Big Data Research ••• (••••) •••–•••

Contents lists available at ScienceDirect

Big Data Research

www.elsevier.com/locate/bdr

Random Forests for Big Data

Robin Genuer [a], Jean-Michel Poggi [b], Christine Tuleau-Malot [c], Nathalie Villa-Vialaneix [d]

[a] ISPED, INSERM U-1219, Univ. Bordeaux & INRIA, SISTM team, France
[b] LMO, Univ. Paris-Sud Orsay & Univ. Paris Descartes, France
[c] Université Côte d'Azur, CNRS, LJAD, France
[d] MIAT, Université de Toulouse, INRA, France

ARTICLE INFO

Article history:
Received 2 November 2016
Received in revised form 2 June 2017
Accepted 7 July 2017
Available online xxxx

Keywords:
Random forest
Big Data
Parallel computing
Bag of little bootstraps
On-line learning
R

ABSTRACT

Big Data is one of the major challenges of statistical science and has numerous consequences from algorithmic and theoretical viewpoints. Big Data always involve massive data but they also often include online data and data heterogeneity. Recently some statistical methods have been adapted to process Big Data, like linear regression models, clustering methods and bootstrapping schemes. Based on decision trees combined with aggregation and bootstrap ideas, random forests were introduced by Breiman in 2001. They are a powerful nonparametric statistical method allowing to consider in a single and versatile framework regression problems, as well as two-class and multi-class classification problems. Focusing on classification problems, this paper proposes a selective review of available proposals that deal with scaling random forests to Big Data problems. These proposals rely on parallel environments or on online adaptations of random forests. We also describe how out-of-bag error is addressed in these methods. Then, we formulate various remarks for random forests in the Big Data context. Finally, we experiment five variants on two massive datasets (15 and 120 millions of observations), a simulated one as well as real world data. One variant relies on subsampling while three others are related to parallel implementations of random forests and involve either various adaptations of bootstrap to Big Data or to "divide-and-conquer" approaches. The fifth variant is related to online learning of random forests. These numerical experiments lead to highlight the relative performance of the different variants, as well as some of their limitations.

1. Introduction

1.1. Statistics in the Big Data world

Big Data is one of the major challenges of statistical science and a lot of recent references start to think about the numerous consequences of this new context from the algorithmic viewpoint and for the theoretical implications of this new framework [1–3]. Big Data always involve massive data: for instance, Thusoo et al. [4] indicate that Facebook© had more than 21 PB of data in 2010. They also often include data streams and data heterogeneity [5]. On a practical point of view, they are characterized by the fact that data are frequently not structured data, properly indexed in a database. Thus, simple queries cannot be easily performed on such data. These features lead to the famous three V (Volume, Velocity and Variety) highlighted by the Gartner, Inc., the advisory company about information technology research.[1] In the most extreme situations, data can even have a size too large to fit in a single computer memory. Then data are distributed among several computers. In this case, the distribution of the data is managed using specific frameworks dedicated to shared storage computing environments, such as Hadoop.[2]

For statistical science, the problem posed by this large amount of data is twofold: first, as many statistical procedures have devoted few attention to computational runtimes, they can take too long to provide results in an acceptable time. When dealing with complex tasks, such as learning of a prediction model or complex exploratory analysis, this issue can occur even if the dataset would be considered of a moderate size for other simpler tasks. Also, as

E-mail addresses: robin.genuer@u-bordeaux.fr (R. Genuer), jean-michel.poggi@math.u-psud.fr (J.-M. Poggi), malot@unice.fr (C. Tuleau-Malot), nathalie.villa-vialaneix@inra.fr (N. Villa-Vialaneix).

http://dx.doi.org/10.1016/j.bdr.2017.07.003
2214-5796/

[1] http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf.

[2] Hadoop, http://hadoop.apache.org is a software environment programmed in Java, which contains a file system for distributed architectures (HDFS: Hadoop Distributed File System) and dedicated programs for data analysis in parallel environments. It has been developed from GoogleFS, The Google File System.

pointed out in [6], the notion of Big Data depends itself on the available computing resources. This is especially true when relying on the free statistical software R [7], massively used in the statistical community, which capabilities are strictly limited by RAM. In this case, data can be considered as "large" if their size exceeds 20% of RAM and as "massive" if it exceeds 50% of RAM, because this amount of data strongly limits the available memory for learning the statistical model itself. For memory demanding statistical methods and implementations, the RAM can even be overloaded with datasets occupying a very moderate amount of the RAM. As pointed out in [3], in the near future, statistics will have to deal with problems of scale and computational complexity to remain relevant. In particular, the collaboration between statisticians and computer scientists is needed to control runtimes that will maintain the statistical procedures usable on large-scale data while ensuring good statistical properties.

1.2. Main approaches to scale statistical methods

Recently, some statistical methods have been adapted to process Big Data, including linear regression models, clustering methods and bootstrapping schemes [8,9]. The main proposed strategies are based on i) *subsampling*, ii) *divide and conquer approach*, iii) *algorithm weakening* and iv) *online* updates.

Subsampling is probably the simplest way to handle large datasets. It is proved efficient to approximate spectral analysis of large matrices using approximate decomposition, such as the Nyström algorithm [10]. It is also a valuable strategy to produce approximate bootstrap scheme [11]. Simple random sampling often produces a representative enough subsample but can be hard to obtain if data are distributed over different computers and the subsample itself has to be built in parallel: online subsampling strategies allowing stratified sampling are presented in [12] and can overcome this problem. Improved subsampling strategies can also be designed, like the core-set strategy used for clustering problem in [13], that extracts a relevant small set of points to perform approximate clustering efficiently. Finally, an alternative to alleviate the impact of the subsampling without the need to use sophisticated subsampling schemes is to perform several subsamplings and to combine the different results [14].

Divide and conquer approach consists in splitting the problem into several smaller problems and in gathering the different results in a final step. This approach is the one followed in the popular MapReduce programming paradigm [15]. Most of the time, the combination is based on a simple aggregation or averaging of the different results but this simple method might lead to biased estimations in some statistical models, as simple as a linear model. Solutions include re-weighting of the different results [16].

Algorithm weakening is a very different approach, designed for methods based on convex optimization problems [17]. This method explicitly treats the trade-off between computational time and statistical accuracy using a hierarchy of relaxed optimization problems with increasing complexity.

Finally, online approaches update the results with sequential steps, each having a low computational cost. It very often requires a specific rewriting of the method to single out the specific contribution of a given observation to the method. In this case, the online update is strictly equivalent to the processing of the whole dataset but with a reduced computational time [18]. However, in most cases, such an equivalence can not be obtained and a modification of the original method is needed to allow online updates [19].

It has to be noted that only a few papers really address the question of the difference between the "small data" standard framework compared to the Big Data in terms of statistical accuracy when approximate versions of the original approach are used to deal with the large sample size. Noticeable exceptions are the article of Kleiner et al. [11] who prove that their "Bag of Little Bootstraps" method is statistically equivalent to the standard bootstrap, the article of Chen and Xie [16] who demonstrate asymptotic equivalence of their "divide-and-conquer" based estimator with the estimator based on all data in the setting of linear regression and the article of Yan et al. [10] who show that the mis-clustering rate of their subsampling approach, compared to what would have been obtained with a direct approach on the whole dataset, converges to zero when the subsample size grows (in an unsupervised setting).

1.3. Random forests and Big Data

Based on decision trees and combined with aggregation and bootstrap ideas, random forests (abbreviated RF in the sequel), were introduced by Breiman [20]. They are a powerful nonparametric statistical method allowing to consider regression problems as well as two-class and multi-class classification problems, in a single and versatile framework. The consistency of RF has recently been proved by Scornet et al. [21], to cite the most recent result. On a practical point of view, RF are widely used [22,23] and exhibit extremely high performance with only a few parameters to tune. Since RF are based on the definition of several independent trees, it is thus straightforward to obtain a parallel and faster implementation of the RF method, in which many trees are built in parallel on different cores. However, direct parallel training of the trees might be intractable in practice, due to the large size of the bootstrap samples. As RF also include intensive resampling, it is to consider adapted bootstrapping schemes for the massive online context, in addition to parallel processing.

Even if the method has already been adapted and implemented to handle Big Data in various distributed environments (see, for instance, the libraries Mahout[3] or MLib, the latter for the distributed framework Spark,[4] among others), a lot of questions remain open. In this paper, we do not seek to make an exhaustive description of the various implementations of RF in scalable environments but we will highlight some problems posed to RF by the Big Data framework, describe several standard strategies that can be used and discuss their main features, drawbacks and differences with the original approach. We finally experiment five variants on two massive datasets (15 and 120 millions of observations), a simulated one as well as real world data. One variant relies on subsampling while three others are related to parallel implementations of random forests and involve either various adaptations of bootstrap to Big Data or "divide-and-conquer" approaches. The fifth variant relates to online learning of RF. To the best of our knowledge, no weakening strategy has been developed for RF.

Since the free statistical software R [7] is *de facto* the esperanto in the statistical community, and since the most flexible and widely used programs for designing random forests are also available in R, we have adopted it for numerical experiments as much as possible. More precisely, the R package **randomForest**, implementing the original RF algorithm using Breiman and Cutler's Fortran code, contains many options together with a detailed documentation. It has then been used in almost all experiments. The only exception is for online RF for which no implementation in R is available. A python library was used, as an alternative tool in order to provide a comparison of online learning with the alternative Big Data variants.

The paper is organized as follows. After this introduction, we briefly recall some basic facts about RF in Section 2. Then, Section 3 is focused on strategies for scaling random forests to Big

[3] https://mahout.apache.org.
[4] https://spark.apache.org/mllib.