A NOVEL PREDICTION APPROACH TO ANALYZE BIG DATA USING K-NEAREST NEIGHBOR ALGORITHM

R.Vinoth [1], Dr.K.Nattar Kannan [2], Dr.KL.Shunmuganathan[3]

[1]Assistant Professor, [2]Associate Professor, Professor[3]
[1] Department of Information Technology, Agni College of Technology, Chennai.
[2]Department of Computer Science and Engineering, Dhanalakshmi College of Engineering, Chennai.
[3]Department of Computer Science and Engineering, Dhanalakshmi College of Engineering, Chennai.
Email: kannannattar@gmail.com

Abstract

Agriculture is the important sources of survival and one of the most important factors in the economic growth of the country. In order to perform analysis on agriculture field that leads to many issues like proper information about current status of soil moisture, climate humidity and temperature. Some devices are developed for improve agriculture production, but it is not successful and sufficient. In this paper, the proposed system process the agriculture data(Big Data) in Hadoop platform to predict the crop yield and to suggest the crop growth thereby improve the quality of yield. In this work, a novel prediction approach using K-nearest neighbor (NPKNN) was proposed to handle and process the large volume of agriculture data set in parallel in Map-Reduce framework. The proposed system has implemented only three nodes. It can be implemented to more number of nodes. A master is setup with two slave nodes in Hadoop distributed environment. The input agriculture test and train data set are in data nodes (slave). The master implement NPKNN algorithm in Map-Reduce frame work to read the data set and analyze it. The output file for each data nodes is written back to Hadoop Distributed File System (HDFS).

Keywords: Big Data, Hadoop Distributed File System, K-nearest neighbor, Prediction.

1. Introduction

Big data is a group of enormous volume of structured and unstructured data from different sources. Big data provide great challenges [1] in terms of data complexity, computational complexity, and system complexity. Due to its complexity it require a new architecture, techniques, algorithms, and analytics to manage it, read out the values and extract the hidden knowledge . Complexity, diversity, frequently changing workloads and rapid evolutions of big data systems raise great challenges in big data benchmarking [3]. Yaxiong Zhao et al [5] presented various schemes for handling the problems of big data analysis through Map Reduce framework over HDFS. Big Data Bench has very low operational intensity and the volume of data input has non negligible impact on micro-architecture characteristics [4].

2. Related works

Recently, Big Data can be applied to health sciences [6] such as healthcare, sensor-based health conditions, Internet-based epidemic surveillance, and food safety monitoring. Pakize and Gandomi [7] presented the comparative analysis of classification Algorithms based on Map-Reduce Model. Model based sensor data approximation [8] reduces the amount of data for query processing. KNN is used for classification and prediction. KNN has an in-memory tree component and is used to store that maps the tree nodes to the modeled data segments. Model based sensor data approximation [8] reduces the amount of data for query processing.

Huan et al [10] proposed a novel data partition scheme to reduce network traffic cost for a Map-Reduce job. They introduced an aggregator to reduce merged traffic from multiple map tasks.. Jeffrey et al [9] proposed a system that parallelizes the computation across large-scale clusters of machines. It handles machine failures and schedules inter-machine communication to make efficient use of the network and disks. It takes more computation cost. Huan Ke et al [10] designed aggregation architecture that used Map-Reduce framework for minimizing the data traffic during the shuffle phase. The aggregators resided anywhere in the cloud.

Kyuseok [11] introduced the Map-Reduce framework based on Hadoop and present the state-of-the-art in Map-Reduce algorithms for query processing, data analysis and data mining. K-Means Clustering Algorithm [12][13] plays a vital role to process high volume data over a distributed environment in Hadoop.

3. Proposed system

The k-Nearest Neighbor technique was implemented on a setup that consisting three nodes

connected over a private LAN and it is given in Fig. 1. One node was used as a Name node and Job Tracker, the other three other nodes were used as Data nodes and Task Trackers. Apache Hadoop version 2.7.2 was installed on all the nodes and the single node and consequent. The training data points were stored in a data set called Agriculture.csv and the testing data points were stored in AgriParams.txt. The required data set were copied into the HDFS. The Hadoop Map-Reduce process until all the data points in the testing dataset was classified. We started with 2500 data points of about 63KB in size and gradually increased the number of points up to 1 million about 100MB in size.

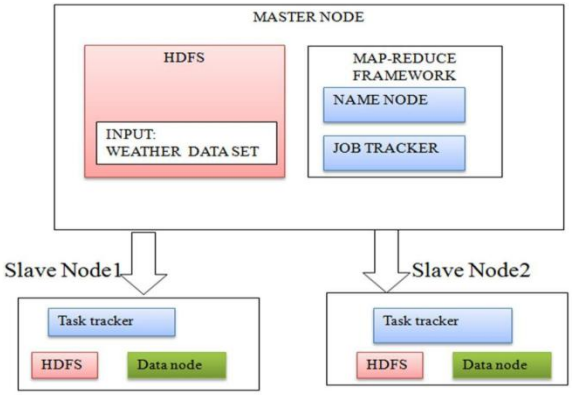

Fig. 1. Multi-node architecture

KNN is a non-parametric lazy learning algorithm. Being a non-parametric algorithm it does not make any assumptions on the underlying data distribution. KNN does not use the training data points to do any generalization and KNN keeps all the training data. So, the training phase is pretty fast. KNN makes decision based on the entire training data set.

3.1. Training phase

KNN Algorithm does not explicitly require any training phase for the data to be classified. The training phase usually involves storing the data vector co-ordinates along with the class label. The class label in general is used as an identifier for the data vector. This is used to classify data vectors during the testing phase.

3.2. Testing phase

The aims of testing phase is to find the class label for the new point for given data points. The algorithm is discussed for k=1 or 1 Nearest Neighbor rule and then extended for k=k or k Nearest Neighbor rule. a) K=1 or 1 nearest Neighbor.

The agriculture datasets are processed on the HDFS. Map function is forked for every job. These maps are run in any node under distributed environment configured under Hadoop configuration. The job distribution is done by the Hadoop cluster setup and datasets are required to be put in HDFS. Figure 2 shows the parallelism of Map-Reduce for agriculture dataset.

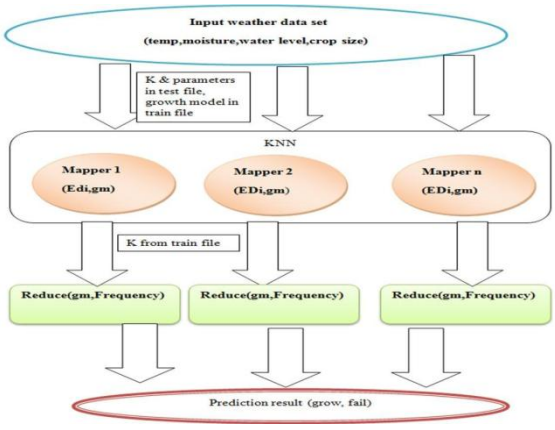

Fig. 2. KNN Map-Reduce implementation

3.3. Map-reduce of KNN

The agriculture data set is processed into Hadoop sequence files on the HDFS Name Node. The data sets were read from the local directory, sequenced, and written back out to a local disk. The resulting sequence files were ingested into the Hadoop file system with the default replica factor of three. The job containing the actual Map-Reduce operation was submitted to the Name Node to be run. Along with the JobTracker, the Head Node schedules and runs the job on the cluster. Hadoop distributes all the mappers across all data nodes that contain the data to be analyzed. On each data node, the input format reader opens up each sequence file for reading and k is initialized to token values.

3.3.1. Procedure for Mapper

1. Load data set containing test data
2. Read K values and other parameters (crop size, water level, temperature, moisture) from test data set.
3. Normalize the parameters.
4. Normalized attributes
5. Write the euclid distance of test data points from all training data points with their growth model in ascending order of distances.
6. Knn.map(EDi ,Gm)

Each Mapper gets the values of each attributes in the form of i,i+1,…,i+n. The Euclid distance is calculated for each attributes and the

intermediate key is derived by the KNN algorithm. Hadoop distributes all the mappers across all data nodes that contain the data to be analyzed. On each data node, the input format reader opens up each sequence data set for reading and k is initialized to token values. Calculating Euclid distance measures directly from the data set where attributes have different measurement scales. So normalize the data and transform all the values to a common scale. Then the normalized value for each attribute is calculated by using equation 1, 2 and 3 taking min &max values for each attributes such as temperature, moisture, and Water level.

3.3.2. Reducer

1. Input: Knn.map(ED_i, Gm)
2. Load Test data set
3. Read Test data point which is K value
4. Set counter for frequency
5. Iterate through high K distance for particular data point and increment counter for the model value
6. For i 0 to k
7. Increase counter value
8. Examine which model has highest counter value
9. Write output data set with most common model along with near K test data points
10. End procedure

The HDFS paths for test and train data set are loaded for mapper and reducer function when KNN map reduce task are initialized as a job.

4. Performance analysis

4.1. Setup

We employ agriculture sensor data from UCI machine learning repository, the size of raw sensor data is upto 100MB including 4million data points. We developed a system using hadoop distributed environment. Experiment performed on single and multi node cluster architecture.

Table 1. Single node specification

Disk space	100 GB
Ram	3 GB
Number of cores	8
Processor speed	2 GHz
Data set size (mb)	3.125,6.25,12.5,25,50,100

Table 2. Multi node specification

Description	Master	Slave
Disk space	100 GB	200 GB
RAM	3 GB	3 GB
Number of cores	12	8
Processor speed	2.7GHz	2GHz
Ethernet connection	100Mbs	100Mbs
Number of nodes	1	2
Data set size (MB)	-	25,50,100

4.2. Performance analysis

The first observation made from the experimental results was that the Map-Reduce KNN classification for smaller datasets of 3.125 Mb having data point of 1, 20,000. This fact is illustrated by the analysis table. Repeating the experiment three times helped us to reliably monitor the time taken to classify. The Algorithm worked well with the dataset chosen for experimentation. The main purpose was to identify the suitability of using Map-Reduce for KNN to predict the growth of plant in agriculture field. In this approach, we stored the data points both training and testing in local data sets. The map-reduce KNN was then applied to assign the testing data points to the closest class label. We noted the time taken to converge to the final classification, number of iterations.

Table 3. Performance analysis

S.No	Data set size	No. of recrods	MAP (ms)	Reduce (ms)	CPT time (ms)	GC time (ms)
1	3.125 Mb	1,20,000	1228	1051	2279	90
2	6.25 Mb	2,40,000	1257	1086	2343	110
3	12.5 Mb	4,80,000	1757	1584	3341	150
4	25 Mb	9,60,000	1792	1592	3384	185
5	50Mb	19,20,000	1950	1936	3886	196
6	100Mb	38,40,000	2210	2002	4212	221

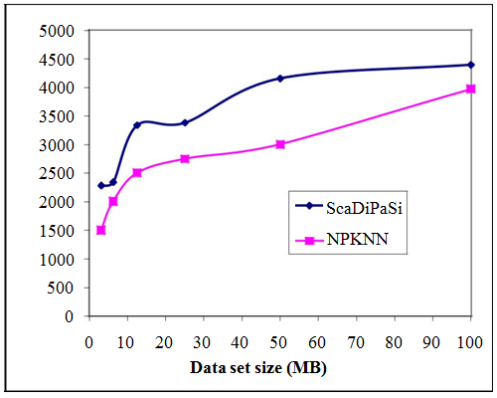

Fig. 3. Data set

The first observation made from the experimental results was that the Map-Reduce KNN classification for smaller datasets of 3.125 Mb having data point of 1, 20,000. This fact is illustrated by the analysis table. Repeating the experiment three times helped us to reliably monitor the time taken to classify. The Algorithm worked well with the dataset chosen for experimentation. The main purpose was to identify the suitability of using Map-Reduce for KNN to predict the growth of plant in agriculture field. In this approach, we stored the data points both training and testing in local data sets. The map-reduce KNN was then applied to assign the testing data points to the closest class label. We noted the time taken to converge to the final classification, number of iterations.

4. Conclusion

In this paper, we proposed a Map-Reduce implementation of KNN algorithm for analysis of big data. The motivation is to predict data from agriculture data set. The process of building algorithm can be very time consuming. Besides, with the volume of dataset increased, the required data cannot fit in memory. To solve above challenges, we therefore proposed a parallel KNN based on Map-Reduce in HDFS. In order to evaluate the efficiency of our method, we conducted experiments on a massive dataset. The empirical results indicated that our Map-Reduce implementation of KNN algorithm exhibited both time efficiency and scalability.

Reference

1. Xiaolong J, Benjamin WW, Xueqi C, Yuanzhuo W, “Significance and Challenges of Big Data Research”, Big Data Research, vol.2, no.2, pp. 59–64, 2015.
2. Pekka P, Daniel P, “Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems”, Big Data Research, vol.2, no.4, pp.166–186, 2015.
3. Min C, Shiewen M, Yunhao L, “Big Data: A Survey”, Journal Mobile Networks and Applications, vol.9, vol.2, pp. 171-209, 2014.
4. W. Gao, Y. Zhu1, Z. Jia, C. Luo, L. Wang, Z. Li, J. Zhan, Y. Qi, Y. He, S. Gong, Xiaona Li,S. Zhang, and B. Qiu. BigDataBench: a Big Data Benchmark Suite from Web Search Engines.in The Third Workshop on Architectures and Systems for Big Data in conjunction with The 40th International Symposium on Computer Architecture, May 2013.
5. Yaxiong Z, Jie WD, “A Data Aware Caching for Big-Data Applications Using The MapReduce Framework”, Proc. 32nd IEEE Conference on Computer Communications, INFOCOM 2013, IEEE Press, pp.35-39, 2013.
6. Tao H, Liang L, Xuexian F, Peng A, Junxia M, Fudi W, “Promises and Challenges of Big Data Computing in Health Sciences”, Big Data Research, vol.2, no.1, pp.2–11, 2015.
7. Pakize, SG, “A Comparative Study of Classification Algorithms Based On MapReduce Model”, International Journal of Innovative Research in Advanced Engineering, vol.1, no.7, pp.251-254, 2014.
8. Punithavathani, DS, Sujatha, K, Jain, JM, “Surveillance of anomaly and misuse in critical networks to counter insider threats using computational intelligence”, Cluster Computing, vol.18,no.1, pp. 435-451, 2013.
9. Jeffrey D, Sanjay G, “MapReduce: Simplified Data Processing on Large Clusters”, Magazine of Communications of the ACM, vol.51.no.1, pp 1-13, 2008.
10. Huan K, Peng L, Song G, “Ivan Stojmenovic, Aggregation on the fly: Reducing traf fic for big data in the cloud”, IEEE Network, vol.29, no.5, pp. 17 – 23, 2015.
11. Kyuseok S, “MapReduce Algorithms for Big Data Analysis”, Lecture Notes in Computer Science, vol.7813, pp 44-48, 2012.
12. Prajesh P. Anchalia, Anjan K. Koundinya, Srinath N. K., "MapReduce Design of K-Means Clustering Algorithm", Proceeding of

International Conference on Information Science and Applications , pp.1-5, 2013.

13. Prajesh PA, Kaushik R, “The k-Nearest Neighbor Algorithm Using MapReduce Paradigm”, Proceeding of Fifth International Conference on Intelligent Systems, Modelling and Simulation, pp.513-518, 2014.

14. Mohammadhossein B, Mahd N, “ScaDiPaSi: An Effective Scalable and Distributable MapReduce-Based Method to Find Patient Similarity on Huge Healthcare Networks”, Big Data Research, vol.2, pp.19–27,2011.