HAR Using Weight Lifting Exercise Data: Excercise Form Prediction

James Kim

March 8, 2017

Executive Summary

Human activity research (HAR) is presented in which a weight lifting exercise data set is used to identify exercise form. Using devices such as Jawbone Up, Nike FuelBand, and Fitbit, it is now possible to collect large amounts of data on personal activity relatively inexpensively. While quantifying how much of a particular activity is performed has frequently been investigated, quantifying how well it is done is just beginning to gain attention. The goal of this investigaiton is to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants who were asked to perform barbell lift exercise correctly and incorrectly in 5 different ways. The data set comes from the Weight Lifting Exercise Dataset of the publication: Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises.

Random forest predictive modeling is used for training and prediction. Selection of features for the predictive modeling and the results of the modeling are presented and discussed. Using 12 features selected out of approximately 60 raw feature data, 99% accuracy (1.4% OOB error rate) is achieved. Finally, a prediction for a test set of 20 observations is presented.

Exploratory Data Analysis & Feature Selection

Data sets in csv form are loaded, consisting of a train data set and a test data set.

Because the number of variables and the number of observations are both very large, a substantial reduction in the number of potential features using intuition about the measurements proves effective over exploring every variable. The effectiveness of the selected feature variables can be confirmed by the outcome of the predictive modeling.

The following describes the basis for the initial intuition-based feature selection.

A. sensor selection:

1. belt sensor most likely would move mostly in the z direction
2. arm sensor would not have much movement except in the case of class B
3. forearm sensor most likely would move mostly in the y and z directions
4. dumbbell sensor would most likely have movement similar to the forearm

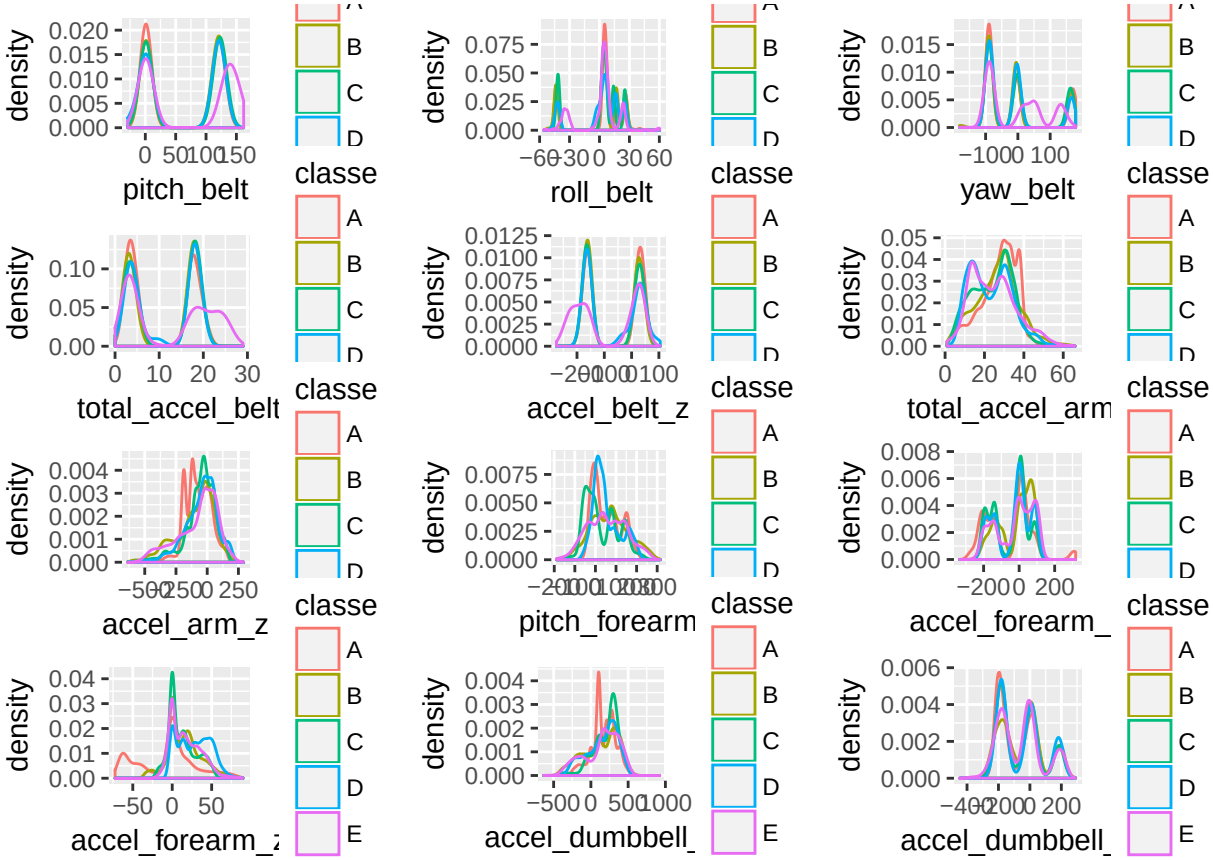
B. yaw, pitch, and roll summarize raw gyro data (i.e. related, so gyro data are not used)

C. magnet data are not used

D. whereas in the journal publications the authors who gathered the data show using statistical parameters derived from the raw data, raw data are directly used in the modeling

From this starting point, feature selection is optimized by examining importance of variables, further adding and removing variables and minimizing prediction error. Shown below are density plots for the selected 12 feature variables distinguished by color according to the classe variable containing the classification result. The density plots are preferable over box plots for feature discovery due to revealing more information regarding distribution. These feature variables are expected to vary extensively according to the classification

for high predictive power, and the density plots show that the selected features do appear to be quite variable with respect to the classification.



Modeling

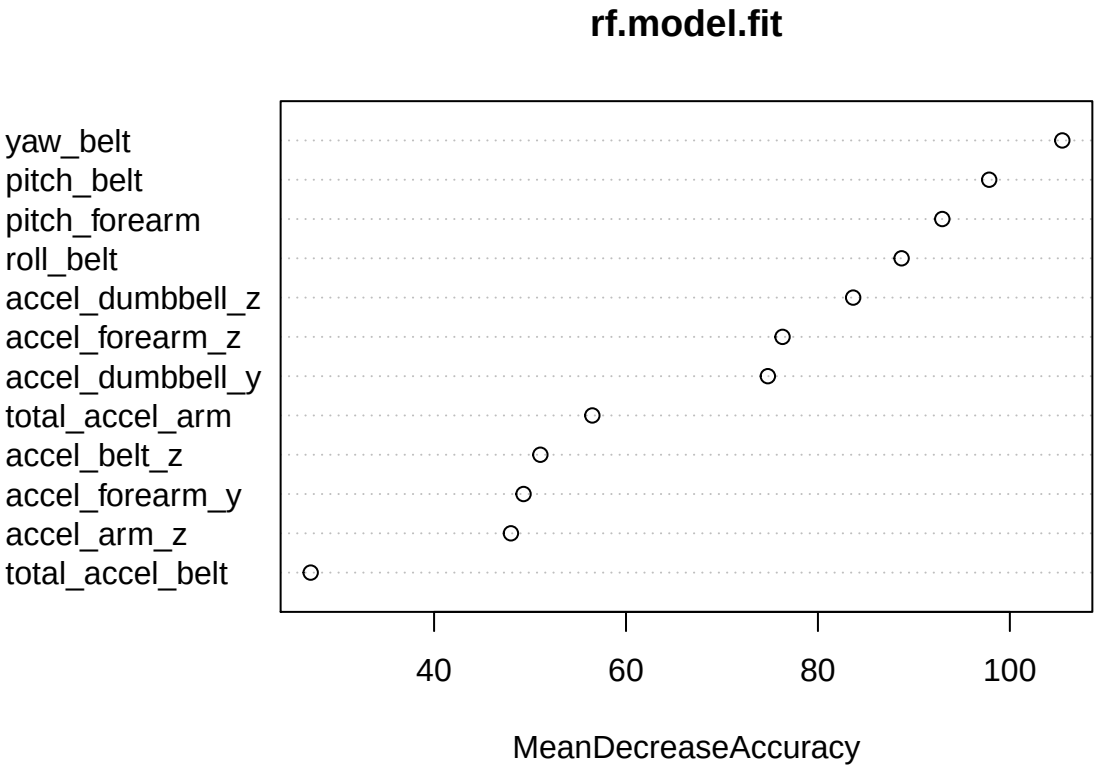
Exploratory random forest modeling is performed with the 12 feature variables, and the code for accomplishing this is shown below (note the code shows only the modeling portion).

```
require(caret);require(randomForest);set.seed(01010)
training.partition<-createDataPartition(training$classe,p=0.7,list=FALSE)
# parallel (75% of the cores) processing enclosure
require(parallel);require(doSNOW)
cluster<-makeCluster(floor(detectCores()*0.75),type='SOCK');registerDoSNOW(cluster)
rf.model.fit<-randomForest(classe~.,training[training.partition,],importance=TRUE)
stopCluster(cluster);registerDoSEQ()
# enclosure end
```

The result of the modeling performed with randomly selected 70% of the data set is presented below. In the result, OOB error rate is 1.4% for the modeling, and this low OOB error rate shows that the feature selection is effective. Also ranked importance of the features is shown in a variable importance plot. In the plot, yaw_belt and pitch_belt features are shown to be of surprisingly high importance (i.e. the accuracy decreases the most without them). This is a surprising result since abdomen movement during a barbell exercise would be thought to be minimal. However, the high ranking may indicate that these variables offer the most distinguishable variability and thus are important.

```
##
## Call:
```

```
##  randomForest(formula = classe ~ ., data = training[training.partition,      ], importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 1.35%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3858   14   18   15    1 0.012288786
## B   28 2605   22    3    0 0.019939804
## C    6   22 2353   15    0 0.017946578
## D    2    1   13 2233    3 0.008436945
## E    1    8    6    7 2503 0.008712871
```



The model fit is used to predict classification for the testing set consisting of the remaining 30% of the data, and a confusion matrix is shown for the classification prediction and the actual classification. As shown, the overall prediction accuracy is high at 99%.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1657   14    2    1    0
##          B    8 1116    6    2    3
##          C    8    8 1012   10    0
##          D    1    1    6  949    1
##          E    0    0    0    2 1078
##
## Overall Statistics
##
##                Accuracy : 0.9876
```

```
##                  95% CI : (0.9844, 0.9903)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9843
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9898   0.9798   0.9864   0.9844   0.9963
## Specificity            0.9960   0.9960   0.9946   0.9982   0.9996
## Pos Pred Value         0.9898   0.9833   0.9750   0.9906   0.9981
## Neg Pred Value         0.9960   0.9952   0.9971   0.9970   0.9992
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2816   0.1896   0.1720   0.1613   0.1832
## Detection Prevalence   0.2845   0.1929   0.1764   0.1628   0.1835
## Balanced Accuracy      0.9929   0.9879   0.9905   0.9913   0.9979
```

Cross validation modeling is then performed to reduce overfitting and improve accuracy when predicting unseen test data sets. The code for accomplishing this is shown (note the code shows only the modeling portion), followed by the result of the modeling. The result of the 10-fold cross validation random forest modeling shows accuracy of 99%, little changed from the exploratory modeling result. It is expected, however, that overfitting is substantially reduced such that a prediciton made on an unseen data set is improved.

```
require(caret)
set.seed(01010)
train.control<-trainControl(method='cv',number=10)
# parallel (75% of the cores) processing enclosure
require(parallel);require(doSNOW)
cluster<-makeCluster(floor(detectCores()*0.75),type='SOCK');registerDoSNOW(cluster)
cv.rf.model.fit<-train(classe~.,data=training,method='rf',trControl=train.control)
stopCluster(cluster);registerDoSEQ()
# enclosure end
```

```
## Random Forest
##
## 19622 samples
##    12 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 17661, 17660, 17660, 17659, 17658, 17660, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9897051  0.9869801
##    7    0.9869531  0.9835006
##   12    0.9833347  0.9789287
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 2.
```

Results

The cross validation random forest modeling is used to predict classfication of a test data set consisting of 20 observables, and the prediction result is shown below. The prediction result is confirmed (via the project quiz) to be 100% accurate, in line with the above results showing high predictive accuracy of the modeling.

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