Predicting classe of exercise based on data from accelerometers on the belt, forearm, arm, and dumbell

by: MonicaPH Date: August, 2014

Background and data

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 (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Objective

Predict the manner in which they did the exercise. This is the “classe” variable in the training set.

Data Processing

Firts it loads the required libraries. Then instructions are included to download, unzip and read the data:

```
#Libraries
library(ggplot2)
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.1.1
```

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

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.1.1
```

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

```
#library(doMC);

#registerDoMC(cores = 4)
Sys.setlocale("LC_TIME", "English")       # Set language to engish
```

```
## [1] "English_United States.1252"
```

```
set.seed(555)

#Downloading
fileurl<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
if(!file.exists("pml-training.csv")){
      download.file(fileurl, destfile="pml-training.csv")
}
fileurl<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
if(!file.exists("pml-testing.csv")){
      download.file(fileurl, destfile="pml-testing.csv")
}

#Reading
train <- read.csv("pml-training.csv")
test <- read.csv("pml-testing.csv",header=TRUE)

#Splitting train for cross validation
trainIndex = createDataPartition(y=train$classe, p = 0.60,list=FALSE)
trainCV = train[trainIndex,]
testCV = train[-trainIndex,]

# Summary of the Data
str(trainCV)
```

```
## 'data.frame':    11776 obs. of  160 variables:
##  $ X                       : int  2 3 4 5 6 7 8 9 12 15 ...
##  $ user_name               : Factor w/ 6 levels "adelmo","carlitos",..: 2 2 2 2 2 2 2 2 2
 2 ...
##  $ raw_timestamp_part_1    : int  1323084231 1323084231 1323084232 1323084232 1323084232
1323084232 1323084232 1323084232 1323084232 1323084232 ...
##  $ raw_timestamp_part_2    : int  808298 820366 120339 196328 304277 368296 440390 484323
 528316 604281 ...
##  $ cvtd_timestamp          : Factor w/ 20 levels "02/12/2011 13:32",..: 9 9 9 9 9 9 9 9 9
 9 ...
##  $ new_window              : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ num_window              : int  11 11 12 12 12 12 12 12 12 12 ...
##  $ roll_belt               : num  1.41 1.42 1.48 1.48 1.45 1.42 1.42 1.43 1.43 1.45 ...
##  $ pitch_belt              : num  8.07 8.07 8.05 8.07 8.06 8.09 8.13 8.16 8.18 8.2 ...
##  $ yaw_belt                : num  -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -
94.4 ...
##  $ total_accel_belt        : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ kurtosis_roll_belt      : Factor w/ 397 levels "","-0.016850",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ kurtosis_picth_belt     : Factor w/ 317 levels "","-0.021887",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ kurtosis_yaw_belt       : Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1 1 1 ...
##  $ skewness_roll_belt      : Factor w/ 395 levels "","-0.003095",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ skewness_roll_belt.1    : Factor w/ 338 levels "","-0.005928",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ skewness_yaw_belt       : Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1 1 1 ...
##  $ max_roll_belt           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_picth_belt          : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_yaw_belt            : Factor w/ 68 levels "","-0.1","-0.2",..: 1 1 1 1 1 1 1 1 1 1
 ...
##  $ min_roll_belt           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_pitch_belt          : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_yaw_belt            : Factor w/ 68 levels "","-0.1","-0.2",..: 1 1 1 1 1 1 1 1 1 1
 ...
##  $ amplitude_roll_belt     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_pitch_belt    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_yaw_belt      : Factor w/ 4 levels "","#DIV/0!","0.00",..: 1 1 1 1 1 1 1 1 1
 1 ...
##  $ var_total_accel_belt    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_roll_belt           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_roll_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_roll_belt           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_pitch_belt          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_pitch_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_pitch_belt          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_yaw_belt            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_yaw_belt         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_yaw_belt            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gyros_belt_x            : num  0.02 0 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0 ...
##  $ gyros_belt_y            : num  0 0 0 0.02 0 0 0 0 0 0 ...
##  $ gyros_belt_z            : num  -0.02 -0.02 -0.03 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 0
 ...
##  $ accel_belt_x            : int  -22 -20 -22 -21 -21 -22 -22 -20 -22 -21 ...
##  $ accel_belt_y            : int  4 5 3 2 4 3 4 2 2 2 ...
##  $ accel_belt_z            : int  22 23 21 24 21 21 21 24 23 22 ...
##  $ magnet_belt_x           : int  -7 -2 -6 -6 0 -4 -2 1 -2 -1 ...
##  $ magnet_belt_y           : int  608 600 604 600 603 599 603 602 602 597 ...
##  $ magnet_belt_z           : int  -311 -305 -310 -302 -312 -311 -313 -312 -319 -310 ...
##  $ roll_arm                : num  -128 -128 -128 -128 -128 -128 -128 -128 -128 -129 ...
##  $ pitch_arm               : num  22.5 22.5 22.1 22.1 22 21.9 21.8 21.7 21.5 21.4 ...
##  $ yaw_arm                 : num  -161 -161 -161 -161 -161 -161 -161 -161 -161 -161 ...
##  $ total_accel_arm         : int  34 34 34 34 34 34 34 34 34 34 ...
##  $ var_accel_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_roll_arm            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_roll_arm         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_roll_arm            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_pitch_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_pitch_arm        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_pitch_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_yaw_arm             : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_yaw_arm          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_yaw_arm             : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gyros_arm_x             : num  0.02 0.02 0.02 0 0.02 0 0.02 0.02 0.02 0.02 ...
##  $ gyros_arm_y             : num  -0.02 -0.02 -0.03 -0.03 -0.03 -0.03 -0.02 -0.03 -0.03 0
 ...
##  $ gyros_arm_z             : num  -0.02 -0.02 0.02 0 0 0 0 -0.02 0 -0.03 ...
##  $ accel_arm_x             : int  -290 -289 -289 -289 -289 -289 -289 -288 -288 -289 ...
##  $ accel_arm_y             : int  110 110 111 111 111 111 111 109 111 111 ...
##  $ accel_arm_z             : int  -125 -126 -123 -123 -122 -125 -124 -122 -123 -124 ...
##  $ magnet_arm_x            : int  -369 -368 -372 -374 -369 -373 -372 -369 -363 -374 ...
##  $ magnet_arm_y            : int  337 344 344 337 342 336 338 341 343 342 ...
##  $ magnet_arm_z            : int  513 513 512 506 513 509 510 518 520 510 ...
##  $ kurtosis_roll_arm       : Factor w/ 330 levels "","-0.02438",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ kurtosis_picth_arm      : Factor w/ 328 levels "","-0.00484",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ kurtosis_yaw_arm        : Factor w/ 395 levels "","-0.01548",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ skewness_roll_arm       : Factor w/ 331 levels "","-0.00051",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ skewness_pitch_arm      : Factor w/ 328 levels "","-0.00184",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ skewness_yaw_arm        : Factor w/ 395 levels "","-0.00311",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ max_roll_arm            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_picth_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_yaw_arm             : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_roll_arm            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_pitch_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_yaw_arm             : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_roll_arm      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_pitch_arm     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_yaw_arm       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ roll_dumbbell           : num  13.1 12.9 13.4 13.4 13.4 ...
##  $ pitch_dumbbell          : num  -70.6 -70.3 -70.4 -70.4 -70.8 ...
##  $ yaw_dumbbell            : num  -84.7 -85.1 -84.9 -84.9 -84.5 ...
##  $ kurtosis_roll_dumbbell  : Factor w/ 398 levels "","-0.0035","-0.0073",..: 1 1 1 1 1 1
1 1 1 1 ...
##  $ kurtosis_picth_dumbbell : Factor w/ 401 levels "","-0.0163","-0.0233",..: 1 1 1 1 1 1
1 1 1 1 ...
##  $ kurtosis_yaw_dumbbell   : Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1 1 1 ...
##  $ skewness_roll_dumbbell  : Factor w/ 401 levels "","-0.0082","-0.0096",..: 1 1 1 1 1 1
1 1 1 1 ...
##  $ skewness_pitch_dumbbell : Factor w/ 402 levels "","-0.0053","-0.0084",..: 1 1 1 1 1 1
1 1 1 1 ...
##  $ skewness_yaw_dumbbell   : Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1 1 1 ...
##  $ max_roll_dumbbell       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_picth_dumbbell      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_yaw_dumbbell        : Factor w/ 73 levels "","-0.1","-0.2",..: 1 1 1 1 1 1 1 1 1 1
 ...
##  $ min_roll_dumbbell       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_pitch_dumbbell      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_yaw_dumbbell        : Factor w/ 73 levels "","-0.1","-0.2",..: 1 1 1 1 1 1 1 1 1 1
 ...
##  $ amplitude_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA ...
##   [list output truncated]
```

Some of the measurements contain lots of missing values. For this prediction, only measurements without NA’s and without empty fields will be considered.Also, sample, user and timestamp data have been removed.

```
trainCV <- trainCV[colSums(is.na(trainCV)) == 0]
trainCV <- trainCV[colSums(trainCV == "") == 0]
trainCV <- trainCV[,8:60]
str(trainCV)
```

```
## 'data.frame':    11776 obs. of  53 variables:
##  $ roll_belt           : num  1.41 1.42 1.48 1.48 1.45 1.42 1.42 1.43 1.43 1.45 ...
##  $ pitch_belt          : num  8.07 8.07 8.05 8.07 8.06 8.09 8.13 8.16 8.18 8.2 ...
##  $ yaw_belt            : num  -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4
 ...
##  $ total_accel_belt    : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ gyros_belt_x        : num  0.02 0 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0 ...
##  $ gyros_belt_y        : num  0 0 0 0.02 0 0 0 0 0 0 ...
##  $ gyros_belt_z        : num  -0.02 -0.02 -0.03 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 0 ...
##  $ accel_belt_x        : int  -22 -20 -22 -21 -21 -22 -22 -20 -22 -21 ...
##  $ accel_belt_y        : int  4 5 3 2 4 3 4 2 2 2 ...
##  $ accel_belt_z        : int  22 23 21 24 21 21 21 24 23 22 ...
##  $ magnet_belt_x       : int  -7 -2 -6 -6 0 -4 -2 1 -2 -1 ...
##  $ magnet_belt_y       : int  608 600 604 600 603 599 603 602 602 597 ...
##  $ magnet_belt_z       : int  -311 -305 -310 -302 -312 -311 -313 -312 -319 -310 ...
##  $ roll_arm            : num  -128 -128 -128 -128 -128 -128 -128 -128 -128 -129 ...
##  $ pitch_arm           : num  22.5 22.5 22.1 22.1 22 21.9 21.8 21.7 21.5 21.4 ...
##  $ yaw_arm             : num  -161 -161 -161 -161 -161 -161 -161 -161 -161 -161 ...
##  $ total_accel_arm     : int  34 34 34 34 34 34 34 34 34 34 ...
##  $ gyros_arm_x         : num  0.02 0.02 0.02 0 0.02 0 0.02 0.02 0.02 0.02 ...
##  $ gyros_arm_y         : num  -0.02 -0.02 -0.03 -0.03 -0.03 -0.03 -0.02 -0.03 -0.03 0 ...
##  $ gyros_arm_z         : num  -0.02 -0.02 0.02 0 0 0 0 -0.02 0 -0.03 ...
##  $ accel_arm_x         : int  -290 -289 -289 -289 -289 -289 -289 -288 -288 -289 ...
##  $ accel_arm_y         : int  110 110 111 111 111 111 111 109 111 111 ...
##  $ accel_arm_z         : int  -125 -126 -123 -123 -122 -125 -124 -122 -123 -124 ...
##  $ magnet_arm_x        : int  -369 -368 -372 -374 -369 -373 -372 -369 -363 -374 ...
##  $ magnet_arm_y        : int  337 344 344 337 342 336 338 341 343 342 ...
##  $ magnet_arm_z        : int  513 513 512 506 513 509 510 518 520 510 ...
##  $ roll_dumbbell       : num  13.1 12.9 13.4 13.4 13.4 ...
##  $ pitch_dumbbell      : num  -70.6 -70.3 -70.4 -70.4 -70.8 ...
##  $ yaw_dumbbell        : num  -84.7 -85.1 -84.9 -84.9 -84.5 ...
##  $ total_accel_dumbbell: int  37 37 37 37 37 37 37 37 37 37 ...
##  $ gyros_dumbbell_x    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ gyros_dumbbell_y    : num  -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02
 ...
##  $ gyros_dumbbell_z    : num  0 0 -0.02 0 0 0 0 0 0 0 ...
##  $ accel_dumbbell_x    : int  -233 -232 -232 -233 -234 -232 -234 -232 -233 -234 ...
##  $ accel_dumbbell_y    : int  47 46 48 48 48 47 46 47 47 47 ...
##  $ accel_dumbbell_z    : int  -269 -270 -269 -270 -269 -270 -272 -269 -270 -270 ...
##  $ magnet_dumbbell_x   : int  -555 -561 -552 -554 -558 -551 -555 -549 -554 -554 ...
##  $ magnet_dumbbell_y   : int  296 298 303 292 294 295 300 292 291 294 ...
##  $ magnet_dumbbell_z   : num  -64 -63 -60 -68 -66 -70 -74 -65 -65 -63 ...
##  $ roll_forearm        : num  28.3 28.3 28.1 28 27.9 27.9 27.8 27.7 27.5 27.2 ...
##  $ pitch_forearm       : num  -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.8 -63.8 -63.8 -63.9
 ...
##  $ yaw_forearm         : num  -153 -152 -152 -152 -152 -152 -152 -152 -152 -151 ...
##  $ total_accel_forearm : int  36 36 36 36 36 36 36 36 36 36 ...
##  $ gyros_forearm_x     : num  0.02 0.03 0.02 0.02 0.02 0.02 0.02 0.03 0.02 0 ...
##  $ gyros_forearm_y     : num  0 -0.02 -0.02 0 -0.02 0 -0.02 0 0.02 -0.02 ...
##  $ gyros_forearm_z     : num  -0.02 0 0 -0.02 -0.03 -0.02 0 -0.02 -0.03 -0.02 ...
##  $ accel_forearm_x     : int  192 196 189 189 193 195 193 193 191 192 ...
##  $ accel_forearm_y     : int  203 204 206 206 203 205 205 204 203 201 ...
##  $ accel_forearm_z     : int  -216 -213 -214 -214 -215 -215 -213 -214 -215 -214 ...
##  $ magnet_forearm_x    : int  -18 -18 -16 -17 -9 -18 -9 -16 -11 -16 ...
##  $ magnet_forearm_y    : num  661 658 658 655 660 659 660 653 657 656 ...
##  $ magnet_forearm_z    : num  473 469 469 473 478 470 474 476 478 472 ...
##  $ classe              : Factor w/ 5 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Model fitting

First, create folds for crossvalidation

```
## 4-fold crossvalidation
cvFolds <- trainControl(method = "cv", number = 4, allowParallel = TRUE,verboseIter = TRUE)
```

Fitting a logistic regression model on the data. The sample number (X), the user’s name (user_name), and the timestamps are excluded from the predictors

```
#model <- train(classe ~ ., method="rf",family="binomial",trControl=cvFolds, data=trainCV[,5
0:60],prox=TRUE)
model <- randomForest(classe ~ ., data = trainCV)

# Check summary accuracy
model
```

```
## 
## Call:
##  randomForest(formula = classe ~ ., data = trainCV) 
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
## 
##         OOB estimate of  error rate: 0.75%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3347    1    0    0    0   0.0002987
## B   17 2256    6    0    0   0.0100921
## C    0   18 2034    2    0   0.0097371
## D    1    0   30 1895    4   0.0181347
## E    0    0    4    5 2156   0.0041570
```

Prediction

Testing the model in the crossvalidation test data.

```
# removing the unused features
testCV <- testCV[names(trainCV)]

# applying model
predTestCV<-predict(model,newdata=testCV)
confusionMatrix(predTestCV,testCV$classe)
```

```
## Warning: package 'e1071' was built under R version 3.1.1
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    9    0    0    0
##          B    0 1509   15    0    0
##          C    0    0 1353   14    6
##          D    0    0    0 1272    1
##          E    0    0    0    0 1435
##
## Overall Statistics
##
##                Accuracy : 0.994
##                  95% CI : (0.992, 0.996)
##     No Information Rate : 0.284
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.993
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             1.000    0.994    0.989    0.989    0.995
## Specificity             0.998    0.998    0.997    1.000    1.000
## Pos Pred Value          0.996    0.990    0.985    0.999    1.000
## Neg Pred Value          1.000    0.999    0.998    0.998    0.999
## Prevalence              0.284    0.193    0.174    0.164    0.184
## Detection Rate          0.284    0.192    0.172    0.162    0.183
## Detection Prevalence    0.286    0.194    0.175    0.162    0.183
## Balanced Accuracy       0.999    0.996    0.993    0.994    0.998
```

```
impFeatures <- varImp(model)
```

The balance accyracy to detect each classe is ~99%

Results

Finally, testing the model in the test set:

```
# subsetting
test <- test[, which(names(test) %in% names(trainCV))]
test2 <- cbind(test,testCV[1:20,53])

# applying model
predTest <- predict(model,newdata=test)


# preparing to write data

pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}

#pml_write_files(predTest)
```