Chapter 37
Clustering Biological Data Using Enhanced k-Means Algorithm

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

Abstract With the advent of modern scientific methods for data collection, huge volumes of biological data are now getting accumulated at various data banks. The enormity of such data and the complexity of biological networks greatly increase the challenges of understanding and interpreting the underlying data. Effective and efficient Data Mining techniques are essential to unearth useful information from them. A first step towards addressing this challenge is the use of clustering techniques, which helps to recognize natural groupings and interesting patterns in the data-set under consideration. The classical k-means clustering algorithm is widely used for many practical applications. But it is computationally expensive and the accuracy of the final clusters is not guaranteed always. This paper proposes a heuristic method for improving the accuracy and efficiency of the k-means clustering algorithm. The modified algorithm is then applied for clustering biological data, the results of which are promising.

Keywords Data mining · clustering · k-means algorithm

37.1 Introduction

Advances in scientific data collection methods have resulted in the large scale accumulation of biological data at various data sources. Owing to the development of novel techniques such as DNA Microarrays for generating data [1], the rate of growth of scientific databases has become tremendous. Hence it is practically

K.A.A. Nazeer (✉) and M.P. Sebastian
Department of Computer Science and Engineering National Institute of Technology Calicut, NIT Campus (PO), Kozhikode, India-673601
e-mail: nazeer@nitc.ac.in; sebasmp@nitc.ac.in

S.-I. Ao and L. Gelman (eds.), *Electronic Engineering and Computing Technology*, Lecture Notes in Electrical Engineering 60, DOI 10.1007/978-90-481-8776-8_37,

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, as discussed in [2] is one of the major data analysis methods which is widely used for many practical applications. 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 process of clustering biological data helps to identify interesting patterns and inherent groupings in the underlying data.

The k-means algorithm proposed by [3] 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 discusses a heuristic method for improving the accuracy and efficiency of the k-means clustering algorithm.

37.2 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 used as the measure 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 change 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 change anymore. This indicates the convergence criterion for the clustering procedure. Pseudocode for the k-means clustering algorithm as given in [4] is listed as Algorithm 1.

The k-means algorithm is an extensively studied algorithm for clustering 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. Accuracy of the final clusters heavily depends on the selection of the initial centroids. The k-means algorithm is computationally expensive also. Its time complexity is $O(nkl)$ where n is the number of data points, k the number of clusters and l the number of iterations.

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 the new mean for each cluster;
Until convergence criterion is met.

37.3 Literature Survey

Several attempts were made by researchers to improve the accuracy and efficiency of the k-means algorithm, as discussed in [5]. A variant of the k-means algorithm is the k-modes method proposed by [6] 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 discussed in [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.

Yuan et al. [7] 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, this method does not suggest any improvement to the time complexity of the k-means algorithm.

Fahim et al. [8] 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.

37.4 Proposed Method

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

Algorithm 2 The Improved 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. Determine the initial centroids of the clusters by using Algorithm 3;
2. Assign the data points to the clusters by using Algorithm 4;

In the first phase, the initial centroids are determined systematically as discussed in [7] so as to produce clusters with improved accuracy. In the second phase of assigning data points to clusters, a variant of the algorithm discussed in [8] is used. It starts by forming the initial clusters based on the relative distance of each data point from the initial centroids. These clusters are subsequently refined by using a heuristic approach. The two phases of the improved method are described below as Algorithms 3 and 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 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 it 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.

Initially, determine the distances between each data point and all other data points in the set. 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, which is taken to be 0.75*(n/k). 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 taking the arithmetic mean of 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

$$d(X, Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} \tag{37.1}$$

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

$$d(X, D) = min\ (d(X, Y), where\ Y \in D) \tag{37.2}$$

The initial centroids obtained in phase 1 are given as input to the second phase, for assigning data point to the appropriate clusters. The steps involved in this phase are described as Algorithm 4.

Algorithm 4 Assigning data points to clusters

Input:
D = d1, d2,.....dn // set of n data items
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 NearestDist[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 NearestDist[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 (NearestDist) are noted. Inclusion of data points in various clusters may alter 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 by a systematic procedure.

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 signifies the convergence criterion. The heuristic method described above results in significant reduction in the number of computations and thus improves the efficiency.

37.5 Computational 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.

37.6 Experimental Results

The improved algorithm was tested with multivariate data taken from the UCI repository of machine learning databases [9]. The iris data, echocardiogram data and the breast cancer data were clustered using the original k-means algorithm and the improved algorithm.

The results of the experiments are tabulated in the Tables 37.1–37.3. For the original k-means algorithm, three experiments each were conducted for the three data sets for different values of the initial centroids. The average values of the accuracy and time taken were then computed and tabulated. For the improved algorithm, the data sets and the value of k are the only inputs required and one experiment each were conducted for the same data sets. The values of accuracy and time taken are then tabulated. Figures 37.1–37.3 illustrate the performance of the improved algorithm compared to the original k-means algorithm. It can be seen that the improved algorithm significantly outperforms the k-means algorithm in terms of accuracy and efficiency.

37.7 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.

Table 37.1 Performance comparison for iris data

Algorithm	Accuracy (%)	Time Taken (mS)
K-means	78.7	71
Enhanced K-means	88.6	69

Table 37.2 Performance comparison for echocardiogram data

Algorithm	Accuracy (%)	Time Taken (mS)
K-means	53.3	73
Enhanced K-means	70	72

Table 37.3 Performance comparison for breast cancer data

Algorithm	Accuracy (%)	Time Taken (mS)
K-means	86.6	72
Enhanced K-means	95	70

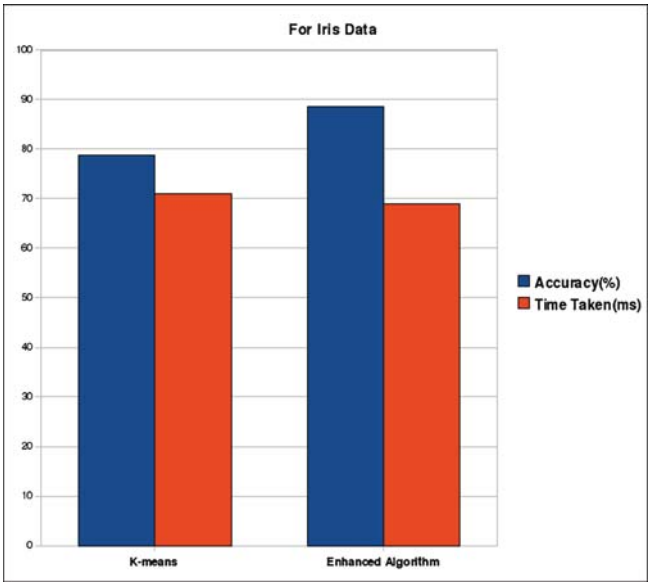

Fig. 37.1 Performance comparison for iris data

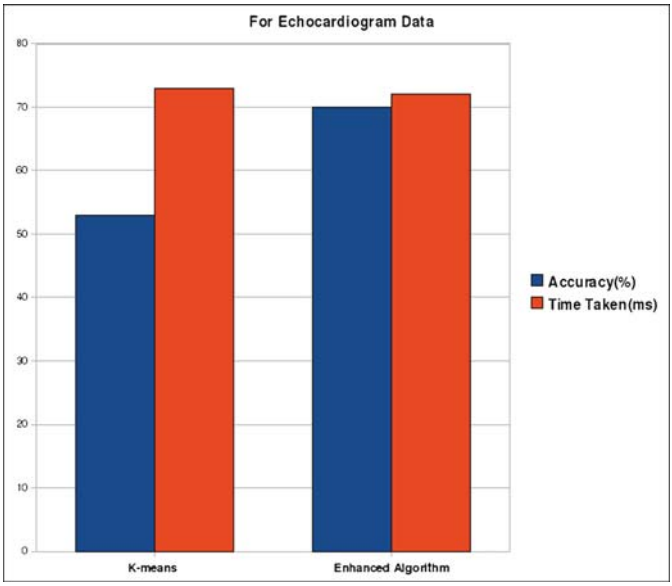

Fig. 37.2 Performance comparison for echocardiogram data

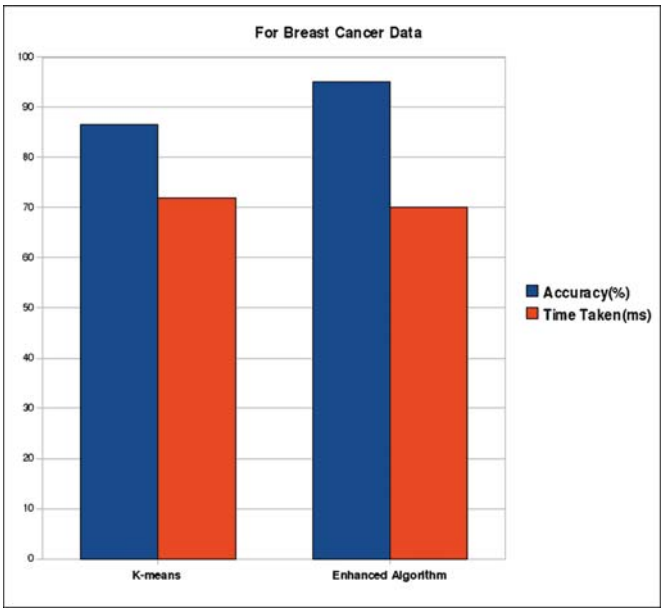

Fig. 37.3 Performance comparison for breast cancer data

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 as an input. Evolving some statistical methods to compute the value of k, based on the distribution of data, is suggested for future research. The method for finding the initial centroids may be refined further to improve the time complexity.

References

1. Daxin J., Chum T., Aidong Z.: Cluster analysis for gene expression data. IEEE Trans. Data Knowl. Eng., **16**(11), 1370–1386 (2004)
2. Han, J.: Data mining concepts and techniques. Morgan Kaufmann Publishers, An imprint of Elsevier, San Francisco, CA (2006)
3. McQueen, J.: Some methods for classification and analysis of multivariate observations. Proc. 5th Berkeley Symp. Math. Statist. Prob. (1), 281–297 (1967)
4. Dunham, M.H.: Data Mining-Introductory and Advanced Concepts. Pearson Education (2006)
5. Huang Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Min. Knowl. Disc. (2), 283–304 (1998)
6. Chaturvedi, J.C.A., Green, P.: K-modes Clustering. J. Classif. (18), 35–55 (2001)

7. Yuan, F., Meng, Z.H., Zhang, H.X., Dong, C.R.: A new algorithm to get the initial centroids. In: Proceedings of the 3rd International Conference on Machine Learning and Cybernetics, pp. 26–29, August 2004
8. Fahim, A.M., Salem, A.M., Torkey, A., Ramadan, M.A.: An efficient enhanced k-means clustering algorithm. J. Zhejiang Univ. **10**(7), 1626–1633 (2006)
9. Merz, C., Murphy, P.: UCI Repository of Machine Learning Databases. http://archive.ics.uci.edu/ml/