A Comparative study of Clustering Algorithms using MapReduce in Hadoop

Dweepna Garg [1], Khushboo Trivedi [2], B.B.Panchal [3]
[1] *Department of Computer Science and Engineering, Parul Institute of Engineering and Technology, Limda, Vadodara, Gujarat, India,*
[2] *Department of Information Technology, Parul Institute of Engineering and Technology, Limda Vadodara, Gujarat, India,*
[3]*Department of Information Technology, Government Engineering College, Modasa, Gujarat, India*

Abstract:-

Hadoop is a distributed file system and an open-source implementation of MapReduce dealing with big data. Facebook, Yahoo, Google etc. makes use of Hadoop to process more than 15 terabytes of new data per day. Data clustering is a part of machine learning and it has high applicability in industries and also in various fields such as image processing, recommendation systems, text analytics etc. In this paper three clustering algorithms are described – K-mean, canopy clustering and Fuzzy K-mean clustering implemented by both MapReduce and sequential approach. MapReduce paradigms are able to solve many problems related to big volume of data by modelling algorithms in map and reduce strategy and high volume data can't be fit into memory for clustering. This is the reason as to why MapReduce paradigms have gained popularity for clustering in big data. These algorithms work with data in portioned form and need to consider the distributed nature of portioned data and model the algorithm accordingly. Apache Mahout, an open source implementation is used in various organizations and was developed by a team of active contributors.

Keywords: MapReduce, Machine Learning, K-mean, Canopy clustering, Fuzzy K-mean clustering.

I. INTRODUCTION

As data and work grow, it takes a longer time to produce results. To produce the result in timely manner, one should start thinking big. A certain way is carried out to spread the work across many computers i.e. one needs to scale out. MapReduce is a programming model which is designed for processing large volumes of data in parallel. It does so by dividing the work into a set of independent tasks. MapReduce programs transform lists of input data elements into lists of output data elements. The MapReduce data elements are immutable, i.e. they cannot be updated. The policy is to place the file into HDFS once and can read the file 'n' number of times.

In this paper, we focus on various clustering algorithms using MapReduce in Hadoop.

I. MAPREDUCE

MapReduce is a programming paradigm which processes the portioned data and aggregates the intermediate results. Google introduced and patented MapReduce. It can also be defined as a software framework supporting distributed computing on large datasets on clusters of computers. Programs can be implemented in any language to run the jobs which are written in MapReduce paradigms. The inspiration of MapReduce came from two functions namely map( ) and reduce( ) functions in functional programming model

map function

A function is applied on the input individual chunks of portioned data in map( ) function in functional programming. This portioning is done by the Hadoop Distributed file system (HDFS). The portioning size is a tunable parameter.

For example, there is a processing function f which converts the input rectangle to a sphere. So parameters would be like – map(f,input)

f (▭) = ⬭

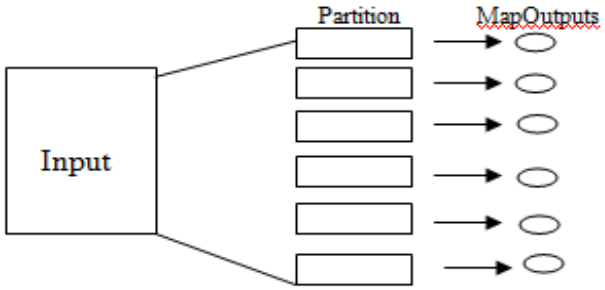

Figure1. map (f,input) applied to input

Map function is applied on individual partition.

The input is taken as a key-value pair in the form of records in Hadoop MapReduce [9]. The function of the map( ) is to take the input values and input key in order to produce the intermediate value list of the processed keys.

map (input_key, input_value) ----> (output_key, intermediate_value_list)

Different input datasets produce different intermediate values and hence it describes the feature that map( ) functions run in parallel.

reduce function

The intermediate values are combined to form a list. This is done after the map( ) is over. That is the intermediate values are combined to get a final result for the same output key. Same as map( ), the reduce( ) also runs in parallel and each of the reduce( ) function runs on a different output key which are genrwted by a map( ) function. reduce( ) function only starts after the end of map( ) function

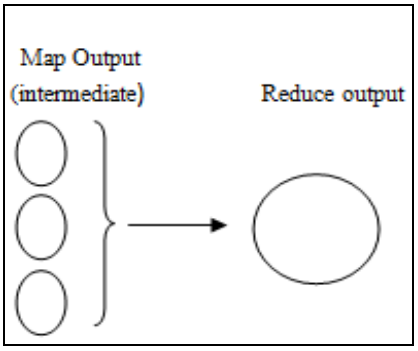

Figure2. reduce (r, intermediate_value_list)

In the above figure r is applied to the intermediate value list to get final reduce output.

r (⬭, ⬭, ⬭) = ◯

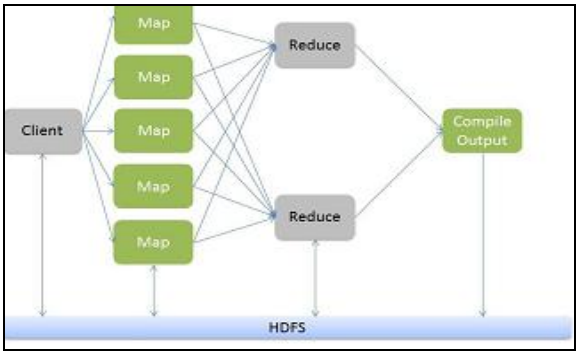

Figure3. MapReduce

II. MACHINE LEARNING

Machine learning [3] is a branch of artificial intelligence which is concerned with the construction and study of systems that can learn from data. Say for example, a machine learning system could be trained on email messages to learn to distinguish between spam and non-spam messages. After learning, it can then be used to classify new email messages into spam and non-spam folders. So basically it is a study which understands the input, make predictions from the input and learn from the feedback. There is a wide variety of machine learning tasks and successful applications. A classic example of machine learning is "optical character recognition", in which printed characters are recognized automatically based on previous examples.

Machine learning is classified into following subcategories such as supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning and learning to learn.

(i) Supervised learning: - It is a machine learning task of inferring a function from labelled training data. The training data consist of a set of training examples. In this, each example consists of an input object (typically a vector) and a desired output value (also called the supervisory signal). It analyzes the training data and produces an inferred function, which can be used for mapping new examples. Google Prediction API is an example of supervised learning. Here a training set is taken as an input and after getting trained, it predicts the language of the input by making a model based on the training input.

(ii) Unsupervised learning: - It does not follow the train-test model. In

machine learning, the problem with this learning is that of trying to find hidden structure in unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning. This unsupervised learning finds the representation of input data. Clustering is one of the best approaches in this.

(iii) Semi-supervised learning: - It is a class of machine learning techniques that combines both supervised and unsupervised learning i.e. it makes use of both labelled and unlabeled data for training. Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data).

(iv) Reinforcement learning: - It is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. It learns from the feedback to the output.

(v) Learning to learn: - in this, experience is used as a part of learning. When one learns about a problem which is similar to some other problem then experience factor comes into action. A better model is built from the experience of various problems and applied to the problems used for learning simultaneously.

III. UNSUPERVISED LEARNING

As stated above, clustering is one of the best approaches of unsupervised learning. Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is used in fields including machine learning, pattern recognition, information retrieval etc. It is a main task of exploratory data mining.

The purpose of using data clustering is as follows:

I. Natural classification: - identifies the degree of similarity among the organisms.

II. Underlying structure: - to gain knowledge about the data, generating hypothesis, detecting anomalies and identifying the salient feature.

III. Compression: - organizes and summarizes the data through cluster prototypes.

The data points can be clustered using K-mean, canopy clustering and Fuzzy k-means clustering. In rest of the section, the above algorithms will be described along with the approach to implement these algorithms: MapReduce and sequential paradigm. These algorithms are useful in recommendation engine, image processing etc.

A. K-MEAN CLUSTERING

It is popular method for cluster analysis in data mining. It is a method of vector quantization originally from signal processing which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

(i) Classical approach :- The steps of the K-mean algorithm is as follows:

Step 1: Place randomly initial group centroids into the space. These represent the "temporary" means of the clusters.

Step 2: Assign each object to the group that has the closest centroid by calculating the squared Euclidean distance from each object to each cluster.

Step 3: For each cluster, recalculate the positions of the centroids and each seed value is now replaced by the respective cluster centroid.

Step 4: If the positions of the centroids didn't change go to the next step, else go to Step 2.

Step 5: End.

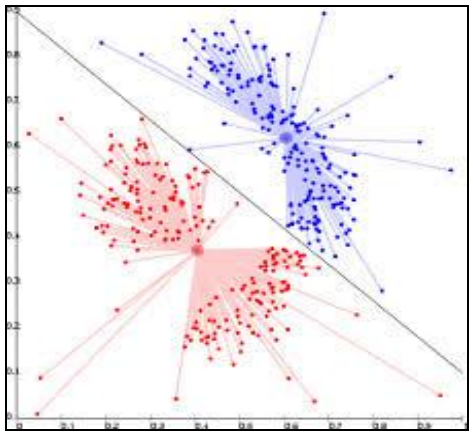

Figure 4: K-Mean clustering

Given a set of observations ($\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k sets ($k \leq n$) $\mathbf{S} = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares (WCSS):

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

where $\boldsymbol{\mu}_i$ is the mean of points in S_i.

This approach does not scale well due to the following reasons:

i. Complexity is high that is of the order k*n*O(distance metric)*num(iterations)

ii. A solution is needed for scaling well with large datasets i.e. the data files of the size of GBs, TB.

MapReduce offers the solution in dealing with large datasets by distributing the tasks to work on smaller chunks of data and making use of distributed system.

Distance metrics: - In order to compute the distance between the points and the cluster centers, the k-mean uses the Euclidean metrics. Mahalanobis distance metric, Manhattan distance are other few other distance metrics which is used by K-mean

(ii) MapReduce Approach

It is based on data portioning and works on keys and values. The assumption of the data points in memory fails in this paradigm. The algorithm is to be designed in such a way so as to parallelize the task and the task does not depend on other splits for any computation.

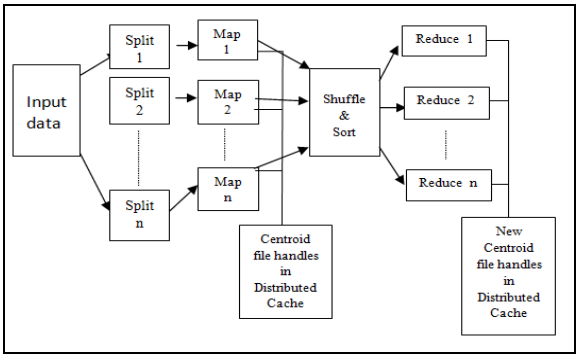

Figure5: Single iteration of K-means on MapReduce

Mappers compute the distance and spills out a key-value pair <centroid_id, datapoint>. This identifies the associativity of the data point with the cluster. Reducers on the other hand work with specific cluster_id and a list of data points associated with it. New mean is computed with the help of a reducer and the reducer also writes to a new centroid file. Now it is the choice of the user regarding the number of iterations to be used, method of algorithm termination or comparison with centroid in previous iteration.

MapReduce based K-mean implementation is also implemented part of Mahout.

B. CANOPY CLUSTERING

It is an unsupervised pre-clustering algorithm, often used as pre processing step for the K-means algorithm or the Hierarchical clustering algorithm [3]. It speeds up the clustering operations on large data sets, whereby the other algorithms may be impractical due to the size of the data set. Briefly the algorithm may be stated as:

i. Cheaply partitioning the data into overlapping subsets (called "canopies")

ii. Perform more expensive clustering, but only within these canopies

Algorithm:

i. Two distance thresholds T1 and T2 are decided such that T1 > T2

ii. A set of points are considered and remove one at random.

iii. Create a Canopy containing this point and iterate through the remainder of the point set.

iv. At each point, if its distance from the first point is < T1, then add the point to the cluster and f the distance is < T2, then remove the point from the set.

v. With the processing in step iv, points close to the original avoids all further processing.

vi. The algorithm loops until the initial set is empty, accumulating a set of Canopies, each containing one or more points. A given point may occur in more than one Canopy.

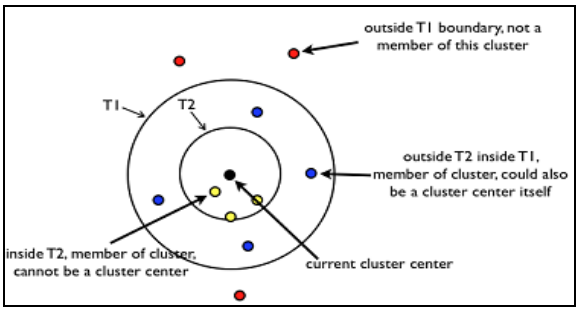

Figure 6: Canopy Clustering

Mahout makes use of just three step processing for finding the canopy centroid:

i. The data is massaged into suitable input format

ii. Each mapper performs canopy clustering on the points in its input set and outputs its canopies' centers. The reducer clusters the canopy centers to produce the final canopy centers

iii. The points are then clustered into these final canopies

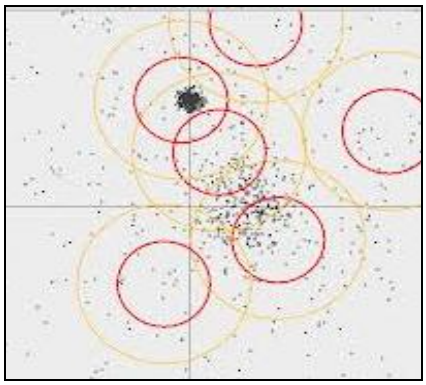

Figure 7: Canopy clustering-Apache Mahout [3]

C. FUZZY K-MEAN CLUSTERING

Fuzzy clustering is a class of algorithms for cluster analysis in which the allocation of data points to clusters is not "hard" but "fuzzy" in the same sense as fuzzy logic. In hard clustering, the given data is divided into distinct clusters and each data element belongs to exactly one cluster. In soft clustering i.e. fuzzy clustering the data elements can belong to more than one cluster, and associated with each element is a set of membership levels. It indicates the strength of the association between a particular cluster and that data element. Hence, fuzzy clustering [4] is a process of assigning these membership levels, and then using them to assign data elements to one or more clusters.

Fuzzy K-Means is also known as Fuzzy C-Means is an extension of K-Means and is one of the popular simple clustering techniques. While K-Means discovers hard clusters, Fuzzy K-Means on the other hand is a more statistically formalized method and discovers soft clusters.

Algorithm:

Fuzzy K-Means works on those objects which can be represented in n-dimensional vector space and a distance measure is defined. The algorithm is similar to k-means and is as follows:

1) Randomly select ‘c’ cluster centers.

2) calculate the fuzzy membership 'µij' using [4]:

$$\mu_{ij} = 1 / \sum_{k=1}^{c} (d_{ij} / d_{ik})^{(2/m-1)}$$

3) compute the fuzzy centers 'vj' using:

$$v_j = (\sum_{i=1}^{n} (\mu_{ij})^m x_i) / (\sum_{i=1}^{n} (\mu_{ij})^m), \forall j = 1, 2, ...c$$

4) Repeat step 2) and 3) until the minimum 'J' value is achieved or $\| U(k+1) - U(k) \| < \beta$.

Here: - ‘k’ is the iteration step.

‘β’ is the termination criterion between [0, 1].

‘J’ is the objective function.

‘U = (µij)n*c’ is the fuzzy membership matrix.

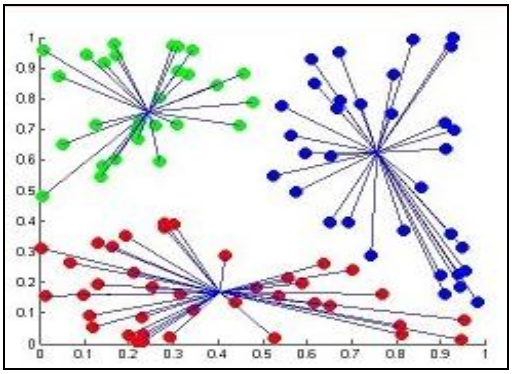

Figure 8: Fuzzy K-mean clustering [7]

(i) MapReduce approach

Similar to K-Means, the program does not modify the input directories. For every iteration carried out, the cluster output is stored in a directory cluster-N. The code sets the number of reduce tasks equal to number of map tasks. So, those many part-0 which are output files. The code uses driver/mapper/combiner/reducer as follows [4]:

a) Fuzzy K-Means Driver – It is similar to KMeansDriver. It iterates over input points and cluster points for specified number of iterations or until it is converged. With every iteration i, a new cluster-i directory is created containing the modified cluster centers obtained during Fuzzy K-Means iteration. This is fed as input clusters in the next iteration. Once Fuzzy K-Means is run for specified number of iterations or until it is converged, a map task is run to output "the point and the cluster membership to each cluster" pair as final output to a directory named "points".

b) Fuzzy K-Means Mapper - During its configure() method it reads the input cluster and then computes cluster membership probability of a point to each cluster. Cluster membership is inversely proportional to distance. Distance is calculated using user supplied distance measure. Output key is encoded clustered and the output values are Cluster Observations containing observation statistics.

c) Fuzzy K-Means Combiner – gets all key_value pairs from the mapper and produces partial sums of the cluster membership probability times input vectors for each cluster. Output key and output values are encoded cluster identifier and Cluster Observations containing observation statistics.

d) Fuzzy K-Means Reducer - Multiple reducers receives certain keys along with the values associated with those keys. An output is a new centroid for the cluster which is produced by the reducer by summing the values. Output key is: encoded cluster identifier and the output value is the formatted cluster identifier.

IV. CONCLUSION

Clustering algorithms are used to find the patterns in data and these clustering algorithms must also scale well with the increasing amount of data. Academia and industry is doing alt to scale the clustering algorithms both vertically and horizontally. MapReduce manages the data partitions and carries on parallel processing on the portioned data. Cluster analysis is sensitive to both the distance metric selected and the criterion for determining the order of clustering. Different approaches may yield different results. The choice of clustering algorithm depends on both the type of data available and on the particular purpose and application. In this paper, algorithms from vast number of machine learning techniques are moulded and focused into MapReduce.

V. FUTURE WORK

Future research will concentrate on centroid generation using canopy clustering and ACO based fuzzy k-means clustering in map reduce paradigm in order to minimize iteration for convergence to reach at final clusters.

References:

[1] Hadoop official site, http://hadoop.apache.org/core/.

[2] Dean, J., and Ghemawat, S.: 'MapReduce: simplified data processing on large clusters', Commun. ACM, 2008, 51, (1), pp. 107-113

[3]Apache Mahout- http://mahout.apache.org

[4]Algorithms- https://cwiki.apache.org/confluence/display/MAHOUT/Algorithms

[5]Ping ZHOU, Jingsheng LEI, Wenjun YE.Large-Scale Data Sets Clustering Based on MapReduce and Hadoop,Journal of Computational Information Systems 7: 16 (2011) 5956-5963

[6] T.Velmurugan,T.Santhanam, A Comparative Analysis Between K-Medoids And Fuzzy C-Means Clustering Algorithms For Statistically Distributed Data Points,Vol 27 No 1 Pp.19-20, Issn:1992-8645

[7] Data Clustering Algorithms:

Https://Sites.Google.Com/Site/Dataclusteringalgorithms/Fuzzy-C-Means-Clustering-Algorithm

[8] A Fuzzy Clustering Algorithm Based On K-Means By Zhen Yan And Dechang Pi In 2009 International Conference On Electronic Commerce And Business Intelligence

[9] Slideshare Site, Http://Www.Slideshare.Net/Varadmeru/Data-Clustering-Using-Map-Reduce