Predicting exercise performance

Michael Rahija

9/14/2019

Background and dataset

The objective of the assignment is to use a dataset to build a model which predicts if individuals perform a certain exercise correctly. The dataset contains about 150 variables for 6 individuals, and each record contains a classe column which groups the movement into categories A, B, C, and D. Class means the excercise was performed correctly, and the other classes indicate some kind of mistake. All information about the dataset can be found here: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har#weight_lifting_exercises

Load data

The first thing to load data and do is drop variables which clearly do not have have any predictive power (i.e. X, user_name, timestamps).

```
library(dplyr)
library(caret)

library(parallel)
library(doParallel)



knitr::opts_chunk$set(echo = TRUE)
#setwd("~/Dropbox/machinelearning")
setwd("C:/Users/Rahija/Dropbox/machinelearning")

df <- read.csv("pml-training-1.csv", stringsAsFactors = F)

df<- select(df, -c("X","user_name","raw_timestamp_part_1","raw_timestamp_part_2","cvtd_timestamp", "new_

testing <- read.csv("pml-testing.csv", stringsAsFactors = F)
```

Cleaning and feature selection

The next thing to do is remove columns with a high percentage of missing values. Missing values in the datasets are either NAs or just completely blank. Columns which are missing more than 90% of values were dropped.

```
blankTest <- function(x, threshold = .90){

  num <- sum( (x == "" | is.na(x)))
  denom <- length(x)
  result <- (num/denom) > threshold

result
}
```

```
colsNA <- sapply(df, blankTest)

df <- df[,!colsNA]
ncol(df)
```

```
## [1] 54
```

We are left with a total of 53 features.

Building and testing model

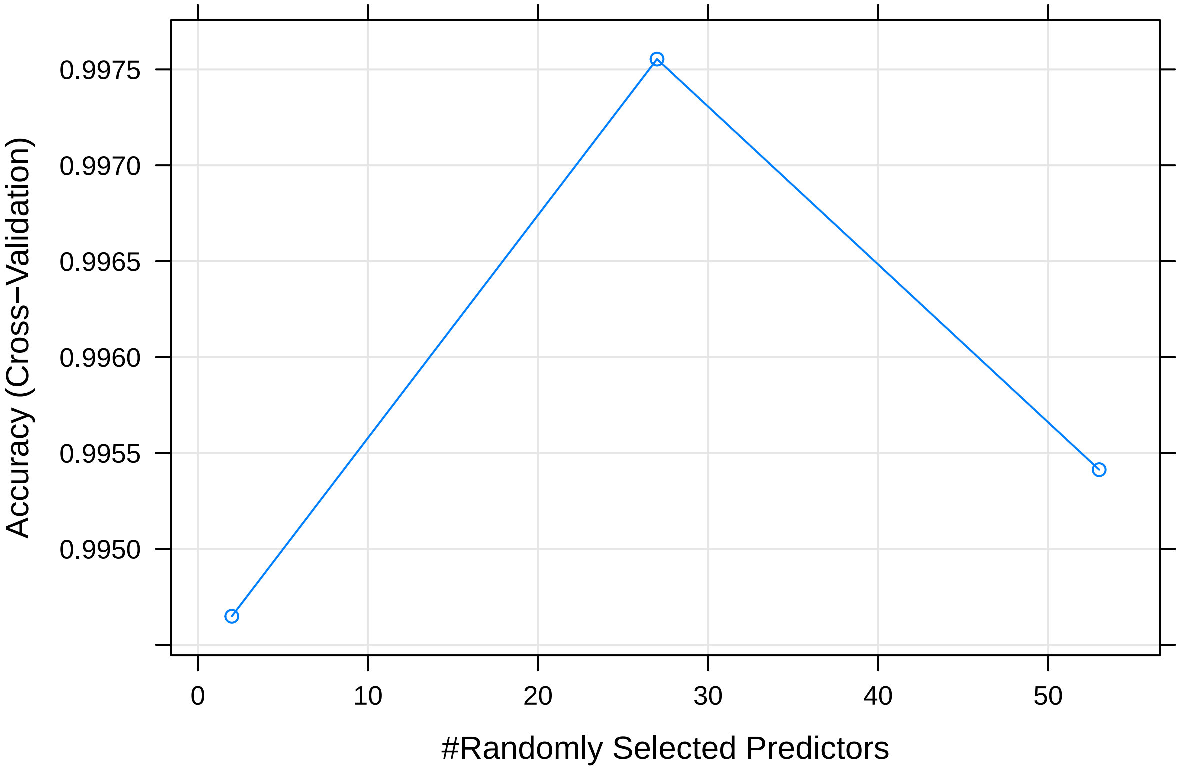
In order to speed-up the operation, we'll implement parallel processing. To reduce model variances, we will use 3 fold cross validation.

```
cluster <- makeCluster(detectCores() - 1) # convention to leave 1 core for OS
registerDoParallel(cluster)

fitControl <- trainControl(method = "cv",
                           number = 3,
                           allowParallel = TRUE)

set.seed(1234)
modelFit <- train(x = df[,-54], y = df$classe, method = "rf", data = t, trControl = fitControl)
#modelFit <- randomForest(x = df[,-1])


stopCluster(cluster)
registerDoSEQ()

modelFit
```

```
## Random Forest
##
## 19622 samples
##    53 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 13081, 13081, 13082
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9946489  0.9932309
##   27    0.9975538  0.9969057
##   53    0.9954132  0.9941978
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

The model fit shows 99% accuracy using 27 predictors. The plot below shows that the prediction accuracy peaks at 27 predictors.

```
plot(modelFit)
```



The OOB error is only .12% signaling that the out of sample error rate will be very low.

```
#modelFit$finalModel
```