BIOS 835: Tree Based Classifiers

Alexander McLain

October 9, 2023

Outline

- Suppose we have two variables X_1 and X_2 and we are trying to classify group status.
- Tree based methods consider breaking the X_1 and X_2 space into blocks.

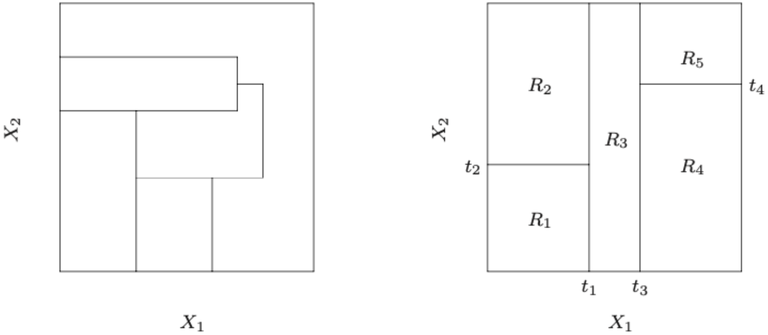


Figure: CART Example from ESL II (page 306).

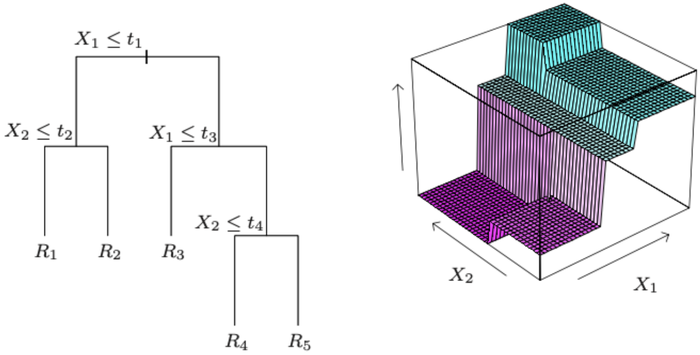


Figure: CART Example from ESL II (page 306).

- The model predict population C_m for $(X_1, X_2) \in R_m$.

Notation

- We now turn to the question of how to grow a regression tree.
- Our data consists of p inputs and a response, for each of N observations: that is, (x_i, y_i) for $i = 1, 2, \ldots, N$, with $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$.
- The algorithm decides what variable to split on and the split point.
- Suppose we have M regions (or nodes) $R_1, \ldots, R_m$ and we predict with:

$$f(x) = \sum_{m=1}^{M} c_m I\left(x \in R_m\right) \tag{1}$$

- If our criterion is minimum sums of squares then

$$\hat{c}_m = \text{ave}\left(y_i \mid x_i \in R_m\right)$$

Splitting

- To find the best partition among all possible splits is not possible (Why?).
- So, we'll use a greedy algorithm which will do the best at each point.
- Starting with all of the data, consider a splitting variable j and split point s, and define the pair of half-planes

$$R_1(j,s) = \{X \mid X_j \leq s\} \text{ and } R_2(j,s) = \{X \mid X_j > s\}$$

- Then we seek the splitting variable j and split point s that solve

$$\min_{j,s}\left[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2\right]$$

Splitting

- For any choice j and s, the inner minimization is solved by

$$\hat{c}_1 = \text{ave}\left(y_i \mid x_i \in R_1(j,s)\right) \text{ and } \hat{c}_2 = \text{ave}\left(y_i \mid x_i \in R_2(j,s)\right)$$

- We can scan through all possibilities of (j, s) to find the best pair.
- Once we find the best split we have two regions and repeat the process for each.

How far to grow?

- Tree size is a tuning parameter governing the model's complexity.
- The preferred strategy is to grow a large tree T_0, stopping the splitting process only when some minimum node size (say 5) is reached.
- The large tree is pruned using **cost-complexity pruning**.
- Denote the **subtree** $T \subset T_0$ to be any tree that can be obtained by pruning T_0.
- We index terminal nodes by m, with node m representing region R_m.
- Let $|T|$ denote the number of terminal nodes in T.

Cost complexity criterion

- Let $N_m = \#\{x_i \in R_m\}$ with

$$\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i$$

$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2$$

and $C(T) = \sum_{m=1}^{|T|} N_m Q_m(T)$ the 'cost' or 'risk' of tree T.

- Define the **cost complexity criterion** as

$$C_\alpha(T) = C(T) + \alpha |T|$$

Weakest link pruning

- For each α one can show that there is a unique smallest subtree T_α that minimizes $C_\alpha(T)$.
- To find T_α we use **weakest link pruning** where you we successively:
  1. collapse the internal node that produces the smallest per-node increase in $\sum_m N_m Q_m(T)$, and
  2. continue until we produce the single-node (root) tree.
- This gives a (finite) sequence of subtrees, and one can show this sequence must contain T_α.
- Estimation of α is achieved by five- or tenfold cross-validation: we choose the value $\hat{\alpha}$ to minimize the cross-validated sum of squares. Our final tree is $T_{\hat{\alpha}}$.

Introduction

- We now turn to the question of how to grow a Classification trees.
- Similar to before data consists of p inputs but now $y_i \in \{1, 2, \ldots, K\}$ is the outcome.
- Let $\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I\left(y_i = k\right)$ denote the proportion of class k observations in node m.
- We classify the observations in node m to class $k(m) = \arg\max_k \hat{p}_{mk}$.

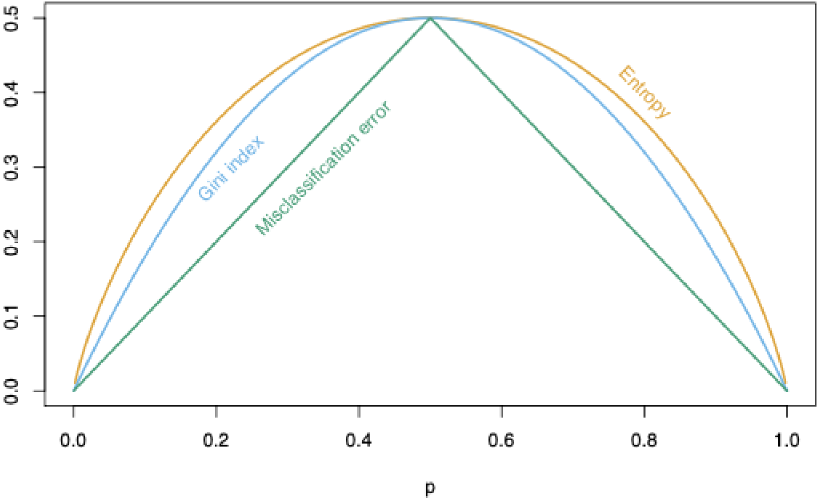
Introduction

- For classification trees, different measures $Q_m(T)$ of node impurity include:

$$\text{Misclassification error:} \quad \frac{1}{N_m} \sum_{i \in R_m} I\left(y_i \neq k(m)\right) = 1 - \hat{p}_{mk(m)}.$$

$$\text{Gini index:} \quad \sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk} \left(1 - \hat{p}_{mk}\right)$$

$$\text{Cross-entropy or deviance:} \quad -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$


0.0
0.1
0.2
0.3
0.4
0.5
0.0
0.2
0.4
0.6
0.8
1.0
p
Gini index
Misclassification error
Entropy