A Random Forest Guided Tour

Gérard Biau
Sorbonne Universités, UPMC Univ Paris 06, F-75005, Paris, France
& Institut Universitaire de France
gerard.biau@upmc.fr

Erwan Scornet
Sorbonne Universités, UPMC Univ Paris 06, F-75005, Paris, France
erwan.scornet@upmc.fr

Abstract

The random forest algorithm, proposed by L. Breiman in 2001, has been extremely successful as a general purpose classification and regression method. The approach, which combines several randomized decision trees and aggregates their predictions by averaging, has shown excellent performance in settings where the number of variables is much larger than the number of observations. Moreover, it is versatile enough to be applied to large-scale problems, is easily adapted to various ad-hoc learning tasks, and returns measures of variable importance. The present article reviews the most recent theoretical and methodological developments for random forests. Emphasis is placed on the mathematical forces driving the algorithm, with special attention given to the selection of parameters, the resampling mechanism, and variable importance measures. This review is intended to provide non-experts easy access to the main ideas.

Index Terms — Random forests, randomization, resampling, parameter tuning, variable importance.

2010 Mathematics Subject Classification: 62G05, 62G20.

1 Introduction

To take advantage of the sheer size of modern data sets, we now need learning algorithms that scale with the volume of information, while maintaining sufficient statistical efficiency. Random forests, devised by L. Breiman in the early 2000s (Breiman, 2001), are part of the list of the most successful methods currently available to handle data in these cases. This supervised

learning procedure, influenced by the early work of Amit and Geman (1997), Ho (1998), and Dietterich (2000), operates according to the simple but effective "divide and conquer" principle: sample small fractions of the data, grow a randomized tree predictor on each small piece, then paste (aggregate) these predictors together.

What has greatly contributed to the popularity of forests is the fact that they can be applied to a wide range of prediction problems and have few parameters to tune. Aside from being simple to use, the method is generally recognized for its accuracy and its ability to deal with small sample sizes and high-dimensional feature spaces. At the same time, it is easily parallelizable and has therefore the potential to deal with large real-life systems. The corresponding `R` package `randomForest` can be freely downloaded on the CRAN website (http://www.r-project.org), while a MapReduce (Jeffrey and Sanja, 2008) open source implementation called *Partial Decision Forests* is available on the Apache Mahout website at https://mahout.apache.org. This allows the building of forests using large data sets as long as each partition can be loaded into memory.

The random forest methodology has been successfully involved in various practical problems, including a data science hackathon on air quality prediction (http://www.kaggle.com/c/dsg-hackathon), chemoinformatics (Svetnik et al., 2003), ecology (Prasad et al., 2006; Cutler et al., 2007), 3D object recognition (Shotton et al., 2011) and bioinformatics (Díaz-Uriarte and de Andrés, 2006), just to name a few. J. Howard (Kaggle) and M. Bowles (Biomatica) claim in Howard and Bowles (2012) that *ensembles of decision trees—often known as "random forests"—have been the most successful general-purpose algorithm in modern times*, while H. Varian, Chief Economist at Google, advocates in Varian (2014) the use of random forests in econometrics.

On the theoretical side, the story of random forests is less conclusive and, despite their extensive use, little is known about the mathematical properties of the method. The most celebrated theoretical result is that of Breiman (2001), which offers an upper bound on the generalization error of forests in terms of correlation and strength of the individual trees. This was followed by a technical note (Breiman, 2004), which focuses on a stylized version of the original algorithm (see also Breiman, 2000a,b). A critical step was subsequently taken by Lin and Jeon (2006), who highlighted an interesting connection between random forests and a particular class of nearest neighbor predictors, further developed by Biau and Devroye (2010). In recent years, various theoretical studies have been performed (e.g., Meinshausen, 2006;

Biau et al., 2008; Ishwaran and Kogalur, 2010; Biau, 2012; Genuer, 2012; Zhu et al., 2012), analyzing more elaborate models and moving ever closer to the practical situation. Recent attempts towards narrowing the gap between theory and practice include that of Denil et al. (2013), who prove the first consistency result for online random forests, and Mentch and Hooker (2014b) and Wager (2014), who study the asymptotic distribution of forests.

The difficulty in properly analyzing random forests can be explained by the black-box flavor of the method, which is indeed a subtle combination of different components. Among the forests' essential ingredients, both bagging (Breiman, 1996) and the Classification And Regression Trees (CART)-split criterion (Breiman et al., 1984) play critical roles. Bagging (a contraction of bootstrap-aggregating) is a general aggregation scheme, which generates bootstrap samples from the original data set, constructs a predictor from each sample, and decides by averaging. It is one of the most effective computationally intensive procedures to improve on unstable estimates, especially for large, high-dimensional data sets, where finding a good model in one step is impossible because of the complexity and scale of the problem (Bühlmann and Yu, 2002; Kleiner et al., 2012; Wager et al., 2013). As for the CART-split criterion, it originates from the influential CART algorithm of Breiman et al. (1984), and is used in the construction of the individual trees to choose the best cuts perpendicular to the axes. At each node of each tree, the best cut is selected by optimizing the CART-split criterion, based on the so-called *Gini impurity* (for classification) or the prediction squared error (for regression).

However, while bagging and the CART-splitting scheme play key roles in the random forest mechanism, both are difficult to analyze with rigorous mathematics, thereby explaining why theoretical studies have so far considered simplified versions of the original procedure. This is often done by simply ignoring the bagging step and/or replacing the CART-split selection by a more elementary cut protocol. As well as this, in Breiman's (2001) forests, each leaf (that is, a terminal node) of individual trees contains a fixed prespecified number of observations (this parameter is usually chosen between 1 and 5). Disregarding the subtle combination of all these components, most authors have focused on stylized, data-independent procedures, thus creating a gap between theory and practice.

The goal of this survey is to embark the reader on a guided tour of random forests. We focus on the theory behind the algorithm, trying to give an overview of major theoretical approaches while discussing their inherent pros and cons. For a more methodological review covering applied aspects of random forests, we refer to the surveys by Criminisi et al. (2011) and Boulesteix

et al. (2012). We start by gently introducing the mathematical context in Section 2 and describe in full detail Breiman's (2001) original algorithm. Section 3 focuses on the theory for a simplified forest model called *purely random forests*, and emphasizes the connections between forests, nearest neighbor estimates and kernel methods. Section 4 provides some elements of theory about resampling mechanisms, the splitting criterion and the mathematical forces at work in Breiman's approach. Section 5 is devoted to the theoretical aspects of associated variable selection procedures. Lastly, Section 6 discusses various extensions to random forests including online learning, survival analysis and clustering problems.

2 The random forest estimate

2.1 Basic principles

As mentioned above, the random forest mechanism is versatile enough to deal with both supervised classification and regression tasks. However, to keep things simple, we focus in this introduction on regression analysis, and only briefly survey the classification case.

Our goal in this section is to provide a concise but mathematically precise presentation of the algorithm for building a random forest. The general framework is nonparametric regression estimation, in which an input random vector $\mathbf{X} \in [0,1]^p$ is observed, and the goal is to predict the square integrable random response $Y \in \mathbb{R}$ by estimating the regression function $m(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$. With this aim in mind, we assume we are given a training sample $\mathcal{D}_n = (\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$ of independent random variables distributed the same as the independent prototype pair $(\mathbf{X}, Y)$. The goal is to use the data set $\mathcal{D}_n$ to construct an estimate $m_n : [0,1]^p \to \mathbb{R}$ of the function m. In this respect, we say that the regression function estimate m_n is (mean squared error) consistent if $\mathbb{E}[m_n(\mathbf{X}) - m(\mathbf{X})]^2 \to 0$ as $n \to \infty$ (the expectation is evaluated over $\mathbf{X}$ and the sample $\mathcal{D}_n$).

A random forest is a predictor consisting of a collection of M randomized regression trees. For the j-th tree in the family, the predicted value at the query point $\mathbf{x}$ is denoted by $m_n(\mathbf{x}; \Theta_j, \mathcal{D}_n)$, where $\Theta_1, \ldots, \Theta_M$ are independent random variables, distributed the same as a generic random variable Θ and independent of $\mathcal{D}_n$. In practice, the variable Θ is used to resample the training set prior to the growing of individual trees and to select the successive directions for splitting—more precise definitions will be given later.

At this stage, we note that the trees are combined to form the (finite) forest estimate

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

In the `R` package `randomForest`, the default value of M (the number of trees in the forest) is `ntree` $= 500$. Since M may be chosen arbitrarily large (limited only by available computing resources), it makes sense, from a modeling point of view, to let M tends to infinity, and consider instead of (1) the (infinite) forest estimate

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

In this definition, $\mathbb{E}_\Theta$ denotes the expectation with respect to the random parameter Θ, conditional on $\mathcal{D}_n$. In fact, the operation "$M\to\infty$" is justified by the law of large numbers, which asserts that almost surely, conditional on $\mathcal{D}_n$,

$$\lim_{M\to\infty}m_{M,n}(\mathbf{x};\Theta_1,\ldots,\Theta_M,\mathcal{D}_n)=m_{\infty,n}(\mathbf{x};\mathcal{D}_n)$$

(see for instance Breiman, 2001, and Scornet, 2014, for more information on this limit calculation). In the following, to lighten notation we will simply write $m_{\infty,n}(\mathbf{x})$ instead of $m_{\infty,n}(\mathbf{x};\,\mathcal{D}_n)$.

Classification. In the (binary) supervised classification problem (Devroye et al., 1996), the random response Y takes values in $\{0,1\}$ and, given $\mathbf{X}$, one has to guess the value of Y. A classifier or classification rule m_n is a Borel measurable function of $\mathbf{x}$ and $\mathcal{D}_n$ that attempts to estimate the label Y from $\mathbf{x}$ and $\mathcal{D}_n$. In this framework, one says that the classifier m_n is consistent if its conditional probability of error

$$L(m_n)=\mathbb{P}[m_n(\mathbf{X})\neq Y|\mathcal{D}_n]$$

satisfies

$$\lim_{n\to\infty}\mathbb{E}L(m_n)=L^\star,$$

where $L^\star$ is the error of the optimal—but unknown—Bayes classifier:

$$m^\star(\mathbf{x})=\begin{cases}1 & \text{if } \mathbb{P}[Y=1|\mathbf{X}=\mathbf{x}]>\mathbb{P}[Y=0|\mathbf{X}=\mathbf{x}]\\ 0 & \text{otherwise.}\end{cases}$$

In the classification situation, the random forest classifier is obtained via a majority vote among the classification trees, that is,

$$m_{M,n}(\mathbf{x};\Theta_1,\ldots,\Theta_M,\mathcal{D}_n)=\begin{cases}1 & \text{if } \frac{1}{M}\sum_{j=1}^{M}m_n(\mathbf{x};\Theta_j,\mathcal{D}_n)>1/2\\ 0 & \text{otherwise.}\end{cases}$$

2.2 Algorithm

We now provide some insight on how the individual trees are constructed and how randomness kicks in. In Breiman's (2001) original forests, each node of a single tree is associated with a hyperrectangular cell. At each step of the tree construction, the collection of cells forms a partition of $[0,1]^p$. The root of the tree is $[0,1]^p$ itself, and the terminal nodes (or leaves), taken together, form a partition of $[0,1]^p$. If a leaf represents region A, then the regression tree outputs on A the average of all Y_i for which the corresponding $\mathbf{X}_i$ falls in A. Algorithm 1 describes in full detail how to compute a forest's prediction.

Algorithm 1 may seem a bit complicated at first sight, but the underlying ideas are simple. We start by noticing that this algorithm has three important parameters:

1. $a_n \in \{1,\dots,n\}$: the number of sampled data points in each tree;
2. `mtry` $\in \{1,\dots,p\}$: the number of possible directions for splitting at each node of each tree;
3. `nodesize` $\in \{1,\dots,n\}$: the number of examples in each cell below which the cell is not split.

The algorithm works by growing M different (randomized) trees as follows. Prior to the construction of each tree, a_n observations are drawn at random with replacement from the original data set; then, at each cell of each tree, a split is performed by maximizing the CART-criterion (see below); lastly, construction of individual trees is stopped when each cell contains less than `nodesize` points. By default in the regression mode, the parameter `mtry` is set to $p/3$, a_n is set to n, and `nodesize` is set to 5. The role and influence of these three parameters on the accuracy of the method will be thoroughly discussed in the next section.

We still have to describe how the CART-split criterion operates. With this aim in mind, we let A be a generic cell and denote by $N_n(A)$ the number of data points falling in A. A cut in A is a pair (j,z), where j is some value (dimension) from $\{1,\dots,p\}$ and z the position of the cut along the j-th coordinate, within the limits of A. Let $\mathcal{C}_A$ be the set of all such possible cuts in A. Then, with the notation $\mathbf{X}_i = (\mathbf{X}_i^{(1)},\dots,\mathbf{X}_i^{(p)})$, for any $(j,z) \in \mathcal{C}_A$, the CART-split criterion takes the form

Algorithm 1: Breiman's random forest predicted value at $\mathbf{x}$.

Input: Training set $\mathcal{D}_n$, number of trees $M > 0$, $a_n \in \{1, \ldots, n\}$, `mtry` $\in \{1, \ldots, p\}$, `nodesize` $\in \{1, \ldots, n\}$, and $\mathbf{x} \in [0,1]^p$.

Output: Prediction of the random forest at $\mathbf{x}$.

for $j = 1, \ldots, M$ **do**
  Select a_n points, with replacement, uniformly in $\mathcal{D}_n$.
  Set $\mathcal{P}_0 = \{[0,1]^p\}$ the partition associated with the root of the tree.
  For all $1 \leq \ell \leq a_n$, set $\mathcal{P}_\ell = \emptyset$.
  Set $n_{nodes} = 1$ and $level = 0$.
  while $n_{nodes} < a_n$ **do**
    if $\mathcal{P}_{level} = \emptyset$ **then**
      $level = level + 1$
    else
      Let A be the first element in $\mathcal{P}_{level}$.
      if A *contains less than* `nodesize` *points* **then**
        $\mathcal{P}_{level} \leftarrow \mathcal{P}_{level} \backslash \{A\}$
        $\mathcal{P}_{level+1} \leftarrow \mathcal{P}_{level+1} \cup \{A\}$
      else
        Select uniformly, without replacement, a subset $\mathcal{M}_{\text{try}} \subset \{1, \ldots, p\}$ of cardinality `mtry`.
        Select the best split in A by optimizing the CART-split criterion along the coordinates in $\mathcal{M}_{\text{try}}$ *(see text for details)*.
        Cut the cell A according to the best split. Call A_L and A_R the two resulting cells.
        $\mathcal{P}_{level} \leftarrow \mathcal{P}_{level} \backslash \{A\}$
        $\mathcal{P}_{level+1} \leftarrow \mathcal{P}_{level+1} \cup \{A_L\} \cup \{A_R\}$
        $n_{nodes} = n_{nodes} + 1$
      end
    end
  end
  Compute the predicted value $m_n(\mathbf{x}; \Theta_j, \mathcal{D}_n)$ at $\mathbf{x}$ equal to the average of the Y_i falling in the cell of $\mathbf{x}$ in partition $\mathcal{P}_{level} \cup \mathcal{P}_{level+1}$.
end
Compute the random forest estimate $m_{M,n}(\mathbf{x}; \Theta_1, \ldots, \Theta_M, \mathcal{D}_n)$ at the query point $\mathbf{x}$ according to (1).

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

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 belonging to A (resp., A_L, A_R), with the convention $0/0 = 0$. For each cell A, the best cut $(j_n^\star, z_n^\star)$ is selected by maximizing $L_n(j,z)$ over $\mathcal{M}_{try}$ and $\mathcal{C}_A$; that is,

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

(To remove some of the ties in the argmax, the best cut is always performed in the middle of two consecutive data points.)

Thus, at each cell of each tree, the algorithm chooses uniformly at random `mtry` coordinates in $\{1, \ldots, p\}$, evaluates criterion (2) over all possible cuts in the `mtry` directions, and returns the best one. The quality measure (2) is the criterion used in the most influential CART algorithm of Breiman et al. (1984). This criterion measures the (renormalized) difference between the empirical variance in the node before and after a cut is performed—the only difference here is that it is evaluated over a subset $\mathcal{M}_{\text{try}}$ of randomly selected coordinates, and **not** over the whole range $\{1, \ldots, p\}$. However, contrary to the CART algorithm, the individual trees are not pruned, and the final cells have a cardinality that does not exceed `nodesize`. Also, each tree is constructed on a subset of a_n examples picked within the initial sample, **not** on the whole sample $\mathcal{D}_n$. When $a_n = n$, the algorithm runs in bootstrap mode, whereas $a_n < n$ corresponds to subsampling (with replacement). Last but not least, the process is repeated M (a large number) times.

Classification. In the classification case, if a leaf represents region A, then a randomized tree classifier takes the simple form

$$m_n(\mathbf{x}; \Theta_j, \mathcal{D}_n) = \begin{cases} 1 & \text{if } \sum_{i=1}^n \mathbf{1}_{\mathbf{X}_i \in A, Y_i = 1} > \sum_{i=1}^n \mathbf{1}_{\mathbf{X}_i \in A, Y_i = 0}, \quad \mathbf{x} \in A \\ 0 & \text{otherwise.} \end{cases}$$

That is, in each leaf, a majority vote is taken over all $(\mathbf{X}_i, Y_i)$ for which $\mathbf{X}_i$ is in the same region. Ties are broken, by convention, in favor of class 0. Algorithm 1 can be easily adapted to do classification by modifying the

CART-split criterion for the binary setting. For any cell A, let $p_{0,n}(A)$ (resp., $p_{1,n}(A)$) be the empirical probability that a data point with label 0 (resp. label 1) falls into A. Then, for any $(j, z) \in \mathcal{C}_A$, the classification CART-split criterion takes the form

$$L_{\text{class},n}(j, z) = p_{0,n}(A)p_{1,n}(A) - \frac{N_n(A_L)}{N_n(A)} \times p_{0,n}(A_L)p_{1,n}(A_L) - \frac{N_n(A_R)}{N_n(A)} \times p_{0,n}(A_R)p_{1,n}(A_R). \tag{3}$$

This criterion is based on the so-called *Gini impurity measure* $2p_{0,n}(A)p_{1,n}(A)$ (Breiman et al., 1984), which has a simple interpretation. Instead of using the majority vote to classify a data point that falls in cell A, one can use the rule that assigns an observation, selected at random from the node, to label ℓ with probability $p_{\ell,n}(A)$, for $j \in \{0, 1\}$. The estimated probability that the item has actually label ℓ is $p_{\ell,n}(A)$. Therefore the estimated probability of misclassification under this rule is the Gini index $2p_{1,n}(A)p_{2,n}(A)$. When dealing with classification problems, it is usually recommended to set `nodesize` $= 1$ and `mtry` $= \sqrt{p}$ (see, e.g., Liaw and Wiener, 2002).

2.3 Parameter tuning

Literature focusing on tuning the parameters M, `mtry`, `nodesize` and a_n is unfortunately rare, with the notable exception of Díaz-Uriarte and de Andrés (2006), Bernard et al. (2008), and Genuer et al. (2010). It is easy to see that the forest's variance decreases as M grows. Thus, more accurate predictions are likely to be obtained by choosing a large number of trees. It is interesting to note that picking a large M does not lead to overfitting, since finite forests converge to infinite ones (Breiman, 2001). However, the computational cost for inducing a forest increases linearly with M, so a good choice results from a trade-off between computational complexity (M should not be too large for the computations to finish in a reasonable time) and accuracy (M must be large enough for predictions to be stable). In this respect, Díaz-Uriarte and de Andrés (2006) argue that the value of M is irrelevant (provided that M is large enough) in a prediction problem involving microarray data sets, where the aim is to classify patients according to their genetic profiles (typically, less than one hundred patients for several thousand genes). For more details we refer the reader to Genuer et al. (2010), who offer a thorough discussion on the choice of this parameter in various regression problems. Another interesting and related approach is by Latinne et al. (2001), who propose a

simple procedure that determines *a priori* a minimum number of tree estimates to combine in order to obtain a prediction accuracy level similar to that obtained with a larger forest. Their experimental results show that it is possible to significantly limit the number of trees.

In the `R` package `randomForest`, the default value of the parameter `nodesize` is 1 for classification and 5 for regression. These values are often reported to be good choices (e.g., Díaz-Uriarte and de Andrés, 2006), despite the fact that this is not supported by solid theory. The effect of `mtry` has been thoroughly investigated in Díaz-Uriarte and de Andrés (2006), who show that this parameter has a little impact on the performance of the method, though larger values may be associated with a reduction in the predictive performance. On the other hand, Genuer et al. (2010) claim that the default value of `mtry` is either optimal or too small. Therefore, a conservative approach is to take `mtry` as large as possible (limited by available computing resources) and set `mtry` $= p$ (recall that p is the dimension of the $\mathbf{X}_i$). A data-driven choice of `mtry` is implemented in the algorithm *Forest-RK* of Bernard et al. (2008).

3 Simplified models and local averaging estimates

3.1 Simplified models

Despite their widespread use, a gap remains between the theoretical understanding of random forests and their practical performance. This algorithm, which relies on complex data-dependent mechanisms, is difficult to analyze and its basic mathematical properties are still not well understood.

This state of affairs has led to polarization between theoretical and empirical contributions to the literature. Empirically focused papers describe elaborate extensions to the basic random forest framework, adding domain-specific refinements that push the state of the art in performance, but come with no clear guarantees. In contrast, most theoretical papers focus on simplifications or stylized versions of the standard algorithm, where the mathematical analysis is more tractable.

A basic framework to assess the theoretical properties of forests involves models that are calibrated independently of the training set $\mathcal{D}_n$. This family of simplified models is often called *purely random forests*. A widespread

example is the *centered forest*, whose principle is as follows: (i) there is no bootstrap step; (ii) at each node of each individual tree, a coordinate is uniformly chosen in $\{1, \ldots, p\}$; and (iii) a split is performed at the center of the cell along the selected coordinate. The operations (ii)-(iii) are recursively repeated k times, where $k \in \mathbb{N}$ is a parameter of the algorithm. The procedure stops when a full binary tree with k levels is reached, so that each tree ends up with exactly 2^k leaves. The parameter k acts as a smoothing parameter that controls the size of the terminal cells (see Figure 1 for an example in two dimensions). It should be chosen large enough in order to detect local changes in the distribution, but not too much to guarantee an effective averaging process in the leaves. In *uniform random forests*, a variant of centered forests, cuts are performed uniformly at random over the range of the selected coordinate, not at the center. Modulo some minor modifications, their analysis is similar.

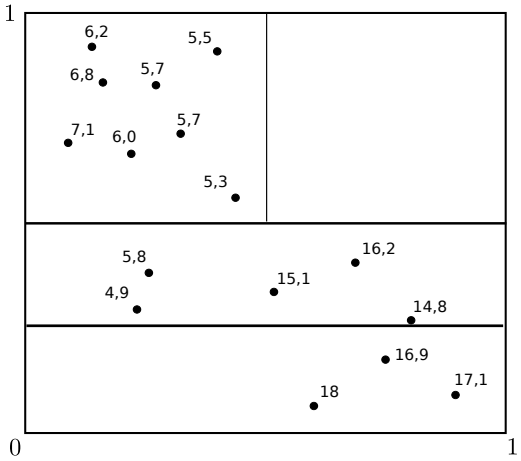

Figure 1: A centered tree at level 2.

The centered forest rule was formally analyzed in Biau et al. (2008) and Scornet (2014), who proved that the method is consistent (both for classification and regression) provided $k \to \infty$ and $n/2^k \to \infty$. The proof relies on a general consistency result for random trees stated in Devroye et al. (1996, Chapter 6). If $\mathbf{X}$ is uniformly distributed in $[0, 1]^p$, then there are on average about $n/2^k$ data points per terminal node. In particular, the choice $k \approx \log n$ corresponds to obtaining a small number of examples in the leaves, in accordance with Breiman's (2001) idea that the individual trees should not be pruned. Unfortunately, this choice of k does not satisfy the condition $n/2^k \to \infty$, so something is lost in the analysis. Moreover, the bagging step

is absent, and forest consistency is obtained as a by-product of tree consistency. Overall, this model does not demonstrate the benefit of using forests in place of individual trees and is too simple to explain the mathematical forces driving Breiman's forests.

The rates of convergence of centered forests are discussed in Breiman (2004) and Biau (2012). In their approaches, the target regression function $m(\mathbf{X}) = \mathbb{E}[Y|\mathbf{X}]$, which is originally a function of $\mathbf{X} = (X^{(1)}, \ldots, X^{(p)})$, is assumed to depend only on a nonempty subset $\mathcal{S}$ (for $\mathcal{S}$trong) of the p features. Thus, letting $\mathbf{X}_{\mathcal{S}} = (X^{(j)} : j \in \mathcal{S})$, we have

$$m(\mathbf{X}) = \mathbb{E}[Y|\mathbf{X}_{\mathcal{S}}].$$

The variables of the remaining set $\{1, \ldots, p\}\backslash\mathcal{S}$ have no influence on the response Y and can be safely removed. In this dimension reduction scenario, the ambient dimension p can be large, much larger than the sample size n, but we believe that the representation is sparse, i.e., that a potentially small number of coordinates of m are active— the ones with indices matching the set $\mathcal{S}$. Letting $|\mathcal{S}|$ be the cardinality of $\mathcal{S}$, the value $|\mathcal{S}|$ characterizes the sparsity of the model: the smaller $|\mathcal{S}|$, the sparser m.

Breiman (2004) and Biau (2012) proved that if the random trees are grown by using coordinates in $\mathcal{S}$ with high probability, and if m satisfies a Lipschitz-type smoothness assumption, then

$$\mathbb{E}\left[m_{\infty,n}(\mathbf{X}) - m(\mathbf{X})\right]^2 = \mathrm{O}\left(n^{\frac{-0.75}{|\mathcal{S}|\log 2 + 0.75}}\right).$$

This equality shows that the rate of convergence of m_n to m depends only on the number $|\mathcal{S}|$ of strong variables, not on the ambient dimension p. This rate is strictly faster than the usual rate $n^{-2/(p+2)}$ as soon as $|\mathcal{S}| \leq \lfloor 0.54p \rfloor$. In effect, the intrinsic dimension of the regression problem is $|\mathcal{S}|$, not p, and we see that the random forest estimate cleverly adapts itself to the sparse framework. This property may be useful for high-dimensional regression, when the number of variables is much larger than the sample size. It may also explain why random forests are able to handle a large number of input variables without overfitting.

An alternative model for pure forests, called *purely uniform random forests* (PURF) is discussed in Genuer (2012). For $p = 1$, a PURF is obtained by drawing k random variables uniformly on $[0, 1]$, and subsequently dividing $[0, 1]$ into random sub-intervals. Although this construction is not exactly recursive, it is equivalent to growing a decision tree by deciding at each level which node to split with a probability equal to its length. Genuer (2012)

proves that PURF are consistent and, under a Lipschitz assumption, that the estimate satisfies

$$\mathbb{E}[m_{\infty,n}(\mathbf{X}) - m(\mathbf{X})]^2 = \mathrm{O}\left(n^{-2/3}\right).$$

This rate is minimax over the class of Lipschitz functions (Stone, 1980, 1982).

It is often acknowledged that random forests reduce the estimation error of a single tree, while maintaining the same approximation error. In this respect, Biau (2012) argues that the estimation error of centered forests tends to zero (at the slow rate $1/\log n$) even if each tree is fully grown (i.e., $k \approx \log n$). This result is a consequence of the tree-averaging process, since the estimation error of an individual fully grown tree does not tend to zero. Unfortunately, the choice $k \approx \log n$ is too large to ensure consistency of the corresponding forest, whose approximation error remains constant. Similarly, Genuer (2012) shows that the estimation error of PURF is reduced by a factor of 0.75 compared to the estimation error of individual trees. The most recent attempt to assess the gain of forests in terms of estimation and approximation errors is by Arlot and Genuer (2014), who claim that the rate of the approximation error of certain models is faster than that of the individual trees.

3.2 Forests, neighbors and kernels

Let us consider a sequence of independent and identically distributed random variables $X_1, \ldots, X_n$. In random geometry, a random observation $\mathbf{X}_i$ is said to be a *layered nearest neighbor* (LNN) of a point $\mathbf{x}$ (from $\mathbf{X}_1, \ldots, \mathbf{X}_n$) if the hyperrectangle defined by $\mathbf{x}$ and $\mathbf{X}_i$ contains no other data points (Barndorff-Nielsen and Sobel, 1966; Bai et al., 2005; see also Devroye et al., 1996, Chapter 11, Problem 6). As illustrated in Figure 2, the number of LNN of $\mathbf{x}$ is typically larger than one and depends on the number and configuration of the sample points.

Surprisingly, the LNN concept is intimately connected to random forests. Indeed, if exactly one point is left in the leaves, then no matter what splitting strategy is used, the forest estimate at $\mathbf{x}$ is but a weighted average of the Y_i whose corresponding $\mathbf{X}_i$ are LNN of $\mathbf{x}$. In other words,

$$m_{\infty,n}(\mathbf{x}) = \sum_{i=1}^{n} W_{ni}(\mathbf{x}) Y_i, \tag{4}$$

where the weights $(W_{n1}, \ldots, W_{nn})$ are nonnegative functions of the sample $\mathcal{D}_n$ that satisfy $W_{ni}(\mathbf{x}) = 0$ if $\mathbf{X}_i$ is not an LNN of $\mathbf{x}$ and $\sum_{i=1}^{n} W_{ni} = 1$.

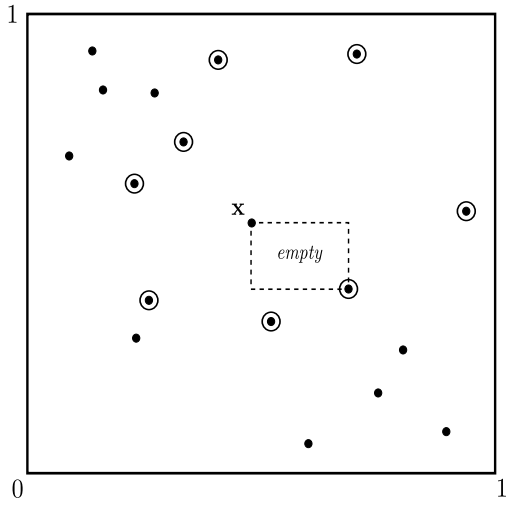

Figure 2: The layered nearest neighbors (LNN) of a point $\mathbf{x}$ in dimension $p = 2$.

This important connection was first pointed out by Lin and Jeon (2006), who proved that if $\mathbf{X}$ is uniformly distributed on $[0,1]^p$ then, provided tree growing is independent of $Y_1, \ldots, Y_n$ (such simplified models are sometimes called *non-adaptive*), we have

$$\mathbb{E}\left[m_{\infty,n}(\mathbf{X}) - m(\mathbf{X})\right]^2 = \mathrm{O}\left(\frac{1}{n_{\max}(\log n)^{p-1}}\right),$$

where $n_{\max}$ is the maximal number of points in the terminal cells (Biau and Devroye, 2010, extended this inequality to the case where $\mathbf{X}$ has a density on $[0,1]^p$). Unfortunately, the exact values of the weight vector $(W_{n1}, \ldots, W_{nn})$ attached to the original random forest algorithm are unknown, and a general theory of forests in the LNN framework is still undeveloped.

It remains however that equation (4) opens the way to the analysis of random forests via a local-averaging approach, i.e., via the average of those Y_i for which $\mathbf{X}_i$ is "close" to $\mathbf{x}$ (Györfi et al., 2002). Indeed, observe, starting from (1), that for a finite forest with M trees, we have

$$m_{M,n}(\mathbf{x}; \Theta_1, \ldots, \Theta_M) = \frac{1}{M}\sum_{j=1}^{M}\left(\sum_{i=1}^{n}\frac{Y_i \mathbb{1}_{\mathbf{X}_i \in A_n(\mathbf{x},\Theta_j)}}{N_n(\mathbf{x},\Theta_j)}\right),$$

where $A_n(\mathbf{x}, \Theta_j)$ is the cell containing $\mathbf{x}$ and $N_n(\mathbf{x}, \Theta_j) = \sum_{i=1}^{n} \mathbb{1}_{\mathbf{X}_i \in A_n(\mathbf{x},\Theta_j)}$

is the number of data points falling in $A_n(\mathbf{x}, \Theta_j)$. Thus,

$$m_{M,n}(\mathbf{x}; \Theta_1, \ldots, \Theta_M) = \sum_{i=1}^{n} W_{ni}(\mathbf{x}) Y_i,$$

where the weights $W_{ni}(\mathbf{x})$ are defined by

$$W_{ni}(\mathbf{x}) = \frac{1}{M} \sum_{j=1}^{M} \frac{\mathbb{1}_{\mathbf{X}_i \in A_n(\mathbf{x}, \Theta_j)}}{N_n(\mathbf{x}, \Theta_j)}.$$

It is easy to see that the W_{ni} are nonnegative and sum to one if the cell containing $\mathbf{x}$ is not empty. Thus, the contribution of observations falling into cells with a high density of data points is smaller than the contribution of observations belonging to less-populated cells. This remark is especially true when the forests are built independently of the data set—for example, PURF—since, in this case, the number of examples in each cell is not controlled. Next, if we let M tend to infinity, then the estimate $m_{\infty,n}$ may be written (up to some negligible terms)

$$m_{\infty,n}(\mathbf{x}) \approx \frac{\sum_{i=1}^{n} Y_i K_n(\mathbf{X}_i, \mathbf{x})}{\sum_{j=1}^{n} K_n(\mathbf{X}_j, \mathbf{x})}, \tag{5}$$

where

$$K_n(\mathbf{x}, \mathbf{z}) = \mathbb{P}_\Theta \left[\mathbf{z} \in A_n(\mathbf{x}, \Theta) \right].$$

The function $K_n(\cdot, \cdot)$ is called the *kernel* and characterizes the shape of the "cells" of the infinite random forest. The quantity $K_n(\mathbf{x}, \mathbf{z})$ is nothing but the probability that $\mathbf{x}$ and $\mathbf{z}$ are connected (i.e., they fall in the same cell) in a random tree. Therefore, the kernel K_n can be seen as a proximity measure between two points in the forest. Hence, any forest has its own metric K_n, but unfortunately the one associated with Breiman's forest is strongly data-dependent and therefore complicated to work with.

It should be noted that K_n does not necessarily belong to the family of Nadaraya-Watson-type kernels (Nadaraya, 1964; Watson, 1964), which satisfy a homogeneous property of the form $K_h(\mathbf{x}, \mathbf{z}) = \frac{1}{h} K((\mathbf{x} - \mathbf{z})/h)$ for some *smoothing parameter* $h > 0$. The analysis of estimates of the form (5) is, in general, more complicated, depending of the type of forest under investigation. For example, Scornet (2015) proved that for a centered forest defined on $[0, 1]^p$ with parameter k, we have

$$K_{k,n}(\mathbf{x}, \mathbf{z}) = \sum_{\substack{k_1, \ldots, k_p \\ \sum_{j=1}^{p} k_j = k}} \frac{k!}{k_1! \ldots k_p!} \left(\frac{1}{p} \right)^k \prod_{j=1}^{p} \mathbb{1}_{\lceil 2^{k_j} x_j \rceil = \lceil 2^{k_j} z_j \rceil}$$

($\lceil \cdot \rceil$ is the ceiling function). As an illustration, Figure 3 shows the graphical representation for $k = 1$, 2 and 5 of the function f_k defined by

$$\begin{array}{rccl} f_k: & [0,1]\times[0,1] & \to & [0,1] \\ & \mathbf{z}=(z_1,z_2) & \mapsto & K_{k,n}\big((\frac{1}{2},\frac{1}{2}),\mathbf{z}\big). \end{array}$$

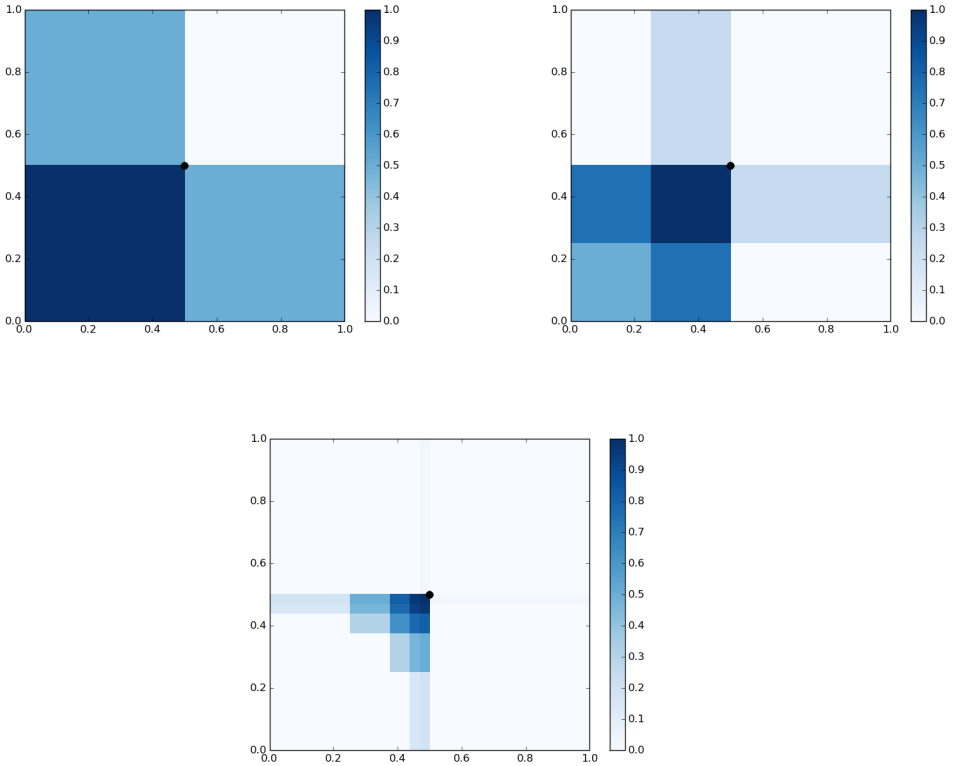

Figure 3: Representations of f_1, f_2 and f_5 in $[0,1]^2$.

The connection between forests and kernel estimates is mentioned in Breiman (2000a) and developed in detail in Geurts et al. (2006). The most recent advances in this direction are by Arlot and Genuer (2014), who show that a simplified forest model can be written as a kernel estimate, and provide its rates of convergence. On the practical side, Davies and Ghahramani (2014) highlight the fact that using Gaussian processes with a specific kernel-based random forest can empirically outperform state-of-the-art Gaussian process methods. Besides, kernel-based random forests can be used as the input for a large variety of existing kernel-type methods such as Kernel Principal Component Analysis and Support Vector Machines.

4 Theory for Breiman's forests

This section deals with Breiman's (2001) original algorithm. Since the construction of Breiman's forests depends on the whole sample $\mathcal{D}_n$, a mathematical analysis of the whole algorithm is difficult. To move forward, the individual mechanisms at work in the procedure have been investigated separately, namely the resampling step and the splitting scheme.

4.1 The resampling mechanism

The resampling step in Breiman's (2001) original algorithm is performed by choosing n times from of n points with replacement to grow the individual trees. This procedure, which traces back to the work of Efron (1982) (see also Politis et al., 1999), is called the *bootstrap* in the statistical literature. The idea of generating many bootstrap samples and averaging predictors is called *bagging* (bootstrap-aggregating). It was suggested by Breiman (1996) as a simple way to improve the performance of weak or unstable learners. Although one of the great advantages of the bootstrap is its simplicity, the theory turns out to be complex. In effect, the bootstrapped observations have a distribution that is different from the original one, as the following example shows. Assume that $\mathbf{X}$ has a density, and note that whenever the data points are sampled with replacement, then with positive probability, at least one observation from the original sample will be selected more than once. Therefore, the resulting $\mathbf{X}_i$ of the bootstrapped sample cannot have an absolutely continuous distribution.

The role of the bootstrap in random forests is still poorly understood and, to date, most analyses are doomed to replace the bootstrap by a subsampling scheme, assuming that each tree is grown with $a_n < n$ examples randomly chosen without replacement from the initial sample (Mentch and Hooker, 2014b; Wager, 2014; Scornet et al., 2015). Most of the time, the subsampling rate a_n/n is assumed to tend to zero at some prescribed rate—an assumption that excludes *de facto* the bootstrap regime. In this respect, the analysis of so-called *median random forests* by Scornet (2014) provides some insight as to the role and importance of subsampling. The assumption $a_n/n \to 0$ guarantees that every single observation pair $(\mathbf{X}_i, Y_i)$ is used in the m-th tree's construction with a probability that becomes small as n grows. It also forces the query point $\mathbf{x}$ to be disconnected from $(\mathbf{X}_i, Y_i)$ in a large proportion of trees. Indeed, if this were not the case, then the predicted value at $\mathbf{x}$ would be overly influenced by the single pair $(\mathbf{X}_i, Y_i)$, which would make the ensemble

inconsistent. In fact, the estimation error of the median forest estimate is small as soon as the maximum probability of connection between the query point and all observations is small. Thus, the assumption $a_n/n \to 0$ is but a convenient way to control these probabilities, by ensuring that partitions are dissimilar enough.

Biau and Devroye (2010) noticed that Breiman's bagging principle has a simple application in the context of nearest neighbor methods. Recall that the 1-nearest neighbor (1-NN) regression estimate sets $r_n(\mathbf{x}) = Y_{(1)}(\mathbf{x})$, where $Y_{(1)}(\mathbf{x})$ corresponds to the feature vector $\mathbf{X}_{(1)}(\mathbf{x})$ whose Euclidean distance to $\mathbf{x}$ is minimal among all $\mathbf{X}_1, \ldots, \mathbf{X}_n$. (Ties are broken in favor of smallest indices.) It is clearly not, in general, a consistent estimate (Devroye et al., 1996, Chapter 5). However, by bagging, one may turn the 1-NN estimate into a consistent one, provided that the size of resamples is sufficiently small. We proceed as follows, via a randomized basic regression estimate r_{a_n} in which $1 \leq a_n \leq n$ is a parameter. The elementary predictor r_{a_n} is the 1-NN rule for a random subsample of size a_n drawn with (or without) replacement from $\mathcal{D}_n$. We apply bagging, that is, we repeat the random sampling an infinite number of times and take the average of the individual outcomes. Thus, the bagged regression estimate $r_n^\star$ is defined by

$$r_n^\star(\mathbf{x}) = \mathbb{E}^\star \left[r_{a_n}(\mathbf{x}) \right],$$

where $\mathbb{E}^\star$ denotes expectation with respect to the resampling distribution, conditional on the data set $\mathcal{D}_n$. Biau and Devroye (2010) proved that the estimate $r_n^\star$ is universally (i.e., without conditions on the distribution of $(\mathbf{X}, Y)$) mean squared consistent, provided $a_n \to \infty$ and $a_n/n \to 0$. The proof relies on the observation that $r_n^\star$ is in fact a weighted nearest neighbor estimate (Stone, 1977) with weights

$$W_{ni} = \mathbb{P}(i\text{-th nearest neighbor of } \mathbf{x} \text{ is the 1-NN in a random selection}).$$

The connection between bagging and nearest neighbor estimation is further explored by Biau et al. (2010), who prove that the bagged estimate $r_n^\star$ achieves optimal rate of convergence over Lipschitz smoothness classes, independently from the fact that resampling is done with or without replacement.

4.2 Decision splits

The coordinate-split process of the random forest algorithm is not easy to grasp, essentially because it uses both the $\mathbf{X}_i$ and Y_i variables to make its

decision. Building upon the ideas of Bühlmann and Yu (2002), Banerjee and McKeague (2007) establish a limit law for the split location in the context of a regression model of the form $Y = m(\mathbf{X}) + \varepsilon$, where $\mathbf{X}$ is real-valued and ε an independent Gaussian noise. In essence, their result is as follows. Assume for now that the distribution of $(\mathbf{X}, Y)$ is known, and denote by $d^\star$ the (optimal) split that maximizes the theoretical CART-criterion at a given node. In this framework, the regression estimates restricted to the left and right children of the cell takes the respective forms

$$\beta^\star_{\ell,n} = \mathbb{E}[Y|X \leq d^\star] \quad \text{and} \quad \beta^\star_{r,n} = \mathbb{E}[Y|X > d^\star].$$

When the distribution of $(\mathbf{X}, Y)$ is unknown, so are $\beta^\star_\ell$, $\beta^\star_r$ and $d^\star$, and these quantities are estimated by their natural empirical counterparts:

$$(\hat{\beta}_{\ell,n}, \hat{\beta}_{r,n}, \hat{d}_n) \in \operatorname*{arg\,min}_{\beta_\ell, \beta_r, d} \sum_{i=1}^{n} \left[Y_i - \beta_\ell \mathbb{1}_{X_i \leq d} - \beta_r \mathbb{1}_{X_i > d} \right]^2.$$

Assuming that the model satisfies some regularity assumptions (in particular, $\mathbf{X}$ has a density f, and both f and m are continuously differentiable), Banerjee and McKeague (2007) prove that

$$n^{1/3}(\hat{\beta}_{\ell,n} - \beta^\star_\ell, \hat{\beta}_{r,n} - \beta^\star_r, \hat{d}_n - d^\star) \xrightarrow{\mathcal{D}} (c_1, c_2, 1) \operatorname*{arg\,max}_t Q(t), \tag{6}$$

where $\mathcal{D}$ denotes convergence in distribution, $Q(t) = aW(t) - bt^2$, and W is a standard two-sided Brownian motion process on the real line. Both a and b are positive constants that depend upon the model parameters and the unknown quantities $\beta^\star_\ell$, $\beta^\star_r$ and $d^\star$. The limiting distribution in (6) allows one to construct confidence intervals for the position of CART-splits. Interestingly, Banerjee and McKeague (2007) refer to the study of Qian et al. (2003) on the effects of phosphorus pollution in the Everglades, which uses split points in a novel way. There, the authors identify threshold levels of phosphorus concentration that are associated with declines in the abundance of certain species. In their approach, split points are not just a means to build trees and forests, but can also provide important information on the structure of the underlying distribution.

A further analysis of the behavior of forest splits is performed by Ishwaran (2013), who argues that the so-called *end-cut preference* (ECP) of the CART-splitting procedure (that is, the fact that splits along non-informative variables are likely to be near the edges of the cell—see Breiman et al., 1984) can be seen as a desirable property. Given the randomization mechanism

at work in forests, there is indeed a positive probability that none of the preselected variables at a node are informative. When this happens, and if the cut is performed, say, at the center of a side of the cell, then the sample size of the two resulting cells is drastically reduced by a factor of two—this is an undesirable property, which may be harmful for the prediction task. In other words, Ishwaran (2013) stresses that the ECP property ensures that a split along a noisy variable is performed near the edge, thus maximizing the tree node sample size and making it possible for the tree to recover from the split downstream. Ishwaran (2013) claims that this property can be of benefit even when considering a split on an informative variable, if the corresponding region of space contains little signal.

There exists a variety of random forest variants based on the CART-criterion. For example, the *Extra-Tree* algorithm of Geurts et al. (2006) consists in randomly selecting a set of split points and then choosing the split that maximizes the CART-criterion. This algorithm has similar accuracy performance while being more computationally efficient. In the *PERT* (Perfect Ensemble Random Trees) approach of Cutler and Zhao (2001), one builds perfect-fit classification trees with random split selection. While individual trees clearly overfit, the authors claim that the whole procedure is eventually consistent since all classifiers are believed to be almost uncorrelated. Let us also mention that additional randomness can be added in the tree construction by considering splits along linear combinations of features. This idea, due to Breiman (2001), has been implemented by Truong (2009) in the package `obliquetree` of statistical computing environment `R`.

4.3 Asymptotic normality and consistency

All in all, little has been proven mathematically for the original procedure of Breiman (2001). Recently, consistency and asymptotic normality of the whole algorithm were proved under simplifications of the procedure (replacing bootstrap by subsampling and simplifying the splitting step). Wager (2014) proves the asymptotic normality of the method and establishes that the infinitesimal jackknife consistently estimates the forest variance. A similar result on the asymptotic normality of finite forests, proved by Mentch and Hooker (2014b), states that whenever M (the number of trees) is allowed to vary with n, and when $a_n = o(\sqrt{n})$ and $\lim_{n\to\infty} n/M_n = 0$, then for a fixed $\mathbf{x}$,

$$\frac{\sqrt{n}(m_{M,n}(\mathbf{x};\Theta_1,\ldots,\Theta_M) - m_{\infty,n}(\mathbf{x}))}{\sqrt{a_n^2\zeta_{1,a_n}}} \xrightarrow{\mathcal{D}} N,$$

where N is a standard normal random variable,

$$\zeta_{1,a_n} = \text{Cov}\left[m_n(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{a_n}; \Theta), m_n(\mathbf{X}_1, \mathbf{X}'_2, \ldots, \mathbf{X}'_{a_n}; \Theta')\right],$$

$\mathbf{X}'_i$ an independent copy of $\mathbf{X}_i$ and Θ' an independent copy of Θ. Note that in this model, both the sample size and the number of trees grow to infinity. Recently, Scornet et al. (2015) proved a consistency result in the context of additive regression models for the pruned version of Breiman's forest. Unfortunately, the consistency of the unpruned procedure comes at the price of a conjecture regarding the behavior of the CART algorithm that is difficult to verify.

We close this section with a negative but interesting result due to Biau et al. (2008). In this example, the total number k of cuts is fixed and `mtry` $= 1$. Furthermore, each tree is built by minimizing the true probability of error at each node. Consider the joint distribution of $(\mathbf{X}, Y)$ sketched in Figure 4 and let $m(\mathbf{x}) = \mathbb{P}[Y = 1|\mathbf{X} = \mathbf{x}]$. The variable $\mathbf{X}$ has a uniform distribution on $[0,1]^2 \cup [1,2]^2 \cup [2,3]^2$ and Y is a function of $\mathbf{X}$—that is, $m(\mathbf{x}) \in \{0,1\}$ and $L^* = 0$—defined as follows. The lower left square $[0,1] \times [0,1]$ is divided into countably infinitely many vertical strips in which the strips with $m(\mathbf{x}) = 0$ and $m(\mathbf{x}) = 1$ alternate. The upper right square $[2,3] \times [2,3]$ is divided similarly into horizontal strips. The middle rectangle $[1,2] \times [1,2]$ is a 2×2 checkerboard. It is easy to see that no matter what the sequence of random selection of split directions is and no matter for how long each tree is grown, no tree will ever cut the middle rectangle and therefore the probability of error of the corresponding random forest classifier is at least $1/6$. This example illustrates that consistency of greedily grown random forests is a delicate issue. Note however that if Breiman's (2001) original algorithm is used in this example (i.e., when all cells with more than one data point in them are split) then one obtains a consistent classification rule.

5 Variable selection

5.1 Variable importance measures

Random forests can be used to rank the importance of variables in regression or classification problems via two measures of significance. The first, called *Mean Decrease Impurity* (MDI), is based on the total decrease in node impurity from splitting on the variable, averaged over all trees. The second, referred to as *Mean Decrease Accuracy* (MDA), stems from the idea that if

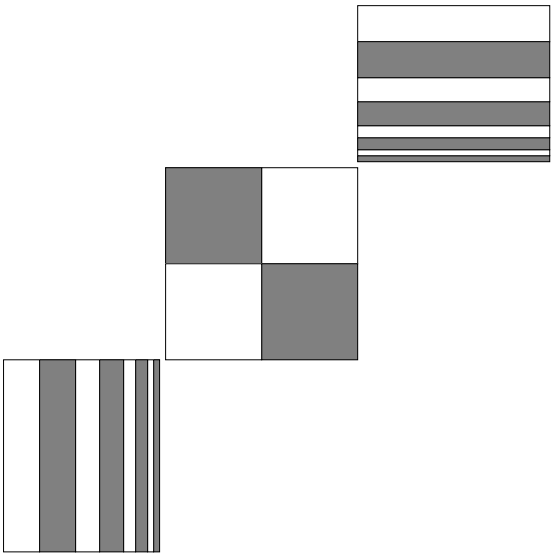

Figure 4: An example of a distribution for which greedy random forests are inconsistent. The distribution of $\mathbf{X}$ is uniform on the union of the three large squares. White areas represent the set where $m(\mathbf{x}) = 0$ and grey where $m(\mathbf{x}) = 1$.

the variable is not important, then rearranging its values should not degrade prediction accuracy.

Set $\mathbf{X} = (\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(p)})$. For a forest resulting from the aggregation of M trees, the MDI of the variable $X^{(j)}$ is defined as

$$\widehat{\mathrm{MDI}}(X^{(j)}) = \frac{1}{M} \sum_{\ell=1}^{M} \sum_{\substack{t \in \mathcal{T}_\ell \\ j^\star_{n,t} = j}} 2 p_{n,t} L_{\mathrm{reg},n}(j^\star_{n,t}, z^\star_{n,t}),$$

where $p_{n,t}(t)$ is the fraction of observations falling in the node t, $\{\mathcal{T}_\ell\}_{1 \leq \ell \leq M}$ the collection of trees in the forest, and $(j^\star_{n,t}, z^\star_{n,t})$ the split that maximizes the empirical criterion (2) in node t. Note that the same formula holds for classification random forests by replacing the criterion $L_{\mathrm{reg},n}$ by its classification counterpart $L_{\mathrm{class},n}$. Thus, the MDI of $X^{(j)}$ computes the weighted decrease of impurity corresponding to splits along the variable $X^{(j)}$ and averages this quantity over all trees.

The MDA relies on a different principle and uses the so-called *out-of-bag* error estimate. In random forests, there is no need for cross-validation or a separate test set to get an unbiased estimate of the test error. It is estimated

internally, during the run, as follows. Since each tree is constructed using a different bootstrap sample from the original data, about one-third of cases are left out of the bootstrap sample and not used in the construction of the m-th tree. In this way, for each tree, a test set—disjoint from the training set— is obtained, and averaging over all these left-out cases and over all trees is known as the out-of-bag error estimate.

To measure the importance of the j-th feature, we randomly permute the values of variable $X^{(j)}$ in the out-of-bag cases and put these cases down the tree. The MDA of $X^{(j)}$ is obtained by averaging the difference in out-of-bag error estimation before and after the permutation over all trees. In mathematical terms, consider a variable $X^{(j)}$ and denote by $\mathcal{D}_{\ell,n}$ the out-of-bag test of the ℓ-th tree and $\mathcal{D}^j_{\ell,n}$ the same data set where the values of $X^{(j)}$ have been randomly permuted. Recall that $m_n(\cdot, \Theta_\ell)$ stands for the ℓ-th tree estimate. Then, by definition,

$$\widehat{\mathrm{MDA}}(X^{(j)}) = \frac{1}{M}\sum_{\ell=1}^{M}\Big[R_n\big[m_n(\cdot,\Theta_\ell),\mathcal{D}^j_{\ell,n}\big] - R_n\big[m_n(\cdot,\Theta_\ell),\mathcal{D}_{\ell,n}\big]\Big], \qquad (7)$$

where R_n is defined for $\mathcal{D} = \mathcal{D}_{\ell,n}$ or $\mathcal{D} = \mathcal{D}^j_{\ell,n}$ by

$$R_n\big[m_n(\cdot,\Theta_\ell),\mathcal{D}\big] = \frac{1}{|\mathcal{D}|}\sum_{i:(\mathbf{X}_i,Y_i)\in\mathcal{D}}(Y_i - m_n(\mathbf{X}_i,\Theta_\ell))^2.$$

It is easy to see that the population version of $\widehat{\mathrm{MDA}}(X^{(j)})$ takes the form

$$\mathrm{MDA}^\star(X^{(j)}) = \mathbb{E}\big[Y - m_n(\mathbf{X}'_j,\Theta)\big]^2 - \mathbb{E}\big[Y - m_n(\mathbf{X},\Theta)\big]^2,$$

where $\mathbf{X}'_j = (X^{(1)},\ldots,X'^{(j)},\ldots,X^{(p)})$ and $X'^{(j)}$ is an independent copy of $X^{(j)}$. For classification purposes, the MDA still satisfies (7) with $R_n(m_n(\cdot,\Theta)$ $,\mathcal{D})$ the number of points that are correctly classified by $m_n(\cdot,\Theta)$ in $\mathcal{D}$.

5.2 Theoretical results

In the context of a pair of categorical variables $(\mathbf{X}, Y)$, where $\mathbf{X}$ takes finitely many values in, say, $\mathcal{X}_1 \times \cdots \times \mathcal{X}_d$, Louppe et al. (2013) consider totally randomized and fully developed trees. At each cell, the ℓ-th tree is grown by selecting a variable $X^{(j)}$ uniformly among the features that have not been used in the parent nodes, and by subsequently dividing the cell into $|\mathcal{X}_j|$ children (so the number of children equals the number of modalities of the

selected variable). In this framework, it can be shown that the population version of $\text{MDI}(X^{(j)})$ for a single tree satisfies

$$\text{MDI}^\star(X^{(j)}) = \sum_{k=0}^{p-1} \frac{1}{\binom{k}{p}(p-k)} \sum_{B \in \mathcal{P}_k(V^{-j})} I(X_j; Y|B),$$

where $V^{-j} = \{1, \ldots, j-1, j+1, \ldots, p\}$, $\mathcal{P}_k(V^{-j})$ the set of subsets of V^{-j} of cardinality k, and $I(X^{(j)}; Y|B)$ the *conditional mutual information* of $X^{(j)}$ and Y given the variables in B. In addition,

$$\sum_{j=1}^{p} \text{MDI}^\star(X^{(j)}) = I(X^{(1)}, \ldots, X^{(p)}; Y).$$

These results show that the information $I(X^{(1)}, \ldots, X^{(p)}; Y)$ is the sum of the importances of each variable, which can itself be made explicit using the information values $I(X^{(j)}; Y|B)$ between each variable $X^{(j)}$ and the output Y, conditional on variable subsets B of different sizes.

Louppe et al. (2013) define a variable $X^{(j)}$ as irrelevant with respect to $B \subset V = \mathcal{X}_1 \times \cdots \times \mathcal{X}_p$ whenever $I(X^{(j)}; Y|B) = 0$. Thus, $X^{(j)}$ is irrelevant if and only if $\text{MDI}^\star(X^{(j)}) = 0$. It is easy to see that if an additional irrelevant variable $X^{(p+1)}$ is added to the list of variables, then the variable importance of any of the $X^{(j)}$ computed with a single tree does not change if the tree is built with the new collection of variables $V \cup \{X^{(p+1)}\}$. In other words, building a tree with an additional irrelevant variable does not change the importances of the other variables.

The most notable results regarding MDA are due to Ishwaran (2007), who studies a slight modification of the criterion via feature noising. To add noise to a variable $X^{(j)}$, one considers a new observation $\mathbf{X}$, take $\mathbf{X}$ down the tree and stop when a split is made according to the variable $X^{(j)}$. Then the right or left child node is selected with probability $1/2$, and this procedure is repeated for each subsequent node (whether it is performed along the variable $X^{(j)}$ or not). The importance of variable $X^{(j)}$ is still computed by comparing the error of the forest with that of the "noisy" forest. Assuming that the forest is consistent and that the regression function is piecewise constant, Ishwaran (2007) gives the asymptotic behavior of $\widehat{\text{MDA}}(X^{(j)})$ when the sample size tends to infinity. This behavior is intimately related to the set of subtrees (of the initial regression tree) whose roots are split along the coordinate $X^{(j)}$.

Let us lastly mention the approach of Gregorutti et al. (2013), who computed the MDA criterion for several distributions of $(\mathbf{X}, Y)$. For example, consider

a model of the form

$$Y = m(\mathbf{X}) + \varepsilon,$$

where $(\mathbf{X}, \varepsilon)$ is a Gaussian random vector, and assume that the correlation matrix C satisfies $C = [\mathrm{Cov}(X_j, X_k)]_{1 \leq j,k \leq p} = (1-c)I_p + c\mathbb{1}\mathbb{1}^\top$ (the symbol $\top$ denotes transposition, $\mathbb{1} = (1, \ldots, 1)^\top$, and c is a constant in $(0, 1)$). Assume, in addition, that $\mathrm{Cov}(X_j, Y) = \tau_0$ for all $j \in \{1, \ldots, p\}$. Then, for all j,

$$\mathrm{MDI}^\star(X^{(j)}) = 2\left(\frac{\tau_0}{1 - c + pc}\right)^2.$$

Thus, in the Gaussian setting, the variable importance decreases as the inverse of the square of p when the number of correlated variables p increases.

5.3 Related works

The empirical properties of the MDA criterion have been extensively analyzed and compared in the statistical computing literature. Indeed, Archer and Kimes (2008), Strobl et al. (2008), Nicodemus and Malley (2009), Auret and Aldrich (2011), and Tolo̧si and Lengauer (2011) stress the negative effect of correlated variables on MDA performance. In this respect, Genuer et al. (2010) noticed that MDA is less able to detect the most relevant variables when the number of correlated features increases. Similarly, the empirical study of Archer and Kimes (2008) points out that both MDA and MDI behave poorly when correlation increases—these results have been experimentally confirmed by Auret and Aldrich (2011) and Tolo̧si and Lengauer (2011). An argument of Strobl et al. (2008) to justify the bias of MDA in the presence of correlated variables is that the algorithm evaluates the marginal importance of the variables instead of taking into account their effect conditional on each other. A way to circumvent this issue is to combine random forests and the *Recursive Feature Elimination* algorithm of Guyon et al. (2002), as in Gregorutti et al. (2013). Detecting relevant features can also be achieved via hypothesis testing (Mentch and Hooker, 2014b)—a principle that may be used to detect more complex structures of the regression function, like for instance its additivity (Mentch and Hooker, 2014a).

As for the tree building process, selecting uniformly at each cell a set of features for splitting is simple and convenient, but such procedures inevitably select irrelevant variables. Therefore, several authors have proposed modified versions of the algorithm that incorporate a data-driven weighing of

variables. For example, Kyrillidis and Zouzias (2014) study the effectiveness of non-uniform randomized feature selection in decision tree classification, and experimentally show that such an approach may be more effective compared to naive uniform feature selection. *Enriched random forests*, designed by Amaratunga et al. (2008) choose at each node the eligible subsets by weighted random sampling with the weights tilted in favor of informative features. Similarly, the *reinforcement learning trees* (RLT) of Zhu et al. (2012) build at each node a random forest to determine the variable that brings the greatest future improvement in later splits, rather than choosing the one with largest marginal effect from the immediate split.

Choosing weights can also be done via regularization. Deng and Runger (2012) propose a *Regularized Random Forest* (RRF), which penalizes selecting a new feature for splitting when its gain is similar to the features used in previous splits. Deng and Runger (2013) suggest a *Guided RRF* (GRRF), in which the importance scores from an ordinary random forest are used to guide the feature selection process in RRF. Lastly, a Garrote-style convex penalty, proposed by Meinshausen (2009), selects functional groups of nodes in trees, yielding to parcimonious estimates. We also mention the work of Konukoglu and Ganz (2014) who address the problem of controlling the false positive rate of random forests and present a principled way to determine thresholds for the selection of relevant features without any additional computational load.

6 Extensions

Weighted forests. In Breiman's (2001) forests, the final prediction is the average of the individual tree outcomes. A natural way to improve the method is to incorporate tree-level weights to emphasize more accurate trees in prediction (Winham et al., 2013). A closely related idea, proposed by Bernard et al. (2012), is to guide tree building—via resampling of the training set and other *ad hoc* randomization procedures—so that each tree will complement as much as possible the existing trees in the ensemble. The resulting *Dynamic Random Forest* (DRF) shows significant improvement in terms of accuracy on 20 real-based data sets compared to the standard, static, algorithm.

Online forests. In its original version, random forests is an *offline algorithm*, which is given the whole data set from the beginning and required to output an answer. In contrast, *online algorithms* do not require that the

entire training set is accessible at once. These models are appropriate for streaming settings, where training data is generated over time and must be incorporated into the model as quickly as possible. Random forests have been extended to the online framework in several ways (Saffari et al., 2009; Denil et al., 2013; Lakshminarayanan et al., 2014). In Lakshminarayanan et al. (2014), so-called *Mondrian forests* are grown in an online fashion and achieve competitive predictive performance comparable with other online random forests while being faster. When building online forests, a major difficulty is to decide when the amount of data is sufficient to cut a cell. Exploring this idea, Yi et al. (2012) propose *Information Forests*, whose construction consists in deferring classification until a measure of *classification confidence* is sufficiently high, and in fact break down the data so as to maximize this measure. An interesting theory related to these greedy trees can be found in Biau and Devroye (2013).

Survival forests. Survival analysis attempts to deal with incomplete data, and particularly right-censored data in fields such as clinical trials. In this context, parametric approaches such as proportional hazards are commonly used, but fail to model nonlinear effects. Random forests have been extended to the survival context by Ishwaran et al. (2008), who prove consistency of *Random Survival Forests* (RSF) algorithm assuming that all variables are factors. Yang et al. (2010) showed that by incorporating kernel functions into RSF, their algorithm KIRSF achieves better results in many situations. Ishwaran et al. (2011) review the use of the *minimal depth*, which measures the predictive quality of variables in survival trees.

Ranking forests. Clémençon et al. (2013) have extended random forests to deal with ranking problems and propose an algorithm called *Ranking Forests* based on the ranking trees of Clémençon and Vayatis (2009). Their approach is based on nonparametric scoring and ROC curve optimization in the sense of the AUC criterion.

Clustering forests. Yan et al. (2013) present a new clustering ensemble method called *Cluster Forests* (CF) in the context of unsupervised classification. CF randomly probes a high-dimensional data cloud to obtain good local clusterings, then aggregates via spectral clustering to obtain cluster assignments for the whole data set. The search for good local clusterings is guided by a cluster quality measure, and CF progressively improves each local clustering in a fashion that resembles tree growth in random forests.

Quantile forests. Meinshausen (2006) shows that random forests provide information about the full conditional distribution of the response variable, and thus can be used for quantile estimation.

Missing data. One of the strengths of random forests is that they can handle missing data. The procedure, explained in Breiman (2003), takes advantage of the so-called *proximity matrix*, which measures the proximity between pairs of observations in the forest, to estimate missing values. This measure is the empirical counterpart of the kernels defined in Section 3.2. Data imputation based on random forests has further been explored by Rieger et al. (2010), Crookston and Finley (2008), and extended to unsupervised classification by Ishioka (2013).

Forests and machine learning. One-class classification is a binary classification task for which only one class of samples is available for learning. Désir et al. (2013) study the *One Class Random Forests* algorithm, which is designed to solve this particular problem. Geremia et al. (2013) have introduced a supervised learning algorithm called *Spatially Adaptive Random Forests* to deal with semantic image segmentation applied to medical imaging protocols. Lastly, in the context of multi-label classification, Joly et al. (2014) adapt the idea of random projections applied to the output space to enhance tree-based ensemble methods by improving accuracy while significantly reducing the computational burden.

References

D. Amaratunga, J. Cabrera, and Y.-S. Lee. Enriched random forests. *Bioinformatics*, 24:2010–2014, 2008.

Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588, 1997.

K.J. Archer and R.V. Kimes. Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis*, 52:2249–2260, 2008.

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

L. Auret and C. Aldrich. Empirical comparison of tree ensemble variable importance measures. *Chemometrics and Intelligent Laboratory Systems*, 105:157–170, 2011.

Z.-H. Bai, L. Devroye, H.-K. Hwang, and T.-H. Tsai. Maxima in hypercubes. *Random Structures & Algorithms*, 27:290–309, 2005.

M. Banerjee and I.W. McKeague. Confidence sets for split points in decision trees. *The Annals of Statistics*, 35:543–574, 2007.

O. Barndorff-Nielsen and M. Sobel. On the distribution of the number of admissible points in a vector random sample. *Theory of Probability and Its Applications*, 11:249–269, 1966.

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.

S. Bernard, S. Adam, and L. Heutte. Dynamic random forests. *Pattern Recognition Letters*, 33:1580–1586, 2012.

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

G. Biau and L. Devroye. On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. *Journal of Multivariate Analysis*, 101:2499–2518, 2010.

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

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

G. Biau, F. Cérou, and A. Guyader. On the rate of convergence of the bagged nearest neighbor estimate. *Journal of Machine Learning Research*, 11:687–712, 2010.

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.

L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

L. Breiman. *Some infinity theory for predictor ensembles*. Technical Report 577, University of California, Berkeley, 2000a.

L. Breiman. Randomizing outputs to increase prediction accuracy. *Machine Learning*, 40:229–242, 2000b.

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

L. Breiman. *Setting up, using, and understanding random forests V4.0*. https://www.stat.berkeley.edu/~breiman/Using_random_forests_v4.0.pdf, 2003.

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

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

P. Bühlmann and B. Yu. Analyzing bagging. *The Annals of Statistics*, 30: 927–961, 2002.

S. Clémençon, M. Depecker, and N. Vayatis. Ranking forests. *Journal of Machine Learning Research*, 14:39–73, 2013.

S. Clémençon and N. Vayatis. Tree-based ranking methods. *IEEE Transactions on Information Theory*, 55:4316–4336, 2009.

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.

N.L. Crookston and A.O. Finley. yaImpute: An `R` package for kNN imputation. *Journal of Statistical Software*, 23:1–16, 2008.

A. Cutler and G. Zhao. PERT – perfect random tree ensembles. *Computing Science and Statistics*, 33:490–497, 2001.

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.

A. Davies and Z. Ghahramani. The random forest kernel and other kernels for big data from random partitions. *arXiv:1402.4293*, 2014.

H. Deng and G. Runger. Feature selection via regularized trees. In *The 2012 International Joint Conference on Neural Networks*, pages 1–8, 2012.

H. Deng and G. Runger. Gene selection with guided regularized random forest. *Pattern Recognition*, 46:3483–3489, 2013.

M. Denil, D. Matheson, and N. de Freitas. Consistency of online random forests. *arXiv:1302.4853*, 2013.

C. Désir, S. Bernard, C. Petitjean, and L. Heutte. One class random forests. *Pattern Recognition*, 46:3490–3506, 2013.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, New York, 1996.

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

T.G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15. Springer, 2000.

B. Efron. *The Jackknife, the Bootstrap and Other Resampling Plans*, volume 38. CBMS-NSF Regional Conference Series in Applied Mathematics, Philadelphia, 1982.

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

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

E. Geremia, B.H. Menze, and N. Ayache. Spatially adaptive random forests. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1332–1335, 2013.

P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63:3–42, 2006.

B. Gregorutti, B. Michel, and P. Saint Pierre. Correlation and variable importance in random forests. *arXiv:1310.5726*, 2013.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46:389–422, 2002.

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

T. Ho. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence*, 20, 832-844, 1998.

J. Howard and M. Bowles. The two most important algorithms in predictive modeling today. In *Strata Conference: Santa Clara*. http://strataconf.com/strata2012/public/schedule/detail/22658, 2012.

T. Ishioka. Imputation of missing values for unsupervised data using the proximity in random forests. In *eLmL 2013, The Fifth International Conference on Mobile, Hybrid, and On-line Learning*, pages 30–36. International Academy, Research, and Industry Association, 2013.

H. Ishwaran. Variable importance in binary regression trees and forests. *Electronic Journal of Statistics*, 1:519–537, 2007.

H. Ishwaran. The effect of splitting on random forests. *Machine Learning*, 99:75–118, 2013.

H. Ishwaran and U.B. Kogalur. Consistency of random survival forests. *Statistics & Probability Letters*, 80:1056–1064, 2010.

H. Ishwaran, U.B. Kogalur, E.H. Blackstone, and M.S. Lauer. Random survival forests. *The Annals of Applied Statistics*, 2:841–860, 2008.

H. Ishwaran, U.B. Kogalur, X. Chen, and A.J. Minn. Random survival forests for high-dimensional data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 4:115–132, 2011.

D. Jeffrey and G. Sanja. Simplified data processing on large clusters. *Communications of the ACM*, 51:107–113, 2008.

A. Joly, P. Geurts, and L. Wehenkel. Random forests with random projections of the output space for high dimensional multi-label classification. In T. Calders, F. Esposito, E. Hüllermeier, and R. Meo, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 607–622, Berlin, 2014. Springer.

A. Kleiner, A. Talwalkar, P. Sarkar, and M.I. Jordan. A scalable bootstrap for massive data. *arXiv:1112.5016*, 2012.

E. Konukoglu and M. Ganz. Approximate false positive rate control in selection frequency for random forest. *arXiv:1410.2838*, 2014.

A. Kyrillidis and A. Zouzias. Non-uniform feature sampling for decision tree ensembles. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4548–4552, 2014.

B. Lakshminarayanan, D.M. Roy, and Y.W. Teh. Mondrian forests: Efficient online random forests. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, pages 3140–3148, 2014.

P. Latinne, O. Debeir, and C. Decaestecker. Limiting the number of trees in random forests. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems*, pages 178–187, Berlin, 2001. Springer.

A. Liaw and M. Wiener. Classification and regression by randomForest. *R news*, 2:18–22, 2002.

Y. Lin and Y. Jeon. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101:578–590, 2006.

G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts. Understanding variable importances in forests of randomized trees. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, pages 431–439, 2013.

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

N. Meinshausen. Forest Garrote. *Electronic Journal of Statistics*, 3:1288–1304, 2009.

L. Mentch and G. Hooker. A novel test for additivity in supervised ensemble learners. *arXiv:1406.1845*, 2014a.

L. Mentch and G. Hooker. Ensemble trees and CLTs: Statistical inference for supervised learning. *arXiv:1404.6473*, 2014b.

E.A. Nadaraya. On estimating regression. *Theory of Probability and Its Applications*, 9:141–142, 1964.

K.K. Nicodemus and J.D. Malley. Predictor correlation impacts machine learning algorithms: Implications for genomic studies. *Bioinformatics*, 25: 1884–1890, 2009.

D.N. Politis, J.P. Romano, and M. Wolf. *Subsampling*. Springer, New York, 1999.

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.

S.S. Qian, R.S. King, and C.J. Richardson. Two statistical methods for the detection of environmental thresholds. *Ecological Modelling*, 166:87–97, 2003.

A. Rieger, T. Hothorn, and C. Strobl. *Random forests with missing values in the covariates*. Technical Report 79, University of Munich, Munich, 2010.

A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. On-line random forests. In *IEEE 12th International Conference on Computer Vision Workshops*, pages 1393–1400, 2009.

E. Scornet. On the asymptotics of random forests. *arXiv:1409.2090*, 2014.

E. Scornet. Random forests and kernel methods. *arXiv:1502.03836*, 2015.

E. Scornet, G. Biau, and J.-P. Vert. Consistency of random forests. *The Annals of Statistics, in press*, 2015.

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.

C.J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5:595–645, 1977.

C.J. Stone. Optimal rates of convergence for nonparametric estimators. *The Annals of Statistics*, 8:1348–1360, 1980.

C.J. Stone. Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics*, 10:1040–1053, 1982.

C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis. Conditional variable importance for random forests. *BMC Bioinformatics*, 9: 307, 2008.

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.

L. Tolo̧si and T. Lengauer. Classification with correlated features: Unreliability of feature ranking and solutions. *Bioinformatics*, 27:1986–1994, 2011.

A.K.Y. Truong. *Fast Growing and Interpretable Oblique Trees via Logistic Regression Models*. PhD thesis, University of Oxford, 2009.

H. Varian. Big data: New tricks for econometrics. *Journal of Economic Perspectives*, 28:3–28, 2014.

S. Wager. Asymptotic theory for random forests. *arXiv:1405.0352*, 2014.

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

G.S. Watson. Smooth regression analysis. *Sankhyā Series A*, pages 359–372, 1964.

S.J. Winham, R.R. Freimuth, and J.M. Biernacka. A weighted random forests approach to improve predictive performance. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 6:496–505, 2013.

D. Yan, A. Chen, and M.I. Jordan. Cluster forests. *Computational Statistics & Data Analysis*, 66:178–192, 2013.

F. Yang, J. Wang, and G. Fan. Kernel induced random survival forests. *arXiv:1008.3952*, 2010.

Z. Yi, S. Soatto, M. Dewan, and Y. Zhan. Information forests. In *Information Theory and Applications Workshop*, pages 143–146, 2012.

R. Zhu, D. Zeng, and M.R. Kosorok. *Reinforcement learning trees*. Technical Report, University of North Carolina, Chapel Hill, 2012.