The Project of Practical Machine Learning

Jianjun Luo

Saturday, May 07, 2016

Overview

The data about personal activity could now be collected with devices such as Jawbone Up, Nike FuelBand, and Fitbit. This project uses data from accelerometers on the belt, forearm, arm, and dumbell. Participants could be classified in 5 different ways according to their manner in which they did the exercise:
1. Class A: exactly according to the specification
2. Class B: throwing the elbows to the front
3. Class C: lifting the dumbbell only halfway
4. Class D: lowering the dumbbell only halfway
5. Class E: throwing the hips to the front

For more details on this project see http://groupware.les.inf.puc-rio.br/har#ixzz481cbSG3W (http://groupware.les.inf.puc-rio.br/har#ixzz481cbSG3W)

Data Processing

The data for this project are from http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har). They are available at https://d396qusza40orc.cloudfront.net/predmachlearn (https://d396qusza40orc.cloudfront.net/predmachlearn)
First, let's download and read in the training and testing dataset. Then, do some exploratory analyses and data clean by removing features with mostly missing values and columns which are not suitable to the prediction. Finally, split the training data into training/validataion sets for the cross validation.

```
rm(list=ls())
# Download two data sets
download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv",destfile
="pml-training.csv", method="libcurl")
download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv", destfile
="pml-testing.csv",method="libcurl")
# Read in training and testing sets
trainingRaw <- read.csv("pml-training.csv",  na.strings=c("", "NA"))
testingRaw <- read.csv("pml-testing.csv", na.strings=c("", "NA"))
# Data's property
dim(trainingRaw)
```

```
## [1] 19622   160
```

```
dim(testingRaw)
```

```
## [1]  20 160
```

```
#summary(trainingRaw)   # output is hidden due to the size
#summary(testingRaw)
# Data clean
training <- trainingRaw[, colSums(is.na(trainingRaw))==0] # remove NA
training <-training[,-c(1:7)]
testing <- testingRaw[, colSums(is.na(testingRaw))==0]
testing <- testing[,-c(1:7)]
# Slice the training data into training and validation sets for cross-validation
library(caret)
```

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

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

```
set.seed(123)
inTrain  <- createDataPartition(y=training$classe, p=0.7, list=FALSE)
trainingSet <- training[inTrain,]
ValidationSet <- training[-inTrain,]
```

Model Development

Two kinds of models are evaluated: Random Forest and Naive Bayes. In order to reduce the out of sample errors, 10-fold cross validation is used when these two models are trained.

```
# k-fold Cross Validation
train_control <- trainControl(method="cv", number=10)
# Random forest model
model_rf <- train(classe ~ ., data=trainingSet, method="rf", trControl=train_control)
```

```
## Loading required package: randomForest
```

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

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

```
##
## Attaching package: 'randomForest'
```

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

```
# Naive Bayes model
model_nb <- train(classe ~ ., data=trainingSet, method="nb", trControl=train_control)
```

```
## Loading required package: klaR
```

```
## Loading required package: MASS
```

The performance of these models are estimated with the validation data set.

```
predict_rf <- predict(model_rf, ValidationSet)
cm_rf <- confusionMatrix(ValidationSet$classe, predict_rf)
predict_nb <- predict(model_nb, ValidationSet)
cm_nb <- confusionMatrix(ValidationSet$classe, predict_nb)
```

As we can see, Random Forest model performed better than the Naive Bayes Model with an accuracy 0.9928632 of Random Forest and 0.7276126 of Naive Bayes Model. Thus, Random Forests model is choosen for the final model.

Typically, the out of sample error is expected to be larger than the in sample error since it is constituted by the new observations not belonging to the training sample. However, the final Random Forest Model is validated with the 10 fold cross validation which evaluates the model on the independent sets, so here the out of sample error could be estimated as 0.0071368.

Finally, using the Random Forest Model, the prediction on the testing data set is made.

```
predict_test <- predict(model_rf, testing)
predict_test
```

```
##  [1] 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
```

Conclusion

In this study, data of personal activity are analyzed and models are developed to predict the class of body movements. Overall, the Random Forest Model provided an outstanding performance and thus be selected as the final model. Consequently, the model correctly predicted 20/20 measurements of the test set in this case.

References

Human Activity Recognition, http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har)
Practical Machine Learning, https://www.coursera.org/learn/practical-machine-learning/home/welcome (https://www.coursera.org/learn/practical-machine-learning/home/welcome)