The Course Project in Practical Machine Learning

Atsuko Yamamoto

April 26, 2015

Executive Summary

This is the Course Project for the class “Practical Machine Learning”. In this report, I predict the manner using the exercise data from accelerometers on the belt, forearm, arm and dumbell of 6 participants. I build the prediction model to predict the objective variable “classe”.

Result

I use caret with random forest as my model with 5 fold cross validation. The model is mtry : 27, accuracy : 0.99, OOB estimate of error rate is less than 1%.

Predictions of testing: B A B A A E D B A A B C B A E E A B B B

Data preprocessing

```
# loading data
temp <- tempfile()
urltrain="https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
urltest="https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
download.file(urltrain,temp,"curl")
pmltrain <- read.csv(temp)
download.file(urltest,temp,"curl")
pmltest <- read.csv(temp)
dim(pmltrain);dim(pmltest)
```

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

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

```
# summary(pmltrain)
```

There are a lot of missing values in csvfiles. I get rid of the clumns that are independent of exercizes and mostly NA.

```
# Cleaning the data
pmldata <- pmltrain[,-1:-7]
naCnt <- apply(pmldata,2,function(x) {sum(is.na(x)|x=="")})
pmldata <- pmldata[,which(naCnt < 19216)]
dim(pmldata)
```

```
## [1] 19622    53
```

Data splitting

Devide the pmldata between training and testing.

```
library(caret)
```

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

```
set.seed(1000)
inTrain <- createDataPartition(y=pmldata$classe, p=0.6, list=FALSE)
training <- pmldata[inTrain,]
testing <- pmldata[-inTrain,]
dim(training); dim(testing)
```

```
## [1] 11776    53
```

```
## [1] 7846   53
```

Training

I use caret with random forest as my model with 5 fold cross validation. Because random forests are usually one of the top performing algorithms along with boosting in any prediction contests.

```
rfmodel <- train(classe ~ .,method="rf", data=training,
                 trControl=trainControl(method="cv", number=5),
                 prox=TRUE, allowParallel=TRUE)
```

```
## Loading required package: randomForest
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
rfmodel
```

```
## Random Forest
##
## 11776 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
##
## Summary of sample sizes: 9420, 9422, 9420, 9421, 9421
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa      Accuracy SD  Kappa SD
##    2    0.9883664  0.9852807  0.004304795  0.005450606
##   27    0.9886217  0.9856048  0.003928684  0.004973259
##   52    0.9857344  0.9819537  0.004385470  0.005550943
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 27.
```

```
rfmodel$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry, proximity = TRUE,      allowParallel = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 27
##
##         OOB estimate of  error rate: 0.84%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3341    4    1    0    2 0.002090800
## B   17 2254    8    0    0 0.010969724
## C    0   13 2033    8    0 0.010223953
## D    0    3   27 1898    2 0.016580311
## E    0    2    5    7 2151 0.006466513
```

```
pred <- predict(rfmodel, testing)
table(pred, testing$classe)
```

```
##
## pred    A    B    C    D    E
##    A 2230   23    0    0    0
##    B    2 1493   11    1    2
##    C    0    2 1352   24    2
##    D    0    0    5 1261    5
##    E    0    0    0    0 1433
```

The model is mtry = 27, and accuracy is 0.99. OOB estimate of error rate is less than 1%. That is a good model.

Predictions of testing

I use my prediction model to predict 20 different test cases.

```
answers <- predict(rfmodel, pmltest)
answers
```

```
##  [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
```