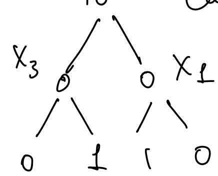
Decision Tree: is a boolean function.

Outputs $+1$ or -1.

0 or 1.

Input : $x \in \{0,1\}^n$ $\quad f(x) \rightarrow \{0,1\}$ ← some function.

x_4 ← each node has a literal, say, x_4. If $x_4 = 0$, move left in the tree. If $x_4 = 1$, move right.



If next node = literal, repeat.

If next node = leaf, output the boolean.

size of decision tree = # of nodes.

depth / height = length of longest path from root to leaf.

Given a set of labeled examples, build a tree with low error.

S = training set $x^1, y^1,$

$\vdots$

x^m, y^m , m = # of training samples.

$y^i \in \{0,1\}$

$x^i \in \{0,1\}^n$, n = # of features.

- Error rate, training error, empirical error rate.

Fix T. (T = decision tree).

$$\text{Error Rate} = \frac{\#\text{ of mistakes } T \text{ makes on } S}{|S|} \quad \leftarrow \text{size of } S.$$

Natural Approach for building decision trees, given a set S:

- Assume: tree is a leaf.

Always $+1$ or always 0.

⇒ If tree can output only 0 or 1, then should output the majority of the labels. If majority $= 0 \rightarrow$ output 0.

- Assume a more interesting tree:

$o \to X$ (root)

How to decide on what is at the root?

Define a potential function $\phi(a)$: this function determines what criterion we use to put at the root of the tree.

Define a potential function $\phi(a)$

$$\phi(a) = \min(a, 1-a).$$

Pick a literal x_i.

Compute $\phi(\Pr_{(x,y)\sim S}(y=0))$

1) Assume 10 (+) examples and 5 (−) examples.

then $\phi(\Pr_{(x,y)\sim S}(y=0)) = \frac{5}{5+10} = \frac{1}{3} \to \phi(\frac{1}{3}) = \min(\frac{1}{3}, \frac{2}{3}) = \frac{1}{3}$

2) Assume 5 (+) and 10 (−):

$\phi(\Pr(y=0)) = \phi(\frac{2}{3}) = \min(\frac{1}{3}, \frac{2}{3}) = \frac{1}{3}$

Error rate for tree with just 1 leaf.

$\phi(\Pr_{(x,y)\sim S}(y=0))$ ← error rate of the trivial decision tree.

Pick literal x_1

x_1

What label? condition on $x_1 = 0$ output majority value.

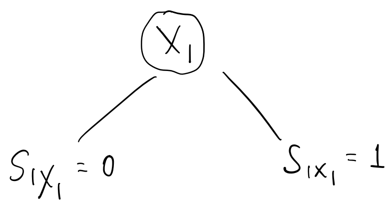
condition on $x_1 = 1$, output majority value.

What is the new error rate?

$$\left.\begin{array}{l} \Pr\limits_{x,y \sim S}[X_1 = 0] \cdot \phi(\Pr(y=0 \mid X_1 = 0)) + \\ \qquad x,y \text{ drawn from } S. \\ \Pr\limits_{x,y \sim S}[X_1 = 1] \cdot \phi(\Pr(y=0 \mid X_1 = 1)) \end{array}\right\} \text{new error rate.}$$

$$\boxed{\text{Gain}(X_1) = \text{Old Rate} - \text{New Rate using } X_1}$$

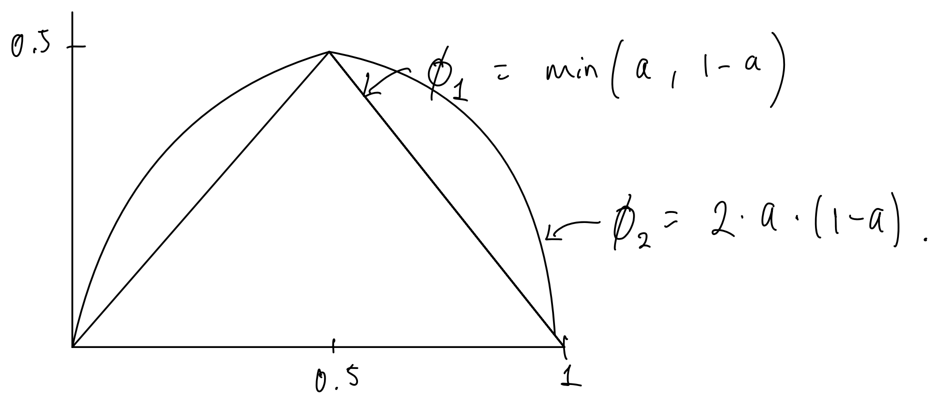
Go thru each literal $X_1 \cdots X_n$, and see which literal maximizes the gain, and put that in the root.



★ Structure of tree is determined by choice of ϕ:

$\phi(a) = \min(a, 1-a)$ corresponded to training error.

$\phi(a) = 2 \cdot a \cdot (1-a)$ corresponds to the "Gini function"



since ϕ_2 is upper bound on $\phi_1 \Rightarrow$
small values of $\phi_2 \Rightarrow$ small values of ϕ_1 ← training error.

ϕ_2 has nicer mathematical properties, is easier to work with; it is smooth.

Example:

$S =$

X_1	X_2	Pos	Neg	
0	0	1	1	2
0	1	2	1	3
1	0	3	1	4
1	1	4	2	6
		10	5	15 $\Rightarrow$ (majority = positive).

$\phi(a) = 2 \cdot a \cdot (1-a)$

$\phi(\Pr_{x,y\sim S}(y=0))$

when $x_1=0$, $x_2=0$, we have 1 (+) example and 1 (-) example in our training set.

what is the phi value of the trivial decision tree? $2\left(\frac{1}{3}\right)\left(\frac{2}{3}\right) = \boxed{4/9}$

$p(\text{negative}) = \frac{5}{15} = 1/3.$

$\phi(a) = 2 \cdot a \cdot (1-a)$

does not matter if use positive or negative.

$= 2\left(\frac{1}{3}\right)\left(\frac{2}{3}\right) = \boxed{4/9}$

pick X_1 or X_2 to be at the root?

Look @ X_1:

$$\underbrace{\Pr(X_1=0)}_{\frac{2+3}{15}=\frac{1}{3}}\ \underbrace{\phi(\Pr(\text{neg}\mid X_1=0))}_{\Pr = 2/5} + \underbrace{\Pr(X_1=1)}_{\frac{4+6}{15}=\frac{2}{3}} \cdot \underbrace{\phi(\Pr(\text{neg}\mid X_1=1))}_{\Pr = \frac{1+2}{10}=\frac{3}{10}}$$

$\phi = 2\cdot\frac{2}{5}\cdot\frac{3}{5}$

$=$

$\phi = 2\cdot\frac{3}{10}\cdot\frac{7}{10}$

$=$

$\Rightarrow \boxed{\frac{11}{25}}$ which is smaller than $4/9$ $\Rightarrow$ made progress!

Now do the same for X_2:

$$p(X_2=0)\cdot\phi\Big(\underbrace{Pr(neg \mid X_2=0)}_{2/6}\Big) + P(\underbrace{X_2=1}_{\frac{3+6}{15}=\frac{3}{5}})\cdot\phi\Big(\underbrace{Pr(neg \mid X_1=1)}_{3/9}\Big)$$

$$\frac{2+4}{15} = \frac{2}{5}$$

$$\Rightarrow \frac{2}{5}\cdot\left(2\cdot\frac{1}{3}\cdot\frac{2}{3}\right) + \frac{3}{5}\cdot\underbrace{\phi\left(\frac{3}{9}\right)}_{4/9} = \boxed{\frac{4}{9}}$$

$$\text{Gain}(X_1) = \left(\frac{4}{9} - \frac{11}{25}\right) > 0$$

$$\text{Gain}(X_2) = \left(\frac{4}{9} - \frac{4}{9}\right) = 0$$

$\Rightarrow$ $\boxed{\text{Pick } X_1}$

If more than X_1 and X_2 $\Rightarrow$ recursive program to pick the best literal to be at the root.

One question is: when should we stop?

- one answer: stop when the gain is extremely small for all literals.
- Pruning: build an enormous tree, and have a parameter indicating how many nodes you want. Avenue of research.
- Random Forest: to build many small decision trees and then take a majority vote of the resulting trees.

$\rightarrow$ Algorithm for building many trees:

1) take training set S, randomly subsample from S to create S'

2) Randomly pick some features from $\{X_1, \cdots, X_n\}$ of size k.

Build a decision tree using S' and the K random features.

Take majority vote.

Can sample with or without replacement.

Read ch 18 of textbook.