Practical Machine Learning Course Project

Tyler T

2018-07-29

Course Project Overview

The objective of this course is to create a prediction model that can accurately classify the proper form used in weight lifting using sensor data. Six participants were fitted with sensors and asked to perform dumbbell curls in one of five different ways, where classification “A” is the correct form and “B” through “E” each representing different but common technique errors for curls.

The background to the research and sensor data can be found at this link:
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)

A Random Forest model is implemented using the Caret package to achieve approximately 2% in-sample error and 3% out-of-sample error when classifying the movements observed by the sensor data.

Therefore, using 52 variables from the sensor data, we are able to consistently predict the “correctness” of dumbbell arm curls with expected 97% accuracy.

Obtain the prepared training and testing data set

```
training <- read.csv(url("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-traini
ng.csv"))
testing <- read.csv(url("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testin
g.csv"))

#For reproducibility, set the random number seed explicitly
set.seed(12345)
```

Preprocessing

Preprocess the data aid in the model creation process, as there are a large number of predictors that may either slow the training process or add undesirable noise to the data.

1. Remove columns that are not relevant to the prediction and may skew the model (e.g. timestamp, row number)
2. Remove near-zero-variance columns. These will add no value to the ML model
3. Remove columns that aren’t complete.

4. Finally, split the training data into a training and validation set. This is required to assess both in- and out-of-sample error before proceeding with the model. Otherwise there is a risk that the model is overfitted to specific observations in the training dataset.

Model Creation

Select a Random Forest (RF) approach as this is a classification problem. By averaging across multiple classification trees, the RF model will likely be less biased than a single decision tree approach. Due to requiring more processing time, run the RF model with parallel processing to speed up the training process.

Cross validation

Apply k-fold cross validation to further reduce the risk of overfitting, by training on multiple “slices” of the training set. Implement cross validation as a fit control parameter in the caret train function, using 5 cross-validation iterations.

```
# Setup parallel processing
cluster <- makeCluster(detectCores() - 1) # convention to leave 1 core for OS
registerDoParallel(cluster)

# Try random forest - requires predictors not to be null
fitControl <- trainControl(method="cv", number=5, allowParallel = TRUE)
rf_fit5 <- train(classe~., data=minitrainset, method="rf",prox=TRUE, trcontrol=fitContro
l)

# Unload parellel processing
stopCluster(cluster)
registerDoSEQ()
```

Results

Test the accuracy of the fitted model (“rf_fit5”) on the training data and the validation data in order to assess in and out of sample errors before proceeding further with the model.

```
# Assess accuracy of prediction (both in sample and out of sample)
# In Sample Error (on training set)
rf_pred_train <- predict(rf_fit5, subset(trainset, select=-c(classe)))
iSampleAcc <- confusionMatrix(rf_pred_train,trainset$classe)$overall[1]
1-iSampleAcc
```

```
##   Accuracy
## 0.02004076
```

```
# Out of Sample Error (on validation set)
rf_pred_valid <- predict(rf_fit5, subset(validset, select=-c(classe)))
oSampleAcc <- confusionMatrix(rf_pred_valid,validset$classe)$overall[1]
1-oSampleAcc
```

```
##   Accuracy
## 0.02956921
```

With an out-of-sample error 97%, expect the model to be an accurate predictor for the correctness of the dumbbell curl technique for any test set of data.