Practical Machine Learning Course Project

Sam H.

Sunday, February 14, 2016

This project examines a weightlifting dataset, which contains various measurements for participants who complete a dumbbell exercise correctly (Class A) or incorrectly (Classes B to E). The prediction algorithm uses the random forest method using 48 variables. The algorithm used all variables in the dataset that did not have null or NA values for most of the samples. The random forest method was chosen because we are estimating a categorical variable and the random forest method blends several decision trees together, creating a better predictor than a single decision tree.

The training set from the provided spreadsheet was split into two groups: a training set (70% of the data) and a test set (30% of the data). Doing this allows us to calculate the accuracy. The following code shows the file that we load into R and how to create the training and test set from the “pml-training.csv” file.

```
training <- read.csv("pml-training.csv")

library(caret)
set.seed(1000)
inTrain <- createDataPartition(y = training$classe, p = 0.7, list = FALSE)
mytrain <- training[inTrain, ]
mytest <- training[-inTrain, ]
```

The random forest method was run using this code:

```
modFit <- train(classe ~ roll_belt + pitch_belt + yaw_belt + total_accel_belt
                    gyros_belt_x + gyros_belt_y + gyros_belt_z +
                    accel_belt_x + accel_belt_y + accel_belt_z +
                    magnet_belt_x + magnet_belt_y + magnet_belt_z +
                    roll_arm + pitch_arm + yaw_arm + total_accel_arm +
                    gyros_arm_x + gyros_arm_y + gyros_arm_z +
                    accel_arm_x + accel_arm_y + accel_arm_z +
                    roll_dumbbell + pitch_dumbbell + yaw_dumbbell +
                    gyros_dumbbell_x + gyros_dumbbell_y + gyros_dumbbell_z +
                    accel_dumbbell_x + accel_dumbbell_y + accel_dumbbell_z +
                    magnet_dumbbell_x + magnet_dumbbell_y + magnet_dumbbell_z
                    roll_forearm  + pitch_forearm + yaw_forearm + total_accel
                    gyros_forearm_x + gyros_forearm_y + gyros_forearm_z +
                    accel_forearm_x + accel_forearm_y + accel_forearm_z +
                    magnet_forearm_x + magnet_forearm_y + magnet_forearm_z,
                data = mytrain,
                method = "rf")
```

The accuracy and the confusion matrix are the following:

```
pred <- predict(modFit, mytest)
mytest$predRight <- pred == mytest$classe
```

```
length(mytest[mytest$predRight, 1])/nrow(mytest)
```

```
## [1] 0.9943925
```

```
table(pred, mytest$classe)
```

```
##
## pred    A    B    C    D    E
##    A 1668    3    0    0    0
##    B    3 1135    4    1    1
##    C    2    1 1020    6    1
##    D    0    0    2  953    4
##    E    1    0    0    4 1076
```

The relative number and percentage of the different classes in the “pml-training.csv” data file are shown below:

```
summary(training$classe)
```

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

```
summary(training$classe)/nrow(training)
```

```
##         A         B         C         D         E
## 0.2843747 0.1935073 0.1743961 0.1638977 0.1838243
```

The most frequent class is Class A with 28.4% of the observations. The accuracy obtained using the random forest is much better than 28.4%, meaning that the random forest algorithm does a better job at predicting the classes than random guessing.