ESAIM: PROCEEDINGS AND SURVEYS, Vol. ?, 2017, 1-10
Editors: Will be set by the publisher

TUNING PARAMETERS IN RANDOM FORESTS

Erwan Scornet[1]

Abstract. Breiman's (2001) random forests are a very popular class of learning algorithms often able to produce good predictions even in high-dimensional frameworks, with no need to accurately tune its inner parameters. Unfortunately, there are no theoretical findings to support the default values used for these parameters in Breiman's algorithm. The aim of this paper is therefore to present recent theoretical results providing some insights on the role and the tuning of these parameters.

1. Introduction

Random forests, designed by Breiman [8], belong to the class of learning algorithms and are among the most used aggregation schemes which predict by combining several weak learners (decision trees in the case of random forests). Facing a single data set, a number of different trees are built by introducing randomness into the initial tree construction process. Random forest prediction is then computed as the average over all tree predictions.

Because random forests appear to have the ability to detect relevant features even in noisy environments, they are very convenient when dealing with high-dimensional feature spaces and hence are often implemented in fields such as chemoinformatics [28], ecology [12, 23], 3D object recognition [26], and bioinformatics [13], just to name a few.

In details, the random forest construction in a regression setting proceeds as follows. Each tree is built using a sample of size a_n drawn from the original data set (either with or without replacement). Only these a_n observations are used to construct the tree partition and to ultimately make the tree prediction. Once the observations have been selected, the algorithm forms a recursive partitioning of the covariates space. In each cell, a number `mtry` of variables are selected uniformly at random among all covariates. Then, the best split is chosen as the one optimizing the CART splitting criterion (details are given in Section 2) only along the `mtry` preselected directions. Equivalently, the procedure selects the split minimizing the quadratic risk of the tree estimate at each step. This process is repeated until each cell contains less than a prespecified number `nodesize` of observations. After tree partition has been completed, the prediction at a new point is computed by averaging observations falling into the cell of the new point. Then each one of the M trees in the forest gives a prediction, and the forest prediction is simply the average of the M predicted values.

All in all, the algorithm depends on several parameters: the subsample size a_n, the number `mtry` of preselected directions for splitting, the tree depth which can be specified in different ways (`nodesize`, or the maximal number of cells `maxnodes`, or the tree level k_n) and the number M of trees.

It is of common belief that implemented default values for these parameters yield good empirical performance in prediction, which is partially why random forests became so popular. One major related drawback is the total absence of theoretical justification for these default values. Whereas the very specific default values $\texttt{mtry} = \sqrt{d}$ (classification setting) and $\texttt{mtry} = d/3$ (regression setting) can be thought of as the consequence of in-depth

[1] CMAP, École Polytechnique, Route de Saclay, 91128 Palaiseau

mathematical analysis, they result in reality from several simulations designed by Breiman in the seminal paper [8]. There is no more theoretical supports for choosing `nodesize` = 1 or `nodesize` = 5, or $a_n = n$ (which corresponds to bootstrap procedure if sampling is done with replacement), nor using $M = 500$ decision trees as default values in the `R` package `randomForest`. Because random forests performance can depend on parameter values and thus can be improved by a proper tuning of these parameters, there is a real need to apprehend their influence on random forest predictions.

The goal of this paper is to present how existing theoretical results provide insights about how to choose parameters in random forests procedure. We do not intend to review exhaustively existing literature on the generic topic of random forests; the interested reader can refer to the survey by [11] and [7] for applied aspects of random forests and to [6] (and the discussions therein) for an overview of theoretical results. In Section 2 of the present paper, we introduce mathematical notations for random forest estimates and briefly discuss the impact of `mtry`. Section 3 focuses on the influence of the number M of trees. Section 4 is devoted to the connection between the tree depth (`nodesize`, `maxnodes` or k_n) and the subsample size a_n.

2. Random forest algorithm and `mtry` parameter

In this paper, we consider a regression framework and assume to be given 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, by Y the response variable and our objective is to estimate the regression function $m(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$. In this context, we use random forests to construct an estimate $m_n : [0,1]^d \to \mathbb{R}$ of m, based on the data set $\mathcal{D}_n$.

2.1. Random forest algorithm

Random forest is a generic term to name an aggregation scheme of decision trees. Due to its popularity and its good empirical performance, Breiman's (2001) forest is one of the most used random forest algorithms, and are subsequently called by some abuse of terms "random forests". Whereas our major focus will be on Breiman's forests, we will have to consider other forest variants. To avoid confusion, we will explicitly mention when we refer to specific forests different from Breiman's forests.

Breiman's forests are based on a collection of M randomized trees. We denote by $m_n(\mathbf{x}, \Theta_j, \mathcal{D}_n)$ the predicted value at point $\mathbf{x}$ given by the j-th tree, where $\Theta_1, \ldots, \Theta_M$ are independent random variables, distributed as a generic random variable Θ, independent of the sample $\mathcal{D}_n$. In practice, the variable Θ contains indexes of observations that are used to build each tree and indexes of splitting candidate directions in each cell. Thus Θ belongs to a large space, which can vary if different randomization processes are at stake, i.e. if forests other than Breiman's are considered. The predictions of the M randomized trees are then averaged to obtain the random forest prediction

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

To lighten notation, when there is no ambiguity, we will omit the explicit dependence on $\mathcal{D}_n$ and simply write, for instance, $m_n(\mathbf{x}, \Theta_m)$.

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 $[0,1]^d$ (see, Figure 1 for an illustration). The procedure is described in Algorithm 1.

So far, we have not made explicit the CART-split criterion used in Algorithm 1. To properly define it, we let A be a generic cell and $N_n(A)$ be the number of data points falling in A. A cut in A is a pair (j, z), where j is a dimension in $\{1, \ldots, d\}$ and z is the position of the cut along the j-th coordinate, within the limits of

Algorithm 1: Breiman's forests algorithm

(1) Grow M trees as follows:
  (a) Prior to the j-th 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 `mtry` coordinates among $\{1, \ldots, d\}$.
  (d) Select the split maximizing the CART-split criterion (see below for details) along the pre-selected `mtry` 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 j-th tree outputs the average 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 of the predictions given by the M trees.

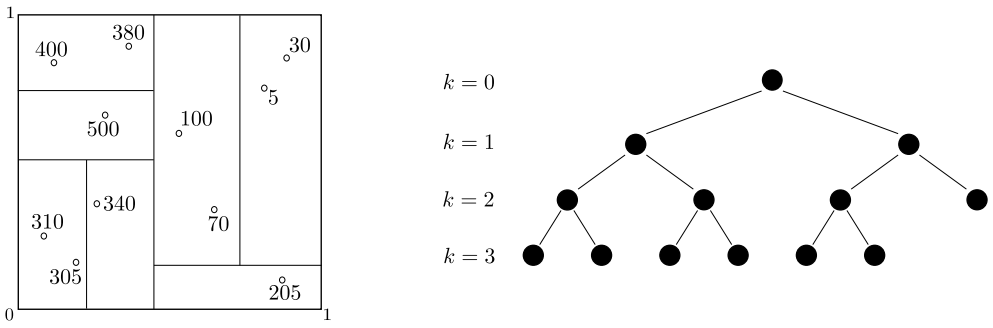

FIGURE 1. Example of a single tree in Breiman forests ($d = 2$)

A. We let $\mathcal{C}_A$ be the set of all such possible cuts in A. Then, with the notation $\mathbf{X}_i = (\mathbf{X}_i^{(1)}, \ldots, \mathbf{X}_i^{(d)})$, for any $(j, z) \in \mathcal{C}_A$, the CART-split criterion [10] takes the form

$$L_n(j,z) = \frac{1}{N_n(A)} \sum_{i=1}^{n} (Y_i - \bar{Y}_A)^2 \mathbb{1}_{\mathbf{X}_i \in A} - \frac{1}{N_n(A)} \sum_{i=1}^{n} (Y_i - \bar{Y}_{A_L} \mathbb{1}_{\mathbf{X}_i^{(j)} < z} - \bar{Y}_{A_R} \mathbb{1}_{\mathbf{X}_i^{(j)} \geq z})^2 \mathbb{1}_{\mathbf{X}_i \in A}, \tag{2}$$

where $A_L = \{\mathbf{x} \in A : \mathbf{x}^{(j)} < z\}$, $A_R = \{\mathbf{x} \in A : \mathbf{x}^{(j)} \geq z\}$, and $\bar{Y}_A$ (resp., $\bar{Y}_{A_L}$, $\bar{Y}_{A_R}$) is the average of the Y_i's belonging to A (resp., A_L, A_R), with the convention $0/0 = 0$. At each cell A, the best cut $(j_n^\star, z_n^\star)$ is finally selected by maximizing $L_n(j,z)$ over $\mathcal{M}_{\text{try}}$ and $\mathcal{C}_A$, that is

$$(j_n^\star, z_n^\star) \in \underset{\substack{j \in \mathcal{M}_{\text{try}} \\ (j,z) \in \mathcal{C}_A}}{\arg\max} L_n(j,z).$$

To remove ties in the arg max, the best cut is always performed at the middle of two consecutive data points, along the selected direction $j_n^\star$. Note that selecting the split that minimizes the CART-split criterion is equivalent to selecting the split such that the local mean square error (second term in (2)) is minimal.

The whole procedure depends on four parameters: the number M of trees, the number a_n of observations used in each tree, the number `mtry` of preselected 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, that is sampling is done with replacement), `mtry` $= d/3$ and `nodesize`$= 5$.

2.2. **Number of candidate variables for splitting**

The first parameter we focus on is the number `mtry` of preselected directions along which the splitting criterion is optimized. To the best of our knowledge, there is no theoretical results highlighting the benefits of randomizing the eligible directions for splitting. From a heuristical point of view, introducing additional randomness via the parameter `mtry` yield more tree diversity. Indeed, if we set `mtry` $= 1$, the splitting variable is chosen uniformly at random among the d initial variables. On the other hand, if `mtry` $= d$, the split is optimized along all possible directions, which means that up to the randomness induced by subsampling, the tree construction is deterministic (given the original data set).

There is also a computational benefit of employing `mtry` $< d$: the splitting criterion is computed along less than `mtry` $< d$ covariates therefore making the algorithm faster than the one where exhaustive search is performed [`mtry` $= d$ as in original CART procedure, see 10]. This is of particular interest in high-dimensional problems, where exhaustive search is prohibited due to computational cost.

From a methodological perspective, the interested reader can refer to [13] and [16] who showed in different contexts that the default value of `mtry` is either optimal or too small. Note also there are implementations of random forests in which `mtry` can be automatically determined [see for example 2].

3. Number of trees

The aim of this section is to study the impact of the number M of trees on the statistical performance of random forests.

3.1. **Risk of finite and infinite forests**

The number of trees in random forests is only used to limit the difference between the finite forest defined by (1) and the infinite forest defined as

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

where $\mathbb{E}_\Theta$ denotes the expectation with respect to the random variables Θ (conditional on all other random variables). Indeed, the law of large numbers states that, conditional on $\mathcal{D}_n$, for any fixed $\mathbf{x} \in [0,1]^d$, almost surely,

$$m_{M,n}(\mathbf{x}, \Theta_1, \ldots, \Theta_M) \underset{M \to \infty}{\to} m_{\infty,n}(\mathbf{x}), \tag{3}$$

The finite forest estimate is nothing but a Monte Carlo approximation of the infinite forest, whose computation is infeasible in practice since it would require knowledge about the dependence of $m_n(\mathbf{x}, \Theta)$ on Θ. To see the influence of M, let us define the risk of $m_{\infty,n}$ as

$$R(m_{\infty,n}) = \mathbb{E}[m_{\infty,n}(\mathbf{X}) - m(\mathbf{X})]^2,$$

and the risk of $m_{M,n}$ as

$$R(m_{M,n}) = \mathbb{E}[m_{M,n}(\mathbf{X}, \Theta_1, \ldots, \Theta_M) - m(\mathbf{X})]^2.$$

As for any other aggregated estimate, the parameter M controls the variance of the Monte Carlo approximation:

$$\begin{aligned}&\mathbb{E}\big[m_{M,n}(\mathbf{X},\Theta_1,\ldots,\Theta_M)-m(\mathbf{X})\big]^2\\&=\underbrace{\mathbb{E}\big[m_{M,n}(\mathbf{X},\Theta_1,\ldots,\Theta_M)-m_{\infty,n}(\mathbf{X})\big]^2}_{\text{Monte Carlo error}}+\underbrace{\mathbb{E}\big[m_{\infty,n}(\mathbf{X})-\mathbb{E}[m_{\infty,n}(\mathbf{X})]\big]^2}_{\text{Estimation error of } m_{\infty,n}}+\underbrace{\mathbb{E}\big[\mathbb{E}[m_{\infty,n}(\mathbf{X})]-m(\mathbf{X})\big]^2}_{\text{Approximation error of } m_{\infty,n}}.\end{aligned}\quad(4)$$

In the light of this result, the larger M, the more accurate the prediction will be in term of mean squared error. Thus, M should be chosen large enough to reach the desired statistical precision and small enough to make the calculations feasible (the computational cost increases linearly with M). Equation (4) presents a decomposition of the mean squared error, and thus for a particular instance of forests, the error can increase by adding a single tree. Luckily, the error will decrease on average with M as precisely stated in Theorem 3.1. As we did above regarding expectation, we will denote by $\mathbb{V}_\Theta$ the variance with respect to Θ.

Theorem 3.1 ([24]). *Assume that $Y = m(\mathbf{X}) + \varepsilon$, where ε is a centered Gaussian noise with finite variance σ^2, independent of $\mathbf{X}$, and $\|m\|_\infty = \sup_{\mathbf{x}\in[0,1]^d} |m(\mathbf{x})| < \infty$. Then, for all $M, n \in \mathbb{N}^\star$,*

$$R(m_{M,n}) = R(m_{\infty,n}) + \frac{\mathbb{E}\Big[\mathbb{V}_\Theta\left[m_n(\mathbf{X},\Theta)\right]\Big]}{M}.$$

In particular,

$$0 \le R(m_{M,n}) - R(m_{\infty,n}) \le \frac{8}{M} \times \Big(\|m\|_\infty^2 + \sigma^2(1+4\log n)\Big).$$

Theorem 3.1 reveals that the risk of infinite forests is lower than the risk of finite forests. Since infinite random forests cannot be computed, Theorem 3.1 should be seen as a way to ensure that $R(m_{M,n})$ is close to $R(m_{\infty,n})$ provided the number of trees is large enough. Indeed, under assumptions of Theorem 3.1, $R(m_{M,n}) - R(m_{\infty,n}) \le \varepsilon$ if

$$M \ge \frac{8(\|m\|_\infty^2 + \sigma^2)}{\varepsilon} + \frac{32\sigma^2 \log n}{\varepsilon}.$$

Note that the "$\log n$" term comes from the Gaussian noise assumption and in all generality, the previous bound can be rewritten as

$$M \ge \frac{8\|m\|_\infty^2}{\varepsilon} + \frac{8}{\varepsilon}\mathbb{E}\left[\max_{1\le i\le n} \varepsilon_i^2\right].$$

Another interesting consequence of Theorem 3.1 is that if $M/\log n \to \infty$ as $n \to \infty$, finite random forests are consistent as soon as infinite random forests are. This allows to derive consistency results for finite forests based on results about infinite forests [see, e.g., 5, 21]. Since the "$\log n$" factor depends on the noise assumption, the asymptotic $M/\log n \to \infty$ changes with the type of noise and, for instance, turns into $M \to \infty$ if the noise is bounded.

3.2. Distribution of finite forests, conditional on $\mathcal{D}_n$

Now that we have assessed how the forest accuracy depends on the number of trees, we take a closer look at the forest predictions and try to determine their distribution. In other words, we examine the asymptotic behavior of the finite forest estimate $m_{M,n}(\bullet, \Theta_1, \ldots, \Theta_M)$ as M tends to infinity. To do so, let us assume for now that the data set $\mathcal{D}_n$ is fixed: this setting is consistent with practical problems, where observations are fixed, and one can grow as many trees as possible. Theorem 3.2 extends the pointwise convergence in (3) to the convergence of the whole functional estimate $m_{M,n}(\bullet, \Theta_1, \ldots, \Theta_M)$, towards the functional estimate $m_{\infty,n}(\bullet)$.

Theorem 3.2 ([24]). *Consider a finite Breiman's forest formed with M trees. Then, conditional on $\mathcal{D}_n$, almost surely, for all $\mathbf{x} \in [0,1]^d$, we have*

$$m_{M,n}(\mathbf{x}, \Theta_1, \ldots, \Theta_M) \underset{M\to\infty}{\to} m_{\infty,n}(\mathbf{x}).$$

Since the set $[0,1]^d$ is not countable, we cannot reverse the "almost sure" and "for all $\mathbf{x} \in [0,1]^d$" statements in (3). Thus, Theorem 3.2 is not a direct consequence of (3). Theorem 3.2 is a first step to prove that infinite forest estimates can be uniformly approximated by finite forest estimates. To pursue the analysis, a natural question is to determine the rate of convergence in Theorem 3.2. The pointwise rate of convergence is provided by the central limit theorem which states that, conditional on $\mathcal{D}_n$, for all $\mathbf{x} \in [0,1]^d$,

$$\sqrt{M}\big(m_{M,n}(\mathbf{x}, \Theta_1, \ldots, \Theta_M) - m_{\infty,n}(\mathbf{x})\big) \underset{M\to\infty}{\overset{d}{\to}} \mathcal{N}\big(0, \tilde{\sigma}^2(\mathbf{x})\big), \tag{5}$$

where $\tilde{\sigma}^2(\mathbf{x}) = \mathbb{V}_\Theta\left[m_n(\mathbf{x}, \Theta)\right] \leq 4 \max_{1\leq i\leq n} Y_i^2$ (as before, $\mathbb{V}_\Theta$ denotes the variance with respect to Θ, conditional on $\mathcal{D}_n$). Theorem 3.3 extends the pointwise convergence in distribution in (5) to the convergence in distribution of the random process

$$m_{M,n}(\bullet, \Theta_1, \ldots, \Theta_M) - m_{\infty,n}(\bullet) = \frac{1}{M}\sum_{m=1}^{M} m_n(\bullet, \Theta_m) - \mathbb{E}_\Theta\left[m_n(\bullet, \Theta)\right].$$

Theorem 3.3 ([24]). *Consider a finite Breiman's forest with M trees. Then, conditional on $\mathcal{D}_n$,*

$$\sqrt{M}\big(m_{M,n}(\bullet, \Theta_1, \ldots, \Theta_M) - m_{\infty,n}(\bullet)\big) \overset{\mathcal{L}}{\to} \mathbb{G}g_\bullet.$$

where $\mathbb{G}$ is a Gaussian process with mean zero and covariate function

$$\mathrm{Cov}_\Theta(\mathbb{G}g_{\mathbf{x}}, \mathbb{G}g_{\mathbf{z}}) = \mathrm{Cov}_\Theta\left(m_n(\mathbf{x}, \Theta), m_n(\mathbf{z}, \Theta)\right).$$

In other words, Theorem 3.2 states that almost surely, the finite forest converges to the infinite forest and Theorem 3.3 states that the corresponding limiting process is Gaussian. This last result provides theoretical foundations to use confidence bounds for the whole forest estimate. Previous theorems are also valid for other type of random forests [for details, see 24]

3.3. **Relation between number of trees and subsampling**

To go further into the analysis of forest predictions, Mentch and Hooker [22] considered the framework where both the size of the data set and the number M_n of trees tends to infinity. To formulate their result, we consider trees with no extra randomness (and thus omit the random variable Θ) built with $a \leq n$ observations, and define for any fixed $\mathbf{x} \in [0,1]^d$,

$$\zeta_{1,a} = \mathrm{Cov}(m_n(\mathbf{x}, \{Z_1, \ldots, Z_a\}), m_n(\mathbf{x}, \{Z_1, Z_2', \ldots, Z_a'\})), \tag{6}$$

where $Z_i' = (X_i', Y_i')$ is an independent copy of $Z_i = (X_i, Y_i)$. For any $c \in \{1, \ldots, a\}$, $\zeta_{c,a}$ has the same expression as (6) but with c observations in common. Theorem 3.4 gives the limiting distribution of forest prediction at a fixed point $\mathbf{x}$ in different regimes. It relies on the assumption that forest predictions do not vary too much when one single observation label in the training set is slightly modified.

Theorem 3.4 ([22]). *Let us consider a random forests whose randomization process entirely lies in sampling the data set, where each tree is built with a_n observations. and set $\alpha = \lim n/M_n$. Assume that the regression*

function m is bounded and that there exists a constant $c > 0$ such that for all $a_n \geq 1$, almost surely,

$$\left|m_n\Big(\mathbf{x}, \Big\{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_{a_n+1}, Y_{a_n+1})\Big\}\Big) - m_n\Big(\mathbf{x}, \Big\{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_{a_n+1}, Y^\star_{a_n+1})\Big\}\Big)\right| \leq c|Y_{a_n+1} - Y^\star_{a_n+1}|,$$

where $Y_{a_n+1} = m(X_{a_n+1}) + \varepsilon_{a_n+1}$, $Y^\star_{a_n+1} = m(X_{a_n+1}) + \varepsilon^\star_{a_n+1}$, $\varepsilon^\star_{a_n+1}$ is an independent copy of ε_{a_n+1} with exponential tail. Assume also that $\lim \zeta_{1,a_n} \neq 0$, $\lim a_n/\sqrt{n} = 0$ and

$$\lim_{n\to\infty} \mathbb{E}\big[m_n(\mathbf{x}, \Theta, \mathcal{D}_n) - m_{\infty,n}(\mathbf{x}, \Theta, \mathcal{D}_n)\big]^2 \neq \infty.$$

Then, we have three different regimes:

(1) If $\alpha = 0$, then

$$\frac{\sqrt{n}(m_{M_n,n}(\mathbf{x}, \Theta, \mathcal{D}_n) - \mathbb{E}[m_{M_n,n}(\mathbf{x}, \Theta, \mathcal{D}_n)])}{a_n\sqrt{\zeta_{1,a_n}}} \xrightarrow{d} \mathcal{N}(0, 1).$$

(2) If $\alpha = \infty$, then

$$\frac{\sqrt{M_n}(m_{M_n,n}(\mathbf{x}, \Theta, \mathcal{D}_n) - \mathbb{E}[m_{M_n,n}(\mathbf{x}, \Theta, \mathcal{D}_n)])}{a_n\sqrt{\zeta_{1,a_n}}} \xrightarrow{d} \mathcal{N}(0, 1).$$

(3) If $0 < \alpha < \infty$, then

$$\frac{\sqrt{M_n}(m_{M_n,n}(\mathbf{x}, \Theta, \mathcal{D}_n) - \mathbb{E}[m_{M_n,n}(\mathbf{x}, \Theta, \mathcal{D}_n)])}{\sqrt{\frac{a_n^2\zeta_{1,a_n}}{\alpha} + \zeta_{a_n,a_n}}} \xrightarrow{d} \mathcal{N}(0, 1).$$

Theorem 3.4 describes the limiting distributions of the finite forest, assuming that the subsample size $a_n = o(\sqrt{n})$. The three different regimes correspond to different values of the ratio n/M_n. It is one of the first results that describes the behaviour of Breiman's forest as a function of more than one parameter, here the subsample size a_n and the number of trees M_n. Theorem 3.4 can be extended to general random forest, where eligible directions for splitting are also randomized [see 22]. Besides, it is worth noticing that there exist procedures to estimate the forest variance ζ_{1,a_n} [see 22, 31].

In brief, finite random forests defined by (1) are nothing but Monte Carlo approximation of infinite forests. Thus finite random forest estimate is asymptotically normal (Theorems 3.3 and 3.4) which allows us to build confidence intervals for random forest predictions [see 22, 31]. Besides, Theorem 3.1 states that the risk decreases with M therefore leading to choose the highest possible value for M given the computational power available. Besides, assuming Gaussian noise, taking M of order $(\log n)^{1+\rho}$ for any nonnegative ρ is sufficient to ensure that the risk of finite forest converges to the risk of the infinite forest.

4. Tree depth and subsampling

Tree shape plays a great part in random forests performance. One key component is the stopping rule, which specifies when to stop the splitting procedure in each cell. As mentioned in Section 2, the original stopping rule in Breiman's algorithm is to stop tree expansion when each cell contains less than `nodesize` observations. A closer look at the `R` package `randomForest` shows that the parameter `maxnodes` can also be used (with `nodesize` or alone) to stop the tree construction. In that case, the tree construction will end when the number of terminal nodes reaches `maxnodes`. We study in this section the influence of tree depth (controlled via parameters `nodesize`, `maxnodes` or tree level k_n) and subsampling on predictive performance.

4.1. **Purely random forests**

Algorithm 2: Centred tree

(1) Consider the whole data set $\mathcal{D}_n$. This entire set will be used in the tree construction.
(2) Consider the root cell $[0,1]^d$.
(3) Select uniformly one coordinate among $\{1, \ldots, d\}$.
(4) Cut the cell at the center of the cell along the preselected direction.
(5) Repeat $(c) - (d)$ for the two resulting cells until each cell has been cut exactly k_n times.
(6) For a query point $\mathbf{x}$, the centred tree estimate outputs the average of the Y_i falling into the same cell as $\mathbf{x}$.

Before considering the entire forest, let us first analyze a single tree. One of the simplest tree to construct is the centred tree, described in Algorithm 2. A centred tree depends on one parameter, the tree level k_n, which models tree shape: the construction stops when each cell has been split k_n times exactly. There is no subsampling parameter since the entire data set is used for each tree.

General results on partitioning estimate whose construction is independent of the data set state that necessary conditions for the consistency of centred trees are $n/2^{k_n} \to \infty$ and $k_n \to \infty$, as $n \to \infty$ [see, e.g. 5]. Indeed, the tree level k_n should grow to infinity with n so that the approximation error decreases: tree estimate approximates the regression function with piecewise constant function in each cell of the partition, which becomes a satisfactory approximation when cell diameters shrink to zero. On the other hand, there should be a large number of observations in each cell so that the estimation error is low. This is a consequence of $n/2^{k_n} \to \infty$ which forces the mean number of observations per cell to tend to infinity. Each tree satisfying these two assumptions is consistent, and thus the forest composed of such trees is consistent too.

The first analysis of random forests was by [9] who found the rate of consistency of centred forest. Later on, [3] extended this result assuming that splits are likely to concentrate around the S informative variables of the model. In this particular setting, the upper bound on the generalization error is given by

$$R(m_{\infty,n}, m) \leq c_1 n^{-3/(4S \log 2 + 3)}, \tag{7}$$

where c_1 is some positive constant, if 2^{k_n} is chosen of order $n^{4S \log 2/(4S \log 2+3)}$. The fact that the upper bound (7) depends only on S and not on the ambient dimension d can explain why random forests perform particularly well in high dimensions, under the assumption that splits concentrate along relevant features. Other purely random forests (whose construction is independent of the training set) has also been studied, for example by [1] who established the rate of convergence of their bias.

Analyzing such forests is not sufficient to understand the outstanding performance of Breiman's forests in high dimensions. Besides, there is still a huge gap between these forests and Breiman forests regarding the number of observations in each cell: cells in Breiman's forest contain less than five observations whereas cells of purely forests contain a large number of data points. A more in-depth analysis is then required to take into account cases where $n/2^{k_n} \simeq 1$.

4.2. **Subsampling and tree depth in median forests**

To deal with deeper trees, and thus to get closer to Breiman's forests, we study in this section median forest [see, for example, 4, for details on median tree]. Median forests construction depends on the X_i making them a good tradeoff between the complexity of Breiman's (2001) forests and the apparent simplicity of purely random forests. Besides, median forests can be tuned such that each leaf of each tree contains exactly one observation, thus being extremely close to Breiman's forests.

We now describe the construction of median forest. In the spirit of Breiman's (2001) algorithm, before growing each tree, data are subsampled: $a_n < n$ points are selected without replacement. Then, each split is

performed at the empirical median along a coordinate, chosen uniformly at random among the d coordinates. 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. Finally, the algorithm stops when each cell has been cut exactly k_n times; k_n is called the tree level. The overall construction process is detailed in Algorithm 3.

Algorithm 3: Median random forests

(1) Grow M trees as follows:
   (a) Prior to the j-th 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 among $\{1, \ldots, d\}$.
   (d) Cut the cell at the empirical median of the X_i falling into the cell, along the preselected direction.
   (e) Repeat $(c) - (d)$ for the two resulting cells until each cell has been cut exactly k_n times.
   (f) For a query point $\mathbf{x}$, the j-th tree outputs the average of the Y_i falling into the same cell as $\mathbf{x}$.
(2) For a query point $\mathbf{x}$, the median forest $m_{M,n}$ outputs the average of the predictions given by the M trees.

The analysis of several random forests can be reduced to that of median forests (or more generally to quantile forests) if their construction depends only on the X_i and if splits do not separate a small fraction of data points from the rest of the sample. This last assumption is true, for example, if $\mathbf{X}$ has a density on $[0,1]^d$ bounded from below and from above, and if some splitting rule forces splits to be performed far from the cell edges. This assumption is explicitly made in the analysis of [21] and [30] to ensure that cell diameters tend to zero as $n \to \infty$, which is a necessary condition to prove the consistency of partitioning estimates, whose construction is independent of the label in the training set [see Chapter 4 in 17]. Nevertheless, there are no results stating that splits in Breiman's (2001) forests are performed far from the cell edges [see 19, for an analysis of the splitting criterion in Breiman's forests]. Theorem 4.1 presents a first consistency result for forests of fully grown median trees.

Theorem 4.1 ([24]). *Assume that $Y = m(\mathbf{X}) + \varepsilon$, where ε is a centred noise such that $\mathbb{V}[\varepsilon|\mathbf{X} = \mathbf{x}] \leq \sigma^2$, where $\sigma^2 < \infty$ is a constant, $\mathbf{X}$ has a density on $[0,1]^d$ and m is continuous. Consider a median forest where k_n is chosen such that each cell contains one or two observations. Then, providing $a_n \to \infty$ et $a_n/n \to 0$, the infinite median random forest is consistent, that is $R(m_{\infty,n}) \to 0$ as $n \to \infty$.*

Each tree in the median forest is inconsistent [see Problem 4.3 in 17], because each leaf contains less than two data points, a number which obviously does not grow to infinity as $n \to \infty$. Thus, Theorem 4.1 shows that median forest combines inconsistent trees to form a consistent estimate. Put differently, Theorem 4.1 proves that there is no caveat to build fully grown median trees (i.e. to construct trees whose terminal nodes contain a small number of observations), provided a subsampling step. Indeed, the subsampling procedure is crucial to prove the consistency, and the condition $a_n/n \to 0$ forces the subsample size to be small compared to the original size of the training set.

Theorem 4.1 deals with median forests where subsampling is performed and trees are fully grown (k_n is maximal). Furthermore, general partitioning theorem states that, as for centred forests, median forests are consistent if $a_n = n$ and $n/2^{k_n} \to \infty$ (the tree construction is stopped at level k_n). A natural question is to know for which values of a_n and k_n the consistency of median forests holds and what the rate of consistency is. This will find an answer in Theorem 4.2.

Theorem 4.2 ([14]). *Assume that $Y = m(\mathbf{X}) + \varepsilon$, where ε is a centred noise such that $\mathbb{V}[\varepsilon|\mathbf{X} = \mathbf{x}] \leq \sigma^2 < \infty$, $\mathbf{X}$ is uniformly distributed over $[0,1]^d$ and m is L-Lipschitz continuous. Then, for all n, for all $\mathbf{x} \in [0,1]^d$,*

$$\mathbb{E}\big[m_{\infty,n}(\mathbf{x}) - m(\mathbf{x})\big]^2 \leq 2\sigma^2 \frac{2^k}{n} + dL^2 C_1 \left(1 - \frac{3}{4d}\right)^k. \tag{8}$$

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

$$k_n = \frac{1}{\ln 2 - \ln \beta}\Big[\ln(n) + C_3\Big], \tag{9}$$

under the condition that $a_n \geq C_4 n^{\frac{\ln 2}{\ln 2 - \ln \beta}}$. For these choices of k_n and a_n, we have

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

Equation (8) stems from the estimation/approximation error decomposition of median forests. The first term in equation (8) corresponds to the estimation error of the forest as in [3] or [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 since letting $n/2^k \to \infty$ and $k \to \infty$ allows us to control respectively the estimation and approximation error. According to Theorem 4.2, 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 $2^k/a_n$. 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 random forests variance reduction is a well-known property, already proved by [15] for a purely random forest, and by [24] for median forests. This highlights a first benefit of median forests over singular trees.

Theorem 4.2 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 1 deals with partially grown median forests, also called small-tree median forests.

Corollary 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 (9). Then, with the assumptions of Theorem 4.2, for all n, for all $\mathbf{x} \in [0,1]^d$,*

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

Trees of the median forest studied in Corollary 1 are not fully developed. This context is similar to centred forest where the construction of each tree is stopped at some level k_n depending on the size of the data set only. The consistency of the forest relies on the consistency of each individual tree which compose it. On the other hand, Corollary 2 deals with median forests where trees are fully grown, therefore establishing the rate of consistency of a particular forest which aggregates inconsistent trees.

Corollary 2 (Fully grown median forest). *Let $\beta = 1 - 3/4d$ and assume that assumptions of Theorem 4.2 hold. Consider a fully grown median forest whose parameters k_n and a_n satisfy $k_n = \log_2(a_n) - 2$. The optimal choice for a_n that minimizes the mean squared error in (8) is then given by (9), that is*

$$a_n = C_4 n^{\frac{\ln 2}{\ln 2 - \ln \beta}}.$$

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

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

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 2. Besides, Corollary 2 provides us with the optimal subsampling size for fully developed median forests.

A closer look at Theorem 4.2 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 (9), 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 4.2. In this way, Corollary 1 and 2 are simply two particular examples of such forests.

The main message behind Corollary 1 and Corollary 2 is that it is equivalent in terms of performance to tune k_n or a_n. Indeed, provided a proper parameter tuning, partially grown median forests without subsampling and fully grown median forests (with subsampling) have similar performance. This means that there is no need for tuning both parameters k_n and a_n simultaneously.

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, 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 2, are to be preferred since their computational time is lower than small-tree forests (described in Corollary 1). However, if one wants to build more accurate predictions, small-tree random forests should probably 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.

4.3. Subsampling and tree depth in Breiman's forests

4.3.1. *Theoretical results*

Since our main goal is the study of Breiman's forests, we would like to extend previous results on median forests to Breiman's forests. Unfortunately, the proof techniques turn out to be much more complex since the tree construction in the original algorithm depends on both the positions X_i and the label Y_i of the data set. Due to this additional difficulty, we are still not able to derive rate of consistency for Breiman's forest as for median forests in Theorem 4.2. To move forward towards the understanding of Breiman's forest, we will need the following assumption on the regression model.

(H1) *The response Y follows*

$$Y = \sum_{j=1}^{d} m_j(\mathbf{X}^{(j)}) + \varepsilon,$$

where $\mathbf{X} = (\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(d)})$ is uniformly distributed over $[0,1]^d$, ε is an independent centred Gaussian noise with finite variance $\sigma^2 > 0$, and each component m_j is continuous.

The main assumption in **(H1)** is the additivity nature of the regression model [for details on additive models, see e.g. 18, 27]. Indeed, under **(H1)**, all covariates $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(d)}$ are independent and since Y is the sum of univariate terms, there are no interaction effects among covariates. These assumptions simplify the theoretical analysis of random forests, which still remain a difficult task since the subsampling step and the CART-split criterion used at each step need to be taken into account. But this context is far from reality since there is no actual (big) data set where all variables are independent and where all interaction effects can be safely omitted.

Theorem 4.3 and Theorem 4.4 assess the consistency of Breiman's forest for two different parameters values. Theorem 4.3 focuses on small-trees forests, i.e. such that the number of terminal nodes `maxnodes` $= t_n$ is small compared to the number of observations a_n used in each tree.

Theorem 4.3 ([25])**.** *Assume that* **(H1)** *is satisfied. Then, provided $a_n, t_n \to \infty$ and $t_n(\log a_n)^9/a_n \to 0$, random forests are consistent, i.e.,*

$$\lim_{n\to\infty} \mathbb{E}\left[m_n(\mathbf{X}) - m(\mathbf{X})\right]^2 = 0.$$

The condition under which Breiman's forests are consistent are very similar to those for which centred forests are consistent. Indeed, recall that centred forests are consistent if $k_n \to \infty$ and $n/2^{k_n} \to \infty$, where k_n is the tree level. Since centred trees are complete binary trees, the number of leaves t_n satisfies $t_n = 2^{k_n}$. And thus the consistency assumptions reduced to $t_n \to \infty$ and $t_n/n \to 0$. Up to the $(\log a_n)^9$ term, Breiman's forest is consistent under the same assumptions. Noteworthy, the $(\log a_n)^9$ comes from the Gaussian noise for one part and from the partition complexity for the other part.

Theorem 4.3 holds for any value of a_n provided $a_n \to \infty$ and consequently the forest is consistent for $a_n = n$ (no subsampling step). The consistency result does not rely on the subsampling step, and a close inspection of the proof of Theorem 4.3 shows that each tree in the forest is consistent. This is the first consistency result for CART.

Enouncing Theorem 4.4 requires an additional assumption **(H2)** which helps to control cuts in the last tree levels.

(H2) *Let $W_i = \mathbb{1}_{\mathbf{X} \overset{\Theta}{\leftrightarrow} \mathbf{X}_i}$ the indicator that $\mathbf{X}$ falls in the same cell as $\mathbf{X}_i$ in the random tree designed with $\mathcal{D}_n$ and the random parameter Θ. Similarly, we let $W'_j = \mathbb{1}_{\mathbf{X} \overset{\Theta'}{\leftrightarrow} \mathbf{X}_j}$, where Θ' is an independent copy of Θ. Accordingly, we define*

$$\psi_{i,j}(Y_i, Y_j) = \mathbb{E}\Big[W_i W'_j \big| \mathbf{X}, \Theta, \Theta', \mathbf{X}_1, \dots, \mathbf{X}_n, Y_i, Y_j\Big]$$
$$\text{and} \quad \psi_{i,j} = \mathbb{E}\Big[W_i W'_j \big| \mathbf{X}, \Theta, \Theta', \mathbf{X}_1, \dots, \mathbf{X}_n\Big].$$

Then, one of the following two conditions holds:

(H2.1) *One has*

$$\lim_{n\to\infty} (\log a_n)^{2d-2} (\log n)^2 \mathbb{E}\left[\max_{\substack{i,j \\ i\neq j}} |\psi_{i,j}(Y_i, Y_j) - \psi_{i,j}|\right]^2 = 0.$$

(H2.2) *Letting $W_{i,j} = (W_i, W'_j)$, there exist a constant $C > 0$ and a sequence $(\gamma_n)_n \to 0$ such that, almost surely,*

$$\max_{\ell_1,\ell_2=0,1} \frac{|Corr(Y_i - m(\mathbf{X}_i), \mathbb{1}_{W_{i,j}=(\ell_1,\ell_2)} | \mathbf{X}_i, \mathbf{X}_j, Y_j)|}{\mathbb{P}^{1/2}\big[W_{i,j} = (\ell_1, \ell_2) | \mathbf{X}_i, \mathbf{X}_j, Y_j\big]} \leq \gamma_n,$$

and

$$\max_{\ell_1=0,1} \frac{|Corr\big((Y_i - m(\mathbf{X}_i))^2, \mathbb{1}_{W_i=\ell_1} | \mathbf{X}_i\big)|}{\mathbb{P}^{1/2}\big[W_i = \ell_1 | \mathbf{X}_i\big]} \leq C.$$

This condition probably shares some similarities with that of [22] in Theorem 3.4 although the connection is not limpid. Under **(H2)**, Theorem 4.4 establishes the consistency of Breiman's forests when trees are fully grown, i.e. $t_n = a_n$.

Theorem 4.4 ([25])**.** *Assume that* **(H1)** *and* **(H2)** *are satisfied and let $t_n = a_n$. Then, provided $a_n \to \infty$ and $a_n \log n/n \to 0$, random forests are consistent, i.e.,*

$$\lim_{n\to\infty} \mathbb{E}\left[m_n(\mathbf{X}) - m(\mathbf{X})\right]^2 = 0.$$

Theorem 4.3 and Theorem 4.4 are the first consistency results for Breiman's (2001) forests. As for median forests, we see that the consistency holds if the tree construction is stopped long before each cell contains one observation (Theorem 4.3), regardless of the subsample size a_n. But the subsampling step allows us to consider

deeper trees (where each terminal node contains one observation) and to prove that the resulting forest is consistent if $a_n \log n/n \to 0$ (Theorem 4.4).

Note that Theorem 4.3 and Theorem 4.4 are not valid in the case of bootstrap. Indeed, all proofs assume that subsampling is performed without replacement, which prevents the resulting data sets from containing the same observations more than once. These data sets are then distributed as the original training set which greatly simplifies the analysis. However, when performing bootstrap, generated data sets can contain several replicates of the same observations.

4.3.2. *Experiments*

We now carry out some experiments to investigate the influence of subsampling size and tree depth on Breiman's procedure. To do so, we start by defining various regression models on which several experiments are based. Throughout this section, we assess the forest performance via the empirical mean squared error. Additional simulations can be found in [14].

Model 1: $n = 800, d = 50, Y = X_1^2 + \exp(-X_2^2)$
Model 2: $n = 600, d = 100, Y = X_1X_2 + X_3^2 - X_4X_7 + X_8X_{10} - X_6^2 + \mathcal{N}(0, 0.5)$
Model 3: $n = 600, d = 100, Y = -\sin(2X_1) + X_2^2 + X_3 - \exp(-X_4) + \mathcal{N}(0, 0.5)$
Model 4: $n = 600, d = 100, Y = X_1 + (2X_2 - 1)^2 + \sin(2\pi X_3)/(2 - \sin(2\pi X_3)) + \sin(2\pi X_4) + 2\cos(2\pi X_4) + 3\sin^2(2\pi X_4) + 4\cos^2(2\pi X_4) + \mathcal{N}(0, 0.5)$
Model 5: $n = 700, d = 20, Y = \mathbb{1}_{X_1>0} + X_2^3 + \mathbb{1}_{X_4+X_6-X_8-X_9>1+X_{10}} + \exp(-X_2^2) + \mathcal{N}(0, 0.5)$
Model 6: $n = 500, d = 30, Y = \sum_{k=1}^{10} \mathbb{1}_{X_k^3<0} - \mathbb{1}_{\mathcal{N}(0,1)>1.25}$

For all regression frameworks, we consider covariates $\mathbf{X} = (X_1, \ldots, X_d)$ that are uniformly distributed over $[-1, 1]^d$. These models comes from various sources [see, e.g., 20, 29]. 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 is evaluated on the test set.

Tree depth. We start by comparing 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 the parameters default values. Small-tree Breiman's forests are similar to Breiman's forests except that tree depth is controlled via the parameter `maxnodes` and that the whole sample $\mathcal{D}_n$ is used to build each tree (without any resampling step). To study only the influence of `maxnodes`, we fix `nodesize` $= 1$ for both Breiman's forests and small-tree Breiman's forests.

We present the mean squared errors of small-tree Breiman's forests for different number of terminal nodes (10%, 30%, 63%, 80% and 100% of the sample size) for **Models 1-6**. The results can be found in Figure 2 in the form of box-plots. We can notice that the forests such that `maxnodes`$= 0.3n$ give similar (**Model 5**) or best (**Model 6**) performances than standard Breiman's forest.

Subsampling. We now study the influence of subsampling on Breiman's forests by comparing the original Breiman's procedure with subsampled Breiman's forests. Subsampled Breiman's 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).

In Figure 3, the mean squared errors of subsampled Breiman's forests is plotted for different subsampling sizes ($0.4n$, $0.5n$, $0.63n$ and $0.9n$) for **Models 1-6**. For every model, we can notice that subsampled forests performance is comparable with that of standard Breiman's forest (Figure 3), as long as the subsampling parameter is well chosen. The large subsample sizes, around $0.9n$, lead to small risk: this may arise from the probably high signal/noise ratio. In each model, when the noise is multiplied by two, the results, exemplified in Figure 4, are less obvious.

According to the theoretical analysis of median forests, we know that there is no need to optimize both the subsample size and the tree depth: optimizing only one of these two parameters leads to the same performance as optimizing both of them. Regarding Breiman's forests, theoretical results are much more difficult to obtain and so far, there is no equivalent upper bound on the risk, which would allow us to determine the joint influence of parameters on forest performance. However, according to empirical results, there is no justification for default values in random forests for subsampling or tree depth, since optimizing either leads to better performance.

5. Conclusion and perspective

Random forests depend on four parameters: the number of trees, the subsample size common to each tree, the tree depth and the number of preselected features for splitting in each cell.

5.1. Number of eligible features for splitting

There is no theory to guide the choice of the number `mtry` of preselected features in each cell. If `mtry` is small, the computational cost of the procedure is small compared to original trees (CART, `mtry` $= d$) and the selection of the splitting direction is almost done uniformly at random over all directions. If `mtry` is too large, the splitting direction is close to the best splitting direction and the calculation time is comparable to original trees (CART, `mtry` $= d$). But the good tradeoff between these two extreme cases is unclear. Empirically, it seems that the default value is too small [see, 13, 16]

5.2. Number of trees

The impact of the number of trees on forest performance is well understood since it corresponds to the number of replicates in a Monte Carlo simulation. To obtain the best performance, we need to add a large number of trees in the forest, knowing that it increases linearly the computational time of the procedure. To do so, Theorem 3.1 provides a lower bound on the number of trees (depending on the regression model) to reach a given accuracy. Empirically, the optimal number of trees is obtained when the forest error reaches its limit as the number of trees grows to infinity. Once the other parameters of the forest have been properly tuned, one can just plot the forest error as a function of M and choose M such that the error roughly reaches its limit.

5.3. Subsample size and tree depth

Tree depth can be controlled via different parameters: `nodesize` which controls the maximal number of observations in each cell, `maxnodes` which limits the number of terminal nodes and the tree level k_n which ensures that each cell has been split at most k_n times.

The subsample size and the tree depth play similar roles in random forest performance. As shown by simulations and the theoretical analysis of median forest, it is sufficient to tune either the subsample size (and consider fully grown trees) or the tree depth (and use the full data set for each tree) to obtain similar or better performance compared to Breiman's forests. Besides, there is no empirical justification for bootstrap compared to a proper tuning of the subsample size.

5.4. Perspective

One promising line of research is to investigate the role of the parameter `mtry` on random forests performance. Indeed, existing results do not take advantage of the extra randomization induced by preselecting `mtry` variables at each cell. For instance, Theorems 4.3 and 4.4 stating the consistency of random forests are valid for any value of `mtry`: in theory, all values of `mtry` are equivalent whereas in practice `mtry` needs to be properly tuned.

Another natural question regarding the results presented in this paper would be whether they can be extended to classification framework. In the context of binary classification, where $Y \in \{0, 1\}$, the extension is straightforward for most of these results since binary classification is related to the problem of estimating the

probability $\mathbb{P}[Y = 1|\mathbf{X} = \mathbf{x}]$, which is a regression task. However, for problems involving more than two classes, the analysis is less obvious and probably require a non-negligible amount of work. For instance, the CART-split criterion used for regression and the one used for two class classification leads to the same splits, and thus theory on CART regression trees shares some similarities with that on CART classification trees (when Y is binary). However these splitting criteria differ when considering multiclass classification problems, which broaden the gap between the theoretical analysis of CART regression trees and CART multiclass classification trees.

References

[1] S. Arlot and R. Genuer. Analysis of purely random forests bias. arXiv:1407.3939, 2014.

[2] S. Bernard, L. Heutte, and S. Adam. Forest-RK: A new random forest induction method. In D.-S. Huang, D.C. Wunsch II, D.S. Levine, and K.-H. Jo, editors, *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, pages 430–437, Berlin, 2008. Springer.

[3] G. Biau. Analysis of a random forests model. *Journal of Machine Learning Research*, 13:1063–1095, 2012.

[4] G. Biau and L. Devroye. Cellular tree classifiers. *Electronic Journal of Statistics*, 7:1875–1912, 2013.

[5] G. Biau, L. Devroye, and G. Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9:2015–2033, 2008.

[6] G. Biau and E. Scornet. A random forest guided tour. *Test*, 25:197–227, 2016.

[7] A.-L. Boulesteix, S. Janitza, J. Kruppa, and I. R. König. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2:493–507, 2012.

[8] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[9] L. Breiman. *Consistency for a simple model of random forests.* Technical Report 670, University of California, Berkeley, 2004.

[10] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees.* Chapman & Hall/CRC, Boca Raton, 1984.

[11] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7:81–227, 2011.

[12] D.R. Cutler, T.C. Edwards Jr, K.H. Beard, A. Cutler, K.T. Hess, J. Gibson, and J.J. Lawler. Random forests for classification in ecology. *Ecology*, 88:2783–2792, 2007.

[13] R. Díaz-Uriarte and S. Alvarez de Andrés. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7:1–13, 2006.

[14] R. Duroux and E. Scornet. Impact of subsampling and pruning on random forests. arXiv:1603.04261, 2016.

[15] R. Genuer. Variance reduction in purely random forests. *Journal of Nonparametric Statistics*, 24:543–562, 2012.

[16] R. Genuer, J. Poggi, and C. Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31:2225-2236, 2010.

[17] L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression.* Springer, New York, 2002.

[18] T. Hastie and R. Tibshirani. Generalized additive models. *Statistical Science*, 1:297–310, 1986.

[19] H. Ishwaran. The effect of splitting on random forests. *Machine Learning*, pages 1–44, 2013.

[20] L. Meier, S. Van de Geer, and P. Bühlmann. High-dimensional additive modeling. *The Annals of Statistics*, 37:3779–3821, 2009.

[21] N. Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7:983–999, 2006.

[22] L. Mentch and G. Hooker. Ensemble trees and clts: Statistical inference for supervised learning. arXiv:1404.6473, 2014.

[23] A.M. Prasad, L.R. Iverson, and A. Liaw. Newer classification and regression tree techniques: Bagging and random forests for ecological prediction. *Ecosystems*, 9:181–199, 2006.

[24] E. Scornet. On the asymptotics of random forests. *Journal of Multivariate Analysis*, 146:72–83, 2016.

[25] E. Scornet, G. Biau, and J.-P. Vert. Consistency of random forests. *The Annals of Statistics*, 43:1716–1741, 2015.

[26] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1304, 2011.

[27] C.J. Stone. Additive regression and other nonparametric models. *The Annals of Statistics*, pages 689–705, 1985.

[28] V. Svetnik, A. Liaw, C. Tong, J.C. Culberson, R.P. Sheridan, and B.P. Feuston. Random forest: A classification and regression tool for compound classification and QSAR modeling. *Journal of Chemical Information and Computer Sciences*, 43:1947–1958, 2003.

[29] M. van der Laan, E.C. Polley, and A.E. Hubbard. Super learner. *Statistical Applications in Genetics and Molecular Biology*, 6, 2007.

[30] S. Wager. Asymptotic theory for random forests. arXiv:1405.0352, 2014.

[31] S. Wager, T. Hastie, and B. Efron. Standard errors for bagged predictors and random forests. arXiv:1311.4555, 2013.

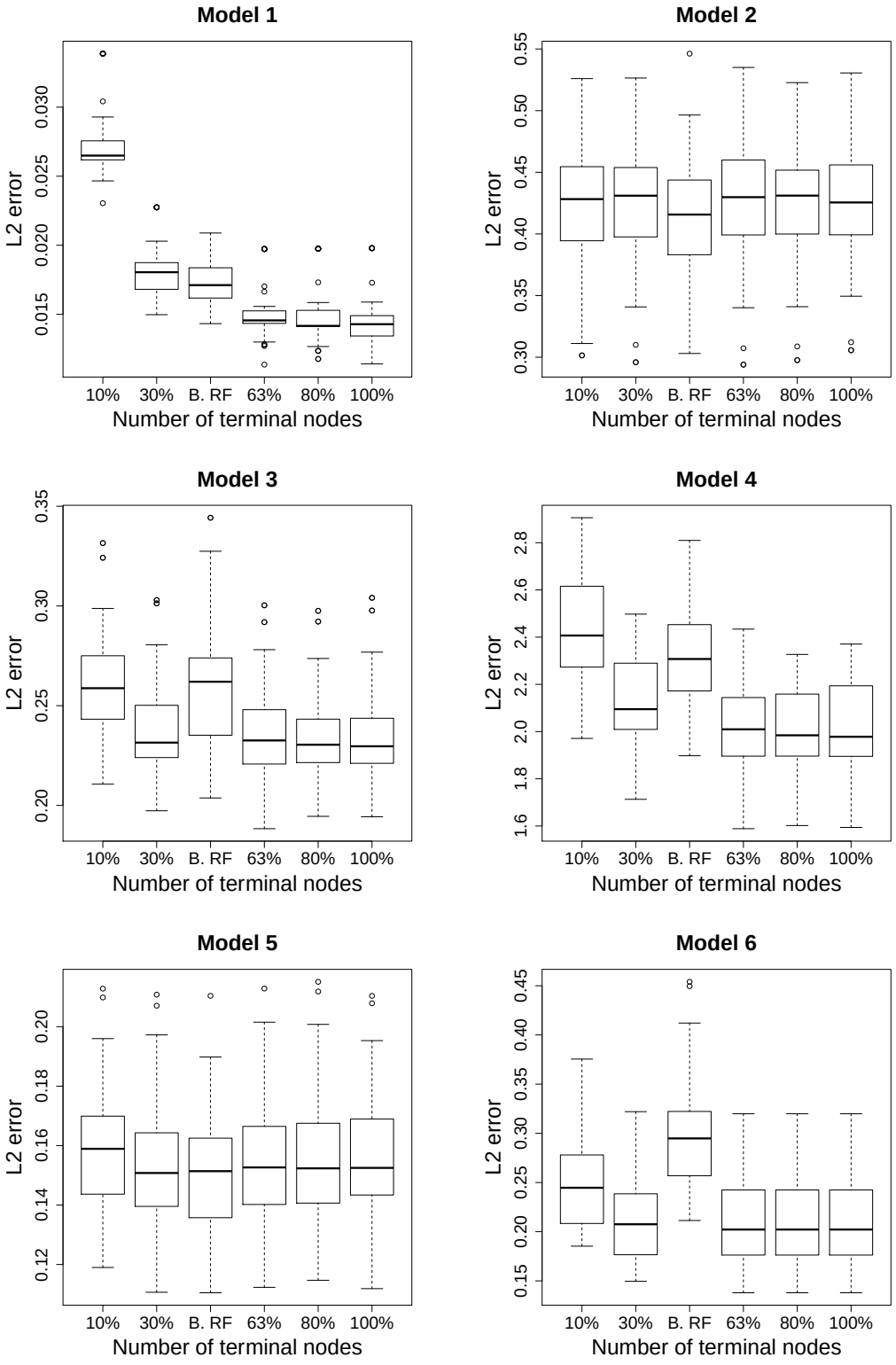

FIGURE 2. Comparison of standard Breiman's forests against several small-tree Breiman's forests in terms of $\mathbb{L}^2$ error.

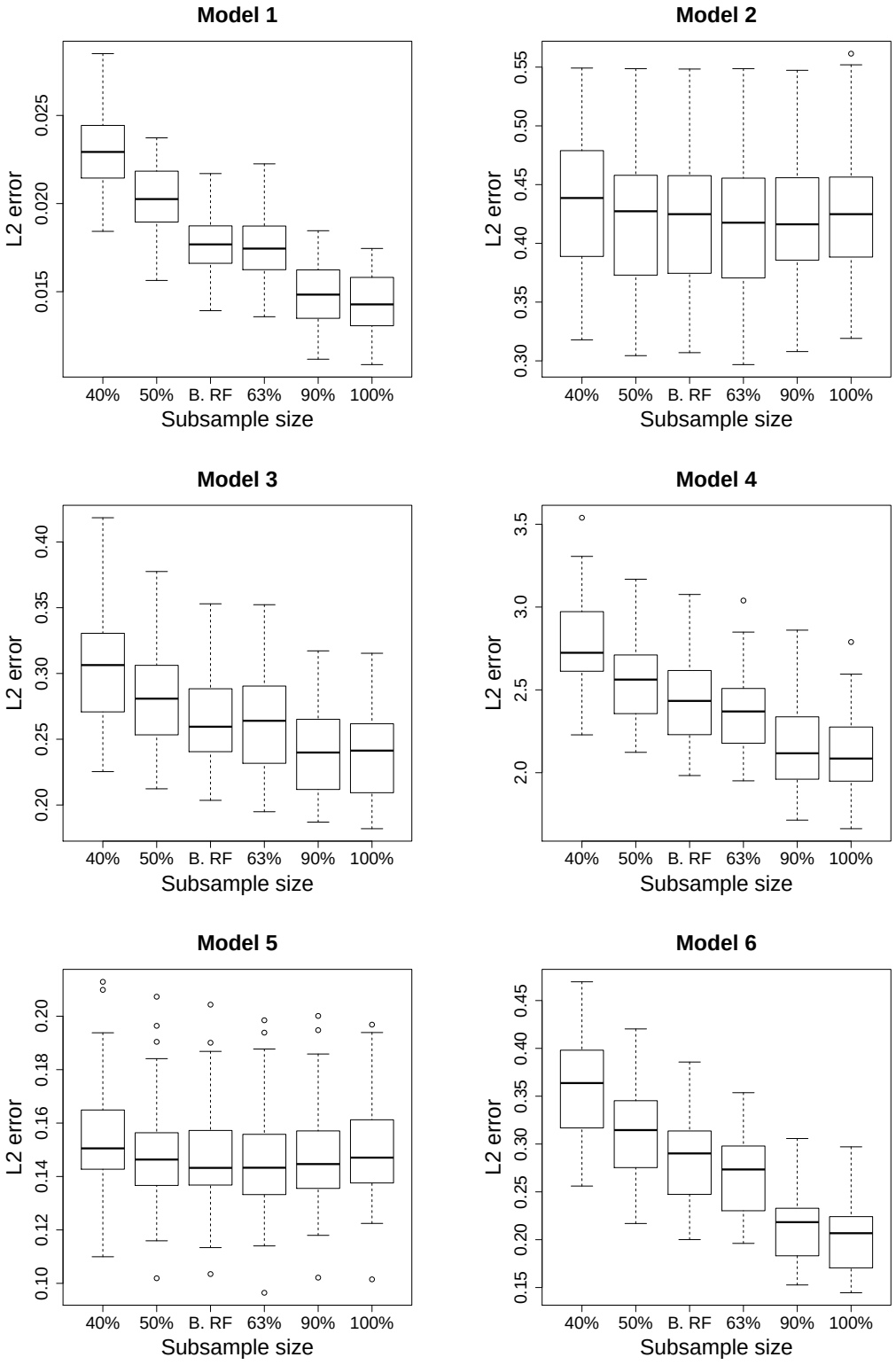

FIGURE 3. Standard Breiman forests versus several subsampled Breiman forests.

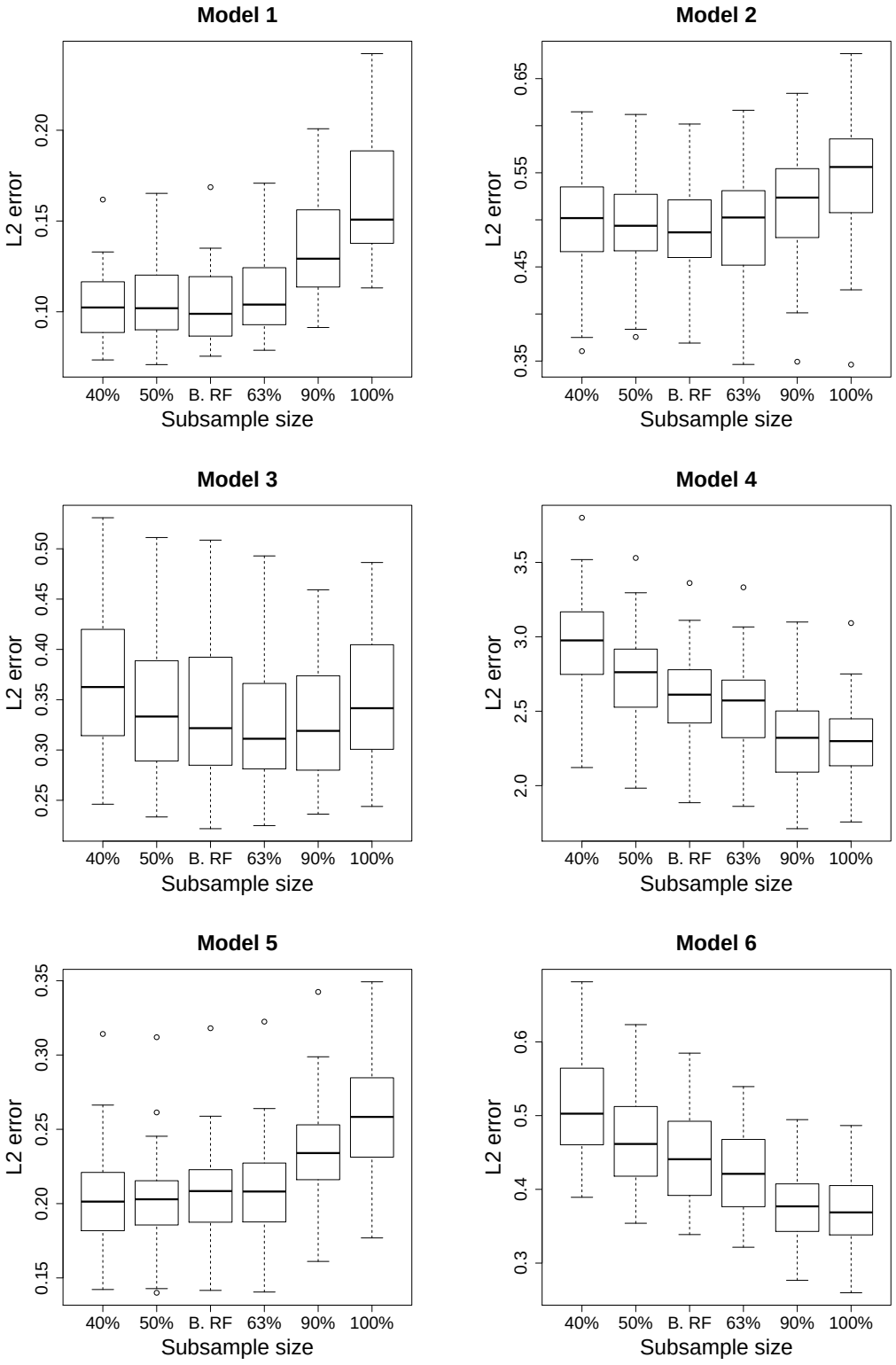

FIGURE 4. Standard Breiman forests versus several subsampled Breiman forests (noisy models).