence criteria is met.

4. Time Complexity

The time complexity of the enhanced method is calculated as follows. In the first step for finding Euclidean distances from origin to all the data points can be done in the O(n) time, where n is the number of data points. To sort the weighted distances using heap sort it will take an O(nlogn) time in both average case and worst case. For assigning the data points to clusters the required time complexity is O(nkl), because the original k-means algorithm procedure is used for assigning the data points to the clusters. The enhanced method proposed in this paper finds the better initial centroids for the original k-means algorithm in less amount of time.

5. Results

We tested both the algorithms for the data sets with known clustering, Iris [6], Ecoli[6], New Thyroid [6], Breast Cancer Wisconsin(Original) [6], Height-Weight [7] and Echocardiogram [6]. The original k-means algorithm is executed 10 times for Iris data, 7 times for Ecoli, New Thyroid, Height-Weight, Echocardiogram data and 5 times for Breast Cancer Wisconsin(Original) data for the different sets of initial centroids. In each experiment the accuracy and time was computed and taken the average accuracy and time of all experiments. Table 1 shows the performance comparison of the algorithms. The results also showed with the help of bar charts in the figures 2, 3, 4, 5, 6 and 7. The results obtained show that the proposed algorithm is producing better clustering results compared to the k-means algorithm in less amount of time.

Data Set	Number of Clusters	Algorithm	Run	Accuracy (%)	Time Taken (sec)
Iris	K = 3	Orinal K-means	10	68.93	0.119
		Proposed Algorithm	1	89.33	0.110
Ecoli	K = 3	Orinal K-means	7	77.14	0.127
		Proposed Algorithm	1	91.91	0.118
New Thyroid	K = 3	Orinal K-means	7	73.15	0.120
		Proposed Algorithm	1	86.04	0.113
Height-Weight	K = 4	Orinal K-means	7	70.28	0.102
		Proposed Algorithm	1	92	0.088
Echocardiogram	K = 2	Original K-means	7	71.42	0.109
		Proposed Algorithm	1	82.24	0.102
Breast Cancer Wisconsin (Original)	K = 2	Orinal K-means	5	96.19	0.142
		Proposed Algorithm	1	96.19	0.136

Table 1. Performance Comparison of the Algorithms

6. Conclusion

One of the most well known partitional clustering algorithms is the *k*-means algorithm, but the original k-means algorithm does not produce the unique clustering results, because of the initial centroids are selected randomly. The results obtained showed that the proposed algorithm is producing better clusters compared to the k-means algorithm in less amount of time.

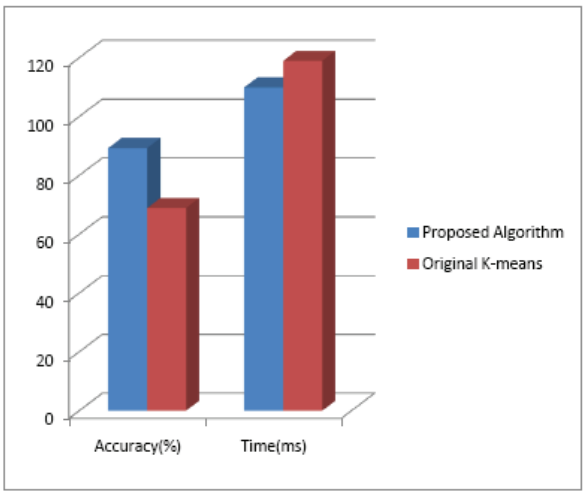

Figure 2. Performance Comparison chart for Iris data

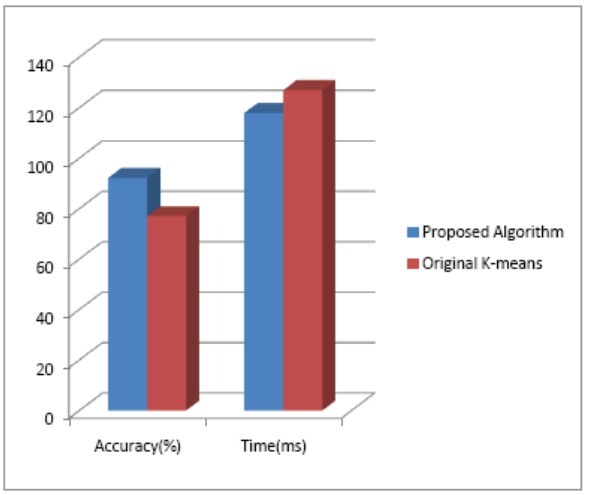

Figure 3. Performance Comparison chart for Ecoli data

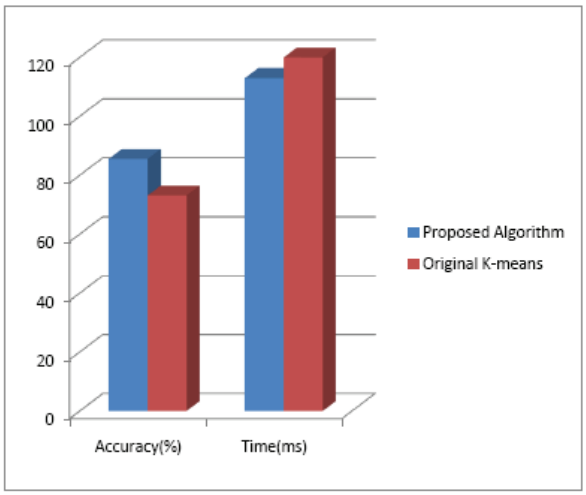

Figure 4. Performance Comparison chart for New Thyroid data

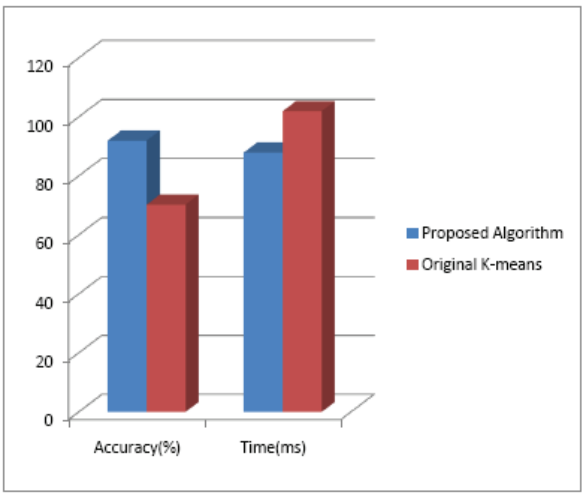

Figure 5. Performance Comparison chart for Height-Weight data

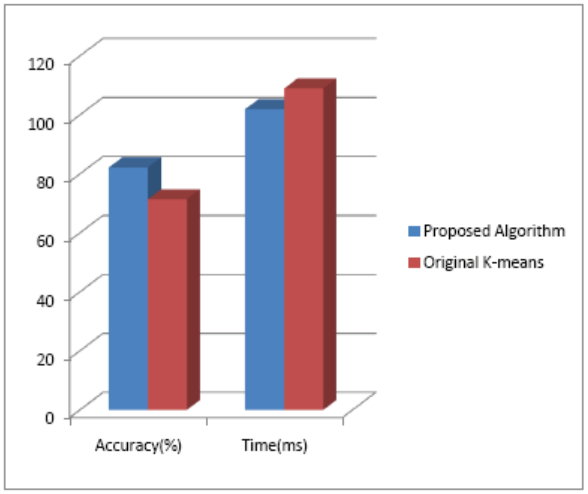

Figure 6. Performance Comparison chart for Echocardiogram data

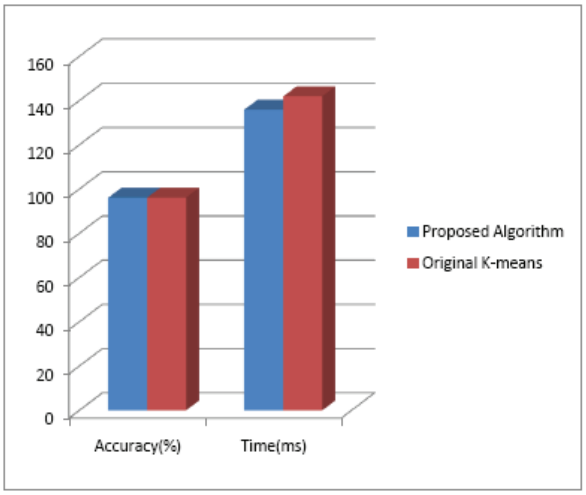

Figure 7. Performance Comparison chart for Breast Cancer Winconsin (original) data

References

[1] Zhang, Chen., Xia, Shixiong. (2009). K-means Clustering Algorithm with Improved Initial center, *In:* Second International Workshop on Knowledge Discovery and Data Mining (WKDD), p.790-792.

[2] Yuan, F Meng, Z H Zhangz, H X Dong, C R (2004). A New Algorithm to Get the Initial Centroids, *In*: Proc. of the 3rd International Conference on Machine Learning and Cybernetics, p. 26-29, August.

[3] Abdul Nazeer, K A., Sebastian, M P (2009). 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 Congresson Engineering (WCE-2009), V. 1, July 1-3, London, U.K.

[4] Dunham, Margaret H (2006). Data Mining-Introductory and Advanced Concepts, Pearson Education.

[5] Elmasri, Navathe, Somayajulu, Gupta,(2006). Fundamentals of Database Systems, Pearson Education, First edition.

[6] Iris, Ecoli, New Thyroid, Echocardiogram and Breast Cancer Wisconsin(Original) data sets are available at http:// archive.ics.uci.edu/ ml/ machine-learning-databases, (accessed on 18-3-10).

[7] Height-Weight Data available at http://www.disable-dworld.com/artman /publish / height-weight-teens.shtml, (accessed on 20-3-10).