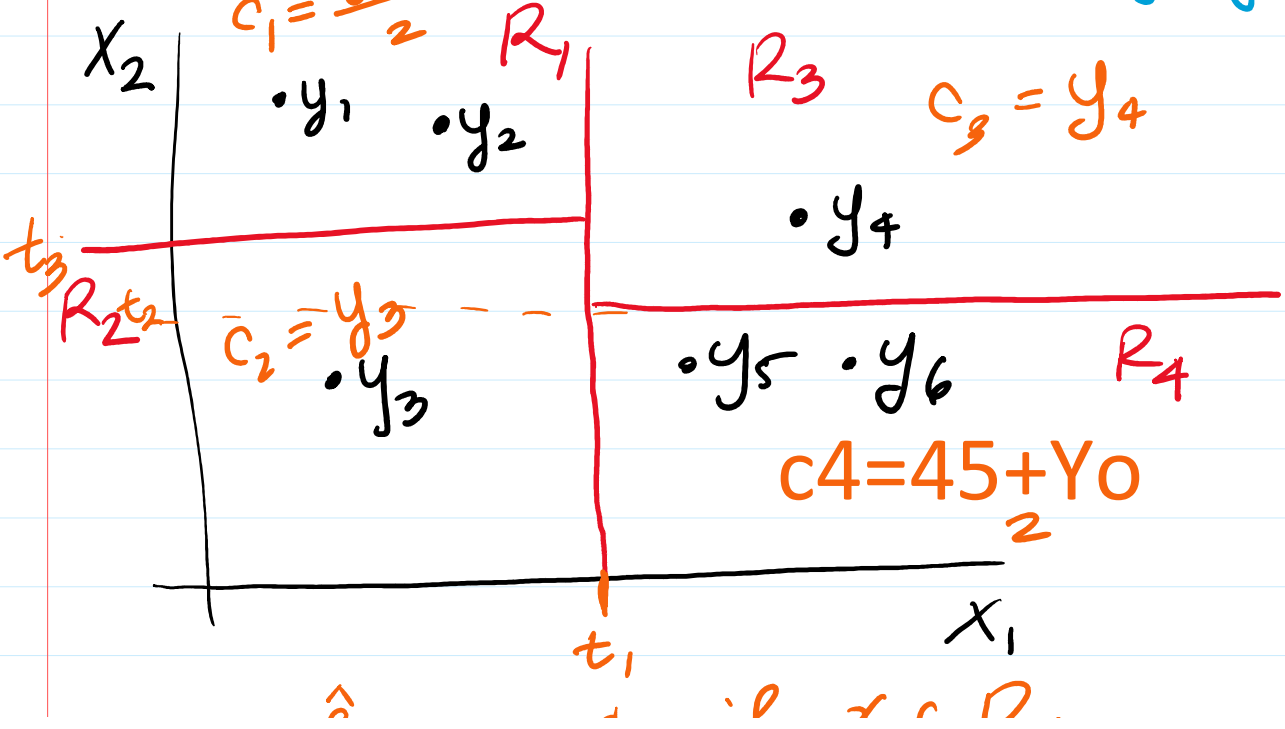
CARTs : Classification and Regression Trees

Regression Trees

Basic idea:

1. break up space of Xs into rectangles
2. on each rectangle, fit some really simple method

(predict y in each rectangle as avg. of training ys)

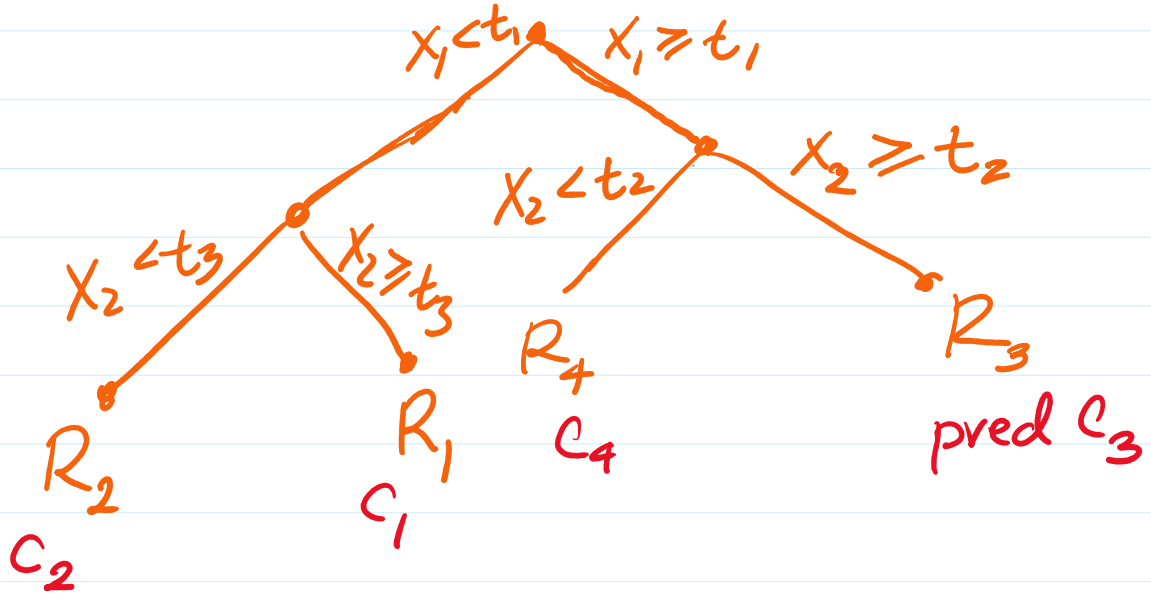


$$\hat{f}(x) = c_i \text{ if } x \in R_i$$

Why called a tree?

Can represent $\hat{f}$ as a decision tree

e.g. (above)



Goal: do this well

Need to decide:

1. which variable to split on (j)
2. where I split the var (t)

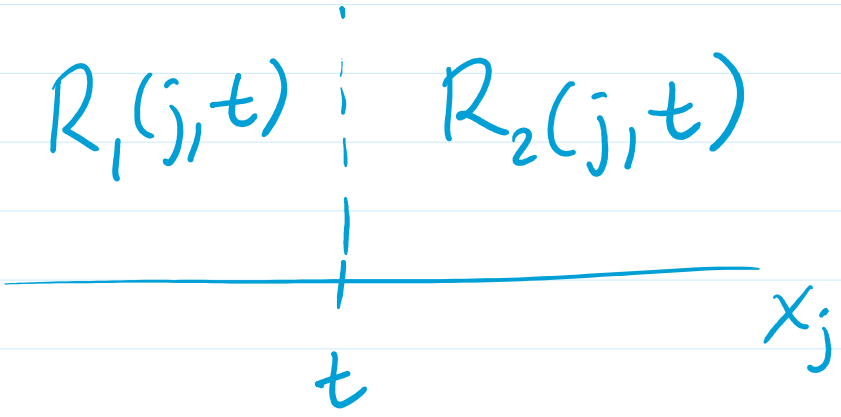
(2) where I split the var (t)

(3) when do I stop.

Optimal: try all possible trees - not computationally tractible

Soln: Use greedy approach

Define $R_1(j,t)$ and $R_2(j,t)$ to be the two half-spaces formed by splitting variable j at point t



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

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

(error I get by splitting var j at point t

$$= \sum_{i \in R_1(j,t)} (y_i - c_1)^2 + \sum_{i \in R_2(j,t)} (y_i - c_2)^2$$

preds in rectangles i.e. means of y_i in each rectangle

Algorithm: fitting tree

1. For each j search over possible t and calc RSS(j,t)
2. Choose j,t to minimize RSS(j,t)
3. Split at var j at point t
4. recursively do this for each new half space.

When do I stop?

→ too many splits → overfit

→ too many splits → overfit

→ too few splits → underfit

(1) set some maximum depth and use something like x-val to set this value

(2) Bad idea: can split until RSS falls below some threshold

problem: bad split may lead to a better split later

Better approach!

(1) grow a really large tree

(2) prune its size (reduce)

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

T T_1

ex.

Cost-Complexity Pruning

penalty strength

penalty

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

size = # of leaf nodes

Want to find tree T that minimizes $C_\alpha(T)$.

Way to do this:

1. grow a really large tree
2. search over sub-trees (i.e. prune parts of tree) to find one that minimizes $C_\alpha(T)$

to find one that minimizes $C_\alpha(T)$

If $\alpha = 0 \Rightarrow$ no pruning (largest possible tree)

$\alpha \to \infty \Rightarrow$ reduce size of tree

Turns out that:

(1) If I prune out parts of tree that increase RSS least I get series of nested sub-trees:

$$T_0 \subset T_1 \subset T_2 \subset \cdots \subset T_N$$

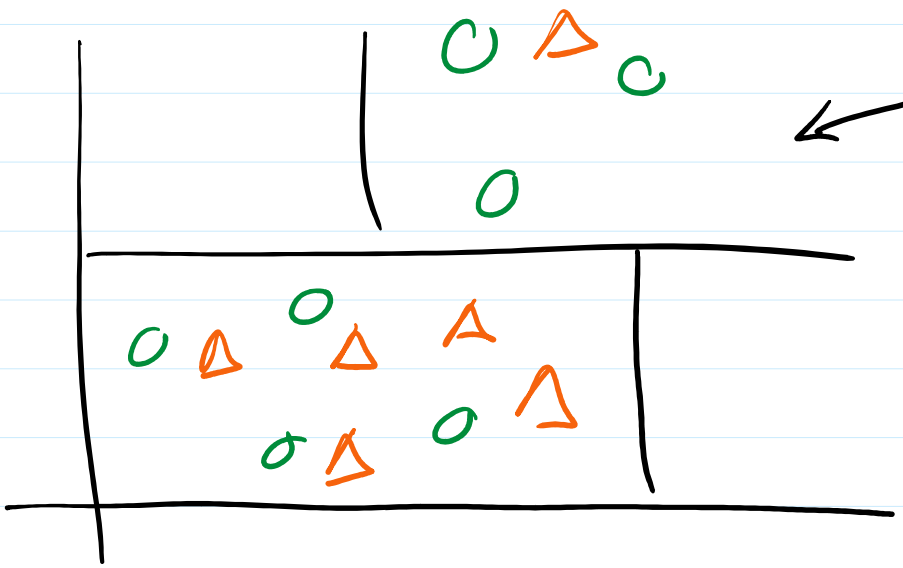
↑ optimal tree that minimizes $C_\alpha(T)$ is somewhere in this sequence

(2) If $\alpha_1 \leq \alpha_2$ then the optimal tree T_{α_1} to $\min C_{\alpha_1}(T)$ contains the tree that $\min_{T_{\alpha_2}} C_{\alpha_2}(T)$.

$$T_{\alpha_1} \supset T_{\alpha_2}$$

$I_{\alpha_1} \cup I_{\alpha_2}$

Classification



predict majority class in each rectangle

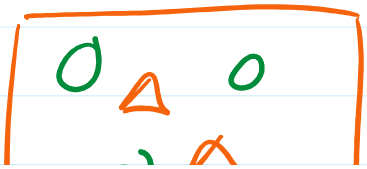
What makes a good split for a classification tree?

Regression ⇝ RSS

Classification ⇝ reduce node/leaf impurity

(increase purity)

Ex. impure



pure