TEST
DOI 10.1007/s11749-016-0481-7

INVITED PAPER

A random forest guided tour

Gérard Biau[1,2] · Erwan Scornet[1]

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.

Keywords Random forests · Randomization · Resampling · Parameter tuning · Variable importance

This invited paper is discussed in comments available at: doi:10.1007/s11749-016-0482-6; doi:10.1007/s11749-016-0483-5; doi:10.1007/s11749-016-0484-4; doi:10.1007/s11749-016-0485-3; doi:10.1007/s11749-016-0487-1.

✉ Gérard Biau
gerard.biau@upmc.fr

Erwan Scornet
erwan.scornet@upmc.fr

[1] Sorbonne Universités, UPMC Univ Paris 06, CNRS, Laboratoire de Statistique Théorique et Appliquées (LSTA), boîte 158, 4 place Jussieu, 75005 Paris, France

[2] Institut universitaire de France, Paris, France

Published online: 19 April 2016

Mathematics Subject Classification 62G02

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 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 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 web site (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. Howard (Kaggle) and 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 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. 2015), 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 consistency of a particular online forest, Wager (2014) and Mentch and Hooker (2015), who study the asymptotic distribution of forests, and Scornet et al. (2015), who show that Breiman's (2001) forests are consistent in an additive regression framework.

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. 2014; Wager et al. 2014). As for the CART-split criterion, it originates from the influential CART program 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 small number of observations, typically between 1 and 5.

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 Sect. 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 connected variable selection procedures. Section 6 discusses various extensions to random forests including online learning, survival analysis and clustering problems. A short discussion follows in Sect. 7.

2 The random forest estimate

2.1 Basic principles

Let us start with a word of caution. The term “random forests” is a bit ambiguous. For some authors, it is but a generic expression for aggregating random decision trees, no matter how the trees are obtained. For others, it refers to Breiman’s (2001) original algorithm. We essentially adopt the second point of view in the present survey.

As mentioned above, the 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 objective 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 $\mathbf{X} \in \mathcal{X} \subset \mathbb{R}^p$ 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 as the independent prototype pair $(\mathbf{X}, Y)$. The goal is to use the data set $\mathcal{D}_n$ to construct an estimate $m_n : \mathcal{X} \to \mathbb{R}$ 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 jth 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. In mathematical terms, the jth tree estimate takes the form

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

where $\mathcal{D}_n^\star(\Theta_j)$ is the set of data points selected prior to the tree construction, $A_n(\mathbf{x}; \Theta_j, \mathcal{D}_n)$ is the cell containing $\mathbf{x}$, and $N_n(\mathbf{x}; \Theta_j, \mathcal{D}_n)$ is the number of (preselected) points that fall into $A_n(\mathbf{x}; \Theta_j, \mathcal{D}_n)$.

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 tend 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 2015a, for more information on this limit calculation). In the following, to lighten notation we will simply write $m_{\infty,n}(\mathbf{x})$ instead of $m_{\infty,n}(\mathbf{x}; \mathcal{D}_n)$.

2.2 Algorithm

We now provide some insight into 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. The root of the tree is $\mathcal{X}$ itself and, at each step of the tree construction, a node (or equivalently its corresponding cell) is split into two parts. The terminal nodes (or leaves), taken together, form a partition of $\mathcal{X}$.

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 (or without) replacement from the original data set. These—and only these—a_n observations (with possible repetitions) are taken into account in the tree building. Then, at each cell of each tree, a split is performed by maximizing the CART-criterion (see below) over `mtry` directions chosen uniformly at random among the p original ones. (The resulting subset of selected coordinates is called $\mathcal{M}_{\text{try}}$.) Lastly, construction of individual trees is stopped when each cell contains less than `nodesize` points. For any query point $\mathbf{x} \in \mathcal{X}$, each regression tree predicts the average of the Y_i (that were among the a_n points) for which the corresponding $\mathbf{X}_i$ falls into the cell of $\mathbf{x}$. We draw attention to the fact that growing the tree and making the final estimation only depends on the a_n preselected data points. 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. $a_n \in \{1, \ldots, n\}$: the number of sampled data points in each tree;
2. `mtry` $\in \{1, \ldots, p\}$: the number of possible directions for splitting at each node of each tree;
3. `nodesize` $\in \{1, \ldots, a_n\}$: the number of examples in each cell below which the cell is not split.

By default, in the regression mode of the `R` package `randomForest`, the parameter `mtry` is set to $\lceil p/3 \rceil$ ($\lceil \cdot \rceil$ is the ceiling function), 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.

Algorithm 1: Breiman's random forest predicted value at **x**.

```
Input: Training set $\mathcal{D}_n$, number of trees $M > 0$, $a_n \in \{1, \ldots, n\}$, mtry $\in \{1, \ldots, p\}$,
       nodesize $\in \{1, \ldots, a_n\}$, and $\mathbf{x} \in \mathcal{X}$.
Output: Prediction of the random forest at $\mathbf{x}$.
for $j = 1, \ldots, M$ do
    Select $a_n$ points, with (or without) replacement, uniformly in $\mathcal{D}_n$. In the following steps, only
    these $a_n$ observations are used.
    Set $\mathcal{P} = (\mathcal{X})$ the list containing the cell associated with the root of the tree.
    Set $\mathcal{P}_{\text{final}} = \emptyset$ an empty list.
    while $\mathcal{P} \neq \emptyset$ do
        Let $A$ be the first element of $\mathcal{P}$.
        if $A$ contains less than nodesize points or if all $\mathbf{X}_i \in A$ are equal then
            Remove the cell $A$ from the list $\mathcal{P}$.
            $\mathcal{P}_{\text{final}} \leftarrow Concatenate(\mathcal{P}_{\text{final}}, A)$.
        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.
            Remove the cell $A$ from the list $\mathcal{P}$.
            $\mathcal{P} \leftarrow Concatenate(\mathcal{P}, A_L, A_R)$.
        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 $\mathcal{P}_{\text{final}}$.
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).
```

We still have to describe how the CART-split criterion operates. As for now, we consider for the ease of understanding a tree with no subsampling, which uses the entire and original data set $\mathcal{D}_n$ for its construction. In addition, 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 $\{1, \ldots, p\}$ and z the position of the cut along the jth coordinate, within the limits of A. Let $\mathcal{C}_A$ be the set of all such possible cuts in A. Then, with the notation $\mathbf{X}_i = (\mathbf{X}_i^{(1)}, \ldots, \mathbf{X}_i^{(p)})$, for any $(j, z) \in \mathcal{C}_A$, the CART-split criterion takes the form

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

where $A_L = \{\mathbf{x} \in A : \mathbf{x}^{(j)} < z\}$, $A_R = \{\mathbf{x} \in A : \mathbf{x}^{(j)} \geq z\}$, and $\bar{Y}_A$ (resp., $\bar{Y}_{A_L}$, $\bar{Y}_{A_R}$) is the average of the Y_i such that $\mathbf{X}_i$ belongs to A (resp., A_L, A_R), with the convention that the average is equal to 0 when no point $\mathbf{X}_i$ belongs to A (resp., A_L, A_R). For each cell A, the best cut $(j_n^\star, z_n^\star)$ is selected by maximizing $L_{\text{reg},n}(j, z)$ over $\mathcal{M}_{\text{try}}$ and $\mathcal{C}_A$; that is,

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

(To remove some of the ties in the argmax, the best cut is always performed in the middle of two consecutive data points.) Let us finally notice that the above optimization program extends effortlessly to the resampling case, by optimizing over the a_n preselected observations instead of the original data set $\mathcal{D}_n$.

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 along the directions in $\mathcal{M}_{\text{try}}$, and returns the best one. The quality measure (2) is the criterion used in the CART algorithm of Breiman et al. (1984). This criterion computes the (renormalized) difference between the empirical variance in the node before and after a cut is performed. There are three essential differences between CART and a tree of Breiman's (2001) forest. First of all, in Breiman's forests, the criterion (2) is evaluated over a subset $\mathcal{M}_{\text{try}}$ of randomly selected coordinates, and not over the whole range $\{1, \ldots, p\}$. Besides, the individual trees are not pruned, and the final cells do not contain more than `nodesize` observations (unless all data points in the cell have the same $\mathbf{X}_i$). At last, each tree is constructed on a subset of a_n examples picked within the initial sample, not on the whole sample $\mathcal{D}_n$; and only these a_n observations are used to calculate the estimation. When $a_n = n$ (and the resampling is done with replacement), the algorithm runs in bootstrap mode, whereas $a_n < n$ corresponds to subsampling (with or without replacement).

2.3 Supervised classification

For simplicity, we only consider here the binary classification problem, keeping in mind that random forests are intrinsically capable of dealing with multi-class problems (see, e.g., Díaz-Uriarte and de Andrés 2006). In this setting (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 probability of error

$$L(m_n) = \mathbb{P}[m_n(\mathbf{X}) \neq Y] \underset{n \to \infty}{\to} L^\star,$$

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

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

If a leaf represents region A, then a randomized tree classifier takes the simple form

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

where $\mathcal{D}_n^\star(\Theta_j)$ contains the data points selected in the resampling step, 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 two-class classification without modifying the CART-split criterion. To see this, take $Y \in \{0, 1\}$ and consider a single tree with no subsampling step. For any generic cell A, let $p_{0,n}(A)$ (resp., $p_{1,n}(A)$) be the empirical probability, given a data point in a cell A, that it has label 0 (resp., label 1). By noticing that $\bar{Y}_A = p_{1,n}(A) = 1 - p_{0,n}(A)$, the classification CART-split criterion reads, for any $(j, z) \in \mathcal{C}_A$,

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

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 the following simple interpretation. To classify a data point that falls in cell A, one uses the rule that assigns a point, uniformly selected from $\{\mathbf{X}_i \in A : (\mathbf{X}_i, Y_i) \in \mathcal{D}_n\}$, 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 error under this rule is the Gini index $2p_{0,n}(A)p_{1,n}(A)$. Note, however, that the prediction strategy is different in classification and regression: in the classification regime, each tree uses a local majority vote, whereas in regression the prediction is achieved by a local averaging.

When dealing with classification problems, it is usually recommended to set $\texttt{nodesize} = 1$ and $\texttt{mtry} = \sqrt{p}$ (see, e.g., Liaw and Wiener 2002).

We draw attention to the fact that regression estimation may also have an interest in the context of dichotomous and multicategory outcome variables (in this case, it is often termed *probability estimation*). For example, estimating outcome probabilities for individuals is important in many areas of medicine, with applications to surgery, oncology, internal medicine, pathology, pediatrics, and human genetics. We refer the interested reader to Malley et al. (2012) and to the survey papers by Kruppa et al. (2014a, b).

2.4 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). According to Schwarz et al. (2010), tuning the forest parameters may result in a computational burden, in particular for big data sets, with hundreds and thousands of samples and variables. To circumvent this issue, Schwarz et al. (2010) implement a fast version of the original algorithm, which they name *Random Jungle*.

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. Interestingly, picking a large M does not lead to overfitting. In effect, following an argument of Breiman (2001), we have

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

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 to obtain a prediction accuracy level similar to that of 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. A simple algorithm to tune the parameter `nodesize` in the classification setting is discussed in Kruppa et al. (2013).

The effect of `mtry` is 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).

Let us finally notice that even if there is no theoretical guarantee to support the default values of the parameters, they are nevertheless easy to tune without requiring an independent validation set. Indeed, the procedure accuracy 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 the observations are left out of the bootstrap sample and not used in the construction of the jth tree. In this way, for each tree, a test set—disjoint from the training set—is obtained, and averaging over all these left-out data points and over all trees is known as the *out-of-bag* error estimate. Thus, the out-of-bag error, computed on the observations set aside by the resampling prior to the tree building, offers a simple way to adjust the parameters without the need of a validation set. (e.g., Kruppa et al. 2013).

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.

As observed by Denil et al. (2014), 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 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 in which partitions do not depend on the training set $\mathcal{D}_n$. This family of simplified models is often called *purely random forests*, for which $\mathcal{X} = [0, 1]^d$. A widespread example is the *centered forest*, whose principle is as follows: (i) there is no resampling 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 final estimation at the query point $\mathbf{x}$ is achieved by averaging the Y_i corresponding to the $\mathbf{X}_i$ in the cell of $\mathbf{x}$. The parameter k acts as a smoothing parameter that controls the size of the terminal cells (see Fig. 1 for an example in two dimensions). It should be chosen large enough 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.

The centered forest rule was first formally analyzed by Breiman (2004), and then later by Biau et al. (2008) and Scornet (2015a), 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 $\mathcal{X} = [0, 1]^p$, 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

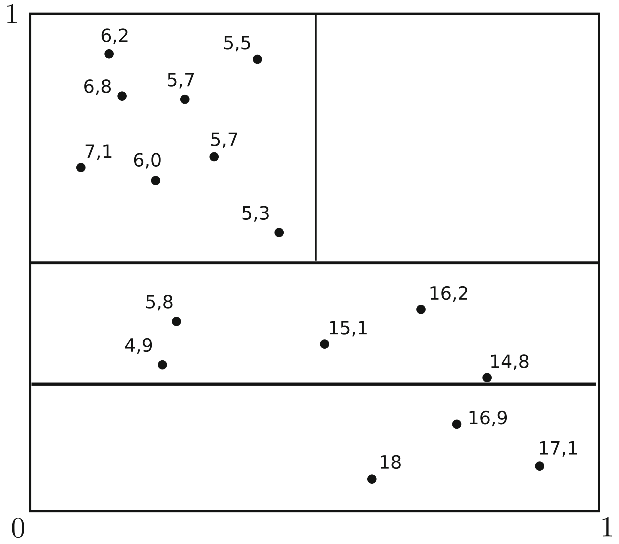

Fig. 1 A centered tree at level 2

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 approach, the covariates $X^{(j)}$ are independent and the target regression function $m(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$, 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

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

The variables of the remaining set $\{1, \ldots, p\}\backslash\mathcal{S}$ have no influence on the function m and can be safely removed. 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 arguments 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. In this dimension-reduction scenario, Breiman (2004) and Biau (2012) proved that if the probability $p_{j,n}$ of splitting along the jth direction tends to $1/S$ and m satisfies a Lipschitz-type smoothness condition, 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 $m_{\infty,n}$ to m depends only on the number $|\mathcal{S}|$ of strong variables, not on the dimension p. This rate is strictly faster than the usual rate $n^{-2/(p+2)}$ as soon as $|\mathcal{S}| \leq \lfloor 0.54p \rfloor$ ($\lfloor\cdot\rfloor$ is the floor function). In effect, the intrinsic dimension of the regression problem is $|\mathcal{S}|$, not p, and we see that the random forest estimate adapts itself to the sparse framework. Of course, this is achieved by assuming that the procedure succeeds in selecting the informative variables for splitting, which is indeed a strong assumption.

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. (Note that as such, the PURF can only be defined for $p = 1$.) 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 $\mathbf{X}_1, \ldots, \mathbf{X}_n$. In random geometry, an 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 Fig. 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 that ignore the resampling step. Indeed, if exactly one point is left in the leaves and if there is no resampling, then no matter what splitting strategy is used, the forest estimate at $\mathbf{x}$ is a weighted average of the Y_i whose corresponding $\mathbf{X}_i$ are LNN of $\mathbf{x}$. In other words,

$$m_{\infty,n}(\mathbf{x}) = \sum_{i=1}^{n} W_{ni}(\mathbf{x}) Y_i, \qquad (3)$$

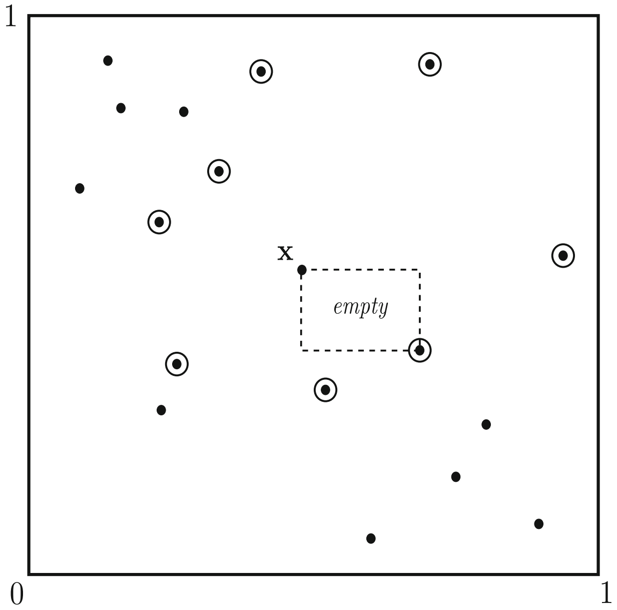

Fig. 2 The layered nearest neighbors (LNN) of a point **x** in dimension $p = 2$

where the weights $(W_{n1}, \dots, W_{nn})$ 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 $\sum_{i=1}^{n} W_{ni} = 1$. This important connection was first pointed out by Lin and Jeon (2006), who proved that if $(\mathbf{X})$ has a density on $([0, 1]^p)$ then, providing tree growing is independent of $Y_1, \dots, Y_n$ (such simplified models are sometimes called *non-adaptive*), we have

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

where $C > 0$ is a constant and $(n_{\max})$ is the maximal number of points in the terminal cells. Unfortunately, the exact values of the weight vector $(W_{n1}, \dots, W_{nn})$ 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 Eq. (3) 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 and without resampling, we have

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

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)

$$m_{\infty,n}(\mathbf{x}) \approx \frac{\sum_{i=1}^n Y_i K_n(\mathbf{X}_i, \mathbf{x})}{\sum_{j=1}^n K_n(\mathbf{X}_j, \mathbf{x})}, \tag{4}$$

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 the CART-splitting strategy 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 translation-invariant 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 (4) is, in general, more complicated, depending of the type of forest under investigation. For example, Scornet (2015b) proved that for a centered forest defined on $[0, 1]^p$ with parameter k, we have

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

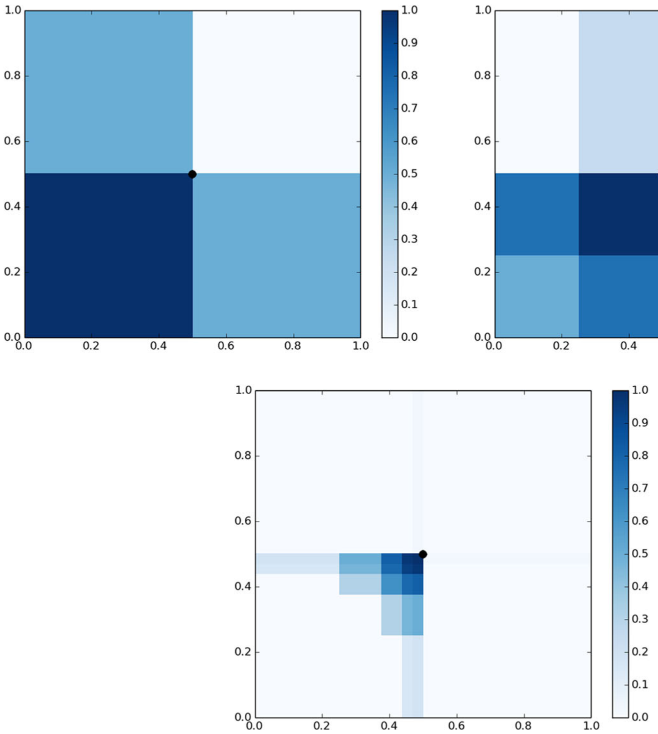

Fig. 3 Representations of f_1, f_2 and f_5 in $[0, 1]^2$

As an illustration, Fig. 3 shows the graphical representation for $k = 1, 2$ and 5 of the function f_k defined by

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

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) plug a specific (random forest-based) kernel—seen as a prior distribution over the piecewise constant functions—into a standard Gaussian process algorithm, and empirically demonstrate that it outperforms the same algorithm ran with linear and radial basis kernels. Besides, forest-based kernels 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 entire 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 n points with replacement to compute the individual tree estimates. 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 distribution of the bootstrap sample $\mathcal{D}_n^\star$ is different from that of the original one $\mathcal{D}_n$, 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 is selected more than once. Therefore, with positive probability, there exist two identical data points in $\mathcal{D}_n^\star$, and the distribution of $\mathcal{D}_n^\star$ cannot be absolutely continuous.

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 2015; 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 (2015a) provides some insight into the role and importance of subsampling.

A median forest resembles a centered forest. Once the splitting direction is chosen, the cut is performed at the empirical median of the $\mathbf{X}_i$ in the cell. In addition, the construction does not stop at level k but continues until each cell contains exactly one observation. Since the number of cases left in the leaves does not grow with n, each tree of a median forest is in general inconsistent (see Györfi et al. 2002, Problem 4.3). However, Scornet (2015a) shows that if $a_n/n \to 0$, then the median forest is consistent, despite the fact that the individual trees are not. The assumption $a_n/n \to 0$ guarantees that every single observation pair $(\mathbf{X}_i, Y_i)$ is used in the jth 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 subbagging, one may turn the 1-NN estimate into a consistent one, provided that the size of subsamples is sufficiently small. We proceed as follows, via a randomized basic regression estimate r_{a_n} in which $1 \leq a_n < n$ 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 subbagging, that is, we repeat the random subsampling an infinite number of times and take the average of the individual outcomes. Thus, the subbagged 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 error 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 local averaging estimate (Stone 1977) with weights

$$W_{ni}(\mathbf{x}) = \mathbb{P}[\mathbf{X}_i \text{ is the 1-NN of } \mathbf{x} \text{ in a random selection of size } a_n].$$

The connection between bagging and nearest neighbor estimation is further explored by Biau et al. (2010), who prove that the subbagged 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 estimate restricted to the left (resp., right) child of the cell takes the form

$$\beta_{\ell,n}^\star = \mathbb{E}[Y|X \leq d^\star] \quad \left(\text{resp., } \beta_{r,n}^\star = \mathbb{E}[Y|X > d^\star]\right).$$

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

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

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

$$n^{1/3} \begin{pmatrix} \hat{\beta}_{\ell,n} - \beta_\ell^\star \\ \hat{\beta}_{r,n} - \beta_r^\star \\ \hat{d}_n - d^\star \end{pmatrix} \overset{\mathcal{D}}{\to} \begin{pmatrix} c_1 \\ c_2 \\ 1 \end{pmatrix} \mathop{\arg\max}_t (aW(t) - bt^2), \tag{5}$$

where $\mathcal{D}$ denotes convergence in distribution, 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 $\beta_\ell^\star$, $\beta_r^\star$ and $d^\star$. The limiting distribution in (5) 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 (assuming that $\mathcal{X} = [0, 1]^d$), then the sample size of the two resulting cells is roughly reduced by a factor of two—this is an undesirable property, which may be harmful for the prediction task. Thus, 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.

It is shown in Scornet et al. (2015) that random forests asymptotically perform, with high probability, splits along the S informative variables (in the sense of Sect. 3.1). Denote by $j_{n,1}(\mathbf{X}), \ldots, j_{n,k}(\mathbf{X})$ the first k cut directions used to construct the cell of $\mathbf{X}$, with the convention that $j_{n,q}(\mathbf{X}) = \infty$ if the cell has been cut strictly less than q times. Assuming some regularity conditions on the regression model, and considering a modification of Breiman's forests in which all directions are preselected for splitting, Scornet et al. (2015) prove that, with probability $1-\xi$, for all n large enough and all $1 \le q \le k$,

$$j_{n,q}(\mathbf{X}) \in \{1, \ldots, S\}.$$

This result offers an interesting perspective on why random forests nicely adapt to the sparsity setting. Indeed, it shows that the algorithm selects splits mostly along the S informative variables, so that everything happens as if data were projected onto the vector space spanned by these variables.

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. As a variant of the original algorithm, Breiman (2001) considered splitting along linear combinations of features (this procedure has been implemented by Truong 2009, in the package `obliquetree` of the statistical computing environment `R`). As noticed by Menze et al. (2011), the feature space separation by orthogonal hyperplanes in random forests results in box-like decision surfaces, which may be advantageous for some data but suboptimal for other, particularly for collinear data with correlated features.

With respect to 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 classification tree, 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. (2015) 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. Splits in random forests are known to be biased toward covariates with many possible splits (Breiman et al. 1984; Segal 1988) or with missing values (Kim and Loh 2001). Hothorn et al. (2006) propose a two-step procedure to correct this situation by first selecting the splitting variable and then the position of the cut along the chosen variable. The predictive performance of the resulting trees is empirically shown to be as good as the performance of the exhaustive search procedure. We also refer the reader to Ziegler and König (2014), who review the different splitting strategies.

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 parsimonious 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.

4.3 Consistency, asymptotic normality, and more

All in all, little has been proven mathematically for the original procedure of Breiman. A seminal result by Breiman (2001) shows that the error of the forest is small as soon as the predictive power of each tree is good and the correlation between the tree errors is low. More precisely, independently of the type of forest, one has

$$\mathbb{E}_{\mathbf{X},Y}[Y - m_{\infty,n}(\mathbf{X})]^2 \leq \bar{\rho}\, \mathbb{E}_{\mathbf{X},Y,\Theta}[Y - m_n(\mathbf{X};\Theta)]^2,$$

where

$$\bar{\rho} = \frac{\mathbb{E}_{\Theta,\Theta'}[\rho(\Theta,\Theta')g(\Theta)g(\Theta')]}{\mathbb{E}_{\Theta}[g(\Theta)]^2},$$

with Θ and Θ' independent and identically distributed,

$$\rho(\Theta,\Theta') = \mathrm{Corr}_{\mathbf{X},Y}\big[Y - m_n(\mathbf{X};\Theta),\, Y - m_n(\mathbf{X};\Theta')\big],$$

and $g(\Theta) = \sqrt{\mathbb{E}_{\mathbf{X},Y}[Y - m_n(\mathbf{X};\Theta)]^2}$. Similarly, Friedman et al. (2009) decompose the variance of the forest as a product of the correlation between trees and the variance of a single tree. Thus, for all $\mathbf{x}$,

$$\mathrm{Var}[m_{\infty,n}(\mathbf{x})] = \rho(\mathbf{x})\sigma(\mathbf{x}),$$

where $\rho(\mathbf{x}) = \mathrm{Corr}[m_n(\mathbf{x};\Theta), m_n(\mathbf{x};\Theta')]$ and $\sigma(\mathbf{x}) = \mathrm{Var}[m_n(\mathbf{x};\Theta)]$.

A link between the error of the finite and infinite forests is established in Scornet (2015a), who shows, provided some regularity assumptions are satisfied, that

$$\begin{aligned}0 &\leq \mathbb{E}[m_{M,n}(\mathbf{X};\Theta_1,\ldots,\Theta_M) - m(\mathbf{X})]^2 - \mathbb{E}[m_{\infty,n}(\mathbf{X}) - m(\mathbf{X})]^2\\ &\leq \frac{8}{M} \times \big(\|m\|_\infty^2 + \sigma^2(1 + 4\log n)\big).\end{aligned}$$

This inequality provides an interesting solution for choosing the number of trees, by making the error of the finite forest arbitrary close to that of the infinite one.

Consistency and asymptotic normality of the whole algorithm were recently proved, replacing bootstrap by subsampling and simplifying the splitting step. So, Wager (2014) shows the asymptotic normality of Breiman's infinite forests, assuming that

(i) cuts are spread along all the p directions and do not separate a small fraction of the data set; and (ii) two different data set are used to, respectively, build the tree and estimate the value within a leaf. He also establishes that the infinitesimal jackknife (Efron 1979) consistently estimates the forest variance.

Mentch and Hooker (2015) prove a similar result for finite forests, which mainly relies on the assumption that the prediction of the forests does not much vary when the label of one point in the training set is slightly modified. These authors show that whenever M_n (the number of trees) is allowed to vary with n, and when $a_n = \mathrm{o}(\sqrt{n})$ and $\lim_{n\to\infty} n/M_n = 0$, then, for a fixed $\mathbf{x}$,

$$\frac{\sqrt{n}\Big(m_{M,n}(\mathbf{x}; \Theta_1, \ldots, \Theta_M) - \mathbb{E}[m_{M,n}(\mathbf{x}; \Theta_1, \ldots, \Theta_M)]\Big)}{\sqrt{a_n^2 \zeta_{1,a_n}}} \overset{\mathcal{D}}{\to} N,$$

where N is a standard normal random variable,

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

$\mathbf{X}'_i$ an independent copy of $\mathbf{X}_i$ and Θ' an independent copy of Θ. It is worth noting that both Mentch and Hooker (2015) and Wager et al. (2014) provide corrections for estimating the forest variance ζ_{1,a_n}.

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 Fig. 4 and let $m(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$. 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 $L^\star = 0$—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. We note, however, that if Breiman's (2001) original algorithm is used in this example (that is, each cell contains exactly one data point) then one obtains a consistent classification rule. We also note that the regression function m is not Lipschitz—a smoothness assumption on which many results on random forests rely.

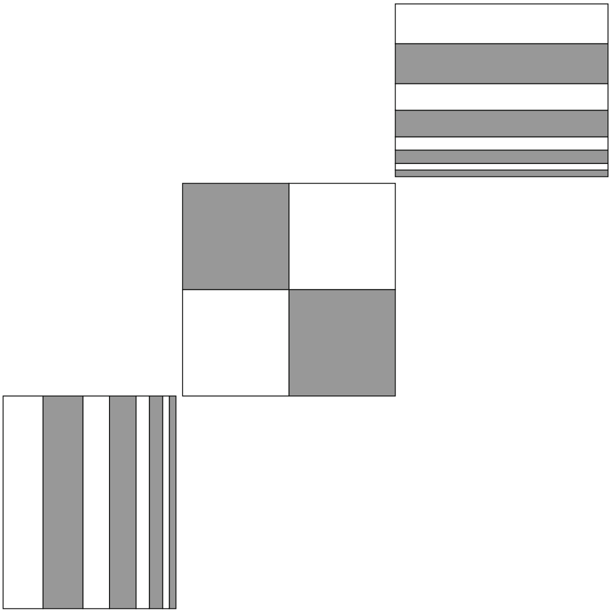

Fig. 4 An example of a distribution for which greedy random forests are inconsistent. The distribution of **X** is uniform on the union of the three large squares. *White* areas represent the set where $m(\mathbf{x}) = 0$ and *grey* where $m(\mathbf{x}) = 1$

5 Variable importance

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; see Breiman 2003a), 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), first defined by Breiman (2001), stems from the idea that if the variable is not important, then rearranging its values should not degrade prediction accuracy.

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

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

where $p_{n,t}$ 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 $(j_{n,t}^{\star}, z_{n,t}^{\star})$ 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 version $L_{\mathrm{class},n}$. 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 out-of-bag error estimate (see Sect. 2.4). To measure the importance of the jth feature, we randomly permute the values of variable $X^{(j)}$ in the out-of-bag observations and put these examples 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 data set of the ℓth 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,

$$\widehat{\mathrm{MDA}}(X^{(j)}) = \frac{1}{M}\sum_{\ell=1}^{M}\Big[R_n\big[m_n(\cdot;\Theta_\ell), \mathcal{D}^j_{\ell,n}\big] - R_n\big[m_n(\cdot;\Theta_\ell), \mathcal{D}_{\ell,n}\big]\Big], \qquad (6)$$

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

$$R_n\big[m_n(\cdot;\Theta_\ell), \mathcal{D}\big] = \frac{1}{|\mathcal{D}|}\sum_{i:(\mathbf{X}_i, Y_i)\in\mathcal{D}} (Y_i - m_n(\mathbf{X}_i;\Theta_\ell))^2. \qquad (7)$$

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

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

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 (6) and (7) since $Y_i \in \{0, 1\}$ (so, $R_n(m_n(\cdot;\Theta), \mathcal{D})$ is also the proportion 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 an infinite ensemble of totally randomized and fully developed trees. At each cell, the ℓth 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 $\mathrm{MDI}(X^{(j)})$ computed with the whole forest satisfies

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

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,

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

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 with respect to V if and only if $\mathrm{MDI}^{\star}(X^{(j)}) = 0$. It is easy to see that if an additional irrelevant variable $X^{(p+1)}$ is added to the list of variables, then, for any j, the variable importance $\mathrm{MDI}^{\star}(X^{(j)})$ 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 alter the importances of the other variables in an infinite sample setting.

The most notable results regarding $\mathrm{MDA}^{\star}$ are due to Ishwaran (2007), who studies a slight modification of the criterion replacing permutation by 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 variable importance $\mathrm{MDA}^{\star}(X^{(j)})$ 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 $\mathrm{MDA}^{\star}(X^{(j)})$ 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. (2016), who compute 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 $C = [\mathrm{Cov}(X^{(j)}, X^{(k)})]_{1 \le j,k \le p} = (1-c)I_p + c\mathbb{1}\mathbb{1}^{\top}$ [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,

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

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 explored 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 Toloşi 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 Toloşi 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. (2016). Detecting relevant features can also be achieved via hypothesis testing (Mentch and Hooker 2015)—a principle that may be used to detect more complex structures of the regression function, like for instance its additivity (Mentch and Hooke 2014).

6 Extensions

6.1 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.

6.2 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 are 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).

6.3 Survival forests

Survival analysis attempts to deal with analysis of time duration until one or more events happen. Most often, survival analysis is also concerned 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 categorical. 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.

6.4 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 relies on nonparametric scoring and ROC curve optimization in the sense of the AUC criterion.

6.5 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.

6.6 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.

6.7 Missing data

One of the strengths of random forests is that they can handle missing data. The procedure, explained in Breiman (2003b), 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 Sect. 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).

6.8 Single class data

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.

6.9 Unbalanced data set

Random forests can naturally be adapted to fit the unbalanced data framework by down-sampling the majority class and growing each tree on a more balanced data set (Chen et al. 2004; Kuhn and Johnson 2013). An interesting application in which unbalanced data sets are involved is by Fink et al. (2010), who explore the continent-wide inter-annual migrations of common North American birds. They use random forests for which each tree is trained and allowed to predict on a particular (random) region in space and time.

7 Conclusion and perspectives

The authors trust that this review paper has provided an overview of some of the recent literature on random forests and offered insights into how new and emerging fields are impacting the method. As statistical applications become increasingly sophisticated, massive and complex data sets require today the development of algorithms that ensure global competitiveness, achieving both computational efficiency and safe with high-dimension models and huge number of samples. It is our belief that forests and their basic principles ("divide and conquer", resampling, aggregation, random search of the feature space) offer simple but fundamental ideas that may leverage new state-of-the-art algorithms.

It remains, however, that the present results are insufficient to explain in full generality the remarkable behavior of random forests. The authors' intuition is that tree aggregation models are able to estimate patterns that are more complex than classical ones—patterns that cannot be simply characterized by standard sparsity or smoothness conditions. These patterns, which are beyond the reach of classical methods, are still to be discovered, quantified, and mathematically described.

It is sometimes alluded to that random forests have the flavor of deep network architectures (e.g., Bengio 2009), insofar as ensemble of trees allow to discriminate between a very large number of regions. Indeed, the identity of the leaf node with which a data point is associated for each tree forms a tuple that can represent a considerable quantity of possible patterns, because the total intersections of the leaf regions can be exponential in the number of trees. This point of view, could be one of the reasons for the success of forests on large-scale data. As a matter of fact, the connection between random forests and neural networks is largely unexamined (Welbl 2014).

Another critical issue is how to choose tuning parameters that are optimal in a certain sense, especially the size a_n of the preliminary resampling. By default, the algorithm runs in bootstrap mode (i.e., $a_n = n$ points selected with replacement) and although this seems to give excellent results, there is to date no theory to support this choice. Furthermore, although random forests are fully grown in most applications, the impact of tree depth on the statistical performance of the algorithm is still an open question.

Acknowledgments We thank the Editors and three anonymous referees for valuable comments and insightful suggestions.

References

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

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

Archer KJ, Kimes RV (2008) Empirical characterization of random forest variable importance measures. Comput Stat Data Anal 52:2249–2260

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

Auret L, Aldrich C (2011) Empirical comparison of tree ensemble variable importance measures. Chemom Intell Lab Syst 105:157–170

Bai Z-H, Devroye L, Hwang H-K, Tsai T-H (2005) Maxima in hypercubes. Random Struct Algorithms 27:290–309

Banerjee M, McKeague IW (2007) Confidence sets for split points in decision trees. Ann Stat 35:543–574

Barndorff-Nielsen O, Sobel M (1966) On the distribution of the number of admissible points in a vector random sample. Theory Probab Appl 11:249–269

Bengio Y (2009) Learning deep architectures for AI. Found Trends Mach Learn 2:1–127

Bernard S, Heutte L, Adam S (2008) Forest-RK: a new random forest induction method. In: Huang D-S, Wunsch DC II, Levine DS, Jo K-H (eds) Advanced intelligent computing theories and applications. With aspects of artificial intelligence. Springer, Berlin, pp 430–437

Bernard S, Adam S, Heutte L (2012) Dynamic random forests. Pattern Recognit Lett 33:1580–1586

Biau G (2012) Analysis of a random forests model. J Mach Learn Res 13:1063–1095

Biau G, Devroye L (2010) On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. J Multivar Anal 101:2499–2518

Biau G, Devroye L (2013) Cellular tree classifiers. Electron J Stat 7:1875–1912

Biau G, Devroye L, Lugosi G (2008) Consistency of random forests and other averaging classifiers. J Mach Learn Res 9:2015–2033

Biau G, Cérou F, Guyader A (2010) On the rate of convergence of the bagged nearest neighbor estimate. J Mach Learn Res 11:687–712

Boulesteix A-L, Janitza S, Kruppa J, König IR (2012) Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. Wiley Interdiscip Rev Data Mining Knowl Discov 2:493–507

Breiman L (1996) Bagging predictors. Mach Learn 24:123–140

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

Breiman L (2000b) Randomizing outputs to increase prediction accuracy. Mach Learn 40:229–242

Breiman L (2001) Random forests. Mach Learn 45:5–32

Breiman L (2003a) Setting up, using, and understanding random forests V3.1. https://www.stat.berkeley.edu/~breiman/Using_random_forests_V3.1.pdf

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

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

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Chapman & Hall/CRC, Boca Raton

Bühlmann P, Yu B (2002) Analyzing bagging. Ann Stat 30:927–961

Chen C, Liaw A, Breiman L (2004) Using random forest to learn imbalanced data. Technical Report 666, University of California, Berkeley

Clémençon S, Depecker M, Vayatis N (2013) Ranking forests. J Mach Learn Res 14:39–73

Clémençon S, Vayatis N (2009) Tree-based ranking methods. IEEE Trans Inform Theory 55:4316–4336

Criminisi A, Shotton J, Konukoglu E (2011) Decision forests: a unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Found Trends Comput Graph Vis 7:81–227

Crookston NL, Finley AO (2008) yaImpute: an R package for *k*NN imputation. J Stat Softw 23:1–16

Cutler A, Zhao G (2001) PERT—perfect random tree ensembles. Comput Sci Stat 33:490–497

Cutler DR, Edwards TC Jr, Beard KH, Cutler A, Hess KT, Gibson J, Lawler JJ (2007) Random forests for classification in ecology. Ecology 88:2783–2792

Davies A, Ghahramani Z (2014) The Random Forest Kernel and creating other kernels for big data from random partitions. arXiv:1402.4293

Deng H, Runger G (2012) Feature selection via regularized trees. In: The 2012 international joint conference on neural networks, pp 1–8

Deng H, Runger G (2013) Gene selection with guided regularized random forest. Pattern Recognit 46:3483–3489

Denil M, Matheson D, de Freitas N (2013) Consistency of online random forests. In: International conference on machine learning (ICML)

Denil M, Matheson D, de Freitas N (2014) Narrowing the gap: random forests in theory and in practice. In: International conference on machine learning (ICML)

Désir C, Bernard S, Petitjean C, Heutte L (2013) One class random forests. Pattern Recognit 46:3490–3506

Devroye L, Györfi L, Lugosi G (1996) A probabilistic theory of pattern recognition. Springer, New York

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

Dietterich TG (2000) Ensemble methods in machine learning. In: Kittler J, Roli F (eds) Multiple classifier systems. Springer, Berlin, pp 1–15

Efron B (1979) Bootstrap methods: another look at the jackknife. Ann Stat 7:1–26

Efron B (1982) The jackknife, the bootstrap and other resampling plans, vol 38. CBMS-NSF Regional Conference Series in Applied Mathematics, Philadelphia

Fink D, Hochachka WM, Zuckerberg B, Winkler DW, Shaby B, Munson MA, Hooker G, Riedewald M, Sheldon D, Kelling S (2010) Spatiotemporal exploratory models for broad-scale survey data. Ecol Appl 20:2131–2147

Friedman J, Hastie T, Tibshirani R (2009) The elements of statistical learning, 2nd edn. Springer, New York

Genuer R (2012) Variance reduction in purely random forests. J Nonparametr Stat 24:543–562

Genuer R, Poggi J-M, Tuleau-Malot C (2010) Variable selection using random forests. Pattern Recognit Lett 31:2225–2236

Geremia E, Menze BH, Ayache N (2013) Spatially adaptive random forests. In: IEEE international symposium on biomedical imaging: from nano to macro, pp 1332–1335

Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. Mach Learn 63:3–42

Gregorutti B, Michel B, Saint Pierre P (2016) Correlation and variable importance in random forests. Stat Comput. doi:10.1007/s11222-016-9646-1

Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46:389–422

Györfi L, Kohler M, Krzyżak A, Walk H (2002) A distribution-free theory of nonparametric regression. Springer, New York

Ho T (1998) The random subspace method for constructing decision forests. Pattern Anal Mach Intell 20:832–844

Hothorn T, Hornik K, Zeileis A (2006) Unbiased recursive partitioning: a conditional inference framework. J Comput Graph Stat 15:651–674

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

Ishioka T (2013) 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, pp 30–36. International Academy, Research, and Industry Association

Ishwaran H (2007) Variable importance in binary regression trees and forests. Electron J Stat 1:519–537

Ishwaran H (2013) The effect of splitting on random forests. Mach Learn 99:75–118

Ishwaran H, Kogalur UB (2010) Consistency of random survival forests. Stat Probab Lett 80:1056–1064

Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS (2008) Random survival forests. Ann Appl Stat 2:841–860

Ishwaran H, Kogalur UB, Chen X, Minn AJ (2011) Random survival forests for high-dimensional data. Stat Anal Data Mining ASA Data Sci J 4:115–132

Jeffrey D, Sanja G (2008) Simplified data processing on large clusters. Commun ACM 51:107–113

Joly A, Geurts P, Wehenkel L (2014) Random forests with random projections of the output space for high dimensional multi-label classification. In: Calders T, Esposito F, Hüllermeier E, Meo R (eds) Machine learning and knowledge discovery in databases. Springer, Berlin, pp 607–622

Kim H, Loh W-Y (2001) Classification trees with unbiased multiway splits. J Am Stat Assoc 96:589–604

Kleiner A, Talwalkar A, Sarkar P, Jordan MI (2014) A scalable bootstrap for massive data. J Royal Stat Soc Ser B (Stat Methodol) 76:795–816

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

Kruppa J, Schwarz A, Arminger G, Ziegler A (2013) Consumer credit risk: individual probability estimates using machine learning. Expert Syst Appl 40:5125–5131

Kruppa J, Liu Y, Biau G, Kohler M, König IR, Malley JD, Ziegler A (2014a) Probability estimation with machine learning methods for dichotomous and multicategory outcome: theory. Biometr J 56:534–563

Kruppa J, Liu Y, Diener H-C, Holste T, Weimar C, König IR, Ziegler A (2014b) Probability estimation with machine learning methods for dichotomous and multicategory outcome: applications. Biometr J 56:564–583

Kuhn M, Johnson K (2013) Applied predictive modeling. Springer, New York

Kyrillidis A, Zouzias A (2014) Non-uniform feature sampling for decision tree ensembles. In: IEEE international conference on acoustics, speech and signal processing, pp 4548–4552

Lakshminarayanan B, Roy DM, Teh YW (2014) Mondrian forests: efficient online random forests. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ (eds) Advances in neural information processing systems, pp 3140–3148

Latinne P, Debeir O, Decaestecker C (2001) Limiting the number of trees in random forests. In: Kittler J, Roli F (eds) Multiple classifier systems. Springer, Berlin, pp 178–187

Liaw A, Wiener M (2002) Classification and regression by randomForest. R News 2:18–22

Lin Y, Jeon Y (2006) Random forests and adaptive nearest neighbors. J Am Stat Assoc 101:578–590

Louppe G, Wehenkel L, Sutera A, Geurts P (2013) Understanding variable importances in forests of randomized trees. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ (eds) Advances in neural information processing systems, pp 431–439

Malley JD, Kruppa J, Dasgupta A, Malley KG, Ziegler A (2012) Probability machines: consistent probability estimation using nonparametric learning machines. Methods Inform Med 51:74–81

Meinshausen N (2006) Quantile regression forests. J Mach Learn Res 7:983–999

Meinshausen N (2009) Forest Garrote. Electron J Stat 3:1288–1304

Mentch L, Hooker G (2014) A novel test for additivity in supervised ensemble learners. arXiv:1406.1845

Mentch L, Hooker G (2015) Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. J Mach Learn Res (in press)

Menze BH, Kelm BM, Splitthoff DN, Koethe U, Hamprecht FA (2011) On oblique random forests. In: Gunopulos D, Hofmann T, Malerba D, Vazirgiannis M (eds) Machine learning and knowledge discovery in databases. Springer, Berlin, pp 453–469

Nadaraya EA (1964) On estimating regression. Theory Probab Appl 9:141–142

Nicodemus KK, Malley JD (2009) Predictor correlation impacts machine learning algorithms: Implications for genomic studies. Bioinformatics 25:1884–1890

Politis DN, Romano JP, Wolf M (1999) Subsampling. Springer, New York

Prasad AM, Iverson LR, Liaw A (2006) Newer classification and regression tree techniques: bagging and random forests for ecological prediction. Ecosystems 9:181–199

Qian SS, King RS, Richardson CJ (2003) Two statistical methods for the detection of environmental thresholds. Ecol Model 166:87–97

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

Saffari A, Leistner C, Santner J, Godec M, Bischof H (2009) On-line random forests. In: IEEE 12th international conference on computer vision workshops, pp 1393–1400

Schwarz DF, König IR, Ziegler A (2010) On safari to random jungle: a fast implementation of random forests for high-dimensional data. Bioinformatics 26:1752–1758

Scornet E (2015a) On the asymptotics of random forests. J Multivar Anal 146:72–83

Scornet E (2015b) Random forests and kernel methods. IEEE Trans Inform Theory 62:1485–1500

Scornet E, Biau G, Vert J-P (2015) Consistency of random forests. Ann Stat 43:1716–1741

Segal MR (1988) Regression trees for censored data. Biometrics 44:35–47

Shotton J, Fitzgibbon A, Cook M, Sharp T, Finocchio M, Moore R, Kipman A, Blake A (2011) Real-time human pose recognition in parts from single depth images. In: IEEE conference on computer vision and pattern recognition, pp 1297–1304

Stone CJ (1977) Consistent nonparametric regression. Ann Stat 5:595–645

Stone CJ (1980) Optimal rates of convergence for nonparametric estimators. Ann Stat 8:1348–1360

Stone CJ (1982) Optimal global rates of convergence for nonparametric regression. Ann Stat 10:1040–1053

Strobl C, Boulesteix A-L, Kneib T, Augustin T, Zeileis A (2008) Conditional variable importance for random forests. BMC Bioinform 9:307

Svetnik V, Liaw A, Tong C, Culberson JC, Sheridan RP, Feuston BP (2003) Random forest: a classification and regression tool for compound classification and QSAR modeling. J Chem Inform Comput Sci 43:1947–1958

Toloşi L, Lengauer T (2011) Classification with correlated features: unreliability of feature ranking and solutions. Bioinformatics 27:1986–1994

Truong AKY (2009) Fast growing and interpretable oblique trees via logistic regression models. PhD thesis, University of Oxford, Oxford

Varian H (2014) Big data: new tricks for econometrics. J Econ Perspect 28:3–28

Wager S (2014) Asymptotic theory for random forests. arXiv:1405.0352

Wager S, Hastie T, Efron B (2014) Confidence intervals for random forests: the jackknife and the infinitesimal jackknife. J Mach Learn Res 15:1625–1651

Watson GS (1964) Smooth regression analysis. Sankhyā Ser A 26:359–372

Welbl J (2014) Casting random forests as artificial neural networks and profiting from it. In: Jiang X, Hornegger J, Koch R (eds) Pattern recognition. Springer, Berlin, pp 765–771

Winham SJ, Freimuth RR, Biernacka JM (2013) A weighted random forests approach to improve predictive performance. Stat Anal Data Mining ASA Data Sci J 6:496–505

Yan D, Chen A, Jordan MI (2013) Cluster forests. Comput Stat Data Anal 66:178–192

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

Yi Z, Soatto S, Dewan M, Zhan Y (2012) Information forests. In: 2012 information theory and applications workshop, pp 143–146

Zhu R, Zeng D, Kosorok MR (2015) Reinforcement learning trees. J Am Stat Assoc 110(512):1770–1784

Ziegler A, König IR (2014) Mining data with random forests: current options for real-world applications. Wiley Interdiscip Rev Data Mining Knowl Discov 4:55–63