Predicting Proper Form in Exercise Using Wearable Data and Random Forest

Connor Gooding

9/21/2017

```
library(ggplot2)
library(caret)
```

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

```
library(randomForest)
```

```
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following object is masked from 'package:randomForest':
##
##     combine
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

About The Data

This report uses data on Qualitative Activity Recognition of Weight Living Exercises taken by Groupware@LES to predict whether or not a subject exercising with dumbbells is doing so with the proper form or failing to meet proper form in one of four common ways. Six subjects performed Unilateral Dumbbell Curls while wearing four inertial measurement units (IMU's). These four units were attached to the belt, arm, forearm, and dumbbell of each subject as they performed the exercises.

The raw data comes in a training set and a test set. These are loaded into R in the following code block:

```
train_url <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
test_url <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

if(!file.exists("data/pml_training.csv"))
{
```

```
    download.file(train_url, destfile = "data/pml_training.csv")
}

if(!file.exists("data/pml_testing.csv"))
{
    download.file(test_url, destfile = "data/pml_testing.csv")
}

train.raw <- read.csv("data/pml_training.csv")
validation <- read.csv("data/pml_testing.csv")
```

Splitting the Data

To prepare a training model, one must partition the raw training data into a training set and testing set. This leaves the raw test set as a validation set and allows for reformulation of the trained model before submitting any final predctions. For this particular model, 70% of the raw training data was used to train the model while the remaining 30% was withheld to test the trained model.

```
inTrain <- createDataPartition(train.raw$classe, p = .7)[[1]]
training <- train.raw[inTrain,]
testing <- train.raw[-inTrain,]
```

Feature Selection

The raw dataset comes with 160 columns, as this is how many features the IMU's capture when running. However, many of these features were captured too sparsely to be of any use in training a prediction model. These sparse columns, which are more than 90% blank or error-filled, are discarded here.

```
sparseCols <- which(apply(training, 2, function(x) mean(is.na(x) | (x == ""))) > .9)
training <- training[,-sparseCols]
testing <- testing[,-sparseCols]
validation <- validation[,-sparseCols]
```

Of the remaining 60 features, 7 more of them should not be used to predict future activity classes. The user's name or what time of day they performed the activity should not factor into whether or not the user is performing the activity correctly in an arbritrary moment in time. In fact, the timestamp data's strong relationship to the activity class is simply an artifact of the experiment structure.

The windowing features are merely metadata taken by the IMU's that would serve better for an exploratory analysis of this data, but not for a true predictive analysis. These should also be discarded.

```
boringCols <- c(1:7)
training <- training[,-boringCols]
testing <- testing[,-boringCols]
validation <- validation[,-boringCols]
```

Prediction

While several training algorithms taught in this class could be used to train a model, using a random forest returns $> 99\%$ accuracy. Other methods, such as linear discriminant analysis and gradient boosting, take a lot of time and report space without ultimately improving the accuracy of the model. Therefore, the random forest algorithm alone is used to predict activity class.

```
mdlRF <- randomForest(classe ~ ., data = training, method = 'class')
mdlRF
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = training, method = "class")
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.54%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3903    2    1    0    0 0.0007680492
## B   12 2640    6    0    0 0.0067720090
## C    0   18 2375    3    0 0.0087646077
## D    0    0   26 2224    2 0.0124333925
## E    0    1    0    3 2521 0.0015841584
```

Testing

Applying the random forest model on the testing data yields 99.3% accuracy. More specifically, the random forest model has a 99.8% positive predictive value of correct form (class A) and specificity $> 99.5\%$ for all five activity classes. This accuracy should be more than satisfactory to expect perfect prediction for the 20 observations in the validation dataset.

```
pred <- predict(mdlRF, newdata = testing)
confusionMatrix(testing$classe, pred)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    1    0    0    0
##          B    9 1128    2    0    0
##          C    0    4 1022    0    0
##          D    0    0    9  955    0
##          E    0    0    0    4 1078
##
## Overall Statistics
##
##                Accuracy : 0.9951
##                  95% CI : (0.9929, 0.9967)
##     No Information Rate : 0.2858
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9938
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9946   0.9956   0.9894   0.9958   1.0000
## Specificity            0.9998   0.9977   0.9992   0.9982   0.9992
```

```
## Pos Pred Value         0.9994   0.9903   0.9961   0.9907   0.9963
## Neg Pred Value         0.9979   0.9989   0.9977   0.9992   1.0000
## Prevalence             0.2858   0.1925   0.1755   0.1630   0.1832
## Detection Rate         0.2843   0.1917   0.1737   0.1623   0.1832
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9972   0.9966   0.9943   0.9970   0.9996
```

Validation

The following output shows the 20 predictions made on the validation set by the random forest model.

```
validation$classe <- predict(mdlRF, newdata = validation)
final <- validation[, 53:54]
t(final)
```

```
##            [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## problem_id " 1" " 2" " 3" " 4" " 5" " 6" " 7" " 8" " 9" "10"  "11"  "12"
## classe     "B"  "A"  "B"  "A"  "A"  "E"  "D"  "B"  "A"  "A"   "B"   "C"
##            [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20]
## problem_id "13"  "14"  "15"  "16"  "17"  "18"  "19"  "20"
## classe     "B"   "A"   "E"   "E"   "A"   "B"   "B"   "B"
```

Conclusion

Using the random forest algorithm on the 53 features related to user motion that contained ample data for analysis yielded a 100% (20/20) prediction accuracy on the validation set. While the out-of-sample error was technically lower than the in-sample error, this is most likely the case because of the small sample size of the validation set. However, the testing out-of-sample error (99.3%) is still demonstrably close to the in-sample error (99.5%), suggesting that the amount of overfitting done by the model to the training set is negligible.

Accuracy could potentially be improved upon if the timestamp variables were used to conduct a time-series analysis and forecasting on the data. However, without accounting for the time-sensitivity, it appears the model trained in this report does a more than adequate job predicting form of a user doing Unilateral Dumbbell Curls.

Bibliography

Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013.