Practical Machine Learning

MJM

April 3, 2016

From "http://groupware.les.inf.puc-rio.br/har#weight_lifting_exercises":

> This human activity recognition research has traditionally focused on discriminating between different activities, i.e. to predict "which" activity was performed at a specific point in time (like with the Daily Living Activities dataset above). The approach we propose for the Weight Lifting Exercises dataset is to investigate "how (well)" an activity was performed by the wearer. The "how (well)" investigation has only received little attention so far, even though it potentially provides useful information for a large variety of applications,such as sports training.
>
> Six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E).

Given the Weight Lifting data set, the goal is to correctly predict which class (*classe*) the performed exercises fall into.

First we load the data into the testing and training variables.

```
testing <- read.csv("pml-testing.csv")
training <- read.csv("pml-training.csv")
```

Then we remove the variables from the input data that is not included in the testing data to provide predictability, using the `nearZeroVar()` function, and also make `classe` a factor variable.
This removes the useless first 7 columns and all other columns that are all NA's in the testing dataset from both testing and training.

```
badcols <- c(1:5,7,nearZeroVar(testing))

training <- training[-badcols]
testing <- testing[-badcols]

training$classe <- as.factor(training$classe)
```

In order to test our algorithms, we need to partition our training data into an actual training dataset and a testing set - but we will call it a "quizzing" set since it is not the actual test.

```
inTrain <- createDataPartition(training$classe,p=2/3,list=FALSE)
mytrain <- training[inTrain,]
quizzing <- training[-inTrain,]
set.seed(311)
```

First we will use a Generalized Boost Model to predict exercises.

```
modelfitgbm <- train(classe ~., method="gbm", data=mytrain)
predictionsgbm <- predict(modelfitgbm,quizzing)
```

```
confusionMatrix(predictionsgbm,quizzing$classe)$overall[1]
```

```
## Accuracy
## 0.963603
```

The results are good - 96% accuracy on the quiz data. We can anticipate getting at least 19 of 20 correct on the test set with this model.

Next we can try random forests to try to improve the results.

```
modelfitrf <- train(classe ~., method="rf", data=mytrain,
                trControl=trainControl(method='cv'),
                number=3, allowParallel=TRUE )
predictionsrf <- predict(modelfitrf,quizzing)
```

```
confusionMatrix(predictionsrf,quizzing$classe)$overall[1]
```

```
##  Accuracy
## 0.9934241
```

The accuracy for this model is even better - over 99%.

```
testpredgbm <- predict(modelfitgbm,testing)
testpredrf <- predict(modelfitrf,testing)
print(data.frame(testpredgbm,testpredrf))
```

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

Both predictions were 100% accurate on the test data (and, obviously, they agreed with each other as well).