Practical Machine Learning - Peer Assessments

Alex (Oleksiy) Varfolomiyev

Executive Summary

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. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

The goal of this HAR project is to use the accelerometers data of 6 participants to predict the "classe" (A, B, C, D or E) of the excersise, labeling how correct was the excercise executed.

Since we have a large number of predictors we use the random forests algorithm to train the model. The developed model gives 0.9924% accuracy on the validation set and was able to make correct prediction for 20 test examples.

Read Data

```
  require(ggplot2)
```

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

```
  require(randomForest)
```

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

```
  require(caret)
```

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

```
  require(lattice)

    fileUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
    fileName <- "./data/activity.csv"
    download.file(fileUrl, destfile = fileName, method = "curl")

    dateDownloaded <- date()
    dateDownloaded
```

```
  if(!exists("datTrainInit")){
    fileName <- "./data/activity.csv"
    datTrainInit <- read.csv(fileName, header = T, sep = ',', na.strings=c("NA",""))
  }
    datTrain <- datTrainInit

    fileUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
    fileName <- "./data/test.csv"
    download.file(fileUrl, destfile = fileName, method = "curl")

    dateDownloaded <- date()
    dateDownloaded

  if(!exists("datTestInit")){
    fileName <- "./data/test.csv"
    datTestInit <- read.csv(fileName, header = T, sep = ',', na.strings=c("NA",""))
  }
    datTest <- datTestInit
```

Process Data

```
rmCol = 1:7
datTrain <- datTrain[ , -rmCol]

n <- nrow(datTrain)
trainColLogic <- colSums(is.na(datTrain)) < 0.7*n
datTrain <- datTrain[ , trainColLogic]

datTest <- datTest[ , -rmCol]

n <- nrow(datTest)
testColLogic <- colSums(is.na(datTest)) < 0.7*n
datTest <- datTest[ , testColLogic]
```

Random Forest Model Fit

```
set.seed(5)

inTrain <- createDataPartition(y = datTrain$classe, p = 0.60, list = FALSE)
datValid  <- datTrain[-inTrain, ]
datTrain  <- datTrain[inTrain, ]

set.seed(7)
if(!exists("fit")) fit <- randomForest(classe ~ . , datTrain, importance = T)

fit
```

```
##
```

```
## Call:
##  randomForest(formula = classe ~ ., data = datTrain, importance = T)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.68%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3339    6    2    0    1 0.002688172
## B   12 2259    8    0    0 0.008775779
## C    0   19 2033    2    0 0.010223953
## D    0    0   19 1909    2 0.010880829
## E    0    0    3    6 2156 0.004157044
```

Model Validation

```
predictionsDataValid <- predict(fit, newdata = datValid)

confusionMatr <- confusionMatrix(predictionsDataValid, datValid$classe)
confusionMatr
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    8    0    0    0
##          B    0 1506   19    0    0
##          C    0    4 1347   14    0
##          D    0    0    2 1272   13
##          E    0    0    0    0 1429
##
## Overall Statistics
##
##                Accuracy : 0.9924
##                  95% CI : (0.9902, 0.9942)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9903
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9921   0.9846   0.9891   0.9910
## Specificity            0.9986   0.9970   0.9972   0.9977   1.0000
## Pos Pred Value         0.9964   0.9875   0.9868   0.9883   1.0000
## Neg Pred Value         1.0000   0.9981   0.9968   0.9979   0.9980
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1919   0.1717   0.1621   0.1821
```

```
## Detection Prevalence   0.2855   0.1944   0.1740   0.1640   0.1821
## Balanced Accuracy      0.9993   0.9945   0.9909   0.9934   0.9955
```

Test Model

```
answers <- predict(fit, newdata = datTest)
answers
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  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
```

```
pml_write_files = function(x){
  n = length(x)
  if(!file.exists("test")) dir.create("test")
  for(i in 1:n){
    filename = paste0("problem_id_", i, ".txt")
    filename = paste0("./test/", filename)
    write.table(x[i], file = filename, quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}
pml_write_files(answers)
```

Conclusion

Developed model accurately predicts the "classe" of 20 test examples. Model accuracy on the validation set is 0.9924%. It is remarcable how accurate the model is considering that we only had 6 participants measurements.