Day 03 - Exercise 01 - Boosting Trees

Victor Calderon

15 August, 2018

Preamble

The exercises here are taken from the `boosting-trees.R` file in the `scripts/day_03` directory.

In order to be able to run the code in here, you need to run the following:

```
## Installing packages
install.packages('rpart')
install.packages('mlbench')
install.packages('caret')
install.packages('gbm')

## Loading packages
library('rpart')
library('mlbench')
library('caret')
library('gbm')
```

These packages are needed in order to run some of the commands.

We can now **load** the necessary libraries:

```
library('rpart')
library('mlbench')
library('gbm')
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:rpart':
##
##     solder
```

```
## Loading required package: lattice
```

```
## Loading required package: splines
```

```
## Loading required package: parallel
```

```
## Loaded gbm 2.1.3
```

```
library('caret')
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:survival':
##
##     cluster
```

Exercise 1

1. Use the 'cv.folds' option to 'gbm' to perform 10-fold cross-validation of the 'gbm' model for breast cancer above by varying 'n.trees'. Select an optimal 'n.trees' and compute the test error using the testing data.

```
# Defining dataset
data('BreastCancer')
BreastCancer$y <- ifelse(BreastCancer$Class == "malignant", 1, -1)
BreastCancer$Class <- NULL
BreastCancer$Id <- NULL
head(BreastCancer)
```

```
##   Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size Bare.nuclei
## 1            5         1          1             1            2           1
## 2            5         4          4             5            7          10
## 3            3         1          1             1            2           2
## 4            6         8          8             1            3           4
## 5            4         1          1             3            2           1
## 6            8        10         10             8            7          10
##   Bl.cromatin Normal.nucleoli Mitoses  y
## 1           3               1       1 -1
## 2           3               2       1 -1
## 3           3               1       1 -1
## 4           3               7       1 -1
## 5           3               1       1 -1
## 6           9               7       1  1
```

We now create a *training* and *testing* datasets, and compute the estimator using the **GBM** classifier:

We define a function `breast_err_calculation` that will modify the data, and will depend on the number of trees and the number of K-fold cross-validations.

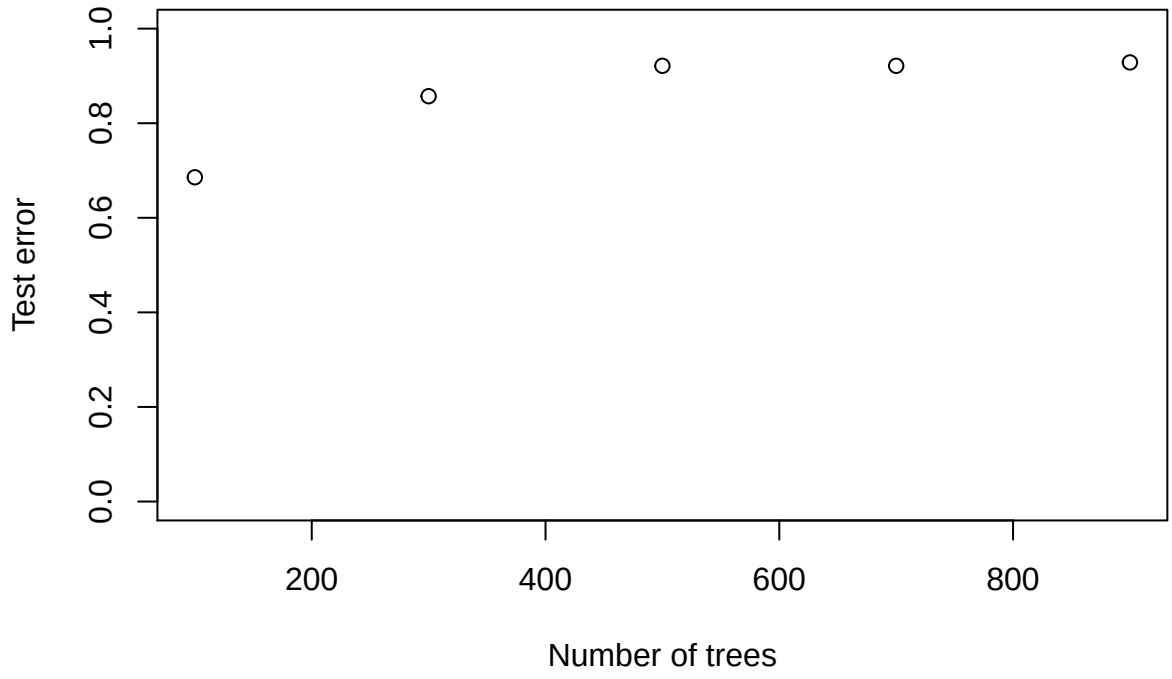
```
breast_err_calculation <- function(dataset, ntrees = 1000, cv_folds = 10){
  # Creating fraction of the training/testing dataset
  trn_idx <- sample(1:nrow(dataset), size = 0.8*nrow(dataset))
  bc_trn <- dataset[ trn_idx,]
  bc_tst <- dataset[-trn_idx,]
  ## GBM Classifier
  bc_trn$y <- (bc_trn$y + 1)/2 ## convert (-1, 1) to (0, 1)
  bc_tst$y <- (bc_tst$y + 1)/2 ## convert (-1, 1) to (0, 1)
  # Computing the GBM classifier
  bc_fit <- gbm(y ~ .,
                distribution = 'bernoulli',
                data=bc_trn,
                n.trees = ntrees,
                cv.folds = cv_folds)
  # Estimating errors
  pre_tst <- predict(bc_fit, bc_tst, n.trees=bc_fit$n.trees, type='response') > 0.5
  gmb_err_mean <- mean(pre_tst == bc_tst$y)

  return(gmb_err_mean)
}
```

We can now compute the errors for each of the different types of `n.trees`:

```
## Looping over different values of `n_trees`

n_init = 100
n_end  = 1000
n_step = 200

n_trees_arr <- seq(n_init, n_end, n_step)
gm_err_arr  <- rep(0, length(n_trees_arr))

for (x in 1:length(n_trees_arr)) {
  gm_err_arr[x] = breast_err_calculation(BreastCancer, ntrees = n_trees_arr[x])
}
```

Now we can go ahead and plot the results of `n_trees_arr` and `gm_err_arr`:

```
plot(n_trees_arr,
     gm_err_arr,
     xlab = 'Number of trees',
     ylab = 'Test error',
     xlim = c(min(n_trees_arr), max(n_trees_arr)),
     ylim = c(0,1))
```



Exercise 2

Using the income data from previous examples to fit a boosted tree ('gbm') for for predicting income, given education and seniority. Be sure to read through the help file to make the appropriate modifications.

```
inc <- read.csv(url("http://www-bcf.usc.edu/~gareth/ISL/Income2.csv"),
                header=T, row.names = 1)
```

Exercise 3

Create a figure using the 'persp' function to display the prediction surface of the boosted tree model trained in part 1. (hint: use the 'plot_inc_data' function from previous lab)