movement type predictor

Renaud Viot

1/10/2018

Excecutive summary

The goal of this analysis is to predict the movement based on several indicators. We have ran two models; boosting and random forest. The random forest model gave us an accuracy of 100%, which bears a risk of overfitting the training test. Cross validated using the boosting model (accuracy of 92%), we decided to go with the random forest as our prediction model.

Background

Human Activity Recognition - HAR - has emerged as a key research area in the last years and is gaining increasing attention by the pervasive computing research community, especially for the development of context-aware systems. There are many potential applications for HAR, like: elderly monitoring, life log systems for monitoring energy expenditure and for supporting weight-loss programs, and digital assistants for weight lifting exercises.

Discovery of the data

Load needed library

In order to perform the analysis, we have used the library listed below:

Load the data

The data to use is available online whici is where we gathered the data:

```
train <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
test <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

training <- read.csv(url(train),
                     na.strings=c("NA","#DIV/0!",""))
testing <- read.csv(url(test),
                    na.strings=c("NA","#DIV/0!",""))
```

Clean the data

In order to perform the analysis, we decided to strip out any variable containing empty or n/a values. We then kept only variables that were giving an indication of movement (x,y,z):

```
trainingsmall <- subset(training,
                        select=colMeans(is.na(training)) == 0)
trainingxyz <- trainingsmall[,grepl("_x|_y|_z|classe", names(trainingsmall))]
trainingxyz$classe = factor(trainingxyz$classe)
dim(trainingxyz)
## [1] 19622    37
```

Fitting models

For this exercise, we have tried to fit two different models: - boosting (accuracy=92% - see confusion matrix in appendix) - random forest (accuracy=100% - see confusion matrix in appendix) The random forest accuracy (100%) drives us to beleive that we are bearing in the model an overfitting risk. We do not believe that combining an overfitted model with another model will be meaningful, hence we will use the random forest model with the awareness that it might be overfitted.

Appendices

Confusion matrices

Boosting

```
modfit2 <- train(classe~., method="gbm", data=trainingxyz, verbose=FALSE)
confusionMatrix(predict(modfit2, trainingxyz),
                trainingxyz$classe)
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 5420  265   80   62   30
##          B   43 3305  132   28   81
##          C   46  176 3152  206   58
##          D   68   25   46 2883   82
##          E    3   26   12   37 3356
## 
## Overall Statistics
## 
##                Accuracy : 0.9232
##                  95% CI : (0.9194, 0.9269)
##     No Information Rate : 0.2844
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 0.9028
##  Mcnemar's Test P-Value : < 2.2e-16
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9713   0.8704   0.9211   0.8965   0.9304
## Specificity            0.9689   0.9821   0.9700   0.9865   0.9951
## Pos Pred Value         0.9254   0.9209   0.8664   0.9288   0.9773
## Neg Pred Value         0.9884   0.9693   0.9831   0.9798   0.9845
## Prevalence             0.2844   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2762   0.1684   0.1606   0.1469   0.1710
## Detection Prevalence   0.2985   0.1829   0.1854   0.1582   0.1750
## Balanced Accuracy      0.9701   0.9262   0.9455   0.9415   0.9628
```

Random Forest

```
modfit3 <- train(classe~., data=trainingxyz, method='rf',
                 trControl=trainControl(method='cv'), number=3)
confusionMatrix(predict(modfit3, trainingxyz),
                trainingxyz$classe)
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 5580    0    0    0    0
##          B    0 3797    0    0    0
##          C    0    0 3422    0    0
##          D    0    0    0 3216    0
##          E    0    0    0    0 3607
## 
## Overall Statistics
## 
##                Accuracy : 1
##                  95% CI : (0.9998, 1)
##     No Information Rate : 0.2844
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2844   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2844   0.1935   0.1744   0.1639   0.1838
## Detection Prevalence   0.2844   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

Apply the fitted model to the testing set

We then applied the random forest model to the testing data set and conclude that the movement for each of the 20 samples are the following:

```
predict(modfit2, testing)
##  [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
```