An Analysis on Machine learning Algorithms implemented on HADOOP Map Reduce

J V N Lakshmi [1], Ananthi Sheshasaayee [2]

[1]Department of Computer Science, SCSVMV University, Kanchipuram, INDIA.
[2]Department of Computer Science, Quaid- e- Millat College for women, Chennai, INDIA.
[1] jlakshmi.research@gmail.com
[2] ananthi.research@gmail.com

***Abstract*— Big data is data that exceeds the processing capacity of conventional database systems. It is not easy to analyse such huge data. This requires machine based systems and technologies in order to process. Map Reduce a distributed parallel programming model runs on Hadoop environment, processes large volumes of data. A parallel programming method can be applicable on linear regression algorithm and Support vector machines algorithm from machine learning community to parallelize speed up on multicore system for efficient timing efficiency.**

Keyword- Big Data, Data Analytics, Support vector machines, Hadoop, Linear Regression, Machine Learning, Map Reduce, Multicore.

I. INTRODUCTION

Big Data: The next frontier for innovation, competition and productivity. The amount of data in our world has been exploding, and analysing large data sets—so-called big data—will become a key basis of competition, underpinning new waves of productivity growth, innovation, and consumer surplus [1].

The hot IT buzzword of 2012, big data has become viable as cost-effective approaches have emerged to tame the volume, velocity and variability of massive data.

Assuming that the volumes of data are larger than those conventional relational database infrastructures can cope with; processing options break down broadly into a choice between massively parallel processing architectures such as, Apache Hadoop-based solutions. Apache Hadoop uses a Distributed file system called HDFS and a parallel paradigm called Map Reduce [3].

Machine Learning techniques are highly scalable for data analytics. In this paper an analysis on Linear Regression and Support vector machines algorithms are proposed for implementing on Hadoop for improving the timing efficiency [2].

In this paper section 2 introduces Hadoop framework, section 3 outlines Map Reduce paradigm. Section 4 discuss on Machine learning and its algorithms. In section 5 conclusions for improving Hadoop Implementation.

II. HADOOP AND HDFS

HADOOP (High Availability Distributed Object Oriented Platform) places no conditions on the structure of the data it can process. At its core, Hadoop is a platform for distributing computing problems across a number of servers [10]. First developed and released as open source by Yahoo, it implements the MapReduce approach pioneered by Google in compiling its search indexes. Hadoop's MapReduce involves distributing a dataset among multiple servers and operating on the data: the "map" stage. The partial results are then recombined: the "reduce" stage [6].

To store data, Hadoop utilizes its own distributed filesystem, HDFS, which makes data available to multiple computing nodes. A typical Hadoop usage pattern involves three stages:

- loading data into HDFS
- MapReduce operations and
- Retrieving results from HDFS.

a. Hadoop Distributed File System

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost

hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets [10].

To understand how it's possible to scale a Hadoop cluster to hundreds (and even thousands) of nodes, you have to start with the Hadoop Distributed File System (HDFS). Data in a Hadoop cluster is broken down into smaller pieces (called blocks) and distributed throughout the cluster. In this way, the map and reduce functions can be executed on smaller subsets of your larger data sets, and this provides the scalability that is needed for big data processing.

III. MAP REDUCE MODEL

The term MapReduce actually refers to two separate and distinct tasks that Hadoop programs perform. The first is the map job, which takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). The reduce job takes the output from a map as input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce job is always performed after the map job [11].

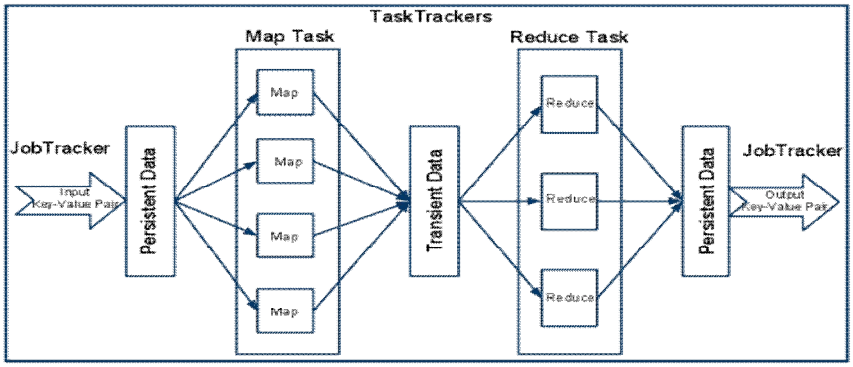


Fig 1: Map Reduce on HDFS Src : [7]

The data flow process given in Fig 1 of Map Reduce with HDFS system is illustrated as below.

- A distributed filesystem spreads multiple copies of the data across different machines. This not only offers reliability without the need for RAID-controlled disks, it offers multiple locations to run the mapping. If a machine with one copy of the data is busy or offline, another machine can be used.
- A job scheduler (in Hadoop, the Job Tracker), keeps track of which MR jobs are executing, schedules individual Maps, Reduces or intermediate merging operations to specific machines, monitors the success and failures of these individual *Tasks*, and works to complete the entire batch job.
- The filesystem and Job scheduler can somehow be accessed by the people and programs that wish to read and write data, and to submit and monitor MR jobs.

Apache Hadoop is such a MapReduce engine. It provides its own distributed filesystem and runs [Hadoop MapReduce] jobs on servers near the data stored on the filesystem - or any other supported filesystem, of which there is more than one.

IV. MACHINE LEARNING ALGORITHMS

Machine learning is a type of artificial intelligence that provides computers with the ability to learn without being explicitly programmed [8]. Machine learning focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data [9].

The process of machine learning is similar to that of data mining. Both systems search through data to look for patterns. However, instead of extracting data for human comprehension -- as is the case in data mining applications -- machine learning uses that data to improve the program's own understanding.Machine learning programs detect patterns in data and adjust program actions accordingly.

a. Support vector Machines

The line that maximizes the minimum margin is a good fit. The model class of "hyper-planes with a margin of m" has a low VC dimension if m is big. This maximum-margin separator is determined by a subset of the datapoints. Datapoints in this subset are called "support vectors". Support vector machines (SVM) are a group of supervised learning methods that can be applied to classification or regression [5].

Support vector machines represent an extension to nonlinear models of the generalized portrait algorithm developed by Vladimir Vapnik [7]. The SVM algorithm is based on the statistical learning theory. It will be useful computationally if only a small fraction of the datapoints is support vectors [12].
Distance from example $\mathbf{x}_i$ to the separator is
Examples closest to the hyper plane are ***support vectors***. ***Margin*** ρ of the separator is the distance between support vectors.

$$r = \frac{\mathbf{w}^T \mathbf{x}_i + b}{\|\mathbf{w}\|}$$

Let training set $\{(\mathbf{x}_i, y_i)\}_{i=1..n}$, $\mathbf{x}_i \in \mathbf{R}^d$, $y_i \in \{-1, 1\}$ separated by a hyper plane with margin ρ. Then for training example $(\mathbf{x}_i, y_i)$:

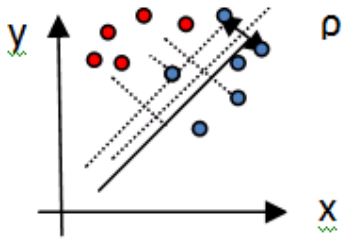


Fig 2 : Support Vectors nearer to the Hyper plane

$$\begin{aligned} \mathbf{w}^T\mathbf{x}_i + b \le -\rho/2 \quad \text{if } y_i = -1 \\ \mathbf{w}^T\mathbf{x}_i + b \ge \rho/2 \quad \text{if } y_i = 1 \end{aligned} \quad \Leftrightarrow \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \ge \rho/2$$

For every support vector $\mathbf{x}_s$ the above inequality is equality. After rescaling **w** and *b* by $\rho/2$ in the equality, then obtain the distance between each $\mathbf{x}_s$ and the hyperplane is

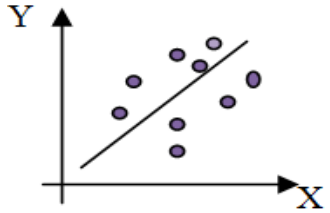
$$r = \frac{y_s(\mathbf{w}^T\mathbf{x}_s + b)}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|} \qquad \rho = 2r = \frac{2}{\|\mathbf{w}\|}$$

Then the margin can be expressed through (rescaled) **w** and b as:

The time complexity calculated for single core is $O(m^2n)$ and multi core is $O(\frac{m^2n}{p} + n\log(p))$. Linear regression speed up is 13.6% for 16 cores [4].

b. ***Linear Regression***
Technique used for the modeling and analysis of numerical data [16]. Regression exploits the relationship between two or more variables so, that can gain information about one of them through knowing values of the other. Regression can be used for prediction, estimation, hypothesis, testing, and modeling causal relationships [15].
Suppose to model the dependent variable Y in terms of predictors, x_1, x_2, ….x_m.
$Y = f(x_1, x_2, \ldots x_m) \rightarrow Y = \vartheta^T x$

$\theta^* = min_\theta \ \sum_{i=1}^{m}(\theta^T xi - yi)^2$



Y = A + BX

Fig 3: Linear Regression for Y = A + BX line

- Typically data is required to estimate function f.
- Therefore, usually have to assume that it has some restricted form, such as linear equations.

The Parameter θ is typically solved by defining x_1, x_2, ….x_m and $\vec{y} = [y_1, \ldots, y_m]^m$ be vectors of target labels and solving normal equations [18].

$$\theta^* = (X^T X)^{-1} X^T Y$$

To compute $\theta^* = A^{-1} b$

$A = X^T X$ and $b = X^T Y$ as follows
$A = \sum_{i=1}^{m} (xixi^T)$ and $b = \sum_{i=1}^{m} xiyi$.

The time complexity calculated for single core is $O(mn^2 + n^3)$ and multi core is $O(\frac{mn^2}{p} + \frac{n^3}{p'} + n^2 \log(p))$. Linear regression speed up is 15% for 16 cores [4].

V. CONCLUSION

In this paper a comparison on two machine learning techniques Linear Regression and Support Vector machines, are implemented on Hadoop for efficient time analysis. Hadoop Map Reduce parallelize on these techniques yields significant results as SVM speed up on 16 cores is 13.6 % and Linear Regression speed up is 15%. The paper proposes the use of Hadoop with machine learning techniques for efficient performance.

REFERENCES

[1] Building Machine learning Algorithms on Hadoop for Bigdata by asha Shranvanthi NagaShree Monika IJET – UK 2013
[2] Verification and Validation of Map Reduce Program Model for parallel SVM on Hadoop Cluster by Kiran Ameersh kumar, Ravi IJCSI 2013
[3] Mahesh Maurya and Sunita Mahajan. “Performance analysis of MapReduce Programs on Hadoop cluster”, World Congress on Information and Communication Technologies 2012.
[4] Map Reduce for Machine Learning on Multicore Cheng- Tao Chu,snag Kyun Kim, Yi-an Lin, Andrew Y Ng with INTEL corporationin 2010.
[5] Chang, E. Y., Zhu, K., Wang, H., Bai, H., Li, J., Qiu, Z. and Cui, H. Parallelizing “Support Vector Machines on Distributed Computers”, 2007.
[6] J Dean and S Ghemawat Map Reduce : simplified data processing on large clusters. Operating systems Design implementation 2004
[7] Vladimire Vapnik : Parallel Support vector machines In NIPS 2004
[8] Gunnar Ratsch, “A Brief Introduction into Machine Learning”, Friedrich Miescher Laboratory of the Max Planck Society, 2004.
[9] http://radar.oreilly.com
[10] Apache Hadoop : http://hadoop.apache.org
[11] www.ibm.com/software/data/infosphere/hadoop/mapreduce
[12] Cortes and V. Vapnik. “Support-vector network”, Machine Learning, 20:273-297, 1995.
[13] Chih-Wei Hsu and Chih-Jen Lin. A Comparison of Methods for Multi-class Support Vector Machines, 13 (2): 415-425, 2002.
[14] C. C. Chang and C. J. Lin, “LIBSVM: A Library for Support Vector Machines”, National Taiwan University, Taipei, Taiwan, 2001.
[15] R E Welsch E KUH Linear Regression In Working paper 1977
[16] Linear Regression using Stata (v.6.3) Oscar Torres-Reyna fro Princeton University.
[17] The Inaugural Coase Lecture “An Introduction to Regression Analysis” Alan O. Sykes* University of Chicago
[18] “Multiple Linear Regression” by Mark Tranmer Mark Elliot from Cathie Marsh Centre for census and survey research