Human Activity Recognition

Raymond Usher

24 de marzo de 2016

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. 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 participants were asked to perform barbell lifts correctly and incorrectly in 5 different ways.

```
Class A: exactly according to the specification
Class B: throwing the elbows to the front
Class C: lifting the dumbbell only halfway
Class D: lowering the dumbbell only halfway
Class E: throwing the hips to the front
```

Initially **MY GOAL** was to use **ONLY** data from **ACCELEROMETERS** on the belt, forearm, arm, and dumbell of 6 participants but after a while (Getting a model that was no precise as required for the Quizz) I include the 4 gyroscopes information.

Expecifically there were devices in

```
Arm
Forearm
Dumbbel
Belt
```

Each device taking accelerometer (x,y,z) and gyroscopes (x,y,z) measures

Load files and libraries

```
setwd("C:/pml_p1/GitFiles")
library(dplyr   , warn.conflicts = FALSE, verbose=FALSE)
library(lattice , warn.conflicts = FALSE, verbose=FALSE)
suppressWarnings( library(ggplot2 , warn.conflicts = FALSE, verbose=FALSE))
suppressWarnings( library(randomForest, warn.conflicts = FALSE, verbose=FALSE))
```

```
## randomForest 4.6-12
```

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

```
library(caret   , warn.conflicts = FALSE, verbose=FALSE)
set.seed(6688)
```

Reading the data from my sources

```
  usein.TrainingAndTesting.rawData  <-read.csv2("../pml-training.csv",sep=",")
  usein.Quizz.rawData <-read.csv2("../pml-testing.csv",sep=",")
```

Knowing the data (Some exploratory data analysis)

```
summary(usein.TrainingAndTesting.rawData$classe)
```

```
##    A    B    C    D    E 
## 5580 3797 3422 3216 3607
```

```
names(usein.TrainingAndTesting.rawData)[grep(".*accel.*|.*gyro.*", names(usein.Traini
ngAndTesting.rawData))]
```

```
##  [1] "total_accel_belt"     "var_total_accel_belt" "gyros_belt_x"        
##  [4] "gyros_belt_y"         "gyros_belt_z"         "accel_belt_x"        
##  [7] "accel_belt_y"         "accel_belt_z"         "total_accel_arm"     
## [10] "var_accel_arm"        "gyros_arm_x"          "gyros_arm_y"         
## [13] "gyros_arm_z"          "accel_arm_x"          "accel_arm_y"         
## [16] "accel_arm_z"          "total_accel_dumbbell" "var_accel_dumbbell"  
## [19] "gyros_dumbbell_x"     "gyros_dumbbell_y"     "gyros_dumbbell_z"    
## [22] "accel_dumbbell_x"     "accel_dumbbell_y"     "accel_dumbbell_z"    
## [25] "total_accel_forearm"  "var_accel_forearm"    "gyros_forearm_x"     
## [28] "gyros_forearm_y"      "gyros_forearm_z"      "accel_forearm_x"     
## [31] "accel_forearm_y"      "accel_forearm_z"
```

Now, I want to filter the data. The instructions for the project are not clear. One part says that I should use the accelerometer data and other says that I can use any variable. For purposes of this project, I am going to work **with accelerometer and gyroscopes data** So, I going to select the important columns using *dplyr* library

```
test.accelerometer.data <- usein.TrainingAndTesting.rawData %>%
  select(
    classe, total_accel_belt, accel_belt_x,accel_belt_y,accel_belt_z,total_accel_arm,
accel_arm_x,accel_arm_y,accel_arm_z,
    total_accel_dumbbell, accel_dumbbell_x,accel_dumbbell_y,accel_dumbbell_z,total_ac
cel_forearm,accel_forearm_x,accel_forearm_y,accel_forearm_z
    ,
    gyros_belt_x, gyros_belt_y, gyros_belt_z, gyros_arm_x, gyros_arm_x, gyros_arm_y,
gyros_arm_z, gyros_dumbbell_x, gyros_dumbbell_y, gyros_dumbbell_z,
    gyros_forearm_x,gyros_forearm_y,gyros_forearm_z
  )

quizz.accelerometer.data <-usein.Quizz.rawData %>%
 select(
    total_accel_belt, accel_belt_x,accel_belt_y,accel_belt_z,total_accel_arm,accel_ar
m_x,accel_arm_y,accel_arm_z,
    total_accel_dumbbell, accel_dumbbell_x,accel_dumbbell_y,accel_dumbbell_z,total_ac
cel_forearm,accel_forearm_x,accel_forearm_y,accel_forearm_z
    ,
    gyros_belt_x, gyros_belt_y, gyros_belt_z, gyros_arm_x, gyros_arm_x, gyros_arm_y,
gyros_arm_z, gyros_dumbbell_x, gyros_dumbbell_y, gyros_dumbbell_z,
    gyros_forearm_x,gyros_forearm_y,gyros_forearm_z
  )
# Convert to numeric gyrcoscopes fields that were loaded as fact data.
AllDevices <-c("belt","arm","dumbbell","forearm")
AllAxis <- c("x","y","z")
for(theDevice in AllDevices ) {
    for( theAxis in AllAxis ){
      name <- paste("gyros_",theDevice,"_",theAxis,sep="")
      test.accelerometer.data [,name] <-  as.numeric(test.accelerometer.data [,name])
      quizz.accelerometer.data[,name] <-  as.numeric(quizz.accelerometer.data[,name])

    }
}
```

Partitioning the data in two sets: one to train and the other for validation

Reserve a 75% for training and 25% for testing.

```
inTrain = createDataPartition(test.accelerometer.data$classe, p = 3/4, list=FALSE)
trainingData = test.accelerometer.data[inTrain,]
validationData = test.accelerometer.data[-inTrain,]
```

Generating the *knowledge*

```
# rf=Random forest, cv=Cross validation, number of folds=3
randomForest.model  <- train(classe ~., method="rf", data=trainingData, trControl=tra
inControl(method='cv', number=3, allowParallel=TRUE ))
# randomForest.model$finalModel
```

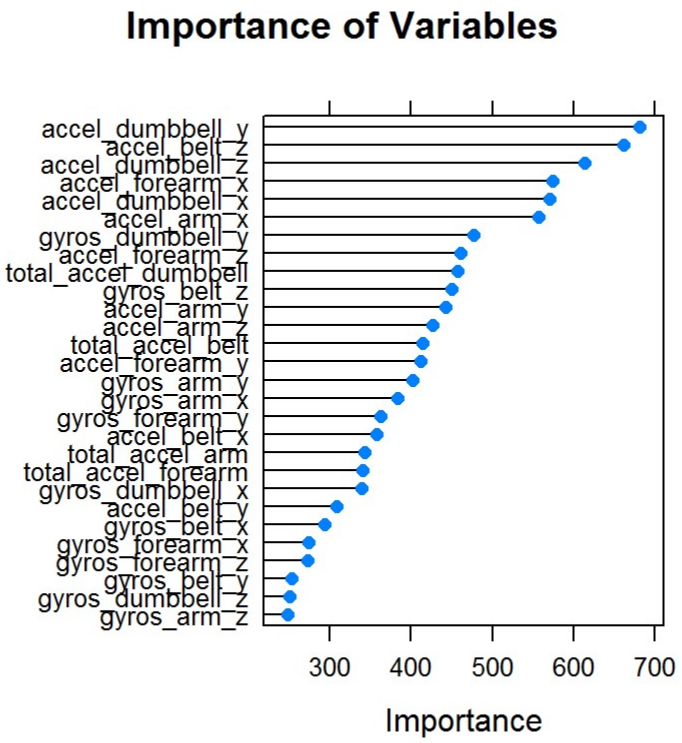
Testing and evaluating precision of the model created

```
trainingPrediction <- predict(randomForest.model , validationData)
confusionMatrix(trainingPrediction, validationData$classe)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1385   16    6    6    0
##          B    1  901    9    2    1
##          C    3   30  836   31    4
##          D    6    1    4  765    0
##          E    0    1    0    0  896
## 
## Overall Statistics
##                                           
##                Accuracy : 0.9753          
##                  95% CI : (0.9706, 0.9795)
##     No Information Rate : 0.2845          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.9688          
##  Mcnemar's Test P-Value : NA              
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9928   0.9494   0.9778   0.9515   0.9945
## Specificity            0.9920   0.9967   0.9832   0.9973   0.9998
## Pos Pred Value         0.9802   0.9858   0.9248   0.9858   0.9989
## Neg Pred Value         0.9971   0.9880   0.9952   0.9906   0.9988
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2824   0.1837   0.1705   0.1560   0.1827
## Detection Prevalence   0.2881   0.1864   0.1843   0.1582   0.1829
## Balanced Accuracy      0.9924   0.9731   0.9805   0.9744   0.9971
```

With an accuracy of 98% the model is acceptable.

```
print(plot(varImp(randomForest.model, scale = FALSE),main="Importance of Variables"))
```



Making Test Set Predictions

Now, I will use the model to predict the label for the observations in the quizz dataset

```
quizzPrediction <- predict(randomForest.model, quizz.accelerometer.data )
# quizzPrediction
```

Conclusion

Random forests algorithm performs nice. I could get a more accurate model using more variables, but my main concern was to use only data that I could understand.

Adressing specific instructions of the project

1) How I built my model

Selecting numeric variables from the accelerators and gyroscopes: arm, belt, dumbbell, forearm. Each device givin 3 measures (x,y,z)

2) How I used Cross Validation

In the function train I instructed that I wanted to use: random forest with cross validation dividing the data en 3 folds. I tried with 5, 7 and more folds, but the processing time increased and the final result was the same.

3) What I think the expected out of sample error is

Overfitting of small quantity of variables. To better the model I shall include more variables.

4) Why I made the choices I did

I selected *random forest* because is one of most winning algorithms in Kaggle competitions. I used only accelerators and gyroscopes data because for me was obvious the measures in that devices should predict the way the lift was made.

Mentions

The training data for this project are available here: [https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)]

The test data are available here: [https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)]

The data for this project come from this source: [http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har)].

Read more: [http://groupware.les.inf.puc-rio.br/har#ixzz3TROgwbfY (http://groupware.les.inf.puc-rio.br/har#ixzz3TROgwbfY)]

Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human ’13) . Stuttgart, Germany: ACM SIGCHI, 2013.