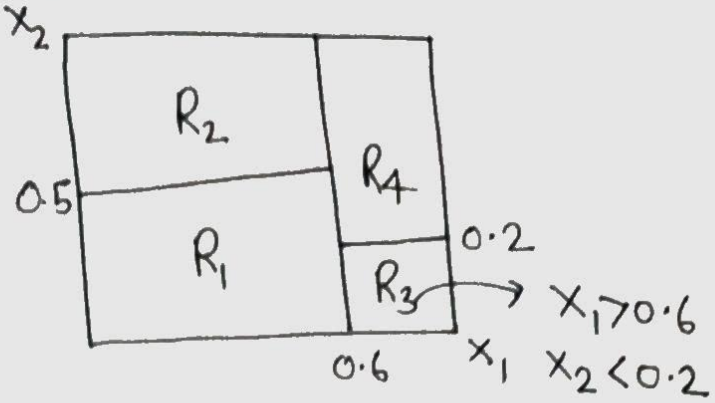
Lecture-33-1 Decision Trees - Introduction

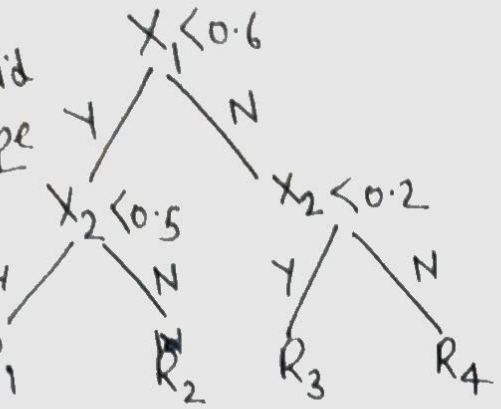
Querying properties of the Data

- Partition I/P space into rectangles.



In theory regions could be of any shape

we use high dimensional rectangular boxes.



- highly Intrepretable
- universal approximator
- Non-parametric

Regression Trees

- Fit a constant in each region.

$$\hat{f}(x) = \sum_{m=1}^{4} c_m I\{(x_1, x_2) \in R_m\}$$

(x_i, y_i) ; $i = 1, \dots N$, $y_i \in \mathbb{R}$ $\forall i$

$x_i = \{x_{i1}, x_{i2}, \dots x_{ip}\}$

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

$$= c_1 I(x \in R_1) + c_2 I(x \in R_2) + c_3 I(x \in R_3) + c_4 I(x \in R_4)$$

- Determine the 'M' regions
- Given regions, find Response. $c_m \rightarrow$ response for region 'm'.

Minimise RSS:

$$= \sum_{j=1}^{M} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

$\hat{y}_{R_m} = \hat{c}_m = \sum_{i \in R_m} f(x_i) \rightarrow$ mean response For training observations

very difficult to divide feature space into J regions.

Approach (greedy)

1. all observations belong to a single region
2. Split feature space, i.e split into two new branches.
3. At each step of tree building process, best split is made.
4. Consider a splitting variable j and cutpoint s, such the splitting the space into regions $R_1: \{X | X_j < s\}$, $R_2: \{X | X_j > s\}$ leads to greatest possible reduction in RSS.

$$R_1(j,s) = \{X | X_j \leq s\}, \quad R_2(j,s) = \{X | X_j > s\}$$

Seek the value of J and S such that

$$\min_{J,S} \left[\sum_{i: x_i \in R_1(J,S)} (y_i - \hat{y}_{R_1})^2 + \sum_{i: x_i \in R_2(J,S)} (y_i - \hat{y}_{R_2})^2 \right]$$

$\hat{y}_{R_1} \rightarrow c_1$ → Mean response

$\hat{y}_{R_2} \rightarrow c_2$ → (Mean response)

↓

we will do seperately.

fix J and find S

$j \in \{1, 2, \cdots p\}$

Strategy to find S → choose values from data points. Sort the points in ascending order and choose 's'.

$n \rightarrow$ Total points. & Tota time: $\boxed{n*p}$ → For one split.

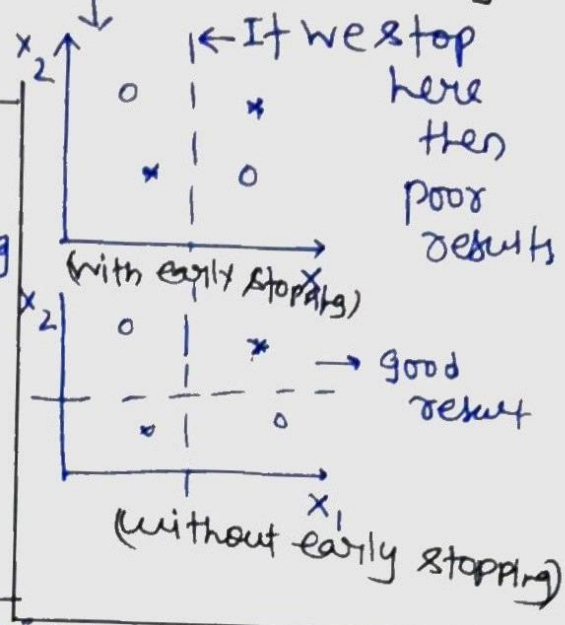
$p \rightarrow J$.

$J \rightarrow$ Can be repeated while growing trees.

Lecture 35 - Stopping criteria & pruning

*Early stopping - May lead into poor decision boundary.

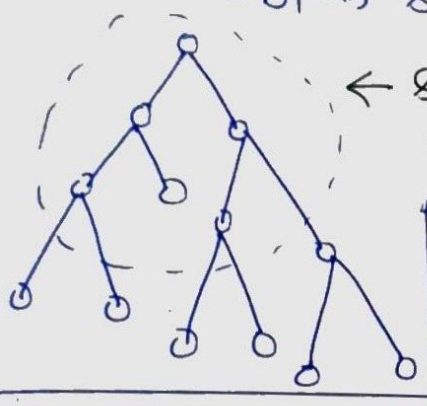
*leaf of a tree is a region



Tree pruning

Above process will lead to overfitting
→ Tree will be very complex.
→ Variance higher
— Simple Tree
less regions → lower variance.
better interpretation

Strategy: Grow Tree T, and the prune it back in order to obtain subtree. → using (Cross-validation)



← seleeting best subtree will be very time consuming → No. of subtrees are very large.

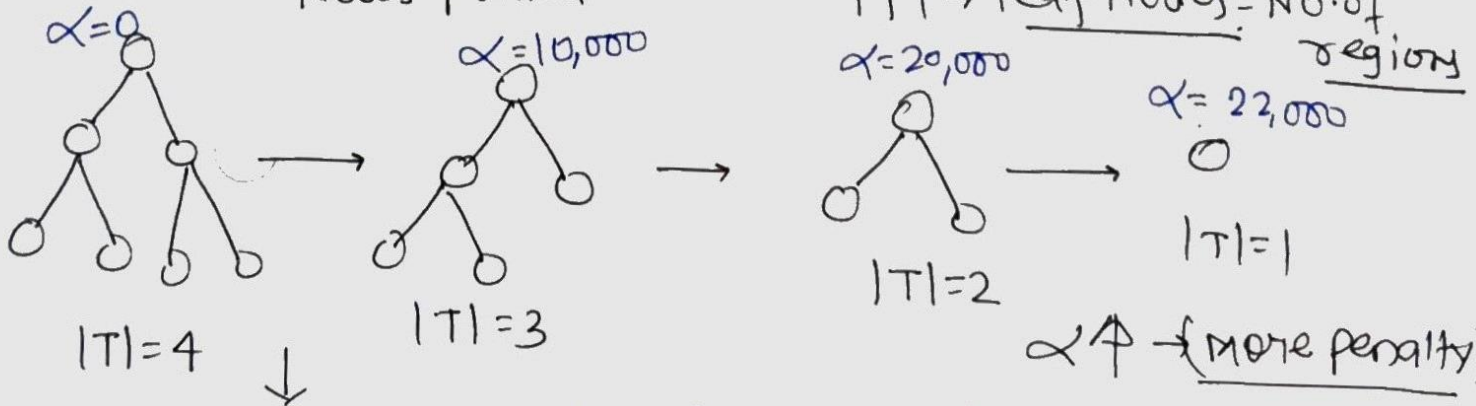
Cost-Complexity pruning: weakes link pruning

$C_\alpha(T)$ = prediction error + $\alpha|T|$

$= \sum_{m=1}^{|T|} (y_i - \hat{y}_{R_m})^2 + \alpha|T|$ ← Complexity penality

$|T|$ → leaf Nodes = No. of regions

As $\alpha \uparrow$ ⇒ Numbere of Nodes pruned.



$\alpha \uparrow$ → (more penalty)

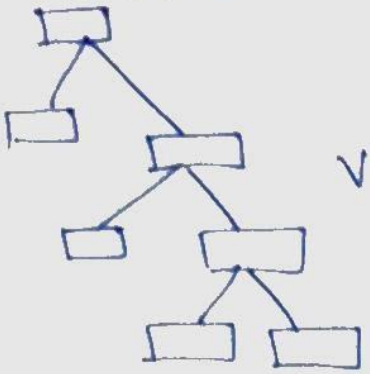
~~we want to penalise tree more having less no. of regions.~~

~~To Compare~~

① we already know that pruned tree will have more error.

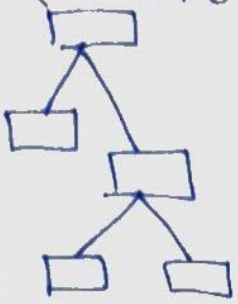
② we need to introduce one more term by which we can compare these trees. by introducing complexity penalty.

SSR = 543·8



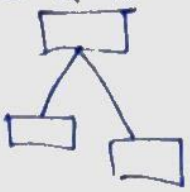
vs

SSR = 5494·8



vs

SSR = 19243



vs

SSR = 288972

Let's take $\alpha = 10{,}000$; Total score = SSR + 10,000*T

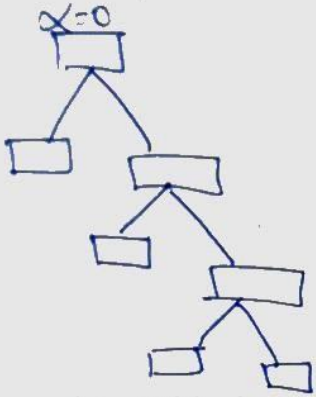
Tree score: 40,543·8 | Tree score = 35,494 | Tree score: 39,243·7 | Tree score: 38,897·2

optimal subtree

using different α can give different result.
we need to find best value for alpha.

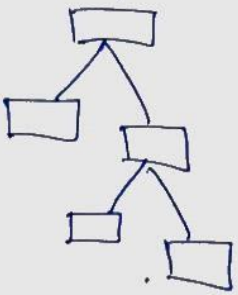
$\alpha = 0$, ⇒ Full sized tree will have lower Tree score

$\alpha = 0$



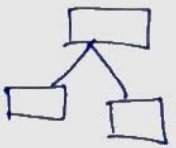
lowest tree score

$\alpha = 10{,}000$



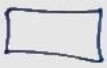
lower tree score

$\alpha = 15{,}000$



(lower tree score)

$\alpha = 22{,}000$



↓

gives lower tree score.

use testing data to get the optimal subtrees for different values of |T|.

Lecture 36

Decision Tree for classification - Loss function.

$$\hat{p}_{mK} = \frac{1}{|R_m|} \sum_{x_i \in R_m} I\{y_i \neq k\}$$

↓ prediction

→ Total Number of ~~Correct predictions~~

↓ Total Number of data points belonging to class k in region 'm'

R_2	R_3
R_1	R_4.

2 classes

$p_{41}, p_{42}, p_{43}, p_{44}, p_{21}, p_{22}, p_{23}, p_{24}$
$p_{11}, p_{12}, p_{13}, p_{14}, p_{31}, p_{32}, p_{33}, p_{44}$

if $x_i \in R_m$ then check which is greater. $[p_{41}, p_{42}, p_{43}, p_{44}]$

$$k(m) = \arg\max_k \hat{p}_{mK} = \max_K [p_{41}, p_{42}, p_{43}, p_{44}]$$

$k(4) =$ $=$

↓ label for region '4'

$$\text{classification error} = \frac{1}{|R_m|} \sum_{x_i \in R_m} I(y_i \neq k(m)) = (1 - \hat{p}_{m(k_m)})$$
$$= (1 - \max_K \hat{p}_{mK})$$

Gini Index $:= \sum_{K=1}^{K} \hat{p}_{mK}(1 - \hat{p}_{mK})$ – Measure of variance across K-classes for a particular region.

Cross entropy. $- \sum_{K=1}^{K} \hat{p}_{mK} \log \hat{p}_{mK}$ – (information gain criteria)

↓ estimated

└ label

Lecture-37.

Decision Trees - Categorical attributes

q- unordered attributes:

Divide into 2 groups. Total No. of grps possible:

$\left(^{q}C_1 + {}^{q}C_2 + {}^{q}C_3 + \cdots {}^{q}C_{q/2}\right) = 2^{q-1} - 1$ → Not feasible to go and pick split points.

in case of regression: you have to look in' points.

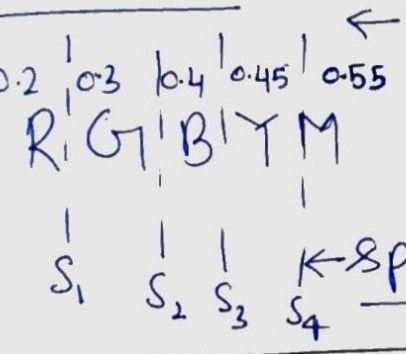
n → No. of data points.

0|1 - binary classification

Suppose q=5 For color attribute:

q=0 Red,
q=1 green
q=3 blue
q=4 yellow
q=5 Magenta.

P(class1| Red), P(class1|green),
P(class1| blue), P(class1|yellow),
P(class1| magenta) → Ascending order. and then make the split.



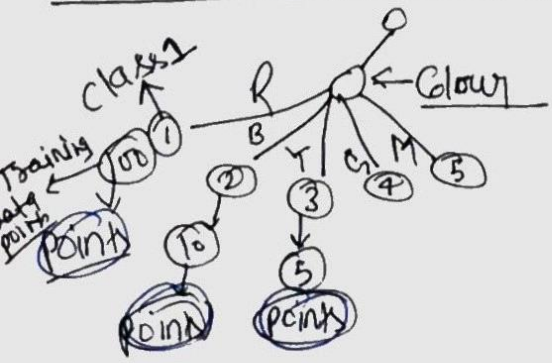
No - Need to consider $(2^{q-1} - 1)$ splits.

Lecture-38 Multiway splits.

Multiclass

Heuristics: -

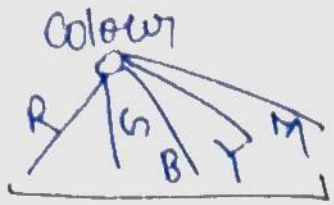
Multiway-splits

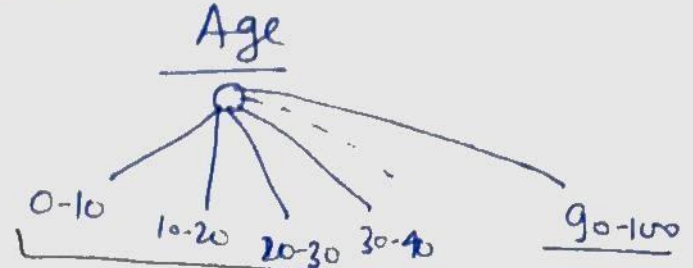


- Problem
  - → Sprawling (spreading) (hard to interpret)
  - → lost interpretality
  - → Variance more.
  - → At each split, data points will be lower..
  - → Tree will become sparse.
  - → overfitting is the problem.

Lecture-38

- Favours attributs with more values.





Error Color > Error Age. Less error in splitting.

C4.5 Algorithm - gain-Ratio

↓

what is the information gain after splitting into 10 parts rather then 6.

→ We can also use binary split to arrive at same → function which is precicted by multiway splits.

↓ binary split is always better > multiway split ~~because~~

↓

we can use recursive approach to make multi-way split.

Multiway splits → avoids the choosing variable (S) (split Point)