Statistical Learning

Course 4 - Decision Trees

Ecole Centrale de Lyon - Bachelor 2nd Year
2024-2025

- This course is inspired form the chapter 8 of An introduction to statistical Learning book
- Here we describe tree-based methods for regression and classification.
- These involve stratifying or segmenting the predictor space into a number of simple regions.
- Since the set of splitting rules used to segment the predictor space can be summarized in a tree, these types of approaches are known as decision-tree methods.

The Basics of Decision Trees

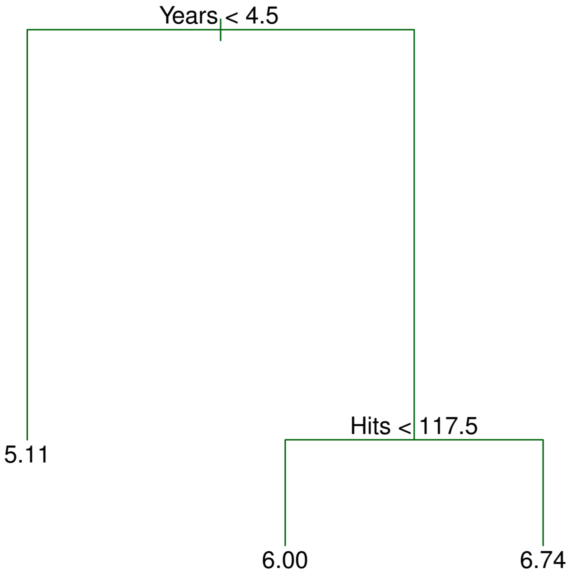
- Decision trees can be applied to both regression and classification problems.
- We first consider regression problems, and then move on to classification.

Hitters data: how would you stratify it?

Aim : Predict a basebal player's Salary based on Years(number of years that he has plyaed in the mamajor leagues) and Hits (number of hits that he made in the previous year).

We work on a log transform of the Salary variable) Salary is color-coded from low (blue, green) to high (yellow,red)

Decision tree for these data



Details of previous figure

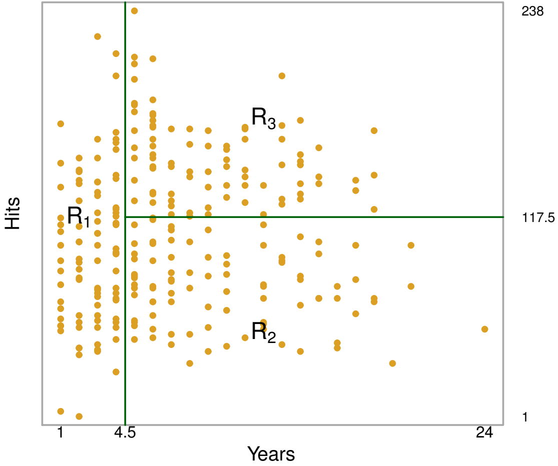
- For the Hitters data, a regression tree for predicting the log salary of a baseball player, based on the number of years that he has played in the major leagues and the number of hits that he made in the previous year.

- At a given internal node, the label (of the form $X_j < t_k$) indicates the left-hand branch emanating from that split, and the right-hand branch corresponds to $X_j \geq t_k$. For instance, the split at the top of the tree results in two large branches. The left-hand branch corresponds to $Years < 4.5$, and the right-hand branch corresponds to $Years \geq 4.5$.

- The tree has two internal nodes and three terminal nodes, or leaves. The number in each leaf is the mean of the response for the observations that fall there.

Results

Overall, the tree segments the players into three regions of predictor space: $R_1 = \{X | Years < 4.5g\}$, $R_2 = \{X | Years >= 4.5, Hits < 117.5g\}$, and $R_3 = \{X | Years >= 4.5, Hits >= 117.5g\}$.



Terminology for Trees

- In keeping with the tree analogy, the regions R_1, R_2, and R_3 are known as terminal nodes.
- 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 referred to as internal nodes
- In the hitters tree, the two internal nodes are indicated by the text $\textit{Years} < 4.5$ and $\textit{Hits} < 117.5$.

Interpretation of Results

- Years is the most important factor in determining Salary, and players with less experience earn lower salaries than more experienced players.
- Given that a player is less experienced, the number of Hits that he made in the previous year seems to play 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, and players who made more Hits last year tend to have higher salaries.
- Surely an over-simplification, but compared to a regression model, it is easy to display, interpret and explain.

The tree-building process-I

1. We divide the predictor space, the set of possible values for $X_1; X_2; \cdots ; X_p$ into J distinct and non-overlapping regions, $R_1; R_2; \cdots ; R_J$.
2. For every observation that falls into the region R_j , we make the same prediction, which is simply the mean of the response values for the training observations in R_j.

The tree-building process -II

In theory, the regions could have any shape. However, 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

  nd boxes $R_1; \cdots ; R_J$ that minimize the RSS, given by

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

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

The tree-building process -III

- Unfortunately, it is computationally infeasible to consider every possible partition of the feature space into J boxes.
- For this reason, we take a top-down, greedy approach that is known as recursive binary splitting.
- The approach is top-down because 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.
- It is greedy because 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.

The tree-building process -IV

- First select the predictor X_j and the cutpoint s such that splitting the predictor space into the regions $\{X_j < s\}$ and $\{X_j \geq s\}$ leads to the greatest possible reduction in RSS. If j and s are given we define the pair of hal-planes

  $$R_1(j, s) = \{X|X_j < s\} \; R_2(j, s) = \{X|X_j > s\}$$

  and we seek the value of j and s that minimize the equation

  $$\sum_{i;x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i;x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2$$

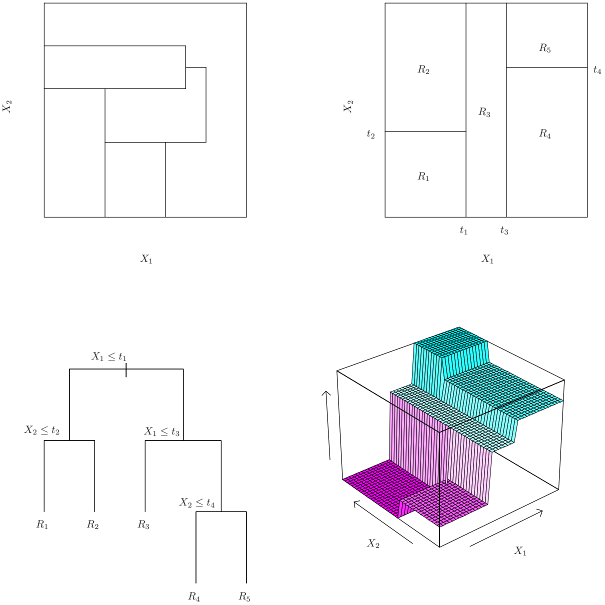
  where $\hat{y}_{R_1}$ is le mean response for training observation in $R_1(j, s)$ and $\hat{y}_{R_2}$ is le mean response for training observation in $R_2(j, s)$.

The tree-building process -V

- 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.
- However, 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; for instance, we may continue until no region contains more than five observations.

Predictions

- 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.
- A five-region example of this approach is shown in the next slide eg $\hat{y}_{R_1} = -5$, $\hat{y}_{R_2} = -7$, $\hat{y}_{R_3} = 0$, $\hat{y}_{R_4} = 2$, $\hat{y}_{R_5} = 4$


X_2
X_1
R_2
R_5
t_4
R_3
t_2
R_4
R_1
t_1
t_3
X_2
X_1
$X_1 \leq t_1$
$X_2 \leq t_2$
$X_1 \leq t_3$
$X_2 \leq t_4$
R_1
R_2
R_3
R_4
R_5
X_2
X_1

Top Left: A partition of two-dimensional feature space that could not result from recursive binary splitting.

Top Right: The output of recursive binary splitting on a two-dimensional example.

Bottom Left: A tree corresponding to the partition in the top right panel.

Bottom Right: A perspective plot of the prediction surface corresponding to that tree.

Pruning a tree-II

- A better strategy is to grow a very large tree T_0, and then prune it back in order to obtain a subtree
- Cost complexity pruning | also known as weakest link pruning | is used to do this
- we consider a sequence of trees indexed by a nonnegative tuning parameter α. For each value of α there corresponds a subtree $T \subset T_0$ such that

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

is as small as possible. Here $|T|$ indicates the number of terminal nodes of the tree T, R_m is the rectangle (i.e. the subset of predictor space) corresponding to the mth terminal node, and $\hat{y}_{R_m}$ is the mean of the training observations in R_m.

Pruning a tree-III

- The tuning parameter α controls a tradeoff between the subtree's complexity and its fit to the training data.
- Select an optimal value $\hat{\alpha}$ using cross-validation.
- Return to the full data set and obtain the subtree corresponding to $\hat{\alpha}$.

Algorithm

1. Use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations.

2. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees, as a function of α.

3. Use K-fold cross-validation to choose α. For each $k = 1; \cdots ; K$:

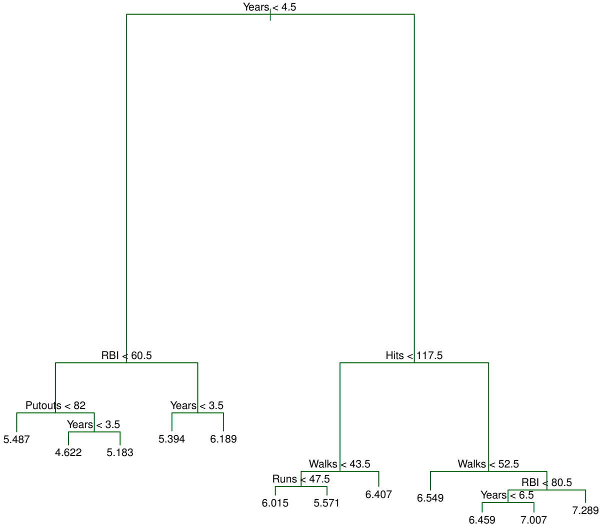
   3.1 1 Repeat Steps 1 and 2 on the $\frac{K-1}{K}$th fraction of the training data, excluding the kth fold.

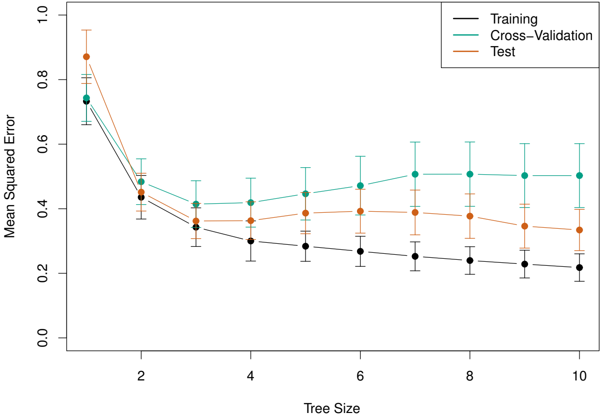
   3.2 Evaluate the mean squared prediction error on the data in the left-out kth fold, as a function of α. Average the results, and pick α to minimize the average error.

4. Return the subtree from Step 2 that corresponds to the chosen value of α.

Hitters data

- First, we randomly divided the data set in half, yielding 132 observations in the training set and 131 observations in the test set.
- Built a large regression tree on the training data and varied α in in order to create subtrees with different numbers of terminal nodes.
- Performed six-fold cross-validation in order to estimate the cross-validated MSE of the trees as a function of α.


Years < 4.5
RBI < 60.5
Hits < 117.5
Putouts < 82
Years < 3.5
Years < 3.5
5.487
4.622
5.183
5.394
6.189
Walks < 43.5
Walks < 52.5
Runs < 47.5
6.015
5.571
6.407
6.549
RBI < 80.5
Years < 6.5
6.459
7.007
7.289


Training
Cross–Validation
Test
Mean Squared Error
0.0
0.2
0.4
0.6
0.8
1.0
2
4
6
8
10
Tree Size

Classification Trees

- Very similar to a regression tree, except that it is used to predict a qualitative response rather than a quantitative one.
- For a clasification tree, we predict that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs.

Classification Trees-II

- Just as in the regression setting, we use recursive binary splitting to grow a clasification tree.
- In the clasification setting, RSS cannot be used as a criterion for making the binary splits
- A natural alternative to RSS is the classi

  cation error rate. this is simply the fraction of the training observations in that region that do not belong to the most common class:

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

  where $\hat{p}_{mk}$ represents the proportion of training observations in the mth region that are from the kth class.
- However classification error is not suficiently sensitive for tree-growing, and in practice two other measures are preferable.

Classification Trees-Gini index and Deviance

- 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. The Gini index takes on a small value if all of the $\hat{p}_{mk}$ are close to zero or one.

- For this reason the Gini index is referred to as a measure of node purity - a small value indicates that a node contains predominantly observations from a single class.

- An alternative to the Gini index is cross-entropy, given by

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

- It turns out that the Gini index and the cross-entropy are very similar numerically.

The case of Two Classes

Let p the proportion in the second class

- Misclassififcation error : $1 - max(p, 1 - p)$.
- Gini Index : $2p(1 - p)$
- Cross entropy : $plogp - (1 - p)log(1 - p)$
- Criteria are similar but cross entropy and gini index are differentiable

Gini index and entropy are used when growing tree. Misclassificetaion is uses when pruning the tree (if prediction accuracy of the final tree is the goal)