machineLearning_courseProject

Li Jiang

Sunday, December 21, 2014

Summary

Large amount of data was collected from devices like Jawbone Up and Nike FuelBand about personal activies to monitor personal activity, regarding both what they do and whether do they do it right. The dataset to analyze consist of data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants to monitor whether they did the lifting correctly and what common mistakes they made. The collected data was used to predict which forms of lifting was performed. After removing redunant and unrelevant variables, I performed both random forest and booting on the training set and got about 0.95 accuracy by 5-fold cross validation for either of them, which is the out of sample error. Then I used these 2 models to predict lifting activites on the testing set and got 18/20 correct, which corresponded to about 90% accuracy, slightly lower than the training set, but expected.

Read in and clear the data

First I read in the training and testing data and only keep variables with sufficient variation and remove variables with considerable amount of NA or blank values, as well as variables apparently not relevant to the prediction, like identity of the performers as well as when they performed the activites. After that, only 55 varialbes were kept and would be used for building prediction model.

```
## load data and library
data(mtcars)
library(car)

setwd("D:/R/R-3.1.2/wd/machineLearning/")
library(caret)
```

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

```
library(ggplot2)
set.seed(123)
training <- read.csv("D:/R/R-3.1.2/wd/machineLearning/pml-training.csv")
testing <- read.csv("D:/R/R-3.1.2/wd/machineLearning/pml-testing.csv")
noNA <- c(4,7,8,9,10,11,37,38,39,40,41,42,43,44,45,46,47,48,49,60,61,62,63,64,65,66,67,68,84,85,86,102,
trainingSub <- training[ ,noNA]
testingSub <- testing[ ,noNA]
testingSub <- testingSub[ ,-length(noNA)]
length(noNA)
```

```
## [1] 55
```

Sample the training set to make prediction model

There are more than 19000 measurements, which significantly slowed down building the prediction model, which is true for both random Forest and Booting, so I randomly sampled 10% of the training data to build initial prediction model, which strongly accelerate the processing time.

```
inTrain <- createDataPartition(trainingSub$classe,p=0.1,list=FALSE)
trainingSubSample <- trainingSub[inTrain,]
```

Model selection

Since the outcome to predict is a classification, rather than a continous measurements, linear regression is not suitable. Since there are more than 2 options for the outcome, generalized linear prediction by binomial distribution is not applicable. As a result, I considered either random forest or booting, which are two most powerful prediction algorithm for classification prediction. I first tried random forest with 5-fold cross validation and got ~96% out of sample error based on cross validation.

```
control <- trainControl(method = "cv", number = 5)
modFit2 <- train(classe~.,method="rf",data=trainingSubSample,trControl = control,prox=FALSE)
```

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

```
pred2<-predict(modFit2,testingSub)
modFit2
```

```
## Random Forest
##
## 1964 samples
##   54 predictor
##    5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
##
## Summary of sample sizes: 1571, 1571, 1572, 1571, 1571
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa  Accuracy SD  Kappa SD
##    2    0.935     0.917  0.01011      0.01282
##   28    0.959     0.948  0.00421      0.00529
##   54    0.955     0.943  0.00526      0.00665
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 28.
```

Then I tried booting with package "gbm" (booting with trees from caret package), also with 5-fold cross validation and got ~94% out of sample error, which is very similar to the random forest prediction.

```
modFit3 <- train(classe~.,method="gbm",data=trainingSubSample,trControl = control,verbose = FALSE)
```

```
## Loading required package: gbm
## Loading required package: survival
## Loading required package: splines
```

```
##
## Attaching package: 'survival'
##
## The following object is masked from 'package:caret':
##
##     cluster
##
## Loading required package: parallel
## Loaded gbm 2.1
## Loading required package: plyr
```

```
pred3<-predict(modFit3,testingSub)
modFit3
```

```
## Stochastic Gradient Boosting
##
## 1964 samples
##   54 predictor
##    5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
##
## Summary of sample sizes: 1571, 1571, 1573, 1570, 1571
##
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy  Kappa  Accuracy SD  Kappa SD
##   1                   50      0.751     0.684  0.02940      0.0371
##   1                  100      0.810     0.759  0.02764      0.0350
##   1                  150      0.839     0.796  0.01443      0.0182
##   2                   50      0.851     0.811  0.01758      0.0222
##   2                  100      0.899     0.872  0.01522      0.0193
##   2                  150      0.919     0.897  0.01486      0.0188
##   3                   50      0.895     0.867  0.01081      0.0137
##   3                  100      0.934     0.917  0.01196      0.0152
##   3                  150      0.944     0.929  0.00796      0.0101
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were n.trees = 150,
##  interaction.depth = 3 and shrinkage = 0.1.
```

The out of sample error suggests that random forest and booting have equally good performance, and hopefully no strong overfitting, so I decide that either model is OK

```
pred2
```

```
##  [1] B A B A A E D D A A C C B A E E A B B B
## Levels: A B C D E
```

```
pred3
```

```
##  [1] B A B A A E D D A A B C B A E E A B B B
## Levels: A B C D E
```

```
identical(pred2,pred3)
```

```
## [1] FALSE
```

Check on test set

I performed a one time check on the test set and got the same result from both random forest and booting. 18/20 was correct, corresponding to a 90% accuracy, slightly worse than the 95% of training set, but still good. It indicated that I don't have strong overfitting or underfitting. The 2 approaches shared the same mistakes (the 8th and 16th of the training set), probably indicating some trace of systematic bias.

Conclusion

Both random forest and booting have around 95% accuracy in the training set and 90% of test set, indicating equally strong predictive power of these 2 approaches, in predicting human activity from different activity parameters.