An Improvement in K-means Clustering Algorithm

Anand Sutariya[1], Prof. Kiran Amin[2]
[1]PG Student, U.V.Patel College of Engineering, Ganpat University, Mehsana, Gujarat
[2]Head, CE Dept., U.V.Patel College of Engineering, Ganpat University, Mehsana, Gujarat

Abstract

Data Mining is the process of extracting information from large data sets through the use of algorithms and techniques drawn from the field of Statistics, Machine Learning and Data Base Management Systems. Cluster analysis is one of the major data analysis methods and the k-means clustering algorithm is widely used for many practical applications. But the original k-means algorithm is computationally expensive and the quality of the resulting clusters heavily depends on the selection of initial centroids. Several methods have been proposed for improving the performance of the k-means clustering algorithm. But still there are many problems in original k-means algorithm. So, we have proposed the improved algorithm of k-means for improving the performance of the algorithm.

1. Introduction

Data mining is a process that uses a variety of data analysis tools to discover patterns and relationships in data that may be used to make valid predictions.

Clustering is the grouping of similar objects and a cluster of a set is a partition of its elements that is chosen to minimize some measure of dissimilarity [1]. Unlike classification which is a supervised learning technique, clustering is a type of unsupervised learning. In clustering, objects in the dataset are grouped into clusters, such that groups are very different from each other and the objects in the same group are very similar to each other. In this case, clusters are not predefined which means that result clusters are not known before the execution of clustering algorithm. These clusters are extracted from the dataset by grouping the objects in it. For some algorithms, number of desired clusters is supplied to the algorithm, whereas some others determine the number of groups themselves for the best clustering result. Clustering of a dataset gives information on both the overall dataset and characteristics of objects in it [2].

1.1 Partitioning Methods

Given a database of n objects or data tuples, a partitioning method constructs k partitions of the data, where each partition represents a cluster. That is, it classifies the data into k groups, which together satisfy the following requirements: (1) each group must contain at least one object, and (2) each object must belong to exactly one group. Given k, the number of partitions to construct, a partitioning method creates an initial partitioning. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. The general criterion of a good partitioning is that objects in the same cluster are "close" or related to each other, whereas objects of different clusters are "far apart" or very different. There are various kinds of other criteria for judging the quality of partitions.

1.2 Hierarchical Methods

A hierarchical method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed. The agglomerative approach, also called the bottom-up approach, starts with each object forming a separate group. It successively merges the objects or groups that are close to one another, until all of the groups are merged into one (the topmost level of the hierarchy), or until a termination condition holds. The divisive approach, also called the top-down approach, starts with all of the objects in the same cluster. In each successive iteration, a cluster is split up into smaller clusters, until eventually each object is in one cluster, or until a termination condition holds.

1.3 K-means Algorithm

K-means is a data mining algorithm which performs clustering. As mentioned previously, clustering is dividing a dataset into a number of groups such that similar items fall into same groups [1]. Clustering uses unsupervised learning technique which means that result clusters are not known before the execution of clustering algorithm unlike the case in classification. Some clustering algorithms takes the number of desired clusters as input while some others decide the number of result clusters themselves.

K-means algorithm uses an iterative procedure in order to cluster database [3]. It takes the number of desired clusters and the initial means as inputs and produces final means as output. Mentioned initial and final means are the means of clusters. If the algorithm is required to produce K clusters then there will be K initial means and K final means. In completion, K-means algorithm produces K final means which answers why the name of algorithm is K-means.

After termination of K-means clustering, each object in dataset becomes a member of one cluster. This cluster is determined by searching throughout the means in order to find the cluster with nearest mean to the object. Shortest distanced mean is considered to be the mean of cluster to which examined object belongs. K-means algorithm tries to group the items in dataset into desired number of clusters. To perform this task it makes some iteration until it converges. After each iteration, calculated means are updated such that they become closer to final means. And finally, the algorithm converges and stops performing iterations.

Steps of Algorithm:

Input:

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

Output: A set of *k* clusters.

Steps:

```
    1. Arbitrarily choose k data-items from D as
    initial  centroids;
    2. Repeat
        Assign each item di to the cluster which
        has the closest centroid;
        Calculate new mean for each cluster;
    Until convergence criteria is met.
```

2. Related Work

Fang Yuan et al. [4] proposed a systematic method for finding the initial centroids. The centroids obtained by this method are consistent with the distribution of data. Hence it produced clusters with better accuracy, compared to the original k- means algorithm. However, Yuan's method does not suggest any improvement to the time complexity of the k-means algorithm.

Sun Shibao et al. [5] proposed an improved k-means algorithm based on weights. This is a new partitioning clustering algorithm, which can handle the data of numerical attribute, and it also can handle the data of symbol attribute. Meanwhile, this method reduces the impact of isolated points and the "noise", so it enhances the efficiency of clustering. However, this method has no improvement on the complexity of time.

K-means algorithm is a popular partition algorithm in cluster analysis, which has some limitations when there are some restrictions in computing resources and time, especially for huge size dataset. Yu-Fang Zhang et al. [6] proposed the improved K-means algorithm is a solution to handle large scale data. However, this method has no improvement on the complexity of time.

K-means algorithm has higher efficiency and scalability and converges fast when dealing with large data sets. However it also has many deficiencies: the number of clusters K needs to be initialized, the initial cluster centers are arbitrarily selected, and the algorithm is influenced by the noise points. In view of the shortcomings of the traditional K-means clustering algorithm, Juntao Wang et al. [7] proposed an improved K-means algorithm using noise data filter. The algorithm developed density-based detection methods based on characteristics of noise data where the discovery and processing steps of the noise data are added to the original algorithm. However, this method has no improvement on the complexity of time.

The k-means algorithm is well known for its efficiency in clustering large data sets. However, working only on numeric values prohibits it from being used to cluster real world data containing categorical values. Zhexu Huang [10] proposed two algorithms which extend the k-means algorithm to categorical domains and domains with mixed numeric and categorical values. However, this method has no improvement on the complexity of time as well as final cluster.

3. Proposed Solution

Based on above related work we can figure out that all the algorithms are suffers in term of time complexity as well as accuracy.

So, we have proposed new Improved K-means algorithm which can handle the above problem very well.

Improved K-means Algorithm

Steps of Algorithm:

Input:

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

Output: A set of k clusters.

Steps:

Phase 1: Determine the initial centroids of the clusters by using Phase 1.

Phase 2: Assign each data point to the appropriate clusters by using Phase 2

In the first phase, the initial centroids are determined systematically so as to produce clusters with better accuracy. The second phase assigns each data point to the appropriate clusters. The two phases of the improved method are described below as Phase 1 and Phase 2.

Phase 1: Finding the initial centroids.

Input:

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

Output: A set of k initial centroids.

Steps:

1. Set m = 1;
2. Compute the distance between each data point and all other data- points in the set D;
3. Find the closest pair of data points from the set D and form a data-point set Am (1<= m <= k) which contains these two data- points, Delete these two data points from the set D;
4. Find the data point in D that is closest to the data point set Am, Add it to Am and delete it from D;
5. Repeat step 4 until the number of data points in Am reaches 0.75*(n/k);
6. If m<k, then m = m+1, find another pair of data points from D between which the distance is the shortest, form another data-point set Am and delete them from D, Go to step 4;
7. For each data-point set Am (1<=m<=k) find the arithmetic mean of the vectors of data points in Am, these means will be the initial centroids.

Phase 1 [4] describes the method for finding initial centroids of the clusters. Initially, compute the distances between each data point and all other data points in the set of data points. Then find out the closest pair of data points and form a set A1 consisting of these two data points, and delete them from the data point set D. Then determine the data point which is closest to the set A1, add it to A1 and delete it from D. Repeat this procedure until the number of elements in the set A1 reaches a threshold. At that point go back to the second step and form another data-point set A2. Repeat this till ’k’ such sets of data points are obtained. Finally the initial centroids are obtained by averaging all the vectors in each data-point set. The Euclidean distance is used for determining the closeness of each data point to the cluster centroids. The distance between one vector X = (x_1, x_2, ...x_n) and another vector Y = (y_1, y_2, …….y_n) is obtained as,

$$d(x,y) = \sqrt{(x1-y1)^2 + (x2-y2)^2 + \cdots + (xn-yn)^2}$$

The distance between a data point X and a data-point set D is defined as

d(X, D) = min (d (X, Y), where Y∈ D).

The initial centroids of the clusters are given as input to the second phase, for assigning data-points to appropriate clusters. The steps involved in this phase are outlined as Phase 2.

Phase 2: Assigning data points to clusters.

```
Input:

    D = {d1, d2,......,dn} // set of n data-points.
    C = {c1, c2,.......,ck} // set of k centroids

Output: A set of k clusters

Steps:

    1. Compute the distance of each data-point
       di (1<=i<=n) to all the centroids
       cj (1<=j<=k) as d(di, cj);

    2. For each data-point di, find the closest
       centroid cj and assign di to cluster j.

    3. Set ClusterId[i] = j; // j: Id of the closest
       cluster

    4. Set Nearest Dist[i] = d(di, cj);

    5. For each cluster j (1<=j<=k), recalculate
       the centroids;

    6. Repeat

    7. For each data-point di,

       7.1 Compute its distance from the centroid
           of the present nearest cluster;
       7.2 If this distance is less than or equal to
           the present nearest distance, the
           datapoint stays in the cluster;
       Else
            7.2.1 For every centroid cj (1<=j<=k)
            Compute the distance d(di, cj);
            Endfor;
            7.2.2 Assign the data-point di to the
            cluster with the nearest centroid cj.
            7.2.3 Set ClusterId[i] = j;
            7.2.4 Set Nearest_Dist[i] = d(di, cj);

     Endfor;
    8. For each cluster j (1<=j<=k), recalculate
       the centroids; until the convergence
       criteria is met.
```

The first step in Phase 2 is to determine the distance between each data-point and the initial centroids of all the clusters. The data-points are then assigned to the clusters having the closest centroids. This results in an initial grouping of the data-points. For each data-point, the cluster to which it is assigned (ClusterId) and its distance from the centroid of the nearest cluster (Nearest_Dist) are noted. Inclusion of data-points in various clusters may lead to a change in the values of the cluster centroids. For each cluster, the centroids are recalculated by taking the mean of the values of its data-points. Up to this step, the procedure is almost similar to the original k-means algorithm except that the initial centroids are computed systematically.

The next stage is an iterative process which makes use of a heuristic method to improve the efficiency. During the iteration, the data-points may get redistributed to different clusters. The method involves keeping track of the distance between each data-point and the centroid of its present nearest cluster. At the beginning of the iteration, the distance of each data-point from the new centroid of its present nearest cluster is determined. If this distance is less than or equal to the previous nearest distance, that is an indication that the data point stays in that cluster itself and there is no need to compute its distance from other centroids. These results in the saving of time required to compute the distances to k-1 cluster centroids. On the other hand, if the new centroid of the present nearest cluster is more distant from the data-point than its previous centroid, there is a chance for the data-point getting included in another nearer cluster. In that case, it is required to determine the distance of the data-point from all the cluster centroids. Identify the new nearest cluster and record the new value of the nearest distance. The loop is repeated until no more data-points cross cluster boundaries, which indicates the convergence criterion. The heuristic method described above results in significant reduction in the number of computations and thus improves the efficiency.

4. Experiment and Results

Improved algorithm of the K-means has been designed and implemented in this project for the purpose of improvement of K-means algorithm in execution time and to get the better accuracy. Original K-means algorithm has also been implemented for the purpose of comparison with Improved K-means algorithm in time and accuracy. Both implementations have been tested on the same environment which is JAVA Programming Language.

The multivariate and sequential Libras Movement dataset has 360 instances and 90 attributes, iris dataset has 150 instances and 4 attributes, wine quality dataset has 6497 instances and 12 attributes, blood transfusion dataset has 748 instances and 5 attributes and TAE(Teaching Assistant Evaluation) dataset has 151 instances and 6 attributes. All these datasets taken from the UCI repository of machine learning database [9], is used for testing the efficiency and accuracy of the improved algorithm. The same data set is given as input to the original k-means algorithm and the improved algorithm.

The results of experiments are tabulated in Table 1 and Table 2. In Table 1 we compare both the algorithm for efficiency in term of time for all the datasets. In Table 2 we compare both the algorithms for accuracy for all datasets.

Table 1. Comparison of efficiency of two algorithms

Algorithms Datasets		K-means	Improved K-means
		Centroid Selected randomly	Centroid computed by program
Libras Movement	Time taken in (ms)	478	261
Iris		20	10
TAE		20	02
Wine quality		426	112
Blood Transfusion		35	05

Table 2. Comparison of accuracy of two algorithms

Algorithms Datasets		K-means	Improved K-means
		Centroid Selected randomly	Centroid computed by program
Libras Movement	Accuracy (in %)	59	61
Iris		66	90
TAE		58	61
Wine quality		26	30
Blood Transfusion		59	64

Figure 1 and Figure 2 depicts the performances of the standard K-means algorithm and the Improved K-means algorithm in terms of the efficiency and accuracy for all the datasets. It can be seen from the results that the improved algorithm significantly outperforms the standard K-means algorithm in terms of efficiency and accuracy.

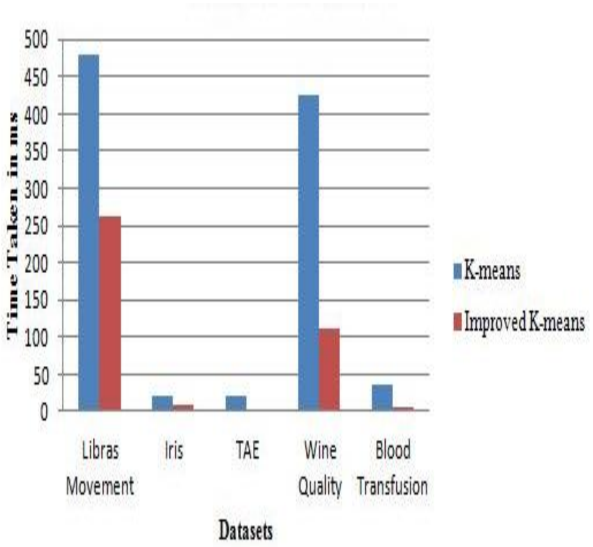

Figure 1. Comparison of efficiency of two algorithms

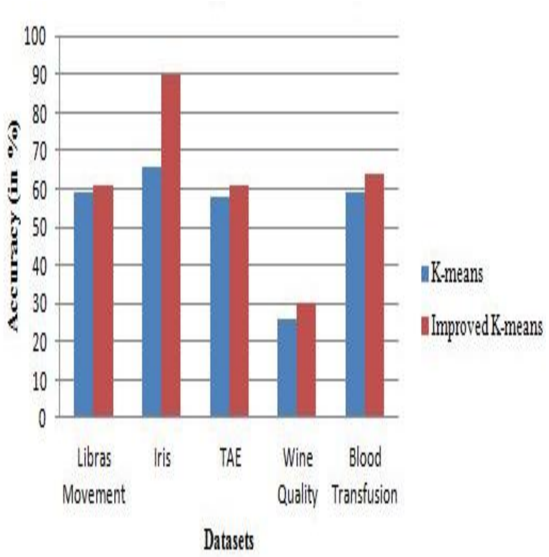

Figure 2. Comparison of accuracy of two algorithms

5. Conclusion

The original k-means algorithm is widely used for clustering large sets of data. But it does not always guarantee for good results, as the accuracy of the final clusters depend on the selection of initial centroids.

Moreover, the computational complexity of the original algorithm is very high because it reassigns the data points a number of times during every iteration of the loop. Here we presents an improved k-means algorithm which combines a systematic method for finding initial centroids and an efficient way for assigning data points to clusters. This method ensures the entire process of clustering in time without sacrificing accuracy of clusters.

6. References

[1] S. Kantabutra, “Parallel K-means Clustering Algorithm on NOWs”, Department of Computer Science, Tufts University, 1999

[2] Berkhin Pavel, “A Survey of Clustering Data Mining Techniques”, Springer Berlin Heidelberg, 2006.

[3] R. Ali, U. Ghani, A. Saeed, **“**Data Clustering and Its Applications”, Rudjer Boskovic Institute, 2001

[4] Yuan F, Meng Z. H, Zhang H. X and Dong C. R, “A New Algorithm to Get the Initial Centroids,” Proc. of the 3rd International Conference on Machine Learning and Cybernetics, pages 26–29, August 2004.

[5] Sun Shibao, Qin Keyun,”Research on Modified k-means Data Cluster Algorithm”, Computer Engineering, vol.33, No.13, pp.200–201, July 2007.

[6] Yu-Fang Zhang, Jia-li mao and Zhong-Yang Xiong, “AN EFFICIENT CLUSTERING ALGORITHM”, Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi’an, 2-5 November 2003

[7] Juntao Wang and Xiaolong Su, “An improved K-means clustering algorithm