# Fuzzy K-mean Clustering in MapReduce on Cloud Based Hadoop

Dweepna Garg[1] ,Khushboo Trivedi[2]
[1]Department of Computer Science and Engineering, Parul Institute of Engineering and Technology, Limda,
[2]Department of Information Technology, Parul Institute of Engineering and Technology, Limda
[1]dweeps1989@gmail.com, [2]khushi.oza@gmail.com

\

***Abstract*— Clustering is regarded as one of the significant task in data mining which deals with primarily grouping of similar data. To cluster large data is a point of concern. Hadoop is a software framework which deals with distributed processing of huge amount of data across clusters of commodity computers using MapReduce programming model. MapReduce allows a kind of parallelization for solving a problem involving large data sets using computing clusters and is also an attractive mean for data clustering involving large datasets. Mahout, a scalable machine learning library is an approach to Fuzzy K-mean clustering which runs on a Hadoop. This paper focuses on studying the performance of different datasets using Fuzzy K-mean clustering in MapReduce on Hadoop. Experimental results depict the execution time of the approach on a multi-node Hadoop cluster which is build using Amazon Elastic Cloud Computing(Amazon EC2).**

***Keywords—Fuzzy K-mean clustering; MapReduce; Hadoop; HDFS; Mahout.***

## I. INTRODUCTION

K-mean algorithm faces a problem of giving a hard partitioning of the data which means that each point is dedicated to one and only one cluster. The data points on the edge of the cluster as well as lying near another cluster may not be as much in the cluster as points in the center of cluster. Hence, Fuzzy K-mean clustering[1] (also known as Fuzzy C-means clustering) given by Bezdek introduced that each point has a probability of belonging to a certain cluster. A coefficient value associated with every point gives the degree of being in the $k^{th}$ cluster and coefficient values should sum to one. Nowadays larger datasets are considered for clustering which do not even fit into main memory.

Apache Hadoop[2,4] was born to solve the problems pertaining to large datasets. With the help of MapReduce, Hadoop fires a query on the large datasets, divide it and then runs it in parallel on multiple nodes. Mahout[3] is a scalable machine learning library which is built on Hadoop and is usually written in Java. In this work, the performance of Multinode Hadoop cluster in Amazon EC2 using Fuzzy K-mean clustering in MapReduce on Hadoop is carried out to compare the results of execution time using three typical datasets.

The rest of the paper deals with the following sections: section II covers about Hadoop, MapReduce and Mahout. Section III covers the Fuzzy K-mean clustering algorithm using both the iterative and MapReduce approach. Section IV shows the experimental setup. Section V covers results. Section VI is conclusion. Future work is described in section VII.

## II. HADOOP , MAPREDUCE AND MAHOUT

Hadoop, an open source framework implementing the MapReduce programming model includes two components namely the Hadoop Distributed File System (HDFS)[4] and MapReduce. HDFS is used for storage of large dataset and MapReduce is used for processing the datasets. In HDFS, the file is split in contiguous chunks each of size 64MB (default block size)[4] and each of these chunk is replicated in different racks. The NameNode in HDFS stores the metadata and the DataNodes stores the blocks from files. Associated with the NameNode and the DataNode is the daemon known as the JobTracker and the TaskTracker respectively. It is the duty of the JobTracker to assign the jobs to the TaskTracker which then processes each of the jobs assigned to it using the MapReduce model. Hadoop, a distributed file system is written in Java.

There are mainly two programs in MapReduce[6], one is the Map and another is Reduce. Dataset is split according to block size of Hadoop. Map( ) function is associated with each block and considers the input pair in the form of a key and value and then processes the input pair thereby generating an intermediate set of <key, value> pairs. The function of Reduce( ) aggregates the intermediate results and generates the final output.

Apache Mahout [3] is a scalable machine learning library which involves clustering, classification and collaborative filtering implemented on top of Apache Hadoop using the MapReduce paradigm and is also written in Java. There are various subcategories of machine learning such as unsupervised learning, supervised learning, semi-supervised learning, learning to learn and reinforcement learning.

ISBN No. 978-1-4799-3914-5/14/$31.00 ©2014 IEEE

Clustering, an unsupervised learning technique groups the samples having similarity in different classes. The groups or classes are referred to as clusters. Samples within a class are of high similarity as compared to the samples of other classes. It is learning by observation process which is mainly used in the areas like data mining, machine learning and statistics. Fuzzy K-mean clustering is based on centroid based clustering technique.

## III. Fuzzy K-Mean Clustering

The fuzzy K-mean clustering, also known as soft clustering is an extension of K-mean clustering. It minimizes the intra-cluster variance. Bezdek introduced the concept of fuzziness parameter (m) in Fuzzy K-mean clustering which determines the degree of fuzziness in the clusters[1]. The algorithm of standard Fuzzy K-mean clustering algorithm is as follows:

1. Choose a number of clusters.
2. Create distance matrix from a point $x_j$ to each of the cluster centers considering the Euclidean distance between the point and the cluster center using the formula:

$$d_{ij} = \sqrt{\sum (x_j - c_i)^2} \qquad (1)$$

where,

$d_{ij}$= Euclidian distance between the $j^{th}$ data point and the $i^{th}$ cluster center

3. The membership matrix is created using:

$$\mu_i(x_j) = \frac{\left(\frac{1}{d_{ij}}\right)^{\frac{1}{m-1}}}{\sum_{k=1}^{p}\left(\frac{1}{d_{kj}}\right)^{\frac{1}{m-1}}} \qquad (2)$$

where,

$\mu_i(x_j)$ = is the membership of $x_j$ in the $i^{th}$ cluster

m= fuzziness parameter

p= number of specified clusters

$d_{kj}$= distance of $x_j$ in cluster $C_k$

For a point in a sample, the total membership must sum to 1. The value of m is kept generally greater than 1 because if it is kept equal to 1, then it resembles K-mean clustering algorithm.

4. The new centroid for each cluster is generated as:

$$C_i = \frac{\sum_i \left[\mu_i(x_j)\right]^m x_j}{\sum_i \left[\mu_i(x_j)\right]^m} \qquad (3)$$

*Stopping criteria*: - The algorithm continues until any centers of the clusters do not change beyond the convergence threshold and neither the points change in the assigned cluster.

The limitation of this iterative algorithm is that number of iterations is increased for forming overlapping clusters thereby increasing the execution time and if large dataset is used then it becomes difficult to handle in main memory. Hence to overcome this problem, MapReduce approach is used.

*MapReduce Approach*

MapReduce approach partitions the large datasets and then computes on the partitioned dataset (known as jobs) in a parallel manner where the individual jobs are processed by the maps and then the sorted output from the maps are processed by the reduce.

*Input*: Data points, randomly selected centroid points, number of clusters.

*Output:* Final centroids and their clustered points.

*Algorithm of Map*:

1. The randomly selected centroid point is considered as key and vector points as value.
2. Calculate the Euclidian distance between centroid point and the vector point using (1).
3. Compute the membership value of each vector point and create the membership matrix using (2).
4. Clusters are generated using the nearest centroid and the data points assigned to that particular cluster.
5. Maintains a cache holding the detail about which vector point is in which cluster.

*Algorithm of Reduce*:

1. Recalculates the centroid for each cluster.

The recalculated centroid would go serially to Map and after that as it iterates, the work would be done in parallel until the centroid converged as depicted in figure 1. Total no. of reducers are less than total no. of mappers(M< N).

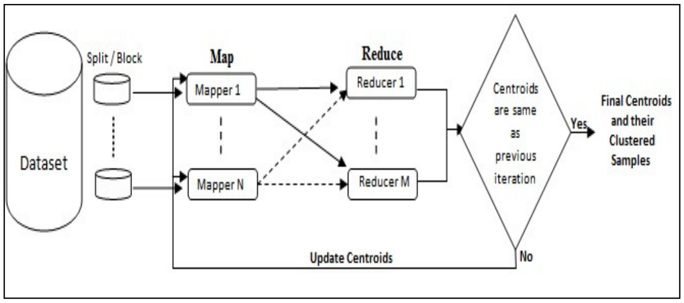

Fig.1: Fuzzy K-means Clustering Algorithm in MapReduce

## IV. Experimental Setup

The experiments were performed on Amazon web service cloud using Elastic Cloud Computing (EC2). Amazon Web Services (abbreviated AWS) is a collection of remote computing services (also called web services) that together make up a cloud computing platform, offered over the Internet by Amazon.com[8]. "m1.medium" and "m1.small" are used as NameNode and DataNode respectivly."m1.medium" has Intel(R) Xeon(R) CPU E5-2650 @ 2.00GHZ processor with 3.7GB RAM, 410GB Hard-disk."m1.small" is based on Intel(R) Xeon(R) CPU E5-2650 @ 2.00GHZ processor, 1.7GB RAM, 160GB hard disk. Hadoop 1.2.1 and 64 bit Java OpenJdk 1.7.0_25 are installed in both of the instances. Installed Operating System is 64 bit Ubuntu server 12.04.3LTS.

For the purpose of testing the performance of Fuzzy K-mean clustering on Hadoop cluster using MapReduce, datasets from well known UCI Machine Learning Repository are used namely Iris dataset[1] and Synthetic control dataset[2]. Iris dataset, of size 8KB has five attributes. Out of which four attributes (Sepal length, sepal width, petal length, petal width) are numeric and fifth attribute has three classes (Iris Setosa, Iris Versicolour, and Iris Virginica) are there for this one non-numeric attribute. It is created by R.A. Fisher. There are total 150 samples in this dataset. Synthetic control dataset of size 284 KB consists 600 samples of synthetically generated control charts. It is contributed by Dr Robert Alcock. The six different classes of control charts are Normal, cyclic, increasing trend, decreasing trend, upward shift and downward shift with each one of the range of 100. KDD cup 1999[3] dataset (10% of full) of size 75 MB is classified as labeled and unlabeled records. There are 41 attributes in each labeled record along with one target value indicating the category name of the attack. The dataset contains a total of 24 training attack types. The labeled dataset as in all around 5 million (4,94,022) records which are used for training the model. Size of full KDD cup 99 dataset is 778MB and it has 48,98,430 records.

## V. Results

To input the dataset for Mahout, preprocessing of the dataset is to be done for converting the data points into the vector format. The vectors are then converted in SequenceFile format (Hadoop file format). The tools for creating the vectors and file conversion are provided by Mahout if the datasets are in .arff format. The Fuzzy K-mean clustering takes in the dataset as the input along with the desired number of clusters, the fuzziness parameter m with the value 1.2 and the distance metric as the Euclidian distance for calculating the distance from centroid point to the vector point. For carrying out the experiment using Iris dataset, 20 maximum iterations and 3 clusters , for synthetic control dataset, 20 maximum iterations and 6 clusters and for KDD cup 1999(10%) ,80 maximum iterations and clusters 2 are considered. For full KDD cup 1999 dataset, we have taken 200 maximum iterations and 24 clusters .We implement Java code for preprocessing of KDD cup 1999 dataset which converts CSV file format to sequence file format. Text attributes are converted in numeric and all samples are in vector format in the sequence file.

Fuzzy K-mean clustering algorithm using MapReduce is tested on iris dataset, synthetic control dataset, KDD cup 1999 dataset (10% of full dataset) and full KDD cup 1999 the results of the same is depicted in the Fig.2.

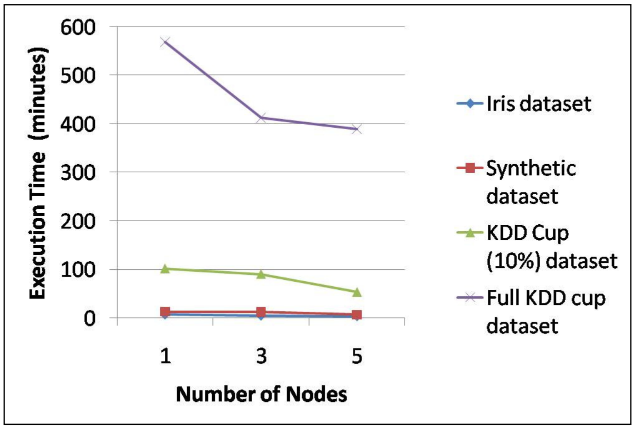

Fig.2: Comparison of Execution time with Number of nodes for different datasets

It is observed in the figure that for small datasets, there is minimum deviation in execution time even if the number of nodes is increased. But for large dataset, where the file size is greater than the block size, a greater deviation is found in execution time on increasing the number of nodes.

On executing the Fuzzy K-mean clustering algorithm in MapReduce of Mahout on different nodes, accuracy remains same and its corresponding confusion matrix is depicted in Table 1 and Table 2. Confusion matrix is made for only iris dataset and KDD cup 1999(10%) dataset because it is easily identified using the class labels. For full KDD cup 1999 dataset, Confusion Matrix is evaluated for 24 types of attacks which are class label. Accuracy of Iris dataset, KDD cup 1999(10%) dataset and full KDD cup dataset are 90%,78.53% and 63.41% respectively. Whereas in synthetic control dataset, inter-cluster quality measure is considered as there is no class label in it. Each test was carried out 5 times to ensure the results.

TABLE 1: CONFUSION MATRIX FOR IRIS DATASET

| | Iris Setosa | Iris Versicolor | Iris Virginica |
|---|---|---|---|
| Iris Setosa<br>(Cluster 1) | 50 | 0 | 0 |
| Iris Versicolor<br>(Cluster 2) | 0 | 50 | 15 |
| Iris Virginica<br>(Cluster 3) | 0 | 0 | 35 |

TABLE 2: CONFUSION MATRIX FOR KDD CUP 1999 (10%) DATASET

| | "Good" normal connections | "Bad" connections (intrusions/attacks) |
|---|---|---|
| Normal<br>(Cluster 1) | 96694 | 13990 |
| Anomaly<br>(Cluster 2) | 92040 | 291297 |

[1] http://archive.ics.uci.edu/ml/datasets/Iris

[2] http://archive.ics.uci.edu/ml/datasets/Synthetic+Control+Chart+Time+Series

[3] https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

## VI. CONCLUSION

We have conducted the experiments using varying size of the dataset and different number of nodes. Mahout, a tool for Fuzzy K-mean clustering allows clustering of large dataset and is scalable thereby producing a significant increase in performance. It is observed that as the number of nodes increases, execution time is reduced. Also it is found that if small dataset i.e iris dataset and synthetic control dataset is considered then we do not get significant gain in performance in terms of execution time. The accuracy of Fuzzy K-mean clustering algorithm remains nearly same even if the number of nodes is increased.

## VII. FUTURE WORK

Optimize the Fuzzy K-mean clustering algorithm by considering the centroid point using certain methodology instead of choosing it randomly which could result in a decrease in number of iterations thereby decreasing the execution time.

## ACKNOWLEDGEMENT

We take this opportunity to express our profound gratitude and deep regards to Professor G.B. Jethava (Head of Information Technology Department), Assistant Professor Yask Patel and Assistant Professor Rahul Joshi of Parul Institute of Engineering and Technology, Limda for their encouragement, guidance, and insightful comments throughout the course of this paper. The blessing, help and pain-staking effort given by them time to time is greatly appreciated.

## REFERENCES

[1] Bezdek, James C, "FCM : THE FUZZY c-MEANS CLUSTERING ALGORITHM", vol. 10, pp191-203, 1984

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

[3] https://mahout.apache.org/

[4] Shvachko, K.; Hairong Kuang; Radia, S.; Chansler, R., "The Hadoop Distributed File System," Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on , vol., no., pp.1,10, 3-7 May 2010

[5] T. White, Hadoop: The Definitive Guide, O'Reilly Media, Yahoo! Press, June 5, 2009.

[6] Ghemawat, H. Gobioff, S. Leung. "The Google file system," In Proc.of ACM Symposium on Operating Systems Principles, Lake George, NY , pp 29–43, Oct 2003

[7] Changqing Ji; Yu Li; Wenming Qiu; Awada, U.; Keqiu Li, "Big Data Processing in Cloud Computing Environments," Pervasive Systems, Algorithms and Networks (ISPAN), 2012 12th International Symposium on , vol., no., pp.17,23, 13-15 Dec. 2012

[8] Amazon EC2: http://aws.amazon.com/ec2/

[9] W. Lee, S. J. Stolfo, and K. W. Mok, "Mining in a Data-Flow Environment: Experience in Network Intrusion Detection", In Proceedings of the 5th ACM SIGKDD, San Diego, CA, 1999b, pp. 114-12

[10] Esteves, R.M.; Pais, R.; Chunming Rong, "K-means Clustering in the Cloud -- A Mahout Test," Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on , vol., no., pp.514,519, 22-25 March 2011

[11] Kwok, Terence,Smith Kate,Lozano, Sebastian,Taniar, David, "Parallel Fuzzy c-Means Clustering for Large Data Sets", pp. 365-374,2002

[12] Zhanquan, Sun Fox, Geoffrey,A Parallel Clustering Method Study Based on MapReduce, July 2012

[13] Kejiang Ye, Xiaohong Jiang, Yanzhang He, Xiang Li, Haiming Yan, Peng Huang, "vHadoop: A Scalable Hadoop Virtual Cluster Platform for MapReduce-Based Parallel Machine Learning with Performance Consideration," Cluster Computing Workshops (CLUSTER WORKSHOPS), 2012 IEEE International Conference on , vol., no., pp.152,160, 24-28 Sept. 2012 doi: 10.1109/ClusterW.2012.32

[14] Anchalia, P.P.; Koundinya, A.K.; Srinath, N.K., "MapReduce Design of K-Means Clustering Algorithm," Information Science and Applications (ICISA), 2013 International Conference on , vol., no., pp.1,5, 24-26 June 2013

[15] Zhou, Ping Lei, JingshengYe, Wenjun,Large-Scale Data Sets Clustering Based on MapReduce and Hadoop, vol 26 pp. 5956-5963,2011

[16] Alina Ene,Sungjin Im, Benjamin Moseley, "Fast clustering using MapReduce", Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, 978-1-4503-0813-7, San Diego, California, USA, pp. 681-689, 2011

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

[18] Description of Multi Node Cluster Setup at: http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-multi-node-cluster/ visited on 21st January, 2012

[19] Sean Owen, R.A., Ted Dunning and Ellen Friedman: 'Mahout in Action' (Manning Publications, 2010. 2010)