“Practical Machine Learning” Course project

Introduction

Our data have already been broken into training and test sets. The first step is to get them. I’ve already downloaded them, but the code to do so is below.

(For a more detailed introduction, see the README)

```
if(file.exists("data/pml-training.csv")) {
    print("File already downloaded")
} else {
    trainingURL <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
    download.file(trainingURL, "data/pml-training.csv")
}
```

```
## [1] "File already downloaded"
```

```
if(file.exists("data/pml-testing.csv")) {
    print("File already downloaded")
} else {
testingURL <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
download.file(testingURL, "data/pml-testing.csv")
}
```

```
## [1] "File already downloaded"
```

```
library(caret)
```

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

We’ll ignore the test data until we’ve selected a model. After reading in the training data, there’s a little tidying to do:

- Replace empty cells and cells with errors with `NA` s
- Remove rows that only have data for the summary variables

```r
train.data <- read.csv("data/pml-training.csv")
train.data[train.data==""] <- NA
train.data[train.data=="#DIV/0!"] <- NA
na.rows <- apply(train.data, 1, function(x){sum(is.na(x))})
na.rm.index <- which(is.na(train.data[1,]))
train.data <- train.data[, -na.rm.index]
```

In order to evaluate our model before submitting it for grading, we'll designate a partition of it for validation

```r
set.seed(12345)
inTrain <- createDataPartition(y=train.data$classe,p=0.7, list=FALSE)
train.data.train <- train.data[inTrain,]
train.data.test <- train.data[-inTrain,]
```

Now that the data are prepared, fitting the model is fairly straightforward. Ignoring user, window and time data, I built a random forest model training `classe` on all of the motion sensor variables, with mostly default arguments. I anticipated that this crude model would need to be adjusted, but it actually performs very well.

```r
set.seed(12345)
modFit <- train(classe ~ ., data=train.data.train[,8:60], method="rf", prox=TRUE)
```

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

I was initially worried the model was overfitting the training set; however, it performs very well on the validation data, and was 100% accurate on the test data.

```r
save(modFit, file = "mod2.rda")
pred.train <- predict(modFit, train.data.train)
```

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

```r
train.data.train$predRight <- pred.train==train.data.train$classe
table(pred.train, train.data.train$classe)
```

```
##
## pred.train    A    B    C    D    E
##          A 3906    0    0    0    0
##          B    0 2658    0    0    0
##          C    0    0 2396    0    0
##          D    0    0    0 2252    0
##          E    0    0    0    0 2525
```

```
pred.test <- predict(modFit,train.data.test)
train.data.test$predRight <- pred.test==train.data.test$classe
table(pred.test,train.data.test$classe)
```

```
##
## pred.test    A    B    C    D    E
##         A 1673   13    0    0    0
##         B    1 1122   14    0    0
##         C    0    4 1008   27    0
##         D    0    0    4  937    3
##         E    0    0    0    0 1079
```

At a glance you can tell that the in-sample accuracy 100%. It perhaps overfits the data, but it is also has 99% out-of-sample accuracy.

```
sum(train.data.test$predRight)/nrow(train.data.test)
```

```
## [1] 0.988785
```

And finally, apply the model to the testing set and export the `.txt` files

```
setwd("data")
test.data <- read.csv("pml-testing.csv")
test.data[test.data==""] <- NA
test.data[test.data=="#DIV/0!"] <- NA
test.data <- test.data[, -na.rm.index]
answers = predict(modFit, test.data)
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(answers)
```