Bagging

Nik Bear Brown

In this lesson we'll learn the how to implement Bagging in R.

Additional packages needed

To run the code you may need additional packages.

- If necessary install the followings packages.

```
install.packages('randomForest');
install.packages('caret');
install.packages('rpart');
install.packages('adabag');
install.packages('ipred');
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
library(caret)
```

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

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

```
##
## Attaching package: 'ggplot2'
```

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

```
library(rpart)
library(adabag)
```

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

```
## Loading required package: doParallel
```

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

```
## Loading required package: parallel
```

```
library(ipred)
```

```
##
## Attaching package: 'ipred'

## The following object is masked from 'package:adabag':
##
##     bagging
```

Data

We will be using the UCI Machine Learning Repository: Adult Data to predict whether income exceeds $50K/yr based on census data. Also known as "Census Income" dataset.

```
data_url <-
'http://nikbearbrown.com/YouTube/MachineLearning/M09/adult.data.txt'
# Adult data set from UCI
adult<- read.csv(url(data_url), header=FALSE)
head(adult)

##   V1                 V2     V3         V4 V5                   V6
## 1 39          State-gov  77516  Bachelors 13        Never-married
## 2 50  Self-emp-not-inc  83311  Bachelors 13   Married-civ-spouse
## 3 38            Private 215646    HS-grad  9             Divorced
## 4 53            Private 234721       11th  7   Married-civ-spouse
## 5 28            Private 338409  Bachelors 13   Married-civ-spouse
## 6 37            Private 284582    Masters 14   Married-civ-spouse
##                    V7             V8     V9     V10  V11 V12 V13
## 1        Adm-clerical  Not-in-family  White    Male 2174   0  40
## 2     Exec-managerial        Husband  White    Male    0   0  13
## 3   Handlers-cleaners  Not-in-family  White    Male    0   0  40
## 4   Handlers-cleaners        Husband  Black    Male    0   0  40
## 5      Prof-specialty           Wife  Black  Female    0   0  40
## 6     Exec-managerial           Wife  White  Female    0   0  40
##              V14    V15
## 1  United-States  <=50K
## 2  United-States  <=50K
## 3  United-States  <=50K
## 4  United-States  <=50K
## 5           Cuba  <=50K
## 6  United-States  <=50K

names(adult)

##  [1] "V1"  "V2"  "V3"  "V4"  "V5"  "V6"  "V7"  "V8"  "V9"  "V10" "V11"
## [12] "V12" "V13" "V14" "V15"

adult.len <- sample(1:nrow(adult), 3*nrow(adult)/4)
head(adult.len)

## [1] 28326  7840  8238  3136 24634  5512
```

```r
train <- adult[adult.len,]
test <- adult[-adult.len,]
head(train)
```

```
##       V1             V2     V3           V4 V5                   V6
## 28326 57  Self-emp-inc 127728  Prof-school 15  Married-civ-spouse
## 7840  33              ? 202498      7th-8th  4           Separated
## 8238  47       Private 586657      Masters 14  Married-civ-spouse
## 3136  60       Private 191188      HS-grad  9  Married-civ-spouse
## 24634 33       Private  58305    Assoc-voc 11  Married-civ-spouse
## 5512  48       Private  94461      HS-grad  9             Widowed
##                       V7             V8     V9     V10   V11 V12 V13
## 28326     Prof-specialty        Husband  White    Male 15024   0  60
## 7840                   ?  Not-in-family  White    Male     0   0  40
## 8238     Exec-managerial        Husband  White    Male     0   0  40
## 3136    Transport-moving        Husband  White    Male     0   0  40
## 24634       Craft-repair        Husband  White    Male     0   0  40
## 5512   Machine-op-inspct  Not-in-family  White  Female     0   0  16
##                  V14    V15
## 28326  United-States   >50K
## 7840       Guatemala  <=50K
## 8238           Japan   >50K
## 3136   United-States  <=50K
## 24634  United-States  <=50K
## 5512   United-States  <=50K
```

```r
head(test)
```

```
##    V1               V2     V3          V4 V5                     V6
## 1  39         State-gov  77516   Bachelors 13          Never-married
## 2  50  Self-emp-not-inc  83311   Bachelors 13     Married-civ-spouse
## 3  38           Private 215646     HS-grad  9               Divorced
## 7  49           Private 160187         9th  5  Married-spouse-absent
## 8  52  Self-emp-not-inc 209642     HS-grad  9     Married-civ-spouse
## 14 32           Private 205019  Assoc-acdm 12          Never-married
##                    V7             V8     V9     V10  V11 V12 V13
## 1        Adm-clerical  Not-in-family  White    Male 2174   0  40
## 2     Exec-managerial        Husband  White    Male    0   0  13
## 3   Handlers-cleaners  Not-in-family  White    Male    0   0  40
## 7       Other-service  Not-in-family  Black  Female    0   0  16
## 8     Exec-managerial        Husband  White    Male    0   0  45
## 14              Sales  Not-in-family  Black    Male    0   0  50
##                V14    V15
## 1   United-States  <=50K
## 2   United-States  <=50K
## 3   United-States  <=50K
## 7         Jamaica  <=50K
## 8   United-States   >50K
## 14  United-States  <=50K
```

Bootstrap aggregating (bagging)

Create ensembles by bootstrap aggregation, i.e., repeatedly randomly re-sampling training data. Not that bagging uses the same learner so bias related to the method isn't addressed by this approach.

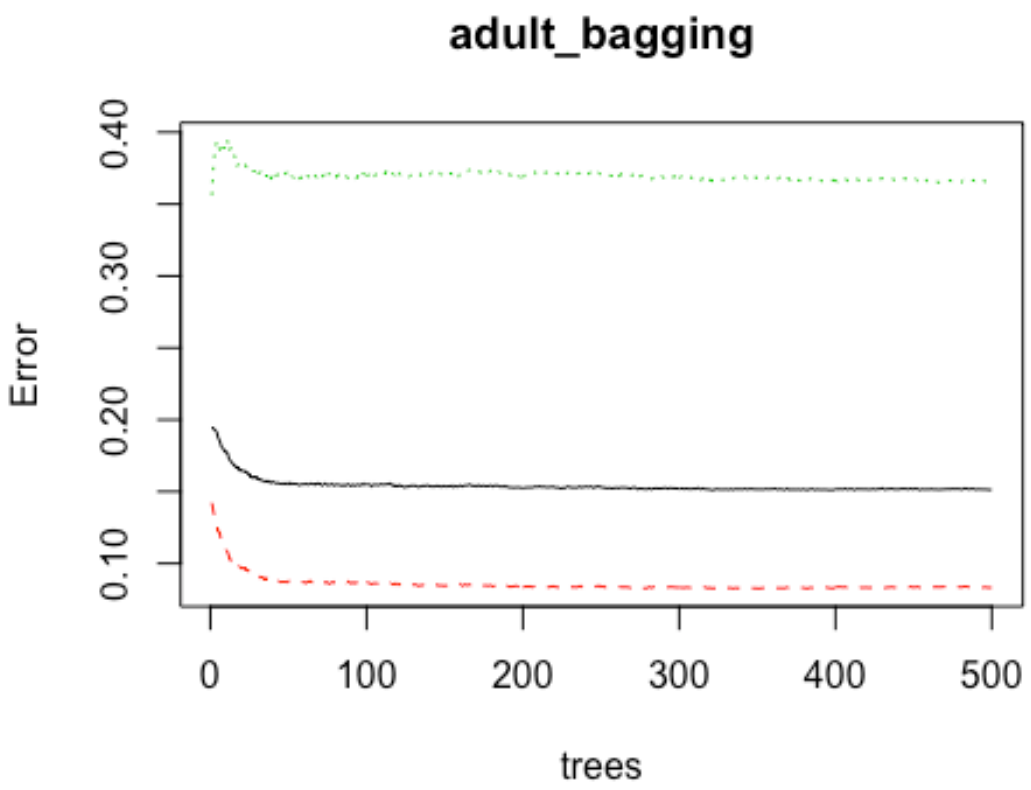
Bootstrap: draw n items from X with replacement

Bootstrap aggregating: combines random learners (often with voting, averaging or median) to create a predictor lesss efected by noise. Unstable and/or noisy algorithms often profit from bagging.

Bagging's usefulness depends on the stability of the base classifiers. If small changes in the sample cause small changes in the base-level classifier, then the ensemble will not be much better than the base classifiers. It reduces variance and helps to avoid overfitting. It is often applied to decision tree methods (random forests) and nearest neighbor classifiers, but it can be used with any type of method.

Bagging in R

```
adult_bagging <- randomForest(V15~.,data=adult, subset=adult.len, mtry=14,
importance=TRUE)
plot(adult_bagging)
```




```
adult_predict <- predict(adult_bagging, test)
adult_predict_confusion <- confusionMatrix(adult_predict, test$V15)
adult_predict_confusion$table
```

```
##            Reference
## Prediction  <=50K  >50K
##      <=50K   5703   670
##      >50K     482  1286
```

```
accuracy <- adult_predict_confusion$overall[1]
accuracy
```

```
## Accuracy
## 0.858494
```

```
# importance of predictors
adult_bagging$importance
```

```
##            <=50K          >50K MeanDecreaseAccuracy MeanDecreaseGini
## V1  0.0022848823  5.918145e-02         0.0160131960       1078.17024
## V2  0.0041011821  2.735627e-03         0.0037718759        320.63197
## V3  0.0003918638 -2.813012e-05         0.0002915477       1610.98068
## V4  0.0194597138 -8.481591e-05         0.0147455717        380.89605
## V5  0.0246041622  1.439993e-02         0.0221364814        697.62623
```

```
## V6  0.0298989987 -4.784828e-03         0.0215260732         76.84384
## V7  0.0132750339  4.614383e-02         0.0212044494        715.02659
## V8  0.0517406634  8.072236e-02         0.0587201955       1845.12548
## V9  0.0003970966  1.160936e-03         0.0005814746         94.31887
## V10 0.0041548991 -4.956942e-04         0.0030323091         55.25202
## V11 0.0339307921  6.959970e-02         0.0425368786        920.22967
## V12 0.0047160754  2.706026e-02         0.0101082449        295.44377
## V13 0.0033366583  2.498712e-02         0.0085583225        624.82185
## V14 0.0016470962 -1.461531e-03         0.0008970615        210.43013
```

```
# ipred package
adult_bagging <- ipredbagg(train$V15, X=train[,-15], nbagg=25,
                           control=rpart.control(minsplit=2, cp=0, xval=0),
                           comb=NULL, coob=FALSE, ns=length(train$V15), keepX
= TRUE)
adult_predict <- predict(adult_bagging, test)
adult_predict_confusion <- confusionMatrix(adult_predict, test$V15)
adult_predict_confusion$table
```

```
##            Reference
## Prediction  <=50K  >50K
##      <=50K   5715   673
##      >50K     470  1283
```

```
accuracy <- adult_predict_confusion$overall[1]
accuracy
```

```
##  Accuracy
## 0.8595996
```

Resources

- [Improve Predictive Performance in R with Bagging via @rbloggers](http://www.r-bloggers.com/improve-predictive-performance-in-r-with-bagging/)
- bagging {adabag} | inside-R | A Community Site for R
- bagging {ipred} | inside-R | A Community Site for R
- Bagging / Bootstrap Aggregation with R