Practical Machine Learning Project

duncan

27 February 2016

Summary

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit 6 participants collected data from accelerometers on the belt, forearm, arm, and dumbell. They were asked to perform 10 repetitions of barbell lifts in 5 different ways (“classe” A to E) (ref [1])

“classe”" A corresponds to the specified (correct) way, while the other 4 correspond to common mistakes.

The goals are:

- to develop a model to predict the “classe” of each exercise from the accelerator values
- to use cross validation and estimate the out-of-sample error
- to use the model to predict the classe of 20 different test cases.

This report was produced using Windows 8.1, RStudio (32 bit), Knit HTML and git.

Examination of the data

The training and test data files were downloaded (ref [2]) and are assumed to be in local storage as pml-training.csv and pml-testing.csv.

Read in data and extract non-NA accelerator columns

```
library(caret)
```

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

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

```
library(MASS)
```

```
Xtrain <- "./pml-training.csv"
trainData  <- read.csv(Xtrain, header=TRUE)

Xtest <- "./pml-testing.csv"
testData  <- read.csv(Xtest, header=TRUE)

## Get variable names containng "accel"
ind <- which(grepl("accel", names(trainData)))
trainAcc <- trainData[,ind]

# remove those starting with var and total
var <- which(grepl("^var", names(trainAcc))  )
trainAcc <- trainAcc[,-var]
tot <- which(grepl("^total", names(trainAcc))  )
trainAcc <- trainAcc[,-tot]

ind <- which(grepl("accel", names(testData)))
testAcc <- testData[,ind]
var <- which(grepl("^var", names(testAcc))  )
testAcc <- testAcc[,-var]
tot <- which(grepl("^total", names(testAcc))  )
testAcc <- testAcc[,-tot]

# add outcome classe column to trainAcc
trainAcc[,"classe"] <- trainData[,"classe"]
set.seed(15651)
```

```
names(trainAcc)  # Here are the variables used
```

```
##  [1] "accel_belt_x"     "accel_belt_y"     "accel_belt_z"
##  [4] "accel_arm_x"      "accel_arm_y"      "accel_arm_z"
##  [7] "accel_dumbbell_x" "accel_dumbbell_y" "accel_dumbbell_z"
## [10] "accel_forearm_x"  "accel_forearm_y"  "accel_forearm_z"
## [13] "classe"
```

```
# A check was made of highly correlated variables (cor>.9)
m <- abs(cor(trainAcc[,-13]))   # 13 is classe
diag(m) <- 0
w <- which(m>0.8,arr.ind=T)
abs(cor(trainAcc[,c(2,3)]))
```

```
##              accel_belt_y accel_belt_z
## accel_belt_y    1.0000000    0.9333854
## accel_belt_z    0.9333854    1.0000000
```

```
#   2 were found and one was removed to avoid bias
trainAcc <- trainAcc[,-3]
testAcc <- testAcc[,-3]
```

Development of the prediction model

Model using randomForest package

```
# Trial of randomForest
library(randomForest)
```

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

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

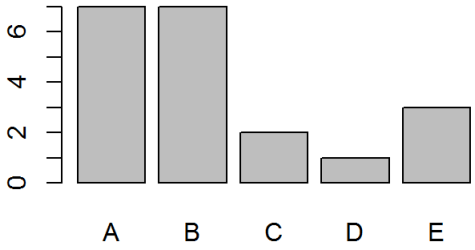
```
##
## Attaching package: 'randomForest'
```

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

```
set.seed(15651)

modFit <- randomForest(classe ~ ., data=trainAcc)

# The predictions for the test data
prf <- predict(modFit,testAcc)
plot(prf)
```



Resampling, Accuracy and Test Predictions

```
# Resampling, Accuracy, and confusion matrix

modFit
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = trainAcc)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 5.55%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 5386   29   75   82    8  0.03476703
## B  150 3490   80   31   46  0.08085331
## C   53   76 3252   27   14  0.04967855
## D   68   14  127 2971   36  0.07618159
## E   17   69   38   49 3434  0.04796230
```

```
# The test predictions
prf
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  B  A  C  A  A  E  D  B  A  A  B  C  B  A  E  E  A  B  B  B
## Levels: A B C D E
```

Investigation of other models

The data was checked for correlated variables. Initially 3 highly correlated variables were found (cor>0.9) which would bias the predictions so two were removed.

A randomForest model was initially tried with caret/train/rf but was found to be very slow and crashed or ran out of memory.

lda and rpart were also trialed and gave different results. Finally a randomForest model was selected and gave good results on the test data.

Linear discriminant analysis (lda) finds linear combinations of the original variables (as in pca) that identify the “classe” groups. It performed poorly on the test data. The code for lda follows:

```
modlda <- train(classe~., data=trainAcc, method="lda")
plda <- predict(modlda,testAcc)
```

cross validation

Separate Training and testing data were provided. Highly correlated variables were reduced to one. randomForest created multiple trees with an OOB error rate of 5.55%.

Rational for choices made

The problem seemed to relate to accelerations of sensors so the variables were pruned to these. No problems were identified with NAs but some variables were highly correlated so were further pruned.

The problem was identified as a classification with 5 outcomes A - E so binary models were discounted.
Tree methods were considered but caret/rf gave problems. Later randomForest was chosen.

Initial experiments identified caret/lda as being simple to set up but it gave low accuracy. It was replaced by randomForest. rpart was also trialed.

Conclusion

Both lda and randomForest methods shown gave different results on the test data. lda had a low accuracy. (Choosing the model based on the results of the test data is clearly overfitting!) Both were easy to use at an elementary level but reading the documentation and selecting options for each proved difficult.

The use of caret/rf appears unsuited for student courses due to time and memory problems but randomForest was ok.

Reference


```
[1] http://groupware.les.inf.puc-rio.br/har (Source of data - Weight Lifting Exercise Dataset).

[2] The training and test data were downloaded from:
  https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv
  https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv
```