Performance Analysis of aScalable Naïve Bayes Classifier on Beyond MapReduce

Myat Cho Mon Oo[1], Thandar Thein[2]
University of Computer Studies, Yangon[1], University of Computer Studies, Maubin[2]
myatchomonoo@ucsy.edu.mm[1], thandartheinn@gmail.com[2]

Abstract

Many real world areas from different sources generate the massive data with large volume of high velocity, complex and variable data. The massive data becomes a challenge for machine learning algorithm because they are difficult to process and extract knowledge using traditional analysis tools. Therefore, a massively scalable and parallel algorithm is needed to process and analyze such massive data. Recently Hadoop MapReduce framework has been adapted for processing large data in an extremely parallel mining. MapReduce may not a very good fit for most of the scalable machine learning that Mahout pioneered. For large scale machine learning on distributed system, Mahout Samsarais used with efficient distributed execution on Spark. This paper analyses the scalability of Naïve Bayes classifier which is implemented by Mahout Samsara. The performance of scalable Naïve Bayes classifier (SNB) on Beyond MapReduce and traditional Naïve Bayes classifier arealso comparedover different data sets. The experimental results show that SNB on Beyond MapReduceis more suitable to classify massive datasets in distributed computing environment and it provides a better accuracy and minimal processing time than traditional Naïve Bayes classifier.

Keywords- Apache Spark, Beyond MapReduce, HDFS, Mahout Samsara,Massive data, Naïve Bayes Classifier

1. Introduction

The scale of data is increasing overwhelmingly during the past decades. The IBM Data Flood Infographic shows that 2.5 quintillion bytes of data are created every day [1].Traditional data mining algorithms are not well suited to process the full value of massive data. Therefore, a massively scalable and parallel algorithm is needed to process and analyze such datasets. For processing huge amount of data, Hadoop is becoming the core technology to solve the huge data problems for large organizations with cloud storage. A commonly used architecture for Hadoop consists of Hadoop Distributed File System (HDFS) for massive data storage and MapReduce for distributed parallel computing.

ISBN 978-99971-0-381-9 © 2017 ICAIT

Machine learning is also ideal for exploiting the opportunities hidden insight in massive data. Apache Mahout is an open source machine learning library built on top of Hadoop to provide distributed analytics capabilities. Although it was originally developed with MapReduce based algorithm, MapReduce was inefficient for most of the scalable machine learning that Mahout pioneered because of its limitation. Also, alternative frameworks such as Spark have finally become much more viable.

Spark absorbs the advantages of Hadoop MapReduce, unlike MapReduce, the intermediate and output results of the Spark jobs can be stored in memory, which is called Memory Computing. Memory Computing improves the efficiency of data computing. So, Spark is better suited for iterative applications, such as Data Mining and Machine Learning [2].

Starting from the release 0.10.0, a new generation of mahout was born for building backend independent programming environment, also called the code name, "**Samsara**". Mahout Samsara is backend-agnostic and uses a Scala-based programming environment to support writing parallel mathematical languages.

This paper aims to analyze the performance and scalability of SNB on Beyond MapReducefor the massive data classification. The remainder of this paper is organized as follow. Section 2 describes the related work of this paper andsection 3 introduces some background information. And then architectureof SNB on beyond MapReduce is presented in section 4. After that the performance evaluation and result discussions are shown in section 5.Finally, the conclusion of this paper is summarized in section 6.

2. Related Work

It has become difficult to process massive amount of data with rapid growth in internet although traditional data mining techniques have achieved good performance for small amount of data scale on single machine. A large number of approaches have been proposed to solve this issue. Most of them based on MapReduce model and distributed file system. The authors [3] proposed Map Reduce-based Naïve Bayes classifier (MP-NB)to process CRIA (Customer Requirement Information Acquisition) classification on large scale mobile data.MP-NB was implemented on

Hadoop Platform with MapReduce programming model to solve data-intensive computing problems.

To analyze the enormous data set, the authors[4] proposed an improved Naïve Bayes classifier for large-scale text classification. They used MapReduce programming model to improve the accuracy of proposed classifier and to provide good scalability and extensibility for text classification.

The main challenges of becoming massive data are high dimensionality, diversity and high analysis value return. For processing massive data, the authors [5] proposed a scalable random forest algorithm based on MapReduce (SMRF). They showed that their new algorithm, SMRF is more suitable to classify massive data sets in distributing computing environment than traditional Random Forest algorithm. Although SMRF algorithm had higher performance while comparing with traditional Random Forest algorithm, it had the equally accuracy degradation because of using MapReduce. MapReduce is inefficient for application that share data across multiple steps, including iterative algorithms or iterative queries.

This paper intends to analyze the performance and scalability of SNB for the massive data classification. By using SNB, it performs efficiently processing on massive datasets in distributed environment and provides good performance results than traditional Naïve Bayes classifier.

3. Preliminaries

This section provides the preliminaries of this paper. First, Apache hadoop and the Spark framework are introduced in Section 3.1 and 3.2. Then Apache Mahout and Mahout Samsara are presented in Section 3.2 and 3.4.

3.1. Apache Hadoop Framework

Apache Hadoop[6] is an open-source software framework for storing and processing large data in a distributed manner. It supports data intensive distributed applications on large clusters built of commodity hardware. This framework is designed to be scalable, which allows the user to add more nodes as the application requires. Hadoop consists of the Hadoop Distributed File System for big data storage and MapReduce engine for distributed processing. Hadoop cluster consists of a single master node and many worker nodes. HDFS is based on master/slave architecture. The development of Hadoop-based data mining techniques has been widely spread, because of its fault-tolerant mechanism and its ease of use.

Despite its popularity, MapReduce becomes inefficient to develop some scalable machine learning algorithms because of its limitations. Since MapReduce is only suitable for batch processing job, implementing interactive jobs and model become expensive due to the huge space consumption by each job. This costs much time for disk I/O operations and also massive resources for communication and storage. To overcome this problem, many distributed processing framework are emerged.

3.2 Spark Framework

Apache Spark [7] is a fast and general engine for large-scale data processing. It has an advanced DAG (Directed Acyclic Graph) execution engine that supports acyclic data flow and in-memory computing. The core abstraction of Spark is Resilient Distributed Dataset (RDD) which has a better ability of computing and fault tolerance. So, Spark can allow us to store a data cache in memory, to perform computation and iteration for the same data directly from memory. It saves huge amount of disk I/O operation time. Therefore, it is more suitable for developing scalable machine learning algorithms.

3.3. Apache Mahout

Apache Mahout [8] is an environment for creating scalable, performant, machine learning applications. It is a machine learning library that runs over the hadoop system for solving clustering, and classification problems. It born a new generation of mahout, linear algebra environment, known as the code name “Mahout Samsara”(release 0.10.0 or later). MapReduce was not a very good fit for most of the machine learning that mahout pioneered. In place of Hadoop MapReduce, Mahout has been focusing on implementing flexible and backend-agnostic machine learning environment.

3.4 Mahout Samsara

Mahout Samsara is a new generation of mahout. It is also known as “Beyond MapReduce” because it is the part of mahout that deals with more advanced back-ends, post-mapreduce generation: Spark, Flink, and H_2O. These backends extend the set of distributed paradigms beyond just MapReduce. Therefore, machine learning algorithms built with the mahout Samsara DSL are better served for iterative nature of applications.

4. Scalable Naïve Bayes Classifier (SNB) on Beyond MapReduce

Naïve Bayes algorithm is a popular algorithm in text classification field. It is a simple probabilistic classifier based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. Samsara currently has two flavors of Naïve Byes implemented in its distribution. The first is standard Multinomial Naïve Bayes ("MNB" or "Bayes") and the latter is a variation on Transformed Weight-normalized Complement Naïve Bayes ("TWCNB" or "CBayes")[9]. In this paper, Samsara's MNB is used for massive data classification in distributed environment. Figure 1 shows a SNB on Beyond MapReduce Architecture.

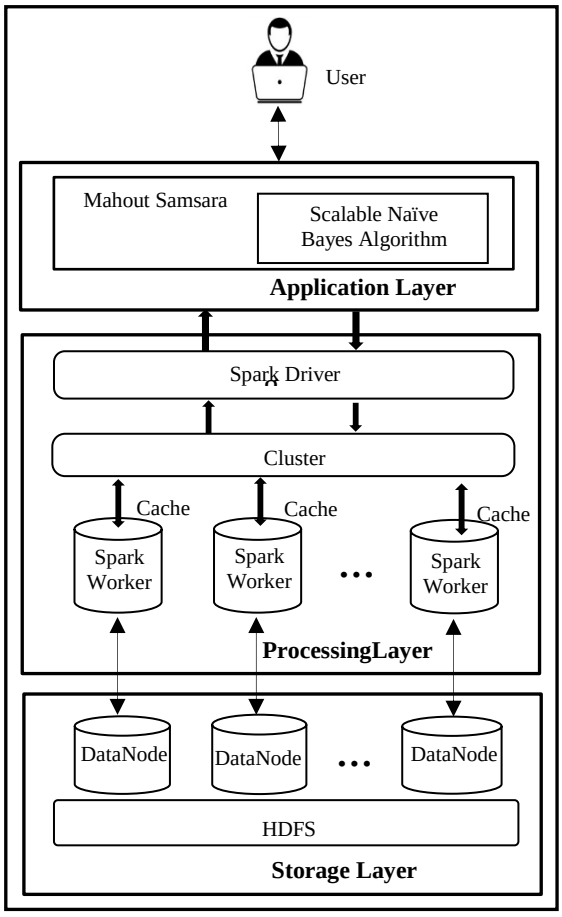

Figure 1. SNB on Beyond MapReduce Architecture

4.1 SNB for Text Classification

The structure of the SNB is shown in Figure 2. It is divided into three stages:initializing, training and label assignment. In the transformation stage, the dataset isacquired and vectorized the document. In order to make good use of the computing resources, Hadoop cluster is implemented. Hadoop is a framework that admits the data in sequence file format. Mahout Samsara over hadoop also admits the data in the sequence file directory. The input data is converted to sequence file format to parse the <text> element of each document. After taking the sequence file conversion phase, the documents are vectorized using ***mahout seq2sparse***. The command ***seq2sparse*** converts the sequence file directory to vector format. The sequence file will be accepted as input and produce the output as vector using a weighting factor like TF-IDF (Term frequency-Inverse Document Frequency) scheme.

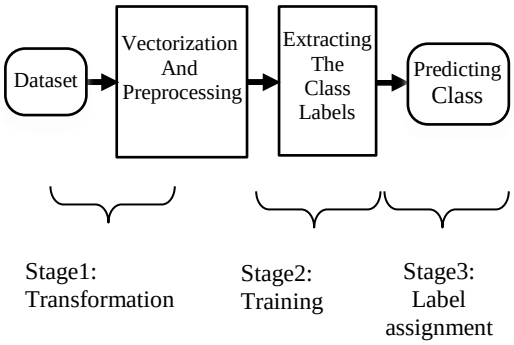

Figure 2. The Structure of SNB

In the training stage, SNB uses the Spark random Split API to split the training and testing sets. Since Spark backend environment is chosen, the algorithm in Mahout Samsara can take advantages of Spark native function. SNB stores the class label of each vectorized document as the row keys. And then, it extracts the all possible document identifier for each document.

SNB on Beyond MapReduce
Step 1: Launch the Mahout Spark-shell
Step 2: Samsara Naïve Bayes imports.
Step 3: Hadoop imports.
Step 4: Read the full datasets into a DRM
Step 5: Extract categories and sum TF-IDF values for each category.
Step 6: Split the data into 60/40 training/testing sets
Step 7: Build the Multinomial Naïve Bayes model on the training set
Step 8: Test the Multinomial Naïve Bayes model on the testing set

Figure 3. SNB on Beyond MapReduce

In the label assignment stage, SNBassigns a label to a vectorized document using a classification function.

It predicts a classification of the document by assigning a class with the largest posterior probability.The SNB on Beyond MapReduce with term weighting scheme is described in figure 3.

5. Performance Evaluation

This section analyzes the performance of SNB on Beyond MapReduce. Firstly, the performance of SNB is compared with traditional Naïve Bayes classifier (NB) in terms of accuracy. And then the scalability of SNB is tested on distributed nodes. The effectiveness in classification for the performance analysis of SNB will be evaluated using confusion matrix which records correctly and incorrectly recognized examples for each class.The four matrices of performance that measure the classification quality for the positive and negative classes independently are:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$TP_{rate}=\frac{TP}{TP+FN} \quad (2)$$

- **True Positive rate**isthepercentageofpositive cases correctly classified as belonging to the positive class.

$$TN_{rate}=\frac{TN}{FP+TN} \quad (3)$$

- **True Negative rate**is the percentage of negative cases correctly classified as belonging to the negative class.

$$FP_{rate}=\frac{FP}{FP+TN} \quad (4)$$

- **False Positive rate**is the percentage of negative cases misclassified as belonging to the positive class.

$$FN_{rate}=\frac{FN}{TP+FN} \quad (5)$$

- **False Negative rate**is the percentage of positive cases misclassified as belonging to the negative class.

5.1. Experimental Environment

This section performs comprehensive experimentto compare accuracy level of SNBon Beyond MapReduce with the traditional multinomial Naïve Bayes algorithm (NB). All the experiments have been carried out over a Hadoop cluster of three computing nodes having high configuration machine of Intel CORE i7 processor, 8 GB of RAM, 1 TB HDD on Linux Ubuntu-16.04 system.The computer cluster is set up with Hadoop, one name node and three data nodes.The specific details of the software used and its configuration are open-sour Apache Hadoop distribution (hadoop 2.6.0), apache spark (1.5.2), maven (3.3.3), scala version (2.10.4), java version (jdk-7.79) and the latest release of Mahout Samsara (0.12.3).The descriptions of datasets used in this paper are presented in table 1.

Table 1.Experimental Datasets

Dataset	Description
D1	Breast-cancers [10] -Wisconsin Diagnostic Breast Cancer (WDBC) dataset -Number of instances : 569 -Number of attributes : 32
D2	Iris [10] -Number of instances : 150 -Number of attributes : 4
D3	Adult [10] -Number of instances : 48842 -Number of attributes : 14
D4	Movie Reviews [11] - Rotten Tomatoes movies review dataset -contain 1000 positive and 1000 negative reviews
D5	Reuter-21578[12] - newswire dataset -contain 21578 newswire documents
D6	OHSUMED[13] - medical abstract dataset -contain 348,566 references and 6 categories

5.2. Performance Evaluation and Result Discussion

SNB is performed in Hadoop cluster distributed computing environment. Weka workbench [14] is adopted to run traditional multinomial Naïve Bayes algorithm (NB). To compare the two algorithms with different datasets, the datasets are partitioned into 60/40 ratio for training and testing.The accuracy measurement of each dataset is shown in Figure 4.

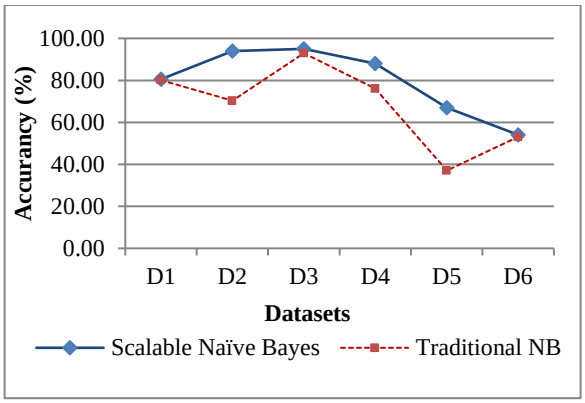

Figure 4. SNB vs. NB

According to the comparison result, the accuracies of SNB in data set D2 and D3 are almost 94% and 95% respectively, which are much higher than NB.In D1 and D4 also, its accuracies achieved beyond 80% and 88% which is 1.4% and 12% larger than NB. In D5 and D6, the two classifiers obtainedequally accuracy degradationbecause this datasets contained more instances from one class than from the other. Therefore, the class imbalanced problem becomes a challenge for massive data mining. However, it can be noted that SNB achieved the better accuracy on six datasets than traditional Naïve Bayes classifier.

5.3. Scalability Test

SNB isparallelized on the distributed hadoop cluster environment. Mahout Spark-shell is used for unseen data during the initial vectorization of the training and testing sets. This section performs the scalability test of SNB with different computing nodes. Itanalyses the scalability of SNB using Enron's email spam datasetin term of accuracy and processing time.In order to verify the scalability of SNB on Beyond MapReduce, the "Enron email" datasets are enlarged according to its formats. The number of ham and spam emails in each Enron dataset is summarized in Table 2.

Table 2. Summary of Enron Email Datasets

Enron Dataset	Number of mails	
	Spam	Ham
E 1	8996	13545
E 2	12671	15045
E 3	17171	16545
Total number of mails 83973		

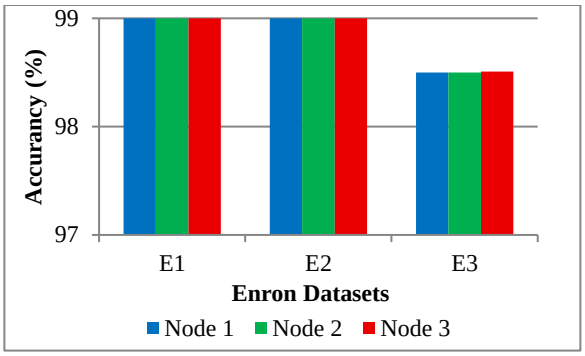

Figure 5.SNB with Different Node Number Classification Result

Figure 5 shows the accuracy level of SNB model on 1, 2 and 3 Data Nodes of hadoop cluster computing environment. According to this result, SNB provides the good accuracy results beyond the 98% with different datasets on each data node.

Figure 6 shows the total processing time of SNB on distributed hadoop cluster environment. For processing massive data, multiple data nodes are simultaneously parallelized. It reduces the processing time to minimal and provides the fast distributed computing. As a result, SNB running with 3 data nodes can providefaster computing time in most dataset classifications.

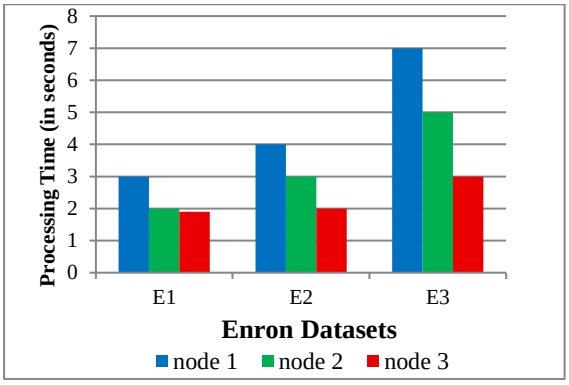

Figure 6.Comparison of Processing Time of SNB

According to the results, the scalable Naïve Bayes classifier on Beyond MapReduce (SNB) can provide agood scalability for massive data classification.

6. Conclusion

Scalability has become one of the core concept slash buzzwords of massive data. Massive data requires a scalable and parallel machine learning algorithm for efficiently processing. Analytical processing complexities are required to take minimal time to get results. In this paper, the performance of a scalable naïve Bayes classifier on Beyond MapReduce is

analyzed. SNB can process the massive data and run multiple tasks simultaneously. Our comparative study can show that SNBon Beyond MapReduce has a better accuracy and faster processing time than the traditional Naïve Bayes algorithm. It also provides the good scalable performance on distributed environment. As future work, the issues in classification of massive datasets with skew class distribution will be considered. Regarding the performance analysis of SNB on Beyond MapReduce, techniques to deal with extremely class imbalanced problem will be studied.

7. References

[1] Big Data Flood Infographic. [Online] Available: https://www.ibm.com/

[2] J. Fu, J. Sun, K. Wang, "SPARK—A Big Data Processing Platform for Machine Learning", in 2016 IEEE International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration , 2016, pp. 48-51

[3] X. Wang, B. Sheng, L. Xue, Z. Xiao, "Classification of Customer Requirements on Map Reduce-based Naïve Bayes". In 2016, IEEE international Conference on Big Data Analysis (ICBDA). IEEE 2016

[4] C. Huaixin, "An improved Naive Bayes Classifier for Large Scale Text". In Proceedings of the 14th International Conference on Applications of Computer Engineering (ACE '15),Seoul, South Korea September 5-7, 2015

[5] J. Han, Y. Liu, X. Sunl, "A Scalable Random Forest Algorithm Based on MapReduce". In 2013,4th IEEE international conference on ICSESS, page 849-832. IEEE 2013

[6] Apache Hadoop. [Online] Available: http://hadoop.apache.org/

[7] Apache Spark. [Online] Available: http://Spark.apache.org/

[8] Apache Mahout. [Online] Available: http://mahout.apache.org/

[9] Dmitriy Lyubimov, Andrew Palumbo.*Apache Mahout: Beyond MapReduce*. 2016

[10]A. Frank and A. Asuncion, "UCI machine learning repository", 2010. [Online] Available: http://archive.ics.uci.edu/ml

[11]Movie Review Data. [Online] Available: https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews

[12]Reuters-21578collection Apte' split. [Online] Available: http://disi.unitn.it/moschitti/corpora.htm

[13]Ohsumed collection. [Online] Available: http://disi.unitn.it/moschitti/corpora.html

[14]Weka tool. [Online] Available: http://www.filehorse.com/download-weka-64/