Human Activity Detection with Machine Learning

Harish Kumar Rongala

January 29, 2017

1. Executive Summary

This document aims to predict how accurately **"Unilateral Dumbbell Biceps Curl"** is performed by the user. Wearable sensors (see Appendix-B) comprised of Inertial measurement units (IMU) provides the data required. In this document machine learning algorithms **rpart, gradient boosting and random forests** were used to predict/classify the activity. This document also discusses the model built, cross-validation and out of sample errors.

2. About Data set

The data set used in this document is licensed under *Creative Commons license* and following publication can be referred to know more about the experiment and contributor.

Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises Read more here

Six young healthy 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).

3. Preparing data

In this approach, data is prepared in 3 steps. First, data is loaded into the working directory. Next, missing values ("NA"/"NaN") are handled. Finally, to test the accuracy of the models, we need a validation set. So, we divide the data set into training and validation sets.

3.1. Loading data

```
## Training data url
url1<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv";

## Download and read file
download.file(url1,"HAR_training.csv");
HAR_train<-read.csv("HAR_training.csv",na.strings = c("","NA"));
dim(HAR_train);
```

```
## [1] 19622   160
```

This data set contains 19622 observations and 160 columns.

3.2. Cleaning data

There are multiple columns containing missing values. It may not be ideal to impute values here, as these columns are redundant and represent maximum, minimum, average, std deviation, variance, amplitude,skewness, and kurtosis of the signal. First 7 columns represent values like user id, subject name, timestamp etc., which are unrelated in classifying the activity. So, we choose to drop them as well.

```
## This variable holds columns to be dropped
drp<-NULL;
##
drp<-append(drp,c(1:7));
drp<-append(drp,grep("^max|^min|^avg|^std|^var|^amp|^skew|^kurt",names(HAR_train)));
## Dropping unwanted columns
HAR_train_clean<-HAR_train[,-drp];
dim(HAR_train_clean);
```

```
## [1] 19622    53
```

After dropping the unwanted columns, our data set contains 53 columns.

3.3. Data Slicing

As we intend to create multiple models and test, it is always a good practice to slice our training data into training and validation set (when the data set is large enough). Here, **70%** of the data set is labeled as training set and rest **30%** is labeled as the validation set.

```
## Creating training and validation sets
suppressPackageStartupMessages(library(caret));
## Set seed for reproducibility
set.seed(1225);
ind<-createDataPartition(HAR_train_clean$classe,p=0.7,list=FALSE);
train_set<-HAR_train_clean[ind,];
valid_set<-HAR_train_clean[-ind,];

dim(train_set);
```

```
## [1] 13737    53
```

```
dim(valid_set);
```

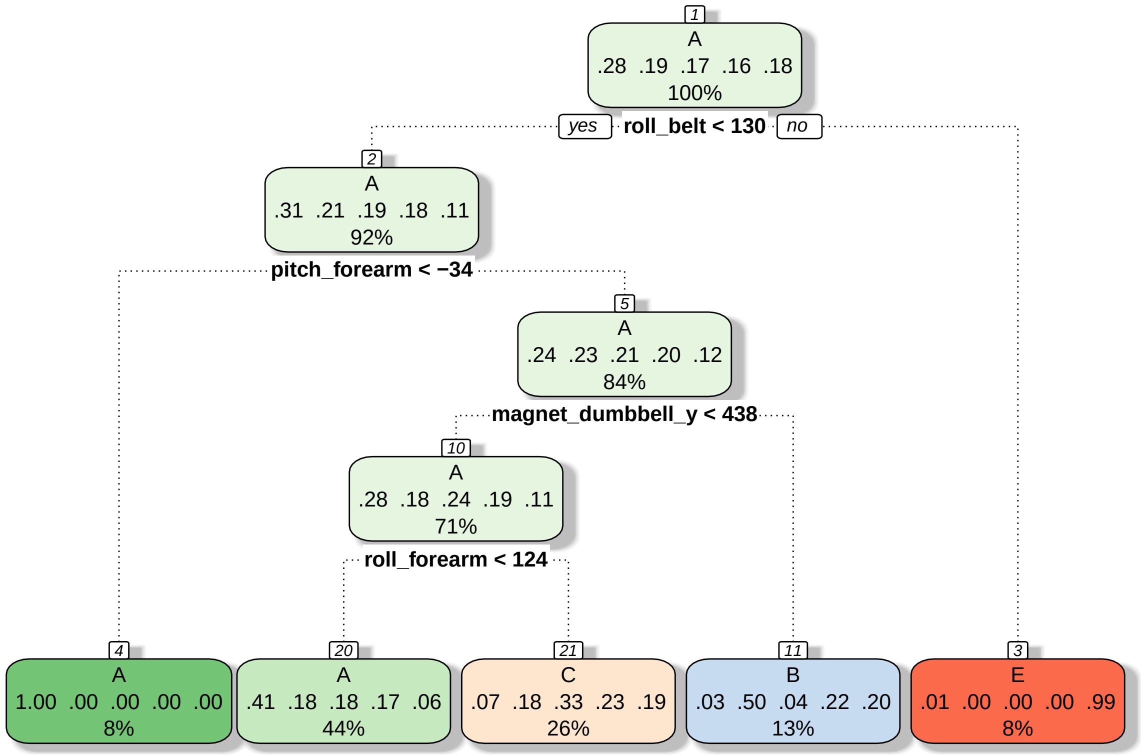
```
## [1] 5885   53
```

4. Model fitting

We start with **rpart** tree classification and end with **gradient boosting method** and **random forests method**. Cross-validation is performed with the K-fold method, K being 10. This ensures the accuracy of the model by training and testing on every observation of the training set and averaging the accuracies.

```
## Set seed for reproducibility
set.seed(1225);
tr_ctrl<-trainControl(method="cv", savePredictions = TRUE, classProbs = TRUE);
## Using rpart classifier
suppressPackageStartupMessages(fit0<-train(classe~.,data=train_set,method="rpart",trControl=tr_ctrl));

## Look at the model
suppressPackageStartupMessages(library(rattle));
fancyRpartPlot(fit0$finalModel);
```



Rattle 2017−Apr−04 14:42:50 Harish_Rongala

```
## Set seed for reproducibility
set.seed(1225);
## Fit Gradient Boosting method
fit1<-train(classe~.,data=train_set,method="gbm",trControl=tr_ctrl);
```

```
## Set seed for reproducibility
set.seed(1225);
## Fit using Random Forests
fit2<-train(classe~.,data=train_set,method="rf",trControl=tr_ctrl);
rf_ind<-fit2$pred$mtry==2;
```

5. Prediction

5.1. Prediction on Validation set

```
## Predict using rpart classifier
suppressPackageStartupMessages(rpart_pr<-predict(fit0,valid_set));
## Predict using Gradient boosting model
suppressPackageStartupMessages(gbm_pr<-predict(fit1,valid_set));
## Predict using Random Forest model
suppressPackageStartupMessages(rf_pr<-predict(fit2,valid_set));

## Look at the accuracy
confusionMatrix(rpart_pr,valid_set$classe);
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1512  471  478  434  147
##          B   29  384   31  169  132
##          C  130  284  517  361  296
##          D    0    0    0    0    0
##          E    3    0    0    0  507
##
## Overall Statistics
##
##                Accuracy : 0.4962
##                  95% CI : (0.4833, 0.509)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3418
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9032  0.33714  0.50390   0.0000  0.46858
## Specificity            0.6367  0.92394  0.77958   1.0000  0.99938
## Pos Pred Value         0.4970  0.51544  0.32557      NaN  0.99412
## Neg Pred Value         0.9430  0.85311  0.88155   0.8362  0.89302
## Prevalence             0.2845  0.19354  0.17434   0.1638  0.18386
## Detection Rate         0.2569  0.06525  0.08785   0.0000  0.08615
## Detection Prevalence   0.5169  0.12659  0.26984   0.0000  0.08666
## Balanced Accuracy      0.7699  0.63054  0.64174   0.5000  0.73398
```

```r
confusionMatrix(gbm_pr,valid_set$classe);
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1654   31    0    2    2
##          B   11 1058   35    3   12
##          C    6   47  978   32    4
##          D    3    3    9  918   15
##          E    0    0    4    9 1049
##
## Overall Statistics
##
##                Accuracy : 0.9613
##                  95% CI : (0.956, 0.966)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.951
##  Mcnemar's Test P-Value : 1.506e-06
##
## Statistics by Class:
```

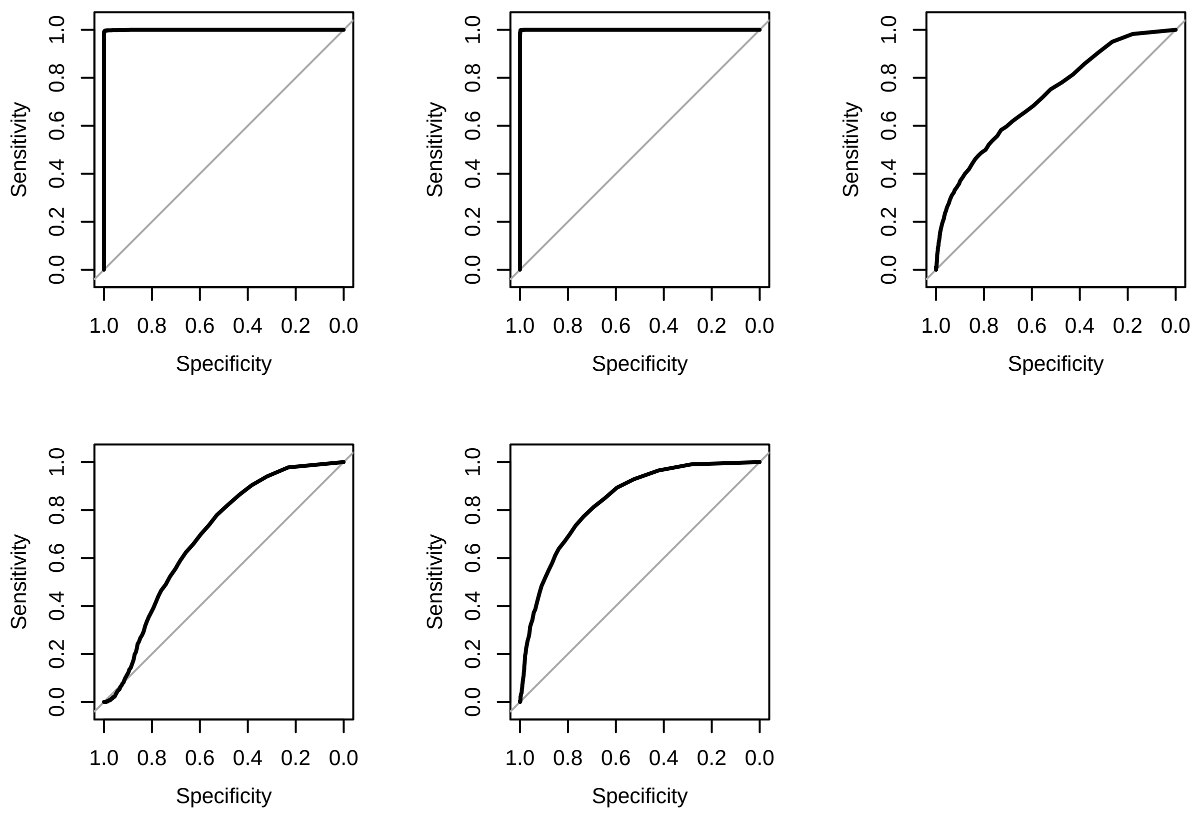
```
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9881   0.9289   0.9532   0.9523   0.9695
## Specificity            0.9917   0.9871   0.9817   0.9939   0.9973
## Pos Pred Value         0.9793   0.9455   0.9166   0.9684   0.9878
## Neg Pred Value         0.9952   0.9830   0.9900   0.9907   0.9932
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2811   0.1798   0.1662   0.1560   0.1782
## Detection Prevalence   0.2870   0.1901   0.1813   0.1611   0.1805
## Balanced Accuracy      0.9899   0.9580   0.9674   0.9731   0.9834
```

```
confusionMatrix(rf_pr,valid_set$classe);
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1671    4    0    0    0
##          B    3 1132    5    0    0
##          C    0    3 1021   19    1
##          D    0    0    0  944    2
##          E    0    0    0    1 1079
##
## Overall Statistics
##
##                Accuracy : 0.9935
##                  95% CI : (0.9911, 0.9954)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9918
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9982   0.9939   0.9951   0.9793   0.9972
## Specificity            0.9991   0.9983   0.9953   0.9996   0.9998
## Pos Pred Value         0.9976   0.9930   0.9780   0.9979   0.9991
## Neg Pred Value         0.9993   0.9985   0.9990   0.9960   0.9994
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2839   0.1924   0.1735   0.1604   0.1833
## Detection Prevalence   0.2846   0.1937   0.1774   0.1607   0.1835
## Balanced Accuracy      0.9986   0.9961   0.9952   0.9894   0.9985
```

5.2. Look at the ROC

From the above results, Random Forests seems to have higher accuracy. It's ROC of each class (each class of the activity) looks as following.



5.3. Prediction on Test set

```
## Testing data url
url2<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv";
download.file(url2,"HAR_test.csv");
HAR_test<-read.csv("HAR_test.csv");

test_pr_rpart<-predict(fit0,HAR_test);
test_pr_gbm<-predict(fit1,HAR_test);
test_pr_rf<-predict(fit2,HAR_test);

test_pr_rpart;
```

```
##  [1] C A C A A C C A A A C C C A C A A A A C
## Levels: A B C D E
```

```
test_pr_gbm;
```

```
##  [1] B A B A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```

```
test_pr_rf;
```

```
##  [1] B A B A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```

6. Conclusion

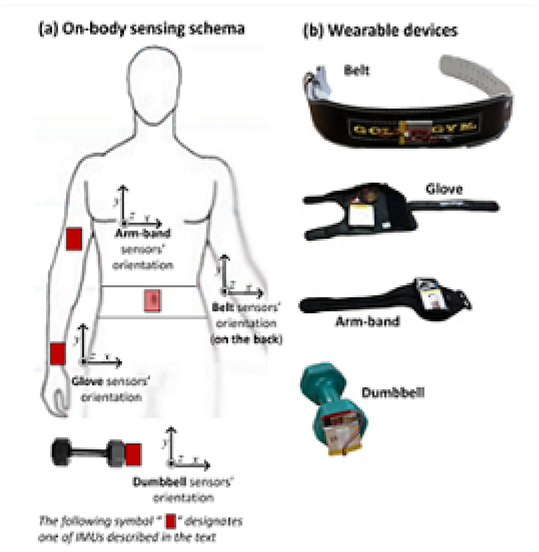
- Out of sample error of rpart classifier is 0.5038
- Out of sample error of Gradient Boosting Model is 0.0387
- Out of sample error of Random Forest model is 0.0065
- In this case, prediction on the test set is same for GBM and Random Forests. However, considering the error rate, we submit the Random forests model as our final model.

7. Appendix

Appendix-A (ROC code)

```
library(pROC);
rf_ind<-fit2$pred$mtry==2;
suppressMessages(library(pROC));
p1<-roc(fit2$pred$obs[rf_ind],fit2$pred$A[rf_ind]);
p2<-roc(fit2$pred$obs[rf_ind],fit2$pred$B[rf_ind]);
p3<-roc(fit2$pred$obs[rf_ind],fit2$pred$C[rf_ind]);
p4<-roc(fit2$pred$obs[rf_ind],fit2$pred$D[rf_ind]);
p5<-roc(fit2$pred$obs[rf_ind],fit2$pred$E[rf_ind]);
```

Appendix-B (Sensing setup)


(a) On-body sensing schema
(b) Wearable devices
Belt
Glove
Arm-band
Dumbbell
Arm-band sensors' orientation
Belt sensors' orientation (on the back)
Glove sensors' orientation
Dumbbell sensors' orientation
The following symbol "■" designates one of IMUs described in the text