HAL open science

Impact of subsampling and tree depth on random forests

Roxane Duroux, Erwan Scornet

To cite this version:

Roxane Duroux, Erwan Scornet. Impact of subsampling and tree depth on random forests. ESAIM: Probability and Statistics, 2018, 22, pp.96-128. 10.1051/ps/2018008 . hal-02925334

HAL Id: hal-02925334

https://hal.science/hal-02925334v1

Submitted on 29 Aug 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ESAIM: PS 22 (2018) 96–128
https://doi.org/10.1051/ps/2018008

ESAIM: Probability and Statistics
www.esaim-ps.org

IMPACT OF SUBSAMPLING AND TREE DEPTH ON RANDOM FORESTS

Roxane Duroux[1] and Erwan Scornet[2,*]

Abstract. Random forests are ensemble learning methods introduced by Breiman [Mach. Learn. **45** (2001) 5–32] that operate by averaging several decision trees built on a randomly selected subspace of the data set. Despite their widespread use in practice, the respective roles of the different mechanisms at work in Breiman's forests are not yet fully understood, neither is the tuning of the corresponding parameters. In this paper, we study the influence of two parameters, namely the subsampling rate and the tree depth, on Breiman's forests performance. More precisely, we prove that quantile forests (a specific type of random forests) based on subsampling and quantile forests whose tree construction is terminated early have similar performances, as long as their respective parameters (subsampling rate and tree depth) are well chosen. Moreover, experiments show that a proper tuning of these parameters leads in most cases to an improvement of Breiman's original forests in terms of mean squared error.

Mathematics Subject Classification. 62G05, 62G20

Received October 26, 2017. Accepted February 10, 2018.

1. Introduction

Random forests are a class of learning algorithms used to solve pattern recognition problems. As ensemble methods, they grow many base learners (in this case, decision trees) and aggregate them to predict. Building several different trees from a single data set requires to randomize the tree building process by, for example, resampling the data set. Thus, there exists a large variety of random forests, depending on how trees are designed and how randomization is introduced in the whole procedure.

Random forests were first introduced by Breiman [5]. They proceed by growing trees based on CART procedure (Classification And Regression Trees) [6], and randomizing both the training set and the splitting variables. Breiman's Breiman [5] random forests have been under active investigation during the last decade mainly because of their good practical performance and their ability to handle high-dimensional data sets. They are acknowledged to be state-of-the-art methods in fields such as genomics [18] and pattern recognition [19], just to name a few. The ease of the implementation of random forests algorithms is one of their key strengths and has greatly contributed to their widespread use. Indeed, a proper tuning of the different parameters of the algorithm is not mandatory to obtain a plausible prediction, making random forests a turn-key solution to deal with large, heterogeneous data sets in many fields (see, *e.g.*, [12]).

Keywords and phrases: Random forests, randomization, parameter tuning, subsampling, tree depth.

[1] Sorbonne Universités, UPMC Univ Paris 06, 75005 Paris, France.
[2] Centre de Mathématiques Appliquées, École polytechnique, UMR 7641, 91128 Palaiseau, France.

* Corresponding author: scornet@polytechnique.edu

Article published by EDP Sciences
© EDP Sciences, SMAI 2018

Several authors studied the influence of the parameters on random forests accuracy. For example, the number M of trees in the forests has been thoroughly investigated by Díaz-Uriarte and de Andrés [11] and Genuer *et al.* [14]. It is easy to see that the computational cost for inducing a forest increases linearly with M, so a good choice for M results from a trade-off between computational complexity and accuracy (M must be large enough for predictions to be stable). Díaz-Uriarte and de Andrés [11] argued that in micro-array classification problems, the particular value of M is irrelevant, assuming that M is large enough (typically over 500). Several recent studies provided theoretical guidance for choosing M. For instance, Scornet [21] gives an explicit upper bound on forest accuracy which depends on the number of trees; Mentch and Hooker [17] and Wager and Athey [27] focus on the pointwise distribution of random forest estimate and establish a central limit theorem for random forests prediction together with a method to estimate their variance. All in all, the role of the number M of trees on the forest prediction is broadly understood.

Breiman's forests depend on three other parameters: the number a_n of data points selected to build each tree, the number m_{try} of preselected variables along which the best split is chosen, and the minimum number `nodesize` of data points in each leaf of each tree. The subsampling rate a_n/n controls the percentage of observations used to create each tree estimate. It turns out that the subsampling step is a key element to design consistent forests based on inconsistent trees [21]. The parameter m_{try} regulates how much randomization is injected into the splitting procedure. A critical case (also known as bagging) is reached by setting $m_{\text{try}} = d$: the best split is chosen among all possible variables, therefore no additional randomness is put on the splitting procedure. The effect of m_{try} was discussed in detail by Díaz-Uriarte and de Andrés [11] and Genuer *et al.* [14] who claimed that the default value ($m_{\text{try}} = d/3$, where d is the number of input variables) is either optimal or too small, therefore leading to no global understanding of this parameter. The story is roughly the same regarding the parameter `nodesize` [11], which governs the tree depth: small `nodesize` values lead to a deep tree whose terminal cells result from a large number of consecutive splits. Unfortunately, there are no theoretical guarantees to support the default values of parameters or any of data-driven tuning process proposed in the literature.

Our objective in this paper is two-fold: (i) to provide a theoretical framework to analyze jointly the influence of the subsampling step and the tree depth (which can be parametrized by either `nodesize` or `maxnode`) on quantile forests, close in spirit to Breiman's original algorithm; (ii) to implement several experiments to see how our theoretical findings can be extended to Breiman's forests. The paper is organized as follows. Section 2 is devoted to notations and presents Breiman's random forests algorithm. To carry out a theoretical analysis of the subsample size and the tree depth, we study in Section 3 quantile forests. We establish an upper bound for the risk of quantile forests and by doing so, we highlight the fact that subsampling and tree depth have similar influence on quantile forest predictions. We first present and discuss results for median forests which are a good starting point for understanding the general theory of quantile forests. To determine the range of validity of these results on Breiman's forests, we implement numerous experiments in Section 4. These experiments tend to indicate that subsample size and tree depth are two faces of the same coin, for Breiman's forests (as well as for quantile forests). Results are discussed in Section 5 and proofs are postponed to Section 6.

2. First definitions

2.1. General framework

In this paper, we consider a training sample $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ of $[0,1]^d \times \mathbb{R}$-valued independent and identically distributed observations of a random pair $(\mathbf{X}, Y)$, where $\mathbb{E}[Y^2] < \infty$. We denote by $\mathbf{X} = (X^{(1)}, \ldots, X^{(d)})$ the input variables and by Y the response variable. We wish to estimate the regression function $m(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$. In this context, we want to build an estimate $m_n : [0,1]^d \to \mathbb{R}$ of m, based on the data set $\mathcal{D}_n$.

Random forests are regression methods based on a collection of M randomized trees. A decision tree is an estimate that recursively split the input space in order to make a prediction (see Fig. 1). Instances of decision trees such as Breiman's trees and median trees are described below. Mathematically speaking, a tree estimate

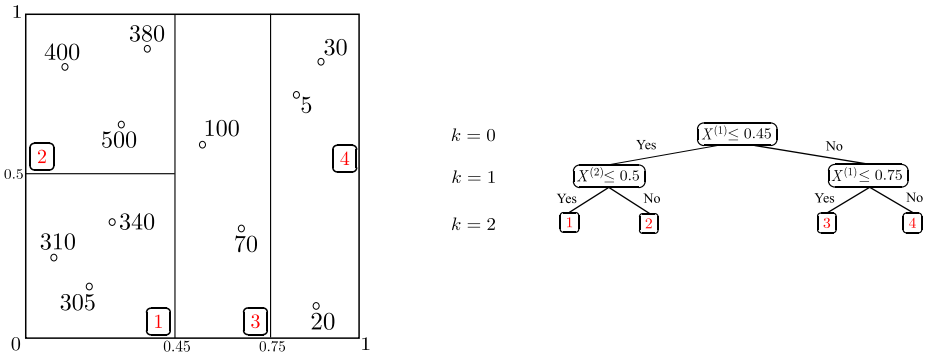


FIGURE 1. A decision tree of depth $k = 2$ in dimension $d = 2$.

satisfies, for all $\mathbf{x} \in [0,1]^d$,

$$m_n(\mathbf{x}, \mathcal{D}_n) = \sum_{i=1}^{n} Y_i \frac{\mathbb{1}_{\mathbf{X}_i \in A_n(\mathbf{x}, \mathcal{D}_n)}}{N_n(\mathbf{x}, \mathcal{D}_n)},$$

where $A_n(\mathbf{x}, \mathcal{D}_n)$ is the terminal cell containing $\mathbf{x}$ and $N_n(\mathbf{x}, \mathcal{D}_n)$ is the number of observations in this cell.

Now, let us consider some randomization of the tree construction: additional randomness, parametrized by Θ, independent of the training set, is introduced in the building process. In practice, the variable Θ can be used to resample the data set or to select the candidate directions or positions for splitting. The corresponding tree estimate at a given point $\mathbf{x} \in [0,1]^d$ then writes

$$m_n(\mathbf{x}, \Theta, \mathcal{D}_n) = \sum_{i=1}^{n} Y_i \frac{\mathbb{1}_{\mathbf{X}_i \in A_n(\mathbf{x}, \Theta, \mathcal{D}_n)}}{N_n(\mathbf{x}, \Theta, \mathcal{D}_n)},$$

where the notations are the same as above. Since we are interested in a collection of trees, we let $\Theta_1, \ldots, \Theta_M$ be independent random variables, distributed as the generic random variable Θ, independent of the sample $\mathcal{D}_n$. The predictions of the M randomized trees are then averaged to obtain the random forest prediction

$$m_{M,n}(\mathbf{x}, \Theta_1, \ldots, \Theta_M, \mathcal{D}_n) = \frac{1}{M} \sum_{j=1}^{M} m_n(\mathbf{x}, \Theta_j, \mathcal{D}_n). \tag{2.1}$$

By the law of large numbers, for any fixed $\mathbf{x}$, conditional on $\mathcal{D}_n$, the finite forest estimate tends to the infinite forest estimate

$$m_{\infty,n}(\mathbf{x}, \mathcal{D}_n) = \mathbb{E}_\Theta \left[m_n(\mathbf{x}, \Theta, \mathcal{D}_n) \right],$$

where $\mathbb{E}_\Theta$ denotes the expectation with respect to the random variable Θ only. For the sake of simplicity, we will omit the explicit dependence in the data set $\mathcal{D}_n$ in all formulas. Thus, $m_{\infty,n}(\mathbf{x}, \mathcal{D}_n)$ will simply be written as $m_{\infty,n}(\mathbf{x})$. Since we carry out our analysis within the $\mathbb{L}^2$ regression estimation framework, we say that $m_{\infty,n}$ is consistent if its risk, $\mathbb{E}[m_{\infty,n}(\mathbf{X}) - m(\mathbf{X})]^2$, tends to zero, as n goes to infinity.

2.2. Breiman's forests

Breiman's [5] forest is one of the most used random forest algorithms. In Breiman's forests, each node of a single tree is associated with a hyper-rectangular cell included in $[0,1]^d$. The root of the tree is $[0,1]^d$ itself and

at each step of the tree construction, a node (or equivalently its corresponding cell) is split in two parts. The terminal nodes (or leaves), taken together, form a partition of $[0,1]^d$. In details, the algorithm works as follows:

1. Grow M trees as follows:
   (a) Prior to the jth tree construction, select uniformly with replacement, a_n data points among $\mathcal{D}_n$. Only these a_n observations are used in the tree construction.
   (b) Consider the cell $[0,1]^d$.
   (c) Select uniformly without replacement m_{try} coordinates among $\{1,\ldots,d\}$.
   (d) Select the split minimizing the CART-split criterion (see [6, 22] for details) along the pre-selected m_{try} directions.
   (e) Cut the cell at the selected split.
   (f) Repeat (c)–(e) for the two resulting cells until each cell of the tree contains less than `nodesize` observations.
   (g) For a query point $\mathbf{x}$, the jth tree outputs the average $m_n(\mathbf{x},\Theta_j)$ of the Y_i falling into the same cell as $\mathbf{x}$.
2. For a query point $\mathbf{x}$, Breiman's forest outputs the average $m_{M,n}(\mathbf{x},\Theta_1,\ldots,\Theta_M)$ of the predictions given by the M trees.

The whole procedure depends on four parameters: the number M of trees, the number a_n of resampled data points in each tree, the number m_{try} of pre-selected directions for splitting, and the maximum number `nodesize` of observations in each leaf. By default in the `R` package `randomForest`, M is set to 500, $a_n = n$ (bootstrap samples are used to build each tree), $m_{\text{try}} = d/3$ and `nodesize` $= 5$.

Let us emphasize that the randomization at work in Breiman's forests is composed of the sampling of data points and the choice of eligible directions for splitting in each cell. Thus, for the jth tree, the variable Θ_j is nothing but a large vector containing first the indices of observations selected to build the jth tree and then the variables eligible for splitting for each cell. Writing explicitly the components of Θ demands a lot of notations and is not vital for our analysis. Hence, we will not fall into this pitfall.

Note that selecting the split that minimizes the CART-split criterion is equivalent to selecting the split such that the two resulting cells have a minimal (empirical) variance (regarding the Y_i falling into each of the two cells).

3. Theoretical results

The numerous mechanisms at work in Breiman's forests, such as the resampling step, the CART-criterion and the trees aggregation, make the whole procedure difficult to theoretically analyze. Most attempts to understand random forest algorithms (see *e.g.*, [4, 8, 15]) have focused on simplified procedures, ignoring the resampling step and/or replacing the CART-split criterion by a data independent procedure more amenable to analysis. On the other hand, recent studies try to dissect the original Breiman's algorithm in order to prove its asymptotic normality [17, 27] or its consistency [22]. When studying the original algorithm, one faces its intrinsic complexity thus requiring high-level mathematics to prove insightful – but rough – results.

In order to provide theoretical guarantees on the parameter values of random forests, we focus in this section on a simplified random forest called quantile forests. Quantile forests construction depends only on the $\mathbf{X}_i$'s making them a good tradeoff between the complexity of Breiman's Breiman [5] forests and the simplicity of totally non-adaptive forests, whose construction is independent of the data set.

The interest of studying quantile forests, compared to forests whose construction is independent of both $\mathbf{X}_i$ and Y_i, lies in the fact that they can benefit from subsampling. Indeed, Scornet [21] proves that subsampled quantile forests are consistent even if each quantile tree in the forest is not, therefore highlighting the nice effect of subsampling on quantile forests. Besides, quantile forests can be tuned such that each leaf of each tree contains exactly one observation, thus being close to Breiman's forests.

We start by analyzing a specific instance of quantile forests called median forests (see, *e.g.*, [3] for details on median tree) and then extend the analysis to general quantile forests.

3.1. Median forests

We now describe the construction of median forest. In the spirit of Breiman's [5] algorithm, before growing each tree, data are subsampled, that is a_n points ($a_n < n$) are selected, without replacement. Then, each split is performed on an empirical median along a coordinate, chosen uniformly at random among the d coordinates. Recall that the median of n real valued random variables $X_1, \ldots, X_n$ is defined as the only $X_{(\ell)}$ satisfying $F_n(X_{(\ell-1)}) \leq 1/2 < F_n(X_{(\ell)})$, where the $X_{(i)}$'s are ordered increasingly and F_n is the empirical distribution function of X. Note that data points on which splits are performed are not sent down to the resulting cells. This is done to ensure that data points are uniformly distributed on the resulting cells (otherwise, there would be at least one data point on the edge of a resulting cell, and thus the data points distribution would not be uniform on this cell). Finally, the algorithm stops when each cell has been cut exactly k_n times, *i.e.*, `nodesize`$= \lceil a_n 2^{-k_n} \rceil$. In other words, each terminal node is nothing but the result of k_n sequential splits of the root $[0,1]^d$ of the tree. The corresponding tree structure is therefore a full binary tree of level k_n. The parameter k_n, also known as the tree depth, is assumed to verify $a_n 2^{-k_n} \geq 4$. The overall construction process is detailed below.

1. Grow M trees as follows:
   (a) Prior to the jth tree construction, select uniformly without replacement, a_n data points among $\mathcal{D}_n$. Only these a_n observations are used in the tree construction.
   (b) Consider the cell $[0,1]^d$.
   (c) Select uniformly one coordinate j among $\{1, \ldots, d\}$ without replacement.
   (d) Cut the cell at the empirical median of the $X_i^{(j)}$ ($1 \leq i \leq n$) falling into the cell, along the preselected direction.
   (e) For each of the two resulting cells, repeat (c)–(d) if the cell has been cut strictly less than k_n times.
   (f) For a query point $\mathbf{x}$, the jth tree outputs the average $m(\mathbf{x}, \Theta_j)$ of the Y_i falling into the same cell as $\mathbf{x}$.
2. For a query point $\mathbf{x}$, the median forest outputs the average $m_{M,n}(\mathbf{x}, \Theta_1, \ldots, \Theta_M)$ of the predictions given by the M trees.

Note that, as many previous works (see, *e.g.*, [17, 27]), our analysis strongly relies on the fact that subsampling is done without replacement. Indeed, if subsampling were to be done with replacement, the distribution of each subsample would not be the same as the distribution of the whole sample (since we assume that $\mathbf{X}$ has a density with respect to Lebesgue measure), and the mathematical analysis would turn out to be much more complicated. However, we will see in Section 4 that empirically, there seems to be no differences between bootstrapping and subsampling without replacement for Breiman's forests, if the subsample size is well chosen.

3.2. Main theorem

We will focus in this section on the risk of the infinite median forest

$$m_{\infty,n}(\mathbf{x}) = \mathbb{E}[m_n(\mathbf{x}, \Theta)],$$

obtained by setting $M \to \infty$. It may seem quite restrictive at first sight because we are interested in the finite median forest $m_{M,n}$ which can be implemented (contrary to the infinite median forest). Fortunately, inequality (3.3) in Theorem 3.1 can be extended to finite median forests by using Theorem 3.3 in Scornet [21] if

$$M \geq C n^{\frac{-\ln \beta}{\ln 2 - \ln \beta}},$$

for some constant $C > 0$ where $\beta = 1 - 3/(4d)$. Thus, the rate of consistency for infinite forests and the analysis provided below are still valid for finite forests.

Theorem 3.1. *Assume that $Y = m(\mathbf{X}) + \varepsilon$, where the noise ε satisfies, for all $\mathbf{x} \in [0,1]^d$, $\mathbb{E}[\varepsilon|\mathbf{X} = \mathbf{x}] = 0$ and $\mathbb{V}[\varepsilon|\mathbf{X} = \mathbf{x}] \leq \sigma^2 < \infty$. Moreover, $\mathbf{X}$ is uniformly distributed on $[0,1]^d$ and m is L-Lipschitz continuous. Then,*

for all n, for all $\mathbf{x} \in [0,1]^d$,

$$\mathbb{E}\big[m_{\infty,n}(\mathbf{x}) - m(\mathbf{x})\big]^2 \leq 2\sigma^2 \frac{2^k}{n} + dL^2 C_1 \left(1 - \frac{3}{4d}\right)^k. \tag{3.1}$$

In addition, let $\beta = 1 - 3/(4d)$. The right-hand side is minimal for

$$k_n = \frac{1}{\ln 2 - \ln \beta}\Big[\ln(n) + C_2\Big], \tag{3.2}$$

under the condition that $a_n \geq C_3 n^{\frac{\ln 2}{\ln 2 - \ln \beta}}$. For these choices of k_n and a_n, we have

$$\mathbb{E}\big[m_{\infty,n}(\mathbf{x}) - m(\mathbf{x})\big]^2 \leq C_4 n^{\frac{\ln \beta}{\ln 2 - \ln \beta}}. \tag{3.3}$$

Equation (3.1) stems from the estimation/approximation error decomposition of median forests. The first term in equation (3.1) corresponds to the estimation error of the forest as in Biau [2] or Arlot and Genuer [1] whereas the second term is the approximation error of the forest, which decreases exponentially in k. Note that this decomposition is consistent with the existing literature on random forests. Two common assumptions to prove consistency of simplified random forests are $n/2^k \to \infty$ and $k \to \infty$, which respectively control the estimation and approximation of the forest. According to Theorem 3.1, making these assumptions for median forests results in their consistency.

Note that the estimation error of a single tree grown with a_n observations is of order $2^k/a_n$. Thus, because of the subsampling step (*i.e.*, since $a_n < n$), the estimation error of median forests $2^k/n$ is smaller than that of a single tree. The variance reduction of random forests is a well-known property, already noticed by Genuer [13] for a totally nonadaptive forest, and by Scornet [21] in the case of median forests. In our case, we exhibit an explicit bound on the forest variance, which allows us to precisely compare it to the individual tree variance therefore highlighting a first benefit of median forests over singular trees.

Now, let us consider the second term in equation (3.1). In the levels close to the root, a split is close to the center of a side of a cell (since $\mathbf{X}$ is uniformly distributed over $[0,1]^d$). Thus, for all k small enough, the approximation error of median forests should be close to that of centred forests studied by Biau [2]. Surprisingly, the rate of consistency of median forests is faster than that of centred forest established in Biau [2], which is equal to

$$\mathbb{E}\big[m^{cc}_{\infty,n}(\mathbf{X}) - m(\mathbf{X})\big]^2 \leq C n^{\frac{-3}{4d\ln 2 + 3}}, \tag{3.4}$$

where $m^{cc}_{\infty,n}$ stands for the centred forest estimate. A close inspection of the proof of Proposition 2.2 in Biau [2] shows that it can be easily adapted to match the (lower) upper bound in Theorem 3.1.

Noteworthy, the fact that the upper bound (3.3) is sharper than (3.4) appears to be important in the case where $d = 1$. In that case, according to Theorem 3.1, for all n, for all $\mathbf{x} \in [0,1]^d$,

$$\mathbb{E}\big[m_{\infty,n}(\mathbf{x}) - m(\mathbf{x})\big]^2 \leq C n^{-2/3},$$

which is the minimax rate over the class of Lipschitz functions (see, *e.g.*, [23, 24]). This was to be expected since, in dimension one, median random forests are simply a median tree which is known to reach minimax rate [10]. Unfortunately, for $d = 1$, the centred forest bound (3.4) turns out to be suboptimal since it results in

$$\mathbb{E}\big[m^{cc}_{\infty,n}(\mathbf{X}) - m(\mathbf{X})\big]^2 \leq C n^{\frac{-3}{4\ln 2 + 3}}. \tag{3.5}$$

Finally, note that Theorem 3.1 provides the pointwise rate of consistency of $m_{\infty,n}$. Thus, median forests are pointwise consistent, which may not be the case of Breiman forests at some very particular query point $\mathbf{x}$ close to the edges of $[0,1]^d$ (see Fig. 3 and the discussion below in [26]).

Theorem 3.1 allows us to derive rates of consistency for two particular forests: the partially grown median forest, where no subsampling is performed prior to building each tree, and the fully grown median forest, where each leaf contains a small number of points. Corollary 3.1 deals with partially grown median forests, also called small-tree median forests.

Corollary 3.1 (Small-tree median forests)**.** *Let $\beta = 1 - 3/(4d)$. Consider a median forest without subsampling (i.e., $a_n = n$) and such that the parameter k_n satisfies (3.2). Under the same assumptions as in Theorem 3.1, we have, for all n, for all $\mathbf{x} \in [0,1]^d$,*

$$\mathbb{E}\big[m_{\infty,n}(\mathbf{x}) - m(\mathbf{x})\big]^2 \leq C_4 n^{\frac{\ln \beta}{\ln 2 - \ln \beta}}.$$

Up to an approximation, Corollary 3.1 is the counterpart of Theorem 2.2 in Biau [2] but tailored for median forests, in which each tree is built using the full training sample and each split is always performed at the median of the selected variable. Indeed, the rate of consistency provided in Theorem 2.2 for centred forests and that of Corollary 3.1 for median forests are similar. Note that, for both forests, the optimal depth k_n of each tree is the same.

Corollary 3.2 handles the case of fully grown median forests, that is forests which contain a small number of points in each leaf. Indeed, note that since $k_n = \log_2(a_n) - 2$, the number of observations in each leaf varies between 4 and 8.

Corollary 3.2 (Fully grown median forest)**.** *Let $\beta = 1 - 3/(4d)$. Consider a fully grown median forest whose parameters k_n and a_n satisfy $k_n = \log_2(a_n) - 2$. Under the same assumptions as in Theorem 3.1, the optimal choice for a_n that minimizes the $\mathbb{L}^2$ error in (3.1) is then given by (3.2), that is*

$$a_n = C_3 n^{\frac{\ln 2}{\ln 2 - \ln \beta}}.$$

In this case, for all n, for all $\mathbf{x} \in [0,1]^d$,

$$\mathbb{E}\big[m_{\infty,n}(\mathbf{x}) - m(\mathbf{x})\big]^2 \leq C_4 n^{\frac{\ln \beta}{\ln 2 - \ln \beta}}.$$

Whereas each individual tree in the fully developed median forest is inconsistent (since each leaf contains a small number of points), the whole forest is consistent and its rate of consistency is provided by Corollary 3.2. Besides, Corollary 3.2 provides us with the optimal subsampling size for fully developed median forests.

Provided a proper parameter tuning, partially grown median forests without subsampling and fully grown median forests (with subsampling) have similar performance. A close look at Theorem 3.1 shows that the subsampling size has no effect on the performance, provided it is large enough. The parameter of real importance is the tree depth k_n. Thus, fixing k_n as in equation (3.2), and by varying the subsampling rate a_n/n one can obtain random forests whose trees are more-or-less deep, all satisfying the optimal bound in Theorem 3.1. In this way, Corollaries 3.1 and 3.2 are simply two particular examples of such forests.

Although our analysis sheds some light on the role of subsampling and tree depth, the statistical performance of median forests does not allow us to choose between small-tree forests and subsampled forests. Interestingly, note that these two types of random forests can be used in two different contexts. If one wants to obtain fast predictions, then subsampled forests, as described in Corollary 3.2, are to be preferred since their computational time is lower than small-tree forests (described in Cor. 3.1). However, if one wants to build more accurate predictions, small-tree random forests have to be chosen since the recursive random forest procedure allows to build several forests of different tree depths in one run, therefore allowing to select the best model among these forests.

3.3. Quantile forests

Although median forests can have a small number of observations in each cell (always larger than one) and are in that sense close to Breiman's forests, the splitting procedure in which splits are performed at the empirical median can seem restrictive compared to Breiman's forests where the split location is optimized according to the CART-criterion.

To circumvent this issue, we consider α-quantile forests whose construction is very similar to median forests but whose splits are not performed at the empirical median but at any empirical quantile of order belonging to the interval $(\alpha, 1-\alpha)$, where $\alpha \in (0, 1/2)$ is a parameter of the algorithm. Note that, even if there is a large choice of possibilities for split locations, we do not allow splits to depend on the outputs Y_i.

As for median forests, the algorithm stops when each cell has been cut exactly k_n times, where we forced the parameter k_n (tree depth) to verify $a_n \alpha^{k_n} \geq 4$, so that terminal nodes are always non empty. Theorem 3.2 and Corollary 3.3 are generalization of Theorem 3.1 and Corollaries 3.1 and 3.2.

Theorem 3.2. *Assume that $Y = m(\mathbf{X}) + \varepsilon$, where the noise ε satisfies, for all $\mathbf{x} \in [0,1]^d$, $\mathbb{E}[\varepsilon|\mathbf{X} = \mathbf{x}] = 0$ and $\mathbb{V}[\varepsilon|\mathbf{X} = \mathbf{x}] \leq \sigma^2 < \infty$. Moreover, $\mathbf{X}$ is uniformly distributed on $[0,1]^d$ and m is L-Lipschitz continuous. Then, for all n, for all $\mathbf{x} \in [0,1]^d$,*

$$\mathbb{E}\big[m_{\infty,n}(\mathbf{x}) - m(\mathbf{x})\big]^2 \leq \frac{2\sigma^2}{n\alpha^k} + dL^2 C_1 \left(1 - \frac{1}{d} + \frac{(1-\alpha)^2}{d}\right)^k, \tag{3.6}$$

where

$$C_1 = \exp\left(\frac{\alpha}{d - 1 + (1-\alpha)^2}\right).$$

In addition, let $\beta = 1 - \frac{1}{d} + \frac{(1-\alpha)^2}{d}$. The right-hand side is minimal for

$$k_n = -\frac{1}{\ln(\alpha) + \ln\beta}\Big[\ln(n) + C_2\Big], \tag{3.7}$$

where

$$C_2 = \ln\left(\frac{dL^2 C_1 \ln(\beta)}{2\sigma^2 \ln(\alpha)}\right),$$

under the condition that $a_n \geq C_3 n^{\frac{\ln\alpha}{\ln\alpha + \ln\beta}}$, where the explicit expression of C_3 is given in Section 6. For these choices of k_n and a_n, there exists a constant $C_4 > 0$ such that

$$\mathbb{E}\big[m_{\infty,n}(\mathbf{x}) - m(\mathbf{x})\big]^2 \leq C_4 n^{-\frac{\ln\beta}{\ln(\alpha) + \ln\beta}}. \tag{3.8}$$

Quantile forests offer a lot more freedom to choose the split location than median forests. However, some important restriction of quantile forests compared to Breiman's forests is that the splitting procedure must not depend on the label Y_i: it is a key-point to prove Theorem 3.2 and Corollary 3.3. To see that quantile forests are still of some interest to study Breiman's forests, we can think of splitting the data set in two parts, using the first one to compute split locations and the second one to actually build the forest, by choosing in each cell, the splitting location that is

(1) the closest of the ideal split location (computed with the first part of the training set);
(2) and a quantile of order between α and $1-\alpha$, where $\alpha \in (0,1)$ is a prespecified parameter.

In that case, the splits are chosen in a data-driven manner (with the first part of the training set) and the proofs remain valid since the Y_i that are averaged in each terminal node are not used to compute the split locations. Therefore, Breiman's forests behaviour can be connected to quantile forests, if the splits of Breiman's forests are all performed between the quantiles of order $(\alpha, 1-\alpha)$. This is a mild assumption if splits are not performed at the cell edges (since there is no lower bound condition on α). Note that the fact that Breiman's splits are performed between quantiles of order α and $1-\alpha$ can be found in other works as in (see [27, 28]) or, with similar ideas, in [9].

Corollary 3.3. *Let $\beta = 1 - \frac{1}{d} + \frac{(1-\alpha)^2}{d}$. Consider one of the two quantile forests below:*

(i) A quantile forest without subsampling (i.e., $a_n = n$) and such that the parameter k_n satisfies (3.2).

(ii) A fully grown quantile forest whose parameters k_n and a_n satisfy $a_n \alpha^k = 4$ and (3.2).

In either case, under the same assumptions as in Theorem 3.2, we have, for all n, for all $\mathbf{x} \in [0,1]^d$,

$$\mathbb{E}\big[m_{\infty,n}(\mathbf{x}) - m(\mathbf{x})\big]^2 \leq C_4 n^{-\frac{\ln \beta}{\ln(\alpha)+\ln \beta}}.$$

Note that if we are only interested in the rate of consistency of quantile forests, we should take $\alpha = 1/2$ which minimizes the right-hand side of inequality 3.8. This is not surprising because this model operates without informations carried out by labels Y_i: the fastest manner to concentrate around the regression function is to create the smallest possible cells, which is done by considering median forests ($\alpha = 1/2$).

4. Experiments

In the light of Section 3, we carry out some simulations to investigate (i) how small-tree forests and subsampled forests compare with Breiman's forests and (ii) the influence of subsampling size and tree depth on Breiman's procedure. To do so, we start by defining various regression models on which the several experiments are based. Throughout this section, we assess the forest performances by computing their empirical $\mathbb{L}^2$ error.

Model 1 : $n = 800, d = 50, Y = \tilde{X}_1^2 + \exp(-\tilde{X}_2^2)$

Model 2 : $n = 600, d = 100, Y = \tilde{X}_1\tilde{X}_2 + \tilde{X}_3^2 - \tilde{X}_4\tilde{X}_7 + \tilde{X}_8\tilde{X}_{10} - \tilde{X}_6^2 + \mathcal{N}(0, 0.5)$

Model 3 : $n = 600, d = 100, Y = -\sin(2\tilde{X}_1) + \tilde{X}_2^2 + \tilde{X}_3 - \exp(-\tilde{X}_4) + \mathcal{N}(0, 0.5)$

Model 4 : $n = 600, d = 100, Y = \tilde{X}_1 + (2\tilde{X}_2 - 1)^2 + \sin(2\pi\tilde{X}_3)/(2 - \sin(2\pi\tilde{X}_3)) + \sin(2\pi\tilde{X}_4) + 2\cos(2\pi\tilde{X}_4) + 3\sin^2(2\pi\tilde{X}_4) + 4\cos^2(2\pi\tilde{X}_4) + \mathcal{N}(0, 0.5)$

Model 5 : $n = 700, d = 20, Y = \mathbb{1}_{\tilde{X}_1>0} + \tilde{X}_2^3 + \mathbb{1}_{\tilde{X}_4+\tilde{X}_6-\tilde{X}_8-\tilde{X}_9>1+\tilde{X}_{10}} + \exp(-\tilde{X}_2^2) + \mathcal{N}(0, 0.5)$

Model 6 : $n = 500, d = 30, Y = \sum_{k=1}^{10} \mathbb{1}_{\tilde{X}_k^3<0} - \mathbb{1}_{\mathcal{N}(0,1)>1.25}$

Model 7 : $n = 600, d = 300, Y = \tilde{X}_1^2 + \tilde{X}_2^2\tilde{X}_3\exp(-|\tilde{X}_4|) + \tilde{X}_6 - \tilde{X}_8 + \mathcal{N}(0, 0.5)$

Model 8 : $n = 500, d = 1000, Y = \tilde{X}_1 + 3\tilde{X}_3^2 - 2\exp(-\tilde{X}_5) + \tilde{X}_6$

For all regression frameworks, we consider covariates $\mathbf{X} = (X_1, \ldots, X_d)$ that are uniformly distributed over $[0,1]^d$. We also let $\tilde{X}_i = 2(X_i - 0.5)$ for $1 \leq i \leq d$. Some of these models are toy models (**Models 1, 5–8**). **Model 2** can be found in van der Laan et al. [25] and **Models 3–4** are presented in Meier et al. [16]. All numerical implementations have been performed using the free R software. For each experiment, the data set is divided into a training set (80% of the data set) and a test set (the remaining 20%). Then, the empirical risk ($\mathbb{L}^2$ error) is evaluated on the test set. For the sake of clarity, we exemplify our findings using only **Models 1–2**. The results for the other models, which exhibit similar behaviours as **Models 1–2**, can be found in the Appendix A.

4.1. Tree depth

We start by studying Breiman's original forests and small-tree Breiman's forests (in which the tree depth is limited). Breiman's forests are the standard procedure implemented in the `R` package `randomForest`, with

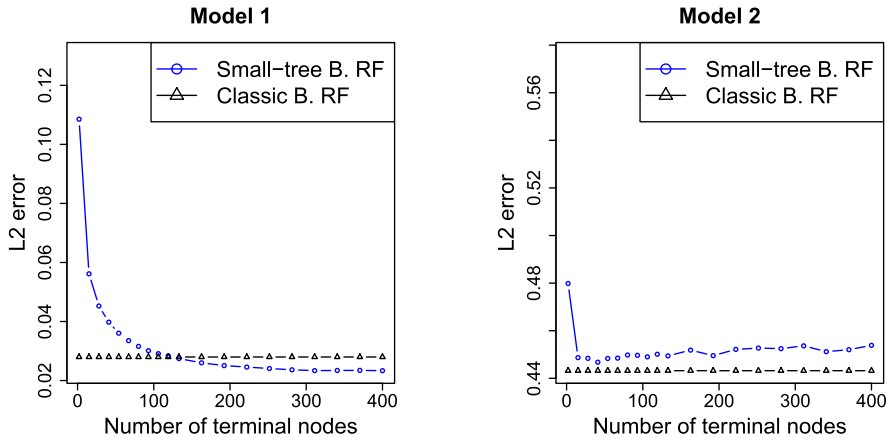


FIGURE 2. Comparison of standard Breiman's forests (B. RF) against small-tree Breiman's forests in terms of $\mathbb{L}^2$ error.

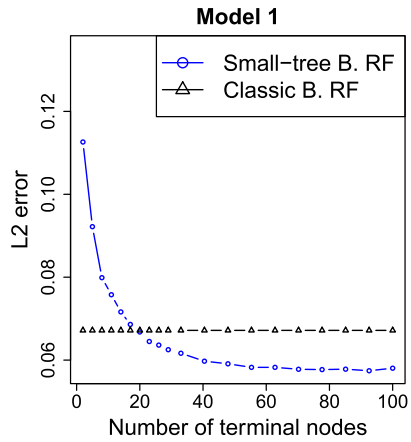
the parameters default values, as described in Section 2. Small-tree Breiman's forests are similar to Breiman's forests except that the tree depth is controlled *via* the parameter `maxnodes` (which corresponds to the number of leaves in each tree) and that the whole sample $\mathcal{D}_n$ is used to build each tree (without any resampling step). However, since there is a competition between parameters `maxnodes` and `nodesize`, we fix `nodesize` $= 1$ only for small-tree Breiman's forests.

In Figure 2, we present, for the **Models 1–2** introduced previously, the evolution of the empirical risk of small-tree forests for different numbers of terminal nodes. We add the representation of the empirical risk of Breiman's original forest in order to compare all forests errors at a glance. The two sub-figures of Figure 2 present forests built with 500 trees. The printed errors are obtained by averaging the risks of 50 forests. Because of the estimation/approximation compromise, we expect the empirical risk of small-tree forests to be decreasing and then increasing, as the number of leaves grows. In most of the models, it seems that the estimation error is too low to be detected, this is why several risks in Figure 2 are only decreasing as for **Model 1** (see Fig. A.1). Regarding that matter, **Model 2** is quite different since the error decreases and then increases, so that there is a real competition between estimation/approximation error. Indeed, **Model 2** is the only model such that the expectation of the signal (*i.e.* of the regression function) is equal to zero. Therefore, signal and noise are comparable in Model 2 in terms of expectation, which explains why we see the effect of both approximation and estimation error.

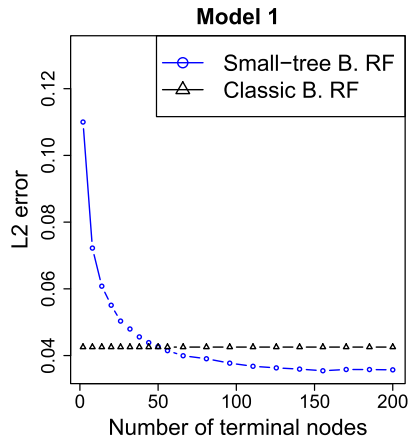
For every model (see Fig. A.1), we can notice that small-tree forests performance is comparable with the one of standard Breiman's forest, as long as the number of leaves is well chosen. For example, for the **Model 1**, a small-tree forest with approximately 110 leaves for each tree has the same empirical risk as the standard Breiman's forest. In the original algorithm of Breiman's forest, the construction of each tree uses a bootstrap sample of the data. For the small-tree forests, the whole data set is used for each tree, and then the randomness comes only from the pre-selected directions for splitting. The performances of bootstrapped and small-tree forests are very alike. Thus, bootstrap seems not to be the cornerstone of the Breiman's forest practical superiority to other regression algorithms. Indeed, as highlighted in the simulations, Breiman's forests are outperformed by small-tree forests, provided a good choice of the tree depth.

In order to study the optimal number of terminal nodes (`maxnodes` parameter in the R algorithm) as a function of the size of the training set, we draw the same curves as in Figure 2, for different training data set sizes ($n = 100$, 200, 300 and 400). For each size of the training set, the optimal `maxnodes` values are plotted in the last graph, where the optimal `maxnodes` value $m^\star$ is defined as

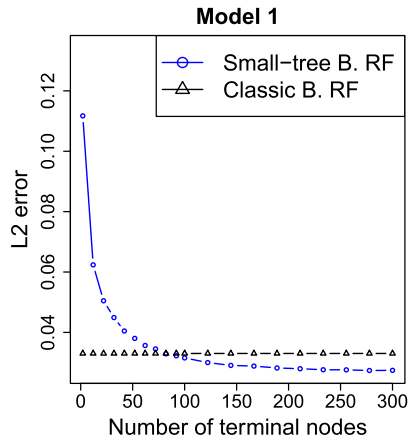
$$m^\star = \min\{m : |\hat{L}_m - \min_r \hat{L}_r| < 0.05 \times (\max_r \hat{L}_r - \min_r \hat{L}_r)\}$$



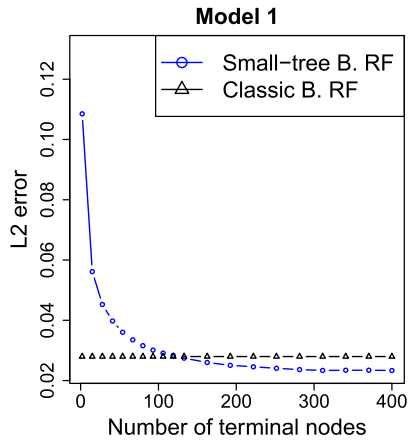
(A) Number of observations: $n = 100$.



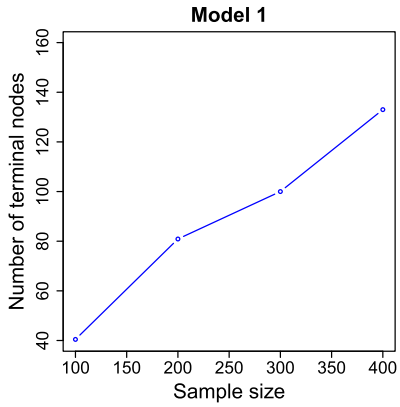
(B) Number of observations: $n = 200$.



(C) Number of observations: $n = 300$.



(D) Number of observations: $n = 400$.



(E) Optimal choices of the number of leaves.

FIGURE 3. (a)–(d) $\mathbb{L}^2$ error of small-tree and standard Breiman's forests in **Model 1** for different sizes of the training set (ranging from 100 to 400); (e) optimal values of the number of terminal nodes in **Model 1**.

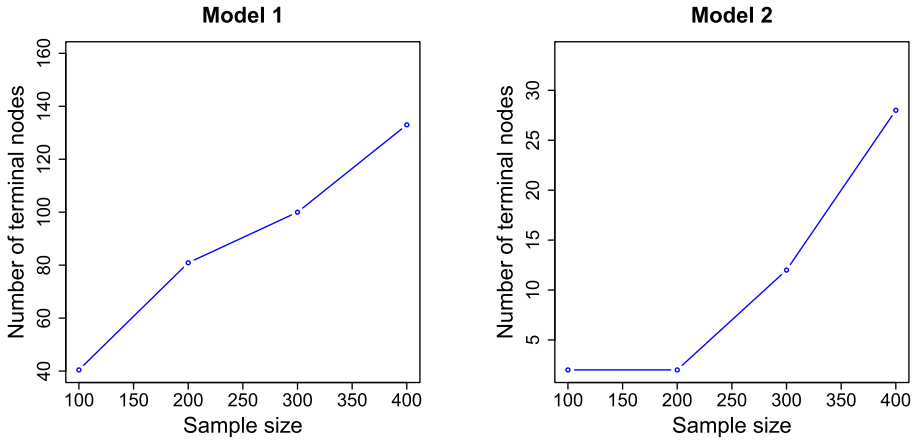


FIGURE 4. Optimal choices of the number of leaves.

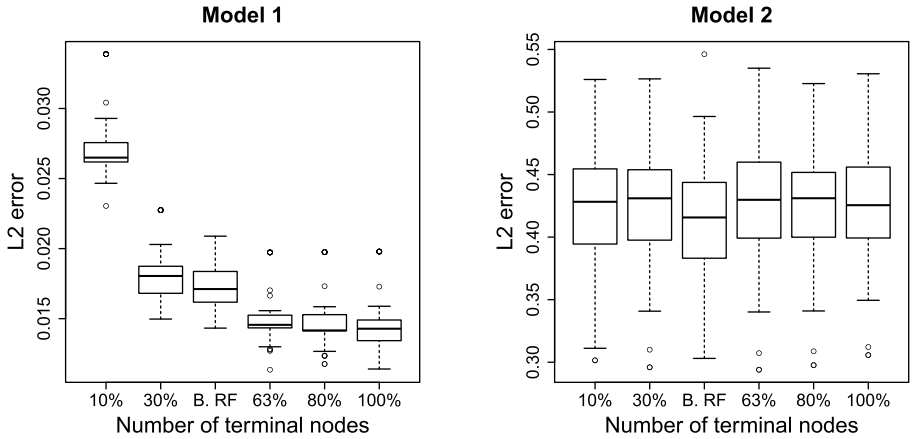


FIGURE 5. Comparison of standard Breiman's forests against several small-tree Breiman's forests in terms of $\mathbb{L}^2$ error.

where $\hat{L}_r$ is the risk of the forest built with the parameter `maxnodes`$= r$. The results can be seen in Figure 3. According to Figure 4e, the optimal `maxnodes` value seems to be proportional to the sample size. For **Model 1**, the optimal value $m^\star$ seems to verify $0.25n < m^\star < 0.3n$. The other models show a similar behaviour, as it can be seen in Figure 4.

We also present the $\mathbb{L}^2$ errors of small-tree Breiman's forests for different number of terminal nodes (10%, 30%, 63%, 80% and 100% of the sample size), when the sample size is fixed. The results can be found in Figure 5 for **Models 1–2** in the form of box-plots (see Fig. A.3 for the other models). We can notice that the forests such that `maxnodes`$= 0.3n$ give similar (**Model 1**) or best (**Model 6**) performances as compared to the standard Breiman's forest.

4.2. Subsampling

In this section, we study the influence of subsampling on Breiman's forests by comparing the original Breiman's procedure with subsampled Breiman's forests. While the effect of subsampling on ensemble methods is a widely documented topic (see, *e.g.*, [7, 29]), we are more interested in the similarities between subsampling and tree depth on the performance of the specific random forests algorithm. Subsampled Breiman's forests

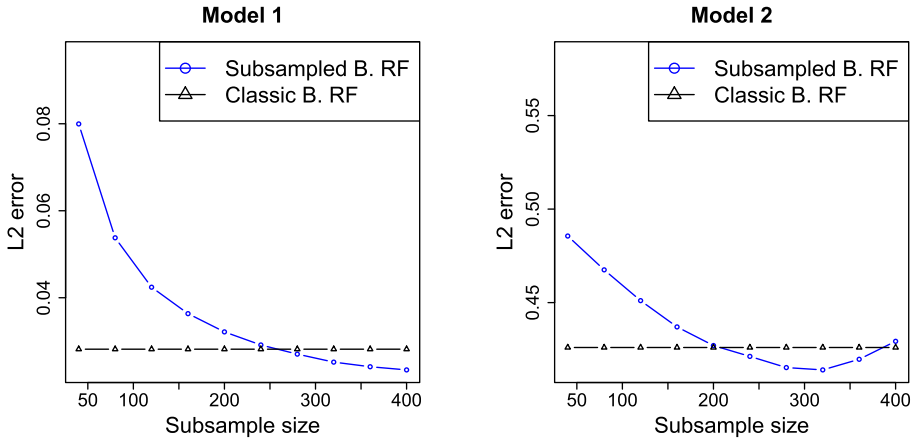


FIGURE 6. Standard Breiman forests *versus* subsampled Breiman forests.

are nothing but Breiman's forests where the subsampling step consists in choosing a_n observations without replacement (instead of choosing n observations among n with replacement), where a_n is the subsample size. Comparison of Breiman's forests and subsampled Breiman's forests is presented in Figure 6 for the **Models 1–2**. More precisely, we can see the evolution of the empirical risk of subsampled forests with different subsampling values, and the empirical risk of the Breiman's forest as a reference. The two sub-figures of Figure 6 present forests built with 500 trees. The printed errors are obtained by averaging the risks of 50 forests.

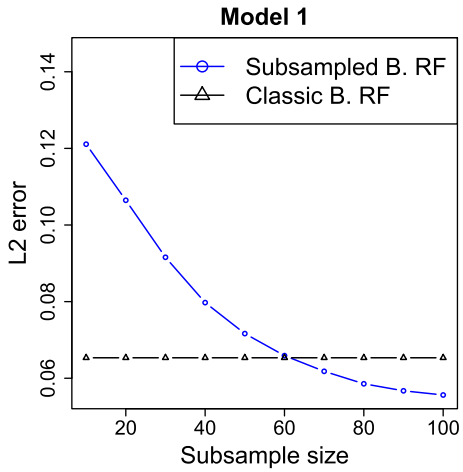
For every model, we can notice that subsampled forests performance is comparable with the one of standard Breiman's forest, as long as the subsampling parameter is well chosen. For example, a forest with a subsampling rate of 50% has the same empirical risk as the standard Breiman's forest, for **Model 2**. Once again, the similarity between bootstrapped and subsampled Breiman's forests moves aside bootstrap as a performance criteria. Thus, as shown in the simulations, the bootstrap step is not a key component of Breiman's forests since they are outperformed by subsampled Breiman's forests (up to a proper tuning of the subsample size, see also [20]).

We want of course to study the optimal subsampling size (`sampsize` parameter in the R algorithm) as a function of the size of the training set. For this, we draw the curves of Figure 6 for different training data set sizes, the same as in Figure 3. We also copy in an other graph the optimal subsample size $a_n^\star$ that we found for each size of the training set. The optimal subsampling size $a_n^\star$ is defined as

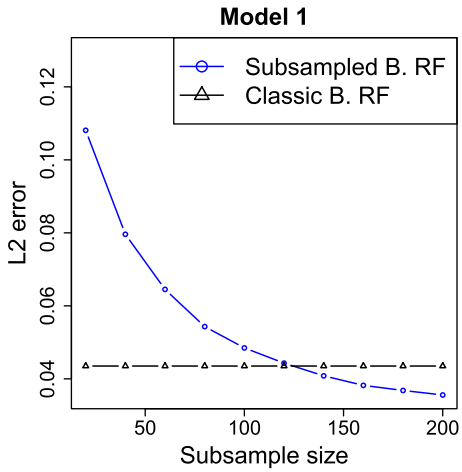
$$a_n^\star = \min\{a : |\hat{L}_a - \min_s \hat{L}_s| < 0.05 \times (\max_s \hat{L}_s - \min_s \hat{L}_s)\},$$

where $\hat{L}_s$ is the risk of the forest with parameter `sampsize` $= s$. The results can be seen in Figure 7. The optimal subsampling size seems, once again, to be proportional to the sample size, as illustrated in Figure 7e. For **Model 1**, the optimal value $a_n^\star$ seems to be close to $0.8n$. The other models show a similar behaviour, as it can be seen in Figure 8.

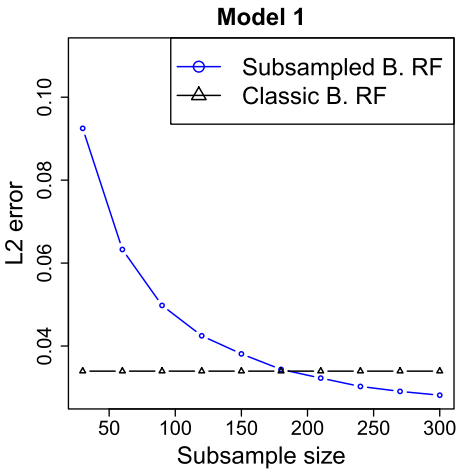
Then we present, in Figure 9, the $\mathbb{L}^2$ errors of subsampled Breiman's forests for different subsampling sizes ($0.4n$, $0.5n$, $0.63n$ and $0.9n$), when the sample size is fixed, for **Models 1–2** (see Fig. A.6 for the other models). We can notice that the forests with a subsampling size of $0.63n$ give similar performances as compared to the standard Breiman's forests. This is not surprising. Indeed, a bootstrap sample contains around 63% of distinct observations. Moreover the high subsampling sizes, around $0.9n$, lead to small $\mathbb{L}^2$ errors. It may arise from the probably high signal/noise rate. We have then implemented experiments with a higher noise. Precisely, we have multiplied the Gaussian noise by a factor 2 except for **Model 6** where it has been multiplied by 4. The results are presented in Figure 10, where we see the influence of signal and noise since the curves decrease and then



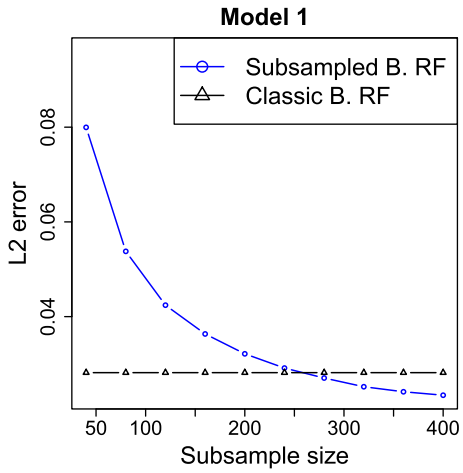
(A) Number of observations: $n = 100$.



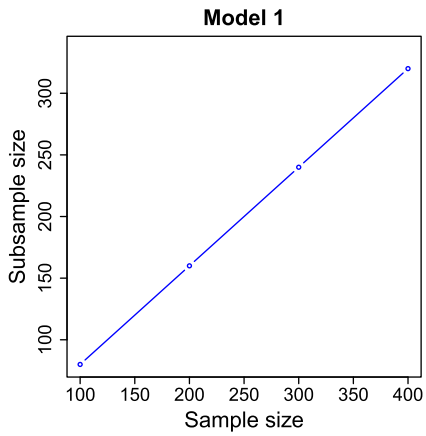
(B) Number of observations: $n = 200$.



(C) Number of observations: $n = 300$.



(D) Number of observations: $n = 400$.



(E) Optimal choices of the subsample size.

FIGURE 7. (a)–(d) $\mathbb{L}^2$ error of subsampled and Breiman's forests in **Model 1** for different sizes of the training set (ranging from 100 to 400); (e) optimal values of the subsample size in **Model 1**.

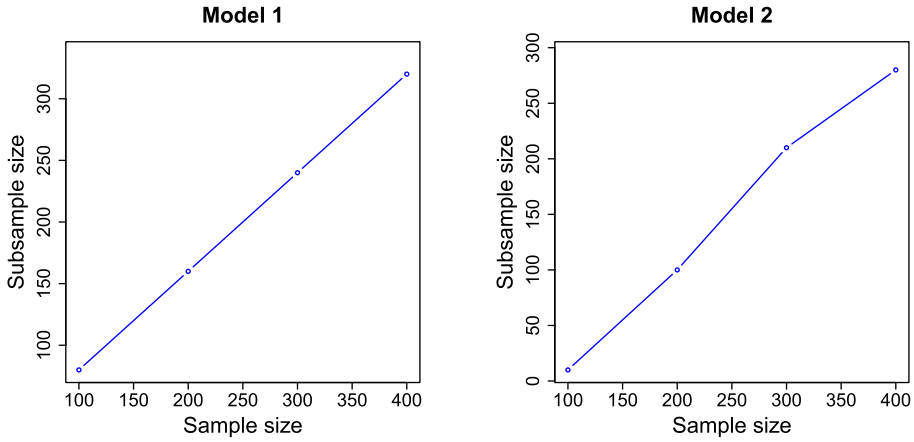


FIGURE 8. Optimal choices of the subsample size.

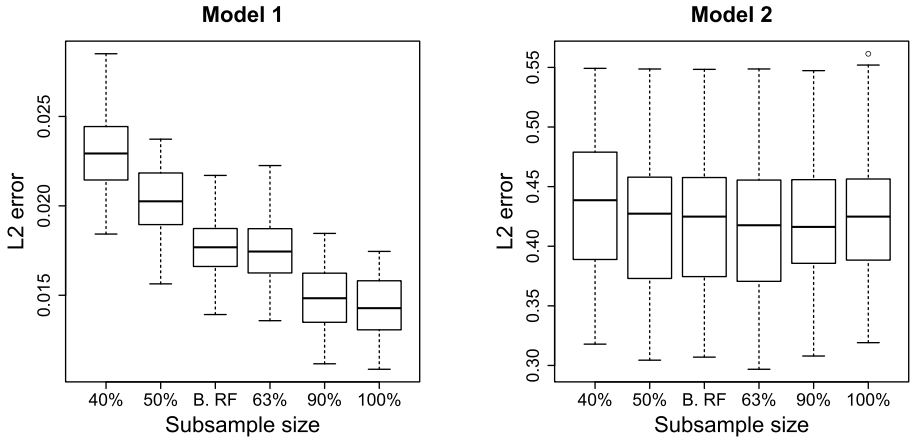


FIGURE 9. Standard Breiman forests *versus* several subsampled Breiman forests.

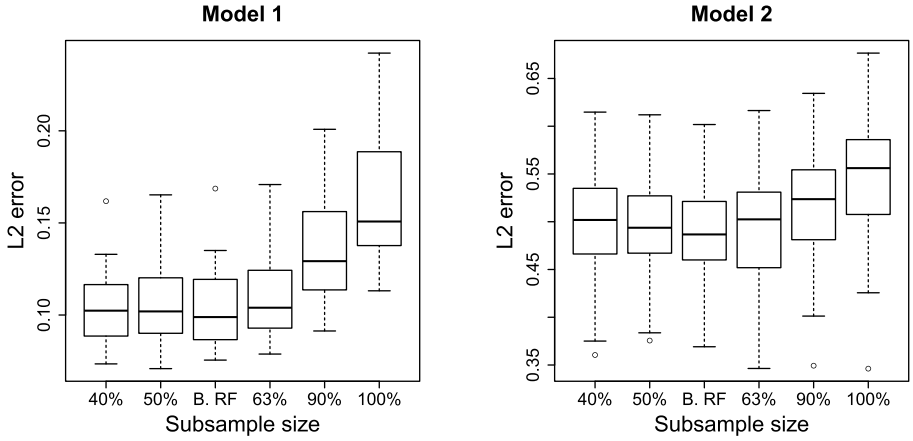


FIGURE 10. Standard Breiman forests *versus* several subsampled Breiman forests (noisy models).

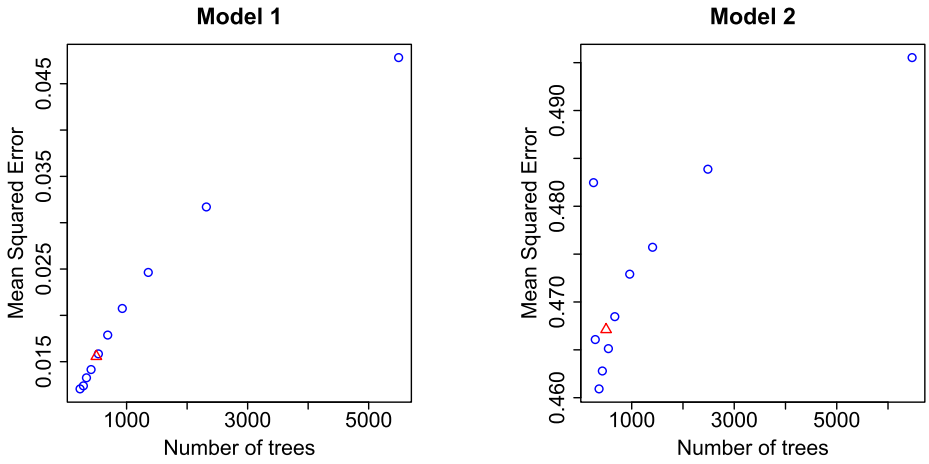


FIGURE 11. Performance of subsampled Breiman's forests (in blue) with same computation cost as Breiman's forests (in red).

increase. That is why we can lawfully use the subsampling size as an optimization parameter for the Breiman's forest performance.

4.3. Computation time

In the previous experiments, the number of trees has been kept fixed since it is generally admitted that this parameter has little impact on the random forest performance. Indeed, the theory tells us that this parameter should be chosen as large as possible, the only limitation being the available computation time.

Since data sets can be really huge, we want to seriously consider the time limitation problem. The general purposes of the following simulation is to find the combination of parameters (number of trees and number of terminal nodes/subsample size) such that the computation time of the corresponding algorithm is the same as Breiman's forests, with the lowest mean squared error.

More precisely, we consider the previous models (**Model 1–8**) with the default number of observations. For each model, we first run a Breiman's forests with a subsampling rate of 63% without replacement and evaluate its error and its computation time. We replicate this procedure ten times and compute the average of the error and the average of computation time.

This averaged computation time is taken as baseline: we want all the forest algorithms to have the same computation time as this first Breiman's forest. To do so, we consider a grid of different values for subsampling rate (from 10% to 100% of the whole data set). For each value, we first run a forest with this subsampling rate (without replacement) with 100 trees and, since the computation time is linear with the number of trees, we calculate the number of trees that will lead to the same computation time as the baseline. Thus for each value of the subsampling rate, we obtain a specific number of trees. Then we can run the corresponding forest and plot its error.

As seen in Figure 11, the number of trees does not influence much the random forest performance. The shape of resulting curves are very similar to that of the previous section, except that the x-axis is reversed. Indeed, increasing the number of trees leads to decreasing the subsample size (since the computation cost is kept fixed). At least for the models we consider here, one should not try to optimize the number of trees but rather the tree construction either by a proper tuning of the maximal number of terminal nodes or the subsample size. The results for all models can be found in Figure A.8 in Appendix A. The same experiments have been done by varying the number of terminal nodes instead of the subsample size. The results are quite similar and are to be found in Figure A.9 in Appendix A.

5. Discussion

In this paper, we studied the role of subsampling step and tree depth in both quantile and Breiman's forests procedure. By analyzing quantile forests, a simpler but close version of Breiman's forests, we show that the performance of subsampled quantile forests and that of small-tree quantile forests are similar, provided a proper tuning of the parameters of interest (subsample size and tree depth, respectively).

The extended experiments have shown similar results: Breiman's forests can be outperformed by either subsampled or small-tree Breiman's forests by properly tuning parameters. Noteworthy, tuning tree depth can be done at almost no additional cost while running Breiman's forests (due to the intrinsic recursive nature of forests). However, if one is interested in a faster procedure, subsampled Breiman's forests are to be preferred to small-tree forests.

As a by-product, our analysis also shows that there is no particular interest in bootstrapping data instead of subsampling: in our experiments, bootstrap is comparable (or worse) than a proper subsampling tuning. This sheds some light on several previous theoretical analysis where the bootstrap step was replaced by subsampling, which is more amenable to analysis. Similarly, proving theoretical results on fully grown Breiman's forests turned out to be extremely difficult. Our analysis shows that there is no theoretical background for considering default parameters in Breiman's forests instead of small-tree or subsampled Breiman's forests, which turn out to be easier to examine.

6. Proofs

Theorem 3.1 is a particular instance of Theorem 3.2 with $\alpha = 1/2$. Therefore, we will only prove Theorem 3.2. The proofs of Corollaries 3.1 and 3.2 are straightforward.

Proof of Theorem 3.2.
Let us start by recalling that the infinite quantile random forest estimate $m_{\infty,n}$ can be written as a local averaging estimate

$$\begin{aligned} m_{\infty,n}(\mathbf{x}) &= \mathbb{E}_{\Theta}[m_n(\mathbf{x},\Theta)] = \mathbb{E}_{\Theta}\Big[\sum_{i=1}^{n} W_{ni}(\mathbf{x})Y_i\Big] \\ &= \sum_{i=1}^{n} W_{ni}^{\infty}(\mathbf{x})Y_i, \end{aligned}$$

where

$$W_{ni}^{\infty}(\mathbf{x}) = \mathbb{E}_{\Theta}[W_{ni}(\mathbf{x},\Theta)] \quad \text{and} \quad W_{ni}(\mathbf{x},\Theta) = \frac{\mathbb{1}_{\mathbf{X}_i \in A_n(\mathbf{x},\Theta)}}{N_n(\mathbf{x},\Theta)}.$$

Since $A_n(\mathbf{x},\Theta)$ is the cell containing $\mathbf{x}$ in the tree built with the random parameter Θ, and $N_n(\mathbf{x},\Theta)$ is the number of observations falling into $A_n(\mathbf{x},\Theta)$, the quantity $\mathbb{1}_{\mathbf{X}_i \in A_n(\mathbf{x},\Theta)}$ indicates whether the observation $\mathbf{X}_i$ belongs to $A_n(\mathbf{x},\Theta)$. The $\mathbb{L}^2$-error of the forest estimate takes then the form

$$\begin{aligned} \mathbb{E}\big[m_{\infty,n}(\mathbf{x}) - m(\mathbf{x})\big]^2 &\le 2\mathbb{E}\Big[\sum_{i=1}^{n} W_{ni}^{\infty}(\mathbf{x})(Y_i - m(\mathbf{X}_i))\Big]^2 \\ &\quad + 2\mathbb{E}\Big[\sum_{i=1}^{n} W_{ni}^{\infty}(\mathbf{x})(m(\mathbf{X}_i) - m(\mathbf{x}))\Big]^2 \\ &= 2I_n + 2J_n. \end{aligned}$$

We can identify the term I_n as the estimation error and J_n as the approximation error, and then work on each term I_n and J_n separately.

6.1. Approximation error

Regarding J_n, by the Cauchy Schwartz inequality,

$$\begin{aligned}
J_n &\leq \mathbb{E}\Bigg[\sum_{i=1}^{n}\sqrt{W_{ni}^{\infty}(\mathbf{x})}\sqrt{W_{ni}^{\infty}(\mathbf{x})}|m(\mathbf{X}_i)-m(\mathbf{x})|\Bigg]^2\\
&\leq \mathbb{E}\Bigg[\sum_{i=1}^{n}W_{ni}^{\infty}(\mathbf{x})(m(\mathbf{X}_i)-m(\mathbf{x}))^2\Bigg]\\
&\leq \mathbb{E}\left[\sum_{i=1}^{n}\frac{\mathbb{1}_{\mathbf{X}_i\in A_n(\mathbf{x},\Theta)}}{N_n(\mathbf{x},\Theta)}\sup_{\substack{\mathbf{x},\mathbf{z},\\ |\mathbf{x}-\mathbf{z}|\leq \mathrm{diam}(A_n(\mathbf{x},\Theta))}}|m(\mathbf{x})-m(\mathbf{z})|^2\right]\\
&\leq L^2\mathbb{E}\left[\frac{1}{N_n(\mathbf{x},\Theta)}\sum_{i=1}^{n}\mathbb{1}_{\mathbf{X}_i\in A_n(\mathbf{x},\Theta)}\left(\mathrm{diam}(A_n(\mathbf{x},\Theta))\right)^2\right]\\
&\leq L^2\mathbb{E}\Big[\left(\mathrm{diam}(A_n(\mathbf{x},\Theta))\right)^2\Big],
\end{aligned}$$

where the fourth inequality is due to the L-Lipschitz continuity of m. Let $V_\ell(\mathbf{x},\Theta)$ be the length of the cell containing $\mathbf{x}$ along the ℓth side. Then,

$$J_n \leq L^2\sum_{l=1}^{d}\mathbb{E}\Big[V_l(\mathbf{x},\Theta)^2\Big].$$

According to Lemma 6.1 specified further, we have

$$\mathbb{E}\Big[V_l(\mathbf{x},\Theta)^2\Big] \leq C_1\left(1-\frac{1}{d}+\frac{(1-\alpha)^2}{d}\right)^k,$$

with

$$C_1 = \exp\left(\frac{\alpha}{d-1+(1-\alpha)^2}\right).$$

Thus, for all k, we have

$$J_n \leq dL^2C_1\left(1-\frac{1}{d}+\frac{(1-\alpha)^2}{d}\right)^k.$$

6.2. Estimation error

Let us now focusing on the term I_n, we have

$$\begin{aligned}
I_n &= \mathbb{E}\Bigg[\sum_{i=1}^{n}W_{ni}^{\infty}(\mathbf{x})(Y_i-m(\mathbf{X}_i))\Bigg]^2\\
&= \sum_{i=1}^{n}\sum_{i=1}^{n}\mathbb{E}\Big[W_{ni}^{\infty}(\mathbf{x})W_{nj}^{\infty}(\mathbf{x})(Y_i-m(\mathbf{X}_i))(Y_j-m(\mathbf{X}_j))\Big]
\end{aligned}$$

$$= \mathbb{E}\left[\sum_{i=1}^{n}(W_{ni}^{\infty}(\mathbf{x}))^2(Y_i - m(\mathbf{X}_i))^2\right]$$
$$\leq \sigma^2\mathbb{E}\left[\max_{1\leq i\leq n} W_{ni}^{\infty}(\mathbf{x})\right],$$

since the variance of ε_i is bounded above by σ^2. Recalling that a_n is the number of subsampled observations used to build the tree, we have

$$n_j \geq \alpha n_{j-1} - 1,$$

which leads to

$$n_k \geq \alpha^k a_n - \frac{1}{1-\alpha} \geq \alpha^k a_n - 2.$$

Therefore,

$$\mathbb{E}\left[\max_{1\leq i\leq n} W_{ni}^{\infty}(\mathbf{x})\right] = \mathbb{E}\left[\max_{1\leq i\leq n} \mathbb{E}_{\Theta}\left[\frac{\mathbb{1}_{\mathbf{X}_i\in A_n(\mathbf{x},\Theta)}}{N_n(\mathbf{x},\Theta)}\right]\right]$$
$$\leq \frac{1}{a_n\alpha^k - 2}\mathbb{E}\left[\max_{1\leq i\leq n} \mathbb{P}_{\Theta}\left[\mathbf{X}_i \in A_n(\mathbf{x},\Theta)\right]\right].$$

Observe that in the subsampling step, there are exactly $\binom{a_n-1}{n-1}$ choices to pick a fixed observation $\mathbf{X}_i$. Since $\mathbf{x}$ and $\mathbf{X}_i$ belong to the same cell only if $\mathbf{X}_i$ is selected in the subsampling step, we see that

$$\mathbb{P}_{\Theta}\left[\mathbf{X}_i \in A_n(\mathbf{x},\Theta)\right] \leq \frac{\binom{a_n-1}{n-1}}{\binom{a_n}{n}} = \frac{a_n}{n}.$$

So,

$$I_n \leq \sigma^2\frac{1}{a_n\alpha^k - 2}\frac{a_n}{n} \leq 2\sigma^2\frac{1}{n\alpha^k},$$

since $a_n\alpha^k \geq 4$. Consequently, we obtain

$$\mathbb{E}\big[m_{\infty,n}(\mathbf{x}) - m(\mathbf{x})\big]^2 \leq I_n + J_n \leq 2\sigma^2\frac{1}{n\alpha^k} + dL^2C_1\left(1 - \frac{1}{d} + \frac{(1-\alpha)^2}{d}\right)^k.$$

6.3. Optimal parameter choice

We now want to prove inequality (3.8) in Theorem 3.2. Regarding Theorem 3.2, we want to find the optimal value of tree depth, in order to obtain the best rate of convergence for the forest estimate. Let $b_1 = \frac{2\sigma^2}{n}$ and $b_2 = dL^2C_1$ and

$$\beta = 1 - \frac{1}{d} + \frac{(1-\alpha)^2}{d}.$$

Then,

$$\mathbb{E}\big[m_{\infty,n}(\mathbf{x}) - m(\mathbf{x})\big]^2 \leq b_1\alpha^{-k} + b_2\beta^k.$$

Let $f : x \mapsto b_1 e^{x\ln(\alpha^{-1})} + b_2 e^{x\ln(\beta)}$. Thus,

$$\begin{aligned} f'(x) &= b_1 \ln(\alpha^{-1}) e^{x\ln(\alpha^{-1})} + b_2 \ln(\beta) e^{x\ln(\beta)} \\ &= b_1 \ln(\alpha^{-1}) e^{x\ln(\alpha^{-1})} \left(1 + \frac{b_2 \ln(\beta)}{b_1 \ln(\alpha^{-1})} e^{x(\ln(\beta)-\ln(\alpha^{-1}))}\right). \end{aligned}$$

Since $\beta \leq 1$, $f'(x) \leq 0$ for all $x \leq x^\star$ and $f'(x) \geq 0$ for all $x \geq x^\star$, where $x^\star$ satisfies

$$\begin{aligned} & f'(x^\star) = 0 \\ \iff \quad & x^\star = \frac{1}{\ln(\alpha^{-1}) - \ln(\beta)} \ln\left(-\frac{b_2 \ln(\beta)}{b_1 \ln(\alpha^{-1})}\right) \\ \iff \quad & x^\star = \frac{1}{\ln(\alpha^{-1}) - \ln\beta}\Big[\ln(n) + C_2\Big], \end{aligned}$$

where $C_2 = \ln\left(-\frac{dL^2 C_1 \ln(\beta)}{2\sigma^2 \ln(\alpha^{-1})}\right)$.

Consequently,

$$\begin{aligned} \mathbb{E}\big[m_{\infty,n}(\mathbf{x}) - m(\mathbf{x})\big]^2 &\leq b_1 \exp(x^\star \ln(\alpha^{-1})) + b_2 \exp(x^\star \ln\beta) \\ &\leq b_1 \exp\left(\frac{1}{\ln(\alpha^{-1}) - \ln\beta}\Big[\ln(n) + C_2\Big] \ln(\alpha^{-1})\right) \\ &\quad + b_2 \exp\left(\frac{1}{\ln(\alpha^{-1}) - \ln\beta}\Big[\ln(n) + C_2\Big] \ln\beta\right) \\ &\leq b_1 \exp\left(\frac{C_2 \ln(\alpha^{-1})}{\ln(\alpha^{-1}) - \ln\beta}\right) \exp\left(\frac{\ln(\alpha^{-1})}{\ln(\alpha^{-1}) - \ln\beta} \ln(n)\right) \\ &\quad + b_2 \exp\left(\frac{C_2 \ln\beta}{\ln 2 - \ln\beta}\right) \exp\left(\frac{\ln\beta}{\ln(\alpha^{-1}) - \ln\beta} \ln(n)\right) \\ &\leq \tilde{b}_1 \exp\left(\left(\frac{\ln(\alpha^{-1})}{\ln(\alpha^{-1}) - \ln\beta} - 1\right) \ln(n)\right) \\ &\quad + \tilde{b}_2 \exp\left(\frac{\ln\beta}{\ln(\alpha^{-1}) - \ln\beta} \ln(n)\right) \\ &\leq B n^{\frac{\ln\beta}{\ln(\alpha^{-1}) - \ln\beta}}, \end{aligned} \tag{6.1}$$

where $B = \tilde{b}_1 + \tilde{b}_2$ with $\tilde{b}_1 = 2\sigma^2 \exp\left(\frac{C_2 \ln(\alpha^{-1})}{\ln(\alpha^{-1}) - \ln\beta}\right)$ and $\tilde{b}_2 = C_2 \exp\left(\frac{C_2 \ln\beta}{\ln(\alpha^{-1}) - \ln\beta}\right)$. Note that this analysis is valid only for $a_n \alpha^{k_n} \geq 4$, that is

$$a_n \geq 4.\alpha^{-\frac{C_2}{\ln(\alpha^{-1}) - \ln\beta}}.n^{\frac{\ln(\alpha^{-1})}{\ln(\alpha^{-1}) - \ln\beta}},$$

We have

$$\begin{aligned} a_n \geq 4\alpha^{-k} \Leftrightarrow & a_n \geq 4\alpha^{-\frac{\ln(n) + C_2}{\ln(\alpha^{-1}) - \ln\beta}} \\ \Leftrightarrow & a_n \geq 4\alpha^{-\frac{C_2}{\ln(\alpha^{-1}) - \ln\beta}} \alpha^{-\frac{\ln(n)}{\ln(\alpha^{-1}) - \ln\beta}} \\ \Leftrightarrow & a_n \geq 4\alpha^{-\frac{C_2}{\ln(\alpha^{-1}) - \ln\beta}} n^{\frac{\ln(\alpha^{-1})}{\ln(\alpha^{-1}) - \ln\beta}}, \end{aligned}$$

where, simple calculations show that

$$\begin{aligned}\alpha^{-\frac{C_2}{\ln(\alpha^{-1})-\ln\beta}} &= \alpha^{-\frac{\ln\left(-\frac{dL^2C_1\ln(\beta)}{2\sigma^2\ln(\alpha^{-1})}\right)}{\ln(\alpha^{-1})-\ln\beta}} \\ &= \left(-\frac{dL^2C_1\ln\left(1-\frac{1}{d}+\frac{(1-\alpha)^2}{d}\right)}{2\sigma^2\ln(\alpha^{-1})}\right)^{\frac{\ln(\alpha^{-1})}{\ln(\alpha^{-1})-\ln\beta}} \\ &\leq \left(\frac{L^2C_1\alpha(2-\alpha)}{2\sigma^2\ln(\alpha^{-1})}\right)^{\frac{\ln(\alpha^{-1})}{\ln(\alpha^{-1})-\ln\beta}}.\end{aligned}$$

Thus, the upper bound (6.1) is valid if $a_n \geq C_3\, n^{\frac{\ln(\alpha^{-1})}{\ln(\alpha^{-1})-\ln\beta}}$, where

$$C_3 = 4\left(\frac{L^2C_1\alpha(2-\alpha)}{2\sigma^2\ln(\alpha^{-1})}\right)^{\frac{\ln(\alpha^{-1})}{\ln(\alpha^{-1})-\ln\beta}}.$$

□

We set up now Lemma 6.1 about the length of a cell that we used to bound the approximation error.

Lemma 6.1. *For all $\ell \in \{1,\dots,d\}$ and $k \in \mathbb{N}^*$, we have*

$$\mathbb{E}\Big[V_l(\mathbf{x},\Theta)^2\Big] \leq C_1\left(1-\frac{1}{d}+\frac{(1-\alpha)^2}{d}\right)^k,$$

with

$$C_1 = \exp\left(\frac{\alpha}{d-1+(1-\alpha)^2}\right).$$

Proof of Lemma 6.1.
Let us fix $\mathbf{x} \in [0,1]^d$ and denote by $n_0, n_1, \dots, n_k$ the number of points in the successive cells containing $\mathbf{x}$ (for example, n_0 is the number of points in the root of the tree, that is $n_0 = a_n$). Note that $n_0, n_1, \dots, n_k$ depends on $\mathcal{D}_n$ and Θ, but to lighten notations, we omit these dependencies. Recalling that $V_\ell(\mathbf{x},\Theta)$ is the length of the ℓth side of the cell containing $\mathbf{x}$, this quantity can be written as a product of independent beta distributions:

$$V_\ell(\mathbf{x},\Theta) \overset{\mathcal{D}}{=} \prod_{j=1}^{k}\big[B(n_j+1, n_{j-1}-n_j)\big]^{\delta_{\ell,j}(\mathbf{x},\Theta)},$$

where $B(\alpha,\beta)$ denotes the beta distribution of parameters α and β, and the indicator $\delta_{\ell,j}(\mathbf{x},\Theta)$ equals to 1 if the jth split of the cell containing $\mathbf{x}$ is performed along the ℓth dimension (and 0 otherwise). Consequently,

$$\begin{aligned}\mathbb{E}\big[V_\ell(\mathbf{x},\Theta)^2\big] &= \prod_{j=1}^{k}\mathbb{E}\Big[\big[B(n_j+1,n_{j-1}-n_j)\big]^{2\delta_{\ell,j}(\mathbf{x},\Theta)}\Big] \\ &= \prod_{j=1}^{k}\mathbb{E}\Big[\mathbb{E}\Big[\big[B(n_j+1,n_{j-1}-n_j)\big]^{2\delta_{\ell,j}(\mathbf{x},\Theta)}\big|\delta_{\ell,j}(\mathbf{x},\Theta)\Big]\Big]\end{aligned}$$

$$
\begin{aligned}
&= \prod_{j=1}^{k} \mathbb{E}\Big[\mathbb{1}_{\delta_{\ell,j}(\mathbf{x},\Theta)=0} + \mathbb{E}\big[B(n_j+1, n_{j-1}-n_j)\big]^2 \mathbb{1}_{\delta_{\ell,j}(\mathbf{x},\Theta)=1}\Big] \\
&= \prod_{j=1}^{k} \left(\frac{d-1}{d} + \frac{1}{d}\mathbb{E}\big[B(n_j+1, n_{j-1}-n_j)\big]^2\right) \\
&= \prod_{j=1}^{k} \left(\frac{d-1}{d} + \frac{1}{d}\frac{(n_j+1)(n_j+2)}{(n_{j-1}+1)(n_{j-1}+2)}\right) \\
&\le \prod_{j=1}^{k} \left(1 - \frac{1}{d} + \frac{(1-\alpha)^2}{d}\frac{(n_{j-1}+\frac{1}{1-\alpha})(n_{j-1}+\frac{2}{1-\alpha})}{(n_{j-1}+1)(n_{j-1}+2)}\right), \qquad (6.2)
\end{aligned}
$$

where the inequality stems from the relation $n_j \le (1-\alpha)n_{j-1}$ for all $j \in \{1, \dots, k\}$. We have

$$
\begin{aligned}
A &= \frac{(n_{j-1}+\frac{1}{1-\alpha})(n_{j-1}+\frac{2}{1-\alpha})}{(n_{j-1}+1)(n_{j-1}+2)} \\
&= 1 + \frac{3(\frac{1}{1-\alpha}-1)n_{j-1} + 2(\frac{1}{(1-\alpha)^2}-1)}{(n_{j-1}+1)(n_{j-1}+2)} \\
&= 1 + \frac{3(\frac{1}{1-\alpha}-1)}{n_{j-1}+1} + \frac{-6(\frac{1}{1-\alpha}-1) + 2(\frac{1}{(1-\alpha)^2}-1)}{(n_{j-1}+1)(n_{j-1}+2)} \\
&\le 1 + \frac{3(\frac{1}{1-\alpha}-1)}{n_{j-1}+1},
\end{aligned}
$$

since the function $x \mapsto 2x^2 - 6x + 4$ is negative for $x = 1/(1-\alpha) \in (1,2)$. Hence,

$$
\mathbb{E}\big[V_\ell(\mathbf{x},\Theta)^2\big] \le \prod_{j=1}^{k} \left(1 - \frac{1}{d} + \frac{(1-\alpha)^2}{d} + \frac{3\alpha(1-\alpha)}{d(n_{j-1}+1)}\right).
$$

Now, since

$$
n_j \ge \alpha n_{j-1} - 1,
$$

we get

$$
n_j \ge \alpha^j a_n - \frac{1}{1-\alpha} \ge \alpha^j a_n - 2,
$$

which leads to

$$
\begin{aligned}
\mathbb{E}\big[V_\ell(\mathbf{x},\Theta)^2\big] &\le \prod_{j=1}^{k} \left(1 - \frac{1}{d} + \frac{(1-\alpha)^2}{d} + \frac{3\alpha(1-\alpha)}{d\alpha^j a_n - d}\Big)\right) \\
&\le \prod_{j=1}^{k} \left(1 - \frac{1}{d} + \frac{(1-\alpha)^2}{d} + \frac{3\alpha(1-\alpha)\alpha^{k-j}}{d\alpha^k a_n - d\alpha^{k-j}}\Big)\right)
\end{aligned}
$$

$$\leq \prod_{j=1}^{k}\left(1-\frac{1}{d}+\frac{(1-\alpha)^2}{d}+\frac{3\alpha(1-\alpha)\alpha^{k-j}}{4d-d\alpha^{k-j}}\right)\Big)$$

(since $\alpha^k a_n \geq 4$)

$$\leq \prod_{j=0}^{k-1}\left(1-\frac{1}{d}+\frac{(1-\alpha)^2}{d}+\frac{\alpha(1-\alpha)\alpha^{j}}{d}\right)\Big).$$

Taking the logarithm of the last expression, we obtain

$$\begin{aligned}\log(\mathbb{E}\big[V_\ell(\mathbf{x},\Theta)^2\big]) &\leq \sum_{j=0}^{k-1}\log\left(1-\frac{1}{d}+\frac{(1-\alpha)^2}{d}+\frac{\alpha(1-\alpha)\alpha^{j}}{d}\right)\\ &\leq k\log\gamma+\sum_{j=0}^{k-1}\log\left(1+\frac{\alpha(1-\alpha)\alpha^{j}}{d\gamma}\right)\\ &\leq k\log\gamma+\frac{\alpha}{d\gamma}),\end{aligned}$$

where $\gamma = 1-\frac{1}{d}+\frac{(1-\alpha)^2}{d}$. Finally,

$$\begin{aligned}\mathbb{E}\big[V_\ell(\mathbf{x},\Theta)^2\big] &\leq \exp\left(k\log\gamma+\frac{\alpha}{d\gamma})\right)\Big)\\ &\leq C_1\gamma^k,\end{aligned}$$

where $C_1 = \exp(\alpha/(d\gamma))$.

□

Appendix A.

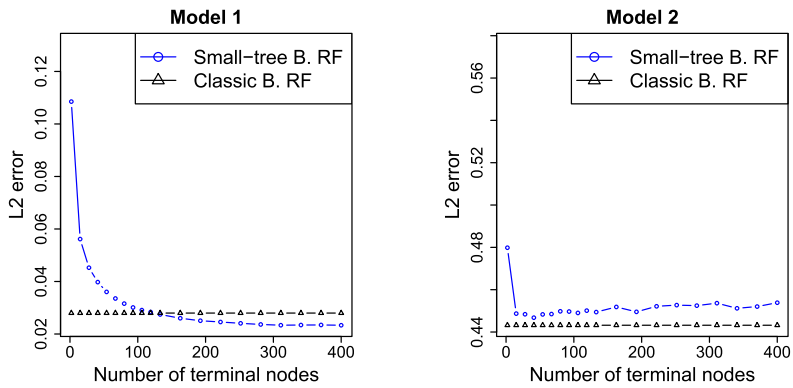


Figure A.1. Comparison of standard Breiman's forests (B. RF) against small-tree Breiman's forests in terms of $\mathbb{L}^2$ error.

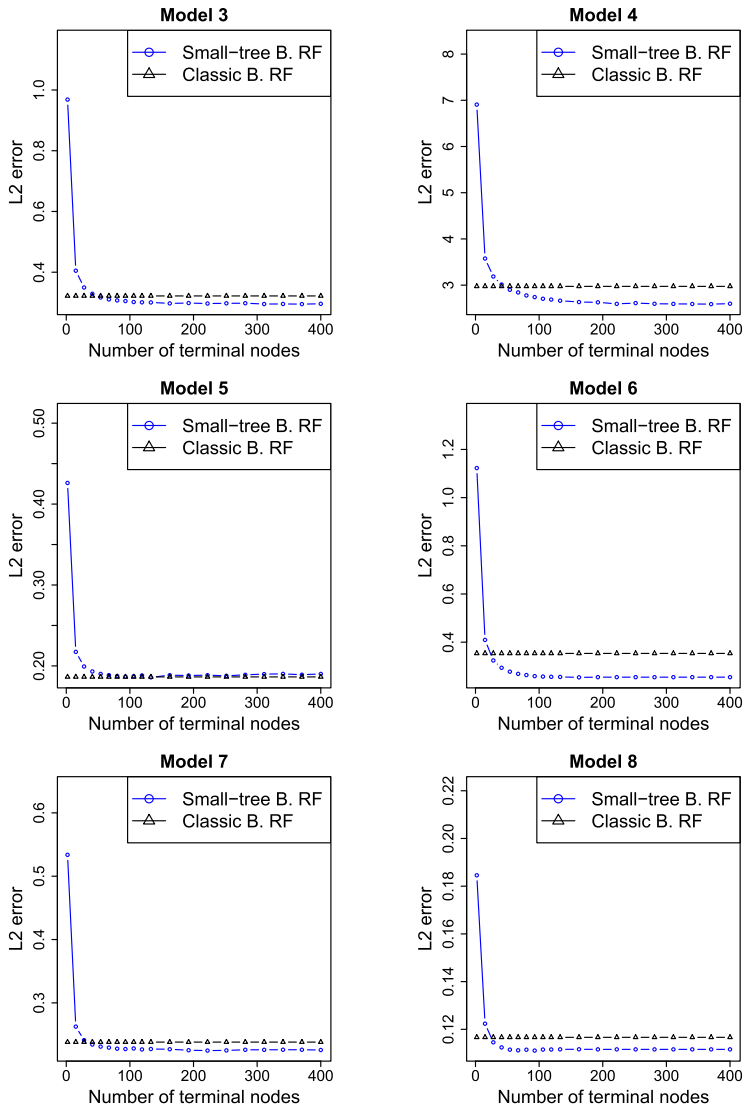


FIGURE A.1. (Continued).

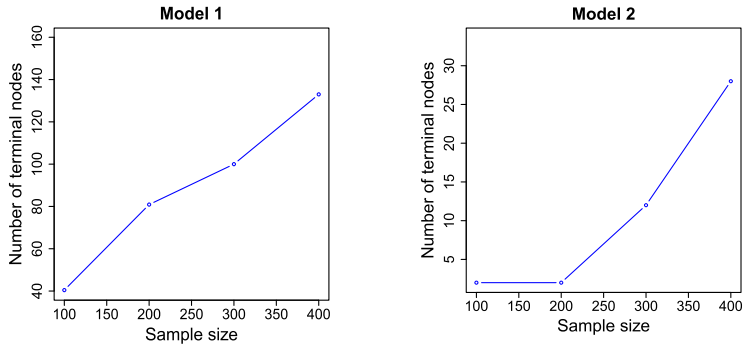


FIGURE A.2. Optimal values of the number of terminal nodes.

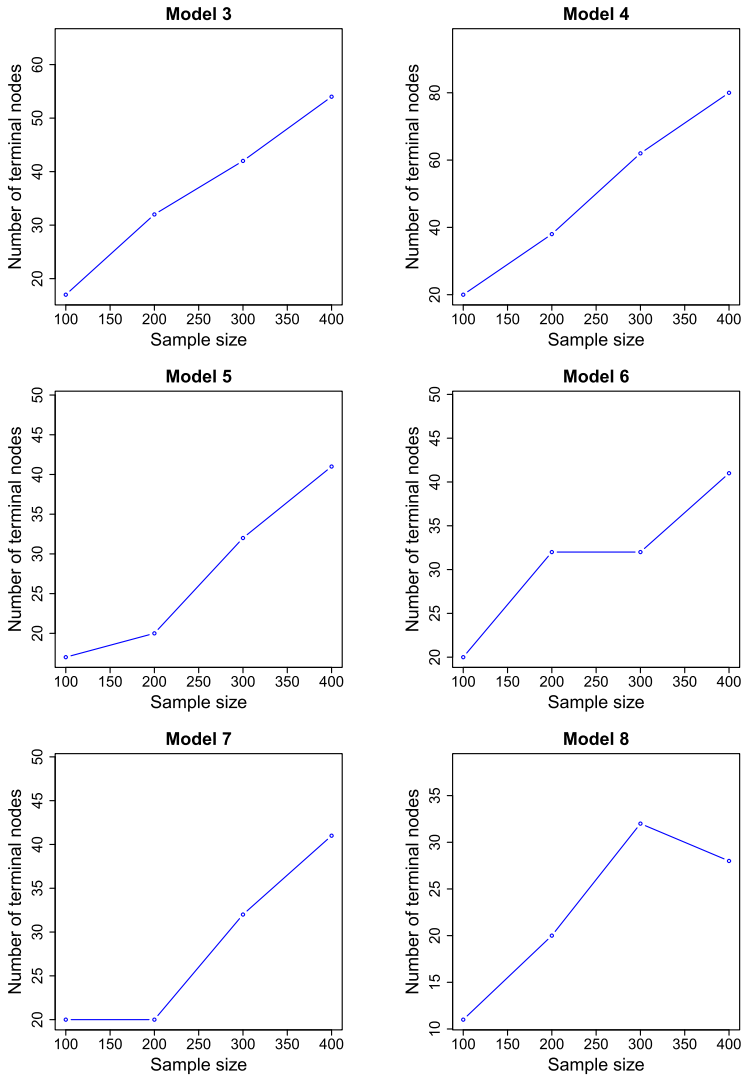


FIGURE A.2. (Continued).

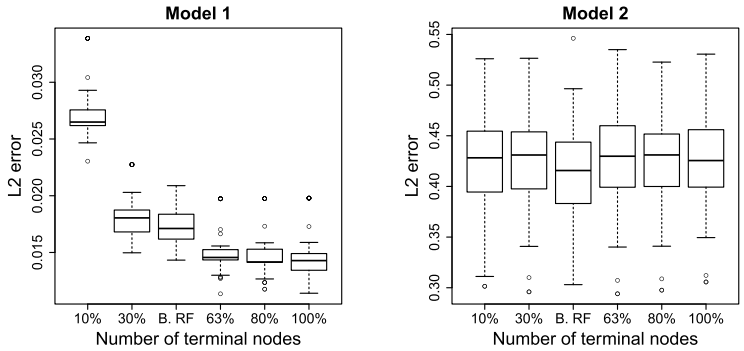


FIGURE A.3. Comparison of standard Breiman's forests against several small-tree Breiman's forests in terms of $\mathbb{L}^2$ error.

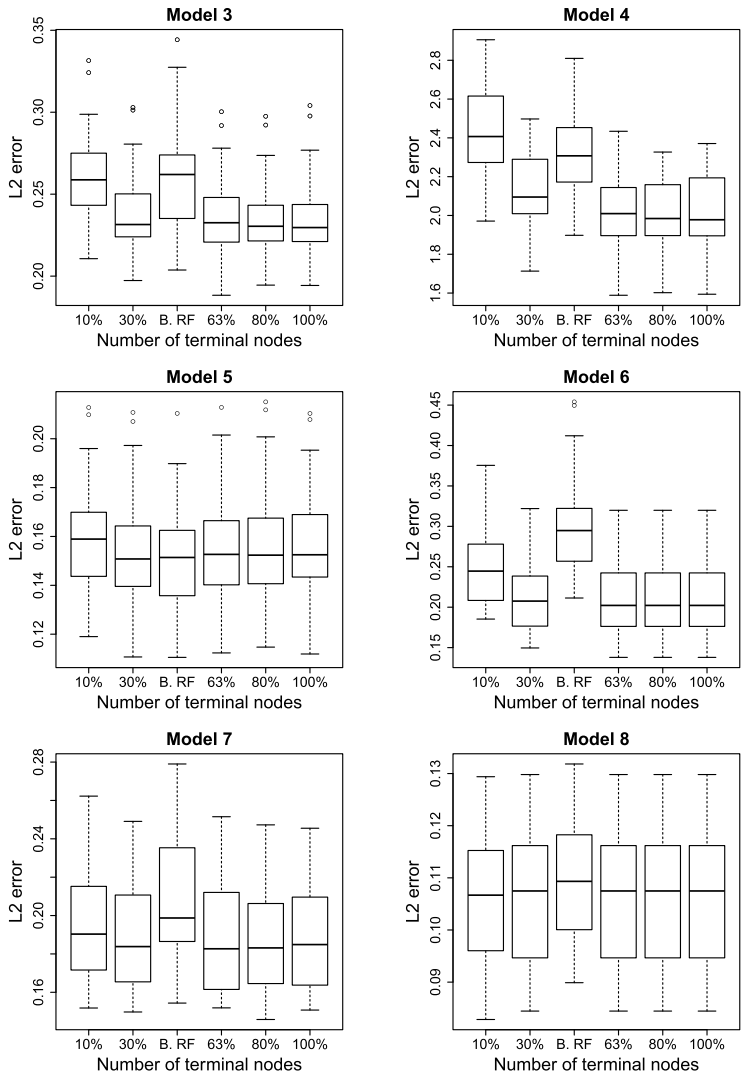


FIGURE A.3. (Continued).

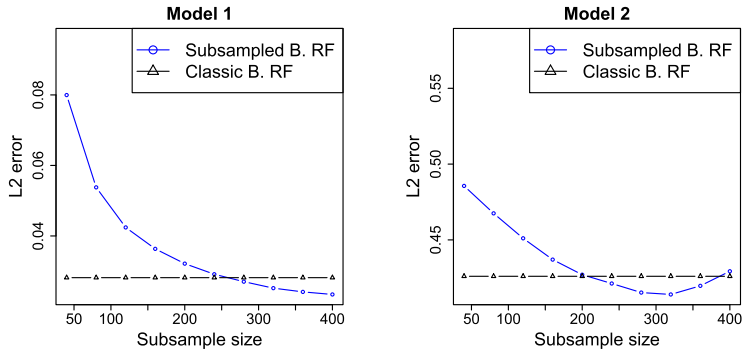


FIGURE A.4. Standard Breiman forests *versus* subsampled Breiman forests.

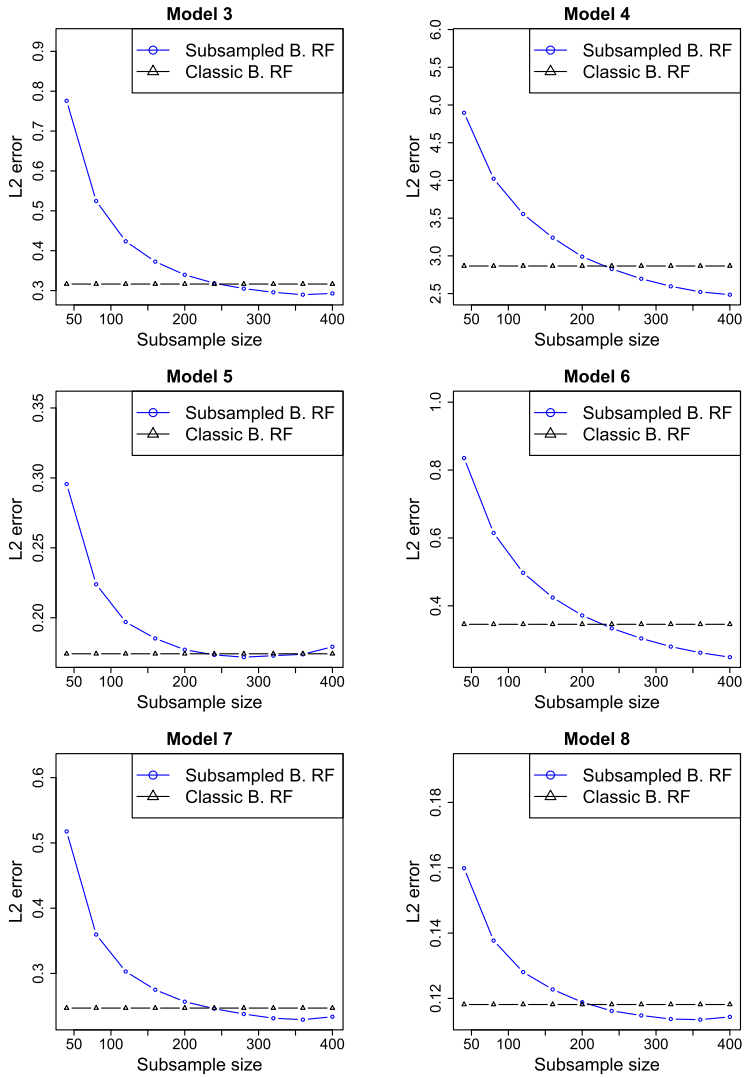


FIGURE A.4. (Continued).

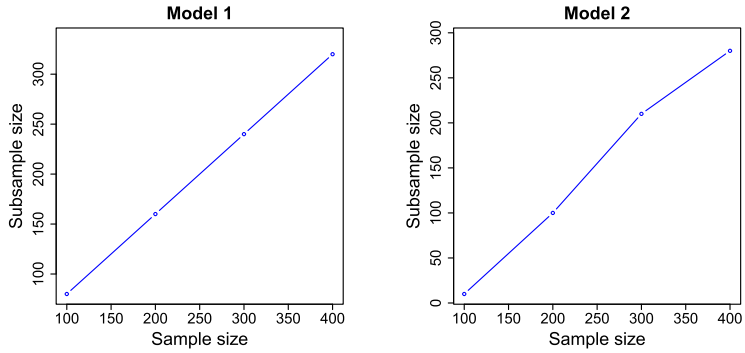


FIGURE A.5. Optimal values of subsample size.

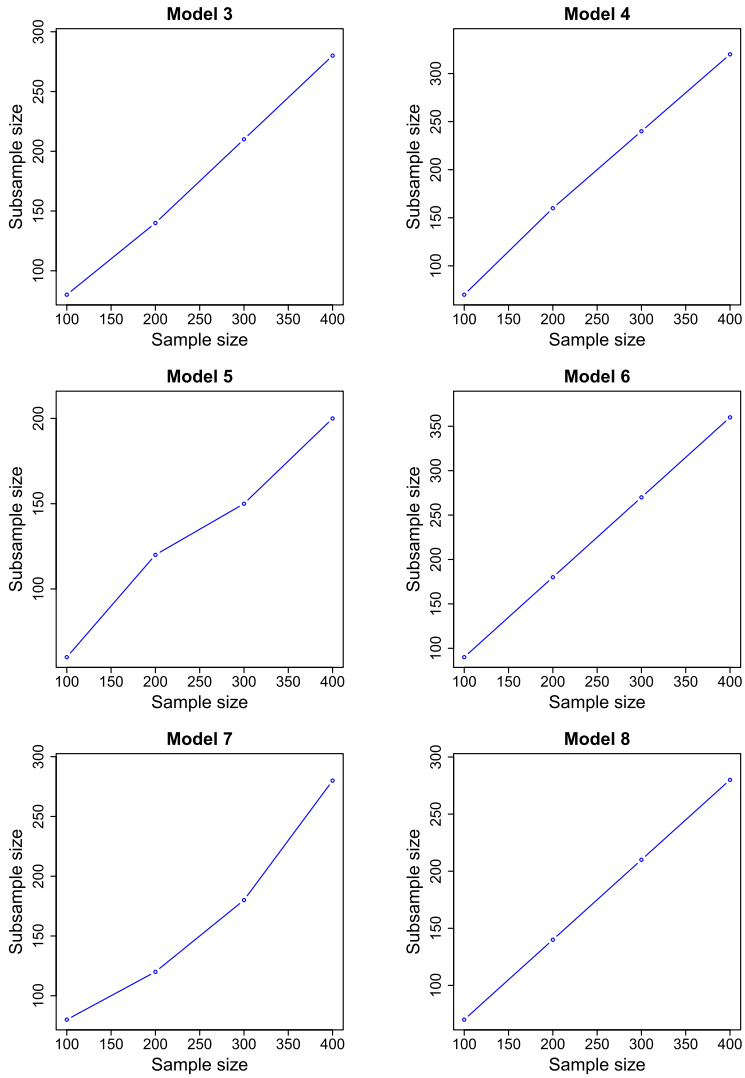


FIGURE A.5. (Continued).

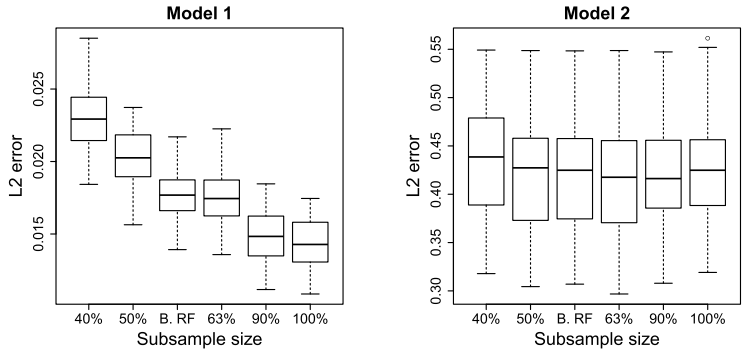


FIGURE A.6. Standard Breiman forests *versus* several subsampled Breiman forests.

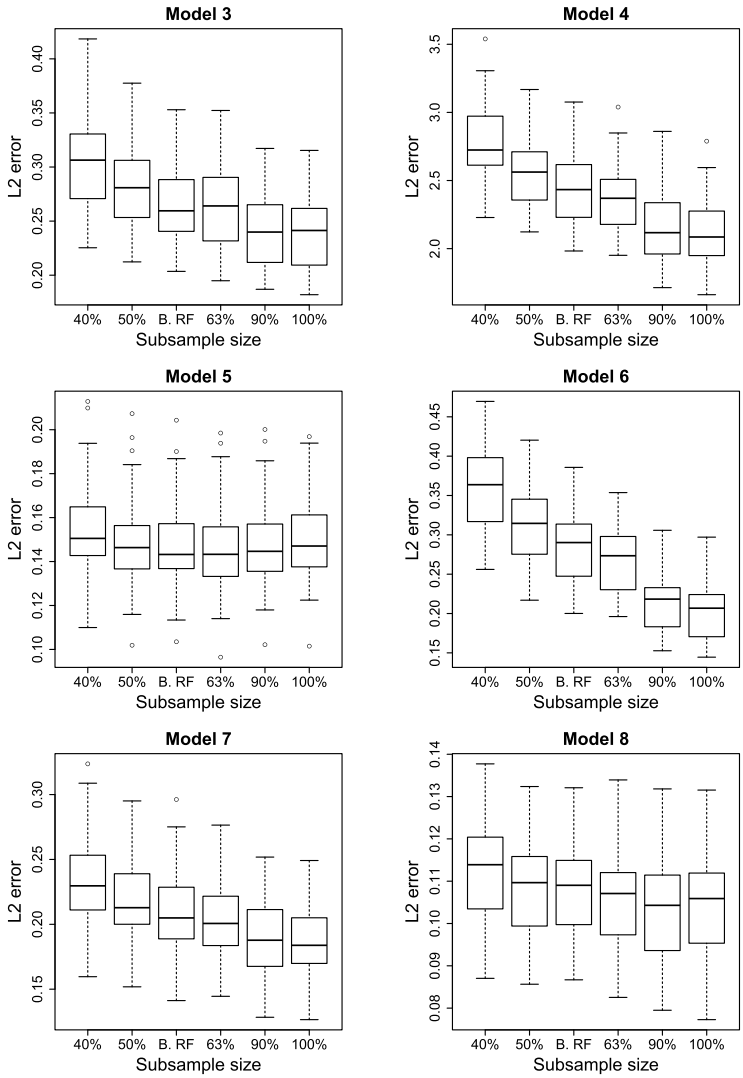


FIGURE A.6. (Continued).

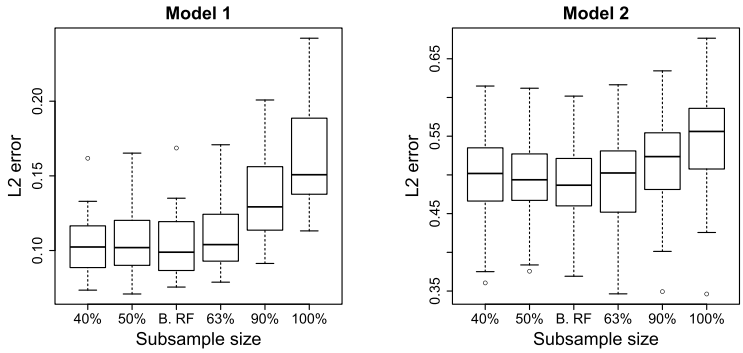


FIGURE A.7. Standard Breiman forests *versus* several subsampled Breiman forests (noisy models).

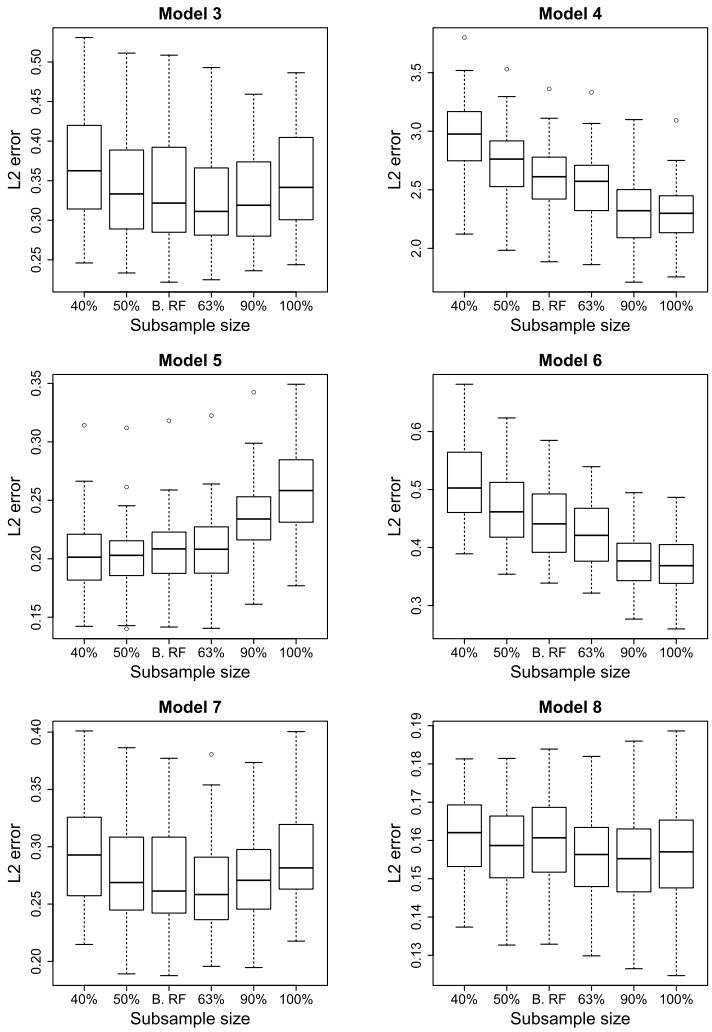


FIGURE A.7. (Continued).

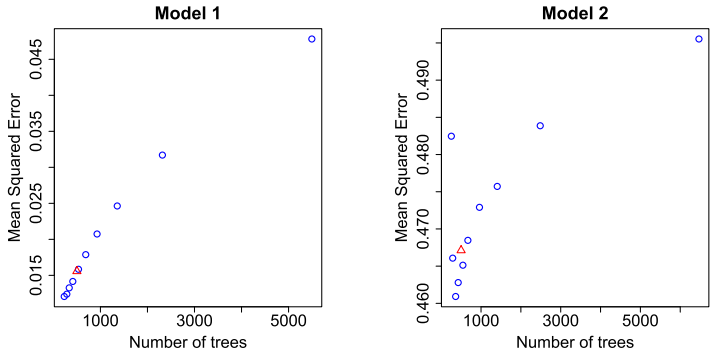


FIGURE A.8. Performance of subsampled Breiman's forests (in blue) with same computation cost as Breiman's forests (in red).

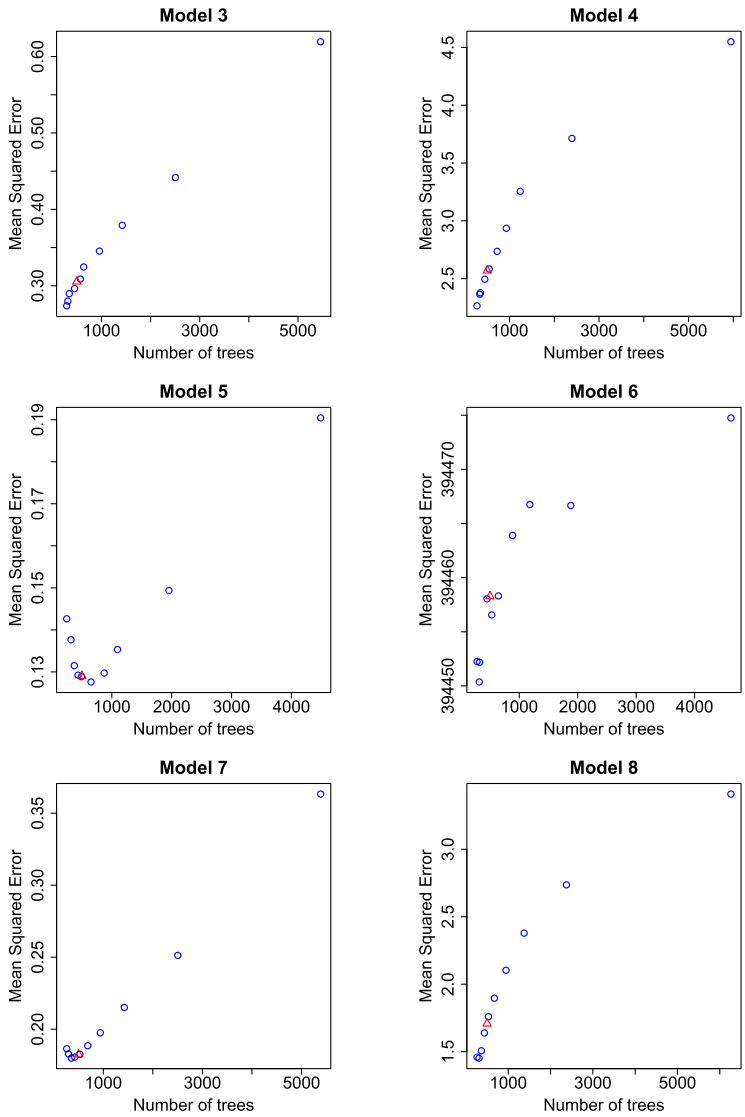


FIGURE A.8. (Continued).

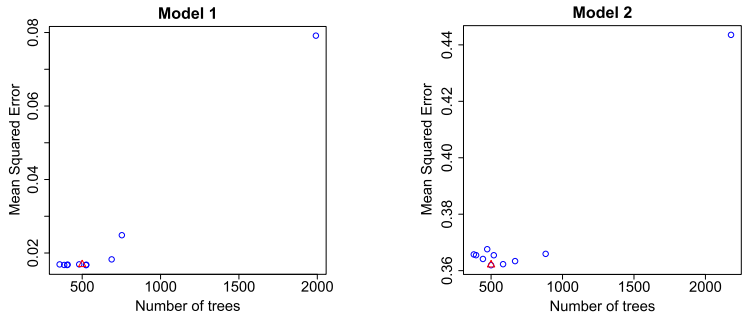


FIGURE A.9. Performance of small-tree Breiman's forests (in blue) with same computation cost as Breiman's forests (in red).

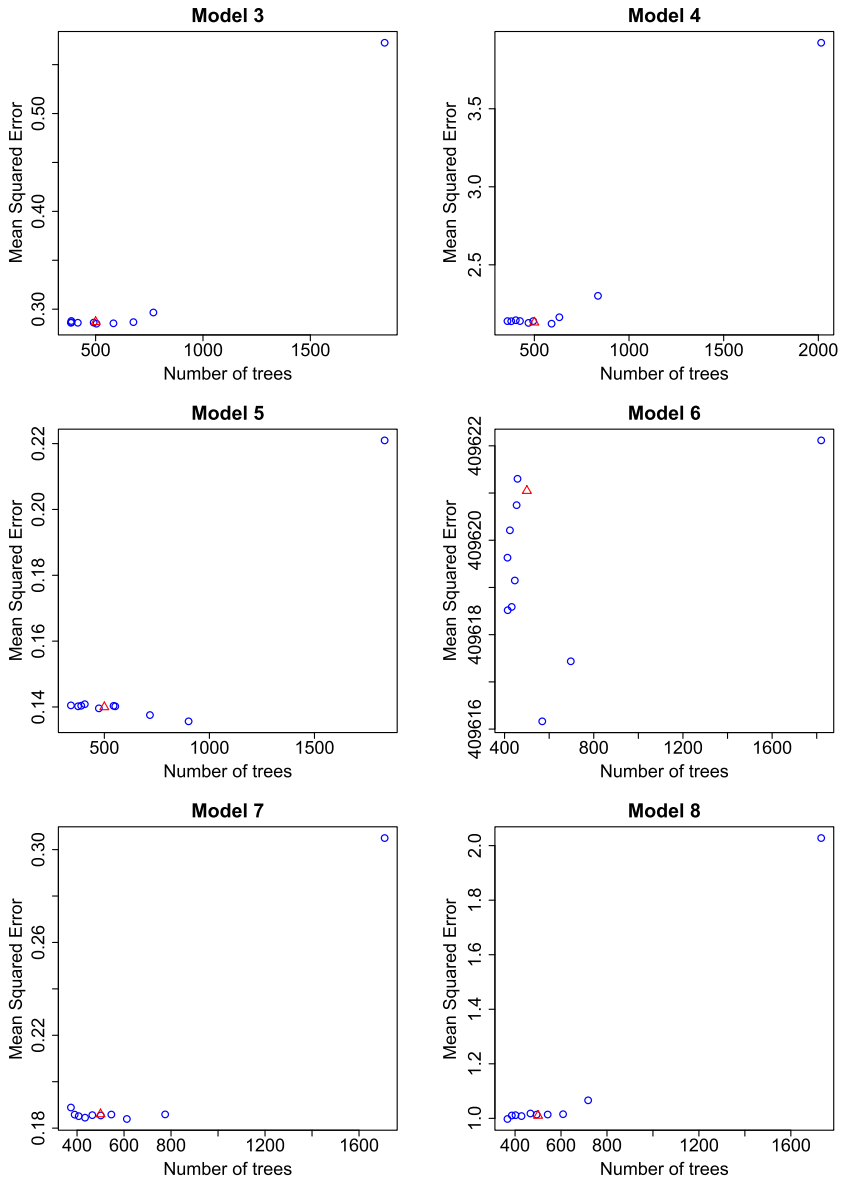


FIGURE A.9. (Continued).

Acknowledgements. We greatly thank the two referees for valuable comments and insightful suggestions.

REFERENCES

[1] S. Arlot and R. Genuer, Analysis of Purely Random Forests Bias. Preprint arXiv:1407.3939 (2014).
[2] G. Biau, Analysis of a random forests model. *J. Mach. Learn. Res.* **13** (2012) 1063–1095.
[3] G. Biau and L. Devroye, Cellular tree classifiers, in Algorithmic Learning Theory. Springer, Cham (2014) 8–17.
[4] G. Biau, L. Devroye and G. Lugosi, Consistency of random forests and other averaging classifiers. *J. Mach. Learn. Res.* **9** (2008) 2015–2033.
[5] L. Breiman, Random forests. *Mach. Learn.* **45** (2001) 5–32.

[6] L. Breiman, J.H. Friedman, R.A. Olshen and C.J. Stone, Classification and Regression Trees. Chapman & Hall, CRC, Boca Raton (1984).
[7] P. Bühlmann, Bagging, boosting and ensemble methods, in Handbook of Computational Statistics. Springer, Berlin, Heidelberg (2012) 985–1022.
[8] M. Denil, D. Matheson and N. de Freitas, Consistency of Online Random Forests. Vol. 28 of *Proc. of ICML'13 Proceedings of the 30th International Conference on International Conference on Machine Learning*, Atlanta, GA, USA June 6–21 (2013) 1256–1264.
[9] M. Denil, D. Matheson and N. de Freitas, Narrowing the gap: random forests in theory and in practice, in *International Conference on Machine Learning (ICML)* (2014).
[10] L. Devroye, L. Györfi and G. Lugosi, A Probabilistic Theory of Pattern Recognition. Springer, New York (1996).
[11] R. Díaz-Uriarte and S. Alvarez de Andrés, Gene selection and classification of microarray data using random forest. *BMC Bioinform.* **7** (2006) 1–13.
[12] M. Fernández-Delgado, E. Cernadas, S. Barro and D. Amorim, Do we need hundreds of classifiers to solve real world classification problems. *J. Mach. Learn. Res.* **15** (2014) 3133–3181.
[13] R. Genuer, Variance reduction in purely random forests. *J. Nonparametric Stat.* **24** (2012) 543–562.
[14] R. Genuer, J. Poggi and C. Tuleau-Malot, Variable selection using random forests. *Pattern Recognit. Lett.* **31** (2010) 2225–2236.
[15] H. Ishwaran and U.B. Kogalur, Consistency of random survival forests. *Stat. Probab. Lett.* **80** (2010) 1056–1064.
[16] L. Meier, S. Van de Geer and P. Bühlmann, High-dimensional additive modeling. *Ann. Stat.* **37** (2009) 3779–3821.
[17] L. Mentch and G. Hooker, Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *J. Mach. Learn. Res.* **17** (2015) 841–881.
[18] Y. Qi, Random forest for bioinformatics, in Ensemble Machine Learning. Springer, Boston, MA (2012) 307–323.
[19] G. Rogez, J. Rihan, S. Ramalingam, C. Orrite and P. H. Torr, Randomized trees for human pose detection, in IEEE Conference on Computer Vision and Pattern Recognition (2008) 1–8.
[20] M. Sabzevari, G. Martínez-Muñoz and A. Suárez, Improving the Robustness of Bagging with Reduced Sampling Size. Université catholique de Louvain (2014).
[21] E. Scornet, On the asymptotics of random forests. *J. Multivar. Anal.* **146** (2016) 72–83.
[22] E. Scornet, G. Biau and J.-P. Vert, Consistency of random forests. *Ann. Stat.* **43** (2015) 1716–1741.
[23] C.J. Stone, Optimal rates of convergence for nonparametric estimators. *Ann. Stat.* **8** (1980) 1348–1360.
[24] C.J. Stone, Optimal global rates of convergence for nonparametric regression. *Ann. Stat.* **10** (1982) 1040–1053.
[25] M. van der Laan, E.C. Polley and A.E. Hubbard, Super learner. *Stat. Appl. Genet. Mol. Biol.* **6** (2007).
[26] S. Wager, Asymptotic Theory for Random Forests. Preprint arXiv:1405.0352 (2014).
[27] S. Wager and S. Athey, Estimation and inference of heterogeneous treatment effects using random forests. *J. Am. Stat. Assoc.* (2018) 1–15.
[28] S. Wager and G. Walther., Adaptive Concentration of Regression Trees, With Application to Random Forests (2015).
[29] F. Zaman and H. Hirose, Effect of subsampling rate on subbagging and related ensembles of stable classifiers, in *International Conference on Pattern Recognition and Machine Intelligence*. Springer (2009) 44–49.