Decision Trees

(ISLR 8.1)

Yingbo Li

Southern Methodist University

STAT 4399

Outline

Tree-based Methods

- We will introduce tree-based methods for regression and classification.
- These involve *stratifying or segmenting* the predictor space into a number of simple regions.
- The set of splitting rules used to segment the predictor space can be summarized in a tree.
- These approaches are also called decision-trees.

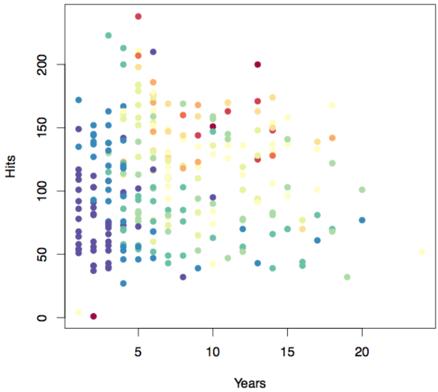
Partitioning up the Predictor Space

- One way to make predictions in a regression problem is to divide the predictor space (i.e. all the possible values for $X_1, X_2, \ldots, X_p$) into distinct regions, say $R_1, R_2, \ldots, R_k$.

- Then for every X that falls in a particular region (say R_j) we make the same prediction,.

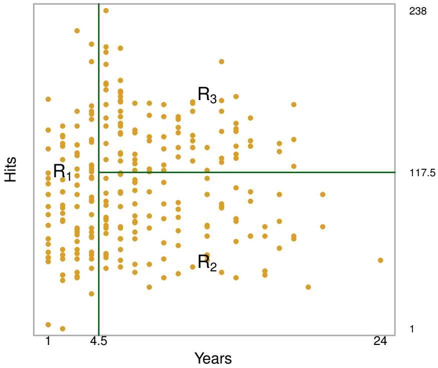
The Baseball Salary Data `Hitters`

- Response: log(`Salary`), color-coded from low (purple) to high (red)
- Predictors:
  1. number of `Years` that a player has played in the major leagues,
  2. number of `Hits` that he made in the previous year.

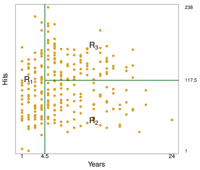


How to stratify the predictor space?

- At a given *internal node*, the label $X_j < t_k$ indicates the left-hand branch, and the right-hand branch corresponds to $X_j \geq t_k$.
- The number in each *terminal node* is the mean of the response for the observations in that region.
  - For example, the predicted salary for a player with who played in the league for 4 years or less is $\$1000 \times e^{5.11} = \$165,670$.



Terminology for Trees



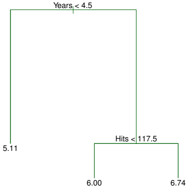
- Decision trees are typically drawn upside down, in the sense that the leaves are at the bottom of the tree.
- The points along the tree where the predictor space is split are called *internal nodes*.
  How many internal nodes are there?
- The regions

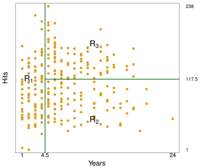
$$R_1 = \{X \mid \texttt{Years} < 4.5\}$$
$$R_2 = \{X \mid \texttt{Years} \geq 4.5,\ \texttt{Hits} < 117.5\}$$
$$R_3 = \{X \mid \texttt{Years} \geq 4.5,\ \texttt{Hits} \geq 117.5\}$$

are known as *terminal nodes*, or *leaves*.

- We refer to the segments of the trees that connect the nodes as *branches*.

Interpretation of Results





- `Years` is the most important factor: players with less experience earn lower salaries.
- Given that a player is less experienced, the number of `Hits` that he made plays little role in his Salary.
- But among players who have been in the major leagues for five or more years, the number of `Hits` made in the previous year does affect salary: players who made more `Hits` last year tend to have higher salaries.
- Compared to a regression model, the regression tree is *easy to display, interpret and explain*.

How to Grow a Tree?

- We choose to divide the predictor space into high-dimensional rectangles, or *boxes*, for simplicity and for ease of interpretation of the resulting predictive model.

- The goal is to find boxes $R_1, \ldots, R_M$ that minimize the RSS, given by

$$\sum_{m=1}^{M} \sum_{i \in R_m} (y_i - \hat{y}_{R_m})^2$$

  where $\hat{y}_{R_m}$ is the mean response for the training observations within the mth box.

Tree-building Details

It is computationally infeasible to consider every possible partition of the feature space into J boxes. So we take a *top-down*, *greedy* approach that is known as *recursive binary splitting*.

- Top-down
  - It begins at the top of the tree and then successively splits the predictor space;
  - each split is indicated via two new branches further down on the tree.
- Greedy
  - At each step of the tree-building process, the best split is made at that particular step,
  - rather than looking ahead and picking a split that will lead to a better tree in some future step.

- We first select the predictor X_j and the cutpoint s such that splitting the predictor space into the regions $\{X \mid X_j < s\}$ and $\{X \mid X_j \geq s\}$ leads to the greatest possible reduction in RSS.

- Next, we repeat the process, looking for the best predictor and best cutpoint in order to split the data further so as to minimize the RSS within each of the resulting regions.

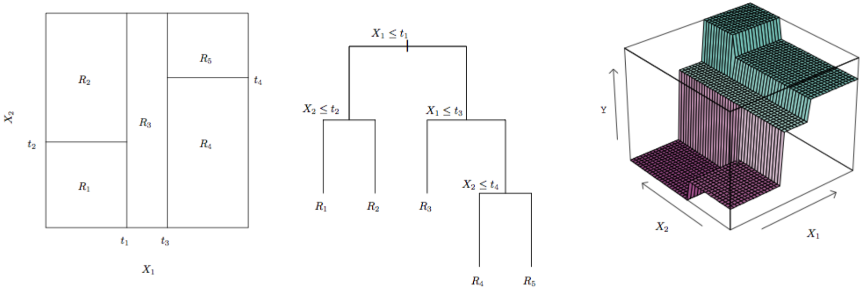
  This time, instead of splitting the entire predictor space, we split one of the two previously identified regions. We now have three regions.

- Again, we look to split one of these three regions further, so as to minimize the RSS.

- The process continues until a stopping criterion is reached; e.g., we may continue until no region contains more than five observations.

A Five-Region Example

- We predict the response for a given test observation using the mean of the training observations in the region to which that test observation belongs.
- Right: the prediction surface.



Pruning a Tree

- A large tree may overfit the data.
- A smaller tree with fewer splits might lead to lower variance and better interpretation at the cost of a little bias.
- One possible alternative is to grow the tree only so long as the decrease in the RSS due to each split exceeds some (high) threshold.
- This strategy results in smaller trees, but is too short-sighted: a seemingly worthless split early on might be followed by a very good split — a split that leads to a large reduction in RSS later on.

Cost Complexity Pruning

- A better strategy is to grow a very large tree T_0, and then *prune* it back to obtain a subtree.

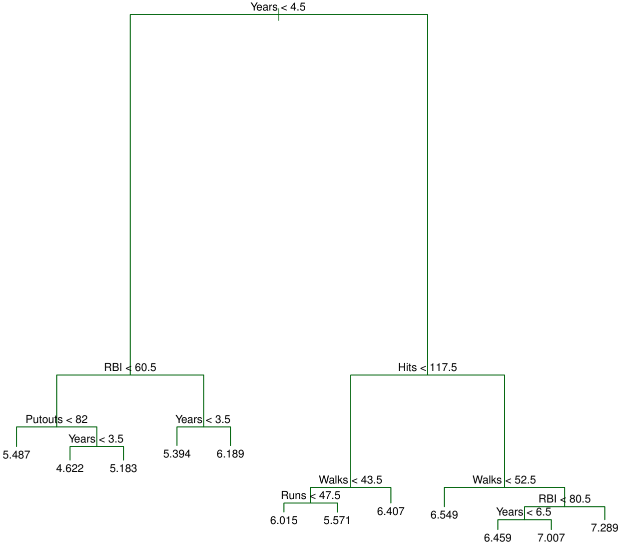
- *Cost complexity pruning*: a penalization method.
  - We consider a sequence of trees indexed by a tuning parameter $\alpha \geq 0$.
  - For each value of α there corresponds a subtree $T \in T_0$ such that

$$\sum_{m=1}^{|T|} \sum_{i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

    is as small as possible. Here $|T|$ is the number of terminal nodes of T.

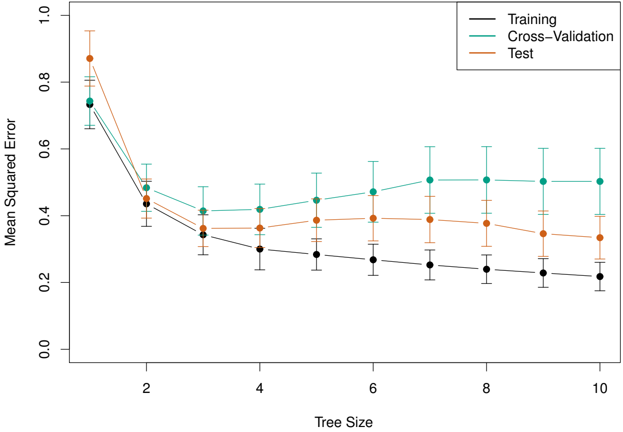
- The turning parameter α
  - It controls a trade-off between the subtree's complexity and its fit to the training data.
  - We select an optimal value $\hat{\alpha}$ using cross-validation, then return to the full data set and obtain the subtree corresponding to $\hat{\alpha}$.

Baseball Example: the Unpruned Tree T_0



Baseball Example: Cross Validation

Cross validation indicated that the minimum test MSE is when the tree size is 3 (i.e. the number of leaf nodes is 3)



Classification Trees

- Very similar to a regression tree, except that it is used to predict a categorical response rather than a continuous one.

- For a classification tree, we predict that each observation belongs to *the most commonly occurring class* of training observations in the region to which it belongs.

- RSS cannot be used as a criterion for making the binary splits. A natural alternative is the classification error rate:

$$1 - \max_k(\hat{p}_{mk})$$

  where $\hat{p}_{mk}$ is the proportion of training points in the mth region that are from the kth class.

- However classification error is not sufficiently sensitive for tree-growing

Gini Index and Cross-Entropy

- The *Gini index* is defined by

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

a measure of total variance across the K classes.
  - G takes on a small value if all of the $\hat{p}_{mk}$ are close to zero or one.
  - The Gini index is referred to as a measure of *node purity* — a small G indicates that a node contains predominantly points from a single class.

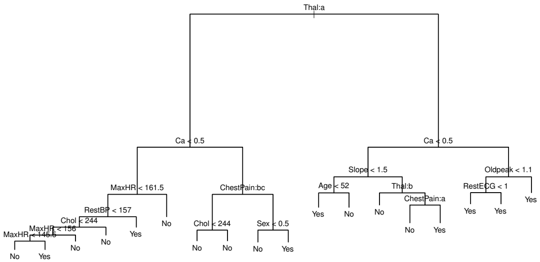
- An alternative is *cross-entropy*:

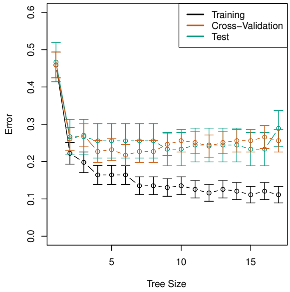
$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk})$$

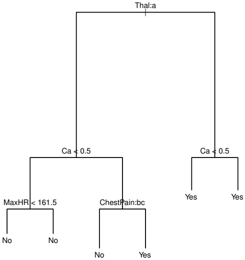
  - This is an equivalent measurement to the deviance.
  - Also indicates node purity.

Heart Data Example

- These data contain 303 patients who presented with chest pain.
- A binary outcome `HD`, indicates whether the patient has a heart disease based on an angiographic test.
- There are 13 predictors including `Age`, `Sex`, `Chol` (a cholesterol measurement), and other heart and lung function measurements.
- Cross-validation yields a tree with six terminal nodes.


Thal:a
Ca < 0.5
Ca < 0.5
MaxHR < 161.5
ChestPain:bc
Slope < 1.5
Oldpeak < 1.1
RestBP < 157
Chol < 244
MaxHR < 156
MaxHR < 145.5
Chol < 244
Sex < 0.5
Age < 52
Thal:b
ChestPain:a
RestECG < 1
No
Yes
No
No
Yes
No
No
No
No
Yes
Yes
No
No
No
Yes
Yes
Yes
Yes


Training
Cross−Validation
Test
Error
Tree Size
0.0
0.1
0.2
0.3
0.4
0.5
0.6
5
10
15


Thal:a
Ca < 0.5
Ca < 0.5
MaxHR < 161.5
ChestPain:bc
No
No
No
Yes
Yes
Yes

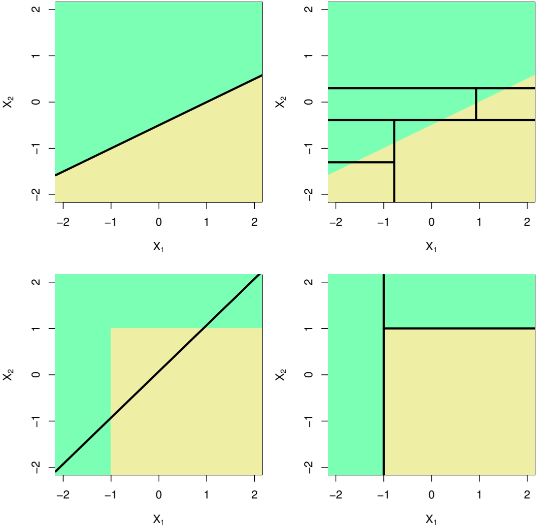
Tree-based Models vs. Linear Models

Linear models

$$f(X) = \beta_0 + \sum_{j=1}^{p} \beta_j X_j$$

Tree-based models

$$f(X) = \sum_{m=1}^{M} c_m \mathbf{1}(X \in R_m)$$



Tree-based Methods: Pros and Cons

Advantages

- Simple and useful for interpretation.
- Trees can be displayed graphically.
- More closely mirror human decision-making.
- Easily handle qualitative predictors without creating dummy variables.

Disdvantages

- Not competitive with the best supervised learning approaches in terms of prediction accuracy.

Combining a large number of trees can often result in dramatic improvements in prediction accuracy, at the expense of some loss interpretation.

- Bagging, random forests, and boosting.
- The latter two methods are among the state-of-the-art methods for supervised learning.