output
html_document
fig_width
6

Human Activity Recognition

Hossein Parsa

Introduction

Human Activity Recognition has traditionally been researched with focus on predicting the time that activities were performed. Another approach is to investigate how well an activity was performed^1. Recently collecting this data has become easy due to development and popularity of wearable devices.

Getting Data and Processing

The Data set is accessible at HAR or Human Activity Recognition. Also more information about the research and methodology is provided at HAR website. The data set contains 19622 records categorized between 5 different classed identified as A, B, C, D and F based on how well the exercises have been performed measured by 160 features. We first read the data and label null, NAs and divided by zero values as missing value. Then split the data into training and testing data.

```
library(caret)
library(ggplot2)
set.seed(0)
fullSet<-read.csv("pml-training.csv", na.string=c("", "NA", "#DIV/0!"))

inTrain <- createDataPartition(y=fullSet$classe, p=0.7, list=FALSE)
data<-fullSet[inTrain,]
testData<-fullSet[-inTrain,]
```

A summary of data shows that some of the features are missing large number of values. Because of low variability in those features, we remove them from The data set and training process.

```
drops<-c()
for (i in 1:dim(data)[2]) if (mean(is.na(data[,i]))>.5) drops <- c(drops, names(data)[i])
drops <- c(drops, "new_window")
data<-data[, !(colnames(data) %in% drops)]
```

```
data<-data[,6:59]
dim(data)
```

Checking for near zero variation among the rest of the features.

```
nearZeroVar(data)
```

```
sum(complete.cases(data))
```

Checking correlation between features.

```
Corr <- cor(data[, names(data) != "classe"])
mcol <- colorRampPalette(c("blue", "white", "red"))(n = 200)
heatmap(Corr, col = mcol)
```

Training

For building the prediction model and training, we use `Gradient Boosting Method` or GBM. We also use `repeated 2-fold cross validation` to reduce the `out of sample error` i.e. to reduce the error when the model is applied to a new data set. Preprocessing methods `center` and `scale` are also included in the training function to normalize the features' data.

```
gbmGrid <-  expand.grid(interaction.depth = c(1, 5, 9),
                        n.trees = (1:30)*50,
                        shrinkage = 0.1)

fitControl <- trainControl(## 2-fold CV,
  method = "repeatedcv",
  number = 2,
  ## repeated 2 times
  repeats = 2,
  allowParallel = TRUE,
  classProbs = TRUE)

gbmFit <- train(classe ~ ., data = data,
                 method = "gbm",
                 trControl = fitControl,
                 preProc = c("center", "scale"),
                 metric = c("ROC","Kappa"),
                 tuneGrid = gbmGrid,
                 verbose = F)
```

Cross Validation Acuracy

The following graph provides `in samsple` accuracy with respect to `Max Tree Depth` and `Boosting Iterations` based on 2-fold cross validation.

```
ggplot(gbmFit)
```

Predicting

Predicting the test data set and evaluating the performance of learning process on this data set also Provides `out of sample accuracy` .

```
Pred <- predict(gbmFit, newdata = testData)
confusionMatrix(Pred, testData$classe)
```

As presented by the output of confusionMatrix function, the accuracy is 99.97%.

Now applying the model to the unseen test set we get the following result.

```
testSet<-read.csv("pml-testing.csv", na.string=c("", "NA", "#DIV/0!"))
predict(gbmFit, newdata = testSet)
```