for classification and regression

→ the tree will be learned from data

→ the prediction will be taken in the leaf

→ Rules:

① every rout to leafs

② every rout to positive leafs

→ no need for one-hot-encoding

→ no need for feature normalization

→ can overfitting easily (especially with high # of features)

→ for multiclass ⇒ no need for any change

non parameters (like Knn)

non linear classifier

** if features are categorical:

1- no need for one-hot-encoding

2- maximum deep = # of features

** if features are continous

→ convert them to binary with thresholds

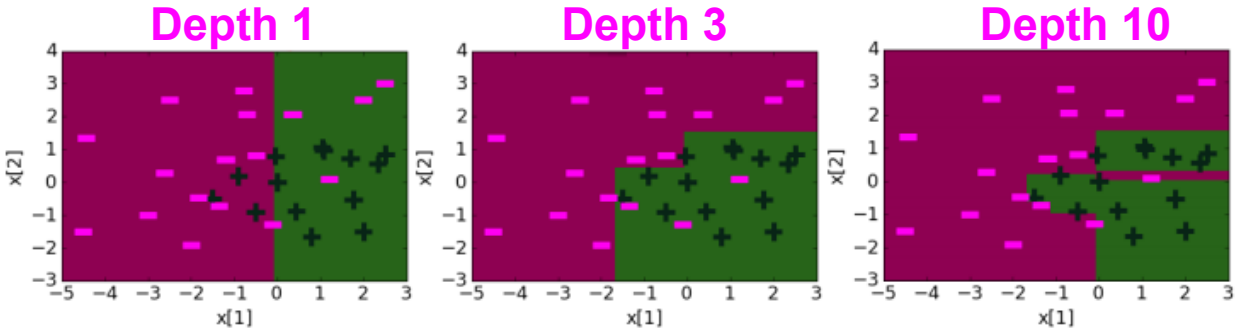
→ what threshold? the one with less error

→ you can use the same feature again with another threshold → tend to overfitting

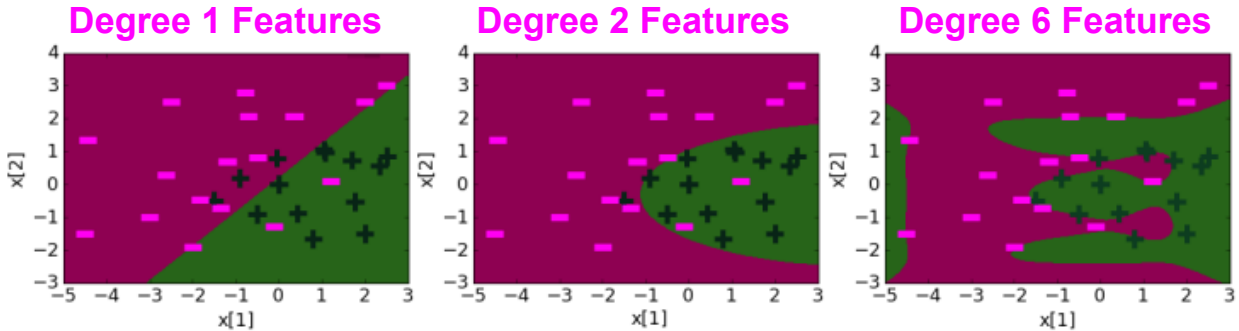
→ no need for normalization

Decision Trees Versus Logistic Regression (6)

Decision Tree decision boundries are axis align



Logistic Regression not axis algin



How to generate a tree??

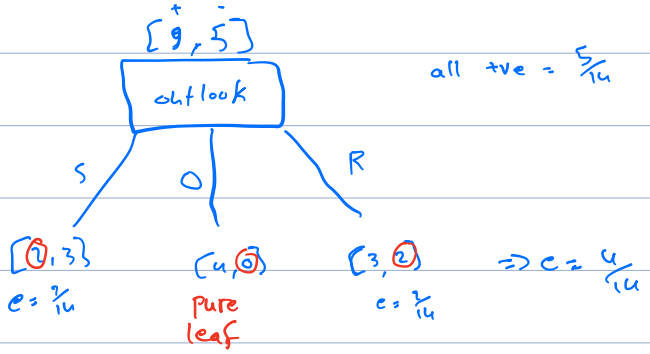
① trail and error ⇒ not fesible

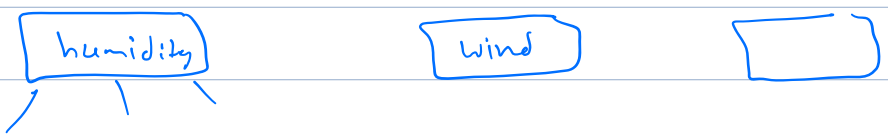
② heuristic

select the best (forward selection)

what is the best?? error based

slide 15





then select the least error rate ⇒ then go more deep unless it's pure

When to stop?-

① pure nodes

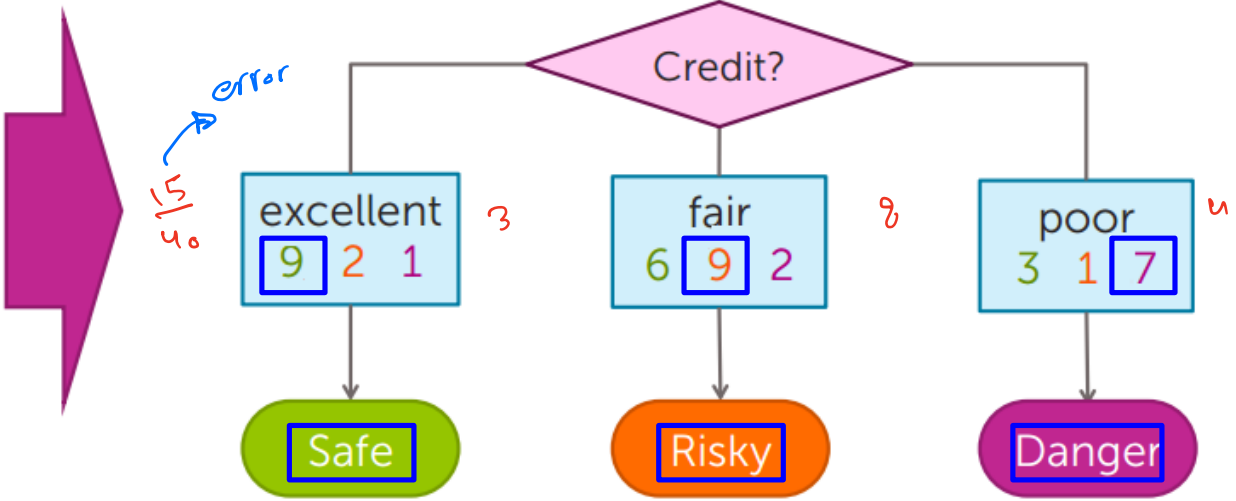
② we already split on all features (if they are categorical)

③ set the depth of tree hyperparameter

Multiclass Decision Stump

N = 40,
1 feature,
3 classes

Credit	y
excellent	safe
fair	risky
fair	safe
poor	danger
excellent	risky
fair	safe
poor	danger
poor	safe
fair	safe
...	...



# of leaf node is an indicator of complixity

How to avoid overfitting ??

1 early stopping

① if you split and the error doesn't improve => stop

② stop if you have small # (hyperparameter) of samples that you split in the node

③ early stopping at certine depth (hyperparameter) (Longest path)

② pruning => cutting from bottom until the J start increasing

J = error rate + λ (hyperparameter) model complexity (# of leaf nodes)

λ = ∞ decision in root (under fit)

λ = 0 standerd (overfit)

Simpler trees are better!!!

more chance to generlaize better

in Regresrion:-

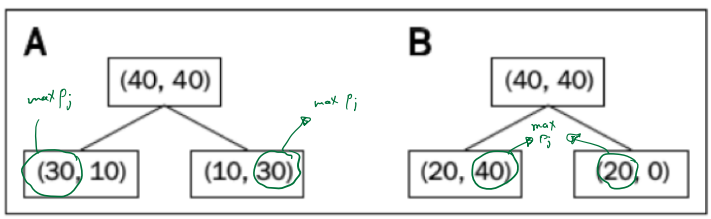
calculate SRR

Different splitting measures

① classification error $= 1 - \max_j p_j$ — what we used above; p_j: probability of class j

② Entropy $= -\sum_j p_j \log_2 p_j$

③ Gini $= 1 - \sum_j p_j^2$



in error ⇒ they are same $e = \frac{20}{80}$

but B is better since has pure

⇒ So error rate is not good enough

General Formula:-

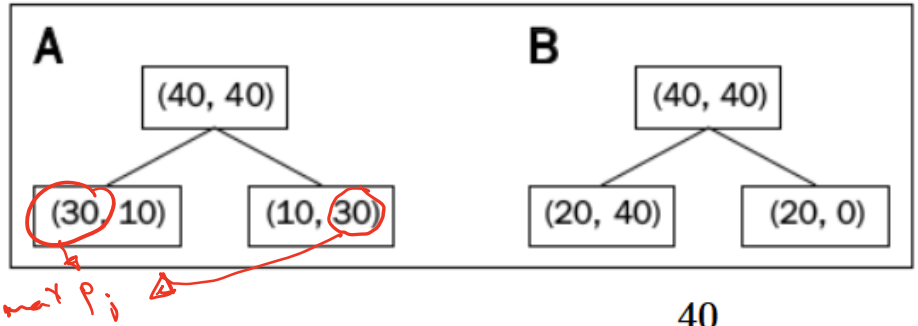
Information Gain — any impurity measure (error, entropy, Gini)

$$IG(D_p) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right})$$

D_p: dataset of the parent

⇒ more gain is better

Example–IG with Classification Error



$$error = 1 - \max p_j$$

$$I_E(D_p) = 1 - \frac{40}{80} = 1 - 0.5 = 0.5$$

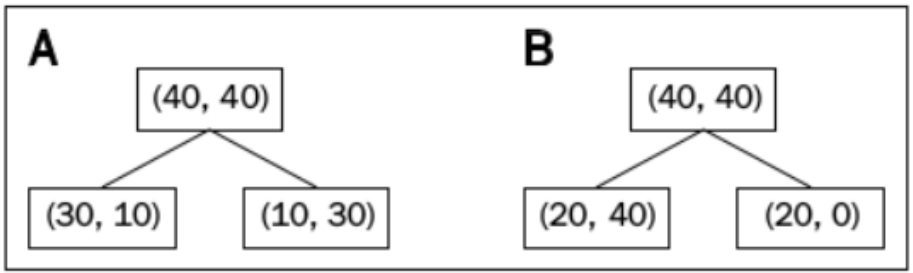
$$A: I_E(D_{left}) = 1 - \frac{30}{40} = 1 - \frac{3}{4} = 0.25$$

$$A: I_E(D_{right}) = 1 - \frac{30}{40} = 1 - \frac{3}{4} = 0.25$$

$$IG(D_p) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right})$$

$$A: IG_E = 0.5 - \frac{40}{80} \times 0.25 - \frac{40}{80} \times 0.25 = 0.5 - 0.125 - 0.125 = 0.25$$

Example–IG with Classification Error



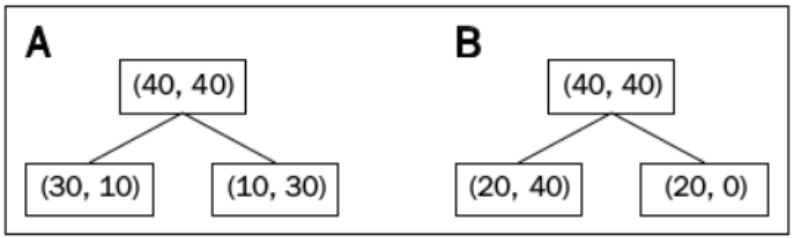
$$error = 1 - \max p_j$$

$$B : I_E(D_{left}) = 1 - \frac{40}{60} = 1 - \frac{2}{3} = \frac{1}{3}$$

$$B : I_E(D_{right}) = 1 - \frac{20}{20} = 1 - 1 = 0$$

$$B : IG_E = 0.5 - \frac{60}{80} \times \frac{1}{3} - \frac{20}{80} \times 0 = 0.5 - 0.25 - 0 = 0.25$$

Example–IG with Entropy



$$Entropy = -\sum_j p_j log_2 p_j$$

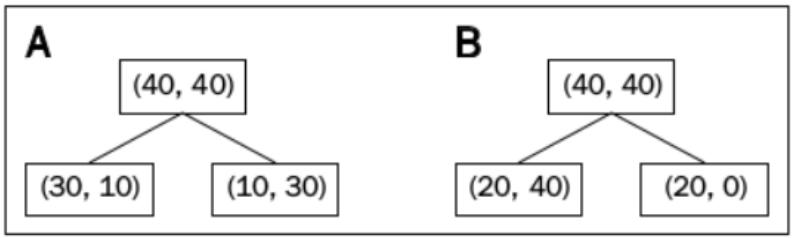
$$I_H(D_p) = -\left(0.5\log_2(0.5) + 0.5\log_2(0.5)\right) = 1$$

$$A: I_H(D_{left}) = -\left(\frac{30}{40}\log_2\left(\frac{30}{40}\right) + \frac{10}{40}\log_2\left(\frac{10}{40}\right)\right) = 0.81$$

$$A: I_H(D_{right}) = -\left(\frac{10}{40}\log_2\left(\frac{10}{40}\right) + \frac{30}{40}\log_2\left(\frac{30}{40}\right)\right) = 0.81$$

$$A: IG_H = 1 - \frac{40}{80} \times 0.81 - \frac{40}{80} \times 0.81 = 0.19$$

Example–IG with Entropy



$$Entropy = -\sum_j p_j log_2 p_j$$

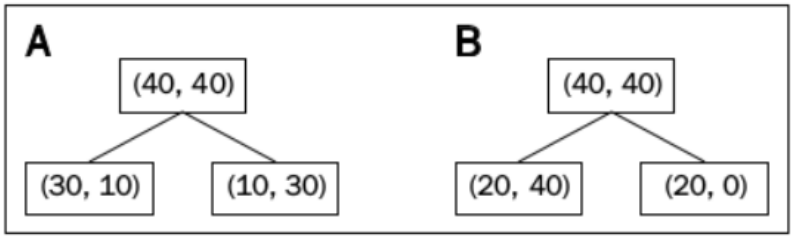
$$B : I_H(D_{left}) = -\left(\frac{20}{60}\log_2\left(\frac{20}{60}\right) + \frac{40}{60}\log_2\left(\frac{40}{60}\right)\right) = 0.92$$

$$B : I_H(D_{right}) = -\left(\frac{20}{20}\log_2\left(\frac{20}{20}\right) + 0\right) = 0$$

$$B : IG_H = 1 - \frac{60}{80} \times 0.92 - \frac{20}{80} \times 0 = 0.31$$

more gain → so it's better

Example–IG with Gini index



$$Gini = 1 - \sum_j p_j^2$$

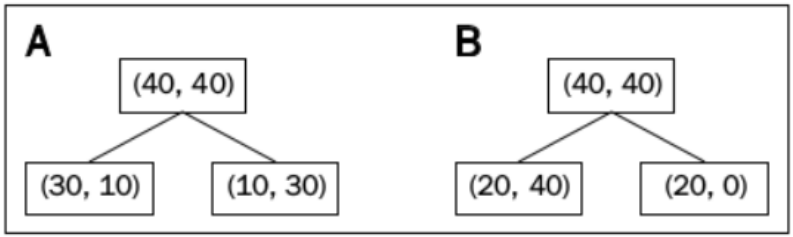
$$I_G(D_p) = 1 - \left(\left(\frac{40}{80}\right)^2 + \left(\frac{40}{80}\right)^2\right) = 1 - (0.5^2 + 0.5^2) = 0.5$$

$$A: I_G(D_{left}) = 1 - \left(\left(\frac{30}{40}\right)^2 + \left(\frac{10}{40}\right)^2\right) = 1 - \left(\frac{9}{16} + \frac{1}{16}\right) = \frac{3}{8} = 0.375$$

$$A: I_G(D_{right}) = 1 - \left(\left(\frac{10}{40}\right)^2 + \left(\frac{30}{40}\right)^2\right) = 1 - \left(\frac{1}{16} + \frac{9}{16}\right) = \frac{3}{8} = 0.375$$

$$A: I_G = 0.5 - \frac{40}{80} \times 0.375 - \frac{40}{80} \times 0.375 = 0.125$$

Example–IG with Gini index



$$Gini = 1 - \sum_j p_j^2$$

$$B : I_G(D_{left}) = 1 - \left(\left(\frac{20}{60} \right)^2 + \left(\frac{40}{60} \right)^2 \right) = 1 - \left(\frac{9}{16} + \frac{1}{16} \right) = 1 - \frac{5}{9} = 0.44$$

$$B : I_G(D_{right}) = 1 - \left(\left(\frac{20}{20} \right)^2 + \left(\frac{0}{20} \right)^2 \right) = 1 - (1 + 0) = 1 - 1 = 0$$

$$B : I_G = 0.5 - \frac{60}{80} \times 0.44 - 0 = 0.5 - 0.33 = 0.17$$

it better