Machine Learning Project

Tavo

Thursday, January 28, 2016

Machine Learning Course Project

A) Project description

This project is part of the course “Practical Machine Learning” from the Data Scientist Specialization on Coursera. The objective is to apply different concepts and R packages learned during the course to a raw data set in order to qualitatively classify an excersise (weight lifting) as correctly or incorrectly executed.

B) Study design and data processing

B.1) Collection of the raw data

The datasets were downloaded from the following links:

- Train set: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)
- Test set: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

The original data was made available by the Human Activity Recognition Project, under Creative Commons license (CC BY-SA), and can be downloaded here:

- http://groupware.les.inf.puc-rio.br/static/har/dataset-har-PUC-Rio-ugulino.zip (http://groupware.les.inf.puc-rio.br/static/har/dataset-har-PUC-Rio-ugulino.zip)

The R packages required to run the code are:

```r
## Load required libraries
library(dplyr)
```

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

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.3
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.2.3
```

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

B.2) Notes on the original data

A dictionary or a full description of the variables used in the experiment wasn't found. However in the paper that describes the original experiment (http://groupware.les.inf.puc-rio.br/public/papers/2013.Velloso.QAR-WLE.pdf (http://groupware.les.inf.puc-rio.br/public/papers/2013.Velloso.QAR-WLE.pdf)) some of the variables were discused and explained. This available information is what is used to further analysis and processing of the dataset.

```r
## Download data (if required) and create two datasets
trainUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
trainName <- "pml-training.csv"

testUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
testName <- "pml-testing.csv"

if(!file.exists(trainName)){ download.file(trainUrl, destfile=trainName, method="curl")}
if(!file.exists(testName)){ download.file(testUrl, destfile=testName, method="curl")}

## Empty cells are treated as NAs
train <- read.csv(trainName, na.strings=c("","NA"))
test <- read.csv(testName, na.strings=c("","NA"))
```

B.3) Creating a tidy dataset

The variables with more than 90% of its rows as “NAs” will be droped. This operation reduces the number of columns from 160 to 60, which (hopefully) will reduce also the time required to run the model. Also the first 7 columns were removed because they are just for identification purposes and doesn't add value to resolve the problem at hand.

```
## Columns with more than 90% of NAs are droped. Also the first 7 indexing column
s.
newTrain <- train[, !(colSums(is.na(train)) > 0.9*nrow(train))]
newTrain <- newTrain[, 8:60]

newTest <- test[, !(colSums(is.na(test)) > 0.9*nrow(test))]
newTest <- newTest[, 8:60]
```

B.4) Variables used

For feature selection, many algorithms could be used (PCA - Principal Components Analysis is an example), for example in the sections 5.1 and 5.2 of the paper about HRA, the autors explain that they selected 17 features using a selection algorithm based on correlation proposed by Hall [3] and that the algorithm was configured to use a Best First strategy based on backtracking.

One of the methods (as explained on the lecture 15, covariate creation) is to study the variance of the variables and discard those wich variance is near zero (they wouldn't add value to the problem). In this case, all of the remaining variables (52) had enought variance to add value to the solution of the problem, however an aditional "selection" will be implicitly made using the principal components analysis during the preprocesing of the model in the next step.

```
## Check for near zero variance of the features
nsv <- nearZeroVar(newTrain, saveMetrics=TRUE)
print(nsv)
```

```
##                      freqRatio percentUnique zeroVar   nzv
## roll_belt             1.101904     6.7781062   FALSE FALSE
## pitch_belt            1.036082     9.3772296   FALSE FALSE
## yaw_belt              1.058480     9.9734991   FALSE FALSE
## total_accel_belt      1.063160     0.1477933   FALSE FALSE
## gyros_belt_x          1.058651     0.7134849   FALSE FALSE
## gyros_belt_y          1.144000     0.3516461   FALSE FALSE
## gyros_belt_z          1.066214     0.8612782   FALSE FALSE
## accel_belt_x          1.055412     0.8357966   FALSE FALSE
## accel_belt_y          1.113725     0.7287738   FALSE FALSE
## accel_belt_z          1.078767     1.5237998   FALSE FALSE
## magnet_belt_x         1.090141     1.6664968   FALSE FALSE
## magnet_belt_y         1.099688     1.5187035   FALSE FALSE
## magnet_belt_z         1.006369     2.3290184   FALSE FALSE
## roll_arm             52.338462    13.5256345   FALSE FALSE
## pitch_arm            87.256410    15.7323412   FALSE FALSE
## yaw_arm              33.029126    14.6570176   FALSE FALSE
## total_accel_arm       1.024526     0.3363572   FALSE FALSE
## gyros_arm_x           1.015504     3.2769341   FALSE FALSE
## gyros_arm_y           1.454369     1.9162165   FALSE FALSE
## gyros_arm_z           1.110687     1.2638875   FALSE FALSE
## accel_arm_x           1.017341     3.9598410   FALSE FALSE
## accel_arm_y           1.140187     2.7367241   FALSE FALSE
## accel_arm_z           1.128000     4.0362858   FALSE FALSE
```

```
## magnet_arm_x         1.000000     6.8239731   FALSE FALSE
## magnet_arm_y         1.056818     4.4439914   FALSE FALSE
## magnet_arm_z         1.036364     6.4468454   FALSE FALSE
## roll_dumbbell        1.022388    84.2065029   FALSE FALSE
## pitch_dumbbell       2.277372    81.7449801   FALSE FALSE
## yaw_dumbbell         1.132231    83.4828254   FALSE FALSE
## total_accel_dumbbell 1.072634     0.2191418   FALSE FALSE
## gyros_dumbbell_x     1.003268     1.2282132   FALSE FALSE
## gyros_dumbbell_y     1.264957     1.4167771   FALSE FALSE
## gyros_dumbbell_z     1.060100     1.0498420   FALSE FALSE
## accel_dumbbell_x     1.018018     2.1659362   FALSE FALSE
## accel_dumbbell_y     1.053061     2.3748853   FALSE FALSE
## accel_dumbbell_z     1.133333     2.0894914   FALSE FALSE
## magnet_dumbbell_x    1.098266     5.7486495   FALSE FALSE
## magnet_dumbbell_y    1.197740     4.3012945   FALSE FALSE
## magnet_dumbbell_z    1.020833     3.4451126   FALSE FALSE
## roll_forearm        11.589286    11.0895933   FALSE FALSE
## pitch_forearm       65.983051    14.8557741   FALSE FALSE
## yaw_forearm         15.322835    10.1467740   FALSE FALSE
## total_accel_forearm  1.128928     0.3567424   FALSE FALSE
## gyros_forearm_x      1.059273     1.5187035   FALSE FALSE
## gyros_forearm_y      1.036554     3.7763735   FALSE FALSE
## gyros_forearm_z      1.122917     1.5645704   FALSE FALSE
## accel_forearm_x      1.126437     4.0464784   FALSE FALSE
## accel_forearm_y      1.059406     5.1116094   FALSE FALSE
## accel_forearm_z      1.006250     2.9558659   FALSE FALSE
## magnet_forearm_x     1.012346     7.7667924   FALSE FALSE
## magnet_forearm_y     1.246914     9.5403119   FALSE FALSE
## magnet_forearm_z     1.000000     8.5771073   FALSE FALSE
## classe               1.469581     0.0254816   FALSE FALSE
```

C Running the Model

The question asked requires to use the given data to classify it into 5 different categories, labeled “A”, “B”, “C”, “D”, “E”, according to the original study, this labels correspond to:

- 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 halfway
- Class E: and throwing the hips to the front

To solve this problem, the random forest algorith is used. According to the material given in the course [4}] this algorithm has various desirable features to solve this problem, among them:

- It is unexcelled in accuracy among current algorithms.
- Runs efficiently on large data bases.

- Can handle many input variables without variable deletion.
- Gives estimates of what variables are important in the classification.
- There is no need for cross-validation or a separate test set to get an unbiased estimate of the test set error.

```
## Split data into train / test sets
set.seed(2442)
intrain <- createDataPartition(y=newTrain$classe, p=0.8, list=FALSE)
training <- newTrain[intrain,]
testing <- newTrain[-intrain,]

## Preprocess and run the model
preProc <- preProcess(training, method="pca", tresh=0.8)
trainPC <- predict(preProc,training)
modelFit <- train(as.factor(training$classe) ~ .,method="rf",data=trainPC)

## Test the results accuracy
testPC <- predict(preProc,testing)
print(confusionMatrix(testing$classe, predict(modelFit,testPC)))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1110    2    4    0    0
##          B   10  741    6    0    2
##          C    0   11  668    5    0
##          D    0    1   26  615    1
##          E    0    5    5    8  703
##
## Overall Statistics
##
##                Accuracy : 0.9781
##                  95% CI : (0.973, 0.9824)
##     No Information Rate : 0.2855
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9723
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9911   0.9750   0.9422   0.9793   0.9958
## Specificity            0.9979   0.9943   0.9950   0.9915   0.9944
## Pos Pred Value         0.9946   0.9763   0.9766   0.9565   0.9750
## Neg Pred Value         0.9964   0.9940   0.9873   0.9960   0.9991
## Prevalence             0.2855   0.1937   0.1807   0.1601   0.1800
## Detection Rate         0.2829   0.1889   0.1703   0.1568   0.1792
## Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      0.9945   0.9847   0.9686   0.9854   0.9951
```

The overall accuracy of the model is 0.9781, and for each class the balanced accuracy is always over 0.96. This high accuracy is expected from this type of algorithm.

After this results the validation is conducted with the validation dataset of 20 rows.

```
## Predict class with the validation set
testOut <- predict(preProc,newTest)
testOut <- predict(preProc, newTest[, -53])
answers <- predict(modelFit, testOut)
```

```
## Loading required package: 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
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
print(answers)
```

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

These answers have an accuracy of 95%, or 19 of 20 classes were correct (given by quiz results). This out of sample error is far below the estimate based on the averall accuracy (97.8%), but the cause is maybe that the validations set is too short (and one sigle error can drop the accuracy 5%).

D) References

[1] 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.

[2] 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.

[3] M. A. Hall. Correlation-based Feature Subset Selection for Machine Learning. PhD thesis, Department of Computer Science, University of Waikato, Hamilton, New Zealand, Apr. 1999.

[4] Breiman, L.; Cutler, A. Random Forests. Berkely University. http://www.stat.berkeley.edu/%7Ebreiman/RandomForests/cc_home.htm (http://www.stat.berkeley.edu/%7Ebreiman/RandomForests/cc_home.htm)