Classification of Exercise Execution Quality

Bill Rowe

May 26, 2016

Executive Summary

The data for this project comes from a Human Activity Recognition (HAR) dataset. Six subjects, aged 20 to 28, were asked to perform 10 repetitons of the Unilateral Dumbbell Biceps Curl, with a realtively light weight, in five different ways. As Vellasco noted in the paper referenced for this dataset, “A key requirement for effective training to have a positive impact on cardio-respiratory fitness is a proper technique. Incorrect technique has been identified as the main cause of training injuries.” Thus, the goal of this analysis is to model and predict the “quality of execution” using accelerometers attached to the subjec’s body and dumbbell.

After cleaning the training dataset by removing the variables that had many missing values and the highly correlated variables, the training dataset was split 70:30 into a subset for training and a subset for validation. Recursive Partitioning and Regression Trees (Rpart), Random Forests (RandomForest), Linear Discriminant Analysis (LDA), k-Nearest Neighbor Classification (knn), Generalized Boosted Regression Models (gbm) and eXtreme Gradient Boosting (xgbTree) were evaulated. GBM and xgbTree had very long run times despite using parallel processing (doParallel). Each model was built using the training subset and evaluated the hold out validation set. Table 1 shows the results for all models. The most accurate model using random forests (out of sample error of 0.61%) was used to predict the “classe” of 20 observations supplied by Coursera.

Exploratory Data Analysis

As mentioned above, the goal is to build a model for and predict the “quality of execution”. Each subject performed 10 reps according to the following directions: repetitons in Class A were the correct technique and repetitions in Classes B-E were common improper or incorrect executions of the bicep curl.

- ***Class A:*** exactly according to the specification
- ***Class B:*** throwing the elbows to the front
- ***Class C:*** lifting the dumbbell only halfway
- ***Class D:*** lowering the dumbbell only half way
- ***Class E:*** throwing the hips to the front.

Data was collected using sensors placed as shown in Figure 1 in the Appendix.

The original dataset has 160 fields or columns. Not all columns are test data. The first column (“X”) is a row identifier, an artifact from querying a database. The next 6 columns identify the user, timestamps and “windows”. There are columns labeled “kurtosis”" and “skewness” that are either blank or have “#DIV/0!” as their value. Other statistical summary columns (“max”, “min”, “var”, “stddev”) and calculated fields (“amplitude”) are often valued as “NA”. The very last column in the training set is the “quality of exercise” indicator, “classe”. (Perhaps Italian for class). The row identifier, user, timestamp, windows and summary statistic and calculated columns were removed with the following code.

```
names.remove <-c(
  "X", "user_name", "raw_timestamp_part_1", "raw_timestamp_part_2", "cvtd_times
tamp",
  "new_window", "num_window")

pml.train <- pml.train[,!names(pml.train) %in% names.remove]
pml.test <- pml.test[,!names(pml.test) %in% names.remove]

# helper function from Stephen Turner
#http://www.gettinggeneticsdone.com/2011/02/summarize-missing-data-for-all.html
propmiss <- function(dataframe) {
  m <- sapply(dataframe, function(x) {
    data.frame(
      nmiss=sum(is.na(x)),
      n=length(x),
      propmiss=sum(is.na(x))/length(x)
    )
  })
  d <- data.frame(t(m))
  d <- sapply(d, unlist)
  d <- as.data.frame(d)
  d$variable <- row.names(d)
  row.names(d) <- NULL
  d <- cbind(d[ncol(d)],d[-ncol(d)])
  return(d[order(d$propmiss), ])
}

#get the fraction missing by variable
names.remove2<-propmiss(pml.train)
c<-names.remove2[which(names.remove2$propmiss > 0.9), 1]
pml.train.clean<-pml.train[,!names(pml.train) %in% names.remove2[which(names.re
move2$propmiss > 0.9), 1]]
pml.test.clean<-pml.test[,!names(pml.test) %in% names.remove2[which(names.remov
e2$propmiss > 0.9), 1]]
#recheck
propmiss(pml.train.clean)
propmiss(pml.test.clean)
```

None of the 52 remaining dependent variables had very low variance and all were kept. However, many machine learning algorithms do not work well with highly correlated independent variables. These were located with caret's find findCorrelation method. Thke following highly correlated variables were removed from the training and test datasets: accel_belt_z, roll_belt, accel_belt_y, accel_arm_y, total_accel_belt, accel_dumbbell_z,accel_belt_x, pitch_belt, magnet_dumbbell_x, accel_dumbbell_y, magnet_dumbbell_y, accel_arm_x, accel_dumbbell_x, accel_arm_z, magnet_arm_y, magnet_belt_z, accel_forearm_y, gyros_forearm_y, gyros_dumbbell_x, gyros_dumbbell_z, gyros_arm_x. This reduces the number of variables in the test and training sets to 31 predictor variables and the "classe" indepedent variable.

```
#do standard tests for 0 variance
zeroVarCols<-nearZeroVar(pml.train.clean, saveMetrics = TRUE)
pml.train.clean<-pml.train.clean[, zeroVarCols$nzv==FALSE]
pml.test.clean<-pml.test.clean[, zeroVarCols$nzv==FALSE]
#really no changes
dim(pml.train.clean)  #19622 and 53
dim(pml.test.clean) #20 and 53

# find columns to remove in order to reduce pair-wise correlations
highlyCorDescr <- findCorrelation(cor(pml.train.clean[,1:52]), cutoff = .75)
pml.train.clean <- pml.train.clean[,-highlyCorDescr]
pml.test.clean <- pml.test.clean[,-highlyCorDescr]
dim(pml.train.clean);dim(pml.test.clean);
#number of variables is now 32

#cast the classe column as a factor
pml.train.clean$classe = factor(pml.train.clean$classe)
```

Model Evaluation

To get an unbiased estimation of the out of sample error for each of the tested models, the supplied training dataset was divided (70:30) into a sub training dataset (13,737 records) and a sub validation dataset (5,885 records). The sub training dataset was used to build the model and the sub validation dataset was used to test the model and get an estimate of the classificaton error and accuracy that could be encountered in predicting "classe" values in an external dataset. Each generated model will be compared on the basis of accuracy, classification error for the validation sub dataset and runtime. The best model would then be used to predict "classe" for the 20 observations in the external testing dataset.

To speed execution, the doParallel package was used with the number of clusters set to 2. Using the makeCluster(detectCores()) caused my pesonal computer to freeze.

Model Results

In order to estimate the accuracy and the out of sample error, the validation sub dataset was used. The following table of results shows the model, estimated accuracy and estimated out of sample error.

Table 1: Comparison of in sample and out of sample model results

Model	rf	xgbTree	gbm	knn	lda	rpart
In training						
Accuracy	1.0000	0.9993	0.9609	0.9239	0.5836	0.5316
LL	0.9997	0.9988	0.9575	0.9194	0.5753	0.5232
UL	1.0000	0.9997	0.9641	0.9283	0.5919	0.5400
No Information Rate	0.2843	0.2843	0.2869	0.2866	0.2982	0.3397
P-Value [Acc > NIR]	< 2.2e-16	< 2.2e-16	< 2.2e-16	< 2.2e-16	< 2.2e-16	< 2.2e-16
Kappa	1.0000	0.9992	0.9505	0.9038	0.4728	0.4104
Out of Sample						
Accuracy	0.9939	0.9901	0.9446	0.8669	0.5799	0.5179
LL	0.9915	0.9873	0.9385	0.8580	0.5672	0.5051
UL	0.9957	0.9925	0.9503	0.8755	0.5926	0.5308
No Information Rate	0.2853	0.2846	0.2890	0.2921	0.2969	0.3499
P-Value [Acc > NIR]	< 2.2e-16	< 2.2e-16	< 2.2e-16	< 2.2e-16	< 2.2e-16	< 2.2e-16
Kappa	0.9923	0.9875	0.9299	0.8316	0.4683	0.3928
Error	0.0061	0.0099	0.0554	0.1331	0.4201	0.4821

External prediction

The most accurate performing model, random forests, was used to predict the value of the “classe” variable in the supplied testing dataset. These results are shown in the following list.

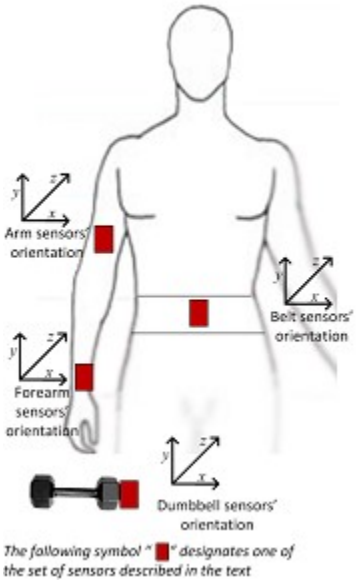
Table 2: Predictons using Random Forest Model

```
pml.test.predict <- predict(rf.model, pml.test, type="class")
pml.test.predict
```

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

Appendix

Figure 1: Locations of sensor used in study.



This image was downloaded from http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har).

System Information

The analysis was run with the following software and hardware.

```
sessionInfo()
```

```
## R version 3.2.5 (2016-04-14)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 10586)
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
##  [1] magrittr_1.5    formatR_1.3     tools_3.2.5     htmltools_0.3.5
##  [5] yaml_2.1.13     Rcpp_0.12.4     stringi_1.0-1   rmarkdown_0.9.5
##  [9] knitr_1.12.3    stringr_1.0.0   digest_0.6.9    evaluate_0.8.3
```

The source code for this document and the analysis is stored in GitHub at https://github.com/wer61537/machine_learning (https://github.com/wer61537/machine_learning). The R code is at https://github.com/wer61537/machine_learning/blob/master/machine_learning.R (https://github.com/wer61537/machine_learning/blob/master/machine_learning.R). The Markdown document is at https://github.com/wer61537/machine_learning/blob/master/machine_learning.Rmd (https://github.com/wer61537/machine_learning/blob/master/machine_learning.Rmd).

References

Ugulino, W.; Cardador, D.; Vega, K.; Velloso, E.; Milidiu, R.; Fuks, H. Wearable Computing: Accelerometers' Data Classification of Body Postures and Movements. Proceedings of 21st Brazilian Symposium on Artificial Intelligence. Advances in Artificial Intelligence - SBIA 2012. In: Lecture Notes in Computer Science. , pp. 52-61. Curitiba, PR: Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-34458-9. DOI: 10.1007/978-3-642-34459-6_6. http://groupware.les.inf.puc-rio.br/work.jsf?p1=10335#ixzz4A5kzyJaD (http://groupware.les.inf.puc-rio.br/work.jsf?p1=10335#ixzz4A5kzyJaD)

Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th Augmented Human (AH) International Conference in cooperation with ACM SIGCHI (Augmented Human'13) . Stuttgart, Germany: ACM SIGCHI, 2013. http://groupware.les.inf.puc-rio.br/work.jsf?p1=11201#ixzz4A5lFFpw7 (http://groupware.les.inf.puc-rio.br/work.jsf?p1=11201#ixzz4A5lFFpw7)