Microarray Gene Expression Data Clustering Using Red Black Tree Based K-Means Algorithm

[1]Jasila E K, [2]K A Abdul Nazeer
[1]Department of Computer Science and Engineering MES College of Engineering, Thrikkanapuram, Malappuram
[2]Department of Computer Science and Engineering, National Institute of Technology Calicut India

ABSTRACT

The need of high quality clustering is very important in the modern era of information processing. Clustering is one of the most important data analysis methods and the k-means clustering is commonly used for diverse applications. Despite its simplicity and ease of implementation, the k-means algorithm is computationally expensive and the quality of clusters is determined by the random choice of initial centroids. Different methods were proposed for improving the accuracy and efficiency of the k-means algorithm. In this paper, we propose a new approach that improves the accuracy of clustering microarray based gene expression data sets. In the proposed method, the initial centroids are determined by using the Red Black Tree and an improved heuristic approach is used to assign the data items to the nearest centroids. Experimental results show that the proposed algorithm performs better than other existing algorithms.

General Terms: K-means clustering, Red Black Tree, Cosine similarity, Heuristic approach.

1. INTRODUCTION

Clustering is a process of grouping the set of data items into disjoint clusters so that similarity between the items in the same cluster are high, and similarity between the items in different clusters are low [1].This paper describes an improved method for cluster analysis of microarray gene expression data [2]. Microarray mainly consists of large number of gene sequences under multiple conditions. We may need to cluster either genes or samples based on the application. In this paper, we have used the Red Black Tree based approach in the first phase to get the initial centroids. The second phase is used to assign each data item to the nearest centroid. In the second phase, the similarity between each data item and the centroids are determined by using the cosine similarity measure.

1.1 K-Means Clustering

The k-means algorithm is very popular because of its simplicity and ease of implementation. This algorithm clusters the data set into k-clusters where the number of clusters k, should be given as an input parameter [3]. The k-means algorithm consists of two phases - in the first phase k data items are randomly selected as the initial centroids and in the second phase each data item is assigned to the nearest centroid by using the Euclidean distance measure. After grouping all the data items in the data set, the centroids are recalculated by taking the mean of all data items in each cluster. This process is repeated until there is no change in the centroids [3]. The psuedocode for the k-means clustering algorithm is given below [4]:

```
Algorithm 1: Original K-means Algorithm
Input: $D= \{d1,d2,....,dn\}$, a set of  $n$ data items an
k, desired number of clusters
Output: A set of k clusters
Steps:
    1. Randomly choose k data items as initial
       centroids from the dataset
    2. Assign each data item di to the closest
       centroid based on the Euclidean distance
```

between each data item and the chosen centroids.

3. Each centroid is recalculated as the average of the data items in that cluster.
4. Steps 2 and 3 are repeated until there is no change in the cluster centroids.

This original k-means algorithm is very expensive especially for large data sets and the quality of clustering is dependent on the randomly selected initial centroids. The time required to execute the original k-means algorithm is proportional to the product of the number of data items, number of clusters and the number of iterations. The time complexity is O(nkl) where n is the number of data items, k is the number of clusters and l is the number of iterations.

2. RELATED WORK

Different approaches were proposed by researchers to improve the accuracy and efficiency of the k-means algorithm [5, 6, 7, and 8]. In the original k-means algorithm, different initial centroids produce different results. Fang Yuan et al.[5] proposed an approach for getting consistent initial centroids. This approach does not give any improvement to the time complexity of the algorithm as compared to original k-means clustering algorithm.

Fahim A M et al.[6] proposed an enhanced heuristic approach for improving the second phase of the original k-means algorithm. In this algorithm some data points remain in its cluster and the others can move to another cluster. The time complexity of this phase requires O(nk). In Fahim's approach, there is no guarantee for the accuracy of final clusters because the initial centroids are chosen randomly.

An enhanced clustering method for improving the accuracy and efficiency of the algorithm by modifying both of the phases, is proposed in [7]. In this approach, the time complexity of the first phase is O(n^2) and second phase is of O(nkl). This approach reduces the number of computations and the accuracy is improved compared to original k-means algorithm. But this enhanced algorithm is computationally expensive.

Rajeev Kumar et al.[8] proposed an enhanced k-means algorithm using Red Black Tree and Min-Heap. In this approach, the final clusters are dependent on the randomly chosen initial centroids and the space complexity of this approach is very high.

Anil K. Jain[1] highlighted different challenges for clustering. Data representation is an important factor which affects the performance of the clustering algorithm. If the data representation is good, then the clusters are very compact and simple clustering algorithms detect them. If the data representation is not good, then the choice of representation will be guided by domain knowledge. Most difficult problem in data clustering is the automatic determination of number of clusters. Clustering algorithms are executed with different values of k, and then the best value of k is selected based on the predefined criterion.

3. PROPOSED METHOD

The proposed algorithm consists of two phases-the first phase is for getting initial centroids and the second phase is to assign data points to appropriate clusters. The method involved in the algorithm is outlined as Algorithm 2.

Algorithm 2: Proposed Algorithm

Input: *D= {d1,d2,….,dn}* ,set of *n* data items and k, the desired number of clusters

Output: A set of k clusters

Steps:

1. Determine the initial centroids using Algorithm 3.
2. Assign the data points to appropriate clusters using Algorithm 4.
3. done

Algorithm 3 describes the first phase of determining the initial centroids. The Red Black Tree [9] data structure is used to arrange the data items.

In Algorithm 3, for each column, find out the minimum and maximum values and determine the range as the difference between maximum and minimum values. Based on the column with the maximum range, build the red black tree. The data points are reordered by carrying out the inorder traversal of the tree. The ordered data points are divided into k sets such that each set contains 0.89*n/k data points. This ensures that the most similar data points are included in each set. Here n is the number of data items and k is the number of clusters. The mean of each set will give the initial centroids.

Algorithm 3: Finding the Initial centroids with Red Black Tree

Input: *D={d1,d2,….,dn}*, set of *n* data points.
k, the desired number of clusters

Output: A set of k initial centroids

Steps:

```
1. For each column of the data set,
   determine the range as the difference
   between the maximum and the minimum
   data point;
2. Identify the column having the maximum
   range;
3. Depending on the column with the
   maximum range, build the Red black Tree
   for the data set.
4. Do the inorder traversal of the tree.
5. Divide the sorted data points into k sets
   such that each set contains 0.89*n/k data
   items.
6. Compute the mean of each set and take
   these means as the initial centroids.
7. done
```

Algorithm 4 is used for assigning data points to appropriate clusters. Each data point is assigned to the closest cluster by using cosine similarity [10, 11] as the distance measure. The similarity between one vector X= (xl, x2... xn) and the other vector Y= (yl, y2... yn) is defined as

$$(\sum_{i=1}^{n} X_i * Y_i)/sqrt(\sum_{i=1}^{n} X_i{}^{2})* sqrt(\sum_{i=1}^{n} Y_i \qquad (1)$$

The cosine similarity ranges between -1 and +1. The similarity value of 1 indicates that the two vectors are exactly the same, -1 means that they are totally dissimilar, and values in between -1 and +1 indicate intermediate similarity or dissimilarity.

In Algorithm 4, initially calculate the similarity between all the data points and the initial centroids. After that the data points are assigned to the clusters with the most similar centroids. The cluster number and its similarity from the centroid of that cluster are stored for each data point. Then the centroids are recalculated. The next step is an iterative stage. In this stage, for each data point compute its similarity from the centroid of the present nearest cluster. If this similarity is greater than or equal to the present highest similarity the data point stays in the cluster, else compute the similarity to all other centroids and the data point is assigned to the centroid having highest similarity. This iterative stage is repeated until the convergence criterion is met. After a few iterations, it can be found that the data points do not move from their clusters and the clusters become almost stabilized. This is considered as the convergence criterion.

Algorithm 4: Assign each data point to the appropriate cluster.

Input: *D={d1,d2,….,dn}* ,set of *n* data points.
k, the desired number of Clusters

Output: A set of k clusters

Steps:

```
1. Compute the similarity of each data-point
   $d_i$ (1 <=i <=n) to all the centroids $c_j$ (1 <= j
   <= k) as similarity ($d_i$,$c_j$).
2. For each data point $d_i$, find the closest
   centroid $c_j$ and assign $d_i$ to cluster j
3. Set ClusterId[i]=j
4. Set Highest_Simi[i] = similarity($d_i$,$c_j$).
5. For each cluster j(1 <=j <= k), recalculate
   the centroids
6. Repeat
7. For each data-point $d_i$,
     a. Compute its similarity from the centroid
        of the present nearest cluster
     b. If this similarity is greater than or equal
        to the present highest similarity, the
        data-point stays in the cluster
     Else
     a. For every centroid $c_j$ (1 <= j <= k)
        Compute the similarity ($d_i$,$c_j$).
     b. Assign the data point $d_i$ to the cluster
        with the nearest centroid $c_j$
     c. Set ClusterId [i] =j;
     d. Set Highest_Simi[i]=similarity($d_i$, $c_j$)
8. End for
9. For each cluster j (1 <= j <= k), recalculate the
   centroids
10. Until the convergence criteria is met.
11. Done
```

4. TIME COMPLEXITY ANALYSIS

In the proposed method, the time complexity for finding out the column with maximum range is O(nm) where n is the number of data items in the data set and m is the number of attributes. For inserting n data items into the red black tree and carrying out inorder traversal requires O(nlogn) and O(n) time respectively. The time complexity of dividing n data items into k sets and finding the mean of each set requires O(n). Thus the overall time complexity of first phase is O(nlogn). As the second phase makes use of a variant of the enhanced method described in [7], it takes O(nkl) time where n is the number of data points, k is the number of clusters and l is the number of iterations.

5. EXPERIMENTAL RESULTS

The different algorithms namely the K-Means, Fang Yuan et al. [5], Fahim et al.[6], Enhanced K-means[7] and the proposed algorithms are implemented with Java [12] as Front end and My SQL as Back end. The experiments are done with three different data sets. The data sets used are microarray gene expression data sets of Cancer subtypes-Bone marrow, Bladder and Lungs [13]. Results of conducting sample based clustering of the above data sets using various algorithms are tabulated in Table 1. Table 2 shows a brief description of the data sets.

The accuracy of the clusters are determined using the cluster purity[14] metric. Purity of cluster C_j is given by

$$\text{Purity }(C_j) = \frac{1}{|cj|} * \max_i(|C_j|_{class=i}) \quad (2)$$

The overall purity of clustering can be measured as a sum of individual cluster purities [14].

$$\text{Purity} = \sum_{j=1}^{k} \frac{|cj|}{n} \text{Purity}(C_j) \quad (3)$$

6. CONCLUSION

The k-means algorithm is the most popular algorithm because of its ease of implementation and simplicity. The computational overhead of the original k-means algorithm is very high and the quality of the final clusters depends on the randomly selected initial centroids. The proposed method makes use of an improved approach for determining the initial centroids and an enhanced heuristic technique for assigning data points to the clusters. Experimental results show that the proposed method produces much better accuracy of clustering compared to other similar methods.

The value of k, the number of clusters, is to be given as one of the inputs in the proposed algorithm also. Efforts can be made to determine the value of k based on the distribution of the data.

Table 1. Comparison of Accuracy using Cluster Purity

Data sets	K-Means	Fang Yuan et al	Fahim et al	Enhanced	Proposed Algorithm
Bone marrow	80.0	79.16	54.16	79.16	94.4
Bladder	65.1	75.0	67.5	75.0	80.0
Lungs	48.6	49.2	49.0	49.2	95.6

Table2. Summary of Data sets

Data set	# of samples	# of clusters
Bone marrow	72	3
Bladder	40	3
Lungs	181	2

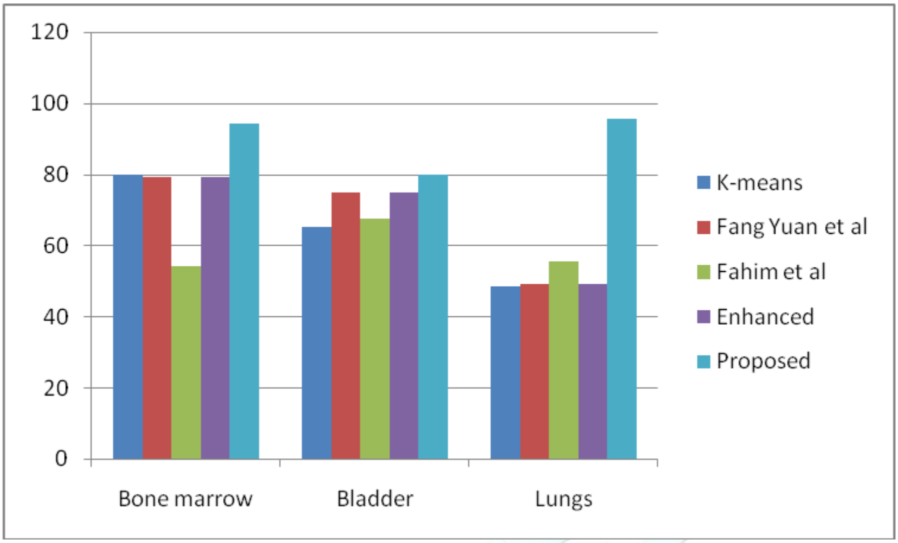

Fig.1. Comparison of Accuracy

7. REFERENCES

[1] A. Jain. Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8):651–666, 2010

[2] L. Dey and A. Mukhopadhyay. Microarray Gene Expression Data Clustering using PSO based K-means Algorithm. International Journal of Computer Science and its Applications.

[3] K-means lustering.http://databases.about.com /datamining/ kmeans.

[4] R. Elmasri and S. B. Navathe. Fundamentals of Database Systems. 4th Edition, 2005

[5] A. Fahim, A. Salem, F. Torkey, and M. Ramadan. An efficient enhanced k-means clustering algorithm. Journal of Zhejiang University- Science A, 7(10):1626–1633, 2006.

[6] Z. C.-R. D. F. Y. Zeng-Hui Meng, Hong-xia. A New Algorithm To Get The Initial Centroids. In Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, volume 1, pages 1191–1193. IEEE, 2004.

[7] K. Nazeer and M. Sebastian. Improving the Accuracy and Efficiency of the k-means Clustering Algorithm. In Proceedings of the World Congress on Engineering, volume 1. Citeseer, 2009.

[8] R. P. Rajeev Kumar and J. Dhar. Enhanced K-Means Clustering Algorithm Using Red Black Tree and Min-Heap. International Journal of Innovation, Management and Technology, 2(1):2010– 0248, 2011.

[9] T. H. C. Charles E Leiseoson Ronald, L Rivest Clifford Stein. Introduction to Algorithms. 2nd Edition, 2001

[10] CosineSimilarity.http://en.wikipedia.org/wiki /Cosine similarity/.

[11] H. X. G. X. Shiwei Zhu, Junjie Wu. Scaling up top-K cosine similarity search. Data and knowledge engineering, 70:60–83, 2011.

[12] Schildt. Java:The Complete Reference. 7th Edition, 2007.

[13] DatasetsAvailable.http://algorithmics.molgen .mpg.de /Supplements/ CompCancer/.

[14] L.P. Chandran and K. A. A. Nazeer. An Improved Clustering Algorithm based on K-Means and Harmony Search Optimization. Recent Advances in Intelligent computational Systems, IEEE, 2011.