An Efficient High Dimensional Data Clustering Using MapReduce

Avinash Dhanshetti
Department of Information Technology,
Pune Institute of Computer Technology,
Pune, India
avinashdhanshetty@gmail.com

Tushar Rane
Department of Information Technology,
Pune Institute of Computer Technology,
Pune, India
ranetushar@yahoo.com

Dr. S. T. Patil
Department of Computer Engineering,
Vishwakarma Institute of Technology,
Pune, India

***Abstract* —Data Clustering is key point used in data processing algorithms for Data Mining. Clustering is a data mining technique used to place data elements into related groups without advance knowledge of the group definitions. Popular clustering techniques include k-means clustering. Clustering is imperative idea in data investigation and data mining applications. In last decade, K-means has been popular clustering algorithm because of its ease of use and simplicity. Now days, as data size is continuously increasing, some researchers started working over distributed environment such as MapReduce to get high performance for big data clustering.**

Keywords: Clustering, Map-Reduce, K-Means, Distributed-Environment.

I. INTRODUCTION

Clustering is a process of grouping objects with some similar properties. Any cluster should exhibit fundamental properties, low between class comparability and similarity. Clustering is an unsupervised learning i.e. it adapts by perception instead of illustrations. There is no predefined class conditions exist for the information focuses. As every other issue of this kind, it manages discovering a structure in a gathering of unstructured data. Clustering is a division of data into groups of similar objects. The following diagram illustrate how data looks after forming the clusters; the similarity criterion consider in diagram is shape. Objects in the datasets belong to the same cluster if they are “close” according to their shape.

Lately, cluster analysis has experienced overwhelming improvement. There are several types of clustering algorithms such as density-based clustering, hierarchical clustering, grid- based clustering, partitional clustering and model-based clustering. Each of them has its own style which results in optimising the performance of algorithm.

The rest of this paper is organized as follows. Section 2 gives brief introduction of different Clustering algorithms. Section 3 gives overview of research work performed for clustering algorithm, to get high clustering performance for big data using MapReduce. As we know, solving clustering problem exactly is NP-hard, even with just two clusters. Regarding the expansion of the data size and the limitation of a single machine, a simple solution is to consider distributed environment for computation. Proposed mechanism for An Efficient Data Clustering using MapReduce is discussed in Section 4 and Section 5 presents the expected results. Section 6 concludes.

According to Vladimir Estivill-Castro, the meaning of a "cluster" cannot be precisely defined, which is one of the reasons why there are so many clustering algorithms [2]. Generally we can say a group of data objects. However, different researchers exploit various cluster models, and for each of these cluster models again different algorithms can be given. Understanding these "cluster models" is key to understanding the differences between the various algorithms. In the following diagram we can 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 example geometrical distance is considered). This is called distance-based clustering.

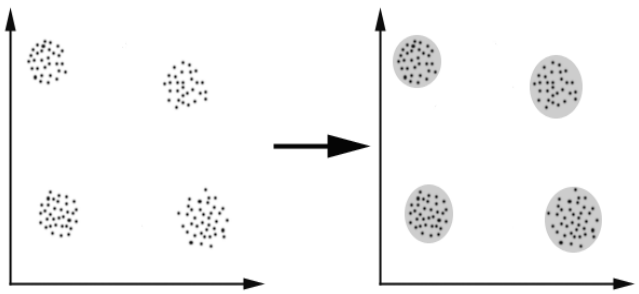

Figure 1. Data Clustering

II. OVERVIEW OF DIFFERENT CLUSTERING METHODOLOGY USING MAP-REDUCE

As of late, with the development of computer technology, some of cluster algorithm were developed quickly and applied broadly.

There are several variations of K-means algorithm; some of them are as following:

- The fuzzy C-means algorithm is a soft clustering method of K-means, where every data point is associated with some value which represents a degree of belongingness to cluster, as in fuzzy logic.
- Some seeding technique is proposed for selecting the initial values (seeds) for the K-means clustering algorithm. K-means++ [9] is one of the techniques to effectively overcome the issues associated with the occasionally results in poor clustering by the standard K-means clustering algorithm.
- The kd-tree (k-dimensional tree) is a space-partitioning data structure for arranging points in a k-dimensional space. Kanungo et al. [10] presented a simple and efficient implementation of K-means clustering algorithm, which is known as filtering algorithm. This algorithm is easy to implement, as it requires a kd-tree as the only major data structure, which results in speeding up each step of K-means.

Zhao W et al. [6] proposed parallel K-means clustering using MapReduce and gave a detailed description for the algorithm, and Moseley B et al. [7] gave the first analysis that shows several partitional clustering algorithms in MapReduce. However, not all big data processing problems can be efficient by parallelism; a research shows that partitional clustering algorithm requires exponentially times iterations [8]. In the meantime, the time for job creation and of big data shuffling are hard to swallow especially when data is of high volume, therefore just parallelism is not enough. Jing Zhang et al. [12] proposed a parallel K-means algorithm based on MPI, called MK-means, which is composed of SK-means (Sequential K-means) and MPI.

III. RELATED WORK

Various methodologies have been proposed for improving performance of big data clustering using MapReduce. Here we have described some of these methods.

Alina Ene et al. [4] proposed approximation algorithms for the k-center and k-median problems that run in a constant number of MapReduce rounds. And also presented the evaluated results that the analysis used for the k-median problem can be extended to the k-means problem in Euclidean space in which a MapReduce algorithm runs in a constant number of rounds and achieves a constant factor approximation.

The algorithm Iterative-Sample [6] maintains a set of sampled points S and a set of points R that contains the set of points that are not well represented by the current sample. The algorithm repeatedly includes new points to the sample. By adding more points to the sample, S represents more accurate points. More points are added to S until the number of remaining points decreases below the threshold. The pivot point v is chosen to determine which points are well represented: if a point x is closer to the sample S than the pivot v, the point x is included in sample S and removed from R. Finally, Iterative-Sample returns the union of S and R. And author proposes the following MapReduce which run on the result of Iterative Sample which says to Map Sample C, which is returned by Iterative Sample algorithm and all of the pairwise distance between points C to a reducer. And the reducer runs a k-center clustering algorithm on C to find final clusters.

Zhao W et al. [8] proposed PKMeans algorithm which needs one kind of MapReduce job. Authors presented three function MAP, REDUCE and COMBINER. The map function, which assigns each sample data set to the closest center while the reduce function performs the procedure of updating the new centres, which results in minimizing network communication cost and a combiner function deals with partial combination of the some values with the same key and map task. The input dataset to the map function is stored on HDFS [13] as a sequence file of <key, value> pairs, which represents a record in the dataset. The key is the offset of the record at the starting point of the data file, and the value is a string of the content. The dataset is split and globally broadcasted to all mappers. Consequently, the distance computations are parallel executed. For each map task, PKMeans construct a global variant centres which is an array containing the information about all centres of the clusters. A mapper can compute the closest center point for each sample. The intermediate values are then composed of two parts: the index of the closest center point and the sample information. After performing map task of each mapper, combiner is applied to combine the intermediate data of the same map task. Since the intermediate data is stored locally, this does not increase the communication cost. In the combiner function, the values of the points assigned to the same cluster are partially added. In order to calculate the mean value of the objects for each cluster and record the number of samples in the same cluster in the same map task. The input to the reduce function is the data obtained from the combiner function of each node of distributed environment. As described in the combiner function, the data includes partial sum of the samples in the same cluster and the sample number. In reduce function, sum all the samples and compute the total number of samples assigned to the same cluster. Therefore, new centres is returned which are used for next iteration.

Caetano Traina et al proposed an adaptive, hybrid method named BoW (Best of both Worlds) that exploits the advantages of the previously described approaches, taking the best of them. There is no universal winner, since it depends on the environment and on the data characteristics. Therefore, the main question here is: When should sampling-and-ignore idea be used and when should it be avoided? Parallel Clustering runs the map, shuffle and reduce phases only once on the whole dataset. Sample and Ignore reduces the amount of data to be shipped to and processed by the reducers, at the cost of a second pass on the input data (in the map phase). Author proposed a cost-based optimization that uses analytics

models to estimate the running time of each clustering strategy. BoW picks the one with the lowest estimated cost.

Authors supported there methodology by evaluating the performance of their proposed algorithm with respect to speed up, scale up and size up. The experiments results show that the Zhao W et al. [4] proposed Parallel k-means clustering based on MapReduce algorithm can process large datasets on commodity hardware effectively.

Xiaoli Cui et al. [10] compared the results with traditional K-means, Kmeans++, Kmeans‖, WMC and DMC clustering algorithms by considering DBI on different sized data sets. Data Sets considered by authors for evaluation with the parallel implementation in the Hadoop framework were from real-world settings and are publicly available from the UC Irvine Machine Learning repository. The Bag of Words Data Set (BoW) consists of 2,351,710,420 points in 3 dimensions and represents features available to (docID, wordID, count) and the individual household electric power consumption Data Set (House) consists of 4,296,075,259 points in 9 dimensions.

IV. PROPOSED MECHANISM

Fuzzy C-Means algorithm works by assigning belongingness to each data object with respective to each cluster center on the basis of distance between the cluster center and the data objects. Closest the data object to the cluster center, the data object belong to that particular cluster. We can say, summation of belongingness of each data object should be equal to one. After each iteration belongingness and cluster centres are updated accordingly. Fuzzy C-Means gives best result for overlapped data set and comparatively better then k-means algorithm. Unlike K-Means where data object must exclusively belong to one cluster center here data point is assigned belongingness to each cluster center as a result of which data point may belong to more than one cluster center. The following diagram illustrates the proposed system architecture.

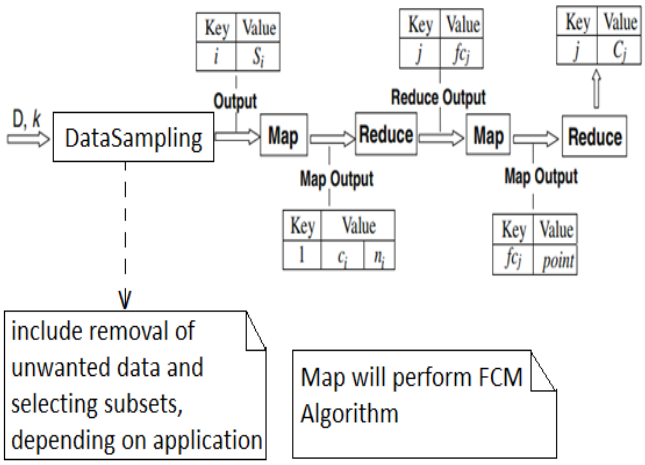

Figure 2. Proposed System Architecture.

Proposed system will be divided in following four modules:

A. Data Sampling:

It is pre-processing step which includes removal of unwanted or corrupted data from data set and select sample subsets.

B. Clustering:

Core module of the system, which performs the data clustering algorithm.

C. Mapper:

Distributing the task with the selected sample sets to the task processor nodes i.e for distributed computing.

D. Reducer:

Aggregating all the resulting clustering which are computed on distributed task processor nodes.

V. EXPERIMENTS AND EXPECTED RESULTS

In order to evaluate our efficient clustering algorithm in practice, we present the experimental setup for evaluating. We set up a cluster of n number of machines, each of them has a Core i3 (4th Gen) 1.7 GHz CPU, 500GB hard drive, 4 GB RAM, 10/100 Mbps Ethernet Controller and with any OS. All nodes are connected to a 100 Mbps Ethernet switch. We use Akka Clustering to form network Cluster for distributed computing and compile the source codes under JDK 1.7 in Eclipse 4.2. Processing Node Actor run on each machine. A single ClusterManager Actor runs on any one Machine. In case, if the Machine crashes where the ClusterManager is running it other ClusterManager Actor will be started on other Node in the cluster. We use live captured network packets as a datasets to evaluate the performance of our efficient clustering algorithm.

A. First step is Data Sampling:

ClusterManager will read the dataset file and remove the corrupted data packets and positive ACK packets which results in -reducing the number packets to be processed. We are performing this step because the number positive ACK are more and they don't use for analysis.

B. Second step is Mapping the dataset file for distributed computing:

As ClusterManager is Singleton Actor running in Cluster environment will read the dataset file divided it among the Processing Node who will perform the C-Means Clustering Algorithm.

C. Third step is perform C-Means Clustering Algorithm :

Once the each of the processing Node gets the different files to process individual they will perform Data Clustering C-Means Algorithm and send response to the ClusterManager.

D. Fourth step is to Merge:

Once ClusterManager gets response from all the Processing Nodes it Merge the result and forms the final clusters.

We compare the performance of our efficient Clustering algorithm against the traditional K-means algorithm by distributing the task among different machines using Akka Clustering and stand-alone on single Machine. For sample testing purpose we captured 20000 live captured network packets after performing Data Sampling step the number of packets reduced to 14842 packets. On stand-alone machine without data sampling it took near about 115 sec and with sampling it took 89 sec and when we tested the same algorithm with data sampling and distributed computing task execution completed within 52 sec. Our Primary aim is to improve the cluster accuracy. Hence, we will be implementing the C-Means clustering Algorithm with the same setup and analyze the performance and cluster accuracy.

VI. CONCLUSION

As data clustering has received a significant amount focus in data mining, many clustering algorithms have been proposed in the previous decades. But still, the continuously increase in the size of data volume makes clustering a challenging task. Here we have proposed an efficient and optimized method for clustering of multi-dimensional dataset using MapReduce. This system will work as pre-processing algorithm in any data mining application or methodology. No matter how good anything can be there is always a scope of improvement. Primary aim of any improvement is its accuracy. Here we have tried to improve the performance of clustering of high dimensional data set using MapReduce.

VII. ACKNOWLEDGMENT

I would like to thank Prof T. A. Rane for his constant guidance and help. I would also like to thank Dr. Emmanuel M. for the encouragement.

VIII. REFERENCES

[1] Avinash Dhanshetti, Tushar Rane, "A Survey on Efficient Big Data Clustering using MapReduce" Data Mining and Knowledge Engineering 7, No 2 (2015): 47-50

[2] Estivill-Castro, Vladimir. "Why so many clustering algorithms: a position paper." *ACM SIGKDD Explorations Newsletter* 4.1 (2002): 65-75

[3] Borthakur, D.:The Hadoop Distributed File System: Architecture and Design (2007)

[4] A. Jain, R. Dubes, "Algorithms for Clustering Data", Prentice Hall, 1988.

[5] Alina Ene, Sungjin Im, Benjamin Mosele, "Fast clustering using MapReduce" Proceedings of the 17th ACM SIGKDD Internation Conference, ACM New York 2011

[6] Zhao W, Ma H, He Q, "Parallel k-means clustering based on MapReduce", Cloud computing- Springer, Berlin Heidelberg 2009.

[7] Ene A, Im S, Moseley B "Fast clustering using MapReduce", Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp. 681–689 2011

[8] Vattani A (2011), "K-means requires exponentially many iterations even in the plane [J]", Discret Comput e Geom 45(4):596–616

[9] D. Arthur, S. Vassilvitskii, "k-means++: the advantages of careful seeding", Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pp. 1027- 1035, 2007.

[10] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, "An Efficient K-means Clustering Algorithm: Analysis and Implementation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 881-892, 2002.

[11] Xiaoli Cui, Pingfei Zhu, Xin Yang, Keqiu Li, Changqing Ji, "Optimized big data K-means clustering using MapReduce " Springer Science Business Media New York 2014

[12] Jing Zhang, Gongquing Wu, Xuegang Hu, Shiying, Shuilong Hao, "A Parallel K-Means Clustering Algorithm with MPI", Parallel Architectures, Algorithms and Programming (PAAP), 2011 Fourth International Symposium, pp. 60-64.

[13] Farnstrom F, Lewis J, Elkan C (2000) Scalability for clustering algorithms revisited [J]. ACM SIGKDD Explor Newsl 2(1):51–57

.