Predict Type of Exercises

FJMA

July 26, 2014

A data set with the results of monitoring while people are doing several types of exercises is provided in [1]. In this document we try to describe an algorithm to predict what kind of exercise has been done. We are going to use the Random Forest algorithm using the commands in the caret package (see [3]).

The data set is loaded with the following instructions:

```
pml<-read.csv("pml-training.csv")
```

The file includes 19622 observations of 160 variables. Some of these variables are missed for most of the observations. We are going to remove all the variables with more than a 30% of missing data.

```
var<-apply(sapply(pml,is.na),2,sum)<19622*0.70
```

We remove the variable X, an index of the individual and the name. These two variables should not be used to predict the type of exercise. The index X is correlated with the type of exercises, but for new data the index should not be useful to select the type of exercise. We also remove the time stamp, using the time when the exercise is done is not what we want to do, we want to to determine the type of exercises with measures done by the different appliances.

```
var[1:5] <- FALSE
pml <- pml[,var]
```

The next step in the preprocessing is going to remove the variable with very low variability since it would not be very useful to predict. In order to do this we use the function nearZeroVar in the caret package

```
library(caret)
```

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

```
var2 <- nearZeroVar(pml,saveMetrics=TRUE)$nzv==FALSE
pml <- pml[,var2]
```

Since the file provided is huge we are going to use a 5% (the sample is divided in 20 groups) of the set as a training set, since the computation time will be too long if all the data set is used

```
set.seed(12345)
folds <- createFolds(y=pml$classe,k=20,list=TRUE,returnTrain=FALSE)
training <- pml[folds[[1]],]
```

We use the data in the first fold calculated in order to train our data and get a prediction model. The method choose is random forest because is one of the most accurate.

```
modFit<-train(classe ~ ., data=training,method="rf",prox=TRUE)
```

```
## Loading required package: randomForest
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
## Loading required package: class
```

```
modFit
```

```
## Random Forest
##
## 981 samples
##  53 predictors
##   5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 981, 981, 981, 981, 981, 981, ...
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa  Accuracy SD  Kappa SD
##   2     0.9       0.8    0.01         0.02
##   30    0.9       0.9    0.02         0.02
##   50    0.9       0.9    0.02         0.02
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 27.
```

We check how good is our prediction with the same data set used for training. With the contigency table we are seeing the degree of agreement of the predicted value with the real value. Parameters such as the accuracy and the kappa index indicates that in sample error or resubstitution error is almost perfect.

```
confusionMatrix(training$classe,predict(modFit,newdata=training))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   A   B   C   D   E
##          A 279   0   0   0   0
##          B   0 190   0   0   0
##          C   0   0 171   0   0
##          D   0   0   0 161   0
##          E   0   0   0   0 180
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.996, 1)
##     No Information Rate : 0.284
##     P-Value [Acc > NIR] : <2e-16
```

```
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             1.000    1.000    1.000    1.000    1.000
## Specificity             1.000    1.000    1.000    1.000    1.000
## Pos Pred Value          1.000    1.000    1.000    1.000    1.000
## Neg Pred Value          1.000    1.000    1.000    1.000    1.000
## Prevalence              0.284    0.194    0.174    0.164    0.183
## Detection Rate          0.284    0.194    0.174    0.164    0.183
## Detection Prevalence    0.284    0.194    0.174    0.164    0.183
## Balanced Accuracy       1.000    1.000    1.000    1.000    1.000
```

In sample error is excellent, but it would be because we are overfitting our model. In order to know what is the real quality of the prediction we are going to study the our of sample erros of resubstitution error. For that we consider another fold of the previously separated groups. The 95% Confidence Interval of the accuracy is above 0.96. Therefore, we can think that our prediction error is going to be lower than a 5%.

```
testing <- pml[folds[[2]],]
confusionMatrix(testing$classe,predict(modFit,newdata=testing))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   A   B   C   D   E
##          A 277   0   1   1   0
##          B   5 178   7   0   0
##          C   0  12 159   1   0
##          D   0   1   6 154   0
##          E   0   1   2   7 171
##
## Overall Statistics
##
##                Accuracy : 0.955
##                  95% CI : (0.94, 0.967)
##     No Information Rate : 0.287
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.943
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.982    0.927    0.909    0.945    1.000
## Specificity             0.997    0.985    0.984    0.991    0.988
## Pos Pred Value          0.993    0.937    0.924    0.957    0.945
## Neg Pred Value          0.993    0.982    0.980    0.989    1.000
## Prevalence              0.287    0.195    0.178    0.166    0.174
## Detection Rate          0.282    0.181    0.162    0.157    0.174
```

```
## Detection Prevalence    0.284    0.193    0.175    0.164    0.184
## Balanced Accuracy       0.990    0.956    0.946    0.968    0.994
```

Cross validation can be done the new data set provided for the assignment. In this case, the predictions are correct what it is consistent with a prediction error lower than a 5%.

```
crossVal<-read.csv("pml-testing.csv")
crossVal <- crossVal[,var]
crossVal <- crossVal[,var2]
predict(modFit,crossVal)
```

References

[1] Groupware@LES. Human Active Recognition.
http://groupware.les.inf.puc-rio.br/har

[2] 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. Read more: http://groupware.les.inf.puc-rio.br/har#ixzz38btFWMdO

[3] http://caret.r-forge.r-project.org/