Are you doing right your exercises?

CWerneck - Claudia Werneck

january, 25, 2018

Abstract

This analysis corresponds to the Project Assignment for the **Practical Machine Learning** course of the Johns Hopkins Bloomberg School of Public Health **Data Science Specialization** at Coursera.

Using devices such as *Jawbone Up*, *Nike FuelBand*, and *Fitbit* it is now possible to collect a large amount of data about personal activity relatively inexpensively.

These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks.

One thing that people regularly do is quantify **how much** of a particular activity they do, but they rarely quantify **how well** they do it.

In this project, the goal is: using data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants, predict the manner in which they did the exercises. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways: A - the correct way and B, C, D e E, four different wrong ways of do the exercise. This is the *“classe”* variable in the training set. It will be select any of the other variables to predict with.

More information is available from the website here: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset) and if you use the document you create for this class, for any purpose, please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

The training and test data for this project are available in this two url’s:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

Data Processing

```
library(caret); library(rattle); library(rpart); library(rpart.plot)
library(randomForest); library(corrplot)
#Load the imported data from local
trainRead<-read.csv("C:/Coursera/08_Practical_Machine_learning/pml-training.csv", na.strings=c("NA","#DIV/0!", ""))
testRead<-read.csv("C:/Coursera/08_Practical_Machine_learning/pml-testing.csv", na.strings=c("NA","#DIV/0!", ""))
dim(trainRead);dim(testRead)
```

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

```
## [1]  20 160
```

```
#Once we want use the columns as predictors, we must eliminate all the columns that do no
t have information.
trainClean<-trainRead[,colSums(is.na(trainRead))==0]
testClean<-testRead[,colSums(is.na(testRead))==0]
#that reduces the columns to only 60 columns
dim(trainClean);dim(testClean)
```

```
## [1] 19622    60
```

```
## [1] 20 60
```

```
#Investigating the data we can see that the seven first columns have a sequencial number
(the first)
#and variations of the timestamp that we are not using for this analysis so we will elimi
nate those columns remaining 53
trainOK<-trainClean[,-c(1:7)]
testOK<-testClean[,-c(1:7)]
dim(trainOK);dim(testOK)
```

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

```
## [1] 20 53
```

```
#And now we are with the Dataset to proceed the study  and will see if there are correlat
ion among the variables used.
###**Create the datasets**
```

```
inTrain<-createDataPartition(trainOK$classe, p=3/4, list=FALSE)
train<-trainOK[inTrain,]
valid<-trainOK[-inTrain,]
```

analysing the principal components, we got that 25 components are necessary to capture .95 of the variance. But it demands alot of machine processing so, we decided by a .80 thresh to capture 80% of the variance using 12 components

```
set.seed(2018)
PropPCA<-preProcess(train,method="pca", thresh=0.8)
PropPCA
```

```
## Created from 14718 samples and 53 variables
##
## Pre-processing:
##   - centered (52)
##   - ignored (1)
##   - principal component signal extraction (52)
##   - scaled (52)
##
## PCA needed 13 components to capture 80 percent of the variance
```

```
#create the preProc object, excluding the response (classe)
preProc  <- preProcess(train[,-53],
                       method = "pca",
                       pcaComp = 12, thresh=0.8)
#Apply the processing to the train and test data, and add the response
#to the dataframes
train_pca <- predict(preProc, train[,-53])
train_pca$classe <- train$classe
#train_pca has only 12 principal components plus classe
valid_pca <- predict(preProc, valid[,-53])
valid_pca$classe <- valid$classe
#valid_pca has only 12 principal components plus classe

###**Choose algorithms to predict**
#####Two methods will be tested, gbm=Generalized Boosted Regression and rf=Random Forest
### GBM  produced the worst result and Once it take a loong time to reprocess, it is dumb
ed.
#fit_gbm<-train(classe ~., data=train_pca, method="gbm")
#print(fit_gbm, digits=4)
#predict_gbm<-predict(fit_gbm,valid_pca)
#(conf_gbm<-confusionMatrix(valid_pca$classe, predict_gbm))
#(accuracy_gbm<-conf_gbm$overall['Accuracy'])
###rf
fitControl<-trainControl(method="cv", number=5, allowParallel=TRUE)

fit_rf<-train(classe ~., data=train_pca, method="rf", trControl=fitControl)
print(fit_rf, digits=4)
```

```
## Random Forest
##
## 14718 samples
##    12 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 11774, 11774, 11775, 11775, 11774
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa
##    2    0.9536    0.9413
##    7    0.9495    0.9361
##   12    0.9416    0.9261
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 2.
```

```
predict_rf<-predict(fit_rf,valid_pca)
(conf_rf<-confusionMatrix(valid_pca$classe, predict_rf))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1367   10   10    5    3
##          B   17  896   31    2    3
##          C   10    9  820   12    4
##          D    7    1   36  759    1
##          E    2    9    7    4  879
##
## Overall Statistics
##
##                Accuracy : 0.9627
##                  95% CI : (0.957, 0.9678)
##     No Information Rate : 0.2861
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9528
##  Mcnemar's Test P-Value : 0.0003434
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9743   0.9686   0.9071   0.9706   0.9876
## Specificity            0.9920   0.9867   0.9912   0.9891   0.9945
## Pos Pred Value         0.9799   0.9442   0.9591   0.9440   0.9756
## Neg Pred Value         0.9897   0.9927   0.9793   0.9944   0.9973
## Prevalence             0.2861   0.1886   0.1843   0.1595   0.1815
## Detection Rate         0.2788   0.1827   0.1672   0.1548   0.1792
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      0.9832   0.9777   0.9492   0.9798   0.9911
```

```
(accuracy_rf<-conf_rf$overall['Accuracy'])
```

```
##  Accuracy
## 0.9626835
```

We can now say that for this dataset, **random forest** method is better than Generalized Boosted Regression and the **accuracy** obtained would be **0.9611**

Results - Prediction on Testing Set

Applying the **Random Forest** to predict the outcome variable classe for the **test** set

```
test_pca <- predict(preProc, testOK[,-53])
test_pca$problem_id <- testOK$problem_id
(predict(fit_rf, test_pca))
```

```
##  [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
```

with those 20 predictions we conclude the Course Project

If you have one of those devices, to get your data, try to do the exercises and discover if you are

doing them in the right way.

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. Cited by 2 (Google Scholar)

Thanks for reading!