Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 78 (2016) 507 – 512

Procedia Computer Science

International Conference on Information Security & Privacy (ICISP2015), 11-12 December 2015, Nagpur, INDIA

Analysis of K-Means and K-Medoids Algorithm For Big Data

Preeti Arora[1], Dr. Deepali[2] , Shipra Varshney[3]

[a]*Bhagwan Parshuram Institute of Technology, New Delhi and 110085, India,*
[b] *Bhagwan Parshuram Institute of Technology, New Delhi and 110085, India,*
[c] *Northern India Engineering College, New Delhi and 110085, India*

Abstract

Clustering plays a very vital role in exploring data, creating predictions and to overcome the anomalies in the data. Clusters that contain collateral, identical characteristics in a dataset are grouped using reiterative techniques. As the data in real world is growing day by day so very large datasets with little or no background knowledge can be identified into interesting patterns with clustering. So, in this paper the two most popular clustering algorithms K-Means and K-Medoids are evaluated on dataset transaction10k of KEEL. The input to these algorithms are randomly distributed data points and based on their similarity clusters has been generated. The comparison results show that time taken in cluster head selection and space complexity of overlapping of cluster is much better in K-Medoids than K-Means. Also K-Medoids is better in terms of execution time, non sensitive to outliers and reduces noise as compared to K-Means as it minimizes the sum of dissimilarities of data objects.

Keywords: Clustering; K-Means,;K-Medoids

1 INTRODUCTION

To identify useful, valid, naive and comprehensible patterns in the data is known as Data Mining. Existing data and by simply analyzing the data the process of data mining works. The most primarily accepted definition of data mining is to turn raw data into useful data or information [2].

*+91-9968351818, +91-9871947800, +91-9811666019
erpreetiarora07@gmail.com,deepalivermani@gmail.com,shipra_vin@yahoo.com

Peer-review under responsibility of organizing committee of the ICISP2015
doi:10.1016/j.procs.2016.02.095

The data mining services receives raw data, Meta data and possibly domain specific knowledge from the client. These days the market is full of Big Data due to enormous growth of data day by day. But after selection, extraction and transformation, usually it's not big" anymore. This is where the big data and data mining are related somehow. Big data isn't a newer term these days. It is a marketing term and not a technical term. On the contrary if the data mining is a process of looking into big data sets for related and significant information and the techniques of mining data without advance knowledge of the group definitions are useful then this process is a good example looking pertinent information from huge data sets. The interpretation in real businesses is to collect huge and large sets of compatible or significant data. Now the middle person or decision makers need to process smaller and specific pieces of data from those huge and massive large sets. Data mining now comes into the picture to unveil the pieces of information. Several techniques for clustering are as follows: [2]

- Partitioning Method
- Hierarchical Method
- Grid- based Method
- Density-based Method

Among all these methods, this paper is aimed to explore partitioning based clustering methods which are K-Means and K-Medoids. These methods are discussed along with their algorithms, strength and limitations.

1.1. K-Means

The K-Means algorithm is a well-known partitioning method for clustering. K-Means clustering method, groups the data based on their closeness to each other according to the Euclidean distance. It takes k_y as input parameter and partition a set of n object from k_y clusters. The mean value of the object is taken as similarity parameter to form clusters. The cluster mean or center is formed by the random selection of k_Y object. By comparing most similarity other objects are assigning to the cluster. For each data vector this algorithm calculates the distance between data vector and each cluster centroid using equation (1).

$$D(Zp,Mj)= \sqrt{(\sum(Zp , Ky-Mj, Ky) \quad .....(1)$$

Zp is p^{th} data point M_J is centroid of j^{th} cluster.

The centroid is recalculated each time respectively after addition of data point in cluster j. It is calculated using equation (2)

$$Mj=1/Nj\sum Zp, \ \nabla Zp \in Cj \quad .....(2)$$

Where Nj is the number of data point in cluster j.

Input: K_y: the number of clusters **D_y**: a data set containing n object

Output: A set of K_y clusters

Algorithm:

1. Input the data set and value of K_y**.**
2. If f K_y= = 1 then Exit.
3. Else
4. Choose k objects from D randomly as the initial cluster centres.
5. For every data point in the cluster j reissue and define every object into the cluster where the object matches, based on the object’s mean value in the cluster.

6. Update cluster means; after that for each cluster calculate the object's mean value.
7. Repeat from step 4 until no data point was assigned otherwise stop.

The satisfying criteria can be either number of iteration or change of position of centroid in consecutive iterations

1.2. K-Medoids

The K-Medoids algorithm is used to find Medoids in a cluster which is centre located point of a cluster. K-Medoids is more robust as compared to K-Means as in K-Medoids we find k as representative object to minimize the sum of dissimilarities of data objects whereas, K-Means used sum of squared Euclidean distances for data objects. And this distance metric reduces noise and outliers.

Drawbacks of K-Means [1] algorithm:

1) To find K-Value is difficult task.

2) It is not effective when used with global cluster.

3) If different initial partitions has been selected than it may vary the result for clusters.

4) Different size and different density cluster is not handled by the algorithm.

We used K-Medoids algorithm that is based on object representative techniques [4] to reduce the drawbacks of K-Means algorithm. Medoids is the data object of cluster which is most centrally located. Medoidss are selected randomly from the K_y data objects to form K_y cluster and other remaining data objects are placed near to Medoids in a cluster. Than process all data objects of cluster to find new Medoids in repeated fashion to represent new cluster in better way. After finding the new Medoids bind all the data objects to the cluster. Location of Medoids change accordingly with each iteration. So k_y clusters are formed representing n data objects [3].

Input: K_y: the number of clusters, D_y: a data set containing n objects.

Output: A set of k_y clusters.

Algorithm:

- Randomly select k_y as the Medoids for n data points.
- Find the closest Medoids by calculating the distance between data points n and Medoids k and map data objects to that.
- For each Medoids m and each data point **o** associated to m do the following:

  Swap **m** and **o** to compute the total cost of the configuration than

  Select the Medoids **o** with the lowest cost of the configuration.
- If there is no change in the assignments repeat steps 2 and 3 alternatively.

2. Implementation

In this paper, the K-Means and K-Medoids algorithm are implemented on dataset transaction10k of KEEL [8] from **http://sci2s.ugr.es/keel/category.php?cat=uns**. The implementation of algorithms is carried out in MATLAB programming Language.

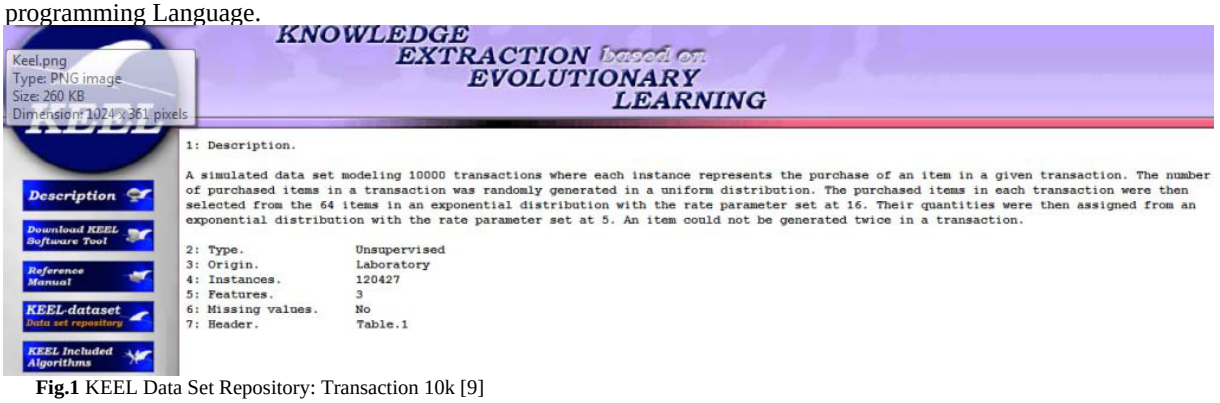

Fig.1 KEEL Data Set Repository: Transaction 10k [9]

The execution time of algorithm is in ms and it may provide different results on different computer. In the transcation10k dataset 10000 transactions represents the purchase of an item for a transaction which was randomly generated with the following data description:

Table 1. Data description of different set with different range of values.

S. No.	Attributes	Types	Range of Values
1	Transaction Id	Int	[0-9999]
2	Item Id	Int	[111-444]
3	Quantity	Int	[1-11]

3. Results

The resulting clusters of the K-Means algorithm is presented in Fig. 2 showing the overlapping of clusters whereas results of K-Medoids are shown in Fig. 3 showing less overlapping as compared to K-Means. This overlapping is reduced due to pair wise distance measure in the K-Medoids algorithm and the K-Means calculates it with sum of squared Euclidean distance metric.

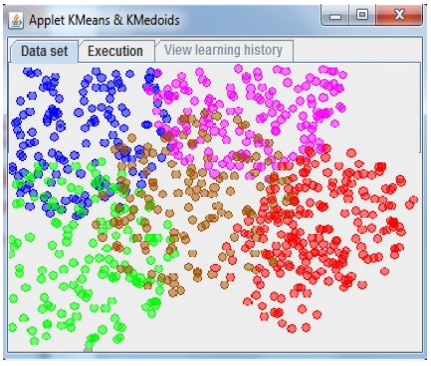

Fig.2 Cluster Overlapping in K-Means Algorithm

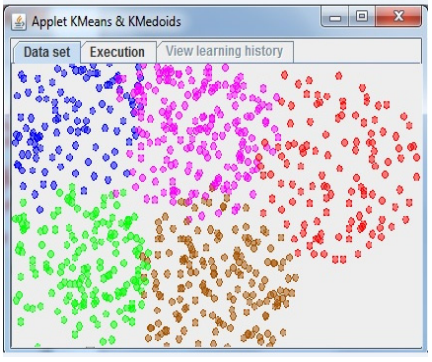

Fig.3 Clusters in K-Medoids Algorithm

The result of Fig. 4 and Fig. 5 shows the cluster head center for the K-Means and K-Medoids respectively. By the result it is clear that the iterative process of K-Medoids replacing representative objects by non representative objects improve the quality of the resulting clusters.

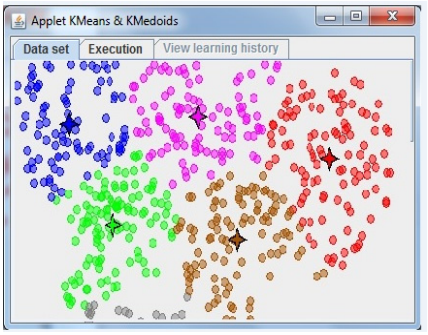

Fig.4 Cluster head of K-Means

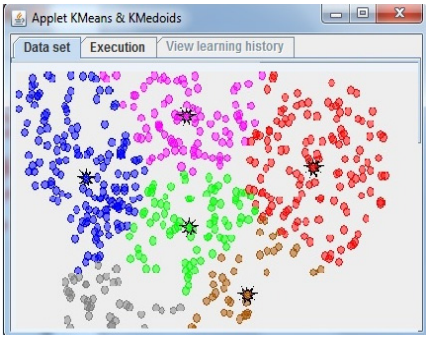

Fig.5 Cluster head of K-Medoids

K-Medoids is more robust as compared to K-Means. As in K-Medoids we find k as representative object to minimize the sum of dissimilarities of data objects whereas, K-Means used sum of squared Euclidean distances for data objects as shown in the Fig.6 & Fig.7. And this distance metric reduces noise and outliers.

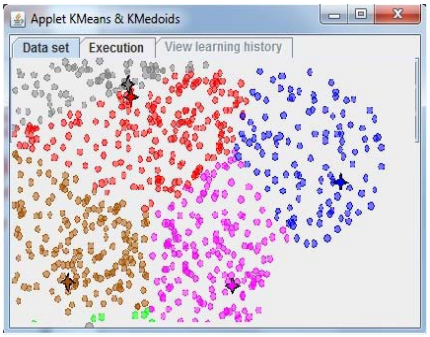

Fig.6 Outliers in K-Means

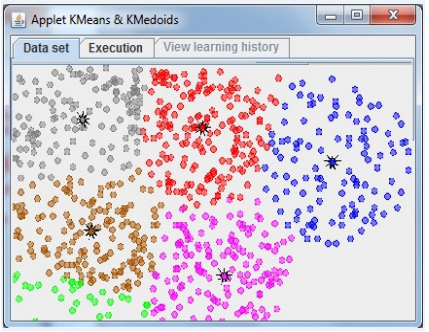

Fig.7 Outliers in K-Medoids

Also the comparison of K-Means & K-Medoids in the form chart for space complexity when the cluster are overlapping and time taken in cluster head selection is shown in Fig.8 which shows K-Medoids is better choice than K-Means.

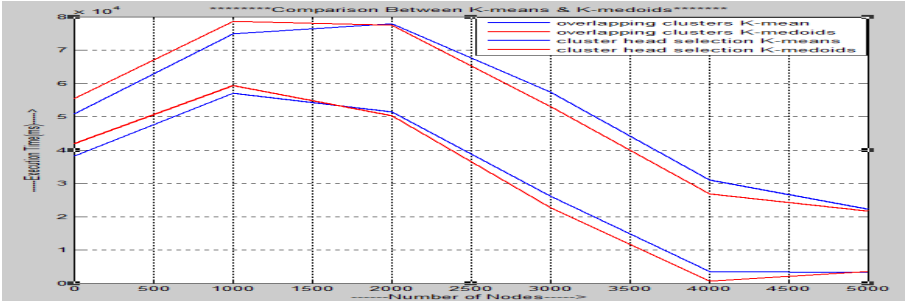

Fig.8 Comparison Chart of K-Means and K-Medoids

In this paper we conclude the results of both K- mean and K-Medoids clustering algorithms with respect to the number of clusters formed and distance metric. The comparison results show that time taken in cluster head selection and space complexity of overlapping of cluster is much better in K-Medoids than K-Means. Also the result of dataset shows that K-Medoids is better in all aspects such as execution time, non sensitive to outliers and reduction of noise but with the drawback that the complexity is high as compared to K-Means.

4. CONCLUSION

In this paper we conclude the results of both K- mean and K-Medoids clustering algorithms with respect to the number of clusters formed and distance metric. The comparison results show that time taken in cluster head selection and space complexity of overlapping of cluster is much better in K-Medoids than K-Means. Also the result of dataset shows that K-Medoids is better in all aspects such as execution time, non sensitive to outliers and reduction of noise but with the drawback that the complexity is high as compared to K-Means.

5. FUTURE WORK

We have tried to obtain accurate results of clustering by using two popular clustering algorithms with the number of clusters formed and distance metric. This metric can be extended using three more distance metrics namely eucledian, manhattan and correlation in our future work. From the experimental results it can be concluded that on changing the value of the distance metric, the results of the clustering algorithm changes. In our future work, we will consider another different distance measures for K-Means algorithm with respect to a big dataset and perform a comparison among them, thereby, try to propose a good one for the task of clustering big data set. Also, we will try to extend our study for another partition clustering algorithms like K-Medoids, CLARA and CLARANS.

References

1. Saurabh Shah & Manmohan Singh “Comparison of A Time Efficient Modified K-Mean Algorithm with K-Mean and K-Medoids algorithm”, International Conference on Communication Systems and Network Technologies, 2012
2. T. Velmurugan,and T. Santhanam, “A Survey of Partition based Clustering Algorithms in Data Mining: An Experimental Approach” An experimental approach Information. Technology. Journal, Vol, 10,No .3 , pp478-484,2011.
3. Shalini S Singh & N C Chauhan ,“K-Means v/s KMedoidss: A Comparative Study”, National Conference on Recent Trends in Engineering & Technology, 2011.
4. “Data Mining Concept and Techniques" ,2nd Edition, Jiawei Han, By Han Kamber.
5. Jiawei Han and Micheline Kamber, “Data MiningTechniques”, Morgan Kaufmann Publishers, 2000.
6. Abhishek Patel,”New Approach for K-Mean and KMedoidss algorithm”, International Journal of Computer Applications Technology and Research,2013.
7. http://www.cs.put.poznan.pl/jstefanowski/sed/DM- 7clusteringnew.pdf
8. http://sci2s.ugr.es/keel/dataset/data/unsupervised/transactions10k-names.txt
9. http://sci2s.ugr.es/keel/category.php?cat=uns#sub1