Unit 8 Week 4 Project Assignments

Weidong

1/27/2020

1. Introduction

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. 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.

In this project, the goal is to use a weight lifting exercise data from Razor inertial measurement units (IMU) worn by 6 participants to develop a prediction model to predict how well one performs weighting lifting.

2. Data

The training data for this projet is available from here: Training Data (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv).

The data set contains data records from six young healthy participants. Each participant worn four Razor inertial measurement units (IMU) at different locations: belt, arm, forearm, and dumbbell. Each IMU provided 9 degrees of freedom measurements (total 36 measurements), including three-axes acceleration, gyroscope and magnetometer data at a joint sampling rate of 45 Hz.

The six participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E). Class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes. Participants were supervised by an experienced weight lifter to make sure the execution complied to the manner they were supposed to simulate. The Class varialbe in the data set is “classe”.

The data set includes 19662 observations of 160 variables. Besides the 36 direct measurements and the Class variable “classe”, the data set contains additional derived variables or features and other information. The paper (http://groupware.les.inf.puc-rio.br/public/papers/2013.Velloso.QAR-WLE.pdf) provides further information.

In this project, we will use the 36 direct measurements from the IMUs as our predictors and the Class variable “classe” as the resposne from the original training data set. The resulted new training set is a clean data set and there is no need for data cleaning or preprocessing.

Using the new 37-feature training data set, we will develop a random forests prediction model. Then we will apply the model to a test data set, which is available here: test data (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

2.1 Create the training data set

```
## The original training dataset and testing data set were saved to
./data/ folder.
## Read in the original training data set:
training <- read.csv("./data/training.csv")

## extract the 36 measurements by the four sensors
varnames <- names(training)
gyros <- varnames[grep("^gyros", varnames)] ## 12 gyroscope measurem
ents
acc <- varnames[grep("^accel", varnames)]   ## 12 acceleration measu
rements
mag <-varnames[grep("^magnet", varnames)]   ## 12 magnetometer measu
rements

## extract the training data set from the original training set
training <- training[, c(gyros, acc, mag, "classe")]
```

2.2 Create the testing data set

```
# load the original testing set
testing <- read.csv("./data/testing.csv")

# create the testing set with the 37 variables
testing <- testing[, c(gyros, acc, mag, "problem_id")]
```

3. Model Fit

Our problem is a classification problem. We decide to fit a random forests prediction model mainly because 1) the random forests method provides good performance in general and 2) we do not assume models for the data and do not need additional preprocessing.

The following code create a random forests model. The testing error rates will be estimated with cross-validation method, which is built-in the function. The default number of the trees is ntree = 500. ntree is set as 100 to accommodate the capacity of my computer and the estimated errors are acceptably low as we will see below. (With ntree = 500, my computer seems to run forever.)

```
library(caret)
```

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

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

```
library(randomForest)
```

```
## randomForest 4.6-14
```

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

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

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

```
set.seed(1)
fitControl <- trainControl(method="cv")
modFit <- train(classe ~., data = training, method="rf",
                trControl= fitControl,
                ntree = 100,
                verbose=F)
```

3.1 Model Information

```
modFit
```

```
## Random Forest
##
## 19622 samples
##    36 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 17658, 17660, 17660, 17660, 17659, 17661
, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9907752  0.9883293
##   19    0.9878193  0.9845898
##   36    0.9838947  0.9796227
##
## Accuracy was used to select the optimal model using the largest v
alue.
## The final value used for the model was mtry = 2.
```

3.2 Summary of the final model performance

```
print(modFit$finalModel)
```

```
##
## Call:
##  randomForest(x = x, y = y, ntree = 100, mtry = param$mtry, verbo
se = ..2)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 1.05%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 5569    4    0    6    1 0.001971326
## B   37 3737   23    0    0 0.015801949
## C    3   30 3385    4    0 0.010812390
## D    7    0   80 3125    4 0.028296020
## E    0    1    2    4 3600 0.001940671
```

As seen from the summary information, the CV estimated error rates - overall and class error rates - are reasonably low.

4. Prediction

```
pred <- predict(modFit, testing)
```

The testing data contains 20 observations. We apply the model from Section 3 and have -

B, A, B, A, A, E, D, B, A, A, B, C, B, A, E, E, A, B, B, B

From the quiz results, the predictions agree with the classifications of the 20 test cases.