Boosting

Nik Bear Brown

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

# Additional packages needed

To run the code you may need additional packages.

* If necessary install the followings packages.

install.packages("adabag");  
install.packages("rpart");  
install.packages("ada");

library(adabag)

## Loading required package: rpart

## Loading required package: mlbench

## Loading required package: caret

## Loading required package: lattice

## Loading required package: ggplot2

library(rpart)  
library(ada)

# 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] 8969 29936 3311 9829 20594 15914

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

## V1 V2 V3 V4 V5 V6  
## 8969 61 Private 215944 9th 5 Divorced  
## 29936 31 Private 83425 Some-college 10 Never-married  
## 3311 36 Private 173804 Some-college 10 Never-married  
## 9829 64 State-gov 216160 Doctorate 16 Married-civ-spouse  
## 20594 35 Self-emp-not-inc 176101 HS-grad 9 Married-civ-spouse  
## 15914 32 Private 48458 HS-grad 9 Never-married  
## V7 V8 V9 V10 V11 V12 V13  
## 8969 Sales Not-in-family White Male 0 0 25  
## 29936 Adm-clerical Unmarried White Female 0 0 40  
## 3311 Sales Not-in-family White Female 0 0 35  
## 9829 Prof-specialty Husband White Male 0 0 50  
## 20594 Farming-fishing Husband White Male 0 0 80  
## 15914 Sales Own-child Black Female 0 1669 45  
## V14 V15  
## 8969 United-States <=50K  
## 29936 United-States <=50K  
## 3311 United-States <=50K  
## 9829 Columbia >50K  
## 20594 United-States >50K  
## 15914 United-States <=50K

head(test)

## V1 V2 V3 V4 V5 V6  
## 1 39 State-gov 77516 Bachelors 13 Never-married  
## 15 40 Private 121772 Assoc-voc 11 Married-civ-spouse  
## 17 25 Self-emp-not-inc 176756 HS-grad 9 Never-married  
## 26 56 Local-gov 216851 Bachelors 13 Married-civ-spouse  
## 28 54 ? 180211 Some-college 10 Married-civ-spouse  
## 29 39 Private 367260 HS-grad 9 Divorced  
## V7 V8 V9 V10 V11 V12 V13  
## 1 Adm-clerical Not-in-family White Male 2174 0 40  
## 15 Craft-repair Husband Asian-Pac-Islander Male 0 0 40  
## 17 Farming-fishing Own-child White Male 0 0 35  
## 26 Tech-support Husband White Male 0 0 40  
## 28 ? Husband Asian-Pac-Islander Male 0 0 60  
## 29 Exec-managerial Not-in-family White Male 0 0 80  
## V14 V15  
## 1 United-States <=50K  
## 15 ? >50K  
## 17 United-States <=50K  
## 26 United-States >50K  
## 28 South >50K  
## 29 United-States <=50K

# Boosting

Boosting involves incrementally building an ensemble by training each new model instance to emphasize the training instances that previous models mis-classified. In these sense it "learns." Unlike Boosting, weights may change at the end of boosting round making certain learners more important than others. In some cases, boosting has been shown to yield better accuracy than Boosting, but it also tends tp propgate bias from the overweighting winning predictor and is more likely to over-fit the training data. By far, the most common implementation of Boosting is Adaboost.

# Boosting in R

# adabag package  
adult\_boosting1 <- boosting(V15~., data=train, mfinal=20,  
 control=rpart.control(maxdepth=5))  
adult\_predict1 <- predict.boosting(adult\_boosting1, newdata=test)  
adult\_predict1$confusion

## Observed Class  
## Predicted Class <=50K >50K  
## <=50K 5780 753  
## >50K 382 1226

accuracy <- 1- adult\_predict1$error  
accuracy

## [1] 0.8605822

# ada package  
adult\_boosting2 <- ada(V15~., data=train,   
 iter=50,nu=1)  
adult\_predict2 <- predict(adult\_boosting2, test)  
adult\_predict\_confusion <- confusionMatrix(adult\_predict2, test$V15)  
adult\_predict\_confusion$table

## Reference  
## Prediction <=50K >50K  
## <=50K 5715 688  
## >50K 447 1291

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

## Accuracy   
## 0.8605822

# Resources

* [An Attempt to Understand Boosting Algorithm(s) via @rbloggers](<http://www.r-bloggers.com/an-attempt-to-understand-boosting-algorithms/>)
* [Boosting](http://www.stat.cmu.edu/~ryantibs/datamining/lectures/25-boost.pdf)
* [boosting {adabag} | inside-R | A Community Site for R](http://www.inside-r.org/packages/cran/adabag/docs/boosting)