Improving the Accuracy and Efficiency of the k-means Clustering Algorithm

K. A. Abdul Nazeer, M. P. Sebastian

Abstract— **Emergence of modern techniques for scientific data collection has resulted in large scale accumulation of data pertaining to diverse fields. Conventional database querying methods are inadequate to extract useful information from huge data banks. 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 in the literature for improving the performance of the k-means clustering algorithm. This paper proposes a method for making the algorithm more effective and efficient, so as to get better clustering with reduced complexity.**

***Index Terms*—Data Analysis, Clustering, k-means Algorithm, Enhanced k-means Algorithm**

I. 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.

Cluster analysis [6] is one of the major data analysis methods which is widely used for many practical applications in emerging areas like Bioinformatics [1, 3]. Clustering is the process of partitioning a given set of objects into disjoint clusters. This is done in such a way that objects in the same cluster are similar while objects belonging to different clusters differ considerably, with respect to their attributes.

The k-means algorithm [6, 7, 8, 10, 11] is effective in producing clusters for many practical applications. 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. Several attempts were made by researchers for improving the performance of the k-means clustering algorithm. This paper deals with a method for improving the accuracy and efficiency of the k-means algorithm.

II. 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 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. 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. Pseudocode for the k-means clustering algorithm is listed as Algorithm 1 [7].

Algorithm 1: The k-means clustering 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.
```

Manuscript received December 9, 2008.

K. A. Abdul Nazeer is with National Institute of Technology Calicut, Kozhikode, India- 673 601. (phone: 91-495-2286818; fax: 91-495-2287250; e-mail: nazeer@ nitc.ac.in).

M. P. Sebastian is with National Institute of Technology Calicut, Kozhikode, India- 673 601. (e-mail: sebasmp@ nitc.ac.in).

The k-means algorithm is the most extensively studied clustering algorithm and is generally effective in producing good results. The major drawback of this algorithm is that it produces different clusters for different sets of values of the initial centroids. Quality of the final clusters heavily depends on the selection of the initial centroids. The k-means algorithm is computationally expensive and requires time proportional to the product of the number of data items, number of clusters and the number of iterations.

III. RELATED WORK

Several attempts were made by researchers to improve the effectiveness and efficiency of the k-means algorithm [4, 5, 12]. A variant of the k-means algorithm is the k-modes [2, 5] method which replaces the means of clusters with modes. Like the k-means method, the k-modes algorithm also produces locally optimal solutions which are dependent on the selection of the initial modes. The k-prototypes algorithm [5] integrates the k-means and k-modes processes for clustering the data. In this method, the dissimilarity measure is defined by taking into account both numeric and categorical attributes.

As shown in Algorithm 1, the original k-means algorithm consists of two phases: one for determining the initial centroids and the other for assigning data points to the nearest clusters and then recalculating the cluster means. The second phase is carried out repetitively until the clusters get stabilized, i.e., data points stop crossing over cluster boundaries.

Fang Yuan et al. [12] 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.

Fahim A M et al. [4] proposed an efficient method for assigning data-points to clusters. The original k-means algorithm is computationally very expensive because each iteration computes the distances between data points and all the centroids. Fahim's approach makes use of two *distance* functions for this purpose- one similar to the k-means algorithm and another one based on a heuristics to reduce the number of distance calculations. But this method presumes that the initial centroids are determined randomly, as in the case of the original k-means algorithm. Hence there is no guarantee for the accuracy of the final clusters.

IV. MODIFIED APPROACH

In the enhanced clustering method discussed in this paper, both the phases of the original k-means algorithm are modified to improve the accuracy and efficiency. The enhanced method is outlined as Algorithm 2.

Algorithm 2: The enhanced method

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 Algorithm 3.

Phase 2: Assign each data point to the appropriate clusters by using Algorithm 4.

In the first phase, the initial centroids are determined systematically so as to produce clusters with better accuracy [12]. The second phase makes use of a variant of the clustering method discussed in [4]. 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. The two phases of the enhanced method are described below as Algorithm 3 and Algorithm 4.

Algorithm 3: 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 datapoint 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 datapoints 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.

Algorithm 3 describes the method for finding initial centroids of the clusters [12]. 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 = (x1, x2, ...x_n) and another vector Y = (y1, y2, …….y_n) is obtained as

$$d(X,Y) = \sqrt{(x1-y1)^2 + (x2-y2)^2 + ... + (x_n - y_n)^2}$$

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

$d(X, D) = \min (d (X, Y), \text{where } Y \in D)$.

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 Algorithm 4.

Algorithm 4: 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 data-point 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. This 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.

V. TIME COMPLEXITY

Phase 1 of the enhanced algorithm requires a time complexity of $O(n^2)$ for finding the initial centroids, as the maximum time required here is for computing the distances between each data point and all other data-points in the set D. In the original k-means algorithm, before the algorithm converges the centroids are calculated many times and the data points are assigned to their nearest centroids. Since complete redistribution of the data points takes place according to the new centroids, this takes *O(nkl)*, where *n* is the number of data-points, *k* is the number of clusters and *l* is the number of iterations. To obtain the initial clusters, Algorithm 4 requires *O(nk)*. Here, some data points remain in its cluster while the others move to other clusters depending on their relative distance from the new centroid and the old centroid. This requires *O(1)* if a data-point stays in its cluster, and *O(k)* otherwise. As the algorithm converges, the number of data points moving away from their cluster decreases with each iteration. Assuming that half the data points move from their clusters, this requires *O(nk/2)*. Hence the total cost of this phase of the algorithm is *O(nk)*, not *O(nkl)*. Thus the overall time complexity of the enhanced algorithm (Algorithm 2) becomes $O(n^2)$, since *k* is much less than *n*.

VI. EXPERIMENTAL RESULTS

The multivariate iris data set, taken from the UCI repository of machine learning databases [9], is used for testing the accuracy and efficiency of the enhanced algorithm. The same data set is given as input to the standard k-means algorithm and the enhanced algorithm. The value of *k*, the number of clusters, is taken as 3.

The results of the experiments are tabulated in Table 1. The standard k-means algorithm requires the values of the initial centroids also as input, apart from the input data values and the value of *k*. The experiment is conducted seven times for different sets of values of the initial centroids, which are selected randomly. The accuracy of clustering is determined by comparing the clusters obtained by the experiments with the three clusters already available in the UCI data set. The percentage accuracy and the time taken for each experiment are computed and the mean values are tabulated. For the enhanced algorithm, the data values and the value of k are the only inputs required since the initial centroids are computed automatically by the program. The percentage accuracy and the time taken in the case of this algorithm are also computed and tabulated.

Table 1. Performance Comparison

Algorithm	**Initial Centroids**	**Accuracy (%)**	**Time taken (ms)**
k-means algorithm (executed 7 times with randomly selected initial centroids)	5.1, 3.5, 1.4, 0.2 4.3, 3, 1.1, 0.1 6.6, 2.9, 4.6, 1.3	52.6	71
	7, 3.2, 4.7, 1.4 6.7, 3.1, 4.4, 1.4 5.1, 3.5, 1.4, 0.2	88.7	69
	7, 3.2, 4.7, 1.4 6.7, 3.1, 4.4, 1.4 7.4, 2.8, 6.1, 1.9	89.3	70
	7.4, 2.8, 6.1, 1.9 6, 3, 4.8, 1.8 6.7, 3.1, 4.4, 1.4	89.3	72
	5.1, 3.5, 1.4, 0.2 4.3, 3, 1.1, 0.1 6, 3, 4.8, 1.8	52.7	70
	6, 3, 4.8, 1.8 5.8, 2.7, 5.1, 1.9 5.1, 3.5, 1.4, 0.2	89.3	72
	5.1, 3.5, 1.4, 0.2 7, 3.2, 4.7, 1.4 6.3, 3.3, 6, 2.5	89.3	71
Mean value	-	78.7	70.7
Enhanced algorithm	computed by the program	88.6	67

Figure 1 depicts the performances of the standard k-means algorithm and the enhanced algorithm in terms of the accuracy and efficiency. It can be seen from the above experiments that the enhanced algorithm significantly outperforms the original k-means algorithm in terms of accuracy and efficiency.

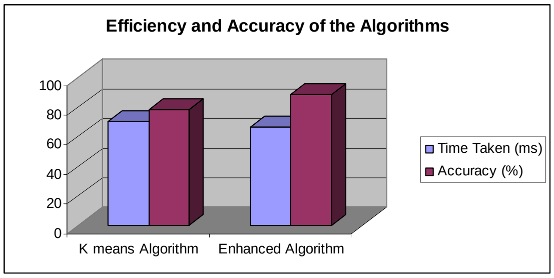

Fig. 1 Efficiency and Accuracy of the algorithms

VII. 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. This paper presents an enhanced 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 $O(n^2)$ time without sacrificing the accuracy of clusters. The previous improvements of the k-means algorithm compromise on either accuracy or efficiency.

A limitation of the proposed algorithm is that the value of *k*, the number of desired clusters, is still required to be given as an input, regardless of the distribution of the data points. Evolving some statistical methods to compute the value of *k*, depending on the data distribution, is suggested for future research. Methods for refining the computation of initial centroids is worth investigating.

REFERENCES

[1] Amir Ben-Dor, Ron Shamir and Zohar Yakini, "Clustering Gene Expression Patterns," *Journal of Computational Biology*, 6(3/4): 281-297, 1999

[2] Chaturvedi J. C. A, Green P, "K-modes clustering," *J. Classification*, (18):35–55, 2001.

[3] Daxin Jiang, Chum Tong and Aidong Zhang, "Cluster Analysis for Gene Expression Data," *IEEE Transactions on Data and Knowledge Engineering*, 16(11): 1370-1386, 2004.

[4] Fahim A.M, Salem A. M, Torkey A and Ramadan M. A, "An Efficient enhanced k-means clustering algorithm," *Journal of Zhejiang University*, 10(7):1626–1633, 2006.

[5] Huang Z, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining and Knowledge Discovery*, (2):283–304, 1998.

[6] Jiawei Han M. K, *Data Mining Concepts and Techniques*, Morgan Kaufmann Publishers, An Imprint of Elsevier, 2006.

[7] Margaret H. Dunham, *Data Mining- Introductory and Advanced Concepts*, Pearson Education, 2006.

[8] McQueen J, “Some methods for classification and analysis of multivariate observations,” *Proc. 5th Berkeley Symp. Math. Statist. Prob.*, (1):281–297, 1967.

[9] Merz C and Murphy P, UCI Repository of Machine Learning Databases, Available: ftp://ftp.ics.uci.edu/pub/machine-learning-databases

[10] Pang-Ning Tan, Michael Steinback and Vipin Kumar, *Introduction to Data Mining*, Pearson Education, 2007.

[11] Stuart P. Lloyd, “Least squares quantization in pcm,” *IEEE Transactions on Information Theory*, 28(2): 129-136.

[12] 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.