Machine Learning Coursera Project

Javier Padilla Jr

August 30, 2016

The application of instruments such as Jawbone Up, Nike FuelBand and Fitbit makes it possible to collect a large amount of data about personal activity in a cost effective manner.

In this project, data will be measured from accelerometers on the belt, forearm, arm, and dumbbell of 6 participants to predict the manner in which they did the exercise.

```r
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```r
library(caret)
```

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

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

```r
library(randomForest)
```

```
## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:Hmisc':
##
##     combine
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(foreach)
library(doParallel)
```

```
## Loading required package: iterators

## Loading required package: parallel
```

```
set.seed(2048)
options(warn=-1)
```

Load the data both from the provided training and test data provided by COURSERA. Some values contained a "#DIV/0!" that I replaced with an NA value.

```
training_data <- read.csv("pml-training.csv", na.strings=c("#DIV/0!") )
evaluation_data <- read.csv("pml-testing.csv", na.strings=c("#DIV/0!") )
```

Cast all columns to the number 8 to the end to be numeric.

```
for(i in c(8:ncol(training_data)-1)) {training_data[,i] = as.numeric(as.character(training_data[,i]))}

for(i in c(8:ncol(evaluation_data)-1)) {evaluation_data[,i] = as.numeric(as.character(evaluation_data[,i
```

Determine and display out feature set.

```
feature_set <- colnames(training_data[colSums(is.na(training_data)) == 0])[-(1:7)]
model_data <- training_data[feature_set]
feature_set
```

```
##  [1] "roll_belt"            "pitch_belt"           "yaw_belt"
##  [4] "total_accel_belt"     "gyros_belt_x"         "gyros_belt_y"
##  [7] "gyros_belt_z"         "accel_belt_x"         "accel_belt_y"
## [10] "accel_belt_z"         "magnet_belt_x"        "magnet_belt_y"
## [13] "magnet_belt_z"        "roll_arm"             "pitch_arm"
## [16] "yaw_arm"              "total_accel_arm"      "gyros_arm_x"
## [19] "gyros_arm_y"          "gyros_arm_z"          "accel_arm_x"
## [22] "accel_arm_y"          "accel_arm_z"          "magnet_arm_x"
## [25] "magnet_arm_y"         "magnet_arm_z"         "roll_dumbbell"
## [28] "pitch_dumbbell"       "yaw_dumbbell"         "total_accel_dumbbell"
## [31] "gyros_dumbbell_x"     "gyros_dumbbell_y"     "gyros_dumbbell_z"
## [34] "accel_dumbbell_x"     "accel_dumbbell_y"     "accel_dumbbell_z"
## [37] "magnet_dumbbell_x"    "magnet_dumbbell_y"    "magnet_dumbbell_z"
## [40] "roll_forearm"         "pitch_forearm"        "yaw_forearm"
## [43] "total_accel_forearm"  "gyros_forearm_x"      "gyros_forearm_y"
## [46] "gyros_forearm_z"      "accel_forearm_x"      "accel_forearm_y"
## [49] "accel_forearm_z"      "magnet_forearm_x"     "magnet_forearm_y"
## [52] "magnet_forearm_z"     "classe"
```

Next, model the data built from our feature set.

```
idx <- createDataPartition(y=model_data$classe, p=0.75, list=FALSE )
training <- model_data[idx,]
testing <- model_data[-idx,]
```

There are several examples of how to perform parallel processing with random forests in R. This enabled the project's predications.

```
registerDoParallel()
x <- training[-ncol(training)]
y <- training$classe

rf <- foreach(ntree=rep(150, 6), .combine=randomForest::combine, .packages='randomForest') %dopar% {
randomForest(x, y, ntree=ntree)
}
```

Conclusions and Test Data Submit

Overall, it can be seen from the confusion matrix that this model is very accurate. The test data was around 99% accurate I expected nearly all of the submitted test cases to be correct. It turned out they were all correct. Thank you for your time.

```
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}


x <- evaluation_data
x <- x[feature_set[feature_set!='classe']]
answers <- predict(rf, newdata=x)

answers
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  B  A  B  A  A  E  D  B  A  A  B  C  B  A  E  E  A  B  B  B
## Levels: A B C D E
```

```
pml_write_files(answers)
```