Principles of machine learning course project

Marius Bakker

February 3, 2019

Principles of machine learning course project

Summary

This write-up summarizes the method and results of designing a prediction algorithem for the ‘weight lifting excercise dataset’. The goal is to predict how subjects performed during a simple weightlifting excersise. A model is estimated and tested on a large data sample and applied to a small sample as part of a quiz excercise. As will be shown a ‘random forest’ model results in a very high prediction accuracy.

Data description

The dataset consists of over 20.000 observations of various subjects performing an excercise, giving movement measures in a variety of ways.

Two data issues arise, first a number of variables are irrelevant for the estimation such as the subject name and the time of the excersise. These are removed manually. Second, a number of variables consists largely of missing variables which are removed from the dataset. The removal is done by eliminating all variables that have no values in the final quiz dataset, since they are by definition not useful for estimating the final quiz results.

The dataset is split into a training (75%) and a test set (25%) to get a full out of sample estimate. The quiz set is kept seperately for the final estimations with the model.

```
set.seed(12345)

data <- read.csv("~/Principles of machine learning assignment/pml-training.csv")

quizset <- read.csv("~/Principles of machine learning assignment/pml-testing.csv")

drop <- sapply(quizset,function(x){all(is.na(x))})

data <- data[,!drop]
data <- data[,-c(1:7)]

quizset <- quizset[,!drop]
quizset <- quizset[,-c(1:7)]

inTrain = createDataPartition(data$class, p = 3/4)[[1]]

training = data[ inTrain,]

testing = data[-inTrain,]

rm(data)
```

Model estimation

Random forest

The random forest model is one of the most powerful available prediction methods, however one of the limitations is the long estimation time. The model is estimated with 100 trees as it already results in an out-of-sample accuracy >99%.

```
setupRF <- trainControl(method="cv", number=3, verboseIter=FALSE)
modRF <- train(classe~.,data = training, method="rf", ntree = 100, trControl = setupRF, na.actio
n = "na.omit")
predRF <- predict(modRF,testing)
confmatRF <- confusionMatrix(predRF,testing$classe)
confmatRF
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1395    9    0    0    0
##          B    0  933    4    0    0
##          C    0    7  847   10    2
##          D    0    0    4  794    5
##          E    0    0    0    0  894
##
## Overall Statistics
##
##                Accuracy : 0.9916
##                  95% CI : (0.9887, 0.994)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9894
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9831   0.9906   0.9876   0.9922
## Specificity            0.9974   0.9990   0.9953   0.9978   1.0000
## Pos Pred Value         0.9936   0.9957   0.9781   0.9888   1.0000
## Neg Pred Value         1.0000   0.9960   0.9980   0.9976   0.9983
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2845   0.1903   0.1727   0.1619   0.1823
## Detection Prevalence   0.2863   0.1911   0.1766   0.1637   0.1823
## Balanced Accuracy      0.9987   0.9911   0.9930   0.9927   0.9961
```

Predicting the quizset

Using the random forest method estimated above, the quiz set is tested giving twenty predictions.

```
QuizPredict <- predict(modRF,quizset)
QuizPredict
```

```
##  [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
```