Comparison of A Time Efficient Modified K-mean Algorithm with K-Mean and K-Medoid algorithm

Saurabh Shah
R.K. University(Rajkot) &
Babaria Institute of Technology,
Dept of Computer Engineering
Vadodara, India
saurabh_er@rediffmail.com

Manmohan Singh
Babaria Institute of Technology,
Dept of Computer Engineering
Vadodara, India
Manmohan_sati@yahoo.co.in

***Abstract*— Clustering analysis is a descriptive task that seeks to identify homogeneous groups of objects based on the values of their attributes. This paper proposes a new algorithm for Modified K-Means clustering which executes like the K-means algorithm and k-medoids algorithms and tests several methods for selecting initial cluster. Modified K-Mean Algorithm is better in terms of number of clusters and execution time comparisons with K-Mean and K-Mediod. Proposed algorithm is evaluated using real data and results are compared with k-Means and k-medoids where it takes reduced time in computation and better performance compared to K-Means and K-Medoids algorithms.**

Keywords- clustering, k-means, k-medoids

I. INTRODUCTION

Clustering is a way of grouping together data samples that are ***similar*** in some way - according to some criteria that you pick. It is a form of ***unsupervised learning***; generally don't have examples demonstrating how the data *should* be grouped together. So, it's a method of ***data exploration,*** a way of looking for patterns or structure in the data that are of interest.
Clustering algorithms-

a) Hierarchical agglomerative clustering

b) K-means clustering and quality measures

c) Self-Organizing Maps (if time)

A. K-means:

K-means (MacQueen, 1967) is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done.. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

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

where $\| x_i^{(j)} - c_j \|^2$ 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.

   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

B. K-Medoids algorithm:

K-medoids is a clustering algorithm related to the *k*-means algorithm and the medoidshift algorithm. Both the *k*-

978-0-7695-4692-6/12 $26.00 © 2012 IEEE
DOI 10.1109/CSNT.2012.100

means and *k*-medoids algorithms are partitional (breaking the dataset up into groups) and both attempt to minimize squared error, the distance between points labeled to be in a cluster and a point designated as the center of that cluster. In contrast to the *k*-means algorithm, *k*-medoids chooses datapoints as centers (medoids or exemplars). *K*-medoids is also a partitioning technique of clustering that clusters the data set of *n* objects into *k* clusters with *k* known a priori. A useful tool for determining *k* is the silhouette. It could be more robust to noise and outliers as compared to *k*-means because it minimizes a sum of general pair wise dissimilarities instead of a sum of squared Euclidean distances. The possible choice of the dissimilarity function is very rich but in our applet we used the squared Euclidean distance. A medoid of a finite dataset is a data point from this set, whose average dissimilarity to all the data points is minimal i.e. it is the most centrally located point in the set.

The most common realization of *k*-medoid clustering is the Partitioning around Medoids (PAM) algorithm and is as follows:

1. Initialize: randomly select *k* of the *n* data points as the medoids
2. Assignment step: Associate each data point to the closest medoid.
3. Update step: For each medoid *m* and each data point *o* associated to *m* swap *m* and *o* and compute the total cost of the configuration (that is, the average dissimilarity of *o* to all the data points associated to *m*). Select the medoid *o* with the lowest cost of the configuration.
4. Repeat alternating steps 2 and 3 until there is no change in the assignments.

C. *Modified K-mean algorithm:*

Efficient K-means Algorithm (Zhang et al., 2003) is an improved modified of k-means which can avoid getting into locally optimal solution in some degree, and reduce the probability of dividing one big cluster into two or more ones owing to the adoption of cluster -error criterion. Algorithm: Modified K-means(S, k), S={x1,x2,…,xn } Input: The number of clusters k1(k1> k) and a dataset containing n objects(Xi+) Output: A set of k clusters (Cij) that minimize the Cluster -error criterion

1. Draw multiple sub-samples {SI, S2, . . . , Sj } from the original dataset;
2. Repeat step 3 for m=l to i
3. Apply K-means algorithm for sub sample Sm for k1 clusters.
4. Compute
5. Choose minimum of minimum distance from cluster center criteria as the refined initial points Zij ,i+ j ∈[1,k1]
6. Now apply k-means algorithm again on dataset S for k1 clusters.
7. Combine two nearest clusters into one cluster and recalculate the new cluster center for the combined cluster until the number of clusters reduces into k.

II. PERFORMANCE COMPARISON OF SEVERAL METHODS FOR SELECTING INITIAL MEDOID:

Performance of iterative clustering algorithms depends highly on the choice of cluster centers in each step. The experiments are pursued on both synthetic and real data sets. Experimental results are reported on two synthetic data sets. In this data set, the average transaction size and average maximal potentially frequent item set size are set to 4 and 5, respectively, while the number of transactions in the dataset is set to 100 K. It is a sparse dataset. The frequent item sets are short and not numerous. For second synthetic data set used, the average transaction size and average maximal potentially frequent item set size are set to 20 and 25, respectively. There exists exponentially numerous frequent item sets in this data set when the support threshold goes down. There are also pretty long frequent item sets as well as a large number of short frequent item sets in it. It contains abundant mixtures of short and long frequent item sets.

III. COMPARISON BETWEEN K-MEAN AND K-MEDIOD ALGORITHM WITH NUMBER OF CLUSTER AND EXECUTION TIME

Table 1. Number of clusters and execution time for K-Mean Algorithms and K-Mediod Algorithms

No. of Clusters	Time Taken To Execute (In milliseconds) K-Mean Algorithms	Time Taken To Execute (In milliseconds) K-Mediod Algorithms
2	21345	20328
3	42175	40385
4	68512	69214
5	74365	73343

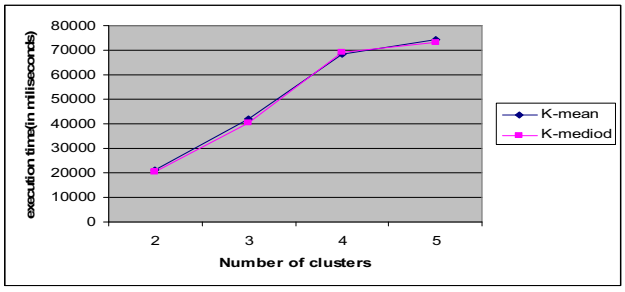

Fig. 1. Comparison Graph

Comments: Above figure shows comparison between K-mean and K-mediod algorithms. As graph show that when number of clusters is less, K-mediod takes less time to execute than the K-mean. If the number of clusters is more than it is again true that K-mediod takes less time to execute than the K-mean. At

the most number of cluster is increased than the execution time taken by K-mediod is less then the K- mean.

Table 2. Number of clusters and execution time for K-Mean, K-Mediod and Modified K-mean Algorithms

No. of Clusters	Time Taken To Execute (In milliseconds) K-Mean Algorithms	Time Taken To Execute (In milliseconds) K-Mediod Algorithms	Time Taken To Execute (In milliseconds) Modified K-Mean
2	**22365**	**20622**	**17321**
3	**42301**	**42281**	**25915**
4	**67812**	**64536**	**47572**
5	**75343**	**72391**	**48343**

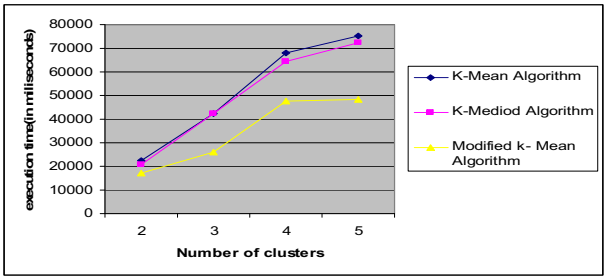

Fig. 2. Comparison Graph

Comments: Above figure shows comparison between K-mean, K-mediods and Modified K-mean algorithms. As graph show that when number of clusters is less, Modified K-mean takes less time to execute than the K- mediods and K- mediods takes less time to execute than the K- mean. If the number of clusters is more than it is again true that Modified K-mean takes less time to execute than the K- mediods and K-mean. At the particular number of the records the execution time taken by Modified K-mean takes approximately equal rather than K-mediods and K-mean. K- mediods

IV. CONCLUSION AND FUTURE SCOPE

In this paper, a new algorithm for modified K-Means clustering is proposed which runs like the K-means clustering. The result from number of cluster shows that the proposed method has better performance than K-means and k-mediod clustering and it takes less computation time than K-mean and k-mediod. Also various methods for selecting initial cluster are presented and compared. This modified K-Means is a better in terms of clustering performance; its computation time is less. So, even the method of selecting initial cluster described in the proposed method is good enough to use when considering both the performance and the execution time.

REFERENCES

[1] S. A. Raut, S. R. Sathe, and A. Raut, "Bioinformatics: Trends in GeneExpression Analysis," *proceedings of 2010 International ConferenceOn Bioinformatics and Biomedical Technology*, 16-18 April 2010,Chengdu, China.

[2] S. A. Raut, S. R. Sathe, and A. P. Raut, "Gene Expression Analysis-AReview for large datasets," *Journal of Computer Science andEngineering*, vol.4, Issue 1, November 2010.

[3] Xiong, H., J. Wu and J. Chen, 2009. K-Meansclustering versus validation measures: A datadistribution perspective. IEEE Trans. Syst., Man,Cybernet.PartB,39:318-331.http://www.ncbi.nlm.nih.gov/pubmed/19095536.R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[4] Berkhin, P., 2002. Survey of clustering data miningtechniques. Technical Report, Accrue Software,Inc.http://www.ee.ucr.edu/~barth/EE242/clustering_survey.pdf

[5] MacQueen, J.B. (1967). Some Methods for Classification and Analysis of Multivariate Observations. In Proc. of 5th Berkley Symposium on Mathematical Statistics and Probability, Volume I: Statistics, pp. 281–297.

[6] S. Ray, and R. H. Turi, "Determination of number of clusters in k-meansclustering and application in colour image segmentation," InProceedings of the 4th International Conference on Advances in PatternRecognition and Digital Techniques, 1999, pp.137-143.

[7] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "Wave-Cluster: AMulti-Resolution Clustering Approach for Very Large SpatialDatabases," Proc. 24th Int. Conf. on Very Large Data Bases. New York,1998, pp. 428-439.

[8] R. Sibson, "SLINK: an optimally efficient algorithm for the single-linkcluster method," The Comp. Journal, 16(1), 1973, pp. 30-34.

[9] T. Zhang, R. Ramakrishnan, and M. Linvy, "BIRCH: An Efficient DataClustering Method for Very Large Databases," Proc. ACM SIGMODInt. Conf. on

[10] Zhang Y. , Mao J. and Xiong Z.: An efficient Clustering algorithm, In Proceedings of Second International Conference on Machine Learning and Cyber netics, November 2003.