A Survey on Efficiency in Big Data Mining

Santhosh Pawar
Department of CS
KSWU Vijayapura

Abstract*-Data mining is the extraction of useful knowledge from a large amount of data from different heterogeneous sources. Nowadays data is growing rapidly, and mining from these massive sets of data become the most complex task. The bigdata mining is the ability to extract information from large complex data due to its volume, velocity, verity, veracity and value. Uncovering of the huge amount of heterogeneous bigdata will maximize the knowledge in the target domain. So bigdata mining become one of the exciting opportunities today. The traditional data mining tools are not capable of handling large distributed data. Effective big data mining requires scalable and efficient solutions that are also useful to all kind of users. So the combination of efficient and user-friendly data mining tools will provide a more effective and scalable bigdata data mining platform for users with all levels of expertise.*

Keywords-Big Data, Data Mining, Frameworks, Distributed Systems.

I. INTRODUCTION

Recent years, it is noticed that there is a tremendous increase in human ability to collect data from different sensors and devices which are in different formats from independent or connected applications. This flooding of data has raised human capability to process, analyse, store and understand these huge datasets. For example, the data in Internet. The Google indexed web pages were around one million in 1998 that is quickly reached 1 billion in 2000 that exceeded 1 trillion in 2008 and go on. This expansion is accelerated by the social networking applications, such as Facebook, Twitter, etc. Big Data is "high-volume, high-velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization" [1]. It has not only these three properties, but also have other two properties veracity and value. All these commonly known as 5 V's. In practice, this term refer the datasets that are difficult to gather, process and find solutions for queries using on-hand data mining tools. The purpose of Big data mining is to go beyond the usual request-response processing, market basket analysis etc. but to design and implement very large scale parallel data mining algorithm. Unveiling the huge volume of interconnected heterogeneous big data has the potential to maximize our knowledge in the target domain. Scalable and efficient solutions are needed for effective big data mining. They should be accessible for users of different levels of expertise. The significant challenges facing, relating to the vast amount of data, includes challenges in (1) system capabilities (2) algorithmic design (3) business models. For efficient and scalable bigdata mining, the infrastructures provided by the distributed systems can be used. In a distributed environment, there will be a collection of hardware devices that can process large amount of data in a distributed fashion. That is, instead of one computer, variety of computers processes the huge volume of data as partitions. There are different types of bigdata mining tools and platforms for extracting useful information from the massive datasets. Hadoop [2] is the most widely used platform for distributed data processing. Hadoop is the open source implementation of Mapreduce. It is Java based programming framework introduced by Yahoo in 2005. It helps to process extremely large set of data in multiple different nodes in the distributed environment. Hadoop is a part of Apache project. Hadoop has mainly two parts: HDFS and Mapreduce. HDFS is Hadoop Distributed File System which is a disk-based file system that spans across the nodes of a distributed system. All the files stored in the HDFS are automatically divided into blocks and distributed in the local disks of each node. The metadata of the blocks are also stored by HDFS. In Hadoop data is taken from disk and processed so that it is not efficient for the applications that often use iterations. Spark [3] is one of the most recent frameworks for distributed data processing that work with Hadoop. Apache Spark is a fast computing technology that can overcome the disadvantages of Hadoop. It can support applications with iterations. Spark increases the processing speed of data because it uses in-memory computation. Spark can do jobs within seconds that take hours in Hadoop. It is 40x faster than Hadoop. Spark supports main-memory caching and possesses a loop aware scheduler. These features enable users to deploy existing Hadoop application logic in Spark via its Scala API.

Spark SQL is a part of Apache Spark framework for structured data processing. It allows to run SQL like queries in Spark. It is a powerful library to run data analytics even by the non-technical people in an organization. Weka [4] is one of the most popular and comprehensive data mining platforms with a user-friendly interface. One of the major properties of Weka is its portability since it is developed in java. So it can be run on any modern platforms. Weka has different kinds of user interfaces and most using one is explorer. Another advantage of Weka is the easiness to use due to its graphical user interface. It also supports all type of data mining operations. The main disadvantage of Weka in case of bigdata is, Weka can support only sequential single node execution. So that the amount of data that can be processed is limited both by amount of memory in a single node and by sequential execution. RapidMiner [5] is another software platform that provides an integrated environment for both machine learning and data mining. It is used for business and commercial applications as well as for research, education, training, rapid prototyping, and application development. More than 1500 data mining operators are present in RapidMiner. This can be used as a stand-alone application. It also provides efficient multi-layered data view. R [6] is a programming language and software environment for statistical analysis and graphical representation. The R language is widely used among statisticians and data miners. It is mostly used for developing statistical software and data analysis and is freely available under the GNU General Public License. It also provides a wide variety of statistical and graphical techniques. It is highly extensible for adding new techniques. But it can not be used for large data processing. Since there are issues in processing huge amount of data efficient and scalable methods should be generated. This include the combination of various tools of data mining and bigdata processing.

II. LITERATURE REVIEW

The data mining tools are combined to form efficient and high performance tools for bigdata mining. These tools provide better performance than individual tools but arise some inabilities while combining the tools. Mahout [9], a community-based Hadoop-related project, aims to provide scalable data mining algorithm. Its libraries do not provide a general framework for building algorithms. So that, quality of the provided solutions varies significantly that depending on the contributor expertise. That leads to a potential decrease in performance [10]. Mahout mainly focuses on implementing specific algorithms, rather than building execution models for algorithm methods. Radoop [11], is an extension of RapidMiner data mining tool with Hadoop. RapidMiner has a graphical user interface that is used to design work-flows which includes cleaning, loading, mining, and visualization tasks. Radoop provides easy-to-use operators to run the distributed process on Hadoop because eventhough Hadoop provides fault-tolerence and better performance, it’s functionalities can be only exploited by developers due to the lack of a user interface. Radoop scales well with increasing dataset size and number of nodes in the cluster. Radoop suffers from the same performance issues as Mahout. Ricardo [12], is the tool by merging the data mining tool R with distributive frameworks. This system uses declarative scripting language and Hadoop to execute R programs in parallel. This system uses R-syntax that is familiar with many analysts. But, due to the overhead produced by compiling the declarative scripts to low-level MapReduce jobs Ricardo suffers from long execution times. RABID [13], is the combination of R and distributive frameworks, especially Spark. It allows the R users to scale their works in distributive manner and still maintain compatibility of R. It uses the interface familiar to R users and 5x faster than Hadoop. But RABID suffers from performance and portability issues. SparkR [14], gives advantage to R users by providing a light-weight front end to Spark system. The existing R packages can be executed in parallel on partitioned datasets and distributing R computations in to nodes. But this system requires the knowledge of statistical algorithms and basic knowledge of the RDD manipulation techniques. RHIPE [18], also aim to extend R for large scale distributed computations. Ris based on C programming language. Since the distributive frameworks like MapReduce are based on java, a bridging problem wil arise when combining the both. R-code is compiled to C-code which uses the Java Native Interface for execution that reduces portability. RHIPE requires the user to learn Mapreduce. The major problem in combining these data mining tools with distributive frameworks is the bridging overhead between java and other programming languages. This will lead to make a decision to select Weka to combine with distributive frameworks since Weka is also written in java and have an intuitive interface. Weka-Parallel [15], is a modification of Weka and it will allow to perform n-fold cross validations in parallel. Since it added parallelism with Weka, it provides a significant increase over original Weka and reduce the time taken to evaluate any datasets using any classifier. It cannot handle with large amount of data because bottleneck may occur. Weka4WS [16], framework extends Weka to support distributive processing in a grid environment. It uses emerging web services to execute tasks in remote servers but do not support parallelism. In this each server execute independent tasks in their own local data. Wegener et al. [19] introduced a novel system architecture for interactive GUI based data mining of large data algorithms. It merge Weka with Hadoop which combines the user-friendly interface of Weka and distributive processing capability of Hadoop. Since Hadoop is not supporting to iterative algorithms, this system also has an overhead on iterative algorithms. Distributed Weka Spark [4], is a recent framework obtained by combining usability of Weka with processing power of Spark. It is a bigdata mining tool that exploits the distributed power of Spark while remaining the interface of Weka. This system build on the top of Spark so that it provides fast in-memory computations and utilizes both parallel and distributive executions. It can also support to perform iterative algorithms. It is 4x faster, on average, than Hadoop. It is also more user intractable. In Distributed Weka Spark the caching strategies are inefficient.

III. CONCLUSION

Nowadays, the information is growing rapidly due to explosion of technology. There will be data from different sources like internet, sensors, other devices etc. that is of different formats. These data should be managed and processed efficiently. The bigdata mining tools provide better performance today, but they are not much user-friendly. People who have the experience in data analytics can get the advantage. To make it more comfortable for non-technical users, the commonly using bigdata mining tools can be combined with tools that have more intuitive interface. So that, more efficient, scalable and user-friendly platforms can be build. It will provide better performance in processing bigdata than single tools.

REFERENCES

[1] M. Beyer and D. Laney, “The importance of big data: A definition,” Stamford, CT:Gartner.

[2] “Apache Hadoop,” http://hadoop.apache.org/.

[3] “Apache Spark,” https://spark.apache.org/.

[4] Koliopoulos, Aris-Kyriakos, et al. "A Parallel Distributed Weka Framework for Big Data Mining using Spark." 2015 IEEE International Congress on Big Data. IEEE, 2015.

[5] https://en.wikipedia.org/wiki/RapidMiner

[6] https://en.wikipedia.org/wiki/R_(programming_language)

[7] Radoop: Analyzing big data with rapidminer and hadoop, 2011.

[8] S. Das, Y. Sismanis, K. S. Beyer, R. Gemulla, P. J. Haas, and J. McPherson, “Ricardo: Integrating R and Hadoop,” in Intl Conf. on Management of Data, 2010, pp. 987–998.

[9] “Apache Mahout,” http://mahout.apache.org/.

[10] E. R. Sparks, A. Talwalkar, V. Smith, J. Kottalam, X. Pan, J. E. Gonzalez, M. J. Franklin, M. I. Jordan, and T. Kraska, “MLI: an API for distributed machine learning,” ICDM, 2013.

[11] Radoop: Analyzing big data with rapidminer and hadoop, 2011.

[12] S. Das, Y. Sismanis, K. S. Beyer, R. Gemulla, P. J. Haas, and J. McPherson, “Ricardo: Integrating R and Hadoop,” in Intl Conf. on Management of Data, 2010, pp. 987–998.

[13] H. Lin, S. Yang, and S. Midkiff, “RABID: A distributed parallel R for large datasets,” in Congress on Big Data, 2014, pp. 725–732.

[14] “SparkR,” http://amplab-extras.github.io/SparkR-pkg/.

[15] S. Celis and D. Musicant, “Weka-parallel: machine learning in parallel,” Carleton College, Tech. Rep., 2002.

[16] D. Talia, P. Trunfio, and O. Verta, “Weka4WS: A WSRFEnabled Weka Toolkit for Distributed Data Mining on Grids,” Knowledge Discovery in Databases, pp. 309–320, 2005.

[17] M. Prez, A. Snchez, P. Herrero, V. Robles, and J. Pea, “Adapting the Weka Data Mining Toolkit to a Grid Based Environment,” Web Intelligence, pp. 492–497, 2005.

[18] “RHIPE,” https://www.datadr.org/, accessed: 2015-03-03.

[19] D. Wegener, M. Mock, D. Adranale, and S. Wrobel, “Toolkit-Based High-Performance Data Mining of Large Data on MapReduce Clusters,” in ICDM, 2009, pp. 296–301.