Given: TS $(\underline{x}_1, y_1) \dots (\underline{x}_n, y_n)$

Goal: Estimate $E(Y \mid \underline{x})$

CART builds a model (prediction rule) that is piecewise constant over axis parallel boxes

$$\hat{f}(\underline{x}) = \sum_{j=1}^{K} c_j \, I(\underline{x} \in N_j)$$

where $N_i \cap N_j = \emptyset$ for $i \neq j$

$\bigcup N_i = \mathbb{R}^p$

- For given $N_1 \dots N_K$ $c_j = \text{mean}(y_i \mid \underline{x}_i \in N_j)$ minimizes resubstitution error.
- Can find optimal partition of $\mathbb{R}^p$ into two boxes ($K=2$) using exhaustive search
- Work for general $K$ grows exponentially in $K$

Idea: Apply splitting procedure for $K=2$ recursively:

Find the best split into 2 boxes $N_1$ $N_2$

Find the best split of $N_1$ $N_{11}$ $N_{12}$

best split of $N_2$ $N_{21}$ $N_{22}$

Greedy optimization $\Rightarrow$ $N_{11}$ $N_{12}$ $N_{21}$ $N_{22}$ will not be the optimal partition into 4 boxes

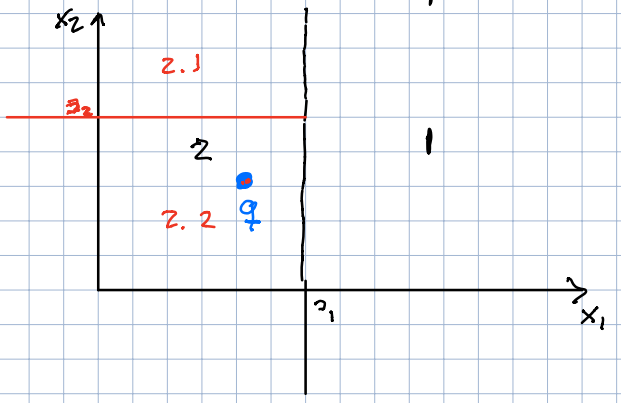
Model can be represented as a binary tree

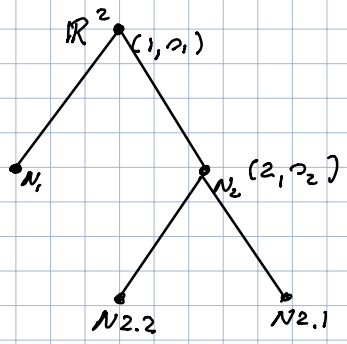
Each node of the tree corresponds to

- a box in predictor space
- a subset of the training sample (all training obs for which predictor vector fall into box)
- a constant $c_j$ : average response for training obs in box $N_j$

In addition, each internal node is associated with a split coordinate and a split point

Illustration for $p = 2$




q

Look-up (making prediction) for query point q is very efficient - simply run q down the tree

Note: Every sub-tree of the tree represents a piecewise constant model

CART can handle unordered categorical predictors.

Suppose categories are $a_1, \ldots a_h$

Need to partition $a_1 \ldots a_h$ into subsets $A_1$ $A_2$

Naive idea: consider all possible partitions

Key fact: We do not have to consider all passible partitions

Suppose $X_u$ is categorical
Define $\bar{y}_j$ = mean $(y_i \mid x_i \in N,\ x_{iu} = a_j)$

Def: A partition of the categories is called monotone if $a_i \in A_1$ $a_j \in A_2 \rightarrow \bar{y}_i \leq \bar{y}_j$

Fact: Need only consider monotone partitions

Treatment of missing predictors.

In practice we often have missing values for some of the predictor vectors.

Bad idea: Throw out all incomplete training observations

Why bad: Incomplete training obs may

still contain useful information

- $E(Y|\underline{x})$ may not depend on missing predictors.
- There might be other predictors highly correlated with the missing predictor.

Missing predictors cause problems at two points

1) Deciding on split coordinate and split point
2) Deciding into which daughter node each obs go.

Solution

(1) Compute reduction in RSS only for those obs for which predictor variable is not missing.

(2) Use surrogate splitting.

Controlling model complexity.

Idea 1: Stop splitting when you run out of data.

Idea 2: Stop splitting if all the $y_i$ for obs in box are the same.

Consequence of idea 1: The prediction rule may have high variance. We are interpolating the training data.