Qualitative Activity Recognition of Weight Lifting Exercises

Peter Prevos

23 November 2014

Background

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is possible to collect a large amount of data about personal activity. These type of devices are part of the quantified self movement — a group of enthusiasts who regularly take measurements about themselves to improve their health, to find patterns in their behavior, or simple 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, the data from accelerometers on the belt, forearm, arm, and dumbbell of six participants will be used. Participants were asked to perform one set of ten repetitions of the Unilateral Dumbbell Biceps Curl (video) correctly (Class A) and incorrectly in four ways:

- Throwing the elbows to the front (Class B)
- Lifting the dumbbell only halfway (Class C)
- Lowering the dumbbell only halfway (Class D)
- Throwing the hips to the front (Class E)

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

This report partially reproduces the research in Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. *Proceedings of 4th Augmented Human (AH) International Conference in cooperation with ACM SIGCHI* (Augmented Human'13). Stuttgart, Germany: ACM SIGCHI, 2013.

Exploratory Analysis

The training set is read into the `data` variable and factor variables are converted to numbers.

```r
data <- read.csv("pml-training.csv")
data[,7:159] <- apply(data[,7:159], 2, function(x){as.numeric(as.character(x))})
```

The data contains a large number of NA values (61% of the data) because many of the variables contain periodic descriptive statistics of other variables. Independent variables with more than 90% of NA values are removed from the data set. This will not influence the error rate of the prediction model since these are summary statistics that highly correlate with the other data. Non predictive variables (X, user_name, raw_timestamp_part_1, raw_timestamp_part_2, cvtd_timestamp, new_window, num_window) can also be removed.

```r
data <- data[,-1:-7]
count_nas <- apply(data, 2, function(var){
        sum(is.na(var))/length(var)*100
        })
data <- data[-which(count_nas>90)]
```

This leaves 0% of the data with NA values and a data set of 52 independent variables and one dependent variable over a total of 19622 observations.

```
vars <- strsplit(names(data[,-53]), "_")
var1 <- unlist(lapply(vars, function(x){x[1]}))
var2 <- unlist(lapply(vars, function(x){x[2]}))
knitr::kable(table(var1, var2), caption="Overview of independent variables.")
```

	accel	arm	belt	dumbbell	forearm
accel	0	3	3	3	3
gyros	0	3	3	3	3
magnet	0	3	3	3	3
pitch	0	1	1	1	1
roll	0	1	1	1	1
total	4	0	0	0	0
yaw	0	1	1	1	1

Table 1: Overview of independent variables.

This overview shows the four modes of measurement: arms, belt, dumbell and forearm. For each of the four modes, the data consists of three values (x, y and z) for acceleration, gyroscope and magnet values. Furthermore, roll, pitch and yaw for each of the four modes is also available. Lastly the total accelleration of each of the four modes is available. This creates a total of 52 combinations, being the number of independent variables.

Analysis

A training and validation set is created from the provided data. The training set is kept small in order to reduce computational load.

```
library(caret)
set.seed(666)
trainIndex = createDataPartition(data$classe, p=0.1, list=FALSE)
training = data[trainIndex,]
validation = data[-trainIndex,]
```

A Random Forest model is fitted over the training data. In random forests, there is no need for cross-validation to determine an unbiased estimate of the test set error. This is estimated internally as the *Out-Of-Bag* (OOB) error rate. To fine tune the model, the `tuneRF` function is used to determine the optimal number of variables randomly sampled as candidates at each split (m_{try}), The number of variables with the lowest Out-of-Bag error estimate is used in the modeling.

```
library(randomForest)
mtry <- tuneRF(training[,-53], training[,53], stepFactor=1.5, trace=F)
```

```
## -0.1094891 0.05
## 0.04379562 0.05
```

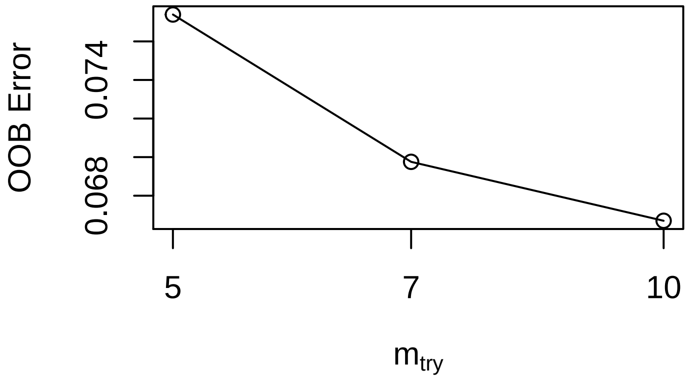

Figure 1: Random forest tuning.

```
mtry_min <- mtry[which.min(mtry)/2]
```

The optimum number of variables to minimize OOB error is 10, which is used to fit the predictive model.

```
fit <- randomForest(classe~., data=training, ntree=100, mtry=mtry_min, prox=T)
```

Results

The model is validated against the validation set.

```
pred <- predict(fit, validation)
accuracy <- confusionMatrix(pred, validation$classe)
```

The selected model achieves an accuracy of 95.1%. The vialised confusion matrix shows the correspondence between prediction and actual excercise classe.

```
par(mar=c(4,2,1,3), xpd=TRUE)
plot(pred, validation$classe, pch=".", col=rainbow(5), xlab="Prediction", ylab="Testing")
legend(1.3,.37, legend=LETTERS[1:5], fill=rainbow(5), box.col=0)
```

Interpretation

Interpretation of random forests is complex since the fitted model cannot be easily understood as it contains 100 individual decision trees. In ther words, it is difficult seeing the trees through the random forest. One way of assesing the model against reality is by determining the importance of the independent variables.

```
par(mar=c(4,4,1,1))
varImpPlot(fit, pch=19, cex=.7, main="")
```

This analysis shows that the roll of the belt (the angle of the belt relative to the ground) is the most important independent variable, this implies that keeping the hips steady contributes greatly to conducting proper unilateral dumbbell biceps curls.

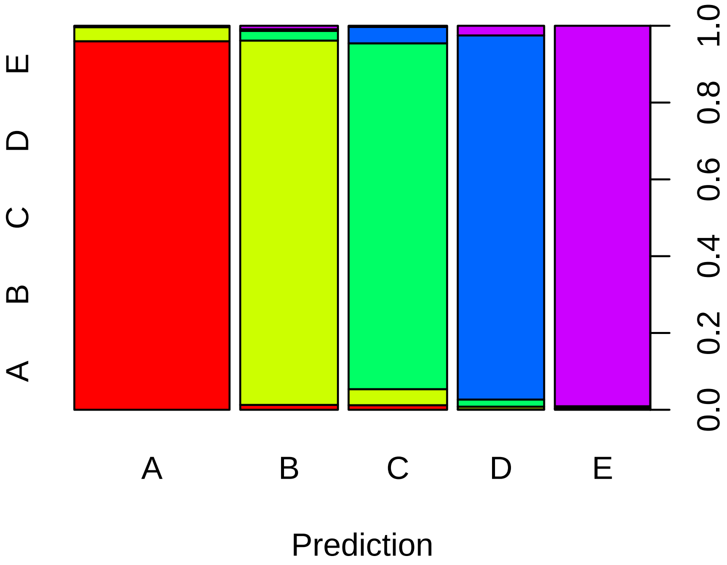

Figure 2: Confusion matrix visualised.

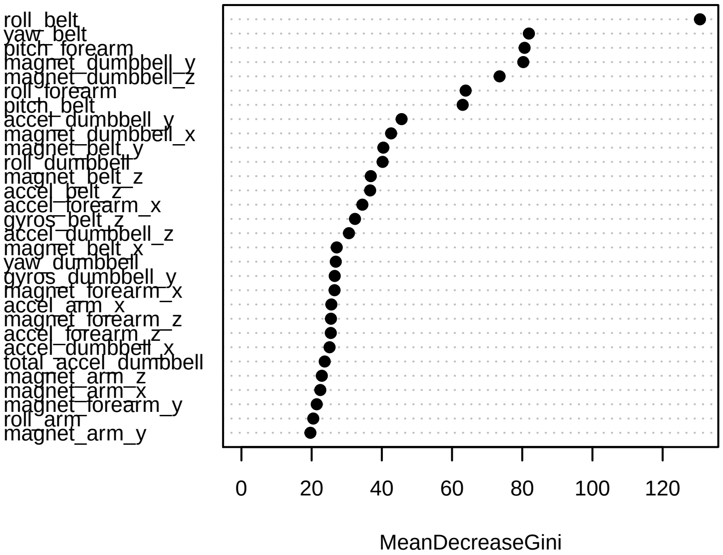

Figure 3: Relative importance of independent variables.