Middle-East Journal of Scientific Research 20 (4): 485-491, 2014
ISSN 1990-9233
© IDOSI Publications, 2014
DOI: 10.5829/idosi.mejsr.2014.20.04.21038

An Efficient Enhanced K-Means Approach with Improved Initial Cluster Centers

[1]G. Sathiya and P. Kavitha

[1]Anna University, India
[2]Tagore Engineering College, Bharath University, Chennai, India

Abstract: Cluster analysis is one of the major data analysis methods and the k-means clustering algorithm is the major technique which is used for many practical applications. But the original k-means algorithm is computationally expensive and the final cluster is greatly depending upon the correctness of the initial centroids, which are selected randomly. Many improvements were already proposed to improve the performance of the k-means, but most of these require additional inputs like threshold values for the number of data points in a set. In this paper a new method is proposed for finding the better initial centroids and to provide an efficient way of assigning the data points to suitable clusters. It reduces the time complexity. This algorithm is easy to implement, which requires a simple data structure to keep some information in each iteration to be used in the next iteration.

Key words: Clustering · Data Mining · Data partitioning · Initial cluster centers · K-means clustering algorithm · Cluster analysis

INTRODUCTION

Advances in scientific data collection methods have resulted in the large scale accumulation of promising data pertaining to diverse fields of science and technology. Owing to the development of novel techniques for generating and collecting data, the rate of growth of scientific databases has become tremendous. Hence it is practically impossible to extract useful information from them by using conventional database analysis techniques. Effective mining methods are absolutely essential to unearth implicit information from huge databases [1].

CLUSTERING is an important tool for a variety of applications in data mining, statistical data analysis, data compression and vector quantization. Clustering is a division of data into groups of similar objects. Each group consists of objects that are similar between themselves and dissimilar to objects of other groups. From the machine learning perspective, Clustering can be viewed as unsupervised learning of concepts. Unsupervised machine learning means that clustering does not depend on predefined classes and training examples while classifying the data objects.

The requirements are given below that should be satisfied by clustering algorithms such as *scalability*, *high dimensionality*, *insensitivity* to order of attributes, *interoperability* and *usability*. 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 Partition algorithms [2]. A hierarchical clustering algorithm divides the given data set into smaller subsets in hierarchical fashion. A partition clustering algorithm partition the data set into desired number of sets in a single step. Numerous methods have been proposed to solve clustering problem. The most popular clustering method is k-means clustering algorithm developed by Mac Queen in 1967. The easiness of k-means clustering algorithm made this algorithm used in several fields.

The k-means clustering algorithm is more prominent since its intelligence to cluster massive data rapidly and efficiently. But the computational complexity of the original k-means algorithm is very high, especially for large data sets. Moreover, this algorithm results in different types of clusters depending on the random choice of initial centroids. The effectiveness of an algorithm depends on the definition of similarity

Corresponding Author: P. Kavitha, Tagore Engineering College, Bharath University - Chennai, India

(distance). Various methods have been proposed in the literature to enhance the accuracy and efficiency of the k-means clustering algorithm. This paper presents an enhanced method for finding the better initial centroids and to provide an efficient way of assigning the data points to suitable clusters with reduced time complexity [3].

This paper is organized as follows. Section 2 presents an overview of k-means algorithm. Section 3 describes a short analysis of the existing clustering methods. Section 4 introduces proposed method. Section 5 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 and future work [4].

The K-Means Clustering Algorithm: This section describes the original k-means clustering algorithm. The idea is to classify a given set of data into *k* number of disjoint clusters, where the value of *k* is fixed in advance. The general characteristics of this approach are given below:

- Each of the *k* clusters C_j is represented by the mean (or weighted average) c_j of its objects, the centroid.
- The clusters are iteratively recomputed to achieve stable centroids.

A popular measure for the “intra-cluster variance” is the *square-error criterion:*

$$E=\sum_{i=1}^{k}\sum_{p\in Ci}\|p-m_i\|^2$$

The algorithm consists of two separate phases: the first phase is to define *k* centroids, one for each cluster. The next phase is to take each point belonging to the given data set and associate it to the nearest centroid. Euclidean distance is generally considered to determine the distance between data points and the centroids. The Euclidean distance between two multi-dimensional data points *X = (x1, x2, x3... xm)* and *Y = (y1, y2, y3... ym)* is described as follows:

$$d(X.Y)=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+..+(x_m-y_m)^2}$$

When all the points are included in some clusters, the first step is completed and an early grouping is done. At this point we need to recalculate the new centroids, as the inclusion of new points may lead to a change in the cluster centroids. Once we find *k* new centroids, a new binding is to be created between the same data points and the nearest new centroid, generating a loop. As a result of this loop, the *k* centroids may change their position in a step by step manner. Eventually, a situation will be reached where the centroids do not move anymore. This signifies the convergence criterion for clustering. Pseudo code for the k-means clustering algorithm is listed as Algorithm 1.

Algorithm 1: The k-means clustering algorithm

```
Require: D = {d1, d2,...... dn} //set of n data items.
k // Number of desired clusters

Ensure: A set of k clusters.

Steps: Arbitrarily choose k data points from D as initial centroids;

Repeat:

    • Assign each point di to the cluster which has the closest centroid;
    • Calculate the new mean for each cluster;

Until: Convergence criteria is met.
```

It generates k points as initial centroids arbitrarily. Each point is then assigned to the cluster with the closest centroid. Then the centroid of each cluster is updated by taking mean of the data points of each cluster. Some data points may move from one cluster to other cluster. Again we calculate new centroids and assign the data points to the suitable clusters. We repeat the assignment and update the centroids, until convergence criteria is met i.e. no point changes clusters, or equivalently, until the centroids remain the same [5].

Although k-means has the great advantage of being easy to implement, it has some drawbacks. The quality of the final clustering results of the k-means algorithm highly depends on the arbitrary selection of 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. Moreover, the k-means algorithm is computationally very expensive also. The computational time complexity of the k-means algorithm is *O(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. So, the computational complexity of the k-means algorithm is rely on the number of data elements, number of clusters and number of iterations [6].

Related Work: The original k-means algorithm is very impressionable to the initial starting points. So, it is quite crucial for k-means to have refine initial cluster centers. Several methods have been proposed in the literature for finding the better initial centroids. And some methods were proposed to improve both the accuracy and efficiency of the k-means clustering algorithm. In this paper, some of the more recent proposals are reviewed.

A. M. Fahim *et al*. [1] proposed an enhanced method for assigning data points to the suitable clusters. It was proposed especially for dataset containing large number of clusters. It discovers the spherical shaped cluster, whose center is the gravity center of points in that cluster, this center moves as new points are added to or removed from it. In this approach, there are two functions described. The first function finds the nearest center for each data point, by computing the distances to the *k* centers and for each data point keeps its distance to the nearest center. The second function is used for assigning the data points into the appropriate cluster by using the *mean squared error value*, *Euclidean distance value*. In Fahim approach the required computational time is reduced when assigning the data elements to the appropriate clusters. But in this method the initial centroids are selected randomly. So this method is very sensitive to the initial starting points and it does not promise to produce the unique clustering results. The time complexity of this enhanced algorithm is O(nk), not O(nkl). K. A. Abdul Nazeer *et al*. [2] proposed an enhanced algorithm to improve the accuracy and efficiency of the k-means clustering algorithm. In this algorithm, two methods are used, one method for finding the better initial centroids. And another method for an efficient way for assigning data points to appropriate clusters with reduced time complexity. In the first method, the initial centroids are determined systematically so as to produce clusters with better accuracy. The second phase makes use of a variant of the clustering methods. It starts by forming the initial clusters based on the relative distance of each data-point from the initial centroids. These clusters are subsequently fine-tuned by using a heuristic approach, thereby improving the efficiency. Here the Euclidean distance is used to find the distance between the centroids and data point. This algorithm produces good clusters in less amount of computational time. The time complexity of this enhanced approach is O(nk), not O(nkl).

Zhang Chen *et al*. [3] proposed the initial centroids algorithm based on k-means that have avoided alternative randomness of initial center.

Fang Yuan [4] 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*. [5] 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.

S. Deelers *et al*. [6] proposed an algorithm follows a novel approach that performs data partitioning along the data axis with the highest variance. This algorithm is mainly focus on the determination of the initial cluster centers. Data in a cell is partitioned using a cutting plane that divides cell in two smaller cells, where inter-cluster distances are large as possible and intra-cluster distances are small as possible. The plane is perpendicular to the data axis with the highest variance and is designed to reduce the sum squared errors of the two cells as much as possible, while at the same time keep the two cells far apart as possible. Cells are partitioned one at a time until the number of cells equals to the predefined number of clusters, *K*. The centers of the *K* cells become the initial cluster centers for *K*-means. The approach has been used successfully for color quantization.

A. Bhattacharya *et al*. [8] proposed a novel clustering algorithm, called Divisive Correlation Clustering Algorithm (DCCA) for grouping of genes having similar pattern of variation in their expression values, without taking the expected number of clusters as an input. To detect clusters with high correlation and biological significance, we use the correlation clustering concept. The algorithm continues clustering until all clusters contain only positively correlated sets of genes. Initially, consider all the genes in one cluster. If no repulsion exists between a pair of genes inside any cluster then stop the process. Otherwise identify a cluster C for which a pair of genes xi, xj have the most negative repulsion value among all the clusters. Then replace the clusters and increase the number of clusters by one. Finally place the genes into the corresponding clusters and find the average correlation value. If there is no change, remain the gene as it is. DCCA is able to produce clusters, without taking the initial centroids and the value of k, the number of desired clusters as an input. The time complexity of the algorithm is high and the cost for repairing from any misplacement is also high.

Enhanced K-Means Algorithm: In this paper we proposed a new approach for finding the better initial centroids with reduced time complexity. For assigning the data points we follow the paper [1, 2]. The pseudo code for the proposed algorithm is outlined as Algorithm 2.

Algorithm 2: The Enhanced method

Require:
D = {d1, d2,...... dn} // set of *n* data items
k // Number of desired clusters

Ensure:
A set of *k* clusters.

Steps:
Phase 1: Determine the initial centroids of the clusters by using Algorithm 3.
Phase 2: Assign each data point to the appropriate clusters by using Algorithm 4.

In the first phase, we are checking the given data set contain the negative value attributes or not. If the data set contains the negative value attributes then we are transforming the all data points in the data set to the positive space by subtracting the each data point attribute with the minimum attribute value in the given data set. Here, the transformation is required, because in the proposed algorithm we calculate the distance from origin to each data point in the data set. So, for the different data points as showed in Fig. 1, we will get the same Euclidean distance from the origin. This will result in incorrect selection of the initial centroids. To overcome this problem all the data points are transformed to the positive space. Then for all the data points as showed in Fig. 1, we will get the unique distances from origin. If data set contains the all positive value attributes then the transformation is not required. The two phases of the enhanced method are described below as Algorithm 3 and Algorithm 4.

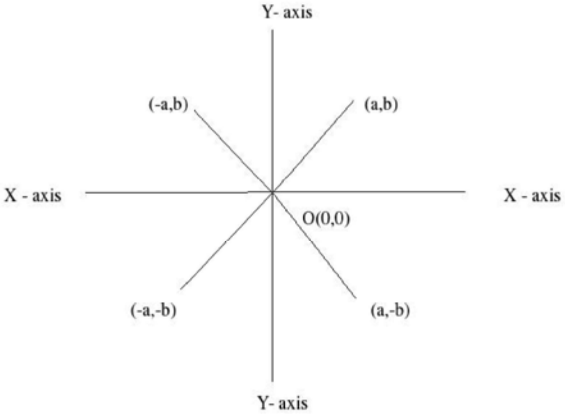

Fig. 1: Data points in Two Dimensional Space

In the next step, for each data point we calculate the distance from origin. 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. These initial centroids lead to the better unique clustering results. Next, for each data point the distance calculated from all the initial centroids. The next stage is an iterative process which makes use of a heuristic approach to reduce the required computational time.

Algorithm 3: Finding the initial centroids

Require:
D = {d1, d2,...... dn} // set of *n* data items
k // Number of desired clusters
Ensure:
A set of *k* clusters.

Steps:

- In the given data set *D*, if the data points contain the both positive and negative attribute values then go to step 2, otherwise go to step 4.
- Find the minimum attribute value in the given data set *D*.
- For each data point attribute, subtract with the minimum attribute value.
- For each data point calculate the distance from origin.
- Sort the distances obtained in step 4. Sort the data points accordance with the distances.
- Partition the sorted data points into k equal sets.
- In each set, take the middle point as the initial centroid.

- Compute the distance between each data point *di* *(1 <= i<= n)* to all the initial centroids *cj (1 <= j <= k)*.
- *Repeat*In the next phase*,* the data points are assigned to the clusters having the closest centroids in the next step. ClusterId of a data point denotes the cluster to which it belongs. NearestDist of a data point denotes the present nearest distance from closest centroid. Next, we have to recalculate the centroids, for each cluster.

Algorithm 4: Assigning data-points to clusters

Require:
D = {d1, d2,......dn} // set of *n* data-points.
c_j //Initial centroid.

Ensure: A set of *k* clusters.

Steps:

- For each data point *di*, find the closest centroid *cj* and assign *di* to cluster *j*.
- Set ClusterId[*i*]=*j*. // *j*:Id of the closest cluster.
- Set NearestDist[*i*]= *d(di, cj)*.
- For each cluster *j (1 <= j <= k)*, recalculate the centroids.
- For each data point *di*,

Compute its distance from the centroid of the present nearest cluster.

If this distance is less than or equal to the present nearest distance, the data point stays in the same cluster.
Else

For every centroid *cj (1<=j<=k)* compute the distance *d(di, cj)*.

End for;
Until the convergence criteria is met.

Then compute its distance from the centroid of the present nearest cluster for each data point. If this distance is less than or equal to the present nearest distance, the data point stays in the same cluster. Otherwise compute the distance for every centroids.

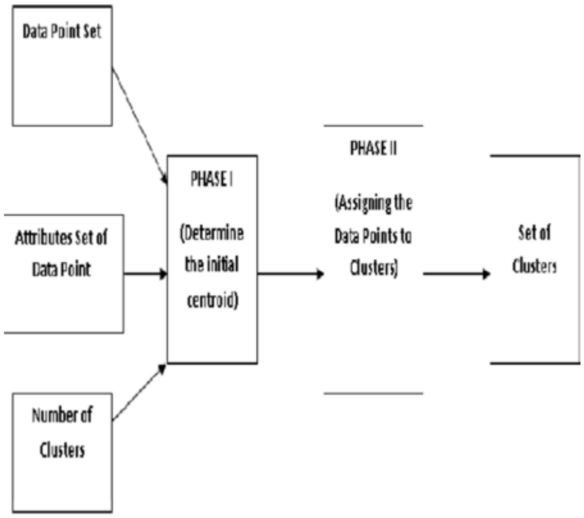
Data Point Set
Attributes Set of Data Point
Number of Clusters
PHASE I (Determine the initial centroid)
PHASE II (Assigning the Data Points to Clusters)
Set of Clusters

Fig. 2: Enhanced k-means Algorithm

Time Complexity: The required time complexity of proposed algorithm for finding the initial centroids is *O(nlogn)* in both average and worst case, where n is the number of data points. The sorting method used for sorting determines the overall time complexity for finding the initial centroids. Since the proposed enhanced method uses heap sort, its overall time complexity becomes *O(nlogn)* in both average and worst case. To get the initial clusters the required time complexity is *O(nk)*. Here, some data points stay in the cluster itself and some other data points move to other clusters based on their relative distance from old centroid and the new centroid.

If the data point stays in the same cluster then the required complexity is *O(*1*)*, otherwise *O(k)*. In each iteration the moving of data points to other clusters is decreases. Assuming, until the convergence criteria is met, half the data points move to the other clusters from their present clusters, this requires *O(nk/2)*. Hence the total time complexity for assigning the data points is *O(nk)*, not *O(nkl)*. Therefore the total time complexity of the proposed algorithm becomes *O(nlogn)*. Hence the proposed algorithm has less time complexity compared to the original k-means clustering algorithm.

RESULTS

We tested both the algorithms for the data sets with known clustering, Iris [11], New Thyroid [11] and Height-Weight [12-17]. The same data sets are used as an input for the original k-means algorithm. Both the algorithms need number of clusters as an input. In additional, for the original k-means algorithm the set of initial centroids also required. The enhanced method finds initial centroids systematically. The enhanced method requires only the data values and number of clusters as inputs. And it does not take any additional inputs like threshold values.

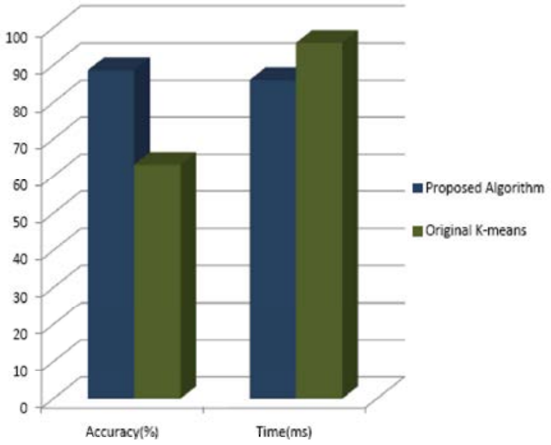

Fig. 3: Performance Comparison chart for Iris Data

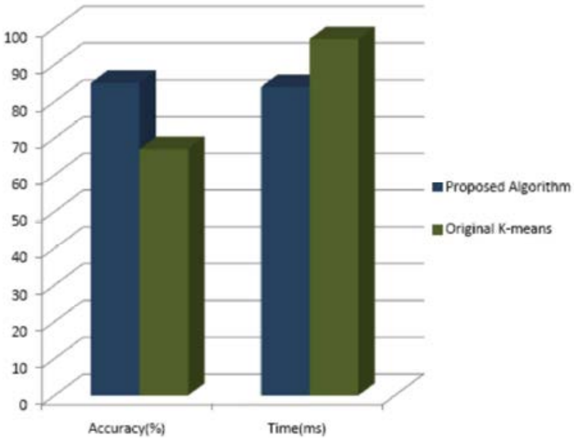

Fig. 4: Performance Comparison chart for New Thyroid Data

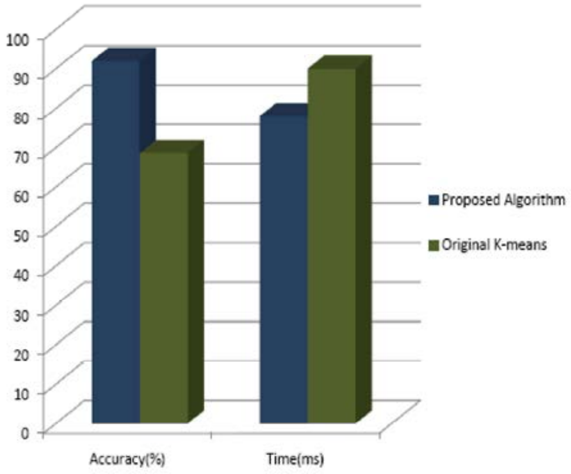

Fig. 5: Performance Comparison chart for Height-weight Data

CONCLUSION

The k-means algorithm is widely used for clustering large sets of data. But the standard algorithm do not always guarantee good results as the accuracy of the final clusters depend on the selection of initial centroids. Moreover, the computational complexity of the standard algorithm is objectionably high owing to the need to reassign the data points a number of times, during every iteration of the loop.

The proposed algorithm is found to be more accurate and efficient compared to the original k-means algorithm. This proposed method finding the better initial centroids and provides an efficient way of assigning the data points to the suitable clusters. This method ensures the total mechanism of clustering in O(nlogn) time without loss the correctness of clusters. This approach does not require any additional inputs like threshold values. The proposed algorithm produces the more accurate unique clustering results. The value of k, desired number of clusters is still required to be given as an input to the proposed algorithm. Automating the determination of the value of k is suggested as a future work.

REFERENCES

1. Fahim, A.M., A.M. Salem, F.A. Torkey and M.A. Ramadan, 2006. An Efficient enhanced k-means clustering algorithm, journal of Zhejiang University, 10(7): 6261633.
2. Abdul Nazeer, K.A. and M.P. Sebastian, July 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 Congress on Engineering (WCE-2009), London, UK. pp: 1.
3. Chen Zhang and Shixiong Xia, K-means 2009. Clustering Algorithm with Improved Initial center, in Second International Workshop on Knowledge Discovery and Data Mining (WKDD), pp: 790-792.
4. Yuan, F., Z.H. Meng, H.X. Zhangz, C.R. Dong, August 2004. A New Algorithm to Get the Initial Centroids, proceedings of the 3rd International Conference on Machine Learning and Cybernetics, pp: 26-29.

5. Koheri Arai and Ali Ridho Barakbah, 2007. Hierarchical K-means, an algorithm for Centroids initialization for k-means, department of information science and Electrical Engineering Politechnique in Surabaya, Faculty of Science and Engineering, Saga University, 36(1).
6. Deelers, S. and S. Auwatanamongkol, Enhancing K-Means Algorithm with Initial Cluster Centers Derived from Data Partitioning along the Data Axis with the Highest Variance, International Journal of Computer Science, 2(4).
7. Mc Queen, J., 1967. Some methods for classification and analysis of multivariate observations, Proc. 5th Berkeley Symp. Math. Statist. Prob., 1: 281-297.
8. Bhattacharya, A. and R.K. De, 2008. Divisive Correlation Clustering Algorithm (DCCA) for grouping of genes: detecting varying patterns in expression profiles, Bioinformatics, 24: 1359-1366.
9. Margaret H. Dunham, 2006. Data Mining-Introductory and Advanced Concepts, Pearson Education.
10. Elmasri, Navathe and Somayajulu, Gupta, 2006. Fundamentals of Database Systems, Pearson Education, First edition.
11. 2010. The UCI Repository website. [Online]. Available: http://archive.ics.uci.edu/.
12. Height-Weight Data, 2010. [Online]. Available: http://www.disabledworld. com/artman/ publish/ height-weight-teens.shtml.
13. Naseer Ahmed, 2013. Ultrasonically Assisted Turning: Effects on Surface Roughness World Applied Sciences Journal, 27(2): 201-206.

14 Tatyana Nikolayevna Vitsenets, 2014. Concept and Forming Factors of Migration Processes Middle-East Journal of Scientific Research, 19(5): 620-624.

15. Shafaq Sherazi and Habib Ahmad, 2014. Volatility of Stock Market and Capital Flow Middle-East Journal of Scientific Research, 19(5): 688-692.
16. Kishwar Sultana, Najm ul Hassan Khan and Khadija Shahid, 2013. Efficient Solvent Free Synthesis and X Ray Crystal Structure of Some Cyclic Moieties Containing N-Aryl Imide and Amide,Middle-East Journal of Scientific Research, 18(4): 438-443.
17. Pattanayak Monalisa and P.L. Nayak, 2013. Green Synthesis of Gold Nanoparticles Using Elettaria cardamomum (ELAICHI) Aqueous Extract World Journal of Nano Science & Technology, 2(1): 01-05.