ISSN No. 0976-5697

Volume 8, No. 5, May-June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Comparative Analysis of K-Means using MapReduce

Humam Siddiqui
Department of Computer Science and Engineering
Jamia Hamdard
Delhi, India

Safdar Tanweer
Department of Computer Science and Engineering
Jamia Hamdard
Delhi, India

Abstract: Continuous growth of digital data has led to concentration in the data mining technique .The actual purpose of data mining is to analyze the larger data set to extract knowledge and interesting patterns. The cluster analysis is an important data partitioning process which distribute data items into different groups (clusters), so that the data items in each cluster can share the common characteristics. Data collected in real time scenarios are more often semi structured and unstructured, that needs to be processed to extract hidden knowledge from it. Here the clustering technique comes into the scenes, there are various clustering algorithms, k Mean is the simplest and popular unsupervised learning algorithm, which has solved many well know clustering problem. K Mean clustering algorithm produces a specific member of disjoint cluster, starting from randomly selected cluster centers. In this paper we have implemented the k mean clustering algorithm for different distance metrics in the MapReduce programming model running in Hadoop distributed environment.

Keywords: K-Means; Data Mining; Big Data; MapReduce; Hadoop.

I. INTRODUCTION

In general, Data can be considered as any set of characters that has been collected and translated for some purpose, mostly analysis. It can be set of characters, including numbers, text, video, pictures, or sound. When this data is processed in such a way so that it is meaningful who receives it then it is called Information. Day by day, the amount of raw data is increasing rapidly. These data are exploding the size of databases also.

So, Data Mining is the process of extracting hidden predictive knowledge or information from large databases. It is also called KDD, i.e., Knowledge Discovery from Database. Data Mining is a powerful tool having great potential which helps companies to focus on relevant data or information from the database [1]. Generally, data mining task is classified as-

- Classification: **Classification** is one of the **data mining** function that is used to assign items in a collection to target classes. The purpose of **classification** is to correctly predict the target class for each **data**.
- **Clustering:** Creating a group of set of objects in such a way so that all objects or data points in the one group (called a **cluster**) have mostly similar feature to each other than to those which are in other groups (**clusters**), is called Clustering.
- **Association rule mining:** In data mining, the probability of the co-occurrence of items in a collection, is determined by association function. **Association rules express t**he relationships between co-occurring items [2][3].

Now these days, we create quintillion bytes of data per in the world. This huge amount of the data is considered as "**Big data"**. So, Big data is a catch-word, which is used to describe a massive volume of both structured data and unstructured data that is so large that it's difficult to process using traditional database and software techniques. Big Data Analytics is essential for the processing of the massive and complex datasets. This massive data is different from structured data in terms of four parameters –volume, velocity, variety and veracity (4V's). The four V's (volume, velocity, variety, veracity) are the challenges of big data management are:

- **Volume:** There are many factors which contribute to the increase in data volume. This was solved due to decreasing the cost of storage, but then other issues emerge, including how to use analytics to create value from relevant and how to determine relevance within large data volumes.
- **Velocity:** Today, data is streaming in at tremendous speed and must be dealt with in a timely manner. Reacting quickly enough to handle with data velocity is one of the challenges for most organizations.
- **Variety:** Now a days, data comes in all types of formats. Structured, Unstructured, numeric data in traditional databases. Managing, and governing different varieties of data is also a challenging task for organizations.
- **Veracity:** Veracity indicates to the trustworthiness of the data. . When organizations are dealing with huge volume, velocity and variety of data, the all of data are not going to be 100% correct, there will be some dirty data. So, big data and analytics technologies work with these kinds of data [4][5].

So, big data is complex and ambiguous one. For achieving goals and objectives of big data Hadoop is used. This Hadoop system include of the Hadoop kernel, MapReduce, distributed file system and data processing and analysis tools. In this research paper, we have represented K-means in Map Reduce based on different distance metric, i.e., Euclidean and Manhattan distance.

II. METHODS AND MATERIALS

For this research work, data set is taken from the UCI Machine Learning Repository which has a huge collection of number of datasets. It is used by the researchers of machine learning also. This dataset is generated through a study of adopted the donor database of Blood Transfusion

Service Center in Hsin-Chu City in Taiwan. The center passes their blood transfusion service bus to one university in Hsin-Chu City to gather blood donated about every three months. This study is done for demonstrating the RFMTC marketing model (a modified version of RFM).To build a FRMTC model, they selected 748 donors at random from the donor database. These 748 donor data, each one included R (Recency - months since last donation), F (Frequency - total number of donation), M (Monetary - total blood donated in c.c.), T (Time - months since first donation), and a binary variable representing whether he/she donated blood in March 2007[6]. In this research work, K-means algorithm is applied with MapReduce for clustering. The work has been carried out on JAVA. The techniques which are used in this paper are-

A. K-means

K-means is one of the simplest, well-known unsupervised clustering algorithms. It assigns data points having similar features in one cluster and other data points in another cluster. K-means firstly takes random data point as centroid of the cluster and then iteratively assign the rest points in that cluster based on their distance from that centroid. K-means is good to use where there is too many data points which are to be clustered. K-means has one major drawback that in this number of the clusters should be known in advance. It is sensitive to noise also [7][8][9][10]. K-means uses the following steps-

1. Define number of clusters that is K.
2. Select k number of data points from given data set, taken as a centroids of the cluster.
3. Calculate the distance between each data point to each centroid and assign data point to the cluster which centroid is nearest to that data point.
4. When all the data points are assigned to some cluster, then recalculate the k-centroids.
5. Repeat the steps 3, 4 until the centroids no longer change or until no point changes its cluster assignment [11][12][13].

In this research paper, we are using two different distance metric. These are-

- Euclidean Distance: This is considered as a straight line distance between two points. If there is are two points x and y in Euclidean n-space such that $x=(x_1,x_2,\ldots,x_n)$, $y=(y_1,y_2,\ldots,y_n)$ then the distance, D(x, y), is calculated[14][15][16]-

$$D(x,y)=D(y,x)=\sqrt{(y_1-x_1)^2+(y_2-x_2)^2+\ldots+(y_n-x_n)^2}$$

$$=\sqrt{\sum_{i=1}^{n}(y_i-x_i)^2}$$

- Manhattan Distance: This distance simply gives the average difference across dimensions. In a plane, if there is are two points p and q such that $p=(p_1,p_2,\ldots,p_n)$, $q=(q_1,q_2,\ldots,q_n)$, then Manhattan distance is calculated as[17][18][19]-

$$D(p,q)=D(q,p)=\sum_{i=1}^{n}|Pi-Qi|$$

B. Hadoop

The Hadoop is a library framework based on Java programming developed mainly to deal with Big Data. It distributes the huge data sets to across clusters of commodity computers and hardware for better processing, using simple programming models. It is designed to execute from single servers to hundreds of machines, each offering local storage and computation. Rather than rely on hardware to deliver high-availability, the library designed work on a distributed environment to detect and handle failures at the application layer, delivers a highly-available service on top of a cluster of commodity hardware, each of which may be prone to failures. It inhabits the ability to process and store huge volume of any kind of data quickly, workings on the distributed environment, achieve the work completion faster as distributing the work and executing them parallel. The more computing nodes you use the more processing power you have, increases direct proportion manners. Data and application processing are protected against hardware failure. If a node fails, tasks are automatically shifted to some other nodes in order to make sure the distributed computing does not fail. Copies of data are maintained at multiple places in order to reduce the effect of data failures [20].

C. MapReduce

MapReduce is a model, programmed for processing data. MapReduce one of the core components of the Hadoop framework. MapReduce programs run by Hadoop written in various languages, i.e., java, ruby, python and c++. It is a model programmed for processing the large data sets. It contains task of data processing and distributes the tasks across number of nodes. It works in a fashion in which it divides the larger problem in a smaller number of tasks and distributes them into number of interconnected nodes working together in a coordinated way. The larger the number of nodes the faster the processing will be.

The Map function transforms a typical dataset into another data set where individual elements are divided into key/value pairs. Map function that uses a (key, value) pair for computation. The Map function results is an intermediate result which works as an input to the reducer function.

The **Reduce** function task is to take the output from the map, consider it as an input for further processing. Reduce process the output from the map and then integrate the data tuples into smaller set of tuples and generates an optimal result .The reduce methods then accumulate the various result and combines them to answer the larger problem. Always it is been executed after the map job is done [21][22].

D. K-Means Clustering Using Mapreduce

The initial stage in implementing the K-mean algorithm in MapReduce is to handle the input and output of the application .The MapReduce programming model works on the key /value pair. Take the key/value pair as an input to the MapReduce job, the input to K-means algorithm must be in key/value pair. Here the key are the cluster centers and the values are the vector of datasets. To implement the MapReduce model with K-mean, two files are created, one storing the cluster centers as the key and the other storing the vectors form of dataset to be clustered. In this research, the basic K-means algorithms approach in the map reduce

model is followed. Initially, both the two files are loaded in HDFS form from the local filesystem to pass as the input to the map function .The map function takes the two files as an input and takes the key as cluster center from one file. Then map function calculates the distance, using the suitable distance calculation, for each vector form of data set values, simultaneously recording the cluster to which the dataset value is nearest .Once all the vectors has been processed ,the dataset values are assigned to the nearest cluster. After the mapper function finishes it task, the centroid or centers of each cluster are updated in the reducer function until it reaches a convergent point. The newly created clusters are written back to the disk which will be loaded as an input to the next iteration [23][24].

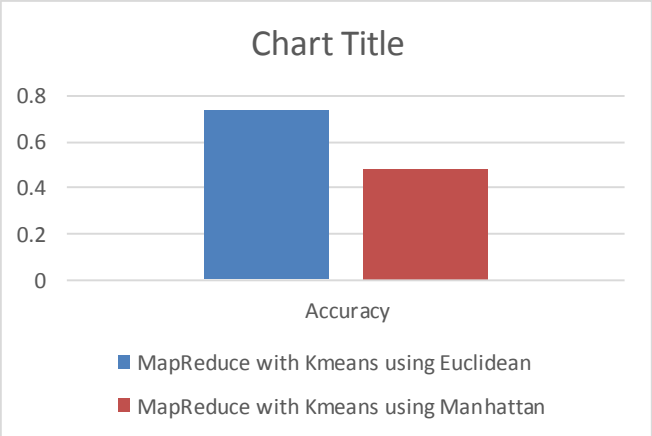
III. EXPERIMENTAL RESULT

In this research, according to the user's requirement, number of clusters can be defined. The well-known, UCI Machine Learning Repository is used for the purpose of testing the efficiency of MapReduce with K-Means using Euclidean and MapReduce with K-Means using Manhattan on JAVA. In this research, Blood Transfusion Service Center Data Set is used which have five attributes and those all five are real .The total numbers of instances are 748. There are two classes, binary variable representing whether he/she donated blood in March 2007- class0 (stands for not donating blood), class1 (stands for donating blood). So, it generates two clusters: cluster0, cluster1. The cluster quality is evaluated using the accuracy of clusters and the elapsed time taken by these two algorithms, i.e., MapReduce with K-means using Euclidean and MapReduce with K-means using Manhattan. Here, to generate these two clusters from data set, first we parse the whole dataset and select randomly the k (2 data points) as initial centroids and save in a separate file called "clusters" and the others are stored as data vector in another file. These two files are fed to the MapReduce implementation of K-means clustering algorithm which is discussed in the previous section. In this implementation the numbers of iterations are restricted to maximum 15 iterations. Hence, it generates 2 cluster files after the processing of vectors and initial clusters. Authors and Affiliations.

A. *Comparison on Accuracy*

The following table shows values of accuracy of different algorithms. The following table has shown clearly that-

- MapReduce with K-means using Euclidean gives higher accuracy than MapReduce with K-means using Manhattan.



1) Fig 1. Comparison of accuracy of Mapreduce with Kmeans using Euclidean and MapReduce with Kmeans using Manhattan.

B. *Comparison on Elapsed Time-*

The following table clearly shows the elapsed time in seconds taken by these two algorithms. It states that-

- MapReduce with K-means using Euclidean takes less elapsed than MapReduce with K-means using Manhattan.

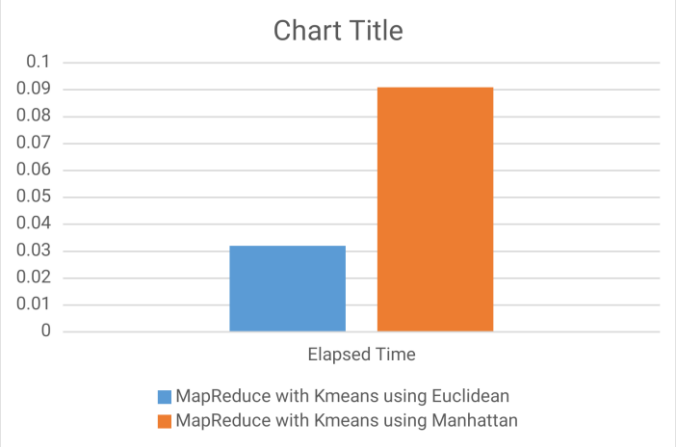


Fig 2. Comparison of elapsed time of MapReduce with Kmeans using Euclidean and MapReduce with Kmeans using Manhattan.

IV. CONCLUSION

In this current time, the data size is growing very rapidly from various sources. It is very necessary to process these huge volume of data to extract useful information hidden in it. Clustering is one such research technique which extract useful information from huge data. Among different clustering algorithm, K-Means algorithm is the simplest and popular algorithm which is widely used for large dataset. The only drawback of this algorithm is that it is needed to define number of clusters in the initial. Hadoop is an open source product introduced by apache which provide data processing on distributive environment.

For mining the information, selecting a proper algorithm is an essential and tough task. While selecting any algorithm one will have to consider other parameters also which can affect the processing of algorithms. So, this paper discussed, the implementation of K-Means Clustering Algorithm over a

distributed environment in MapReduce programing model using different distance metric and also shown that while implementing K-means with MapReduce one can get more accurate clusters using Euclidean. So, K-means using Euclidean with MapReduce will give better performance than K-means using Manhattan in MapReduce.

.

V. REFERENCES

[1] HAN J. and Kamber M., Data Mining: Concepts and Techniques, Morgan Kaufimann Publishers,2001.

[2] H. Gulati, "Predictive analytics using data mining technique," 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2015, pp. 713-716.

[3] A. K. Jain, M. N. Murty and P. J. Flynn, "Data Clustering: A review", ACM Computing Surveys, vol. 31, no. 3, 1999.

[4] Apache Hive. Available at http://hive.apache.org

[5] 23 C. Lakshmi , V. V. Nagendra Kumar," Survey Paper on Big Data" ,International Journal of Advanced Research in Computer Science and Software Engineering 6(8), August- 2016, pp. 368-381

[6] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[7] A. Rakhlin and A. Caponnetto, "Stability of K-Means clustering", Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2007, pp. 216–222.

[8] Li C. and Biswas G., "Unsupervised learning with Mixed Numeric and Nominal Data", IEE Transactions on Knowledge and Engineering, vol. 14,no 4,pp. 673-690,2002.

[9] Longjiang Guo, Chunyu Ai, Xiaoming Wang, Zhipeng Cai and Yingshu Li, "Real time clustering of sensory data in wireless sensor networks," 2009 IEEE 28th International Performance Computing and Communications Conference, Scottsdale, AZ, 2009, pp. 33-40. doi: 10.1109/PCCC.2009.5403841

[10] M. Danishvar, A. Mousavi, P. Sousa and R. Araujo, "Event-clustering for real-time data modeling," 2013 IEEE International Conference on Automation Science and Engineering (CASE), Madison, WI, 2013, pp. 362-367. doi: 10.1109/CoASE.2013.6653911

[11] N. Afshan, S. Qureshi and S. M. Hussain, "Comparative study of tumor detection algorithms," 2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom), Greater Noida, 2014, pp. 251-256.

[12] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 881-892, Jul 2002.doi: 10.1109/TPAMI.2002.1017616

[13] B. Thuraisingham, L. Khan, C. Clifton, J. Maurer and M. Ceruti, "Dependable real-time data mining," Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'05), 2005, pp. 158-165.

[14] Grabusts, P. "The choice of metrics for clustering algorithms." In Proceedings of the 8th International Scientific and Practical Conference (Vol. 2). ISSN 1691-5402 ISBN 978-9984-44-071-2 Pp. 70 -76, 2011.

[15] J. Ye, Z. Zhao and H. Liu, "Adaptive Distance Metric Learning for Clustering," 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, 2007, pp. 1-7. doi: 10.1109/CVPR.2007.383103

[16] R. Bansal, N. Gaur and S. N. Singh, "Outlier Detection: Applications and techniques in Data Mining," 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), Noida, 2016, pp. 373-377. doi: 10.1109/CONFLUENCE.2016.7508146

[17] Riabov A., Liu Z., Wolf L., Yu S. and Zhang L "Clustering Algorithms for Content-Based Publication-Subscription Systems", in Proceedings of the 22nd International Conference on Distributed Computing Systems(ICDCS'02), USA,pp.133,2002.

[18] S. Okada and T. Nishida, "Online incremental clustering with distance metric learning for high dimensional data," The 2011 International Joint Conference on Neural Networks, San Jose, CA, 2011,pp.2047-2054. doi: 10.1109/IJCNN.2011.6033478

[19] --Shraddha Pandit, Suchita Gupta, "A Comparative Study On Distance Measuring approaches for Clustering", International Journal of Research in Computer Science eISSN 2249-8265 Volume 2 Issue 1 (2011) pp. 29-31 © White Globe Publications.

[20] Apache Pig. Available at http://pig.apache.org

[21] Parmeshwari P. Sabnis, Chaitali A.Laulkar , "SURVEY OF MAPREDUCE OPTIMIZATION METHODS", ISSN (Print): 2319-2526, Volume -3, Issue -1, 2014

[22] Parmeshwari P. Sabnis, Chaitali A.Laulkar , "SURVEY OF MAPREDUCE OPTIMIZATION METHODS", ISSN (Print): 2319-2526, Volume -3, Issue -1, 2014.

[23] N. Akthar, M. V. Ahamad and S. Ahmad, "MapReduce Model of Improved K-Means Clustering Algorithm Using Hadoop MapReduce," 2016 Second International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, 2016, pp. 192-198. doi: 10.1109/CICT.2016.46

[24] P. P. Anchalia, "Improved MapReduce k-Means Clustering Algorithm with Combiner," 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, Cambridge, 2014, pp. 386-391. doi: 10.1109/UKSim.2014.11