project

Marco

17 de mayo de 2015

Downloading data

The data were downloaded with the next commands

URL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"

download.file(URL, destfile = "training.csv", method="curl")

URL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

download.file(URL, destfile = "test.csv", method="curl")

Processing the data, I deleted the first column because it contained a consecutive number unuseful for the algoritm.

```
data <- read.csv("training.csv", na.strings=c("NA", "#DIV/0!", ""), stringsAsFactors = T)
data <- data[,-1]
##classes <- sapply(data, class)
```

The code is very slow, and I see that there are many columns with NA, I looked for the sums of NA with this code colSums(is.na(data)), and I found that there are many columns with more than 19000 NA's, so I deleted the columns with more than 19 NAs, so I have 67 variables less (only 59 and before I had 160)

```
##data <- data[,colSums(is.na(data))<19000]
colswithNA <- colSums(is.na(data))<19000
data <- data[,colswithNA]
```

We are going to predict the classe. First I selected my training and test data set

```
require(caret)
```

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

```
##require(grid); require(doMC);registerDoMC(6)
train=sample(1:nrow(data),11776)
inTrain <- createDataPartition(y=data$classe, p=0.6, list=F)
training <- data[inTrain,]
testing <- data[-inTrain,]
##modFit <- train(classe ~ ., method="rf", data=training, prox=T)
##modFit
```

Caret pakage was very slowly, so I used the RandomForest package to process my model

```
set.seed(100)
library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
model2 <- randomForest(classe~., data=training)
model2
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = training)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.14%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3347    1    0    0    0 0.0002986858
## B    0 2279    0    0    0 0.0000000000
## C    0    5 2048    1    0 0.0029211295
## D    0    0    5 1923    2 0.0036269430
## E    0    0    0    2 2163 0.0009237875
```

The OOB estimate of error rate was 0.18%, maybe this OOB error rate underestimate the true error

Accuracy with the testing set

```
set.seed(100)
predicted <- predict(model2,testing)
table(testing$classe, predicted)
```

```
##    predicted
##        A    B    C    D    E
##   A 2231    1    0    0    0
##   B    4 1514    0    0    0
##   C    0    1 1366    1    0
##   D    0    0    2 1284    0
##   E    0    0    0    0 1442
```

This model is very accurate, I got only 3 errors from 7846 tests. So I can perform the submissions

Getting the answers for the submissions (these data were processed previously)

```
##predicted <- predict(model2,test)
##predicted
```

```
##pml_write_files = function(x){
#  n = length(x)
#  for(i in 1:n){
#    filename = paste0("problem_id_",i,".txt")
#    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col##.names=FALSE)
 # }
#}
```

```
#pml_write_files(predicted)
```