INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume3, Issue2)

A Survey on Partition Based Parallel Data Mining Algorithms for Clustering

Monika Aal
M.Tech Student, Computer Engineering
C.U. Shah University
Gujarat, India
aalmonika94@gmail.com

Abstract: Volumes of data are exploding in both scientific and commercial domains. Data mining techniques that extract information from the huge amount of data have become popular in many applications. Algorithms are designed to analyze those volumes of data automatically inefficient ways so that users can grasp the intrinsic knowledge latent in the data. Clustering is important in data analysis and data mining applications. Clustering is a division of data into a group of similar objects. Each group called a cluster consists of objects that are similar between themselves and dissimilar between comparing to objects of other groups. This paper is aimed to study of all the parallel data mining algorithms based on partition.

Keywords: Clustering, Partition Algorithm, K-Means, K-Medoid.

I. INTRODUCTION

Clustering techniques have a wide use and importance nowadays. This importance tends to increase the amount of data grows and the processing power of the computers increases. Clustering applications are used extensively in various fields such as artificial intelligence, pattern recognition, economics, ecology, psychiatry, and marketing. Data clustering is under vigorous development. Contributing areas of research include data mining, statistics, machine learning, spatial database technology, biology, and marketing. Owing to the huge amounts of data collected in databases, cluster analysis has recently become a highly active topic in data mining research [1]. A good clustering method will produce better quality clusters with

High intraclass relationship
Low inter-class relationship

The value of clustering result based on both the similarity measure used by the method and its implementation. The value of clustering method is also measured by its capability [2]. Data clustering algorithms can be divided into following categories [2,3].

PARTITIONING ALGORITHMS: Build various partitions and then evaluate them by some measure.
HIERARCHY ALGORITHMS: Create a hierarchical breakdown of the set of data (or objects) using some criterion.
DENSITY BASED: Built on connectivity and density functions.
GRID BASED: Grid based clustering is depending on a multiple-level granularity structure.
MODEL BASED: A model is offered for each of the clusters and the idea is to find the best fit of that model to each other.

II. PARTITIONING ALGORITHMS

The most well-known and commonly used partitioning methods are *k-means*, *k-medoids*, and their variations. Partitional clustering techniques create a one-level partitioning of the data points. There are a number of such techniques, but we shall only describe two approaches in this section: K-means and K-medoid. Both these techniques are based on the idea that a center point can represent a cluster. For K-means we use the notion of a centroid, which is the mean or median point of a group of points. Note that a centroid almost never corresponds to an actual data point. For K-medoid we use the notion of a medoid, which is the most representative (central) point of a group of points. Partitional techniques create one level (un-nested) partitioning of the data points. If *K* is the desired number of clusters, then partitional approaches typically find all *K* clusters at once.

Clustering

***K-means*:** Each cluster is represented by the center of the cluster.
K-medoids: Each cluster is represented by one of the the objects in the cluster [4].

III. K-MEANS CLUSTERING ALGORITHM

The K-Means algorithm is used for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input: k: the number of clusters, D: a data set containing n objects

Output: A set of k clusters.

Method

1. Arbitrarily choose k objects from D as the initial cluster centers;
2. Repeat
3. Based on mean value of the elements in the cluster, (re)assign each element to the cluster to which the element is the most similar;
4. Update the cluster means, that is, calculate the mean value of the object for each cluster;
5. Until no change;

The distance between two elements is calculated using the Euclidian distance measure.

The *k*-means algorithm takes the input parameter, *k*, and partitions a set of *n* objects into *k* clusters so that the resulting intracluster similarity is high but the intercluster similarity is low.

$$E = \sum_{i=1}^{k} \sum_{p \epsilon Ci} (p - m_i)^2$$

Where E is the sum of the square error for all objects in the data set; p is the point in space representing a given object, and mi is the mean of cluster Ci (both p and mi are multidimensional). In other words, for each object in each cluster, the distance from the object to its cluster center is squared, and the distances are summed. This criterion tries to make the resulting k clusters as compact and as separate as possible [1, 5].

Limitation of K-Means algorithm

1) To find K-Value is a difficult task.
2) It is not effective when used with the global cluster.
3) If different initial partitions have been selected then it may vary the result for clusters.
4) Different size and different density cluster are not handled by the algorithm.

IV. K-MEDOID ALGORITHM

The k-means method uses centroid to represent the cluster and it is sensitive to outliers. This means a data object with an extremely large value may disrupt the distribution of data. K-Medoids method overcomes this problem by using Medoids to represent the cluster rather than centroid. A Medoids is the centrally positioned data object in a cluster. Here, k data objects are selected randomly as Medoid to represent k cluster and remaining all data objects are placed in a cluster having Medoids nearest (or most similar) to that data object. After handling all data objects, new Medoids is determined which can represent cluster in a better way and the whole process is repeated. Again all data objects are bound to the clusters depend on the new Medoids. In each repetition, Medoids change their location step by step. In other words, Medoids move in each repetition. This process is continued until no any Medoids change [6].

Input: k, the number of clusters; Dataset (D) containing n objects.

Output: A set of *k* clusters.

Method

1. arbitrarily choose N objects in Dataset as the initial representative objects
2. repeat
3. assign each remaining object to the cluster with the nearest representative object
4. randomly select a non-representative object
5. Compute the total cost of swapping old Medoid object with a newly selected on- Medoid object
6. If the total cost of swapping is less than zero (<0), then perform that swap operation to form the new set of k-Medoid.
7. until no change;

Conceptually, this is done in the following way. The distance of each non-selected point from the closest candidate medoid is calculated, and this distance is summed over all points. This distance represents the "cost" of the current configuration. All possible swaps of a non-selected point for a selected one are considered, and the cost of each configuration is calculated.

$$\text{cost}(\text{x}, \text{c}) = \sum_{i=1}^{d} (\text{xi} - \text{ci})$$

Where *x* is any data object, *c* is the medoid, and *d* is the dimension of the object.

V. EXPERIMENTS

This paper illustrates the use of K-Means and K-medoid clustering algorithms in Rapid Miner. Sample data set used for the analysis is based on the birth and death list data "birth death rates.CSV" [16]. Dataset contains attributes such as ID, BIRTH, DEATH, that all are numeric data. Analyzing the data manually is very difficult and tedious for large data sets it becomes very easy to analyze such data using Rapid Miner, WEKA, MATLAB, Hadoop, etc. Below figures are shows the birth and death data set after preprocessing. Here eight clusters are select in the analysis process.

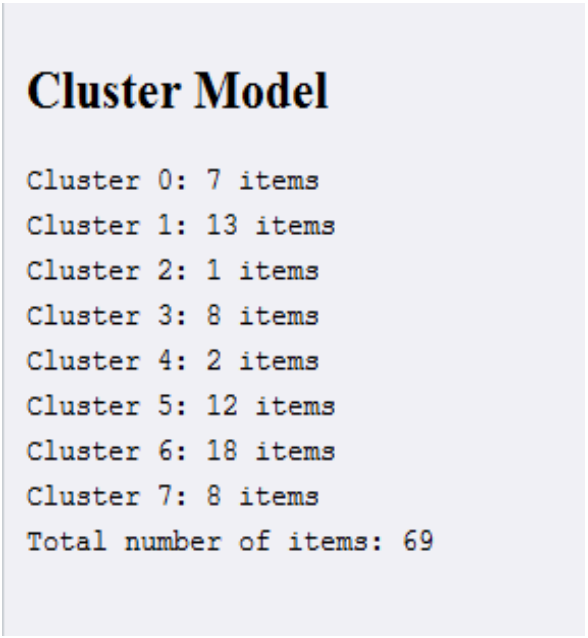
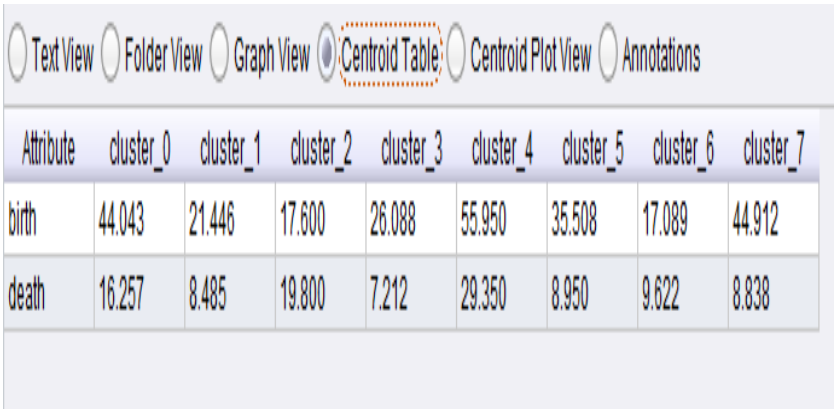


Figure 1.K-means (Text View)

Figure 1 show the Text View of all eight clusters define how many items in particular cluster for K-means algorithm.



Attribute	cluster_0	cluster_1	cluster_2	cluster_3	cluster_4	cluster_5	cluster_6	cluster_7
birth	44.043	21.446	17.600	26.088	55.950	35.508	17.089	44.912
death	16.257	8.485	19.800	7.212	29.350	8.950	9.622	8.838

Figure 2.K-means (Centroid table)

Figure 2 show the Centroid point for all eight clusters, and in K-means Centroid points are may be or may not from data set's item.

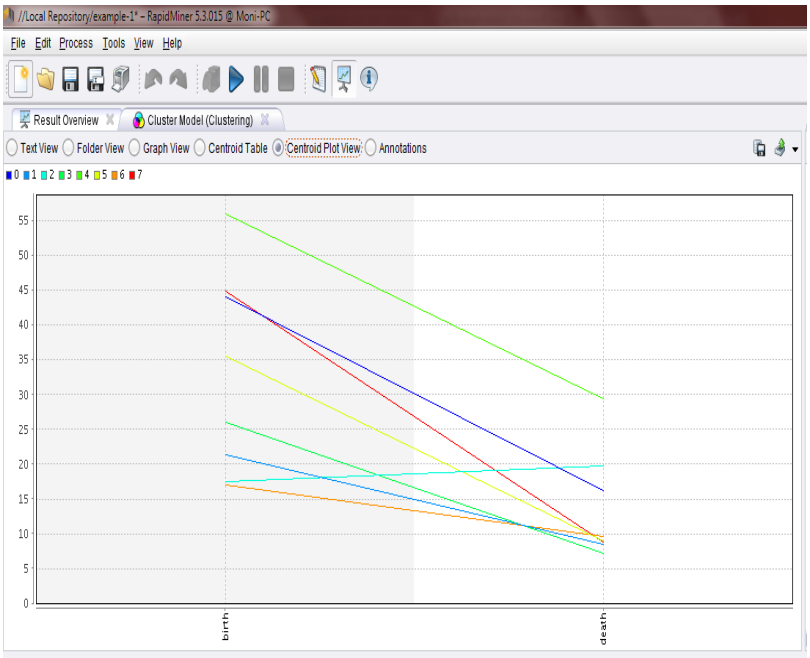


Figure 3.K-means (Centroid Plot View)

Figure 3 show the Centroid Plot View for all eight clusters for K-means algorithm.

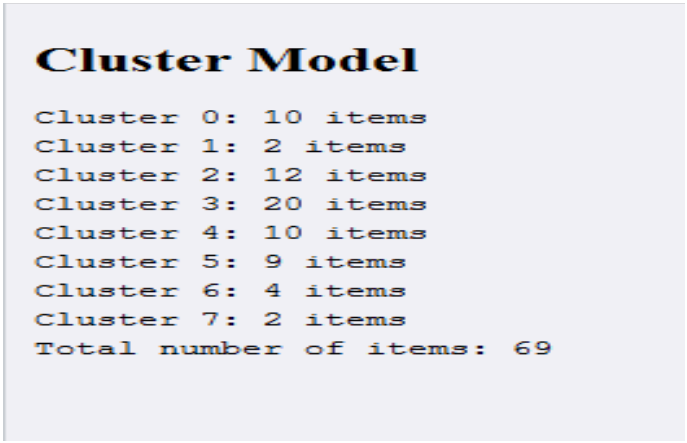
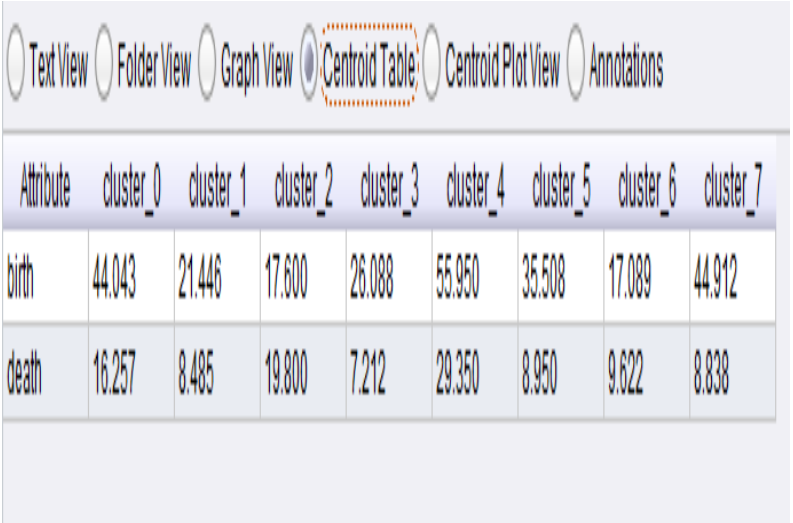


Figure 4.K-medoid (Text View)

Figure 4 show the Text View of all eight clusters define how many items in particular cluster for the K-medoid algorithm.



Attribute	cluster_0	cluster_1	cluster_2	cluster_3	cluster_4	cluster_5	cluster_6	cluster_7
birth	44.043	21.446	17.600	26.088	55.950	35.508	17.089	44.912
death	16.257	8.485	19.800	7.212	29.350	8.950	9.622	8.838

Figure 5.K-medoid (Centroid table)

Figure 5 show the Centroid point for all eight clusters, and in K-medoid Centroid points are all ways from data set's item.

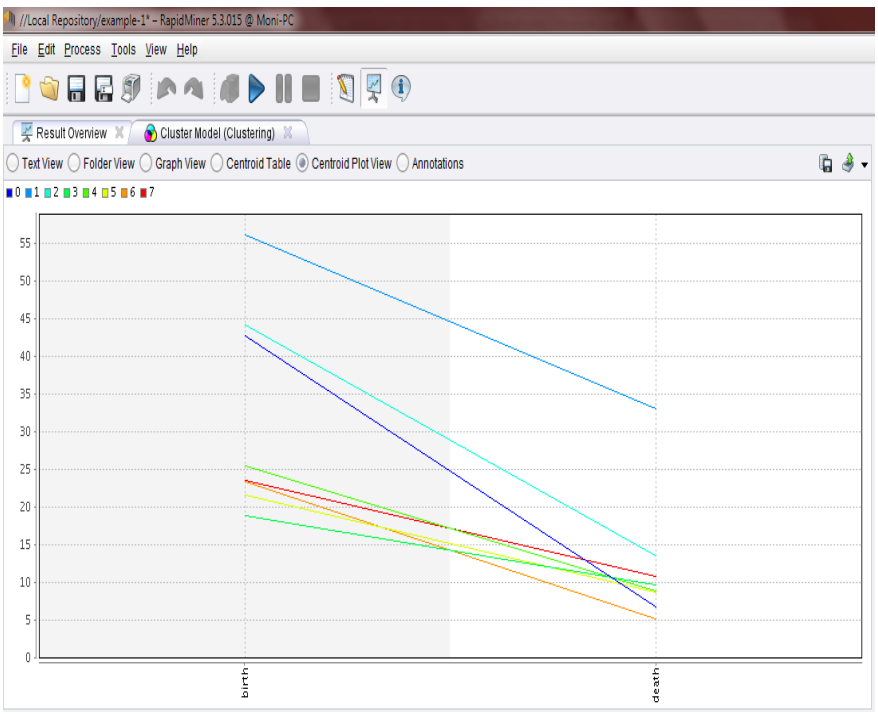


Figure 6.K-mediod (Centroid Plot View)

Figure 6 show the Centroid Plot View for all eight clusters for the K-medoid algorithm.

VI. COMPARISON OF K-MEANS AND K-MEDOID

Table 1.Comparison of K-means and K-medoids
Table 1 defines the comparison between the k-means and k-medoid algorithm.

Parameters	k-means	k-medoids
Complexity	$O(i\ k\ n)$	$O(i\ k\ (n-k)^2)$
Efficiency	Comparatively more	Comparatively less
Implementation	Easy	Complicated
Sensitive to Outliers?	Yes	No
Advance specification of No. of clusters 'k'	Required	Required
Does initial partition affects result and Runtime?	Yes	Yes
Optimized for	Separated clusters	Separated clusters, small dataset

PROPOSED WORK

Survey of different clustering algorithm from different research papers find modified some techniques. In [9] author modified K-means algorithm with Calculates the distance matrix once and uses it for finding new medoids at every iterative step. In this paper define the new modified k-Medoids algorithm is cannot suitable for large data set, that's why now proposed work on K-Medoids algorithm to implement in a parallel environment. In parallel environment apply map and reduce method using K-Medoids algorithm and getting output for large data set, because the map and reduce throw do parallel process on data set and get a fast result and improve the process of K-medoids.

REFERENCES

[1].Dr. Aishwarya Batra, "Analysis and Approach: K-Means and K-Medoids Data Mining Algorithms".
[2].Sunita Kumari, Abha Kaushik, "A Survey on Clustering Problem with Optimized K-medoid Algorithm", ISSN: 2348-4098 Volume 02 ISSUE 04 April-May 2014.
[3].Preeti Arora, Dr. Deepali, Shipra Varshney, "Analysis of K-Means and K-Medoids Algorithm For Big Data", International Conference on Information Security & Privacy (ICISP2015), 11-12 December 2015.
[4].Megha Mandloi, " A Survey on Clustering Algorithms and K-Means", International Journal of Research in Engineering Technology and Management ISSN 2347 – 7539.
[5].Rupali Patil, Shyam Deshmukh, K Rajeswari, "Analysis of Simple K-Means with Multiple Dimensions using WEKA", International Journal of Computer Applications (0975 – 8887) Volume 110 – No. 1, January 2015.
[6].Khaled A. Alenezi, Mohammad A. Alahmad, Walid Aljoby, "Propose Parallelization of K-Medoid Clustering Algorithm", Journal of Advanced Computer Science and Technology Research, Vol.4 No.4, December 2014, 101-107.
[7].Meenakshi Sharma, "Data Mining: A Literature Survey", International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-3, Issue-2),feb-2014.
[8].Shraddha Masih, Sanjay Tanwani, "Data Mining Techniques in Parallel and Distributed Environment-A Comprehensive Survey", International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 3, March 2014.
[9].Hae-Sang Park, Jong Seok Lee and Chi-Hyuck Jun, "A K-means-like Algorithm for K-medoids Clustering and Its Performance" Department of Industrial and Management Engineering, POSTECH San 31 Hyoja-dong, Pohang 790-784, S. Korea.
[10].Hae-Sang Park, Chi-Hyuck Jun, "simple and fast algorithm for K-medoids clustering", Expert Systems with Applications 36 (2009) 3336–3341.
[11].Tapas Kanungo, Nathan S. Netanyahu, Angela Y. Wu, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation", IEEE Transaction on pattern analysis and machine intelligence, vol. 24, no. 7, July 2002.
[12].Shailendra Singh Raghuwanshi, PremNarayan Arya, "Comparison of K-means and Modified K-mean algorithms for Large Data-set", International Journal of Computing, Communications, and Networking, Volume 1, No.3, November – December 2012.
[13].Harshawardhan S. Bhosale, Prof. Devendra P. Gadekar, "A Review Paper on Big Data and Hadoop", International Journal of Scientific and Research Publications, Volume 4, Issue 10, October 2014.
[14].Mansi shah, Vatika Tayal, "Future of Big Data beyond Batch Processing", IJSRD - International Journal of Scientific Research & Development, Vol. 3, Issue 01, 2015.
[15].http://www2.cs.uregina.ca/~dbd/cs831/notes/kdd/3_algs_and_methods.html
[16].https://vincentarelbundock.github.io/Rdatasets/datasets.html

