Random Forest

Calvin Seto

January 14, 2016

Random Forest

mtry = number of variables randomly sampled as candidates at each split Defaults: Classification sqrt(p) Regression p/3 where p = number of variables

```
library(caret)
```

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

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

```
setwd("~/Dropbox/jhudatascience/8_Practical_Machine_Learning/CourseProject")

# setwd("C:/Users/Calvin/Calvinsbiz/Dropbox/jhudatascience/8_Practical_Machine_Learning/CourseProject")

pmlTrain1 <- read.csv("data/pml-training.csv", stringsAsFactors = FALSE, na.strings = c("#DIV/0!","","NA

pmlTrain1MissingCounts <- sapply(pmlTrain1, function(x)sum(is.na(x)))
pmlTrain1Complete <- pmlTrain1MissingCounts[pmlTrain1MissingCounts==0]
pmlTrain2 <- pmlTrain1[,names(pmlTrain1Complete)]

inTrain <- createDataPartition(y=pmlTrain2$classe,
                               p=0.75, list=FALSE)
training <- pmlTrain2[inTrain,]
testing <- pmlTrain2[-inTrain,]

predictors <- training[,8:59]
outcome <- as.factor(training[,60])

# configure parallel
library(parallel)
library(doParallel)
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

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


# seed
set.seed(168)
```

```r
# default is bootstrap
fitRFControl <- trainControl(method="cv",
                             number=10
)

# fitRFGrid <- expand.grid(mtry=
# )

"Start Time "; Sys.time()
```

```
## [1] "Start Time "
```

```
## [1] "2016-01-14 13:08:39 EST"
```

```r
fitRF <- train(x=predictors,
             y=outcome,
             data=training,
             method="rf",
             trControl=fitRFControl
)
```

```
## Loading required package: randomForest
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
"End Time "; Sys.time()
```

```
## [1] "End Time "
```

```
## [1] "2016-01-14 13:46:09 EST"
```

```r
stopCluster(cluster)

# show model summary
fitRF
```

```
## Random Forest
##
## 14718 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 13246, 13246, 13247, 13247, 13246, 13246, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa      Accuracy SD  Kappa SD
##    2    0.9933417  0.9915769  0.002455430  0.003106666
##   27    0.9923902  0.9903734  0.001777764  0.002249025
##   52    0.9841686  0.9799700  0.002379580  0.003012168
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 2.
```

```
fitRF$resample
```

```
##     Accuracy     Kappa Resample
## 1  0.9945652 0.9931262   Fold02
## 2  0.9945652 0.9931247   Fold01
## 3  0.9952413 0.9939804   Fold03
## 4  0.9870924 0.9836709   Fold06
## 5  0.9918478 0.9896847   Fold05
## 6  0.9952413 0.9939794   Fold04
## 7  0.9938900 0.9922696   Fold07
## 8  0.9925221 0.9905395   Fold10
## 9  0.9945652 0.9931257   Fold09
## 10 0.9938859 0.9922674   Fold08
```

```
confusionMatrix.train(fitRF)
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentages of table totals)
##
##           Reference
## Prediction    A    B    C    D    E
##          A 28.4  0.1  0.0  0.0  0.0
##          B  0.0 19.2  0.1  0.0  0.0
##          C  0.0  0.0 17.3  0.3  0.0
##          D  0.0  0.0  0.0 16.1  0.0
##          E  0.0  0.0  0.0  0.0 18.3
```

```
# Make predictions and make table

pred <- predict(fitRF,testing)
testing$predRight <- pred==testing$classe
table(pred,testing$classe)
```

```
##
## pred    A    B    C    D    E
##    A 1395    3    0    0    0
##    B    0  945    7    0    0
##    C    0    1  848   16    0
##    D    0    0    0  788    1
##    E    0    0    0    0  900
```