Lecture 22 : trees

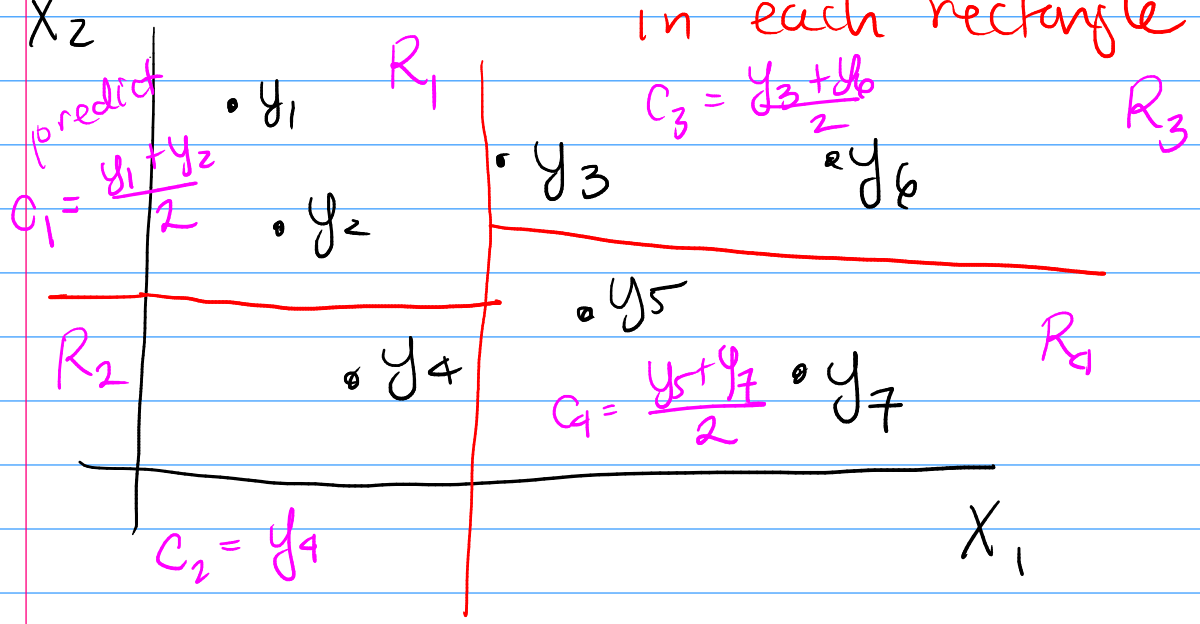
CARTs - Classification and regression trees

Regression Trees

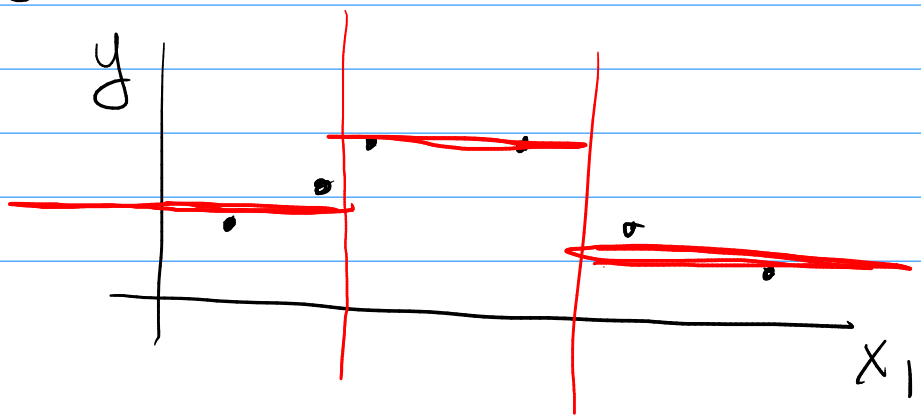
Basic idea:

1. break up X-space into rectangles
2. fit a simple model on each rectangle

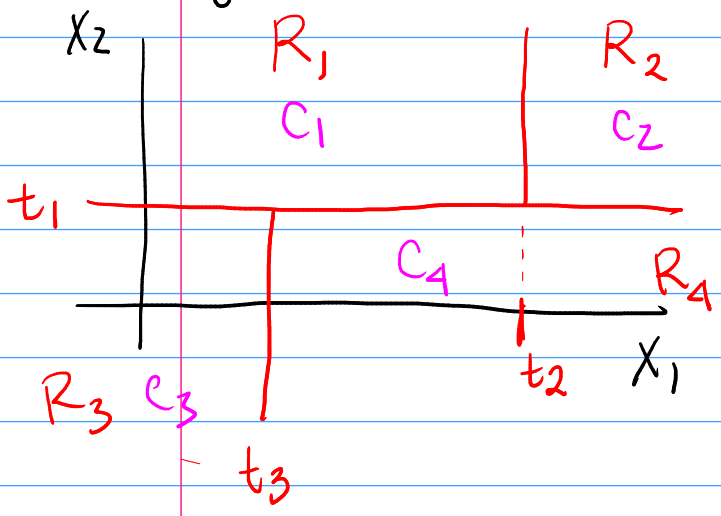
(predict the mean of train ys in each rectangle)

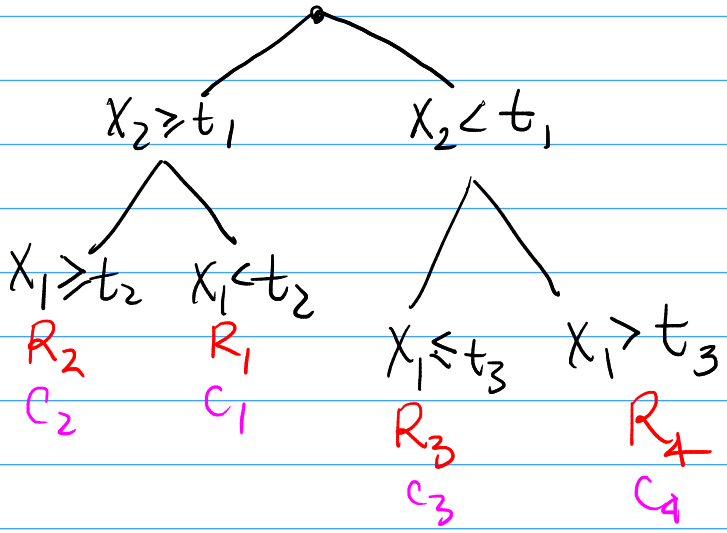


$$\hat{y} = \hat{f}(x) = c_i \text{ if } x \in R_i$$



Why called a tree? Can represent as a decision tree.





Goal: build a good tree

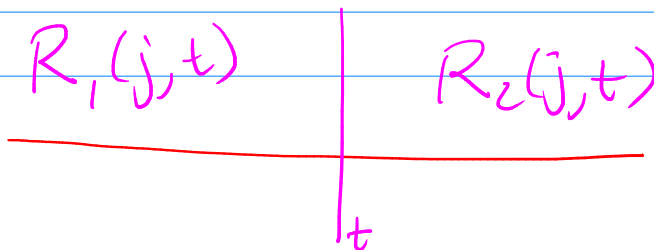
Need to decide

1. which variable I split on
2. where I split this variable
3. when do I stop splitting

Optimal approach: try all trees - not computationally tractible

So use same greedy approach.

Define $R_1(j,t)$ and $R_2(j,t)$ to be the half-spaces if we split variable j at point t



We can define the RSS if we make this split as

$$RSS(j,t) = RSS(R_1(j,t)) + RSS(R_2(j,t))$$

$$= \sum (y_i - c_1)^2 + \sum (y_i - c_2)^2$$

i in $R_1(j,t)$; mean of ys in $R_1(j,t)$; similar for $R_2(j,t)$

Algorithm to choose j and t

1. For each var j search over possible t and calc. $RSS(j,t)$
2. Choose j and t that minimize $RSS(j,t)$.
3. recursively do this for each half space.

When do we stop?

→ too many splits danger is overfitting

→ too few danger is underfitting

Bad strategy very greedy approach and split until RSS falls below some threshold

problem: a bad split might lead to

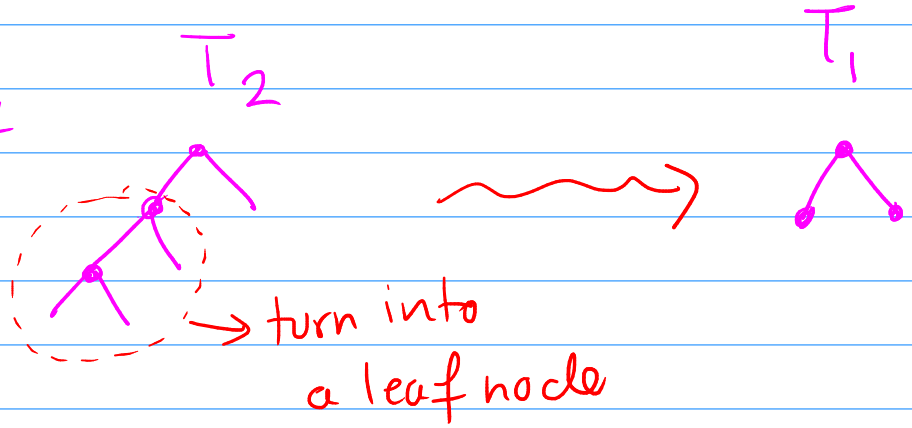
an even better split later

better approach:

1. grow a really large tree (overfit)
2. reduce its size by pruning

Aside: a tree T_1 is called a subtree of T_2 if I can get T_1 by "collapsing" part of T_2

Ex.



In particular use "cost-complexity pruning"

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

$\alpha \geq 0$

$|T|$: size of tree i.e. # of leaf nodes

Given some $\alpha \geq 0$

(1) grow a large tree

(2) search over sub-trees and choose sub-tree that minimizes $C_\alpha(T)$

If $\alpha = 0 \Rightarrow$ Choose largest subtree i.e. remove nothing

$\alpha = \infty \Rightarrow$ choose smallest subtree i.e. remove everything and don't split

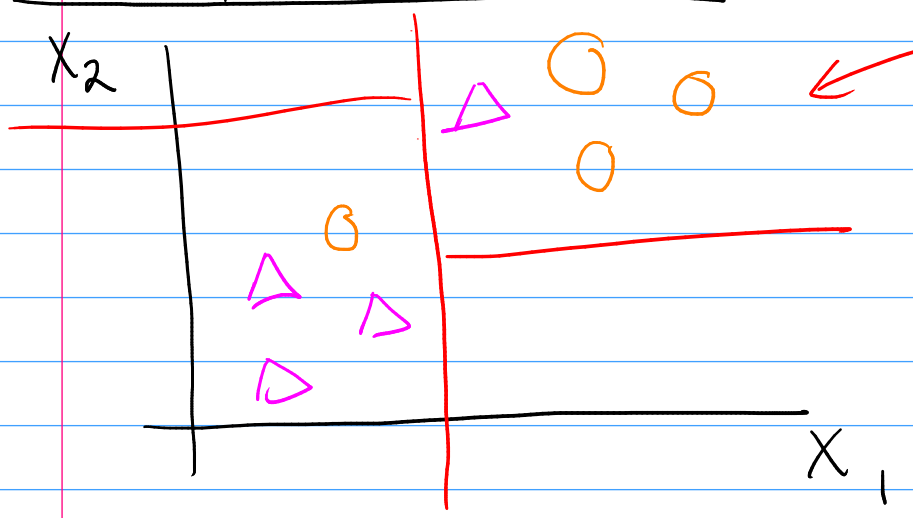
---

Turns out that if $\alpha_1 \leq \alpha_2$ then the optimal tree for α_1 (call it T_1) and optimal tree for α_2 (call it T_2) are related so that

$T_2 \subset T_1$ (sub-tree)

So as a increase α we get a nested series of trees.

Classification Trees



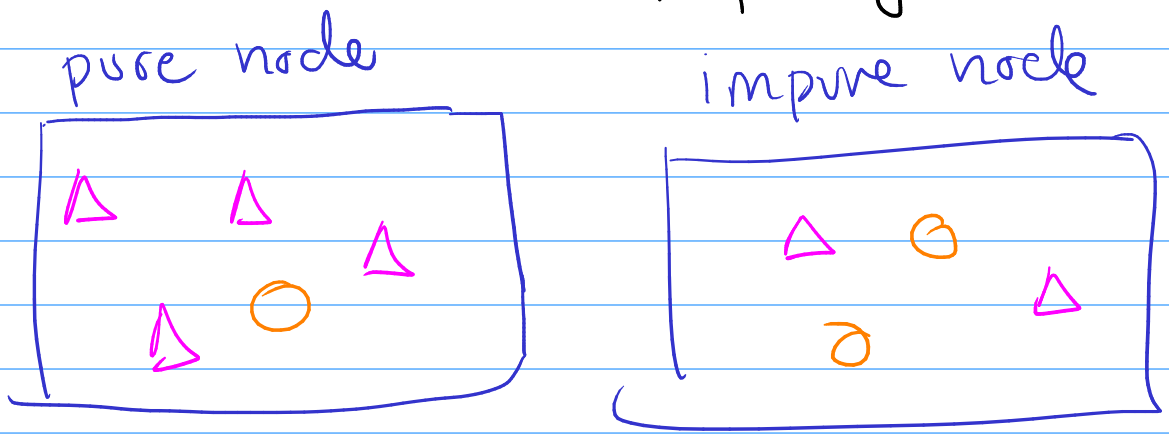
predict the majority (plurality) class in each rectangle

What makes a good split?

Regression Trees: RSS

Classification Trees: node (rectangle) impurity

Ex,



Now we split to reduce node impurity

Node Impurity Measures If p_k = pct. of class k in node

(1) mis. class. rate: $1 - p_{\hat{k}}$ where $\hat{k}$ = most common class

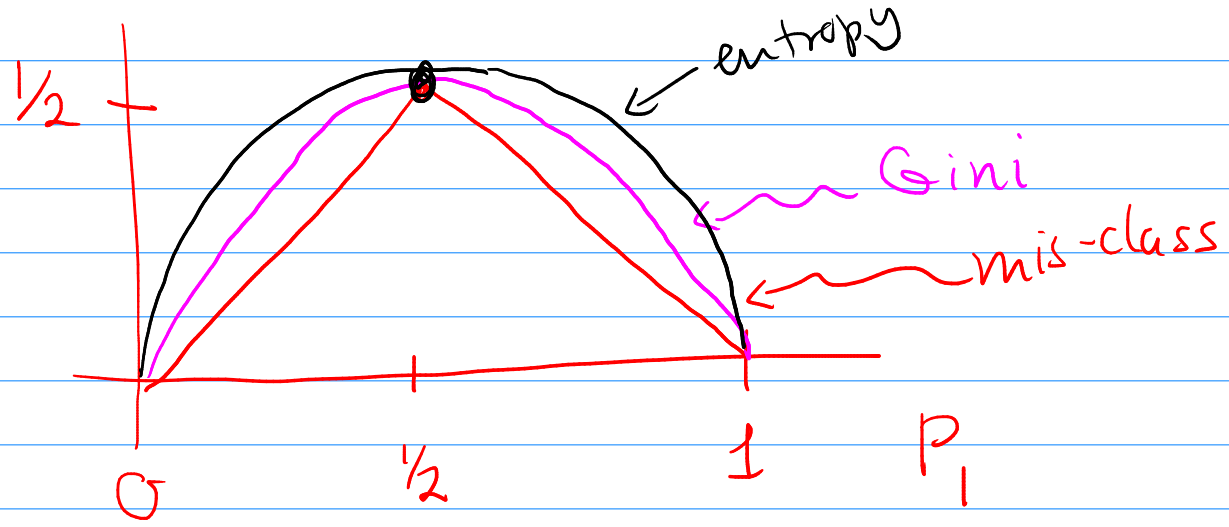
(2) Gini-Index

$$\sum_k p_k(1-p_k)$$

For all:
pure ≈ 0
impure $\approx 1/K$

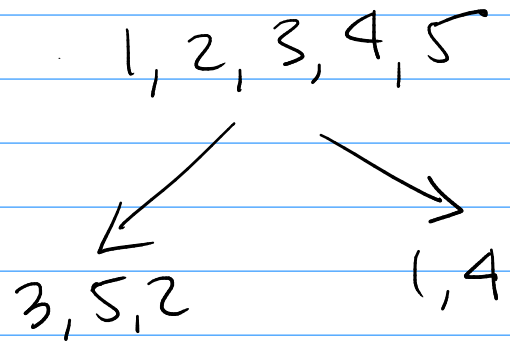
(3) Entropy: $\sum_k p_k \log p_k$

$K=2$ (two-class problem)



What about categorical vars?

Splitting a cat. var. means dividing the cats into two groups:



Problem: If I have q levels in my cat. var then I have 2^{q-1}

possible splits.

CARTs can deal w/ missing data nicely.

Cat. vars just add a "missing" category

Numeric vars. Keep track of "surrogate" splits i.e. splits using other vars that have similar divisions

idea: use surrogate if var is missing.