Distributed Stochastic Aware Random Forests - Efficient Data Mining for Big Data

Joaquim Assunção, Paulo Fernandes, Lucelene Lopes, Silvio Normey
Computer Science Department
PUCRS University
Porto Alegre, Brazil
{*joaquim.assuncao, paulo.fernandes, lucelene.lopes, silvio.gomez*}*@pucrs.br*

***Abstract*—Some top data mining algorithms, as ensemble classifiers, may be inefficient to very large data set. This paper makes an initial proposal of a distributed ensemble classifier algorithm based on the popular Random Forests for Big Data. The proposed algorithm aims to improve the efficiency of the algorithm by a distributed processing model called MapReduce. At the same time, our proposed algorithm aims to reduce the randomness impact by following an algorithm called Stochastic Aware Random Forests - SARF.**

I. Introduction

The three V's (Velocity, Volume and Variability) of the Big Data imposes three great challenges [1]. The first challenge is how we can use and understand Big Data when it is created in different formats such as video, text, tuples, documents, extensible records and objects. The second challenge is how to capture the most important data and deliver it to the correct people in real time. The third challenge is how to store, to analyze and to understand the data considering its size and the available computational capacity.

Focusing on this third challenge, the Data Mining area offers computational tools to analyze and extract valuable information. Some top algorithms in this area are very efficient in medium and small data sets, but these algorithms does not scale for large and complex data sets. Some of these algorithms are based of random decisions, and, therefore, such algorithms have another potential problem due to the impact of randomness.

For instance, Bagging and Boosting classification algorithms are sensible to random decisions. A recent work [2] proposes an extension to the traditional Random Forests classification algorithm [3], called Stochastic Aware Random Forests (SARF) that reduces the impact of randomness, by producing less correlated models. Such initiative supports the proposal of an algorithm suitable to very large data sets that is globally distributed, hence, suitable to handle Big Data, and additionally capable to generate less correlated distributed models.

Among a myriad of distributed processing techniques, the MapReduce programming model [4] seems the suitable option to handle large data sets stored in distributed machines. MapReduce principle is a two-phase process: the Map phase when the working nodes (workers) are mapped and the local tasks are performed; and the Reduce phase when the results obtained by the workers are gather to perform the final task.

This paper describes the initial proposal of a distributed Data Mining algorithm for Big Data called Distributed Stochastic Aware Random Forests - DSARF. This algorithm is based on two basic components: SARF to build the local models and MapReduce to process large scale data in a distributed (and parallel) way.

II. DSARF - Distributed Stochastic Aware Random Forests

The main objective of DSARF is to provide fast model built according to very large training data set distributively available. Our concern is to provide an algorithm capable to speed up the model creation by handling the training data set in a distributed way. After, it assembles such distributed models in order to provide an unified model to perform the classification of an often large number of instances.

A. Main Components

The MapReduce programming model defines one master node and the remaining nodes are defined as workers. The traditional MapReduce model behaves by assigning Map tasks to the workers, and after completion of these tasks, the workers are assigned to Reduce tasks. Our proposal changes this general programming model by including an exchange of results among the workers at the end of the Map phase.

Specifically (Figure 1), the proposed algorithm assumes that each of n workers have access to its own training data set and the master node starts the mapping phase by: *(i)* assigning a different pseudorandom number generator (PNG) to each worker; and ordering each worker to generate k classification models (decision trees) according to its data set and its assigned PNG. As result, each worker will produce k decision trees, and broadcast their trees to all other workers.

After completion of all workers' Map tasks, all workers will posses nk decision trees. The Reduce phase will correspond to the classification of instances that now can be made by each worker independently. All workers will be able to classify the instances with their nk classification models and furnish the classification result by an appropriated voting algorithm [3].

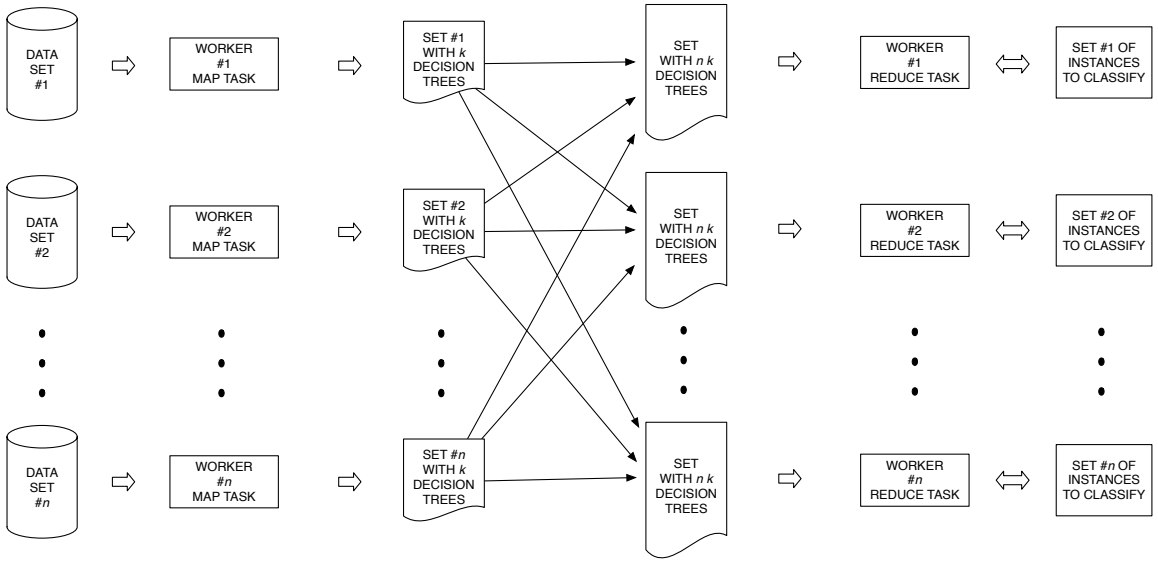

Figure 1. DSARF overview.

B. Implementations Concerns

The overview presented in Figure 1 does not cover some implementation details that are key points to the proposed algorithm. Such details bring some concerns that may be foresee at this proposal state of DSARF algorithm. However, they will be better assessed during the actual DSARF implementation and first experiments over real big data sets.

The first one, already mentioned previously, is to assign different PNG to each worker. This concern will decrease the possible correlation among the random choices (instance and attribute samples) to be taken by the generation of decision trees in each worker. Such concern with randomness impact must also be present inside each worker by choosing different sets of random seeds as stated by Fernandes *et al.* work [2]. Such concern is vital to the quality improvement of the classification performed by DSARF.

Another interesting implementation detail concerns the exchange of generated decision trees among workers. The communication costs to broadcast decision trees may be non negligible, specially considering the time each worker may take to complete its own map tasks. Such concern may affect considerably the time efficiency of DSARF implementation.

Other natural concerns to be taken into account are the ones naturally found in classification algorithms, and, more generally yet, concerns common to any data mining efforts. These concerns are related to the size (number of instances) of each data set associated to each worker. Such concern may demand some kind of pre-processing (homogenization of data set sizes) or even weighting of generated classifiers. It also expected to be concerned with general distribution processing pitfalls, such as load balancing, scheduling, and processing *versus* communication costs.

III. Final Considerations and Next Steps

This work lay down the basic ideas to a novel algorithm to perform distributed data mining of Big Data. The proposed algorithm aims to improve the efficiency by optimizing the time to generate classification models, but also to reduce the impact of randomness, which is a known problem for ensemble classifiers algorithms.

It is important to recall that this paper describes a work in progress that is limited to the proposal of basic components and a brief analysis of foreseeable implementation issues. The natural future steps to this work are the actual implementation, and the experiments over real data. Nonetheless, the original contribution of this paper resides in combining state of the art approaches in data mining and distributed processing areas to tackle Big Data challenges.

References

[1] R. Cattell, "Scalable sql and nosql data stores," *SIGMOD Rec.*, vol. 39, no. 4, pp. 12–27, May 2011. [Online]. Available: http://doi.acm.org/10.1145/1978915.1978919

[2] P. Fernandes, L. Lopes, S. Normey, and D. Ruiz, "Stochastic aware random forests - a variation less impacted by randomness," in *Proc. of the Twenty Sixth Int. FLAIRS Conf. (FLAIRS 2013)*, C. Boonthum-Denecke and G. M. Youngblood, Eds. AAAI Press, 2013, pp. 146–149.

[3] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1010933404324

[4] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," in *Proc. of the 6th Symposium on Opearting Systems Design & Implementation*. Berkeley, CA, USA: USENIX Association, 2004, pp. 10–10. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251254.1251264