DMML, 16 Jan 2020

Decision Trees

Items with attributes

$(a_1, \ldots, a_n)$ & class c

Training data → Model

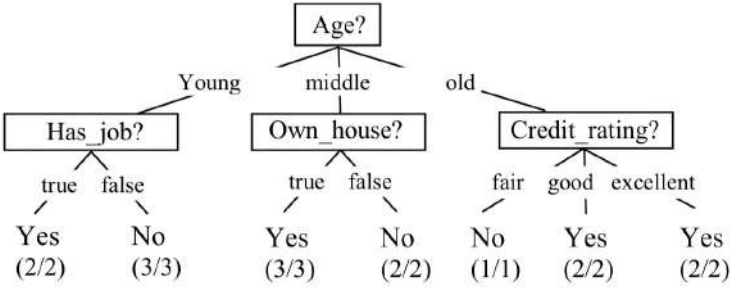
Adaptively ask question

Prefer small trees

Computationally hard

Greedy heuristic

ID	Age	Has_job	Own_house	Credit_rating	Class
1	young	false	false	fair	**No**
2	young	false	false	good	**No**
3	young	true	false	good	**Yes**
4	young	true	true	fair	**Yes**
5	young	false	false	fair	**No**
6	middle	false	false	fair	**No**
7	middle	false	false	good	**No**
8	middle	true	true	good	**Yes**
9	middle	false	true	excellent	**Yes**
10	middle	false	true	excellent	**Yes**
11	old	false	true	excellent	**Yes**
12	old	false	true	good	**Yes**
13	old	true	false	good	**Yes**
14	old	true	false	excellent	**Yes**
15	old	false	false	fair	**No**



Greedy heuristic

Reduce "impurity" as much as possible

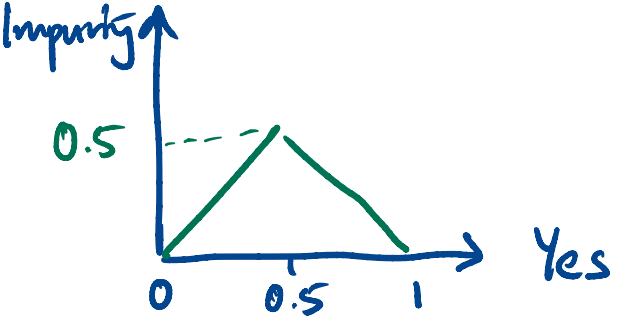
"Pure" = all items in the set have same class

Impure set – verdict is majority

Minority case = error = misclassification

Impurity = Misclassification Rate

Max impurity = 0.5

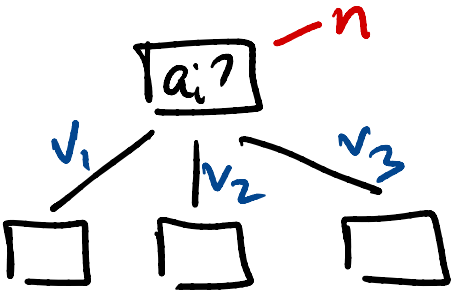


Tree building algorithm

Available attributes $(a_1, \ldots, a_k)$

Evaluate current misclassificate rate

For each a_i



$a_i \leq \begin{matrix} v_1 - n_1 \\ v_2 - n_2 \\ v_3 - n_3 \end{matrix}$ n

Impurity i_1 i_2 i_3

n_1 n_2 n_3

Wted avg $\frac{n_1}{n} \cdot i_1 + \frac{n_2}{n} \cdot i_2 + \frac{n_3}{n} \cdot i_3$

Compute weighted avg of impurity for each a_i

Among these, choose the best one $\rightarrow a_j$

For each child node, apply the same algorithm with $(a_1, \ldots, a_{j-1}, a_{j+1}, \ldots a_k)$ as available attributes

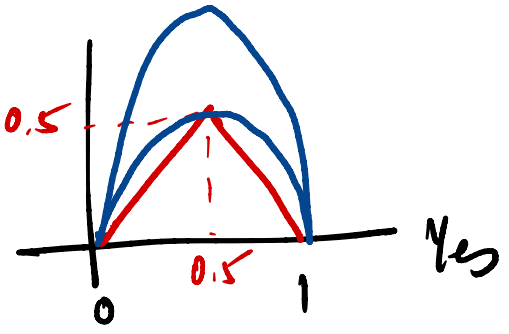
Stopping?

1. No more available attributes
2. Node is pure

1. Impurity = Error = "Cost"

Incrementally reducing cost by refining the model

2. Is there a better notion of impurity?



Better to have a measure that penalizes impurity non-linearly

Information Theory [Shannon]

Message alphabet → Encoding → Transmission Alphabet

$\{a, b, \ldots, y, z\}$ $\qquad$ $\{0, 1\}$

26 $\qquad$ $2^4 < 26 \leq 2^5$

Can we use frequency information to improve this?

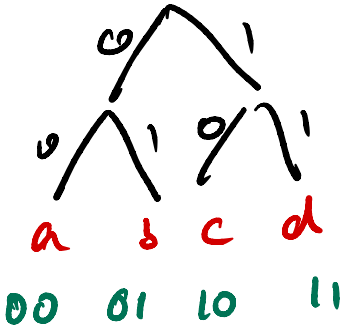
Interpret frequencies as probabilities

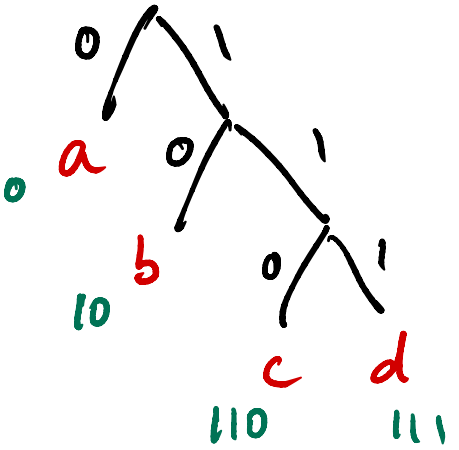
Example $\{a, b, c, d\}$ $\xrightarrow[\{0,1\}]{\text{encode}}$ 2 bits

Message length N $\longrightarrow$ 2N

a – 1/2
b – 1/4
c – 1/8
d – 1/8

Variable length encoding – shorter codes for frequent letters





200 letters input → Naive → 400 bits

100 a × 1 + 50 b × 2 + 25 c × 3 + 25 d × 3

100 + 100 + 75 + 75 = 350

Huffman Coding

Shannon - entropy $-\sum p_i \log_2 p_i$

p_i = probability of ith letter

$\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}$

$$-\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{4}\log\frac{1}{4} + \frac{1}{8}\log\frac{1}{8} + \frac{1}{8}\log\frac{1}{8}\right)$$

$$\overset{-1}{\log\frac{1}{2}} \quad \overset{-2}{\log\frac{1}{4}} \quad \overset{-3}{\log\frac{1}{8}} \quad \overset{-3}{\log\frac{1}{8}}$$

$$-\frac{1}{2} + -\frac{1}{2} + -\frac{3}{8} + -\frac{3}{8}$$

$$-\left(-1 + \left(-\frac{3}{4}\right)\right)$$

$$-\left(-1.75\right)$$

$$= 1.75 \implies$$ avg length per letter is 1.75 bits (lower bd

200 length input $\times$ 1.75 = 350 chars

Borrow entropy as a measure of impurity

N cases $\begin{cases} M \text{ Yes} & - P_Y = \frac{M}{N} \\ N-M \text{ No} & - P_N = \frac{N-M}{N} \end{cases}$

$$-\left(P_Y \log P_Y + P_N \log P_N\right)$$

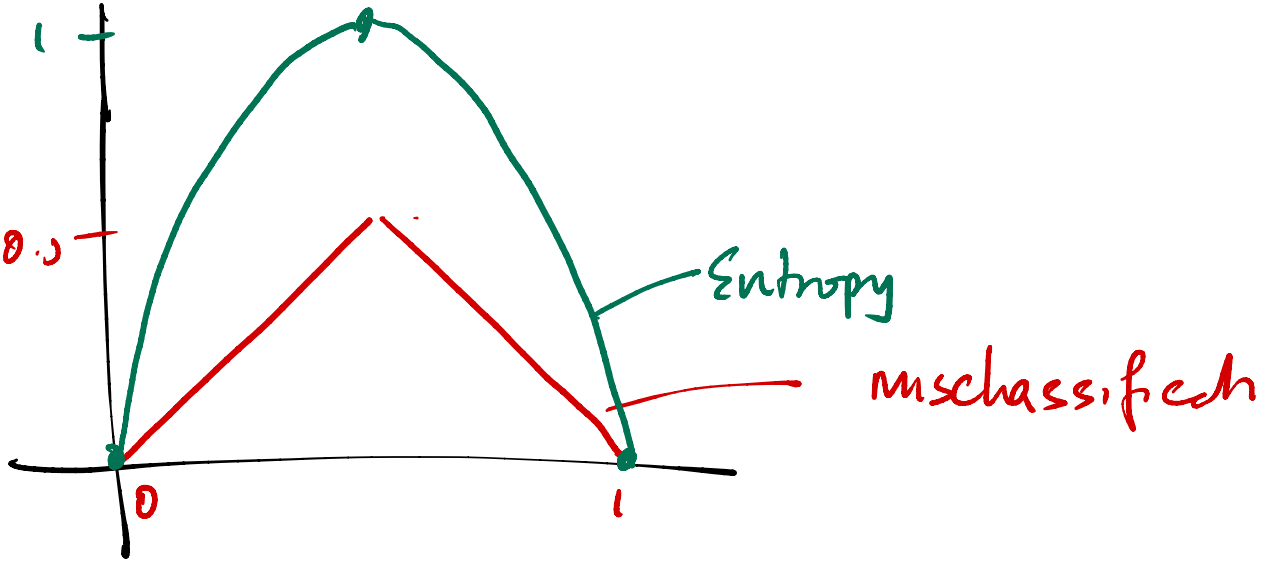
$P_Y = 1, P_N = 0$ $\quad -(1 \underbrace{\log 1}_{0} + 0 \underbrace{\log 0}_{?}) = 0$

$P_Y = 0, P_N = 1$

For entropy $0 \log 0$ defined $= 0$

$P_y = P_N = \frac{1}{2}$ $\quad -\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right)$

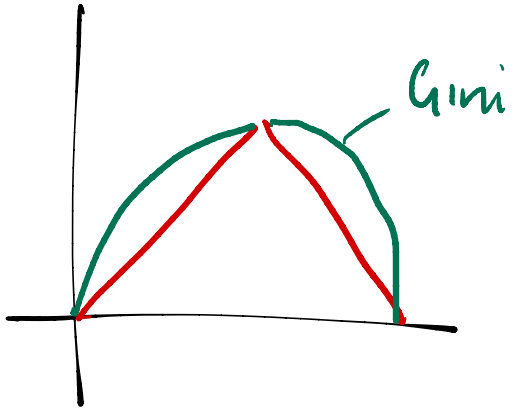
$= \quad -\left(-\frac{1}{2} + -\frac{1}{2}\right) \quad = +1$



Another option

Economics - Equality of distribution of wealth/resources

$$1 - \sum p_i^2$$



$P_Y = 1 \quad P_N = 0 \qquad 0$

$P_N = 1 \quad P_Y = 1 \qquad 0$

$P_Y = P_N = \frac{1}{2} \qquad 1 - \left(\frac{1}{4} + \frac{1}{4}\right) = \frac{1}{2}$

$\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$

Entropy $- \left(\frac{1}{3} \log \frac{1}{3}\right) \cdot 3$

$\log_2 3$

Gini $1 - \sum \left(\frac{1}{3}\right)^2 = \frac{2}{3}$

Most implementations use Gini index

Any such function should do

Shannon Information is inverse of entropy

For us entropy = impurity

Reduce impurity = Gain information