Qualitative Assessment of Exercising

Manas Sharma

March 12, 2016

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, we use data from accelerometers on the belt, forearm, arm, and dumbbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. Given the data from accelerometers, the goal is to predict the class of action (variable `classe`) which can be one of the following:

- exactly according to the specification (A)
- throwing elbows to the front (B)
- lifting the dumbbell only halfway (C)
- lowering the dumbbell only halfway (D)
- throwing the hips to the front (E)

More information is available from the website here (see the section on the Weight Lifting Exercise Dataset).

Data

The training and test data for this project are available here and here, respectively.

```
library(RCurl)
train.url <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
test.url <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
train.data <- read.csv(text=getURL(train.url), na.strings=c("", "NA"))
test.data <- read.csv(text=getURL(test.url), na.strings=c("", "NA"))
```

The first column is the Index variable and hence can be omitted from the data set.

```
train.data$X <- NULL
```

Similarly, the user and time information should not have any effect on whether barbell lifts are performed correctly or not.

```
col.rm <- c("user_name", "raw_timestamp_part_1",
                    "raw_timestamp_part_2", "cvtd_timestamp")
```

```
for (col in col.rm) {
    train.data[, col] <- NULL
}
```

Many of the columns in the data set have majority of missing values. Therefore, we will remove these columns (or, features) from our training and test data sets since imputation is not an option.

```
col.NAs <- apply(train.data, 2, function(x) {sum(is.na(x))})
train.data <- train.data[, which(col.NAs == 0)]
```

Some of the variables have near constant values, i.e. almost zero variance. Hence, we can remove these zero variance predictors from our data since they have few unique values relative to the number of observations.

```
library(caret)
nsv <- nearZeroVar(train.data)
train.data <- train.data[-nsv]
test.data <- test.data[-nsv]
```

The final set of predictors used for classification are:

```
names(train.data)
```

```
##  [1] "num_window"           "roll_belt"            "pitch_belt"
##  [4] "yaw_belt"             "total_accel_belt"     "gyros_belt_x"
##  [7] "gyros_belt_y"         "gyros_belt_z"         "accel_belt_x"
## [10] "accel_belt_y"         "accel_belt_z"         "magnet_belt_x"
## [13] "magnet_belt_y"        "magnet_belt_z"        "roll_arm"
## [16] "pitch_arm"            "yaw_arm"              "total_accel_arm"
## [19] "gyros_arm_x"          "gyros_arm_y"          "gyros_arm_z"
## [22] "accel_arm_x"          "accel_arm_y"          "accel_arm_z"
## [25] "magnet_arm_x"         "magnet_arm_y"         "magnet_arm_z"
## [28] "roll_dumbbell"        "pitch_dumbbell"       "yaw_dumbbell"
## [31] "total_accel_dumbbell" "gyros_dumbbell_x"     "gyros_dumbbell_y"
## [34] "gyros_dumbbell_z"     "accel_dumbbell_x"     "accel_dumbbell_y"
## [37] "accel_dumbbell_z"     "magnet_dumbbell_x"    "magnet_dumbbell_y"
## [40] "magnet_dumbbell_z"    "roll_forearm"         "pitch_forearm"
## [43] "yaw_forearm"          "total_accel_forearm"  "gyros_forearm_x"
## [46] "gyros_forearm_y"      "gyros_forearm_z"      "accel_forearm_x"
## [49] "accel_forearm_y"      "accel_forearm_z"      "magnet_forearm_x"
## [52] "magnet_forearm_y"     "magnet_forearm_z"     "classe"
```

Model

We will use Random Forest classifier to predict the action class. To measure the accuracy of the model, we will perform a 10-fold cross validation with 80:20 split on each fold, i.e. 80% of the data will be used for training and remaining 20% will be used for testing.

```
library(randomForest)
set.seed(123)
```

```
obs <- c()
preds <- c()
for(i in 1:10) {
    intrain = sample(1:dim(train.data)[1], size=dim(train.data)[1] * 0.8,
replace=F)
    train.cross = train.data[intrain,]
    test.cross = train.data[-intrain,]
    rf <- randomForest(classe ~ ., data=train.cross)
    obs <- c(obs, test.cross$classe)
    preds <- c(preds, predict(rf, test.cross))
}
```

The confusion matrix for predictions on cross validation folds is:

```
conf.mat <- confusionMatrix(table(preds, obs))
conf.mat$table
```

```
##      obs
## preds     1     2     3     4     5
##     1 11223     4     0     0     0
##     2     0  7533     9     0     0
##     3     0     3  6809    29     0
##     4     0     0     1  6435     6
##     5     2     0     0     3  7193
```

The model seems to be classifying well enough, with the accuracy of 99.85%. Finally, let's train the random forest on the entire data set so that the model can be used to predict the class of an action, given a set of activity measurements.

```
library(randomForest)
model <- randomForest(classe ~ ., data=train.data)
```