Bagging

From a course by Davy Paindaveine and Nathalie Vialaneix

Camille Mondon

1 Introduction

- Designed by Breiman (1996).
- The bootstrap has other uses than those described in the last chapter.
- In particular, it allows us to design **ensemble methods** in **statistical learning**.
- **Bagging** (**B**ootstrap **Agg**regat**ing**), which is the most famous approach in this direction, can be applied to both **regression** and **classification**.
- Below, we mainly focus on **bagging of classification trees**, but it should be clear that bagging of regression trees can be performed similarly.

2 Classification trees

2.1 The classification problem

- In classification, one observes (X_i, Y_i), $i = 1, \ldots, n$, where
  - X_i collects the values of p predictors on individual i, and
  - $Y_i \in \{1, 2, \ldots, K\}$ is the class to which individual i belongs.
- The problem is to classify a new observation for which we only see x, that is, to bet on the corresponding value $y \in \{1, 2, \ldots, K\}$.
- A classifier is a mapping

$$\begin{aligned} \phi_{\mathscr{S}} : \mathscr{X} &\rightarrow \{1, 2, \ldots, K\} \\ x &\mapsto \phi_{\mathscr{S}}(x), \end{aligned}$$

  that is designed using the sample $\mathscr{S} = \{(X_i, Y_i),\ i = 1, \ldots, n\}$.

```r
library(boot)
data(channing)
channing <- channing[, c("sex", "entry", "time", "cens")]
n <- nrow(channing)
channing[sample(1:n, 4), ]
```

	sex	entry	time	cens
262	Female	890	115	0
416	Female	994	46	1

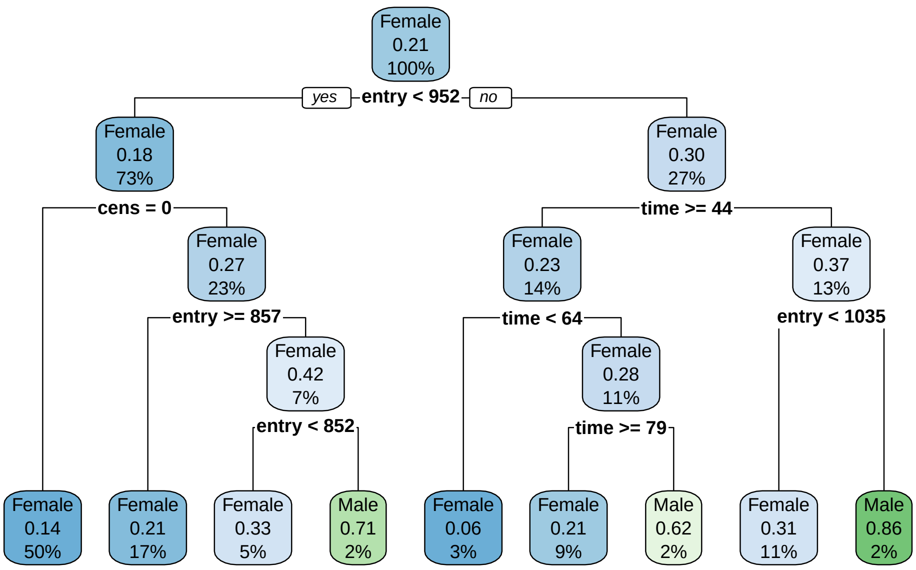
	sex	entry	time	cens
10	Male	837	108	1
220	Female	810	85	0

Predict `sex` $\in$ {Male, Female} on the basis of two numerical predictors (`entry`, `time`) and a binary one (`cens`).

2.2 Classification trees

In Chapter 3 of this course, we learned about a special type of classifiers $\phi_{\mathscr{S}}$, namely classification trees (Breiman et al. 1984).

```r
library(rpart)
library(rpart.plot)
fitted.tree <- rpart(sex ~ .,
  data = channing,
  method = "class"
)
rpart.plot(fitted.tree)
```



(+) Interpretability
(+) Flexibility
(–) Stability
(–) Performance

The process of **averaging** will reduce variability, hence, **improve stability**. Recall indeed that, if $U_1, \ldots, U_n$ are uncorrelated with variance σ^2, then

$$\mathrm{Var}[\bar{U}] = \frac{\sigma^2}{n}.$$

Since unpruned trees have low bias (but high variance), this reduced variance will lead to a low value of

$$\text{MSE} = \text{Var} + (\text{Bias})^2$$

which will ensure a **good performance**.

How to perform this **averaging**?

3 Bagging of classification trees

3.1 Bagging

Denote as $\phi_{\mathscr{S}}(x)$ the predicted class for predictor value x returned by the classification tree associated with sample $\mathscr{S} = \{(X_i, Y_i),\ i = 1, \ldots, n\}$.

Bagging of this tree considers predictions from B bootstrap samples

$$\begin{array}{ccccc} \mathscr{S}^{*1} & = & ((X_1^{*1}, Y_1^{*1}), \ldots, (X_n^{*1}, Y_n^{*1})) & \rightsquigarrow & \phi_{\mathscr{S}^{*1}}(x) \\ \vdots & & & & \vdots \\ \mathscr{S}^{*b} & = & ((X_1^{*b}, Y_1^{*b}), \ldots, (X_n^{*b}, Y_n^{*b})) & \rightsquigarrow & \phi_{\mathscr{S}^{*b}}(x) \\ \vdots & & & & \vdots \\ \mathscr{S}^{*B} & = & ((X_1^{*B}, Y_1^{*B}), \ldots, (X_n^{*B}, Y_n^{*B})) & \rightsquigarrow & \phi_{\mathscr{S}^{*B}}(x) \end{array}$$

then proceeds by **majority voting** (i.e., the most frequently predicted class wins):

$$\phi_{\mathscr{S}}^{\text{Bagging}}(x) = \underset{k \in \{1,\ldots,K\}}{\text{argmax}} \#\{b : \phi_{\mathscr{S}^{*b}}(x) = k\}$$

3.2 Toy illustration: bagging with $B = 3$ trees

```
library(boot)
set.seed(20)
d <- sample(1:n, replace = TRUE)
fitted.tree <- rpart(sex ~ ., data = channing[d, ], method = "class")
rpart.plot(fitted.tree)
```

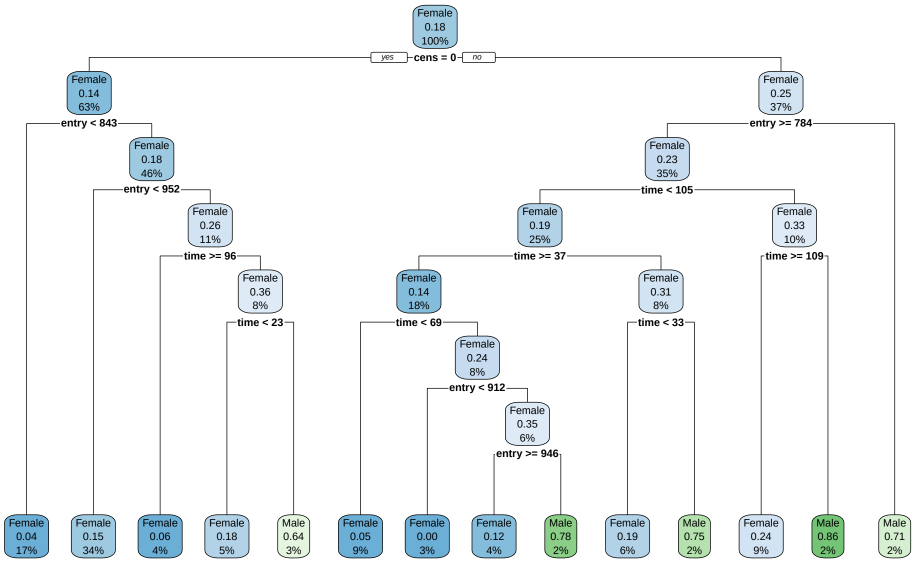


Figure 1: Classification tree from the first bootstrap sample.

```
channing[1, 2:4]
```

entry	time	cens
782	127	1

```
predict(fitted.tree, channing[1, ],
  type = "class"
)
```

```
   1
Male
Levels: Female Male
```

```
d <- sample(1:n, replace = TRUE)
fitted.tree <- rpart(sex ~ ., data = channing[d, ], method = "class")
rpart.plot(fitted.tree)
```

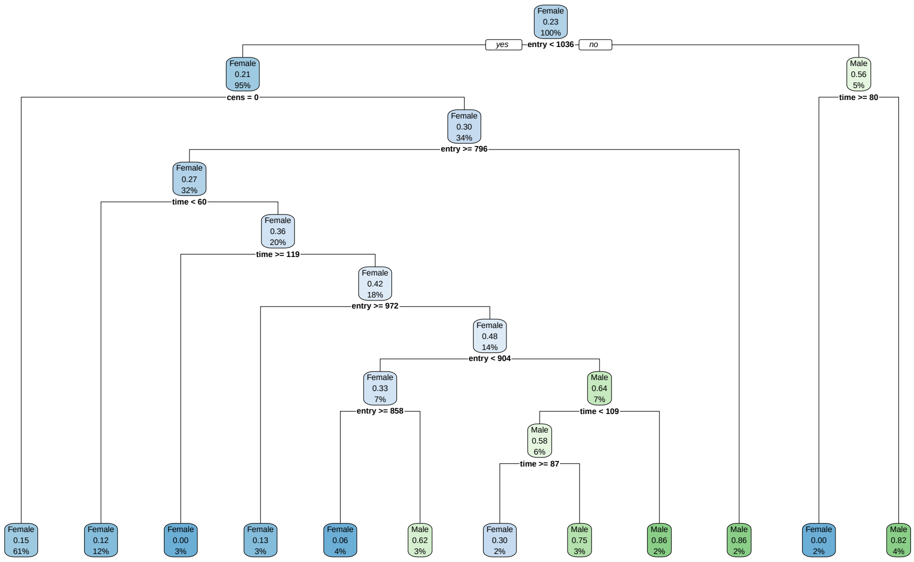


Figure 2: Classification tree from the second bootstrap sample.

```r
channing[1, 2:4]
```

entry	time	cens
782	127	1

```r
predict(fitted.tree, channing[1, ],
  type = "class"
)
```

```
   1
Male
Levels: Female Male
```

```r
d <- sample(1:n, replace = TRUE)
fitted.tree <- rpart(sex ~ ., data = channing[d, ], method = "class")
rpart.plot(fitted.tree)
```

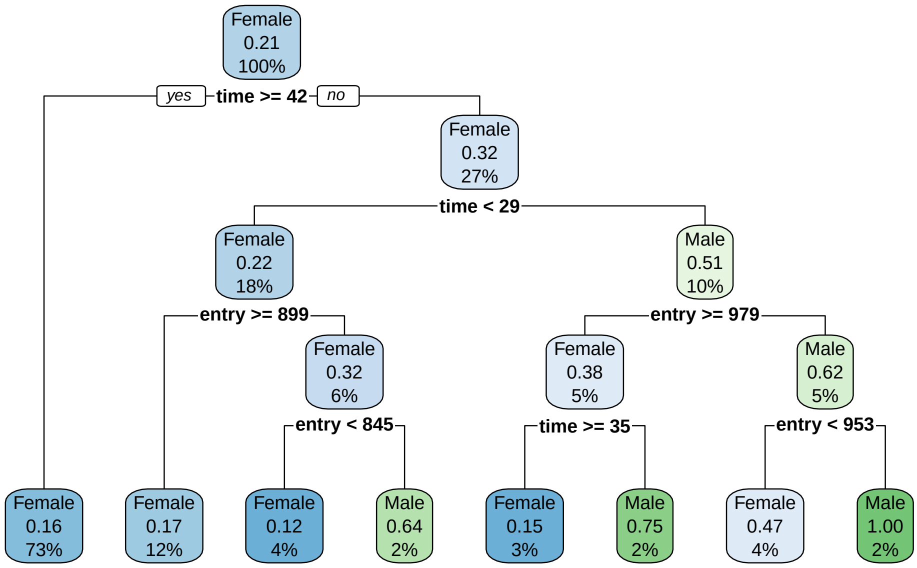


Figure 3: Classification tree from the third bootstrap sample.

```
channing[1, 2:4]
```

entry	time	cens
782	127	1

```
predict(fitted.tree, channing[1, ],
  type = "class"
)
```

```
     1
Female
Levels: Female Male
```

For $x = (\text{entry}, \text{time}, \text{cens}) = (782, 127, 1)$,

- **two** (out of the $B = 3$ trees) **voted for Male**
- **one** (out of the $B = 3$ trees) **voted for Female**,

the bagging classifier will thus classify x into **Male**.

Of course, B is usually much larger ($B = 500$? $B = 1000$?), which requires automating the process (through, e.g., the `boot` function).

4 How much do you gain?

4.1 A simulation

We repeat $M = 1000$ times the following experiment:

(1) Split the data set into a training set (of size 300) and a test set (of size 162);
(2) (a) **train** a classification tree on the training set and evaluate its **test** error (i.e., misclassification rate) on the test set;
   (b) do the same with a bagging classifier using $B = 500$ trees.

This provides $M = 1000$ test errors for the **direct** (single-tree) approach, and $M = 1000$ test errors for the **bagging** approach.

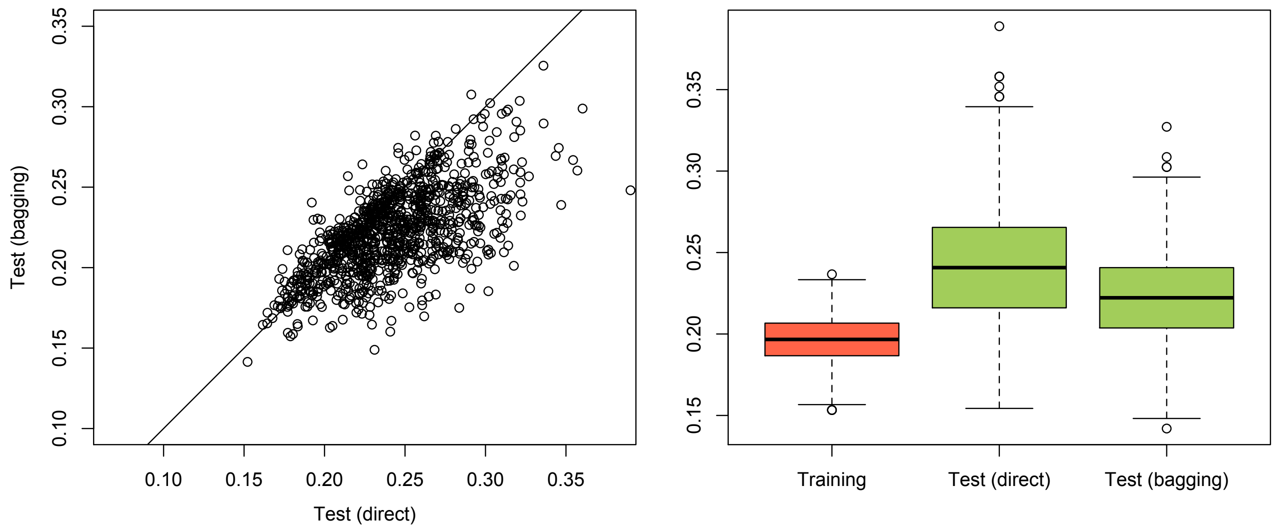


Figure 4: Results of the simulation (Q-Q plot and boxplot).

5 Estimating the prediction accuracy

5.1 Estimating the prediction (lack of) accuracy

Several strategies to estimate prediction accuracy of a classifier:

(1) Compute a test error (as above): Partition the data set $\mathscr{S}$ into a training set $\mathscr{S}_{\text{train}}$ (to train the classifier) and a test set $\mathscr{S}_{\text{test}}$ (on which to evaluate the misclassification rate e_{test}).

---

(2) Compute an L-fold cross-validation error:

Partition the data set $\mathscr{S}$ into L folds $\mathscr{S}_\ell$, $\ell = 1, \dots, L$. For each ℓ, evaluate the test error $e_{\text{test},\ell}$ associated with training set $\mathscr{S} \setminus \mathscr{S}_\ell$ and test set $\mathscr{S}_\ell$.

Table 5: L-fold cross-validation framework.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Run ℓ=1	Test	Train	Train	Train	Train
Run ℓ=2	Train	Test	Train	Train	Train
Run ℓ=3	Train	Train	Test	Train	Train
Run ℓ=4	Train	Train	Train	Test	Train
Run ℓ=5	Train	Train	Train	Train	Test

Then the (L-fold) 'cross-validation error' is:

$$e_{\mathrm{CV}} = \frac{1}{L}\sum_{\ell=1}^{L} e_{\mathrm{test},\ell}$$

---

(3) Compute the Out-Of-Bag (OOB) error[1]:

For each observation X_i from $\mathscr{S}$, define the OOB prediction as

$$\phi_{\mathscr{S}}^{\mathrm{OOB}}(X_i) = \underset{k\in\{1,\ldots,K\}}{\mathrm{argmax}}\#\{b : \phi_{\mathscr{S}^{*b}}(X_i) = k \text{ and } (X_i, Y_i) \notin \mathscr{S}^{*b}\}$$

This is a **majority voting** discarding, quite naturally, bootstrap samples that use (X_i, Y_i) to train the classification tree. The OOB error is then the corresponding misclassification rate

$$e_{\mathrm{OOB}} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}[\phi_{\mathscr{S}}^{\mathrm{OOB}}(X_i) \neq Y_i]$$

5.2 Final remarks

- **Bagging of trees can also be used for regression**. The only difference is that majority voting is then replaced with an averaging of individual predicted responses.
- **Bagging is a general device that applies to other types of classifiers**. In particular, it can be applied to k-NN classifiers (we will illustrate this in the practical sessions).
- **Bagging affects interpretability of classification trees**. There are, however, solutions that intend to measure importance of the various predictors (see the next section).

References

Breiman, Leo. 1996. "Bagging Predictors." *Machine Learning* 24 (2): 123–40. https://doi.org/10.1007/BF00058655.

Breiman, Leo, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. 1984. *Classification And Regression Trees*. 1st ed. Routledge. https://doi.org/10.1201/9781315139470.

[1]This is for bagging procedures only.