

Statistics 452: Statistical Learning and Prediction

Chapter 8, Part 2: Bagging and Random Forests

Brad McNeney

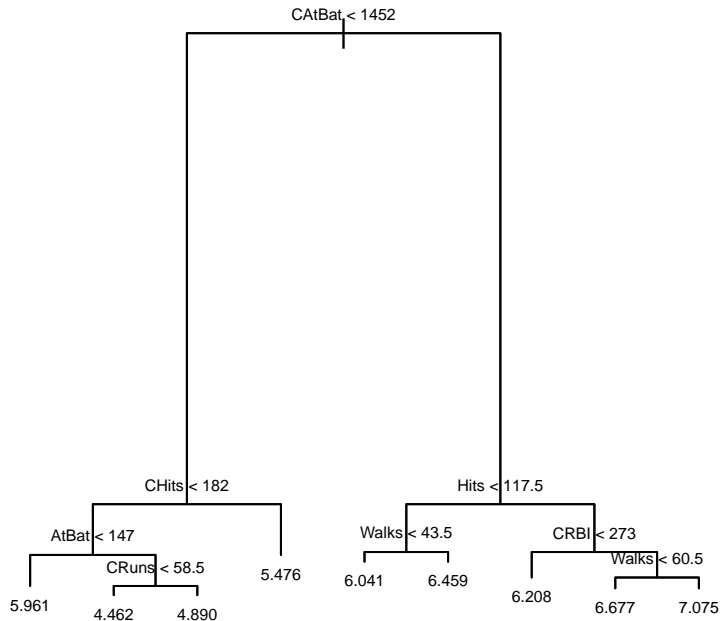
2017-11-06

Example: The Hitters Data Set

- ▶ Example tree grown, but not pruned.

```
library(ISLR)
data(Hitters)
library(dplyr)
Hitters <- mutate(Hitters, lSalary = log(Salary)) %>%
  select(-Salary) %>%
  na.omit()
library(tree)
t1 <- tree(lSalary ~ ., data=Hitters)
```

```
plot(t1)
text(t1,cex=.5)
```



Bootstrapping

- ▶ A bootstrap sample from n observations is obtained by drawing n observations, with replacement, from the sample

```
set.seed(543)
n <- 6
x <- round(rnorm(n),1)
sort(x)
```

```
## [1] -1.0 -0.2  0.2  0.4  0.8  1.4
```

```
sort(sample(x,size=n,replace=TRUE)) # bootstrap sample 1,
```

```
## [1] -0.2  0.4  0.4  0.8  0.8  0.8
```

```
sort(sample(x,size=n,replace=TRUE)) # bootstrap sample 2
```

```
## [1] -1.0 -0.2  0.2  0.2  0.2  0.2
```

```
sort(sample(x,size=n,replace=TRUE)) # bootstrap sample 3
```

```
## [1] -0.2 -0.2 -0.2  0.2  0.4  0.8
```

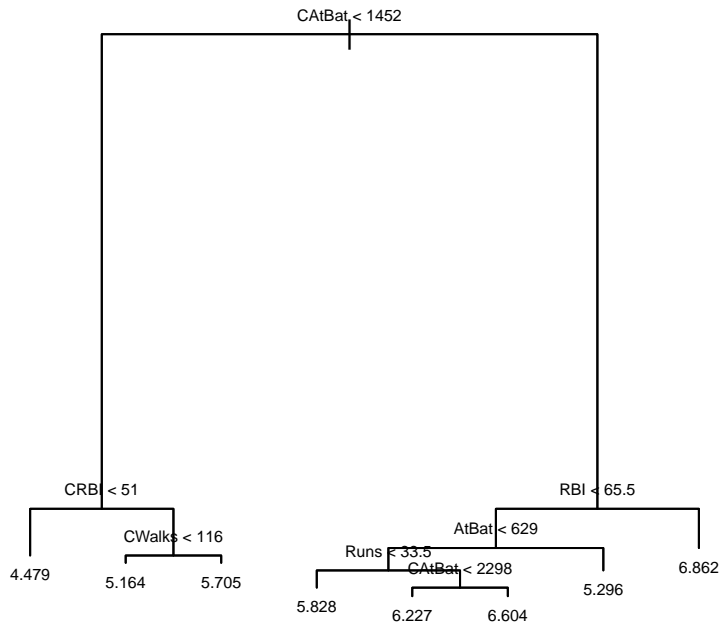
Empirical Distribution

- ▶ The data-generating process in our example is a normal distribution.
- ▶ The empirical distribution of our sample is the distribution under which each sampled value is equally likely.
- ▶ Resampling amounts to sampling from the empirical distribution.
- ▶ If empirical close to data-generating distribution, resampled samples are like new samples from the data-generating distribution.

Example: Tree from a Bootstrap Sample

```
# Use sample_n() from dplyr  
n <- nrow(Hitters)  
bHitters <- sample_n(Hitters,size=n,replace=TRUE)  
t1b <- tree(lSalary ~ .,data=bHitters)
```

```
plot(t1b)
text(t1b,cex=.5)
```



Bootstrap Aggregation (Bagging)

- ▶ For a general statistical learning method that provides an estimate, $\hat{f}(x)$ of the model $f(x)$, we can aggregate over estimates fit to bootstrap samples.
- ▶ If $\hat{f}^b(x)$ is the fitted model from the b th bootstrap sample, the aggregate estimator is

$$\hat{f}_{\text{avg}}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^b(x),$$

where B is the number of bootstrap samples.

- ▶ Bagging has been found to be useful for improving predictions from decision trees.
 - ▶ Grow each tree “deep” and don’t prune.
 - ▶ Average predictions over trees.

Bias-Variance Tradeoff

- ▶ The deeply-grown trees have high variance, but low bias.
- ▶ Averaging many deep trees is an attempt to reduce the variance.
- ▶ However, if trees are highly correlated (e.g., similar first few splits) then averaging does not reduce variance.
 - ▶ Return to this point when we discuss random forests.

Bagging for Classification

- ▶ When the outcome Y is categorical, one approach to the aggregate prediction of category is the “majority vote”:
 - ▶ Record the predicted class from each tree (each bootstrap sample) and take the most common prediction as the aggregate prediction.

Estimating Test Error with Out-of-Bag (OOB) Observations

- ▶ On average about $2/3$ are and $1/3$ of observations are not resampled
 - ▶ See small bootstrap samples of size 6 above.
- ▶ Those not resampled are called out-of-bag (OOB)
- ▶ Regression trees: The OOB prediction for the i th observation is the average of its predicted values from trees grown on bootstrap samples that did not include i .
 - ▶ The OOB MSE is the average squared error between observed values and their OOB predictions.
- ▶ Classification trees: The OOB prediction for the i th observation is the majority vote from trees grown on bootstrap samples that did not include i .
 - ▶ The OOB classification error is the proportion of misclassified OOB predictions.
- ▶ OOB errors are valid because the predictions use only trees not fit to the corresponding observation.

Variable Importance

- ▶ There isn't one tree to interpret now.
- ▶ Need an overall measure of the importance of a variable.
- ▶ For a single tree, a variable's importance can be measured by the amount that the node homogeneity (RSS for regression, Gini for classification) is decreased by splitting on the variable.
 - ▶ Homogeneity of tree including variable minus homogeneity if variable was not available to split on.

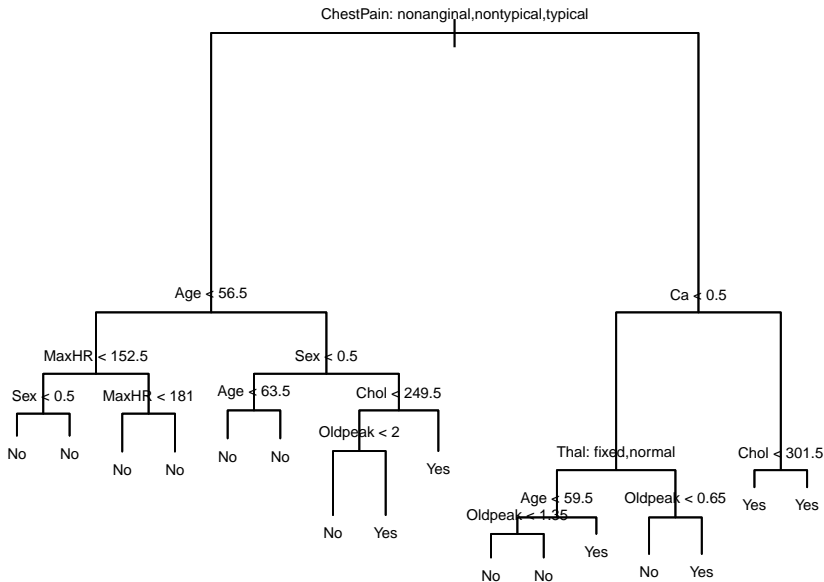
Example: Heart Data

```
uu <- url("http://www-bcf.usc.edu/~gareth/ISL/Heart.csv")
Heart <- read.csv(uu,row.names=1)
Heart <- na.omit(Heart)
dim(Heart) # Train on 2/3, test on 1/3
```

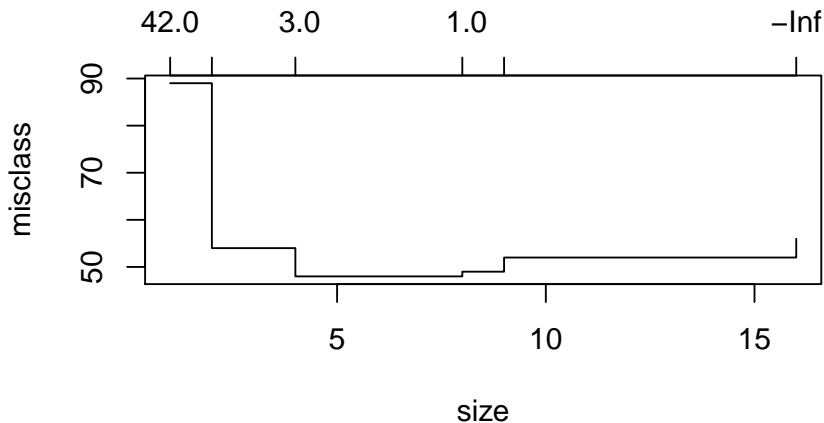
```
## [1] 297 14
```

```
train <- sample(1:nrow(Heart),size=2*nrow(Heart)/3,replace=FALSE)
t1 <- tree(AHD ~ ., data=Heart,subset=train)
```

```
plot(t1)
text(t1,cex=.5,pretty=0)
```



```
cv.t1 <- cv.tree(t1,FUN=prune.misclass) # use class'n err for CV  
plot(cv.t1) # Suggest size 4
```



```
t1.best <- prune.tree(t1,best=4)
summary(t1.best)
```

```
##
## Classification tree:
## snip.tree(tree = t1, nodes = c(7L, 4L, 5L, 6L))
## Variables actually used in tree construction:
## [1] "ChestPain" "Age"      "Ca"
## Number of terminal nodes: 4
## Residual mean deviance: 0.8502 = 164.9 / 194
## Misclassification error rate: 0.2121 = 42 / 198
```

```
hpred <- predict(t1.best,newdata=Heart[-train,],type="class")
table(hpred,Heart[-train,]$AHD)
```

```
##
## hpred No Yes
##    No  45  24
##    Yes  4  26
```

```
28/nrow(Heart[-train,])
```

```
## [1] 0.2828283
```

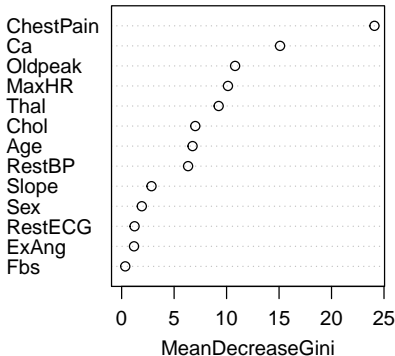
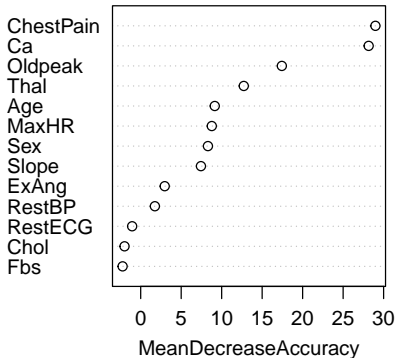

Bagging on Heart Data

```
library(randomForest)
set.seed(1)
bag.heart <- randomForest(AHD ~ ., data=Heart, subset=train,
                          mtry=13, importance=TRUE)
bag.heart
```

```
##
## Call:
## randomForest(formula = AHD ~ ., data = Heart, mtry = 13, importance = TRUE,
##               Type of random forest: classification
##               Number of trees: 500
## No. of variables tried at each split: 13
##
##               OOB estimate of error rate: 20.71%
## Confusion matrix:
##      No Yes class.error
## No  97  14   0.1261261
## Yes 27  60   0.3103448
```

```
varImpPlot(bag.heart) # or importance(bag.heart)
```

bag.heart



► See `help("importance")` for a definition of accuracy.

Classification Error for Bagged Trees

```
bag.hpred <- predict(bag.heart,newdata=Heart[-train,],type="class")  
table(bag.hpred,Heart[-train,]$AHD)
```

```
##  
## bag.hpred No Yes  
##      No  44  13  
##      Yes  5  37
```

```
18/nrow(Heart[-train,])
```

```
## [1] 0.1818182
```

Random Forests

- ▶ Bootstrap datasets may yield correlated trees.
 - ▶ E.G., if we have one or two strong predictors, all trees might share the first few splits.
- ▶ Random forests avoid this by restricting the predictors to a random subset.
 - ▶ New subset at each split.
 - ▶ Subset size? Text: "... typically we choose $m \approx \sqrt{p}$ ".

Random Forests on Heart Data

```
bag.heart <- randomForest(AHD ~ ., data=Heart,subset=train,  
                          mtry=sqrt(13),importance=TRUE)
```

```
bag.heart
```

```
##
```

```
## Call:
```

```
## randomForest(formula = AHD ~ ., data = Heart, mtry = sqrt(13),
```

```
importa
```

```
##           Type of random forest: classification
```

```
##           Number of trees: 500
```

```
## No. of variables tried at each split: 4
```

```
##
```

```
##           OOB estimate of  error rate: 19.19%
```

```
## Confusion matrix:
```

```
##      No Yes class.error
```

```
## No   97  14   0.1261261
```

```
## Yes  24  63   0.2758621
```

```
bag.hpred <- predict(bag.heart,newdata=Heart[-train,],type="class")  
table(bag.hpred,Heart[-train,]$AHD)
```

```
##
```

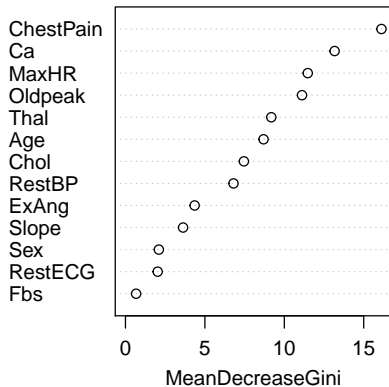
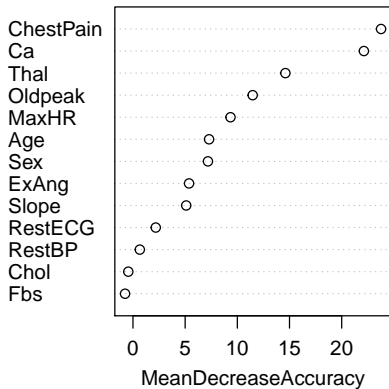
```
## bag.hpred No Yes
```

```
##      No   43  11
```

```
##      Yes   6  39
```

```
varImpPlot(bag.heart) # or importance(bag.heart)
```

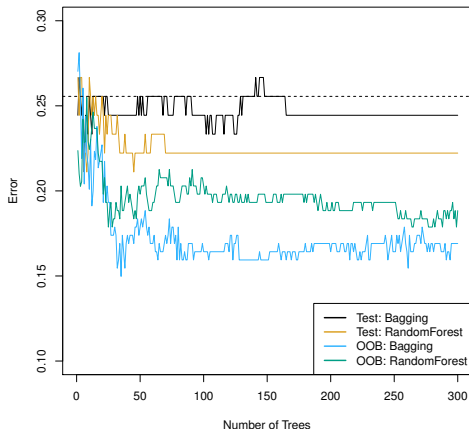
bag.heart



- Chest pain still most important, but not by such a wide margin.

Number of Trees

- ▶ Bagging and random forests are not very sensitive to the number of trees B .
- ▶ Can try different values and make sure B large enough that error has stabilized.



- ▶ Text, Fig. 8.8
Note OOB error, lower than test error, but convenient.