Practical Machine Learning Assignment

Vipul Chadda

17 May 2018

Overview

This report is part of the final assignment of the Practical Machine Learning course on Coursera. Quoting directly from the assignment question, “One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it.” The goal is to predict how well a particular exercise is done by using data from 4 accelerometers.

The data

The data used is from http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har).

The training data provided has a collection of raw measurements from the accelerometers as well as calculated/derived measurements. The derived measurements are made at particular intervals and are calculated from the raw measurements in that interval. The number derived measurements are very less.

For the prediction, we will ignore the derived measurements as well as the time-stamp related columns. A new data frame is created with only the columns required for predictions.

```r
library(caret)
```

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

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

```r
library(parallel)
library(doParallel)
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```r
set.seed(610)
initialData <- read.csv("pml-training.csv", na.strings = c("NA", "#DIV/0!", ""))

ignoreCols <- grep("kurtosis_|skewness_|max_|min_|amplitude_|var_total_|avg_|stddev_|var_", c
olnames(initialData))
cleanData <- initialData[, -ignoreCols]

cleanData <- cleanData[, -c(1,3:7)]
```

This cleaned data is divided into training and testing sets using the `createDataPartition` method in the caret package.

```
inTrain <- createDataPartition(cleanData$classe, p=0.7, list = FALSE)
training <- cleanData[inTrain,]
testing <- cleanData[-inTrain,]
```

The model

Building the model using random forest method as it is highly accurate. The cons of using this method are speed and overfitting. To improve on these points, 3-fold cross validation and parallel processing is used.

Parallel processing is achived using parallel and doParallel packages.

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

# For cross validation
fitControl <- trainControl(method = "cv", number = 3, allowParallel = TRUE)

fitRF <- train(classe ~ ., method="rf", data=training,trControl = fitControl)
```

The accuracy of the modal comes up to `0.9896252` which gives an out of sample error rate of approximately 1.1 % using the cross validation data.

Using the testing set, the accuracy comes up to `0.9922` , making the out of sample error rate as 0.78 %.

```
predRF <- predict(fitRF, testing)
confusionMatrix(predRF, testing$classe)
```

Conclusion

Using random forest method with 3-fold cross validation gives a prediction accuracy of 99.22 % of how a particular activity is performed based on the data used by 4 accelerometers.