Predicting Exercise Quality with Machine Learning

Erich Meschkat

Sunday, November 23, 2014

Load in libraries

```
library(randomForest)
set.seed(11)
```

Read in train and test data from web.

```
train.url <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
test.url <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
download.file(train.url, "c:/program files/r/pmltrain.csv")
download.file(test.url, "c:/program files/r/pmltest.csv")
train <- read.csv("c:/program files/r/pmltrain.csv")
test <- read.csv("c:/program files/r/pmltest.csv")
```

Subset data to include only columns that DO NOT have NAs in the test set (leaves 60 factors, but includes some ID categories)

```
test.index <- colSums(is.na(test)) == 0
test.sub <- test[which(test.index == TRUE)]
train.sub <- train[which(test.index == TRUE)]
```

We will use a Random Forest alogrithm to develop a model. This is a good fit because it does not require scaling our parameters, and can accomodate a large number of features. Though computationally expensive, we have trimmed our features enough in the previous step to allow this model to run on a normal laptop.

```
rfmodel <- randomForest(x=train.sub[,8:59], y=train.sub$classe )
rfmodel
```

Call: randomForest(x = train.sub[, 8:59], y = train.sub$classe) Type of random forest: classification Number of trees: 500 No. of variables tried at each split: 7

```
    OOB estimate of  error rate: 0.3%
```

Confusion matrix: A B C D E class.error A 5578 2 0 0 0 0.0003584229 B 11 3783 3 0 0 0.0036871214 C 0 11 3409 2 0 0.0037989480 D 0 0 20 3194 2 0.0068407960 E 0 0 2 5 3600 0.0019406709

The Random Forest performs extrememly well, with an Out of Sample error rate of less than 1% (0.28%). Given the nature of the Random Forest Alogrithm, cross validation is included in the error rate.

Let's predict the classifaction of the test set.

```
rftest <- predict(rfmodel, test.sub[8:59])
paste(rftest)
```

```
##  [1] "B" "A" "B" "A" "A" "E" "D" "B" "A" "A" "B" "C" "B" "A" "E" "E" "A"
## [18] "B" "B" "B"
```

Looks like we predicted all 20 correctly!!

Below is a plot showing the error rate of our model as the number of trees increases.

```
plot(rfmodel, type="l")
```