Statistics 452: Statistical Learning and Prediction

Chapter 8, Part 1: Introduction to Tree-Based Methods

Brad McNeney

Decision Trees

- Basic idea: Recursively split the space of predictors into regions.
  - The prediction for a region is a summary of the training responses in that region, such as the mean (regression) or mode (classification).
  - Represent the splits on different predictors as a tree $\Rightarrow$ decision trees.
- Single trees are not typically competitive for prediction accuracy, but we can produce multiple decision trees and use the consensus from these as the prediction.
- Decision trees can be used for regression or classification.

Regression Trees

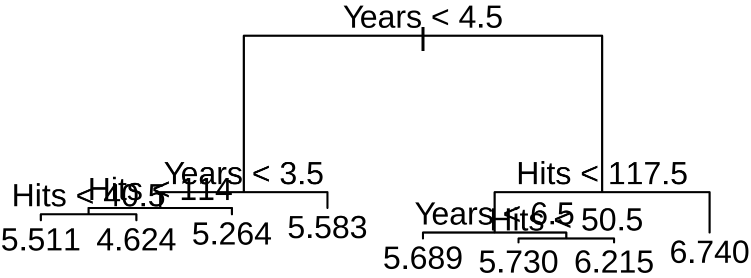
- Example: `Hitters` data

```
library(ISLR)
data(Hitters)
head(Hitters,n=3)
```

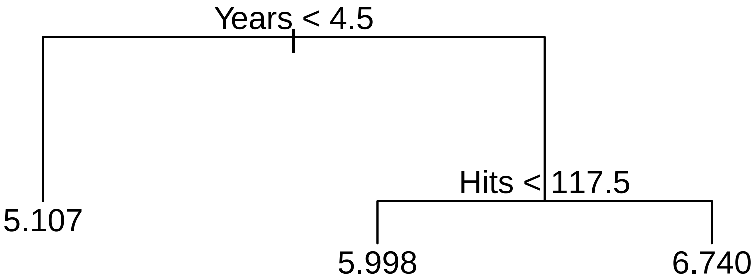
```
##                   AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun
## -Andy Allanson      293   66     1   30  29    14     1    293    66      1
## -Alan Ashby         315   81     7   24  38    39    14   3449   835     69
## -Alvin Davis        479  130    18   66  72    76     3   1624   457     63
##                   CRuns CRBI CWalks League Division PutOuts Assists Errors
## -Andy Allanson       30   29     14      A        E     446      33     20
## -Alan Ashby         321  414    375      N        W     632      43     10
## -Alvin Davis        224  266    263      A        W     880      82     14
##                   Salary NewLeague
## -Andy Allanson        NA         A
## -Alan Ashby          475         N
## -Alvin Davis         480         A
```

```
library(dplyr)
Hitters <- mutate(Hitters,lSalary = log(Salary)) %>% na.omit()
```

```r
library(tree)
t1 <- tree(lSalary ~ Years + Hits,data=Hitters)
plot(t1)
text(t1)
```



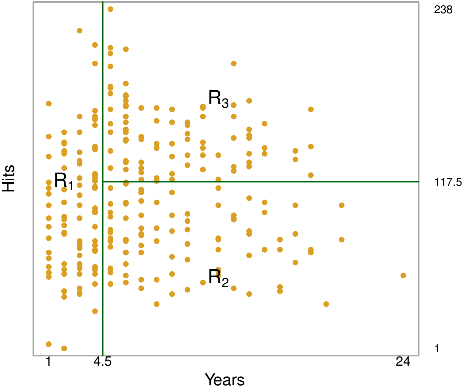
```r
t1.pr <- prune.tree(t1,best=3)
plot(t1.pr)
text(t1.pr)
```



Tree Splits

- Tree represents a series of splits
  - First split on `Years`: Players with < 4.5 go in left child branch, those with ≥ 4.5 go in right child branch.
  - Second split is on `Hits` for the branch with `Years` ≥ 4.5: Players with < 117.5 go in left child branch, those with ≥ 117.5 go in right child branch.
- The resulting regions are called *terminal nodes* or *leaves* of the tree.

Illustration of Leaves for Hitters



- Text, Fig. 8.2: Three regions from partitioning on (i) `Years` and then (ii) `Hits` within `Years` ≥ 4.5.

Predictions for each Region

- The predictions are the mean log-Salary for players within each region; e.g.,

```
with(Hitters,mean(lSalary[Years<4.5])) # R1
```

```
## [1] 5.10679
```

```
with(Hitters,mean(lSalary[Years>=4.5&Hits<117.5])) #R2
```

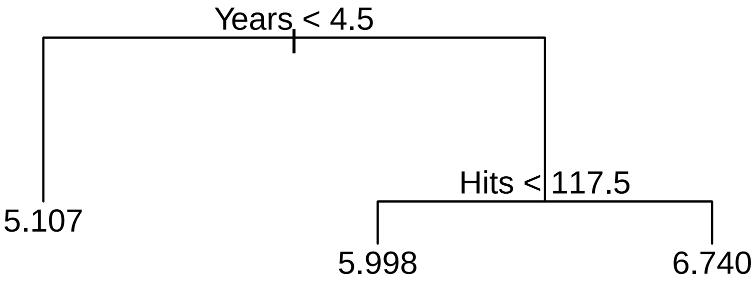
```
## [1] 5.99838
```

```
with(Hitters,mean(lSalary[Years>=4.5&Hits>=117.5])) #R3
```

```
## [1] 6.739687
```

Interpretation

```r
plot(t1.pr); text(t1.pr)
```



- Years is most important, with newer players earning the lowest salaries.
- Within more experienced players, those who get more hits get more $.

Stratification of the Feature Space

1. Divide the predictor space, or possible values of $X = (X_1, \ldots, X_p)$ into J distinct non-overlapping regions, $R_1, \ldots, R_J$.
2. For every observation in R_j, the prediction is the mean response of observations in R_j.

Stratification Into "Boxes"

- Restrict the regions to be "boxes" (high-dimensional rectangles).
- Goal: Find the boxes $R_1, \ldots, R_J$ that are most homogeneous with respect to the outcome; that is, that minimize the RSS

$$\sum_{j=1}^{J} \sum_{i: x_i \in R_j} (y_i - \hat{y}_{R_j})^2,$$

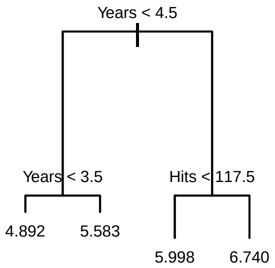
  where $\hat{y}_{R_j}$ is the mean outcome in box R_j.
- Rather than search all possible boxes, create them by recursive splitting.
  - Start with the entire feature space.
  - Consider splitting on each variable and at each data observed value.
  - Find the split that creates the two most homogeneous regions.
  - Repeat.
  - Stop when, say, no leaf has more than a pre-specified number of observations.

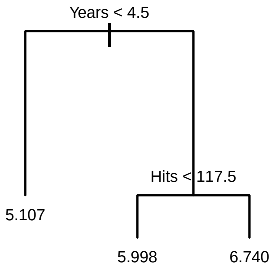
Tree Pruning

- Such a partition may predict the training observations well, but may overfit and yield poor predictions on test data.
- We could stop the splitting when the decrease in the RSS becomes "small".
- But stopping may miss good splits lower in the tree that reduce RSS.
- Strategy is to grow a big tree, T_0, and then "prune" it; i.e., successively remove branches.

Illustration of Pruning

```
t1.pr2 <- prune.tree(t1,best=4)
par(mfrow=c(1,2))
plot(t1.pr2); text(t1.pr2,cex=.5)
plot(t1.pr); text(t1.pr,cex=.5)
```





- Starting with the left-hand tree, the split at 3.5 on `Year` is removed, or pruned off.
- The remaining tree, on the right, is called a subtree.

Criterion for Finding the Best Subtree

- Could consider removing all possible branches, and evaluating the resulting subtree by CV-estimated test set error.
  - Too computationally expensive.
  - An alternative is *cost complexity pruning*
- For a given value of a tuning parameter α, we seek the subtree T that minimizes

$$\sum_{j=1}^{|T|} \sum_{i: x_i \in R_j} (y_i - \hat{y}_{R_j})^2 + \alpha |T|,$$

where $|T|$ is the number of leaves in tree T.

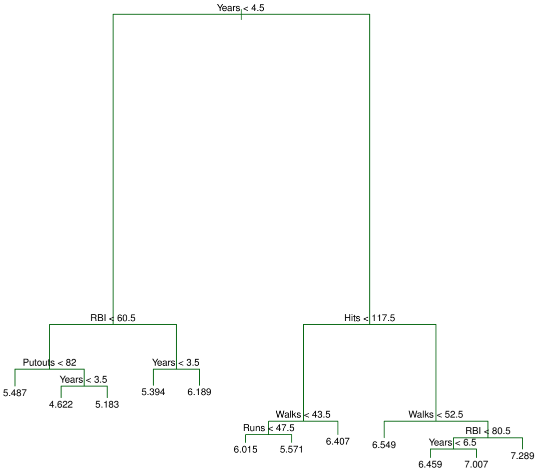
Sequence of α Values

- When $\alpha = 0$ the best subtree is T_0 itself.
- As α increases, we start to pay for extra nodes, and at some α_1, the best subtree T_1 will be a strict subtree with one branch removed.
- Weakest link cutting is an algorithm for finding the α_1 that will yield a new optimal tree and finding this new optimal tree.
  - New tree: Collapse the internal node that produces the smallest increase in RSS.
- We end up with a sequence of α-values $\alpha_0 = 0, \alpha_1, \ldots, \alpha_k$ for some k, and corresponding best trees $T_0 \supset T_1 \supset \ldots \supset T_k$.

Choosing α and its Best Tree

- Now use CV to select the best value of α.

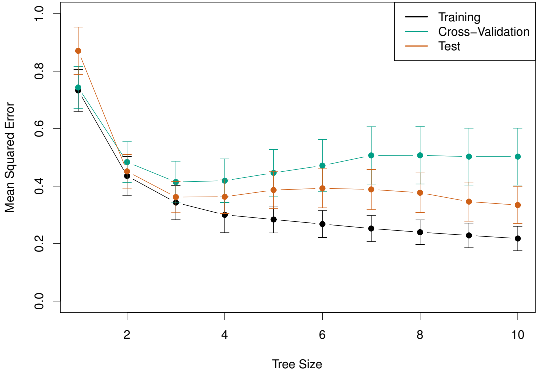
Illustration with Hitters Data



- Text, Fig. 8.4: Unpruned regression tree for Hitters data

Estimated MSE

- Split Hitters into 132 training and 131 test obs'ns.
- Apply CV to the training half to select α, or equivalently $|T|$.



- Text, Fig. 8.5: MSE based on training (black), test set (orange) or CV (green). (No details on SEs.) Based on CV take $|T| = 3$.

Classification Trees

- Same idea as regression trees, but for predicting a qualitatitve response.
- Prediction for a region is the most common class in the region.
- Regions are chosen to minimize a measure of heterogeneity within the region.
  - Instead of RSS, use classification error rate as measure of heterogeneity?

Leaf Heterogeneity Measures

- Classification error is not sensitive enough for tree growing.
- Instead prefer the Gini index (sum of variances; small when all $\hat{p}$'s near 0 or 1)

$$G_m = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

or the cross-entropy (also small when all $\hat{p}$'s near 0 or 1)

$$D_m = \sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk},$$

where $\hat{p}_{mk}$ is the proportion of observations in region m that are from class k and K is the number of classes of the response.

  - G_m and D_m are near zero when all but one $\hat{p}_{mk}$ near zero.

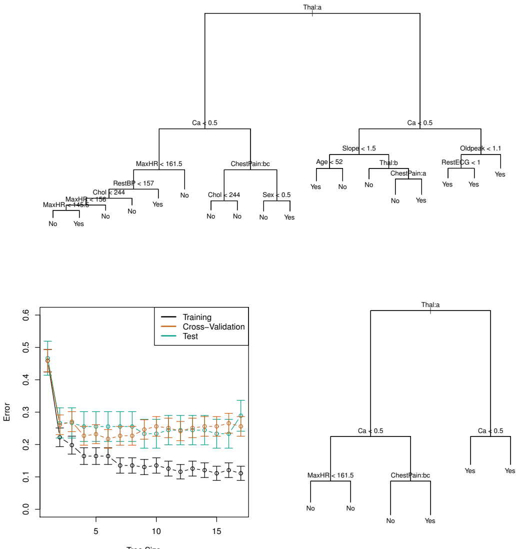
Example: Heart Data

```r
uu <- url("http://faculty.marshall.usc.edu/gareth-james/ISL/Heart.csv")
Heart <- read.csv(uu,row.names=1)
head(Heart,n=3)
```

```
##   Age Sex    ChestPain RestBP Chol Fbs RestECG MaxHR ExAng Oldpeak Slope
## 1  63   1      typical    145  233   1       2   150     0     2.3     3
## 2  67   1 asymptomatic    160  286   0       2   108     1     1.5     2
## 3  67   1 asymptomatic    120  229   0       2   129     1     2.6     2
##   Ca       Thal AHD
## 1  0      fixed  No
## 2  3     normal Yes
## 3  2 reversable Yes
```

- Binary outcome HD (heart disease Yes or No)

Best Treee for Heart Data



- Text, Fig. 8.6: Unpruned tree, estimated MSE, best tree

```
# Value b of Thal is normal when I read it in
```

Trees Versus Linear Models

- Which is better?
- Depends on the data-generating model.
  - If approximately linear, or approximately constant over regions.
- Interpretability: Statisticians think of linear models as interpretable, but for many decision trees are more natural.
- Trees are generally not as accurate at making predictions, though.
- Can improve predictive ability by aggregating many decision trees.