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 4.6-12

## 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: mlbench

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](https://archive.ics.uci.edu/ml/datasets/Adult) 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] 8718 21045 17531 13301 28165 8835

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

## V1 V2 V3 V4 V5 V6  
## 8718 64 State-gov 114650 9th 5 Married-civ-spouse  
## 21045 60 Self-emp-inc 181196 Some-college 10 Married-civ-spouse  
## 17531 38 Private 220783 HS-grad 9 Divorced  
## 13301 37 Private 240810 Assoc-acdm 12 Married-civ-spouse  
## 28165 55 ? 141807 HS-grad 9 Never-married  
## 8835 23 Local-gov 144165 Bachelors 13 Never-married  
## V7 V8 V9 V10 V11  
## 8718 Craft-repair Husband White Male 0  
## 21045 Exec-managerial Husband White Male 0  
## 17531 Other-service Not-in-family White Female 0  
## 13301 Craft-repair Husband White Male 0  
## 28165 ? Not-in-family White Male 13550  
## 8835 Prof-specialty Own-child Amer-Indian-Eskimo Male 0  
## V12 V13 V14 V15  
## 8718 0 40 United-States <=50K  
## 21045 0 40 United-States >50K  
## 17531 0 20 United-States <=50K  
## 13301 0 45 United-States <=50K  
## 28165 0 40 United-States >50K  
## 8835 0 30 United-States <=50K

head(test)

## V1 V2 V3 V4 V5 V6  
## 2 50 Self-emp-not-inc 83311 Bachelors 13 Married-civ-spouse  
## 8 52 Self-emp-not-inc 209642 HS-grad 9 Married-civ-spouse  
## 20 43 Self-emp-not-inc 292175 Masters 14 Divorced  
## 24 43 Private 117037 11th 7 Married-civ-spouse  
## 25 59 Private 109015 HS-grad 9 Divorced  
## 29 39 Private 367260 HS-grad 9 Divorced  
## V7 V8 V9 V10 V11 V12 V13  
## 2 Exec-managerial Husband White Male 0 0 13  
## 8 Exec-managerial Husband White Male 0 0 45  
## 20 Exec-managerial Unmarried White Female 0 0 45  
## 24 Transport-moving Husband White Male 0 2042 40  
## 25 Tech-support Unmarried White Female 0 0 40  
## 29 Exec-managerial Not-in-family White Male 0 0 80  
## V14 V15  
## 2 United-States <=50K  
## 8 United-States >50K  
## 20 United-States >50K  
## 24 United-States <=50K  
## 25 United-States <=50K  
## 29 United-States <=50K

# Bootstrap aggregating (bagging)

Create ensembles by [bootstrap aggregation](https://en.wikipedia.org/wiki/Bootstrap_aggregating), 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.

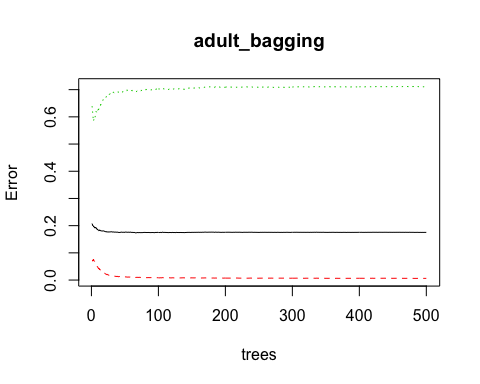
Bootstrap: draw n items from X with replacement

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 6120 1406  
## >50K 40 575

accuracy <- adult\_predict\_confusion$overall[1]  
accuracy

## Accuracy   
## 0.8223805

# importance of predictors  
adult\_bagging$importance

## <=50K >50K MeanDecreaseAccuracy MeanDecreaseGini  
## V1 0.0049306561 0.0144419426 0.0072111415 1085.84109  
## V2 0.0024906046 0.0054135851 0.0031887608 286.75055  
## V3 -0.0001988956 -0.0005500772 -0.0002857537 1626.98627  
## V4 0.0137654219 -0.0097888796 0.0081132433 143.84481  
## V5 0.0203861588 -0.0029534161 0.0147879156 998.69198  
## V6 0.0355076407 -0.0040624276 0.0260138458 73.45320  
## V7 0.0094921128 0.0181764887 0.0115749347 636.33629  
## V8 0.0429096933 -0.0031834822 0.0318485641 1825.19284  
## V9 0.0003204685 0.0001255798 0.0002739902 85.63695  
## V10 0.0030428349 -0.0012649488 0.0020088294 48.08797  
## V11 0.0363483998 0.1129306290 0.0547238671 964.89803  
## V12 0.0053528498 0.0420863700 0.0141673485 303.82477  
## V13 0.0050087254 0.0041056740 0.0047908228 623.62269  
## V14 -0.0002026956 -0.0001912583 -0.0001998350 207.64357

# 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 5694 748  
## >50K 466 1233

accuracy <- adult\_predict\_confusion$overall[1]  
accuracy

## Accuracy   
## 0.8508783

# 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](http://www.inside-r.org/packages/cran/adabag/docs/bagging)
* [bagging {ipred} | inside-R | A Community Site for R](http://www.inside-r.org/packages/cran/ipred/docs/bagging)
* [Bagging / Bootstrap Aggregation with R](http://amunategui.github.io/bagging-in-R/)