A Review on Clustering Algorithms Applicable for Map Reduce

Dr.Rajashree Shettar
Professor, Dept. of CSE
R V College of Engineering
Bangalore, India

Mr.Bhimasen.V.Purohit
PG Student, Dept. of CSE
R V College of Engineering
Bangalore, India

Abstract – Big data is a new trend and big data analytics is gaining more importance among the data analyzers. Many researches are going around applying different algorithm to extract useful information from a large data set. The processing of big data can be handled through parallelizing the work. For this Hadoop has come with a new software programming model called as MapReduce. Only few Data mining algorithms are successfully implemented in MapReduce fashion and are applied to massive dataset. In this survey different clustering algorithms which can be applied for big data using MapReduce are discussed.

Key Terms — Hadoop, MapReduce, K-Means, MRDBSCAN, Canopy clustering.

I. INTRODUCTION

Extracting the knowledge from a data which will be useful to understand the data pattern is called data mining. Data mining has many techniques like supervised learning like classification and unsupervised learning like clustering. Classifying the data objects into groups with similar characters is called clustering. Clustering is major technique which can be applied to many industrial settings. A huge community of researchers and developers are using the clustering algorithms to apply in many fields like image processing, gene sequence analysis, weather forecasting etc.

The data produced in Flickr a public picture sharing site, is around 3 TB every single day. 2.5 quintillion bytes of data will be generated all over the globe every day. This increased use of internet has created enormous amount of digitized data, processing of this big data poses different challenges for analyzers [1]. Extracting required patterns or hidden information from this huge data will become tedious if the conventional approaches are used. Also the normal methods are time consuming and less efficient. Hence parallel processing of data has gained major importance.

As the MapReduce paradigm has gained popularity among researchers for parallel processing of data, many data mining algorithms are now implemented using this technique. MapReduce model of programming allows the working with huge data. The conventional algorithms like classification, clustering algorithms need to be implemented with MapReduce so that they can be used to analyze the huge data more efficiently [2].

II. BACKGROUND

A. Hadoop

Hadoop is the new tool to process big data. Hadoop is open software developed by apache. Hadoop has in built capacities to handle big data. Hadoop comes with two major aspects. One is HDFS (Hadoop Distributed File System) which is to handle the data, and another one is MapReduce which is a programming model.

HDFS stored the files in blocks of 64MB. It is designed to store the files of varying size from 100MB to GB, TB. Hadoop can be established as single node cluster or multinode cluster. Every Hadoop cluster has five running nodes namely Namenode, Secondary Namenonde, Datanode, Jobtracker, Tasktracker. Namenode maintains the metadata of HDFS. Secondary Namenode is responsible for validating Namenode and updating its information. Datanode stores the data. Jobtracker is the one which takes job from the user and divides it into tasks. These tasks are given Tasktrackers which will carry out the required task [2]. Figure.1 shows the data flow in HDFS.

![Fig.1 – Data flow in HDFS](image)

B. Map Reduce Paradigm

Hadoop map reduce is a software programming model which can be implemented to process large amount of data. A mapreduce task breaks down the input data into chunks. Mapreduce task has three phases. The map phase is the first phase where the data is split among the nodes of a cluster and <key, value> pairs are generated. The output of this map is sorted and given as input to reduce phase. After this, the results are combined to provide a final result.

Mapreduce works in completely parallel fashion and hence the processing speed will be high. It is also responsible to handle data among all machines, scheduling tasks, and re-executing the failed tasks [3].

III. CLUSTERING ALGORITHMS

A. K-Means algorithm

K-means algorithm is the most widely used algorithm in clustering. The algorithm takes an input k, which refers to the number of clusters that should be generated and n which
is set of objects. Initially the cluster centers are chosen randomly. Then using any distance functions like Euclidean distance or Manhattan distance, the centroids are calculated. The algorithm works iteratively until the clusters are finalized.

This algorithm can be implemented in mapreduce pattern as follows.

Map function: The HDFS stores input data as sequence file of <key, value> pairs [4]. Every <key, value> pair represents a record. The data is split and distributed across all mappers. Global variant “centers” is created which stores the information about cluster centers. A mapper then computes the closest center for each record.

Combine function: After mapping, the combiner will combine intermediate data of same mapper. The intermediate data is located in local disk of host. In this phase the value of points are summed up.

Reduce function: the output of combine function is given as input to this phase. Here all the values from all nodes are summed and final result will be obtained. New centers are generated which can be used for further iterations [5].

The figure.2 shows the data flow in K-Means algorithm [9].

**B. DBSCAN Algorithm**

Another important concept in clustering is density based cluster. The idea here is the points that form a dense region are grouped to form a cluster. A fixed threshold value is used to find the dense region. DBSCAN (Density Based Spatial clustering of Applications with Noise) is an important density based algorithm which is widely used in researches. The major advantages of this DBSCAN are it handles noise data efficiently [6].

DBSCAN algorithm, though efficient than other clustering algorithms, cannot handle massive data. It again suffers from bottleneck. Hence mapreduce pattern is applied to DBSCAN. The mapreduce technique works as follows.

**Fig.2 – K-Means on Mapreduce**

First data set should be partitioned and distributed among the nodes for processing. The DBSCAN requires a global index structure, but it creates extra communication cost. Therefore a distributed index is used for efficient parallel processing [7].

The data points of same cluster may be scattered among different nodes. These scattered points should be merged. The figure.3 shows the phases of mapreduce based DBSCAN.

**Fig.3 – Phases of MR-DBSCAN**

The first phase in this process is called as “Partition with reduced boundary points”. Here the input data is divided among nodes of cluster and minimization of boundary points will be carried out. This minimization will help in increasing the efficiency of clustering, and also merging will be done in better manner. The second phase is “DBSCAN-Map” in which DBSCAN algorithm is executed on every node. This execution uses kd-tree [8] space index. It will be executed locally on divided and assigned data. The third phase which is known as “DBSCAN-Reduce” where the point indexes between partitions are found. Also the clustering ID (CID) of the points is calculated. “Merge Result” is the fourth phase in this process. Here the results obtained from DBSCAN-Reduce phase are merged. After merging, based on boundary points, the global structure of clusters is discovered. The last phase is Relabel phase, in which the results from local clustering from every data partition are relabeled and global clusters are identified.

**C. Canopy Clustering**

Canopy clustering is an unsupervised clustering algorithm. It is used as pre-clustering method. The output is given as input to other classic clustering algorithms. This pre-clustering helps in speeding up of clustering in other major clustering algorithms. This helps in processing large datasets [9].

The algorithm has following steps,

1. A list of data points named X is created
2. two threshold values T1 and T2 are decided, where T1>T2.
3. Randomly pick one data point, which represents a canopy centroid.
4. Calculate distance d for all points from this centroid point.
5. IF d<T1 add the point to canopy. If d<T2 remove the point from X.
6. Repeat the steps 3-5, till all points are visited.

The mapreduce function involves, generating <key, value> pairs of centroid and other points in map phase. In reduce phase the results are combined.

**D. MinHash Clustering**

MinHash clustering is a type of probabilistic method. It belongs to the family Locality Sensitive Hashing (LSH) algorithms. This type of clustering used where clustering
should be done based on range of dimensions of the data points [10]. The probability of data point pair assigned to a single cluster is proportional to the overlap between set of items [9].

CONCLUSION AND FUTURE WORK

The generation of data which has led to the new trend of Big-data needs efficient algorithms to work upon that. Conventional algorithms suffer from performance bottleneck when applied on large set of data. Hence the concept of MapReduce has been evolved.

The K-Means algorithm discussed in this survey is efficient for large set of data but it cannot handle noisy data. The DBSCAN algorithm is well known for handling noise in data set. The mapreduce version of this algorithm also discussed here. Canopy clustering and Minhash clustering also can be implemented in MapReduce fashion.

The future work is suggested to implement the combination of two or more these algorithms for better performance. The combination of K-Means and DBSCAN algorithm can be implemented on MapReduce so that the deficiencies of both algorithms can be overcome and a better accurate clustering can be achieved. And also, the implementation can be done on Apache Mahout.

REFERENCES


