Chapter 1
Introduction to Random Forests with R

Abstract The two algorithms discussed in this book were proposed by Leo Breiman: CART trees, which were introduced in the mid-1980s, and random forests, which emerged just under 20 years later in the early 2000s. This chapter offers an introduction to the subject matter, beginning with a historical overview. Some notations, used to define the various statistical objectives addressed in the book, are also introduced: classification, regression, prediction, and variable selection. In turn, the three R packages used in the book are listed, and some competitors are mentioned. Lastly, the four datasets used to illustrate the methods' application are presented: the running example (spam), a genomic dataset, and two pollution datasets (ozone and dust).

1.1 Preamble

The two algorithms discussed in this book were proposed by Leo Breiman: CART (Classification And Regression Trees) trees, which were introduced in the mid-1980s (Breiman et al. 1984), and random forests (Breiman 2001), which emerged just under 20 years later in the early 2000s. At the confluence of statistics and statistical learning, this shortcut among Leo Breiman's multiple contributions, whose scientific biography is described in Olshen (2001) and Cutler (2010), provides a remarkable figure of these two disciplines.

Decision trees are the basic tool for numerous tree-based ensemble methods. Although known for decades and very attractive because of their simplicity and interpretability, their use suffered, until the 1980s, from serious justified objections. From this point of view, CART offers to decision trees the conceptual framework of automatic model selection, giving them theoretical guarantees and broad applicability while preserving their ease of interpretation.

But one of the major drawbacks, instability, remains. The idea of random forests is to exploit the natural variability of trees. More specifically, it is a matter of disrupting the construction by introducing some randomness in the selection of both individuals and variables. The resulting trees are then combined to construct the final prediction,

R. Genuer and J.-M. Poggi, *Random Forests with R*, Use R!,
https://doi.org/10.1007/978-3-030-56485-8_1

rather than choosing one of them. Several algorithms based on such principles have thus been developed, for many of them, by Breiman himself: Bagging (Breiman 1996), several variants of the Arcing (Breiman 1998), and Adaboost (Freund and Schapire 1997).

Random forests (RF in the following) are therefore a nonparametric method of statistical learning widely used in many fields of application, such as the study of microarrays (Díaz-Uriarte and Alvarez De Andres 2006), ecology (Prasad et al. 2006), pollution prediction (Ghattas 1999), and genomics (Goldstein et al. 2010; Boulesteix et al. 2012), and for a broader review, see Verikas et al. (2011). This universality is first and foremost linked to excellent predictive performance. This can be seen in Fernández-Delgado et al. (2014) which crowns RF in a recent large-scale comparative evaluation, whereas less than a decade earlier, the article in Wu et al. (2008) with similar objectives mentions CART but not yet random forests! In addition, they are applicable to many types of data. Indeed, it is possible to consider high-dimensional data for which the number of variables far exceeds the number of observations. In addition, they are suitable for both classification problems (categorical response variable) and regression problems (continuous response variable). They also allow handling a mixture of qualitative and quantitative explanatory variables. Finally, they are, of course, able to process standard data for which the number of observations is greater than the number of variables.

Beyond the performance and the easy to tune feature of the method with very few parameters to adjust, one of the most important aspects in terms of application is the quantification of the explanatory variables' relative importance. This concept, which is not so much examined by statisticians (see, for example, Grömping 2015, in regression), finds a convenient definition in the context of random forests that is easy to evaluate and which naturally extends to the case of groups of variables (Gregorutti et al. 2015).

Therefore, and we will emphasize this aspect very strongly, RF can be used for variable selection. Thus, in addition to a powerful prediction tool, it can also be used to select the most interesting explanatory variables to explain the response, among a potentially very large number of variables. This is very attractive in practice because it helps both to interpret more easily the results and, above all, to determine influential factors for the problem of interest. Finally, it can also be beneficial for prediction, because eliminating many irrelevant variables makes the learning task easier.

1.2 Notation

Throughout the book, we will adopt the following notations. We assume that a learning sample is available:

$$\mathcal{L}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

composed of n couples of independent and identically distributed observations, coming from the same common distribution as a couple (X, Y). This distribution is, of course, unknown in practice and the purpose is precisely to estimate it, or more specifically to estimate the link that exists between X and Y.

We call the coordinates of X the "input variables" (or "explanatory variables" or "variables"), where we note X^j for the jth coordinate, and we assume that $X \in \mathcal{X}$, a certain space that we will specify later. However, we assume that this space is of dimension p, where p is the (total) number of variables.

Y refers to the "response variable" (or "explained variable" or "dependent variable") and $Y \in \mathcal{Y}$. The nature of the regression or classification problem depends on the nature of the space $\mathcal{Y}$:

- If $\mathcal{Y} = \mathbb{R}$, we have a regression problem.
- If $\mathcal{Y} = \{1, \ldots, C\}$, we have a classification problem with C classes.

1.3 Statistical Objectives

Prediction

The first learning objective is prediction. We are trying, using the learning sample $\mathcal{L}_n$, to construct a predictor:

$$\widehat{h} : \mathcal{X} \to \mathcal{Y}$$

which associates a prediction $\widehat{y}$ of the response variable corresponding to any given input observation $x \in \mathcal{X}$.

The "hat" on $\widehat{h}$ is a notation to specify that this predictor is constructed using $\mathcal{L}_n$. We omit the dependence over n for the predictor to simplify the notations, but it does exist.

More precisely, we want to build a powerful predictor in terms of prediction error (also called generalization error):

- In regression, we will consider here the mathematical expectation of the quadratic error: $\mathrm{E}\left[(Y - \widehat{h}(X))^2\right]$.
- In classification, the probability of misclassification: $\mathrm{P}\left(Y \neq \widehat{h}(X)\right)$.

The prediction error depends on the unknown joint distribution of the random couple (X, Y), so it must be estimated. One classical way to proceed is, using a test sample $\mathcal{T}_m = \{(X'_1, Y'_1), \ldots, (X'_m, Y'_m)\}$, also drawn from the distribution of (X, Y), to calculate an empirical test error:

- In regression, it is the mean square error: $\frac{1}{m}\sum_{i=1}^{m}\left(Y'_i - \widehat{h}(X'_i)\right)^2$.
- In classification, the misclassification rate: $\frac{1}{m}\sum_{i=1}^{m}\mathbf{1}_{Y'_i \neq \widehat{h}(X'_i)}$.

In the case where a test sample is not available, the prediction error can still be estimated, for example, by cross-validation. In addition, we will introduce later on a specific estimate using random forests.

Remark 1.1 In this book, we focus on regression problems and/or supervised classification ones. However, RF have been generalized to various other statistical problems.

First, for survival data analysis, Ishwaran et al. (2008) introduced Random Survival Forests, transposing the main ideas of RF to the case for which the quantity to be predicted is the time to event. Let us also mention on this subject the work of Hothorn et al. (2006).

Random forests have also been generalized to the multivariate response variable case (see the review by Segal and Xiao 2011, which also provides references from the 1990s).

Selection and importance of variables
A second classical objective is variable selection. This involves determining a subset of the input variables that are actually useful and active in explaining the input–output relationship. The quality of a subset of selected variables is often assessed by the performance obtained with a predictor using only these variables instead of all the initial sets.

In addition, we can focus on constructing a hierarchy of input variables based on a quantification of the importance of the effects on the output variable. Such an index of importance therefore provides a ranking of variables, from the most important to the least important.

1.4 Packages

We will mainly focus on R three packages (R Core Team 2018):

- **rpart** (Therneau and Atkinson 2018) for tree methods, in Chap. 2.
- **randomForest** (Liaw and Wiener 2018) for random forests, in Chaps. 3 and 4.
- **VSURF** (Genuer et al. 2018) for variable selection using random forests, in Chap. 5.

Remark 1.2 Regarding the variants of random forests discussed in the previous section, the **randomForestSRC** package (Ishwaran and Kogalur 2017) provides a unified implementation of RF for regression, supervised classification, in a survival context as well as for the multivariate response case.

1.5 Datasets

1.5.1 Running Example: Spam Detection

We will illustrate the application of the different methods on the very classical `spam` data for educational purposes, as a running example.

This well-known and freely available dataset is due to an engineer from Hewlett-Packard company, named George, who analyzed a sample of his professional emails:

- The observations are the 4,601 emails, of which 2,788 (i.e., 60 %) are desirable emails and 1,813 (i.e., 40 %) are undesirable emails, i.e., spam.
- The response variable is therefore binary: `spam` or `non-spam`. We will rename the category `non-spam` to `ok` to make some graphs easier to read.
- There are $p = 57$ explanatory variables: 54 are proportions of occurrences of words or characters, such as `$` (denoted `charDollar`), `!` (denoted `charExclamation`), `free`, `money`, and `hp`, two are related to the lengths of the capital letter sequences (the average, `capitalAve`, the longest, `capitalLong`), and finally the last is the number of capital letters in the mail, `capitalTotal`. These variables are classical and are defined using standard text analysis procedures, allowing observations characterized by texts to be statistically processed through numerical variables.

The statistical objectives stated above are formulated for this example as follows: first, we want to build a good spam filter: a new email arrives, we have to predict if it is spam or not. Secondly, we are also interested in knowing which variables are the most important for the spam filter (here, words or characters).

To assess the performance of a spam filter, the dataset is randomly split into two parts: 2,300 emails are used for learning while the other 2,301 emails are used to test predictions.[1]

So we have a problem of **2-class classification** ($C = 2$) with a number of individuals ($n = 2,300$ for learning, model building) much larger than the number of variables ($p = 57$). In addition, we have a large test sample ($m = 2,301$) to evaluate an estimate of the prediction error.

Let us load the dataset into R, available in the **kernlab** package (Karatzoglou et al. 2004); let us rename the category `nonspam` to `ok` and fix the learning and test sets:

[1] Other usual choices are 70% of data for learning, 30% for test or even 80–20%: we choose 50–50% to stabilize estimation errors and reduce computational times.

```
> data("spam", package = "kernlab")
> set.seed(9146301)
> levels(spam$type) <- c("ok", "spam")
> yTable <- table(spam$type)
> indApp <- c(sample(1:yTable[2], yTable[2]/2),
    sample((yTable[2] + 1):nrow(spam), yTable[1]/2))
> spamApp <- spam[indApp, ]
> spamTest <- spam[-indApp, ]
```

Remark 1.3 The command `set.seed(9146301)` allows fixing the seed of the random numbers generator in R. Thus, if the previous instruction block is executed several times, there will be no variability in the learning and test samples.

1.5.2 Ozone Pollution

The `Ozone` data is used in many papers and is one of the classical benchmark datasets since the article of Breiman and Friedman 1985.

The objective here is to predict the maximum ozone concentration associated with a day in 1976 in the Los Angeles area, using 12 weather and calendar variables. The data consist of 366 observations and 13 variables, each observation is associated with a day. The 13 variables are as follows:

- `V1` Months: `1` = January, …, `12` = December.
- `V2` Day of the month `1` to `31`.
- `V3` Day of the week: `1` = Monday, …, `7` = Sunday.
- `V4` Daily maximum of hourly average of ozone concentrations.
- `V5` 500 millibar (*m*) pressure height measured at Vandenberg AFB.
- `V6` Wind speed (*mph*) at Los Angeles International Airport (LAX).
- `V7` Humidity (%) at LAX.
- `V8` Temperature (*degrees F*) measured at Sandburg, California.
- `V9` Temperature (*degrees F*) measured at El Monte, California.
- `V10` Inversion base height (*feet*) at LAX.
- `V11` Pressure gradient (*mmHg*) from LAX to Daggett, California.
- `V12` Inversion base temperature (*degrees F*) to LAX.
- `V13` Visibility (*miles*) measured at LAX.

So it is a problem of **regression** where we have to predict `V4` (the daily maximum ozone concentration) using the other 12 variables, nine meteorological variables (`V5` to `V13`), and three calendar variables (`V1` to `V3`).

In many cases, only continuous explanatory variables are considered. Here, the tree methods allow all of them to be taken into account, even if including `V2` the day of the month is a priori irrelevant.

This dataset is available in the `mlbench` package (Leisch and Dimitriadou 2010) and can be loaded into R using the following command:

```
> data("Ozone", package = "mlbench")
```

1.5.3 Genomic Data for a Vaccine Study

The dataset `vac18` is from an HIV prophylactic vaccine trial (Thiébaut et al. 2012). Expressions of a subset of 1,000 genes were measured for 42 observations corresponding to 12 negative HIV participants, from 4 different stimuli:

- The candidate vaccine (LIPO5).
- A vaccine containing the Gag peptide (GAG).
- A vaccine not containing the Gag peptide (GAG-).
- A non-stimulation (NS).

The prediction objective here is to determine, in view of gene expression, the stimulation that has been used. So it is a **4-class high-dimensional classification** problem. It should be noted that this prediction problem is an intermediate step in order to reach the actual objective which is the selection of the most useful genes for the discrimination between the different vaccines.

We load the `vac18` data, available in the `mixOmics` package (Le Cao et al. 2017):

```
> data("vac18", package = "mixOmics")
```

1.5.4 Dust Pollution

These data are published in Jollois et al. (2009).

Airborne particles come from various origins, natural or human-induced, and the chemical composition of these particles can vary a lot. In 2009, Air Normand, the air quality agency in Upper Normandy (Haute-Normandie), had about ten devices measuring the concentrations of PM10 particles of diameter of less than 10 µ/m, expressed in µ/g/m^3, in average over the past quarter of an hour. European regulation rules set the value of 50 µ/g/m^3 (as a daily average) as the limit not to exceed more than 35 days in the year for PM10.

We focused on a subnetwork of six PM10 monitoring stations: three in Rouen GCM (industrial), JUS (urban), and GUI (near traffic); two in Le Havre REP (traffic) and HRI (urban); and finally a rural station in Dieppe AIL.

The data considered for the six stations are

- For weather: rain PL, wind speed VV (max and average), wind direction DV (max and dominant), temperature T (min, max, and average), temperature gradient GT (Le Havre and Rouen), atmospheric pressure PA, and relative humidity HR (min, max, and average).
- For pollutants: dust (PM10), nitrogen oxides (NO, NO2) for urban pollution and sulfur dioxide (SO2) for industrial pollution: in addition to those measured at each station, pollutants measured nearby are added:
  - For GUI: addition of SO2 measured at JUS.
  - For REP: addition of SO2 measured in Le Havre (MAS station).
  - For HRI: addition of NO and NO2 measured at Le Havre (MAS station).

Let us load the data for the JUS station, included in the **VSURF** package:

```
> data("jus", package = "VSURF")
```