PS5841

Data Science in Finance & Insurance

Decision Tree

Yubo Wang

Spring 2022

Decision Trees

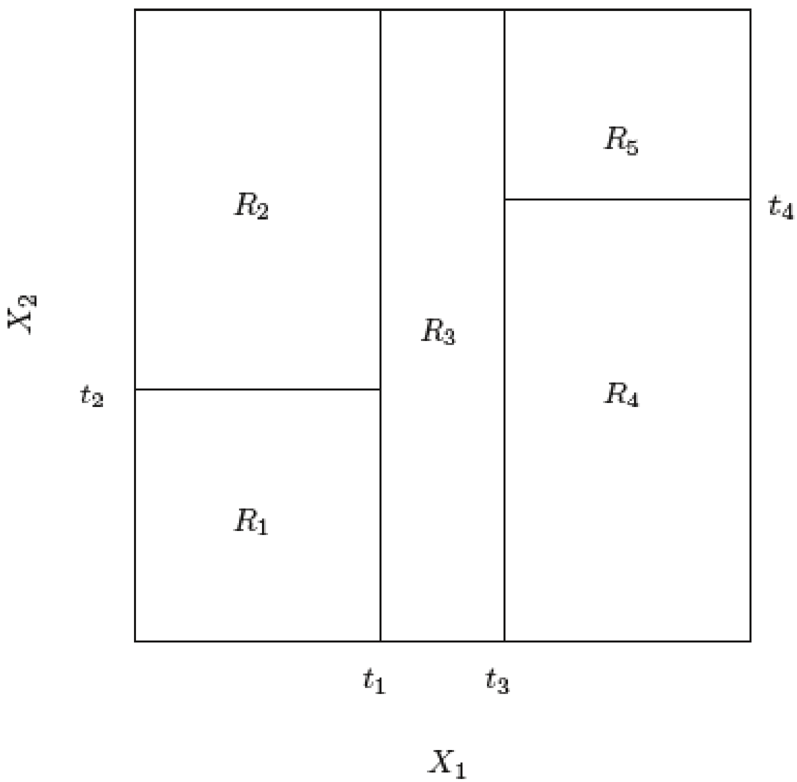
- Prediction via stratification of the feature space
  - Divide the predictor space into high-dimensional rectangles that minimizes “loss” via recursive binary splitting
  - Prediction based on the mean (for regression) or the most commonly occurring class (for classification) of training responses in the same terminal node

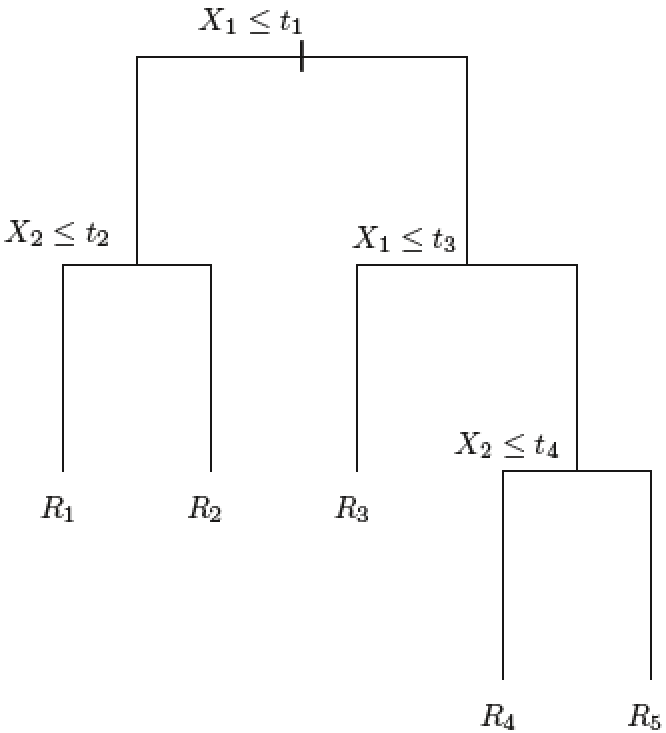
Recursive Binary Splitting

- Top-Down
  - Start from the top of the tree
- Greedy
  - The best split for a particular node is made at that particular step only, rather than taking into account of future steps
- Each split involves a cut-point s which splits a predictor X_j into two partitions
- Split a qualitative predictor into

$$\{X \mid X_j \text{ in classes up to } s\} \text{ and}$$
$$\{X \mid X_j \text{ in the rest of the classes}\}$$

Example





$$R_-(j,s) = \{X \mid X_j < s\}, R_+(j,s) = \{X \mid X_j \geq s\}$$

Find the values of j (feature) and s (cut point) that minimize “loss”

$$\sum_{i:\boldsymbol{x}_i \in R_-(j,s)} loss(y_i, \hat{y}_{R_-}) + \sum_{i:\boldsymbol{x}_i \in R_+(j,s)} loss(y_i, \hat{y}_{R_+})$$

Split Criteria ("loss")

- Regression
  - RSS

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

- Classification
  - Gini index
  - (Cross) Entropy

$\hat{p}_{R_m C_k}$ for Classification

- The proportion of training observations in the m-th region R_m that are from the k-th class C_k

$$\hat{p}_{mk} = \hat{p}_{R_m C_k} = \frac{n_{R_m C_k}}{n_{R_m}}$$

Classification Error Rate

- For the m-th region R_m

$$E_{R_m} = 1 - \max_k \left(\hat{p}_{R_m c_k}\right)$$

- Overall error rate
  - is the sum of the error rates over all regions
- Classification error rate is not sufficiently sensitive for tree-growing and is rarely used as a split criterium

Gini Index (1)

- For the m-th region R_m

$$G_{R_m} = \sum_{k=1}^{K} \hat{p}_{R_m c_k}\left(1 - \hat{p}_{R_m c_k}\right)$$

  – a measure of variance across the K classes for observations in that region

  – G_{R_m} will take on a small value if the m-th node is pure, containing predominantly observations from a single class

Gini Index (2)

- Overall Gini index

$$G = \sum_{k=1}^{K} \frac{n_{R_m}}{N} G_{R_m}$$

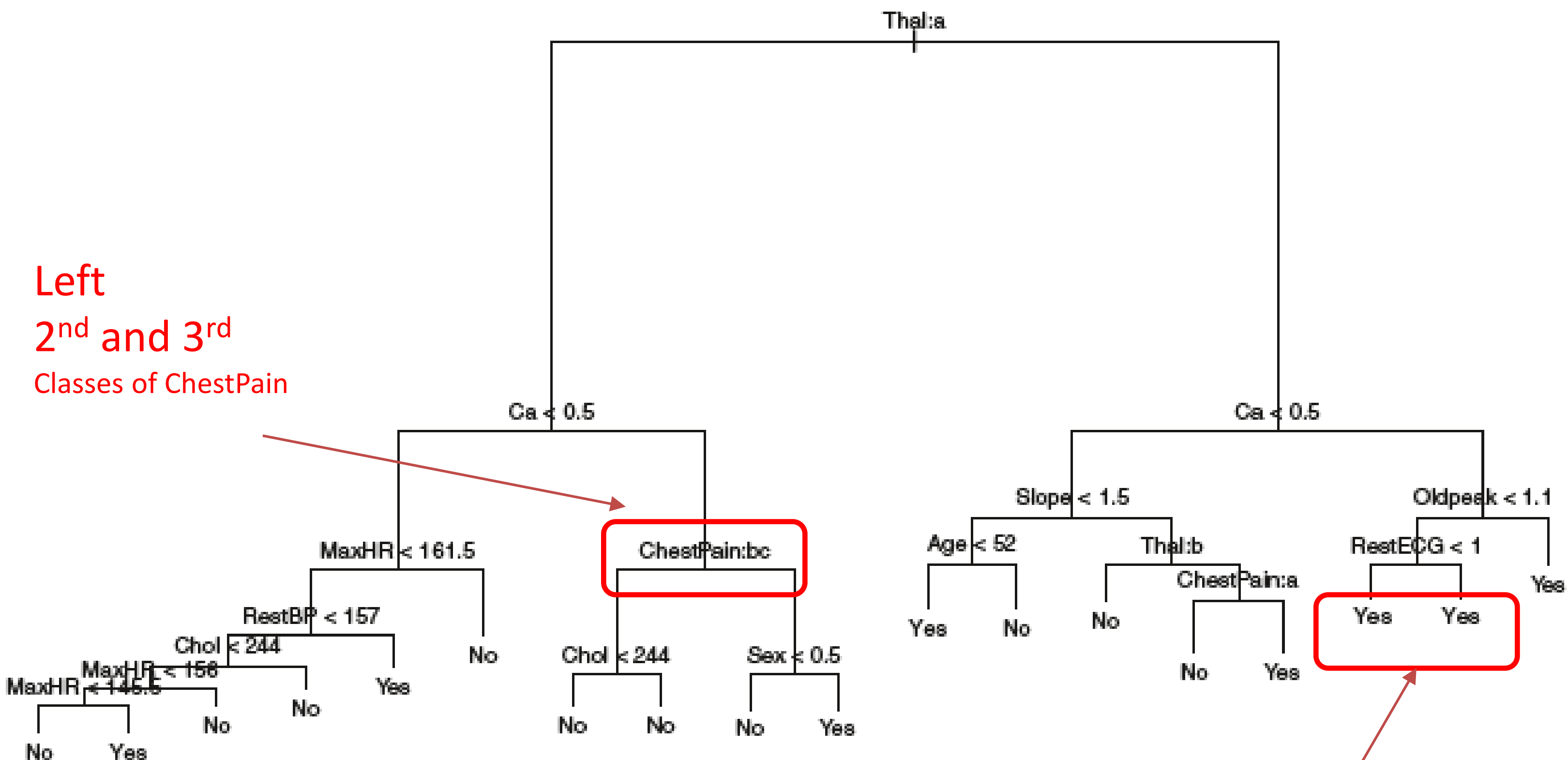
  - the pooled variance involving regional variances

Example: Binary Split on Gini Index

- 2-class responses and 2-D features X_1 and X_2
- Find the optimal split for predictor X_1
  - Find $s_{X_1}^*$ that minimizes G as $G^{X_1}(s_{X_1}^*)$
- Find the optimal split for predictor X_2
  - Find $s_{X_2}^*$ that minimizes G as $G^{X_2}(s_{X_2}^*)$
- If $G^{X_1}\left(s_{X_1}^*\right) < G^{X_2}(s_{X_2}^*)$, the current step splits X_1, otherwise, the current step splits X_2

Split, Node Purity

Left
2nd and 3rd
Classes of ChestPain



Split for node purity
Left $\hat{p}_{mk} = 0.64$
Right $\hat{p}_{mk} = 1.00$

- Node purity – the degree to which a node contains predominantly observations from a single class

Entropy

- For the m-th region R_m

$$D_{R_m} = -\sum_{k=1}^{K} \hat{p}_{R_m c_k} \log \hat{p}_{R_m c_k}$$

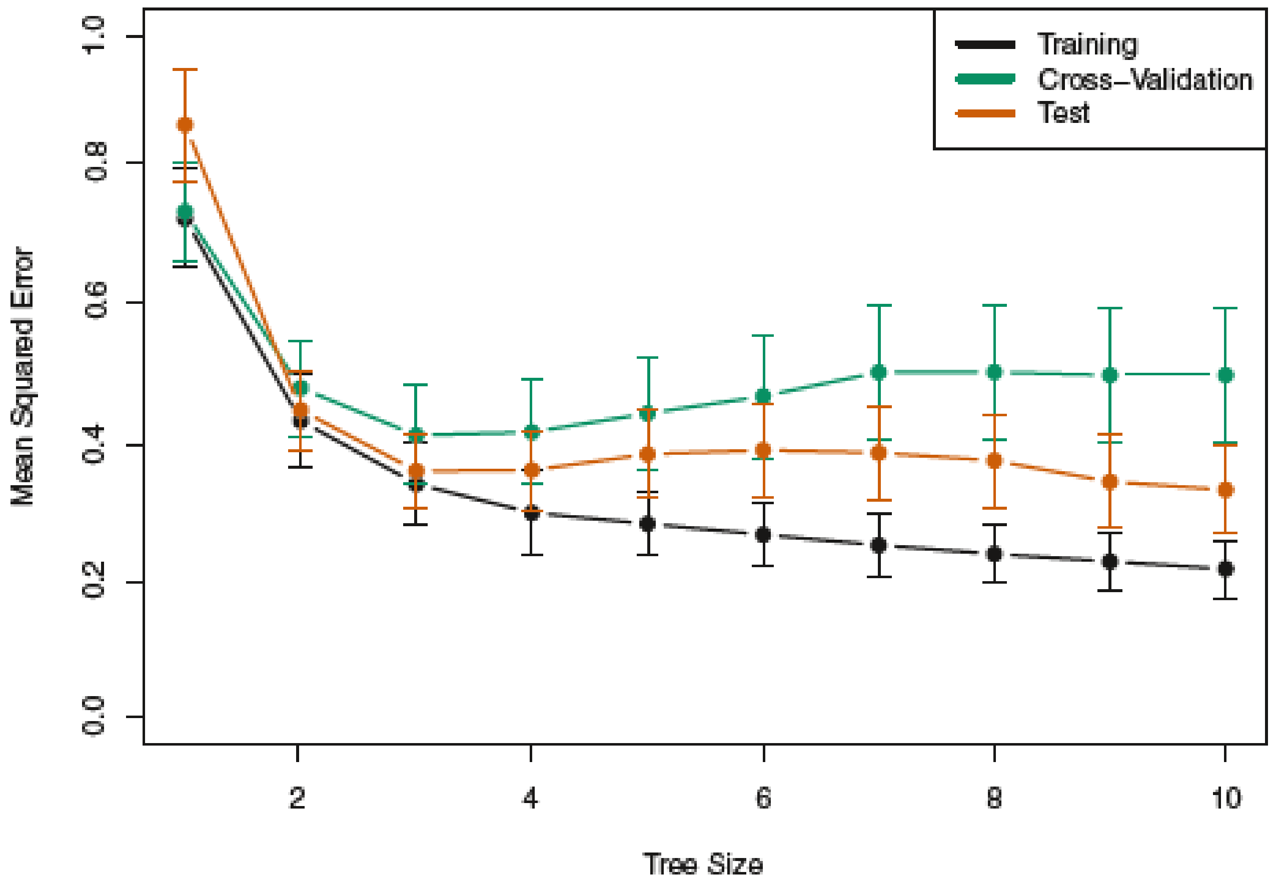
  - D_{R_m} will take on a small value if the m-th node is pure, containing predominantly observations from a single class
- Overall entropy
  - is the sum of entropy over all regions
- The Gini index and the entropy are quite similar numerically

Cost Complexity Pruning (Weakest Link Pruning)

- Pruning a tree to manage the risk of overfitting by a large tree T_0
- Each value of the tuning parameter $(\alpha \geq 0)$ corresponds to a subtree $T \subset T_0$ which minimizes

$$\text{error\_rate} + \alpha|T|$$

Example: Pruning



Building a Pruned Tree

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. That is, divide the training observations into K folds. For each $k = 1, \ldots, K$:

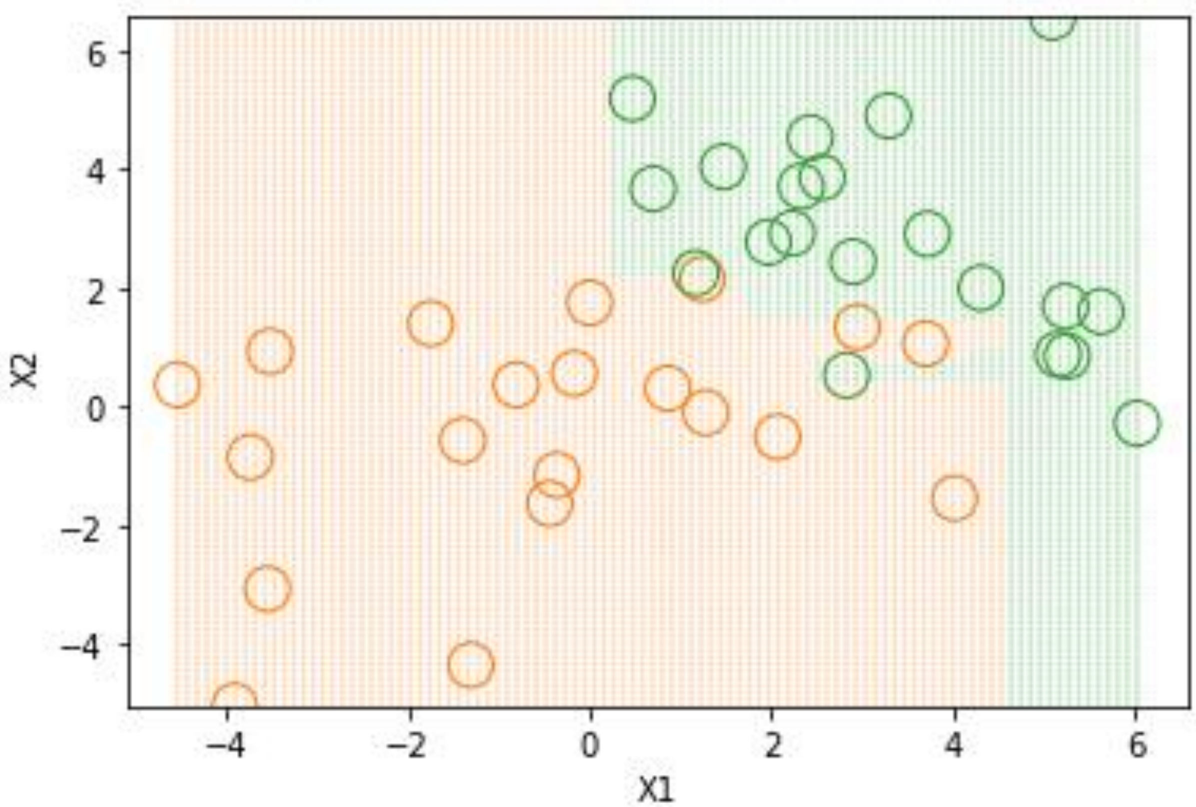
   (a) Repeat Steps 1 and 2 on all but the kth fold of the training data.

   (b) Evaluate the loss for pruning (RSS or Error Rate) on the data in the left-out kth fold, as a function of α.

   Average the results for each value of α, and pick α to minimize the average error.

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

Decision Boundary
random forest



That was