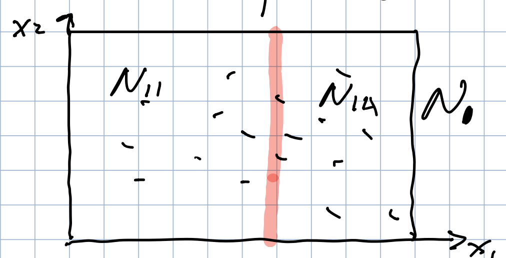
CART: Controlling model complexity

Idea: Stop splitting when you run out of data
Problem: Prediction rule may have high variance

Idea: Stop splitting when the reduction in RSS
is $< \alpha\%$ for some α.

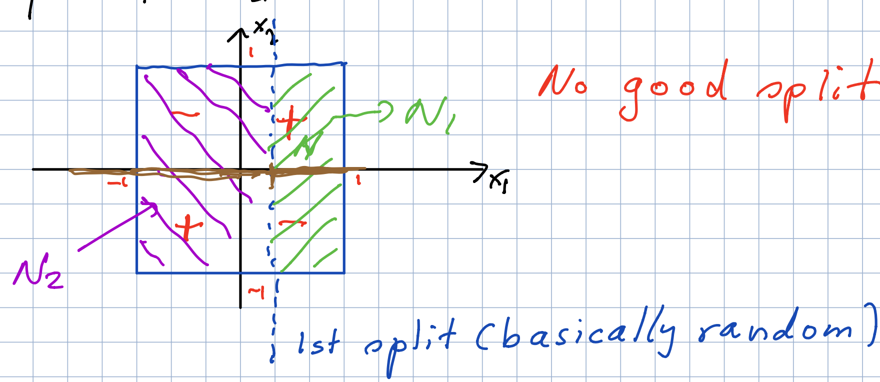


RSS(N_1) resid ss of squares for box N_1

$$RSS(N_{11}) + RSS(N_{12}) \leq RSS(N_1)$$

Problem: Next split may not help much, but splits further down the tree may help.

Example: $(X_1, X_2) \sim U[-1, 1]^2$
$$Y = X_1 * X_2$$

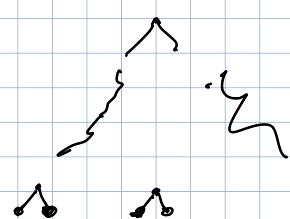


First (random) split breaks symmetry.
Further splits help.

Better idea:

- Grow big tree and then prune back
- Estimate optimal tree size by cross-validation

Problem: Estimating # of leaves by CV does not work: There are many subtrees with the same # of leaves.



← n leaves

For example, there are $n/2$ ways of making a tree with $(n-1)$ leaves out of tree with n leaves. This motivates

Minimal cost-complexity pruning.

Define $c(t)$ "cost of node t"

$c(t) = RSS(t)$ if t is a leaf

$c(t) = c(\ell(t)) + c(r(t)) + \lambda$ otherwise

λ = cost of increasing complexity by splitting

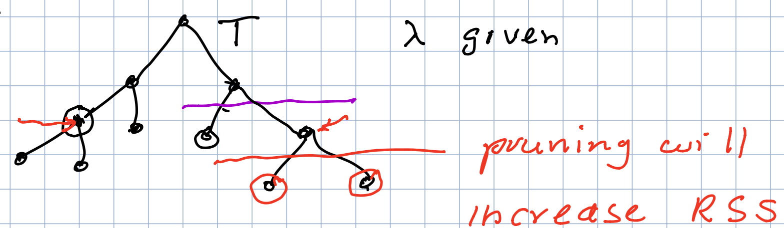
$\ell(t)$ $r(t)$: left and right daughters of t

T: original tree (split down to bucket size 1)

$T(\lambda)$: Minimum cost subtree for complexity parameter λ

Important: For given λ $T(\lambda)$ can be found by bottom up recombination

- Traverse the tree depth first
- Recombine nodes if $RSS(t) \leq c(\ell(t)) + c(r(t)) + \lambda$



- $\lambda = 0$ $\quad T(0) = T$ original tree
- $\lambda = \infty$ $\quad T(\infty) =$ root node
- $\lambda_1 > \lambda_2 \Rightarrow T(\lambda_1)$ is subtree of $T(\lambda_2)$

- If T has K leaves then there are at most K different subtrees that can be obtained by choosing different λs.

Bagging (Bootstrap aggregation)
(invented by Leo Breiman)

Given: TS $S = \{(\underline{x}_1, y_1) \ldots (\underline{x}_n\ y_n)\} \sim iid\ (\underline{X}, Y)$

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

Let $\hat{f}(\underline{x}, S)$ be the estimate of $f(\underline{x})$ based on training sample S.

Let $S_1^* \ldots S_B^*$ be Bootstrap samples from S

"Bootstrap sample" is sample of size n drawn with replacement from S.

Define bagged estimate

$$\hat{f}_{Bag}(\underline{x}, S) = \frac{1}{B} \sum_{i=1}^{B} \hat{f}(\underline{x}, S_i^*)$$