International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

A Competitive Clustering on Bigdata

N. Narasimha Swamy[1], Anita Kumari Singh[2]

[1]M.Tech, Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, India

[2]Phd Scholar, Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, India

Abstract: ***The vast increase in the volume of data has created a need for new applications and algorithms to quickly analyse the large scale of data. Many of the cluster analysis techniques like K-Means are used to compute the data in distributed systems, its accuracy depends on the initial seeding of centroids. The improvisation of K-Means algorithm shows good initial seeding but it suffers with the serial nature i.e., it takes long time on large data sets. In this paper we propose a new algorithm with Map Reduce implementation to address the above problems. Our algorithm provides parallel processing of data by dividing it into number of subsets using Hadoop with MapReducing methods. Our work provides good initial centers in a less time and also produces fast and accurate cluster analysis on large scale data.***

Keywords: K-Means, K-Means++, MapReduce, Mahout, HDFS

1. Introduction

Now- a- days data is growing rapidly with an exponential rate. We need to develop new tools to analyse this abundant amount of data quickly. Even though with the developing computer hardware and high speed internet facilities the problem of handling large scale data is prevailing. With reference to these problems, a few efficient methods like divide and conquer, sampling, distributed computing, incremental learning[2] etc., have been implemented. Thus to improve the computation time of data analytics we are using the dimensionality reduction methods. To retrive knowledge from large amount of data we need new analyzing techniques like cluster analysis.

As the data is very large, applying the traditional clustering techniques may increase the cost abundently. So we need new clustering techniques to improvise the performance of the data extraction. The Algorithms like K-Means are used to cluster large data in a parallel manner but it faces major problem while initializing the initial cluster centers. An effective center initialisation provides good scalable clusters, but to provide the initial centers we must have a prior idea about the data. It is much more difficult to provide the effective centers when the data consists the big data properties like high volume of data, high dimensionality and distributive data storage. Arthur and Vassilvitskii, 2007 introduces K-Means++[3] algorithm to improvise the initial seeding by taking each center at a far distance from each other. K-Means++ also faces two major problems, firstly it is a serial algorithm so it takes very high amount of time to process the data and it is a stocahastic algorithm i.e., it produces different results on the same initial conditions. To improvise the effective initial seeding and execution time of the data we are introducing the new algorithm called Ball K-Means which is also called as competitive K-Means[1] with a mapreduce implementation.

2. Related Work

Clustering is a process of grouping similar objects into a cluster. It is used in many applications like searching, effecient browsing, document classification, recommendation systems, etc. Clustering is mainly categorized into 5 types they are partition based, hierarchical based, model based, grid based and density based clustering. Among those partitioning based clustering is simple and straight forward to implement on the large data. The K-Means[6] algorithm is one of the partition based clustering method it randomly selects the k initial centers where k is the parameter initially taken from the user and further assign the data points to the cluster centers which are near to the center by calculating the distance with each center. The mean of the clusters are calculated and are considered as new centers. The clustering is performed until the effective clusters are obtained. The distance functions used in the clustering are manhattan distance, Euclidean distance, minkowski distance.

Map reducing is a programming model in Hadoop framework. It is used to develop applications for processing the vast amount of data in parallel way and produces very fast results. MapReduce can take the input from the Hadoop Distributed File System (HDFS), process the data and produces the result onto HDFS. At first the data is splitted according to the block size and is stored in multiple systems due to the nature of replication in Hadoop[5]. The data which is splitted is taken as input for the map function and is processed according to the logic present in the map function and produces the result. The output of the map method is taken as input for the reduce method and is further processed according to the logic present in the reduce method and finally produces the desired result. In between the map and reduce methods the hadoop framework will execute the predefined methods for sorting and shuffling the data. Map Reduce[4] can execute the data parallelly in many systems and combines the results. The process itself shows the high reduction of execution cost in the project. Map Reducing is the good solution for analyzing large amounts of data.

Many of the data mining algorithms are implemented in mapreduce on hadoop. We will produce a tool with inbuilt programs to implement on the large scale data. Mahout is a open source project which is built on top of the hadoop. Mahout is started as a sub project of Lucene. It implements machine learning algorithms including recommendation systems, classification and clustering. Many companies uses Mahout for their project areas due to its scalability. The top

features of mahout are - The algorithms produces effictive results in the distributed environment and it fits good with the cloud environment, It provides in-built code and is easy to use algorithms on very large scale data. It has inbuilt distributed fitness function capability. It produces fast results in analysing large datasets. It provides many mapreduce algorithms[7][8] for clustering like k-means, fuzzy k-means, canopy, mean shift and dirichlet algorithms. Which scales well with distributed large amount of data.

The Mahout data flow is shown in fig 1. It takes the input from the file and convert it into a sequence file because mahout applies the algorithms only on the sequence files and produces the results also in the form of sequence files. In between it performs the map and reduce methods according to the user code.

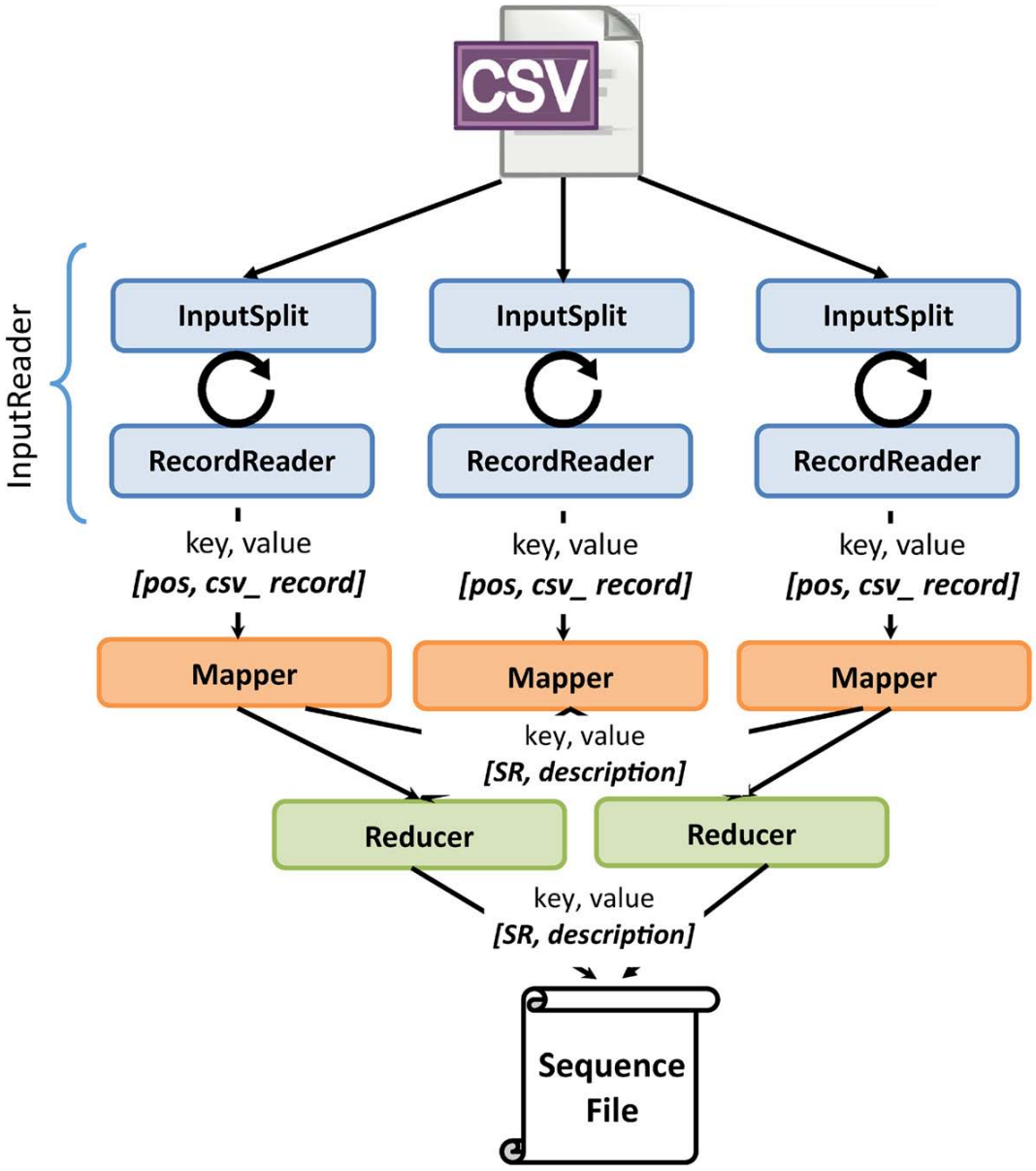

Figure 1: Data Flow Diagram

3. Methodology

The work flow of K-Means++ is shown in below algorithm2 it first selects the initial centroid and then selects the other centroids one by one in a sequential way. Initially a set of data points X and number of centroids k are given.

Algorithm 1:

1: IC ← a single data point uniformly sampled at random from X
2: While || IC || <k do
3: For each data point dp ∈ X, compute D(dp, ic),
where D is the shortest distance from x to the closest ic ∈ IC
4: Sample dp ∈ X with probability
$$D2(dp , ic) = \Sigma D 2 (dp , ic)$$
5: IC ← IC ∪ {dp}
6: End while
7: K-Means on X using the set of initial centroids IC

Initialise the single centroid by picking random sample from the dataset X and calculate the distance from the remaining points and find the data point with the highest probability and add it to the initial centroid set. Repeat the steps 2 to 5

until k centers are occurred. Perform the K-Means algorithm for the obtained centers at step7 and obtain the final clusters for the dataset.

Performing K-Means++ algorithm on large scale data is time consuming due to its serial nature. To solve this problem we will partition the data into m subsets and perform the CK-Means algorithm on each subset in parallel. The results of each subset are again scored using a fitness function to get the k intial centroids.

Algorithm 2:

1: Partition X into x 1 , x 2 , ..., x m
2: For each i ∈ {1, 2, ..., m} do
3: Run K-Means++ on x i to get k centroids IC i and clusters clx i
4: S i = f (clx i)
5: C ← IC i , where i ← Best – fit(S i)
6: Run K-Means on X with C as initial centroids to obtain cluster analysis output

In step 3 we perform the K-Means++ algorithm parallelly on the subsets of data and get the k centers for each subset. Among these centers again we obtain the k centers using the fitness function. Perform the K-Means algorithm for the obtained centers in step 6 and retrieve the effecient clusters. Here steps 3-5 clearly shows the parallelization of the computation and reduces the time of execution.

Dataset used in the experiment is the hypercube dataset which is a generated by using synthetic data generator. The dataset having 10k points with 10 dimensionality.

The equipment used for performing the cluster analysis is a Hadoop cluster with 5 Nodes each of 8GB, 1334MHz DDR3 RAM; 1TB Hard Disk with inter core i5 processor;Ubuntu 14.04 and Hadoop version 2.7.0

We can apply the kmeans++ algorithm and CK-Means algorithm on the dataset and the corresponding execution time is noted in the table by varying the k, where k is the number of centers.

Table1: execution time in milli seconds

Algorithm	K=50	K=200
Kmeans++	4281ms	4765ms
CK-Means	856ms	953ms

Hence from the above results it is clearly shown that our algorithm produces the accurate clusters in very less time.

4. Conclusion

K-Means algorithm with parallel processing will produce the fast clusters but it suffers with the initial cluster center initialization. Although K-Means++ improves the initial seeding yet it shows the serial nature. So in this paper we parallelize the K-Means++ to improve the fast and effective clusters over the large datasets. Our CK-Means will produce the accurate clusters by taking the advantage of Map Reduce and K-Means++ algorithms. We found that our CK-Means algorithm scales well with large scale and with high dimensional data. Observing our results we will clearly prove that accurate clusters are formed very fast by using our CK-Means algorithm.

References

[1] Esteves, R.M., Hacker, T. ; Chunming Rong, (2013) Competitive K-Means, a New Accurate and Distributed K-Means Algorithm for Large Datasets
[2] Chun-Wei Tsai, Chin-Feng Lai, Han Chieh Chao,(2015), Big data analytics: a survey, journal of BigData
[3] Arthur, D. and Vassilvitskii, S. (2007) K-Means++: the advantages of careful seeding', SODA '07 Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms
[4] Ekanayake, J., Pallickara, S. and Fox, G. (2008) 'MapReduce for data intensive scientific analyses', in eScience '08, IEEE Fourth International Conference on, pp.277–284.
[5] Holmes, A. (2012) Hadoop in Practice, Manning Publications Co., New York.
[6] Tapas Kanungo, David M. Mount, Nathan S, (2002), An Efficient k-Means Clustering Algorithm: Analysis and Implementation.
[7] Dean, J. and Ghemawat, S. (2008) 'MapReduce: simplified data processing on large clusters', Commun. ACM, Vol. 51, No. 1, pp.107–113, doi:10.1145/1327452.1327492.
[8] Esteves, R.M.,Pais, R. ; Chunming Rong, (2011), K-means Clustering in the Cloud - A Mahout Test a conference of Advanced Information Networking and Applications (WAINA)