Chapter 6: Decision Tree

Qing (Wendy) Wang

Spring, 2016

Outline:

- Regression Tree
- Classification Tree
- Advantages and Disadvantages
- Advanced Tree-based Methods

Introduction

- Tree-based methods can be used for both regression (quantitative response) and classification (categorical response).
- The term CART refers to *Classification And Regression Tree* that was first introduced by Breiman et al. in 1984.[1]
- Each tree is build upside-down based on a greedy approach known as *recursive binary splitting*. The most important predictor is first split into two decision groups, followed by other predictors in subsequent splits.

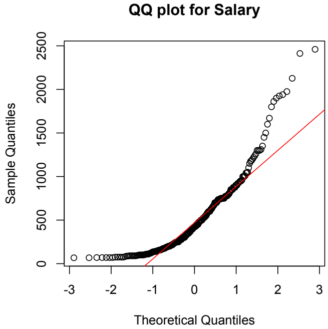
[1] Breiman, Friedman, Olshen, and Stone (1984). *Classification and regression trees*. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software.

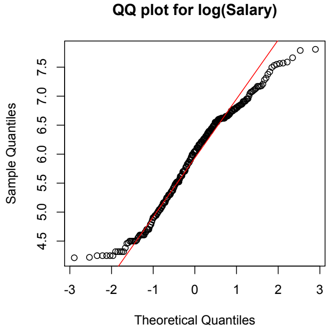
Regression Tree

- Let's first consider regression trees where the response of interest is quantitative.
- To motivate regression trees, we begin with a simple example based on *Hitters* data set. The goal is to predict a baseball player's salary based on Years (the number of years that he has played in the major leagues) and Hits (the number of hits that he made in the previous year).
- The data set, named *Hitters*, is available in "ISLR" package in **R**.

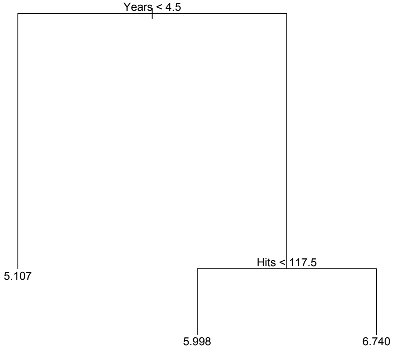
```
> library("ISLR")
> head(Hitters)
                  AtBat Hits HmRun Runs RBI Walks Years CAtBat C
-Andy Allanson      293   66     1   30  29    14     1    293
-Alan Ashby         315   81     7   24  38    39    14   3449
-Alvin Davis        479  130    18   66  72    76     3   1624
-Andre Dawson       496  141    20   65  78    37    11   5628
-Andres Galarraga   321   87    10   39  42    30     2    396
-Alfredo Griffin    594  169     4   74  51    35    11   4408
                  Division PutOuts Assists Errors Salary NewLeag
-Andy Allanson           E     446      33     20     NA
-Alan Ashby              W     632      43     10  475.0
-Alvin Davis             W     880      82     14  480.0
-Andre Dawson            E     200      11      3  500.0
-Andres Galarraga        E     805      40      4   91.5
-Alfredo Griffin         W     282     421     25  750.0
> Hitters.new = na.omit(Hitters) #remove observations with NA's
```

We will also take a natural logarithm of the response to have a more bell-shaped distribution.





The regression tree for predicting the log of Salary:



- Years is a more significant predictor than Hits for predicting baseball players' salary.
- Among players with less experience ($\leq$4 years), number of Hits does not play a major role.
- Among players with more experience ($\geq$5 years), having at least 118 Hits the previous year leads to higher salary.

Concepts:

- Decision trees are often drawn upside down, in the sense that the leaves are at the bottom of the tree.
- The final branches of the tree are called *terminal nodes* or *leaves*.
- The points along the tree where the predictor space is split are referred to as *internal nodes*.

Remarks:

- Overall, the regression tree stratifies the players into three regions of predictor space:
  1. players who have played for four or fewer years;
  2. players who have played for five or more years and who made fewer than 118 hits the previous year;
  3. players who have played for five or more years and who made at least 118 hits the previous year.
- We can denote these three regions as

$$\begin{aligned} R_1 &= \{(X_1, X_2) \mid \text{Years} < 4.5\} \\ R_2 &= \{(X_1, X_2) \mid \text{Years} \geq 4.5, \ \text{Hits} < 117.5\} \\ R_3 &= \{(X_1, X_2) \mid \text{Years} \geq 4.5, \ \text{Hits} \geq 117.5\} \end{aligned}$$

  where $X_1 =$Years and $X_2 =$Hits.
- Future predictions: given observed (x_1, x_2) if $(x_1, x_2) \in R_j$, then the predicted log-salary for that player is the mean log-salary in R_j $(1 \leq j \leq 3)$.

Remarks:

- From the **R** output tree, for any observations that satisfy feature space R_1 the predicted salary is $e^{5.107} = 165.17$ thousands; for R_2 the predicted salary is $e^{5.998} = 402.63$ thousands; for R_3 the predicted salary is $e^{6.740} = 845.56$ thousands.
- Note that the data were collected in 1986. Baseball players in major leagues make a lot more than these figures nowadays.

More Details:

- In general, suppose there are J terminal leaves of a regression tree. We can divide the predictor space into J *distinct and non-overlapping* regions based on binary feature splitting. We denote the J regions as $R_1, \ldots, R_J$.
- For every observation i falling in region R_j, the predicted response $\hat{y}_i = \hat{y}_{R_j} =$ mean responses of all observations in region R_j.
- **Splitting Criterion**: The way to select which predictor to split first and how to choose the cut-off value to split a predictor is by minimizing the residual sum of squares

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

How can we fit a regression tree in R?

R Help: To fit a regression tree in **R**, one can use the *tree()* function in "tree" package. The syntax of *tree()* function is quite similar to that of *lm()* function.

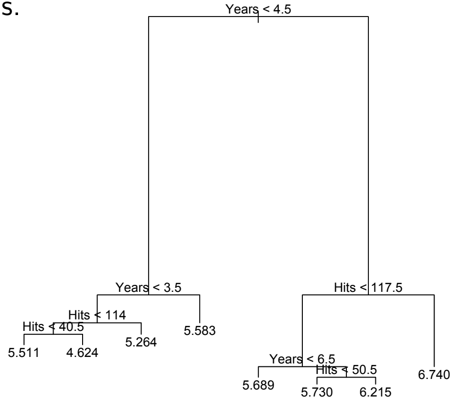
```
 tree(response~x1+x2+...,data)
```

In the Hitters example, we have

```
install.packages("tree")
library("tree")
Hitters.tree = tree(log(Salary)~Years+ Hits,data=Hitters.new)
plot(Hitters.tree)
text(Hitters.tree,pretty=0)
```

Tree Pruning

- The tree below (obtained from the R code) has more terminal nodes than shown in the earlier slide.
- This tree may give good predictions for the given data set but may result in poor predictions on independent (future) observations.



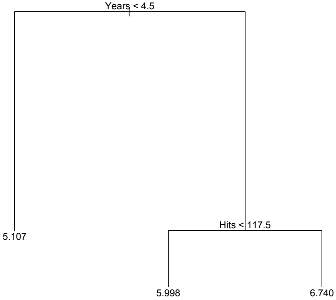
Tree Pruning

- In general, a smaller tree with fewer split might lead to less variance and better interpretation at the cost of a little bias.
- A better strategy is to grow a very large tree (as shown in the previous slide), and then *prune* it back to obtain a subtree.
- Intuitively, the goal is to find a subtree that leads to the smallest test error rate. The test error rate can be estimated using cross-validation.

R Help: *prune.tree()* function.

```
Hitters.prune = prune.tree(Hitters.tree,best=3)
plot(Hitters.prune)
text(Hitters.prune,pretty=0)
```

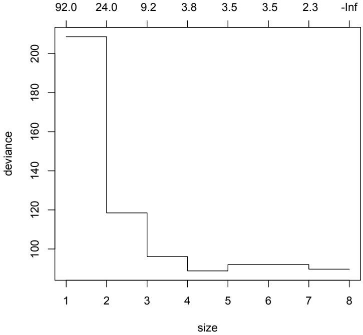
The option "best=3" tells **R** to prune the tree and return the *best* subtree with three terminal leaves.



R Help: One can also use cross-validation to prune the tree and identify what size (# terminal nodes) is optimal for the given data set.

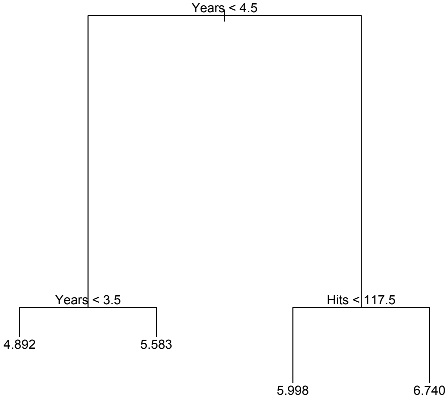
```
>  Hitters.tree = tree(log(Salary)~Years+ Hits,data=Hitters.new)
> result = cv.tree(Hitters.tree,K=10,FUN=prune.tree)
> plot(result)
```

where “K=10” means to use 10-fold cross-validation (default option), and “FUN=prune.tree” means to use prune.tree function to trim the original tree.



Using ten-fold cross-validation, a regression tree of four terminal nodes seems to yield the best performance.

```
> tree.new = prune.tree(Hitters.tree,best=4)
> plot(tree.new)
> text(tree.new,pretty=0)
```



Making predictions in R: *predict()* function

Suppose we want to predict the salaries of two baseball players: one with 4 years of experience and had 80 hits the previous year; the other with 8 years of experience and had 110 hits the previous year.

```
> predict(tree.new,newdata=data.frame(Years=c(4,10),
+ Hits=c(100,150)))
[1] 5.582812 6.739687 #these are on log-scale
> exp(5.5828) #transform back to the original scale
[1] 265.8148
> exp(6.3797)
[1] 589.7508
```

The fitted regression tree would predict the first player's salary to be 265.8148 thousands and predict the second player's salary to be 589.7508 thousands.

Classification Trees

- A classification tree is *very similar* to a regression tree except that it is used to predict a categorical response rather than a quantitative one.
- Difference 1:
  - Regression trees: the *predicted response* for an observation is given by the mean response of the training observations that belong to the same terminal leave.
  - Classification trees: the *predicted class/categorical* is the most commonly occurring class of the training observations of the same terminal leave.

Classification Trees

- Difference 2:
  - Regression trees: the shape and structure of a regression tree is determined by minimizing the residual sum of squares.
  - Classification trees: the shape and structure of a regression tree is determined by minimizing the classification error rate.
  - **Def**: Since we assign an observation in a given region to the most commonly occurring class of the training observations in that region, the classification error rate is simply the fraction of the training observations in that region that do not belong to the most common class.

Remarks:

- For classification trees, we are often interested not only in the class prediction, but also in the *class proportions* among the training observations that fall into the same terminal leave.

Example: Heart disease

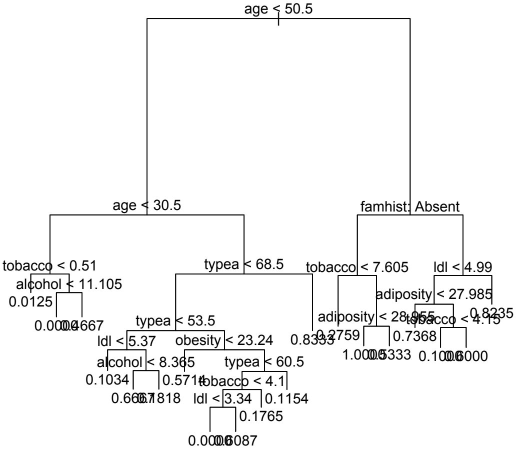
Consider a sample of males in a heart-disease high-risk region of the Western Cape, South Africa. The response is binary outcome for coronary heart disease (1=presence,0=absence). The list of predictor variables in the data are shown below.

Our goal is to build a classification tree to identify important characteristics that lead to heart disease for these people.

sbp	systolic blood pressure
tobacco	cumulative tobacco (kg)
ldl	low densiity lipoprotein cholesterol
adiposity famhist	family history of heart disease (Present, Absent)
typea	type-A behavior
obesity	a measure of obesity
alcohol	current alcohol consumption
age	age at onset

We start with building up the full tree without using cross-validation.

```
> mydata = read.table(file.choose(),sep=",",header=TRUE)
> #notice the use of sep=","
> heart = mydata[,-1] #remove first column of ID
> attach(heart)
> head(heart)
  sbp tobacco  ldl adiposity famhist typea obesity alcohol age c
1 160   12.00 5.73     23.11 Present    49   25.30   97.20  52
2 144    0.01 4.41     28.61  Absent    55   28.87    2.06  63
3 118    0.08 3.48     32.28 Present    52   29.14    3.81  46
4 170    7.50 6.41     38.03 Present    51   31.99   24.26  58
5 134   13.60 3.50     27.78 Present    60   25.99   57.34  49
6 132    6.20 6.47     36.21 Present    62   30.77   14.14  45
> is.factor(chd)
[1] FALSE
> chd=factor(chd)
> heart.tree=tree(chd~.,data=heart)
> plot(heart.tree)
> text(heart.tree,pretty=0)
```

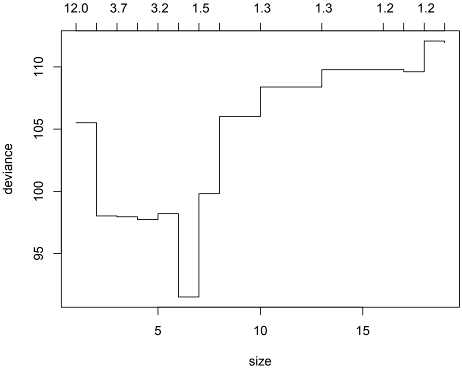


Remarks:

- From the classification tree it seems that Age is the most significant predictor for having heart disease, and family history of heart disease is also a very important predictor.
- Clearly the full classification tree is quite messy. It is highly likely that this regression tree is overfitting the given data set, therefore won't yield satisfactory prediction performance.

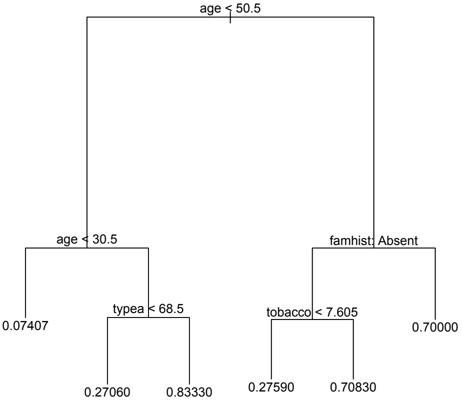
Next, we use *cv.tree()* function to identify the best size of the classification tree based on ten-fold cross-validation.

```
> result = cv.tree(heart.tree,FUN=prune.tree,K=10)
> plot(result)
```



Note: Size 6 is suggested as the optimal choice.

```
> new.tree = prune.tree(heart.tree,best=6)
> plot(new.tree)
> text(new.tree,pretty=0)
```



Note: The number at each terminal node is the probability that observations belonging to that leave have heart disease.

According to the best classification tree of size six, the predictor space is partitioned into six disjoint regions.

$$\begin{aligned}
R_1 &= \{\text{age} < 30.5\} \\
R_2 &= \{30.5 \leq \text{age} < 50.5, \text{type-A behavior} < 68.5\} \\
R_3 &= \{30.5 \leq \text{age} < 50.5, \text{type-A behavior} \geq 68.5\} \\
R_4 &= \{\text{age} \geq 50.5, \text{no family history}, \text{tobacco} < 7.605\} \\
R_5 &= \{\text{age} \geq 50.5, \text{no family history}, \text{tobacco} \geq 7.605\} \\
R_6 &= \{\text{age} \geq 50.5, \text{has family history of heart disease}\}
\end{aligned}$$

Based on the proportions of people with heart disease in each region, we would predict any individual falling into $R_1, R_2,$ and R_4 free of heart disease, while anyone falling into $R_3, R_5,$ and R_6 has heart disease

Advantages of CART

- Trees are very easy to explain to people. In fact, they are even easier to explain than linear regression!
- Some people believe that decision trees more closely mirror human decision-making than do the regression and classification approaches seen in previous chapters.
- Trees can be displayed graphically, and are easily interpreted even by a non-statistician.
- Trees can easily handle categorical predictors without the need to create dummy indicator variables.

Disadvantages of CART

- Unfortunately, trees generally *do not* have the same level of predictive accuracy as some of the other regression and classification approaches.

There are some techniques in modern statistics (data mining, machine learning) that use trees as building blocks to construct *more powerful* prediction models.

Advanced Tree-based Methods

- Below is a list of some very powerful data mining techniques that are built upon simple decision trees.[2]
- List of tree-based methods
  - Bagging: also called bootstrap aggregating, is a general-purpose procedure for reducing the variance of a statistical learning method.

    Idea: Make copies of a large number of training data sets based on bootstrap method, and then build a separate tree using each bootstrapped training sample. Aggregate the results from all built trees.
  - Random Forests: provide an improvement over bagged trees by way of a small tweak that *decorrelates* the trees.
  - Boosting: grow trees in a sequential way, i.e. each tree is grown using information from previously grown trees.

[2] These topics are available for your final projects. Some reference papers of these techniques will be posted on Blackboard later.