Topic 4: Tree-based Models

1. Tree-Based Methods
2. Regression Trees
3. Classification Trees

Suggested Reading

- "Principles of data mining", Hand, *et al.* Chap. 5.2 An example : The CART algorithm for building tree classifiers.

- "Modern applied statistics with S-Plus ", Venables and Ripley. Chap. 10.

- "The elements of statistical learning : data mining, inference, and prediction", Trevor, Tibshirani and Friedman. Chap. 9.2, main reference.

1. Tree-Based Methods

- Partitioning the feature space into rectangles.

  Feature space — space that includes all possible events.

- Two important references:

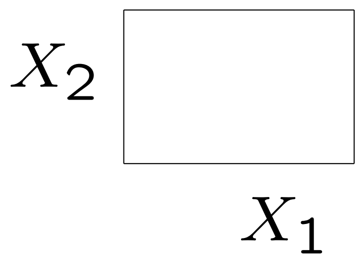
  CART. the book by Breiman, *et al*.

  C4.5 google the reference.

- Example: regression tree.

Regression Tree-1

- Setting: Response Y, inputs (explanatory variables) X_1, X_2.
- Feature space: all points (X_1, X_2).



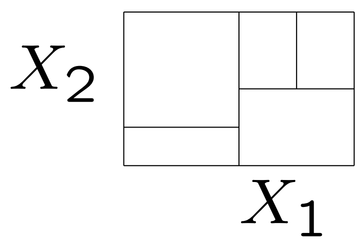
- The difficulty with a general regression model: there is no a, b, c such that

$$Y = aX_1 + bX_2 + c.$$

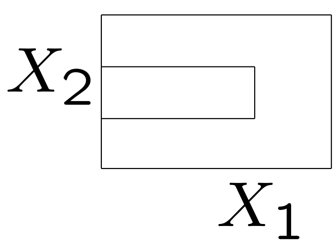
Regression Tree-2

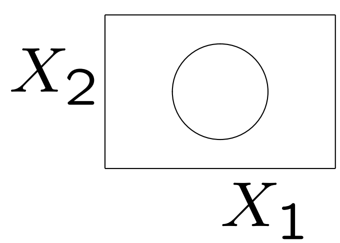
- Strategy: Recursively and dyadically partition the feature space so that within each small region, a regression model can be fit.

- A possible partition:

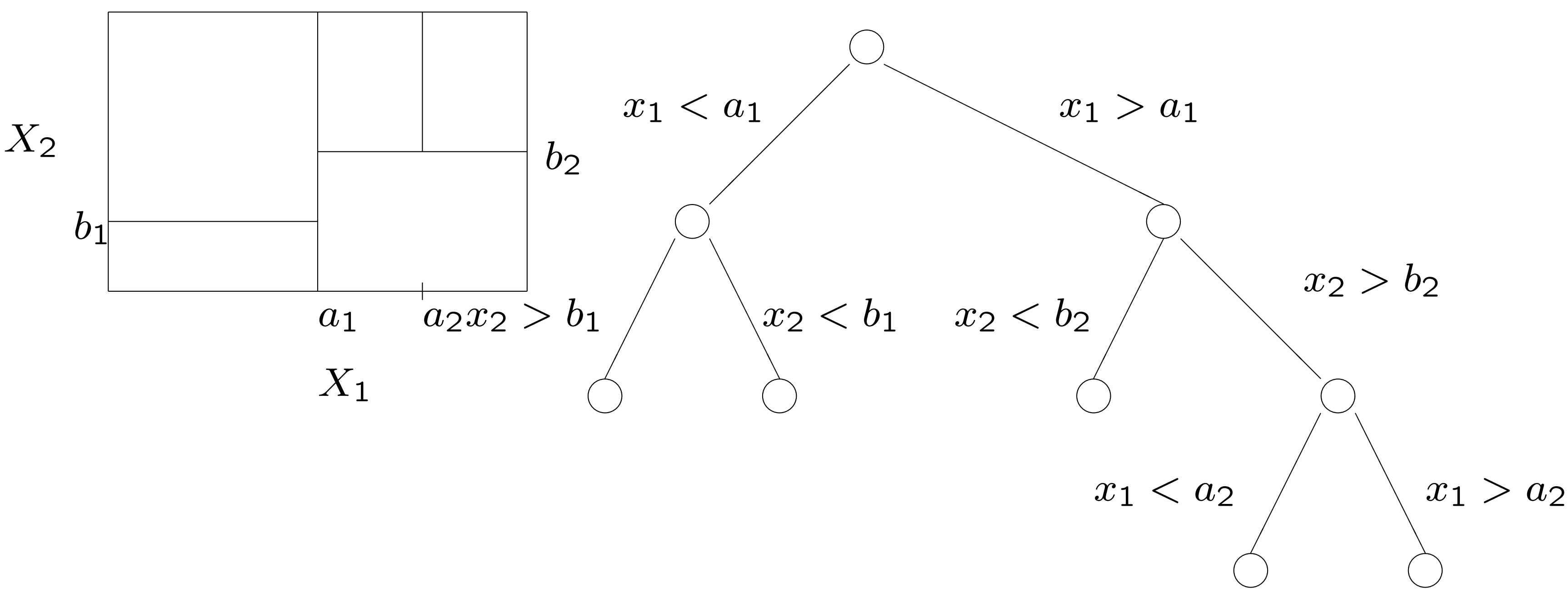


- Impossible partitions:





Why this is a tree method? — "A recursive dyadic partition can be associated with a binary tree."



A regression model

We only consider constant fit in each region.

$$\widehat{f}(X) = \sum_{m=1}^{5} C_m I\{(X_1, X_2) \in R_m\}.$$

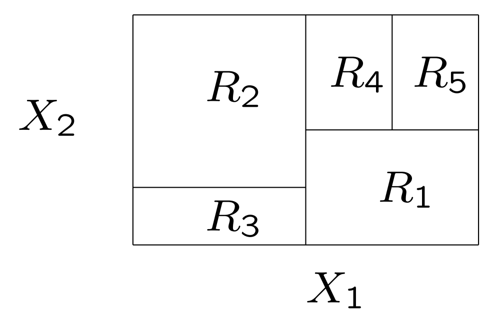
Here

X — (X_1, X_2)

C_m — a constant

$I\{(X_1, X_2) \in R_m\}$ — indicator function

$$I\{(X_1, X_2) \in R_m\} = \begin{cases} 1 & \text{if}(X_1, X_2) \in R_m \\ 0 & \text{otherwise.} \end{cases}$$



C_m's need to be determined.

- How to generalize?

  For example, replace C_m with "$a_m X_1 + b_m X_2 + c_m$" (a linear model).

- Advantage of the recursive dyadic tree:

  - interpretability.

  - easy to generalize to high dimensional space.

2. Regression Trees

How to grow a regression tree?

Notations:

x_i — ith input, $\qquad i = 1, 2, ..., N.$

$x_i = (x_{i1}, x_{i2}, ..., x_{ip})$ — p - variates.

y_i — ith response.

(x_i, y_i) — ith observation.

R_i — ith region in the recursive dyadic partitioning.

C_m — the constant value in each region.

The model:

$$f(x) = \sum_{m=1}^{M} C_m I\{x \in R_m\}.$$

Things need to be determined:

- estimate C_m's.
- determine regions.

- Estimating C_m's.

  If we want to minimize the sum of square which is $\sum_{i=1}^{N}(y_i - f(x_i))^2$, and suppose that R_m's are fixed, then

  $$\widehat{C}_m = \text{ave}(y_i | x_i \in R_m).$$

  This is because

  $$\sum_{i=1}^{N}(y_i - f(x_i))^2 = \sum_{m=1}^{M} \sum_{x_i \in R_m} (y_i - C_m)^2$$

  is equivalent to choose C_m such that $\sum_{i, x_i \in R_m}(y_i - C_m)^2$ is minimized. The solution is $\widehat{C}_m = \text{ave}(y_i | x_i \in R_m)$.

- How to specify regions?

  More complicated analysis is postponed. Here we study a look-ahead strategy.

  – Key idea: Starting from the entire space, at each step, partition a region into two smaller regions based on the rule "$x_j \leq s$" or "$x_j > s$". Within each region, suppose we can choose x_j and $x_j = s$ as the splitting criterion, there will be two regions.

Continuation...

$$R_1(j,s) = \{x|x_j \le s\},$$
$$R_2(j,s) = \{x|x_j > s\}.$$

We need j, s that solve

$$G(j,s) = \min_{j,s}[\min_{C_1} \sum_{x_i \in R_1(j,s)} (y_i - C_1)^2 + \min_{C_2} \sum_{x_i \in R_2(j,s)} (y_i - C_2)^2].$$

Based on a previous argument we have

$$\hat{C}_1 = \text{ave}(y_i|x_i \in R_1(j,s)),$$
$$\hat{C}_2 = \text{ave}(y_i|x_i \in R_2(j,s)).$$

- For fixed j, s can be determined quickly, $G(j,s)$

$$
\begin{aligned}
&= \min_{j,s}[\sum_{x_i \in R_1(j,s)} (y_i - C_1)^2 + \sum_{x_i \in R_2(j,s)} (y_i - C_2)^2] \\
&= \min_{j,s}[\sum y_i^2 - \widehat{C}_1^2.\#\{x_i \in R_1(j,s)\} - \widehat{C}_2^2.\#\{x_i \in R_2(j,s)\}]
\end{aligned}
$$

Denote

$y_{(1)}$	$y_{(2)}$	$\cdots$	$y_{(N)}$	ordered y_i
S_1	S_2	...	S_N	partial sum sequence

Equivalent to: $\max_k S_k^2/k + (S_N - S_k)^2/(N-k)$.

- The procedure is applied to all x_j's. The order of complexity is $O(N)$.

- The size of the tree.
  - Large tree $\Rightarrow$ small residual sum of squares.

    Large tree $\rightarrow$ overfit.

    An extreme case is that when the tree is large enough, the residual sum of square is zero.

- Penalization on the size of the tree.
  - $|T|$ — size of a tree, # of nodes.
  - $RSS(T)$ — residual sum of squares.

$$RSS(T) = \sum_{m=1}^{|T|} RSS_m(T, x, y)$$

$$RSS_m(T, x, y) = \sum_{x_i \in R_m} (y_i - \widehat{C}_m)^2$$

$$\widehat{C}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i$$

$R_1, R_2, ..., R_{|T|}$ — regions associated with the tree

N_m — number of observations in R_m.

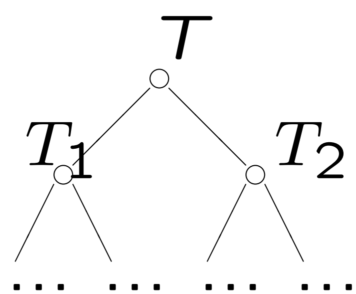
- Cost complexity criterion

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

  Here α is a parameter. In one look-ahead step, if the reduction of RSS is less than α, then the above criterion will favor no partition.

- Large $\alpha \rightarrow$ small tree.
  Small $\alpha \rightarrow$ large tree.

- A tree pruning algorithm: For a fixed tree T, a sub-tree of T, denoted by T_0, which minimizes $C_\alpha(T'), (T' \in T)$ can be quickly solved by following a tree pruning procedure.
  - Additivity of the cost complexity function: Both function $RSS(T)$ and function $|T|$ is additive.



$$RSS(T) = RSS(T_1) + RSS(T_2), \qquad |T| = |T_1| + |T_2|.$$

Key idea: a bottom-up pruning algorithm.

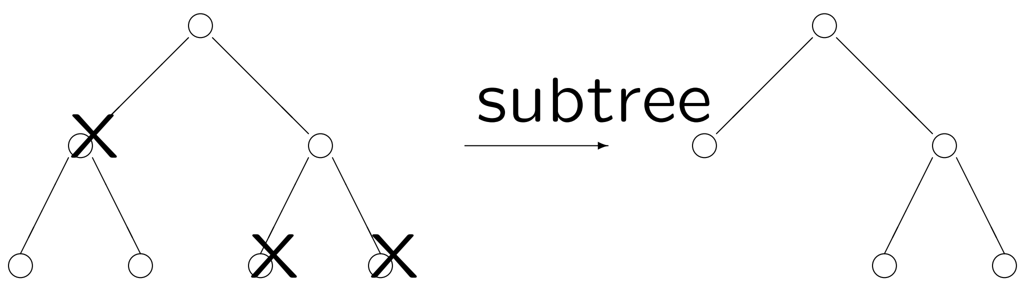
- Tree pruning algorithm

$$C_\alpha(N) \quad \text{vs} \quad C_\alpha(N_1) + C_\alpha(N_2)$$

  - $C_\alpha(N)$ corresponds to "fit with no split".
  - $C_\alpha(N_1) + C_\alpha(N_2)$: "fit with splitting".

This procedure is repeated until we reach the top node of the tree. The final survival subtree is the tree that minimizes the function $C_\alpha(N)$.

A *subtree*:



- How to choose α?

  What is *cross validation*?

  observations:

$$\boxed{p_1} \quad \boxed{p_2} \quad \boxed{p_3} \quad \boxed{p_4} \quad \boxed{p_5} \quad \cdots\cdots$$

p_i - ith subset of the observations.
$\hat{f}^{-k}(x,\alpha)$ - model fitted while assuming parameter α and kth subset is excluded.

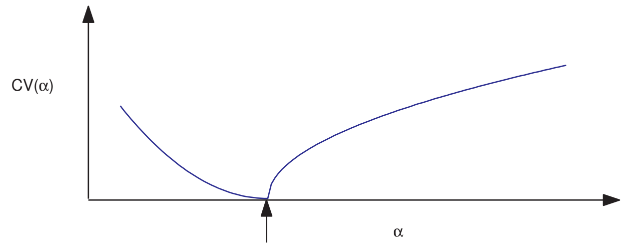
$$CV(\alpha) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i,\alpha))$$

$$CV(\alpha) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i, \alpha))$$

N	—	#of observations.
(x_i, y_i)	—	ith observation.
$k(i)$	—	the index of the subset used above.
		e.g., if (x_i, y_i) is in the first subset, k(i)=1.
L	—	loss function.

$CV(\alpha)$ provides an estimate of the test error curve.

We pick the α_0 w (α).



Cross validation finds the α that minimizes the objective.

- Ten folded cross validation: the observations are equally divided into 10 subsets.

- Five folded cross validation: 5 subsets.

- Leave-one-out CV: each observation is a subset of "$k(i) = i$" or equivalently.

3. Classification Trees

Recall in a regression tree, we have,

$$f(x) = \sum_{m=1}^{M} C_m I(x \in R_m).$$

In classification, $y_i \in \{1, 2, ..., k\}$,

response:

$$y = k, \quad \text{w.p. } p_{mk} \quad \text{while } x_i \in R_m.$$

We have

$$\widehat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k).$$

Recall in a regression model,

$$RSS(T) = \sum_{m=1}^{|T|} \sum_{x_i \in R_m} (y_i - \widehat{C}_m)^2.$$

In classification, it is changed to misclassification error:

$$Q_m(T) = \frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \widehat{p}_{mk(m)}.$$

$$k(m) = \arg\max_k \widehat{p}_{mk}$$

- Gini index:

$$\sum_{k \neq k'} \widehat{p}_{mk}\widehat{p}_{mk'} = \sum_{k=1}^{K} \widehat{p}_{mk}(1 - \widehat{p}_{mk})$$

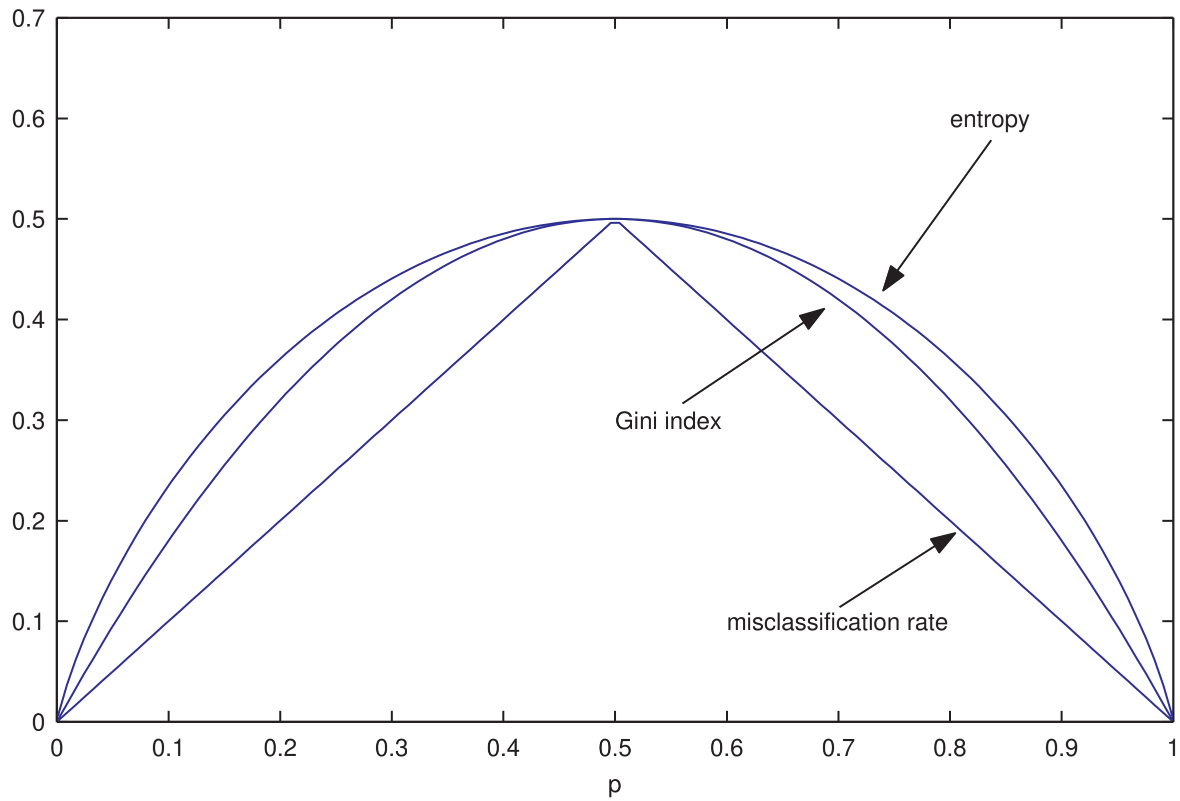
  Overall:

$$\sum_{m} \frac{N_m}{N_T} \sum_{k=1}^{K} \widehat{p}_{mk}(1 - \widehat{p}_{mk}).$$

- Entropy:

$$-\sum_{k=1}^{K} \widehat{p}_{mk} \log \widehat{p}_{mk}.$$

Comparing three criteria: misclassification rate, Gini index, and entropy.



Gini index and cross entropy are more sensitive.

Example: two class problem (400,400).

A split (300,100) and (100, 300)

B split (200, 400) and (200, 0)

The Gini index and entropy are lower for B split, while the misclassification rates are the same.

$$\text{misclassification rate} = \frac{200\text{misclassified}}{800} = 0.25.$$

Gini index for A: $1/2 \times 1/4 \times 3/4 + 1/2 \times 1/4 \times 3/4 = 3/16.$

Gini index for B: $600/800 \times 1/3 \times 2/3 + 0 = 1/6 < 3/16.$