Barbell lifts classification using wearables

JoPM

18 septembre 2016

Introduction

The quantified self movement is growing. Many differents device exist to record data while training. For this project, the data was recored using accelerometers on the belt, forearm, arm, and dumbell. This report explain who a model was develop to classify barbell lift in 5 class.

General info

The data came from 6 participants who execute Unilateral Dumbbell Biceps Curl in 5 different maners : 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).

```r
# package / library
library(caret)
```

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

```
## Loading required package: ggplot2
```

```r
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
```

Data

```r
trainUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

trainset <- read.csv(url(trainUrl))
testset <- read.csv(url(testUrl))
```

Data exploration and cleaning

The training set as 160 variables. For each variable, there are 19622 observations. As for the test set, there are 20 observations with 19622 observations.

The "classe" varible is the outcome. There are 5 the different levels:

```r
levels(trainset$classe)
```

```
## [1] "A" "B" "C" "D" "E"
```

There seems to have a lot of variable with missing values. Thoses variables will be exclude from the analysis as they don't offer information to the classification model. In addition, you want to develop a model that is independant of the person. For this reason, the "user_name" variable was exclude. Other variables are excludes : "X" as it represent only the index "cvtd_timestamp" as only timestamp were keep also, many variable were eliminate as they contain "" and/or "#DIV/0!" as variable levels.

```r
# cleaning data
trainset_new <- trainset[ , apply(trainset, 2, function(x) !any(is.na(x)))]
trainset_new$user_name <- NULL
trainset_new$X <- NULL
trainset_new$cvtd_timestamp <- NULL
trainset_new$kurtosis_yaw_forearm <- NULL
trainset_new$skewness_yaw_belt <- NULL
trainset_new$kurtosis_yaw_belt <- NULL
trainset_new$amplitude_yaw_belt <- NULL
trainset_new$kurtosis_yaw_dumbbell <- NULL
trainset_new$skewness_yaw_dumbbell <- NULL
trainset_new$amplitude_yaw_dumbbell <- NULL
trainset_new$skewness_yaw_forearm <- NULL
trainset_new$amplitude_yaw_forearm <- NULL
```

To eliminate more variables to obtain a more simple model, all the variables with close to zero variance were eliminate.

```r
NZV <- nearZeroVar(trainset_new, saveMetrics=TRUE)
idx <- NZV$nzv
trainset_new <- trainset_new[!idx]

D <- dim(trainset_new)
```

After that, the new train set dimension are 19622 observations and 56 variables.

Models creation

The large size of the training set allow to do cross-validation by subsetting a training and a test set. It is usefull, as it permit to optimize the model with the training set. The test set could only be use to caracterize the final model. The cross-validation was made via random subsampling: 60% of the training data form the substrainset and the 40% remaining the subtestset.

```
inTrain = createDataPartition(trainset_new$classe, p = 0.6)[[1]]
subtrainset = trainset_new[ inTrain,]
subtestset = trainset_new[-inTrain,]

D_train <- dim(subtrainset)
D_test <- dim(subtestset)
```

The subtrain set dimension is 11776 observations and 56 variables. The subtrain set dimension is 7846 observations and 56 variables.

As there are many way to determine a predicting model,only three models were investigated. The models are “random forest”, “boosted trees” and “linear discriminant analysis”. The models use all the remaining variables to start with. For each model, the prediction was made with the subtestset.

Random forest

```
modelRF <- randomForest(classe ~ ., data = subtrainset)
predicRF <- predict(modelRF, newdata = subtestset)
confusionMatrix(predicRF, subtestset$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2226    3    0    0    0
##          B    6 1514    4    0    0
##          C    0    1 1364    3    0
##          D    0    0    0 1282    1
##          E    0    0    0    1 1441
##
## Overall Statistics
##
##                Accuracy : 0.9976
##                  95% CI : (0.9962, 0.9985)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9969
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9973   0.9974   0.9971   0.9969   0.9993
## Specificity            0.9995   0.9984   0.9994   0.9998   0.9998
## Pos Pred Value         0.9987   0.9934   0.9971   0.9992   0.9993
## Neg Pred Value         0.9989   0.9994   0.9994   0.9994   0.9998
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2837   0.1930   0.1738   0.1634   0.1837
## Detection Prevalence   0.2841   0.1942   0.1744   0.1635   0.1838
## Balanced Accuracy      0.9984   0.9979   0.9982   0.9984   0.9996
```

Boosted Trees

sorry, to long to process.

```
# modelGBM_t2 <- train(classe ~ ., method = 'gbm', data = subtrainset)
# predicGBM <- predict(modelGBM, newdata = subtestset)
# confusionMatrix(predicGBM, subtestset$classe)
```

Linear discriminant analysis

```
modelLDA <- train(classe ~ ., method = 'lda', data = subtrainset)
```

```
## Loading required package: MASS
```

```
predicLDA <- predict(modelLDA, newdata = subtestset)
confusionMatrix(predicLDA, subtestset$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1829  235  137   74   58
##          B   64  992  155   62  196
##          C  144  163  904  144  119
##          D  189   66  137  963  129
##          E    6   62   35   43  940
##
## Overall Statistics
##
##                Accuracy : 0.7173
##                  95% CI : (0.7072, 0.7273)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6422
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8194   0.6535   0.6608   0.7488   0.6519
## Specificity            0.9102   0.9246   0.9120   0.9206   0.9772
## Pos Pred Value         0.7840   0.6753   0.6133   0.6489   0.8656
## Neg Pred Value         0.9269   0.9175   0.9272   0.9492   0.9257
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2331   0.1264   0.1152   0.1227   0.1198
## Detection Prevalence   0.2973   0.1872   0.1879   0.1891   0.1384
## Balanced Accuracy      0.8648   0.7891   0.7864   0.8347   0.8145
```

Final model

The boosted trees model was eliminate as it take a very large amount of time to compute as designed. On the contrary, the linear discriminant analysis was execute very fast, but the accuracy resulting is low. The resulting chosen model is the random forest. The accuracy is very good. As the training wasn't done many time, the model should not be overfit.

The testset should be transform to contain the same variable as in the new trainset. After it is done, it is possible to apply the chosen model to the new testset in order to obtain the prediction.

```r
t <- colnames(trainset_new)
var_names <- t[1]
x <- 2:length(t)-1
for (i in x){var_names <- c(var_names, t[i])}
testset_new <- subset(testset, select=var_names)

predicTestSet <- predict(modelRF, newdata = testset_new)
```

The prediction for the 20 samples in the testset are: B, A, B, A, A, E, D, B, A, A, B, C, B, A, E, E, A, B, B, B

Expected out-of-sample error

The out-of-sample error is "1-accuracy" obtain from cross-validation. The accuracy is the proportion of correct outcome over the total data of the subtestset. The "Expected out-of-sample error" would be "1-expected accuracy" of the testing data set.

Conclusion

In conclusion, the random forest model seems the best in consideration of accuracy, complexity and time consumption. Some possible improvement would be to decrease the number of variable needed to accomplish this classification.