COMPARISON OF PARTITION BASED CLUSTERING ALGORITHMS

M.D. Boomija, M.C.A., M.Phil.,
Lecturer, Department of MCA,
Prathyusha Institute of Technology and Management,
Poonamallee -Tiruvallur High Road,
Aranvoyalkuppam, Chennai – 602 025.

Abstract

Data mining refers to extracting or "mining" knowledge from large amounts of data. Clustering is one of the most important research areas in the field of data mining. Clustering means creating groups of objects based on their features in such a way that the objects belonging to the same groups are similar and those belonging in different groups are dissimilar.

In this paper, the most representative partition based clustering algorithms are described and categorized based on their basic approach. The best algorithm is found out based on their performance. Two of the clustering algorithms, namely, Centroid based k-means, Representative object based k-medoids are implemented by using JAVA and their performance is analyzed based on their clustering quality. The randomly distributed data points are taken as input to these algorithms and clusters are found out for each algorithm. The algorithm's performance is analyzed by different runs on the input data points. The experimental results are given as both graphical as well as tabular representation.

Keywords: K Means, K Medoids, clustering, Clara, Clarans.

1. Introduction

Clustering can be considered as the most important unsupervised learning problem; so, as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data. A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. We can show this with a simple graphical example:

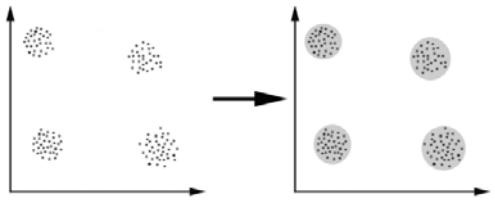


Fig.1 A Graphical Example for Clusters

In this case we easily identify the 4 clusters into which the data can be divided; the similarity criterion is distance: two or more objects belong to the same cluster if they are "close" according to a given distance (in this case geometrical distance). This is called distance-based clustering.

2. Partition-based algorithms

The aim of the partition-based algorithms is to decompose the set of objects into a set of disjoint clusters where the number of the resulting clusters is predefined by the user. The algorithm uses an iterative method, and based on a distance measure it updates the cluster of each object.

The most representative partition-based clustering algorithms are

- k-Means
- k-Medoids
- CLARA
- CLARANS

The advantage of the partition-based algorithms that they use an iterative way to create the clusters, but the drawback is that the number of clusters has to be determined in advance and only spherical shapes can be determined as clusters.

3. K-Means Clustering Algorithm

K-Means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The main idea is to define k centroids, one for each cluster. The better choice is to place the Centroids as much as possible far away from each other. This algorithm aims at minimizing an objective function, in this case a squared error function [1].

The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 ,$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centers.

The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Fig.2 k-Means algorithm

3.1 Experimental Results

Example:

Presented here in tabular and graphical form are the results of different experimental runs. Hundred random data points are input to this algorithm. The number of clusters and data points given by the user. The algorithm is repeated for thousand times to get efficient output. The cluster centers (Centroids) are calculated for each cluster by its mean values and clusters are formed depending upon the distance between data points[2].

The experimental results are shown below :

Five hundred uniformly distributed random points are taken as input as shown in Fig 3. Number of clusters chosen by user is 10. The output of one of the trial is shown in Figure 4. The result of the algorithm is given as table format in Table 2 and graphical format in Figure 5.

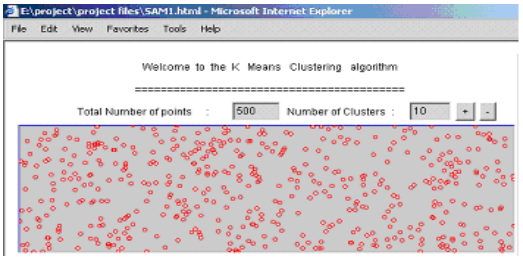


Fig.3 The random 500 data points

Number of random data points -> 500
Number of Clusters -> 10

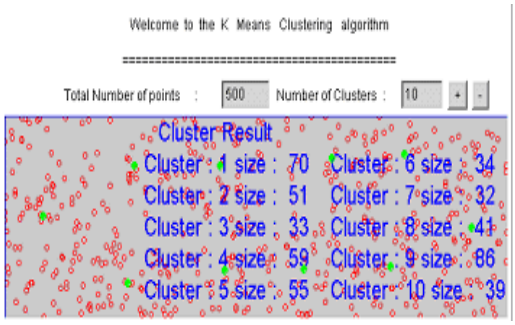


Fig.5 Data Points = 500, k = 10

Table 2 Experimental results for different runs

cluster	Total Number of Data Points				
1	41	39	45	62	v63
2	44	43	58	70	69
3	37	71	54	41	33
4	47	46	52	58	32
5	58	39	58	38	45
6	48	57	43	44	47
7	78	45	45	55	36
8	35	36	56	50	58
9	46	63	44	31	52
10	66	61	45	51	65

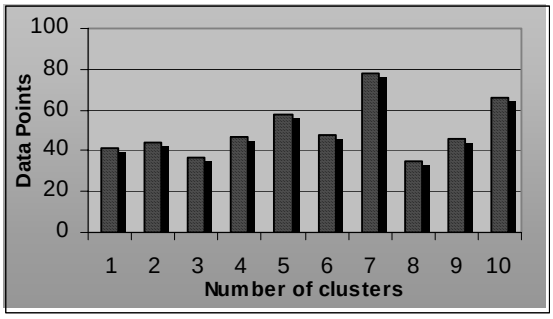


Fig. 6 Graphical Representation

4. k–Medoids Clustering Algorithm

The k-means algorithm is sensitive to outliers since an object with an extremely large value may substantially distort the distribution of data. Instead of taking the mean value of the objects in a cluster as a reference point, the medoid can be used, which is the most centrally located object in a cluster. K-Medoids method uses representative objects as reference points instead of taking the mean value of the objects in each cluster. [3]

Algorithm: k-Medoids

Input: The number of clusters k and a database containing n objects

Output: A set of k clusters that minimizes the sum of the dissimilarities of all the objects to their nearest medoid.

Method:

Arbitrarily choose k objects as the initial medoids;

- Repeat
- Assign each remaining object to the cluster with the nearest medoid
- Randomly select a non medoid object,o_{random}
- Compute the total cost,S of swapping o_j with o_{random}
- If $S < 0$ then swap o_j with o_{random} to form the new set of k medoid
- Until no change

4.1 Experimental Results

Example 1:

The results are presented here in tabular and graphical form for many experimental runs.
Number of clusters chosen by user is 5.
The output is shown in figure 7.

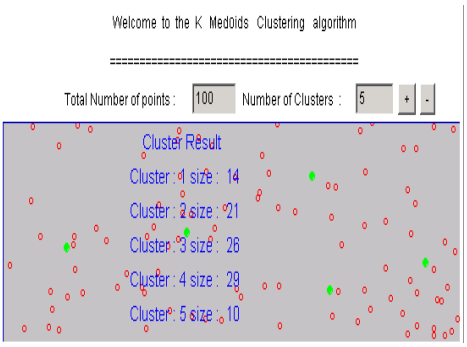


Fig.7 The output for 100 data set

The result of the algorithm is given as table format in Table 3. and graphical format in Figure 8.

Number of random data points -> 500

Number of Clusters -> 10

cluster	Total Number of Data Points				
	Run1	Run2	Run3	Run4	Run5
1	54	46	58	53	62
2	48	55	66	37	68
3	48	83	30	66	38
4	45	44	49	56	37
5	40	37	46	57	46
6	65	37	43	48	48
7	42	60	45	36	36
8	51	49	51	44	53
9	60	42	49	50	52
10	47	47	63	53	60

Table 3 Experimental results for different runs

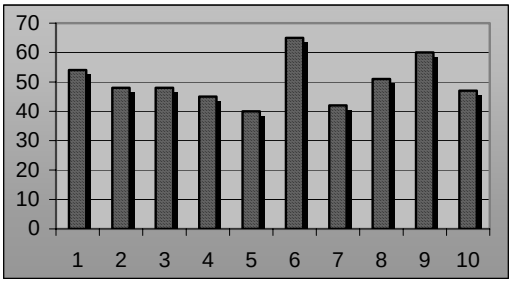
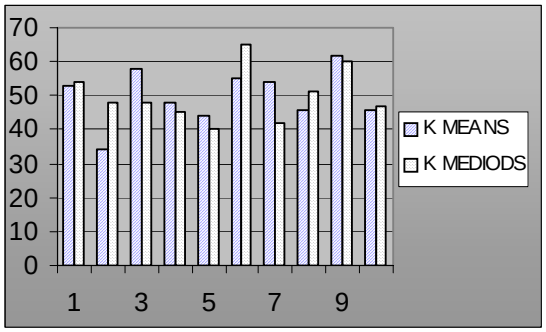


Fig.8 Graphical Representation k =10

5 Comparison

k-means and k-medoids

The k-medoids method is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean. [4] However, its processing is more costly than the k-mean method. The comparison is given as graphical representations in Figure 9.



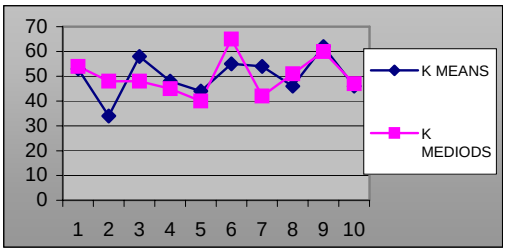


Figure 9 Comparison

6. Partition Methods In Large Databases

From k-medoids to CLARANS

A typical k-medoids partitioning algorithm works effectively for small data sets, but does not scale well for large data sets. To deal with larger data sets, a sampling-based method, called Clara (clustering large applications) can be used.

The idea behind CLARA is as follows:

Instead of taking the whole set of data into consideration, a small portion of the actual data is chosen as a representative of the data. Medoids are then chosen from this sample Partitioning Around Medoids. If the sample is selected in a fairly random manner, it should closely represent the original data set. The representative objects (medoids) chosen will likely be similar to those that would have been chosen from the whole data set. Clara draws multiple samples of the data set, applies PAM on each sample, and returns its best clustering as the output. [5]

The effectiveness of CLARA depends one the sample size. Notice that PAM searches for the best k medoids among a given data set, whereas CLARA searches for the best k medoids among the selected sample for the data set. CLARA cannot find the best clustering if any sampled medoid is not among the best k medoids. A k-medoids type algorithm called CLARANS (Clustering Large Applications based upon RANdomized Search) was proposed that combines both sampling technique with PAM. However, unlike CLARA, CLARANS does not confine itself to any sample at any given time. While CLARA has a fixed sample with some randomness in each step of the search, CLARANS draws a sample with some randomness in each step of the search. The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of k medoids. The clustering obtained after replacing a single medoid is called the neighbor of the current clustering. If a better neighbor is found, CLARANS moves to the neighbor's node and the process starts again; otherwise the current clustering produces a local optimum. [6]

7. Conclusion

The choice of clustering algorithm depends both on the type of data available and on the particular purpose and application. The partition based algorithms work well for finding spherical-shaped clusters in small to medium-sized databases.

The k-medoids method is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean. But its processing is more costly than the k-means method. The k-medoids method works effectively for small data sets, but does not scale well for large data sets. To deal with larger data sets, a sampling-based method, called CLARA can be used. The effectiveness of CLARA depends on the sample size. CLARA cannot find the best clustering if any sampled medoid is not among the best k medoids. CLARANS is the most effective portioning method among all. It enables the detection of outliers. For the future enhancement, these algorithms are combined together to form the hybrid algorithm which is more efficient to form the clusters than all other algorithms.

References

[1] Jiawei Han & Micheline Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann Publishers, New Delhi, 2001

[2] Zhao, Tong, Nehorai, Arye, and Porat, Boaz" K-Means Clustering-Based Data Detection and Symbol-Timing Recovery for Burst-Mode Optical Receiver" IEEE transactions on Communications Vol. 54. No 8. Aug 2006. 1492-1501.

[3] Zhong Wei, et al. "Improved K-Means Clustering Algorithm for Exploring Local Protein Sequence Motifs Representing Common Structural Property" IEEE Transactions on Nanobioscience, Vol.4., No.3. Sep. 2005. 255-265.

[4] Coomans, I. Broeckaert, M. Jonckheer, D.L. Massart "Comparison of Multivariate Discrimination Techniques for Clinical Data. Application to the Thyroid Functional State. Methods of Information in Medicine" Vol.22, (1983) 93- 101

[5] D.L. Davies and D.W.Bouldin, A cluster separation measure, IEEE Trans. Pattern Anal.MachineIntell.Vol.1, 1979,pp.224-227.

[6] P.Dempster, N.M. Laird, and D.B. Rubin "Maximum Likelihood from Incomplete Data via the EM algorithm", Journal of the Royal Statistical Society, Series B, vol. 39,1977,1:1-38.

Biography:

Boomija M D, is presently serving as a Lecturer, Department of Computer Applications, Prathyusha Institute of Technology and Management, Aranvoyalkuppam, Chennai. She has received her M.C.A. from Madurai Kamaraj University on 2001, Madurai. She has obtained M.Phil.(CS) from Alagappa University on 2008. She has six years teaching experience. Her area of interest includes Object Oriented Programming, Middleware Technologies, Data Mining.