



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. 2016.
hal-01287521v3

HAL Id: hal-01287521

<https://hal.archives-ouvertes.fr/hal-01287521v3>

Submitted on 7 May 2019

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.

Impact of subsampling and tree depth on random forests.

Roxane Duroux

Sorbonne Universités, UPMC Univ Paris 06, F-75005, Paris, France
roxane.duroux@upmc.fr

Erwan Scornet

Centre de Mathématiques Appliquées
École polytechnique
UMR 7641, 91128 Palaiseau, France
erwan.scornet@polytechnique.edu

Abstract

Random forests are ensemble learning methods introduced by ? 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 show that forests based on subsampling and 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.

Index Terms — Random forests, randomization, parameter tuning, subsampling, tree depth.

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.

One of the most popular random forests is that of ? which grows trees based on CART procedure (Classification And Regression Trees, ?) and randomizes both the training set and the splitting variables. Breiman’s (?) 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 (?) and pattern recognition (?), 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.

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 ? and ?. 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). ? 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 . ? gives an explicit upper bound on forest accuracy which depends on the number of trees. ? and ? 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 make each tree estimate. It turns out that the subsampling step is a key element to design consistent forests based on inconsistent trees (?). The parameter m_{try} regulates how much randomization is injected into the splitting procedure. A critical case 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 ? and ? 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` (?), 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 median forest, 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 median forests. We establish an upper bound for the risk of median forests and by doing so, we highlight the fact that subsampling and tree depth have similar influence on median forest predictions. To confirm that these results still hold for Breiman’s forests, we implement numerous experiments in Section 4. 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 $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)\}$ 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 $\mathbf{X} = (X^{(1)}, \dots, X^{(d)})$ 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 $m_n : [0, 1]^d \rightarrow \mathbb{R}$ of m , based on the data set \mathcal{D}_n .

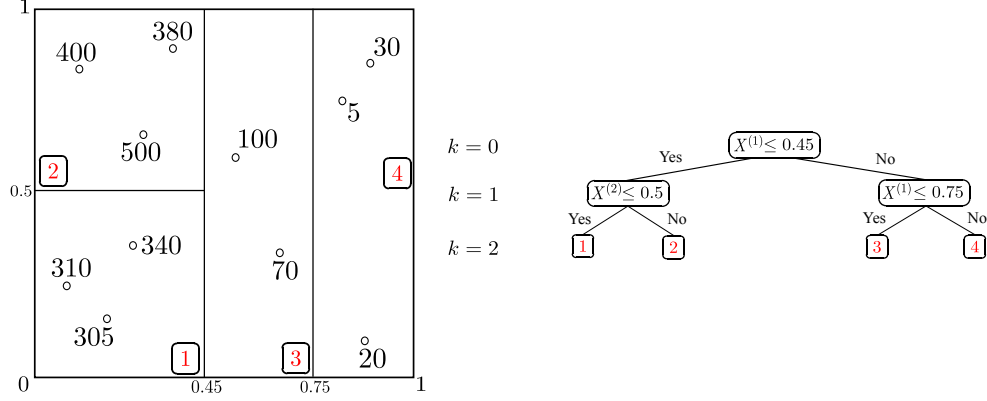


Figure 1: A decision tree of depth $k = 2$ in dimension $d = 2$.

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 Figure 1). Instances of decision trees such as Breiman's trees and median trees are described below. Mathematically speaking, a tree estimate 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, where some randomness 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 $\Theta_1, \dots, \Theta_M$ 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

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

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

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

where \mathbb{E}_Θ 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 (?) 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 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 m_{try} coordinates among $\{1, \dots, d\}$.
 - (d) Select the split minimizing the CART-split criterion (see ??, 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 j -th 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 $m_{M,n}(\mathbf{x}, \Theta_1, \dots, \Theta_M)$ 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 j -th tree, the variable Θ_j is nothing but a large vector containing first the indices of observations selected to build the j -th 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., ???) 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 (??) or its consistency (?). When studying the original algorithm, one faces the complexity of the algorithm 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 median forest (see, for example, ?, for details on median tree). Median forests construction depends only on the \mathbf{X}_i ’s making them a good tradeoff between the complexity of Breiman’s (?) forests and the simplicity of totally non adaptive forests, whose construction is independent of the data set.

The interest of studying median 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, ? proves that subsampled median forests are consistent even if each median tree in the forest is not, therefore highlighting the nice effect of subsampling on median forests. Besides, median forests can be tuned such that each leaf of each tree contains exactly one observation, thus being close to Breiman’s forests.

3.1 Median Forests

We now describe the construction of median forest. In the spirit of Breiman’s (?) 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, \dots, 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` = $\lceil a_n 2^{-k_n} \rceil$. 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 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 j among $\{1, \dots, d\}$ 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) 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 $m(\mathbf{x}, \Theta_j)$ 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 $m_{M,n}(\mathbf{x}, \Theta_1, \dots, \Theta_M)$ of the predictions given by the M trees.

3.2 Main theorem

Instead of studying the risk of the finite median forest $m_{M,n}$, we focus on the risk of its infinite counterpart

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

obtained by setting $M \rightarrow \infty$. Theorem 3.1 presents an upper bound of the mean squared error of the infinite median random forests $m_{\infty,n}$.

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 $\mathbf{x} \in [0, 1]^d$,*

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

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

$$k_n = \frac{1}{\ln 2 - \ln \beta} \left[\ln(n) + C_2 \right], \quad (3)$$

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

$$\mathbb{E}[m_{\infty,n}(\mathbf{x}) - m(\mathbf{x})]^2 \leq C_4 n^{\frac{\ln \beta}{\ln 2 - \ln \beta}}. \quad (4)$$

In practice, we cannot compute infinite forests. Thus, we are naturally interested in the consistency of finite forests. Fortunately, inequality (4) in Theorem 3.1 can be extended to finite median forests by using Theorem 3.3 in ? if

$$M = O\left(n^{\frac{-\ln \beta}{\ln 2 - \ln \beta}}\right).$$

Thus, the rate of consistency and the following analysis are still valid for finite forests.

Equation (2) stems from the estimation/approximation error decomposition of median forests. The first term in equation (2) corresponds to the estimation error of the forest as in ? or ? 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 $n/2^k \rightarrow \infty$ and $k \rightarrow \infty$, 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 ? for a totally non adaptive forest, and by ? 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 (2). 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 ?. Surprisingly, the rate of consistency of median forests is faster than that of centred forest established in ?, which is equal to

$$\mathbb{E}[m_{\infty,n}^{cc}(\mathbf{X}) - m(\mathbf{X})]^2 \leq Cn^{-\frac{3}{4d \ln 2 + 3}}, \quad (5)$$

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

Noteworthy, the fact that the upper bound (4) is sharper than (5) appears to be important in the case where $d = 1$. In that case, according to Theorem 3.1, for all n , for all $\mathbf{x} \in [0, 1]^d$,

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

which is the minimax rate over the class of Lipschitz functions (see, e.g., ??). This was to be expected since, in dimension one, median random forests are simply a median tree which is known to reach minimax rate (?). Unfortunately, for $d = 1$, the centred forest bound (5) turns out to be suboptimal since it results in

$$\mathbb{E}[m_{\infty,n}^{cc}(\mathbf{X}) - m(\mathbf{X})]^2 \leq Cn^{-\frac{3}{4 \ln 2 + 3}}. \quad (6)$$

Finally, note that Theorem 3.1 provides the pointwise rate of consistency of $m_{\infty,n}$. Thus, median forests are pointwise consistent, contrary to Breiman's forests which are thought to be inconsistent for some particular query point \mathbf{x} (see, e.g., ?).

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 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 (3). Under the same assumptions as in Theorem 3.1, we have, for all n , for all $\mathbf{x} \in [0, 1]^d$,*

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

Up to an approximation, Corollary 1 is the counterpart of Theorem 2.2 in ? 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 1 for median forests are similar. Note that, for both forests, the optimal depth k_n of each tree is the same.

Corollary 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 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 (2) is then given by (3), that is*

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

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

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

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 2. Besides, Corollary 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), 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, Corollary 1 and 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 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 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.

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 $\mathbf{X} = (X_1, \dots, X_d)$ 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 (**Model 1, 5-8**). **Model 2** can be found in ? and **Models 3-4** are presented in ?. 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.

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 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` for both Breiman's forests and small-tree Breiman's forests.

In Figure 2, we present, for the **Models 1-8** 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. Every sub-figure of Figure 2 presents 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.

For every model, 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), we draw the same curves as in Figure 2, for different learning data set sizes ($n = 100, 200, 300$ and 400). For each size of the learning set, the optimal `maxnodes` values are plotted in the last graph, where the optimal `maxnodes` value m^* is defined as

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

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 the last sub-figure in Figure 3, the optimal `maxnodes` value seems to be proportional to the sample size. For **Model 1**, the optimal value m^* seems to verify $0.25n < m^* < 0.3n$. 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, for **Models 1-8**. The results can be found in Figure 5 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 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. 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), 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 1-8** introduced previously. 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. Every sub-figure of Figure 6 presents 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 ?).

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

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

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 the last sub-figure of Figure 7. For **Model 1**, the optimal value a_n^* 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 1-8**. We can notice that the forests with a subsampling size of $0.63n$ give similar performances than 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. In each model, when the noise is increasing, the results, exemplified in Figure 10, are less obvious. That is why we can lawfully use the subsampling size as an optimization parameter for the Breiman's forest performance.

5 Discussion

In this paper, we study the role of subsampling step and tree depth in both median and Breiman's forests procedure. By analyzing median forests, a simpler but close version of Breiman's forests, we show that the performance of subsampled median forests and that of small-tree median 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.

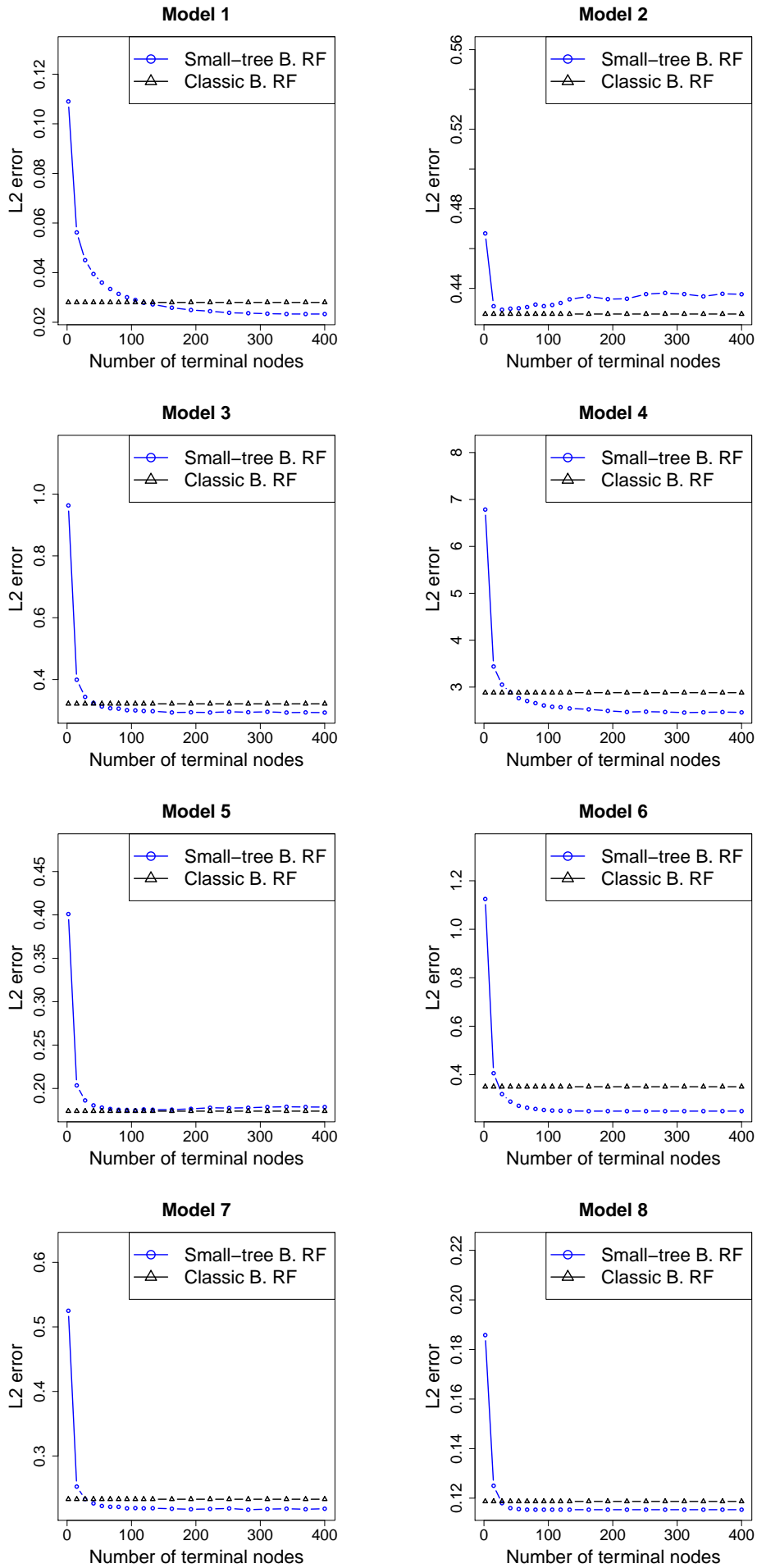


Figure 2: Comparison of standard Breiman's forests (B. RF) against small-tree Breiman's forests in terms of \mathbb{L}^2 error.

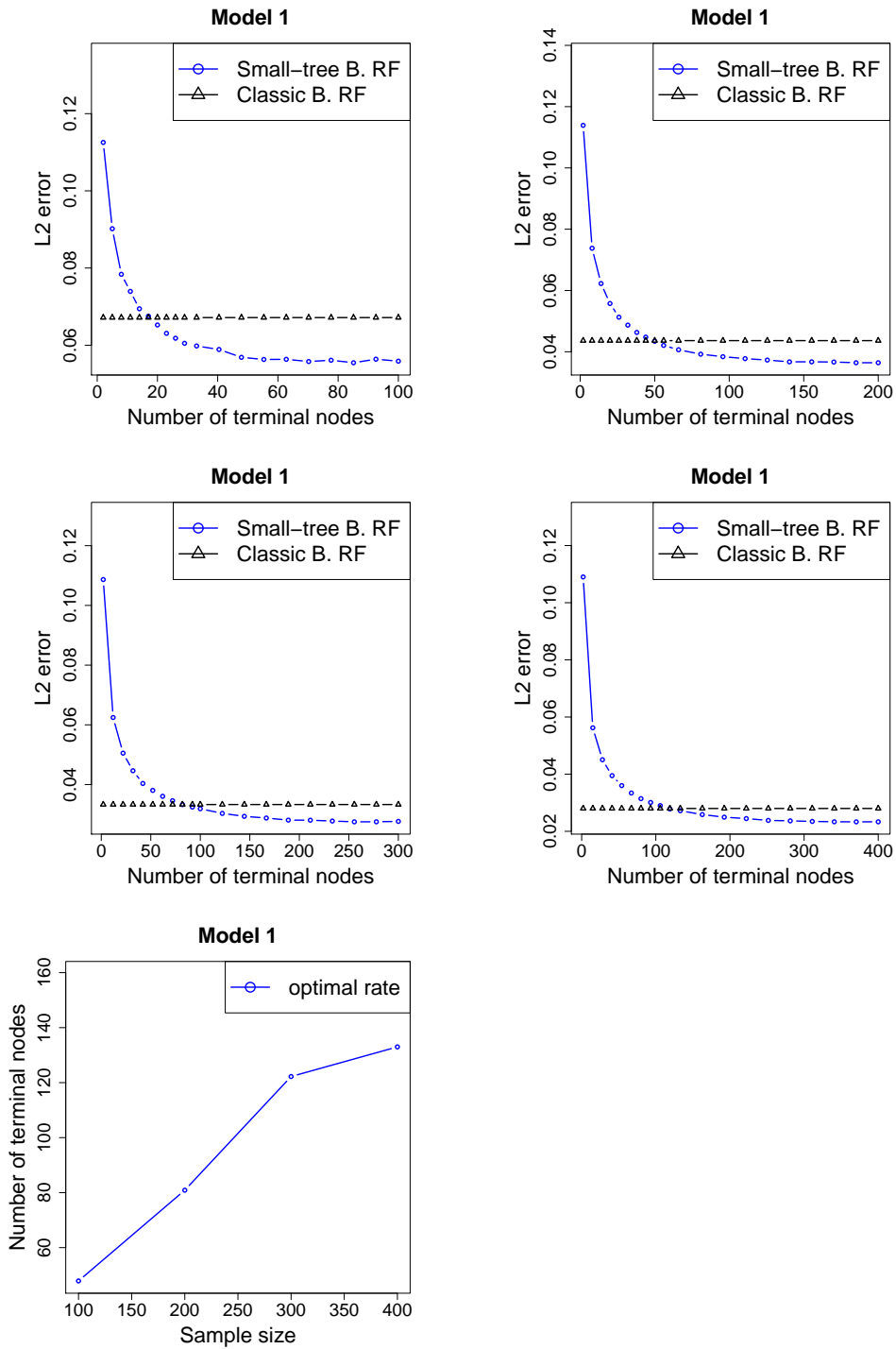


Figure 3: First four plots: \mathbb{L}^2 error of small-tree and standard Breiman's forests in **Model 1** for different sizes of the training set (ranging from 100 to 400); last plot: optimal values of the number of terminal nodes in **Model 1**.

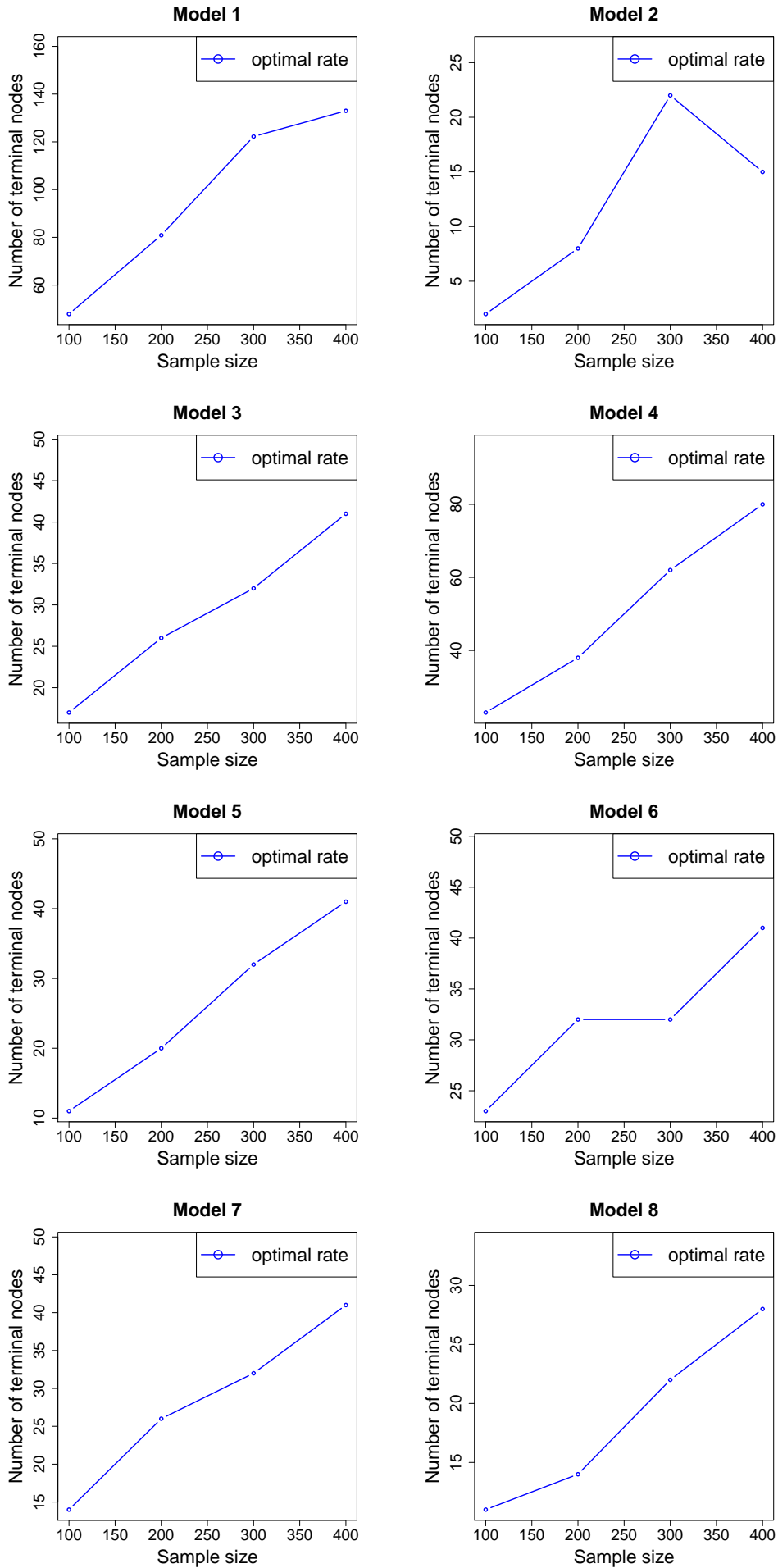


Figure 4: Optimal values of the number of terminal nodes.

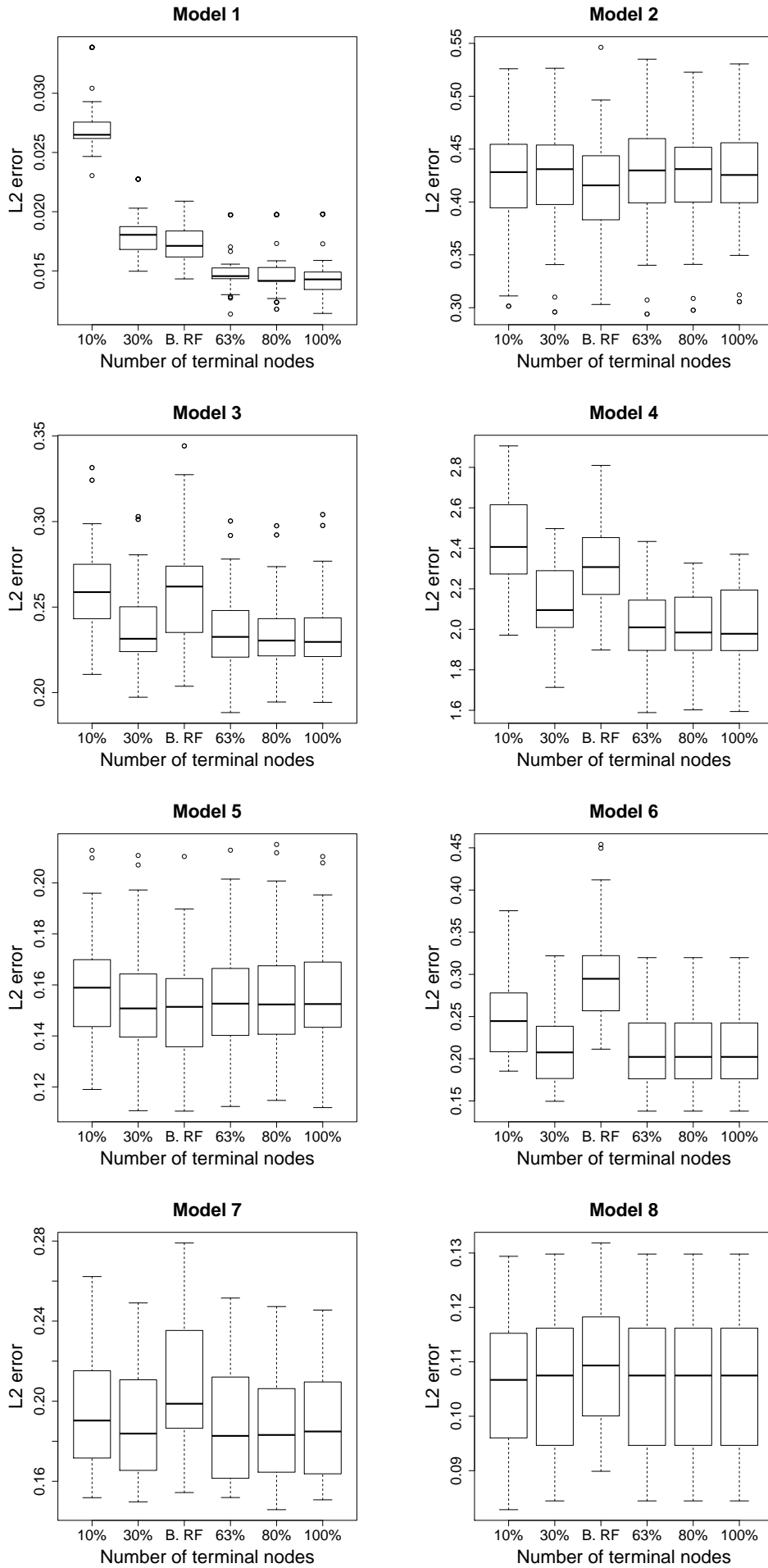


Figure 5: Comparison of standard Breiman's forests against several small-tree Breiman's forests in terms of \mathbb{L}^2 error.

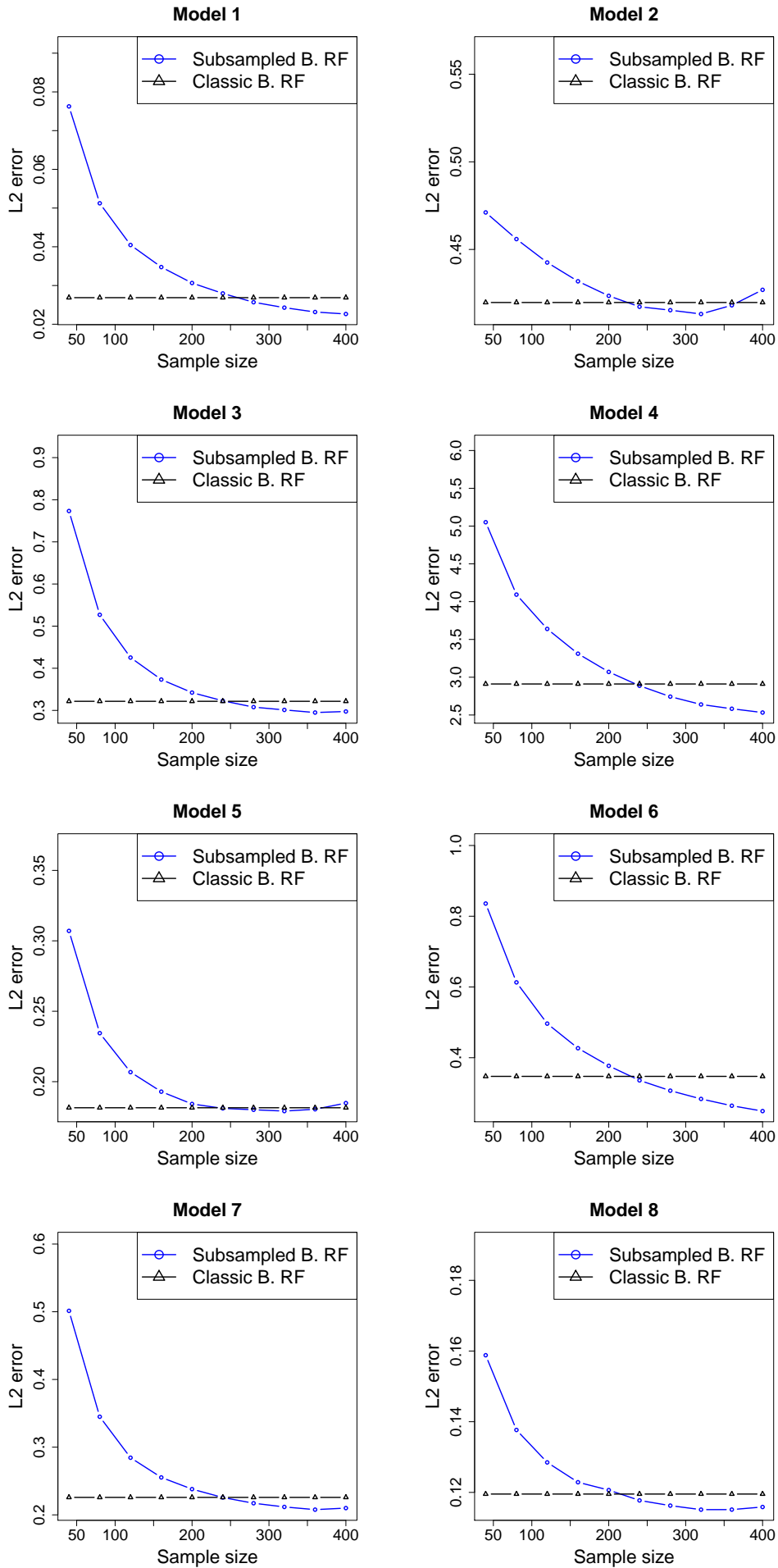


Figure 6: Standard Breiman forests versus subsampled Breiman forests.

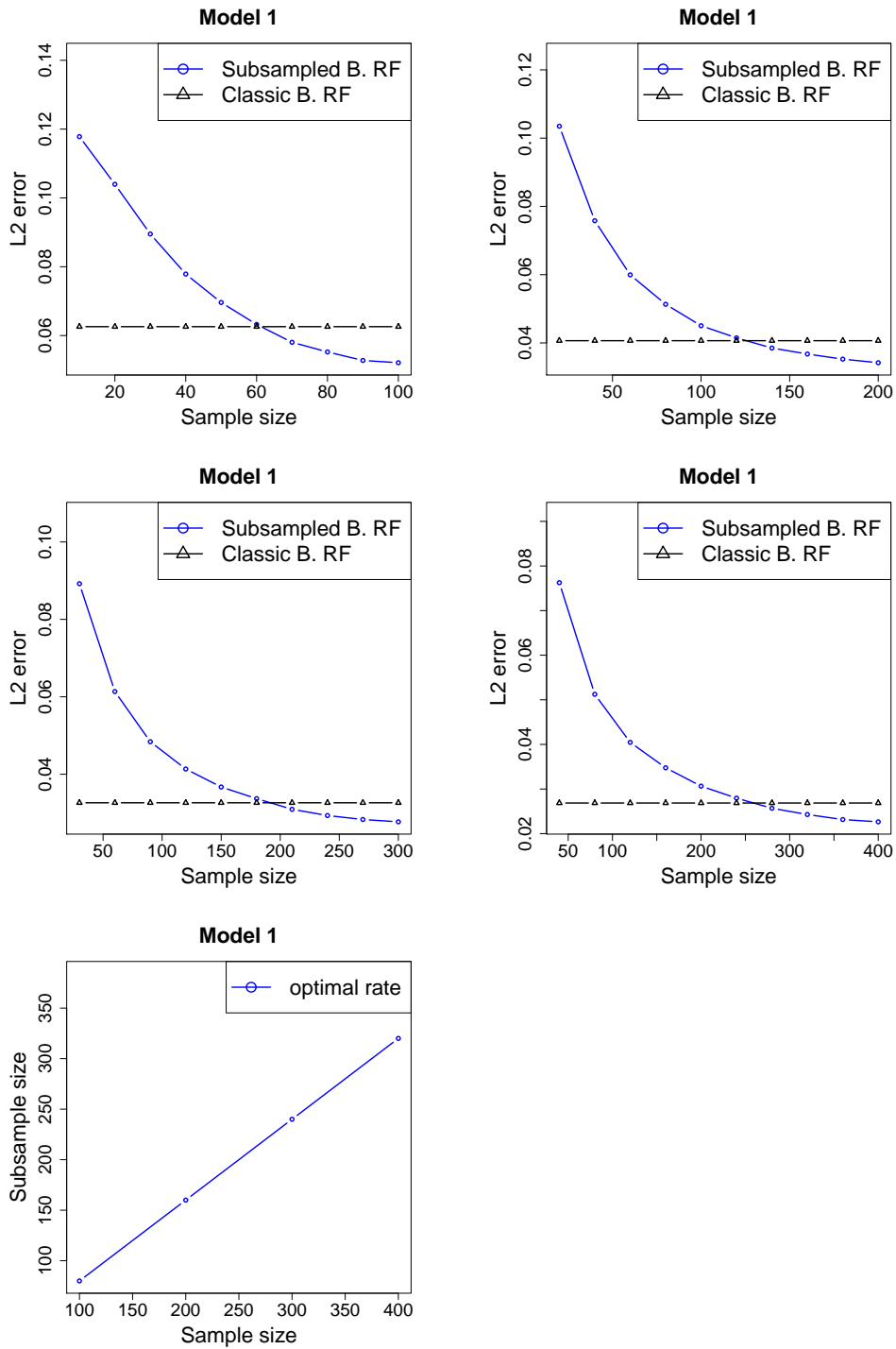


Figure 7: First four plots: \mathbb{L}^2 error of subsampled and Breiman's forests in **Model 1** for different sizes of the training set (ranging from 100 to 400); last plot: optimal values of the subsample size in **Model 1**.

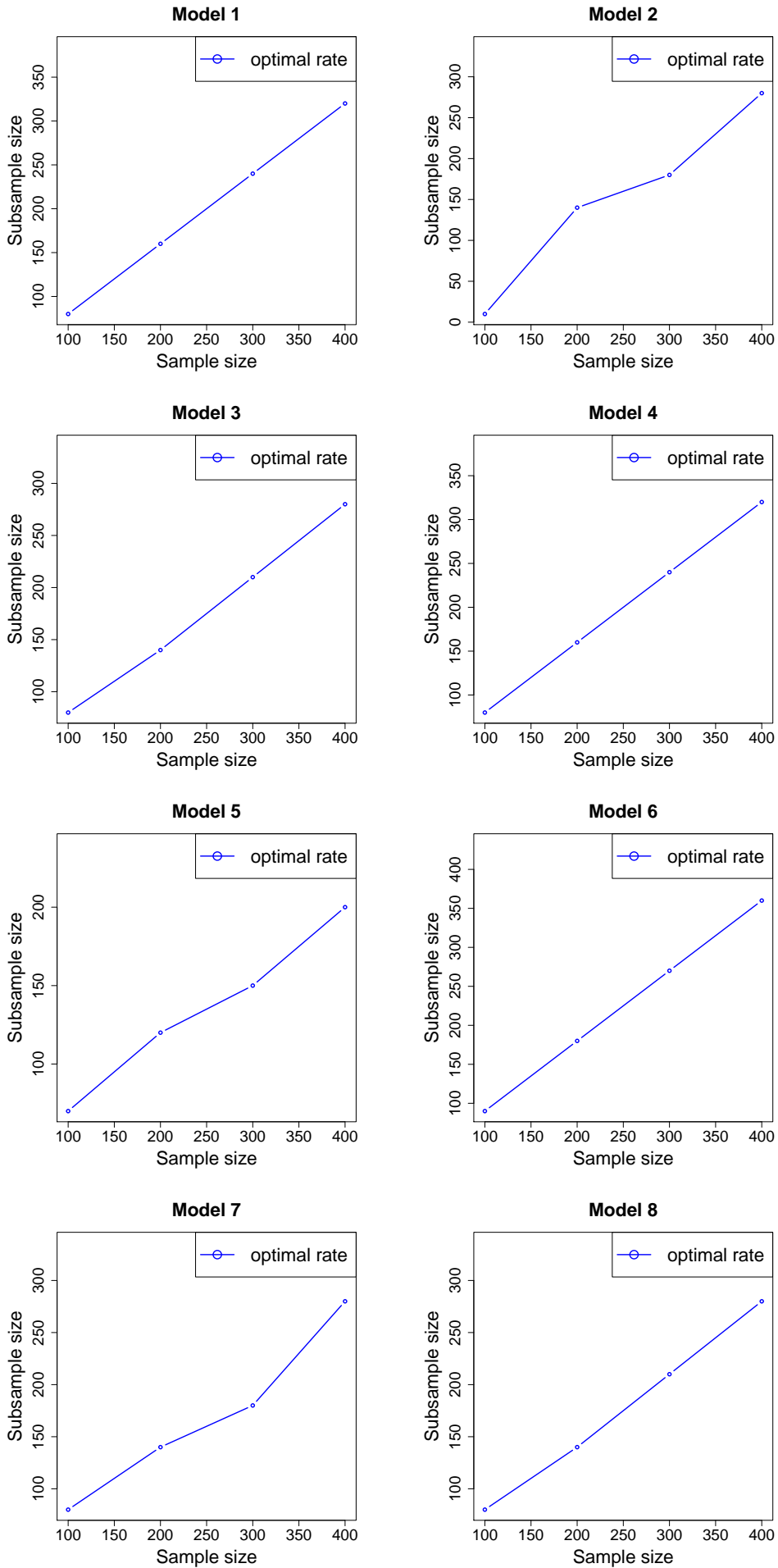


Figure 8: Optimal values of subsample size.

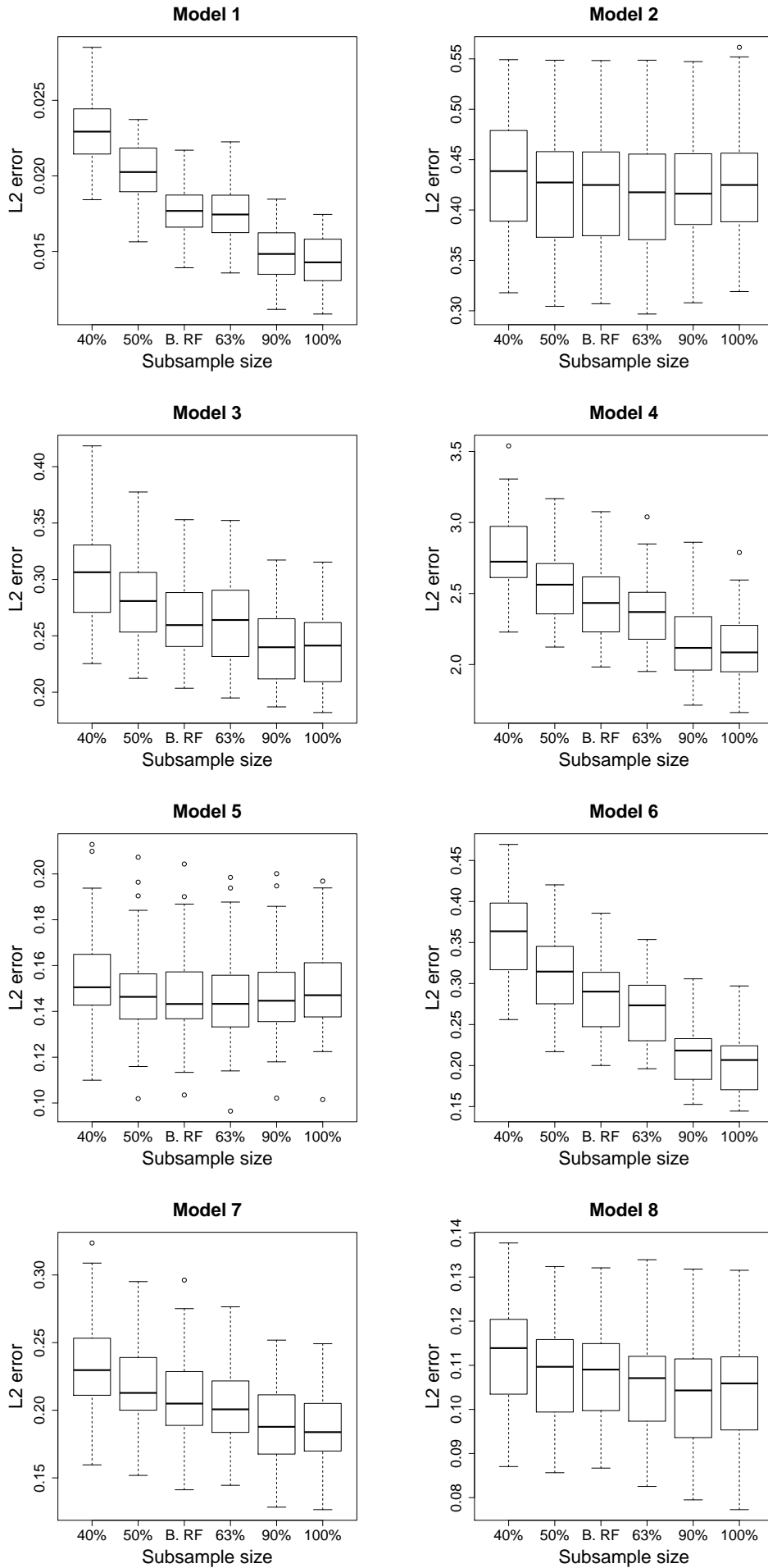


Figure 9: Standard Breiman forests versus several subsampled Breiman forests.

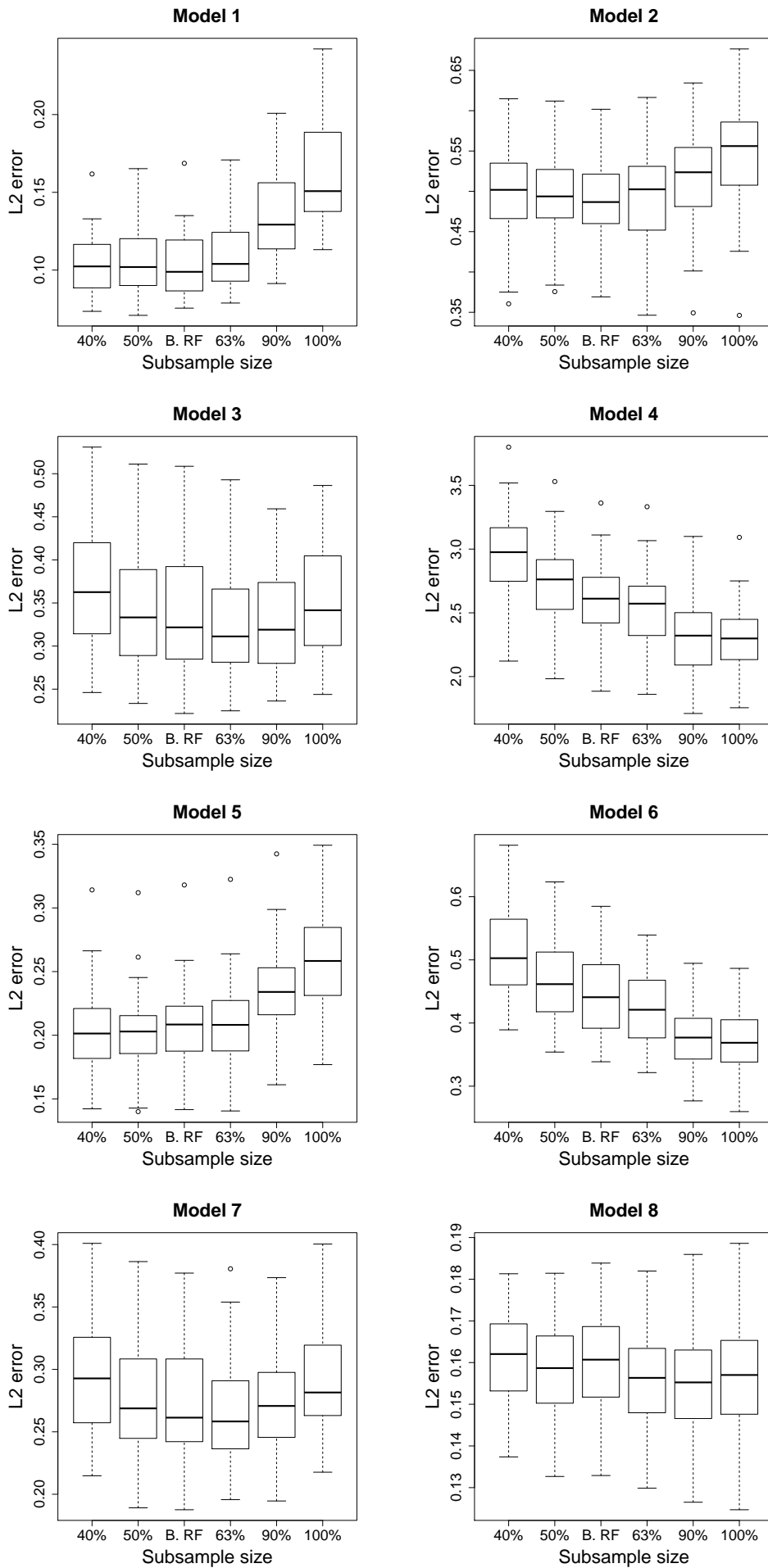


Figure 10: Standard Breiman forests versus several subsampled Breiman forests (noisy models).

6 Proofs

Proof of Theorem 3.1. Let us start by recalling that the infinite median random forest estimate $m_{\infty,n}$ can be written as a local averaging estimate

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

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

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

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.

Approximation error. Regarding J_n , by the Cauchy Schwartz inequality,

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

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 ℓ -th side. Then,

$$J_n \leq L^2 \sum_{\ell=1}^d \mathbb{E}\left[V_{\ell}(\mathbf{x}, \Theta)^2\right].$$

According to Lemma 1 specified further, we have

$$\mathbb{E}\left[V_i(\mathbf{x}, \Theta)^2\right] \leq C_1 \left(1 - \frac{3}{4d}\right)^k,$$

with $C_1 = \exp(12/(4d - 3))$. Thus, for all k , we have

$$J_n \leq dL^2 C_1 \left(1 - \frac{3}{4d}\right)^k.$$

Estimation error. Let us now focusing on the term I_n , we have

$$\begin{aligned} I_n &= \mathbb{E}\left[\sum_{i=1}^n W_{ni}^\infty(\mathbf{x})(Y_i - m(\mathbf{X}_i))\right]^2 \\ &= \sum_{i=1}^n \sum_{j=1}^n \mathbb{E}\left[W_{ni}^\infty(\mathbf{x})W_{nj}^\infty(\mathbf{x})(Y_i - m(\mathbf{X}_i))(Y_j - m(\mathbf{X}_j))\right] \\ &= \mathbb{E}\left[\sum_{i=1}^n (W_{ni}^\infty(\mathbf{x}))^2 (Y_i - m(\mathbf{X}_i))^2\right] \\ &\leq \sigma^2 \mathbb{E}\left[\max_{1 \leq i \leq n} W_{ni}^\infty(\mathbf{x})\right], \end{aligned}$$

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 can note that

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

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

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

So,

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

since $a_n/2^k \geq 4$. Consequently, we obtain

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

Optimal parameter choice We now want to prove inequality (4) in Theorem 3.1. Regarding Theorem 3.1, we want to find the optimal value of tree depth, in order to obtain the best rate of convergence for the forest estimate.

Let $c_1 = \frac{2\sigma^2}{n}$ and $c_2 = dL^2C_1$ and $\beta = \left(1 - \frac{3}{4d}\right)$. Then,

$$\mathbb{E}[m_{\infty,n}(\mathbf{x}) - m(\mathbf{x})]^2 \leq c_1 2^k + c_2 \beta^k.$$

Let $f : x \mapsto c_1 e^{x \ln 2} + c_2 e^{x \ln(\beta)}$. Thus,

$$\begin{aligned} f'(x) &= c_1 \ln 2 e^{x \ln 2} + c_2 \ln(\beta) e^{x \ln(\beta)} \\ &= c_1 \ln 2 e^{x \ln 2} \left(1 + \frac{c_2 \ln(\beta)}{c_1 \ln 2} e^{x(\ln(\beta) - \ln 2)}\right). \end{aligned}$$

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

$$\begin{aligned} f'(x^*) &= 0 \\ \iff x^* &= \frac{1}{\ln 2 - \ln(\beta)} \ln\left(-\frac{c_2 \ln(\beta)}{c_1 \ln 2}\right) \\ \iff x^* &= \frac{1}{\ln 2 - \ln(\beta)} \left[\ln\left(\frac{1}{c_1}\right) + \ln\left(-\frac{c_2 \ln(\beta)}{\ln 2}\right)\right] \\ \iff x^* &= \frac{1}{\ln 2 - \ln\left(1 - \frac{3}{4d}\right)} \left[\ln(n) + \ln\left(-\frac{dL^2C_1 \ln\left(1 - \frac{3}{4d}\right)}{2\sigma^2 \ln 2}\right)\right] \\ \iff x^* &= \frac{1}{\ln 2 - \ln \beta} \left[\ln(n) + C_2\right], \end{aligned}$$

where $C_2 = \ln\left(-\frac{dL^2C_1 \ln\left(1 - \frac{3}{4d}\right)}{2\sigma^2 \ln 2}\right)$.

Consequently,

$$\begin{aligned} \mathbb{E}[m_{\infty,n}(\mathbf{x}) - m(\mathbf{x})]^2 &\leq c_1 \exp(x^* \ln 2) + c_2 \exp(x^* \ln \beta) \\ &\leq c_1 \exp\left(\frac{1}{\ln 2 - \ln \beta} \left[\ln(n) + C_2\right] \ln 2\right) \\ &\quad + c_2 \exp\left(\frac{1}{\ln 2 - \ln \beta} \left[\ln(n) + C_2\right] \ln \beta\right) \\ &\leq c_1 \exp\left(\frac{C_2 \ln 2}{\ln 2 - \ln \beta}\right) \exp\left(\frac{\ln 2}{\ln 2 - \ln \beta} \ln(n)\right) \\ &\quad + c_2 \exp\left(\frac{C_2 \ln \beta}{\ln 2 - \ln \beta}\right) \exp\left(\frac{\ln \beta}{\ln 2 - \ln \beta} \ln(n)\right) \\ &\leq \tilde{c}_1 n^{\frac{\ln 2}{\ln 2 - \ln \beta} - 1} + \tilde{c}_2 n^{\frac{\ln \beta}{\ln 2 - \ln \beta}} \\ &\leq C_4 n^{\frac{\ln\left(1 - \frac{3}{4d}\right)}{\ln 2 - \ln\left(1 - \frac{3}{4d}\right)}}, \end{aligned} \tag{7}$$

where $C_4 = \tilde{c}_1 + \tilde{c}_2$ with $\tilde{c}_1 = 2\sigma^2 \exp\left(\frac{C_2 \ln 2}{\ln 2 - \ln \beta}\right)$ and $\tilde{c}_2 = C_2 \exp\left(\frac{C_2 \ln \beta}{\ln 2 - \ln \beta}\right)$. Note that this analysis is valid only for $a_n 2^{-k_n} \geq 4$, that is

$$a_n \geq 4.2^{\frac{C_2}{\ln 2 - \ln \beta}} \cdot n^{\frac{\ln 2}{\ln 2 - \ln \beta}},$$

where, simple calculations show that

$$2^{\frac{C_3}{\ln 2 - \ln \beta}} = \left(\frac{3L^2 e^{12/(4d-3)}}{8\sigma^2 \ln 2} \right)^{\frac{\ln 2}{\ln 2 - \ln \beta}}.$$

Thus, the upper bound (7) is valid if $a_n \geq C_3 n^{\frac{\ln 2}{\ln 2 - \ln \beta}}$, where

$$C_3 = 4 \left(\frac{3L^2 e^{12/(4d-3)}}{8\sigma^2 \ln 2} \right)^{\frac{\ln 2}{\ln 2 - \ln \beta}}.$$

□

We set up now Lemma 1 about the length of a cell that we used to bound the approximation error.

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

$$\mathbb{E} \left[V_\ell(\mathbf{x}, \Theta)^2 \right] \leq C_1 \left(1 - \frac{3}{4d} \right)^k,$$

with $C_1 = \exp(12/(4d - 3))$.

Proof of Lemma 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 ℓ -th side of the cell containing \mathbf{x} , this quantity can be written as a product of independent beta distributions:

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

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 j -th split of the cell containing \mathbf{x} is performed along the

ℓ -th dimension (and 0 otherwise). Consequently,

$$\begin{aligned}
\mathbb{E}[V_\ell(\mathbf{x}, \Theta)^2] &= \prod_{j=1}^k \mathbb{E} \left[[B(n_j + 1, n_{j-1} - n_j)]^{2\delta_{\ell,j}(\mathbf{x}, \Theta)} \right] \\
&= \prod_{j=1}^k \mathbb{E} \left[\mathbb{E} \left[[B(n_j + 1, n_{j-1} - n_j)]^{2\delta_{\ell,j}(\mathbf{x}, \Theta)} \mid \delta_{\ell,j}(\mathbf{x}, \Theta) \right] \right] \\
&= \prod_{j=1}^k \mathbb{E} \left[\mathbb{1}_{\delta_{\ell,j}(\mathbf{x}, \Theta)=0} + \mathbb{E}[B(n_j + 1, n_{j-1} - n_j)]^2 \mathbb{1}_{\delta_{\ell,j}(\mathbf{x}, \Theta)=1} \right] \\
&= \prod_{j=1}^k \left(\frac{d-1}{d} + \frac{1}{d} \mathbb{E}[B(n_j + 1, n_{j-1} - n_j)]^2 \right) \\
&= \prod_{j=1}^k \left(\frac{d-1}{d} + \frac{1}{d} \frac{(n_j + 1)(n_j + 2)}{(n_{j-1} + 1)(n_{j-1} + 2)} \right) \\
&\leq \prod_{j=1}^k \left(\frac{d-1}{d} + \frac{1}{4d} \frac{(n_{j-1} + 2)(n_{j-1} + 4)}{(n_{j-1} + 1)(n_{j-1} + 2)} \right) \\
&\leq \prod_{j=1}^k \left(1 - \frac{1}{d} + \frac{1}{4d} \frac{n_{j-1} + 4}{n_{j-1} + 1} \right), \tag{8}
\end{aligned}$$

where the first inequality stems from the relation $n_j \leq n_{j-1}/2$ for all $j \in \{1, \dots, k\}$. We have the following inequalities.

$$\begin{aligned}
\frac{n_{j-1} + 4}{n_{j-1} + 1} &\leq \frac{a_n + 2^{j+1}}{a_n - 2^{j-1}} = \frac{a_n + 2^{j+1}}{a_n(1 - \frac{2^{j-1}}{a_n})} \\
&\leq \frac{a_n + 2^{j+1}}{a_n} \left(1 + \frac{2^{j-1}}{a_n} \frac{1}{1 - \frac{2^{j-1}}{a_n}} \right) \\
&\leq \left(1 + \frac{2^{j+1}}{a_n} \right)^2,
\end{aligned}$$

since

$$\frac{2^{j-1}}{a_n} \leq \frac{2^{k-1}}{a_n} \leq \frac{1}{2}.$$

Going back to inequality (8), we find

$$\begin{aligned}
\mathbb{E}[V_\ell(\mathbf{x}, \Theta)^2] &\leq \prod_{j=1}^k \left[1 - \frac{1}{d} + \frac{1}{4d} \left(1 + \frac{2^{j+1}}{a_n} \right)^2 \right] \\
&\leq \prod_{j=1}^k \left[1 - \frac{3}{4d} + \frac{3}{d} \frac{2^{j-1}}{a_n} \right] \\
&\leq \prod_{j=1}^k \left[1 - \frac{3}{4d} + \frac{3}{d} \frac{2^k}{a_n} 2^{j-k} \right] \\
&\leq \prod_{j=0}^{k-1} \left[1 - \frac{3}{4d} + \frac{3}{d} 2^{-j-1} \right].
\end{aligned}$$

Moreover, we can notice that

$$\begin{aligned} \ln \left(\prod_{j=0}^{k-1} \left[1 - \frac{3}{4d} + \frac{3}{d} 2^{-j-1} \right] \right) &= k \ln \left(1 - \frac{3}{4d} \right) + \sum_{j=0}^{k-1} \ln \left(1 + 6 \frac{2^{-j}}{4d-3} \right) \\ &\leq k \ln \left(1 - \frac{3}{4d} \right) + \frac{12}{4d-3}. \end{aligned}$$

This yields to the desired upper bound

$$\mathbb{E} \left[V_l(\mathbf{x}, \Theta)^2 \right] \leq C_1 \left(1 - \frac{3}{4d} \right)^k,$$

with $C_1 = \exp(12/(4d-3))$.

□

The proofs of Corollary 1 and 2 are straightforward.

References