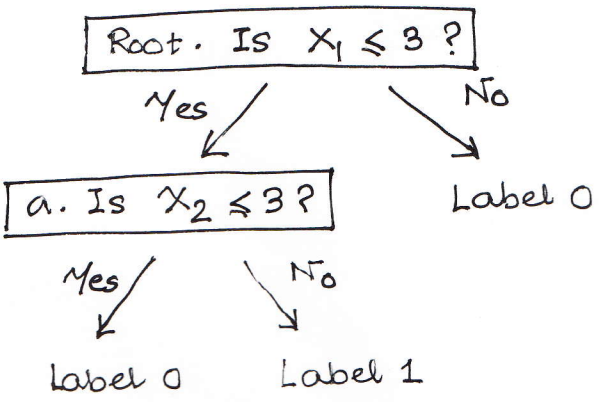
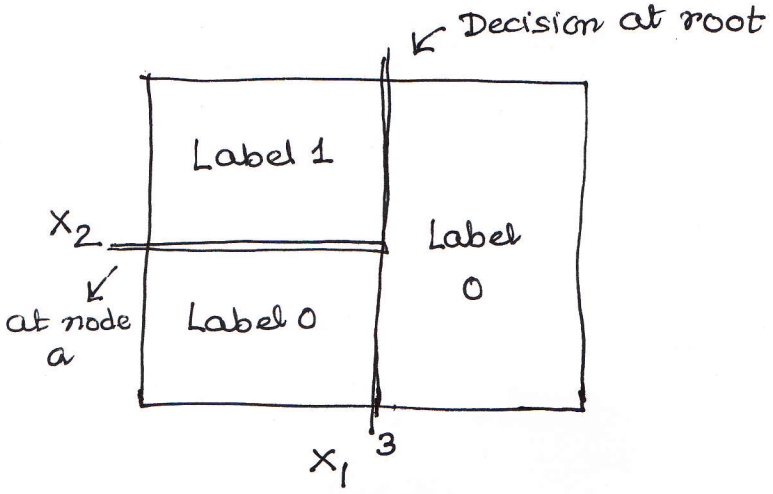
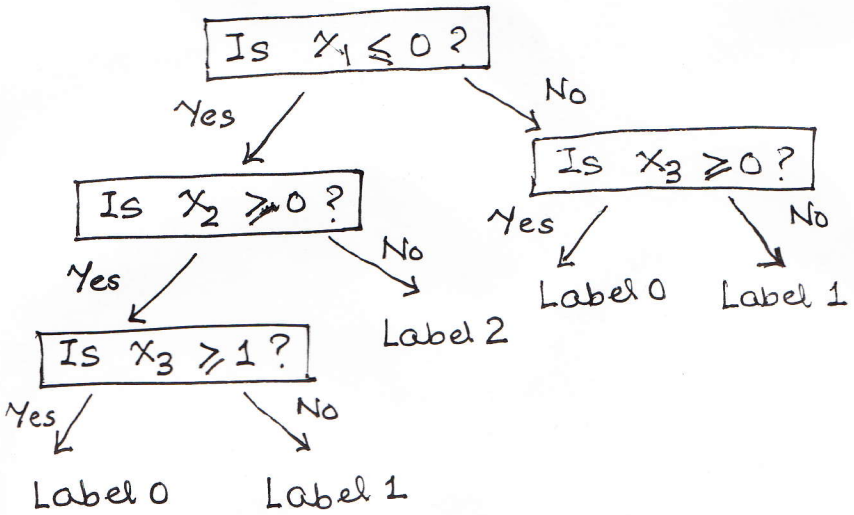
Lecture 4: Decision Trees

Example 1: Features X_1, X_2



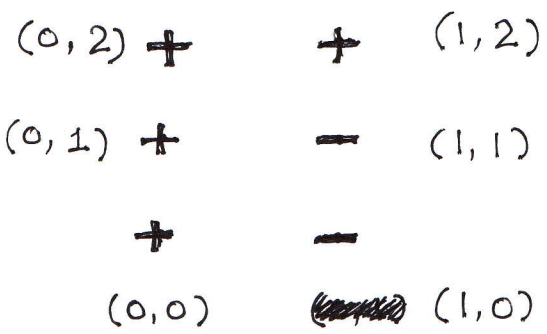


Example 2: Features X_1, X_2, X_3

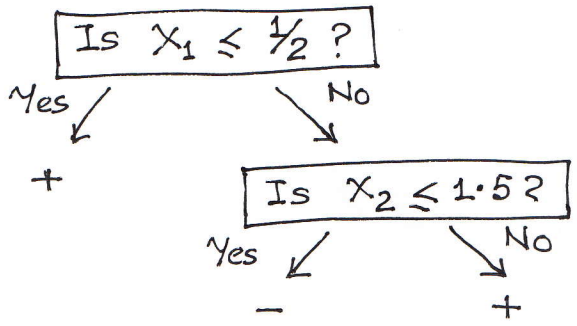


- Each node is based on a feature
- Each node: a decision is made based on the value of this feature
- Each leaf corresponds to a label (same label may appear on many leaves)

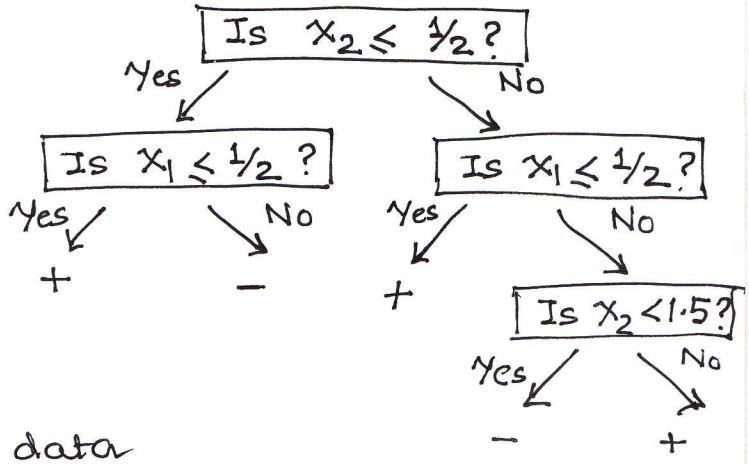
Example 3: Dataset:



Tree 1:



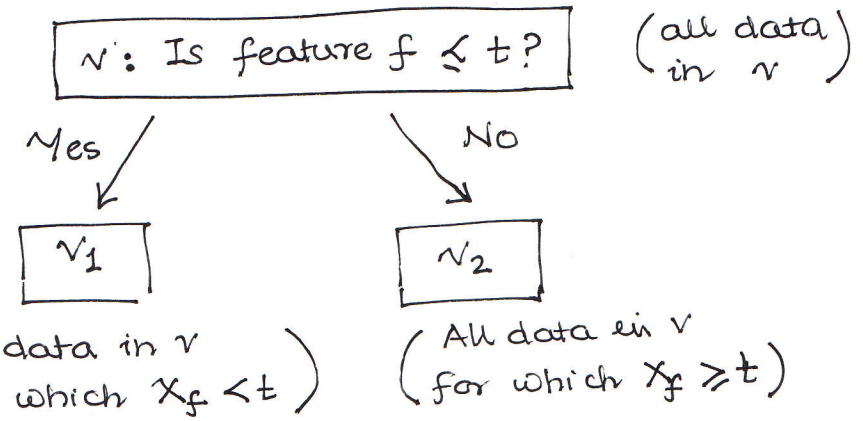
Tree 2:



- Both trees classify the training data correctly; but which one is better?
- Answer: The simpler one. (here, tree 1)
- In general, finding the smallest decision tree to fit a training data set is computationally difficult.
- A commonly used algorithm that proceeds greedily is the ID3 Decision Tree algorithm.
- Convention: Rule at each node of the form:
  "Is $X_f \leq t$?"

ID3 Algorithm:

1. Initially, Whole training data is at root.
2. While there is an impure node:
   (a) Pick any impure node v
   (b) Pick a feature f and threshold t along which to "split" the data at v. Done according to "Splitting Rule" (to be described later)
   (c) Modify tree as:



If any of v_1 or v_2 is pure (i.e. has data of only one label), then ~~predict~~ make it a leaf that predicts this label.

Impure Node: Node with data points with multiple labels

Each node in the algorithm corresponds to a subset of the training data.

Notes:

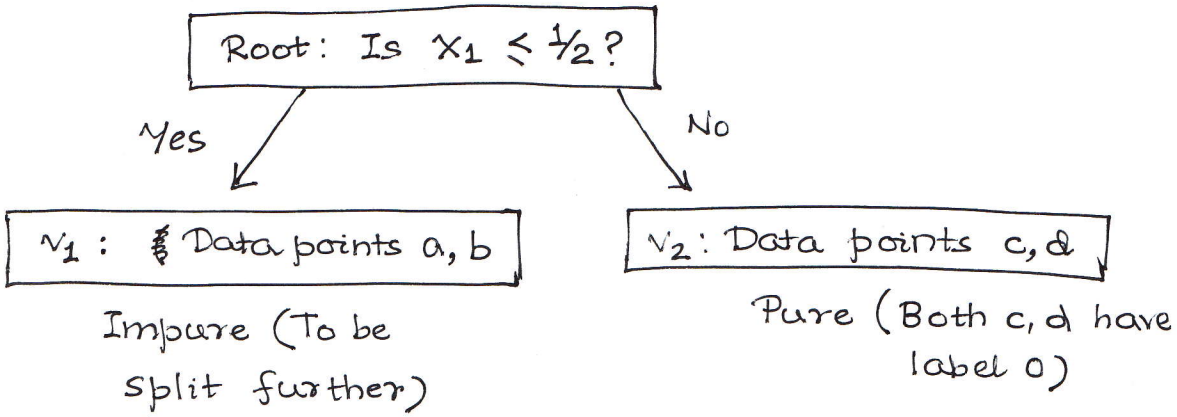
1. Convention: Use a threshold "in the middle" of two values.
   eg. Node has data points: (0,0), (0,1), (1,1), and if we split along feature 1, use threshold $1/2$.
2. More discussion of splitting rule coming up.

Example: Training data: $\underset{a}{((0,0),1)}, \underset{b}{((0,1),0)}, \underset{c}{((1,0),0)}, \underset{d}{((1,1),0)}$

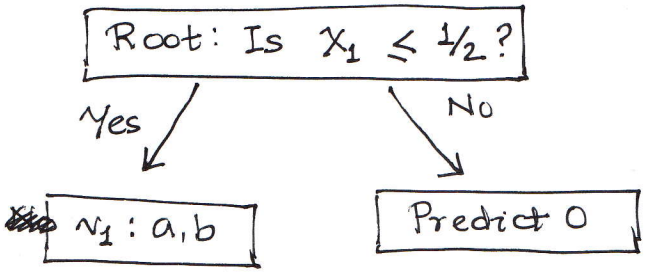
1. Initially: Root has (a, b, c, d)

Root: (a, b, c, d) Impure.

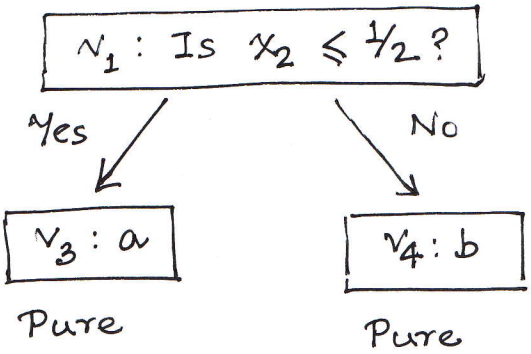
2. Suppose we pick x_1, threshold $1/2$ (thru Selection Rule).



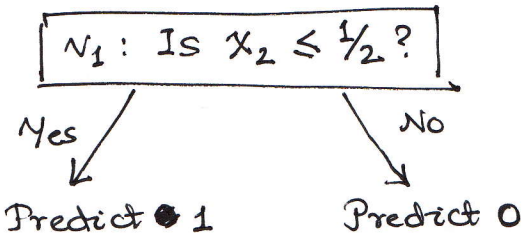
3.



4. Suppose we pick feature x_2, threshold $1/2$ (to split v_1)

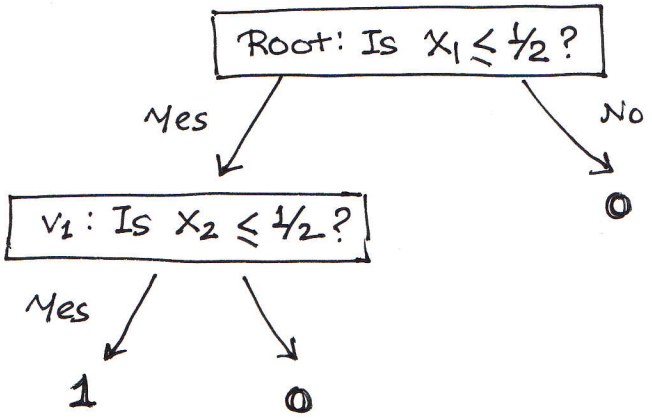


5.



All nodes are pure at this point, so we are done.

Complete Decision Tree:



Splitting Rule: How to choose a feature and threshold along which to split a node?

In ID3 Decision Trees, we choose one that reduces uncertainty the most.

How to measure uncertainty? Through a notion in information theory, called entropy

Entropy: Let X be a random variable that takes values $v_1, .., v_K$ with probabilities $p_1, .., p_K$. Then, entropy of X, denoted by $H(X)$ is defined as:

$$H(X) = -p_1 \log p_1 - p_2 \log p_2 - \cdots - p_K \log p_K$$

(Note: Use the convention that $0 \cdot \log 0 = 0$)

Some Entropy Values:

1. X: 0/1 random variable (r.v). $Pr(X=1) = 1/2$.

$p_0 = Pr(X=0) = 1/2$

$p_1 = Pr(X=1) = 1/2$

$$H(X) = -p_0 \log p_0 - p_1 \log p_1 = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = \log 2$$

$= 0.69$ (Usually, we use natural logarithm for these calculations.)

2. X: 0/1 r.v. $Pr(X=1) = 0$.

$p_0 = 1, \quad p_1 = 0.$

$$H(X) = -1\log 1 - 0\log 0 = 0$$

3. X is a r.v. which takes values $1, 2, .., k$ w.p. $\frac{1}{k}$ each.

$$H(X) = -\frac{1}{k}\log\frac{1}{k} - \frac{1}{k}\log\frac{1}{k} - \cdots - \frac{1}{k}\log\frac{1}{k}$$

$$= -k \cdot \frac{1}{k}\log\frac{1}{k} = \log k.$$

For 0/1 r.v, closer $Pr(X=0)$ is to a $1/2$, higher is the entropy.

Properties of Entropy:

1. $H(X)$ does not depend on the exact values taken by X, only on the probabilities of distinct values.

2. $H(X) \geq 0$ (always). Why?

$$H(X) = -\sum_{i=1}^{k} p_i \log p_i, \text{ where } \log p_i \leq 0, \text{ as } p_i \leq 1.$$

3. If X takes k values, $H(X) \leq \log k$. This maximum is achieved only when each value occurs w.p. $1/k$.

Conditional Entropy:

Let X, Z be two r.v.s. The conditional entropy of X given Z is defined as:

$$H(X|Z) = \sum_z \Pr(Z=z)\, H(X|Z=z)$$

(Intuitively, average entropy of X given that we know Z).

Example: 1

Joint Distribution of X, Z:

Z \ X	0	1
0	$\frac{1}{3}$	$\frac{1}{3}$
1	$\frac{1}{3}$	0

$\Pr(Z=0) = \frac{2}{3}$ $\quad \Pr(Z=1) = \frac{1}{3}$

For any x,

$$\Pr(X=x|Z=0) = \frac{\Pr(X=x, Z=0)}{2/3}$$

$$\Pr(X=x|Z=1) = \frac{\Pr(X=x, Z=1)}{1/3}$$

So: Conditional distributions are:

$X|Z=0$: $\Pr(X=0|Z=0) = 1/2$

$\Pr(X=1|Z=0) = 1/2$

$X|Z=1$: $\Pr(X=0|Z=1) = 1$

$\Pr(X=1|Z=1) = 0$

$$H(X|Z=0) = -\tfrac{1}{2}\log\tfrac{1}{2} - \tfrac{1}{2}\log\tfrac{1}{2} = \log 2$$

$$H(X|Z=1) = -1\log 1 - 0\log 0 = 0$$

So: $H(X|Z) = \frac{2}{3}\cdot\log 2 + \frac{1}{3}\cdot 0 = \frac{2}{3}\log 2.$

Example 2:

Joint distribution:

Z \ X	0	1	2
0	0	1/2	1/4
1	1/12	1/12	1/12

$Pr(Z=0) = 3/4$

$Pr(Z=1) = 1/4.$

Conditional distributions of X, given Z:

$Pr(X=0|Z=0) = 0$ $\quad$ $Pr(X=0|Z=1) = 1/3$

$Pr(X=1|Z=0) = 2/3$ $\quad$ $Pr(X=1|Z=1) = 1/3$

$Pr(X=2|Z=0) = 1/3$ $\quad$ $Pr(X=2|Z=1) = 1/3.$

$$H(X|Z=0) = -0\log 0 - \frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3} = \log 3 - \frac{2}{3}\log 2$$

$$H(X|Z=1) = -3\cdot\frac{1}{3}\log\frac{1}{3} = \log 3$$

$$\begin{aligned} H(X|Z) &= Pr(Z=0)\,H(X|Z=0) + Pr(Z=1)\,H(X|Z=1) \\ &= \frac{3}{4}\cdot\left(\log 3 - \frac{2}{3}\log 2\right) + \frac{1}{4}\log 3 \\ &= \log 3 - \frac{1}{2}\log 2 \end{aligned}$$

Properties of Conditional Entropy:

1. Suppose $X = Z$. What is $H(X|Z)$?

$$H(X|Z) = \sum_z Pr(Z=z)\, H(X|Z=z).$$

Now, given that $Z=z$, we know that $X=z$ w.p. 1.

So, $H(X|Z=z) = 0$.

Thus, $H(X|Z) = \sum_z Pr(Z=z) \cdot 0 = 0$.

(This may hold even if $H(X)$ is very large!)

2. Suppose X, Z are independent. What is $H(X|Z)$?

Since X, Z are independent, $Pr(X=x \mid Z=z)$ for any z and x, is equal to $Pr(X=x)$.

i.e. $X|Z=z$ has exactly the same distribution as X.

So. $H(X|Z=z) = H(X)$

$$H(X|Z) = \sum_z Pr(Z=z)\, H(X|Z=z) = H(X) \sum_z Pr(Z=z) = H(X)$$

Information Gain(Z) $= H(X) - H(X|Z)$

(essentially, how much entropy of X is reduced because we know Z).

It can be shown that information gain (IG) is ≥ 0 (always)

* Splitting Rule: We pick the feature and threshold which maximizes information gain.

$$IG(Z) = H(X) - H(X|Z).$$

Here X: distribution of labels at a node, say at node v.

(e.g. $\frac{2}{5}$ data points have label 0, $\frac{3}{5}$ label 1)

Each Z corresponds to a feature f and threshold t, takes 2 values (0 for No, 1 for Yes)

$X|Z=0$: distribution of labels at v among those data points for which feature $f \geq t$

$X|Z=1$: distribution of labels at v among those data points for which feature $f < t$.

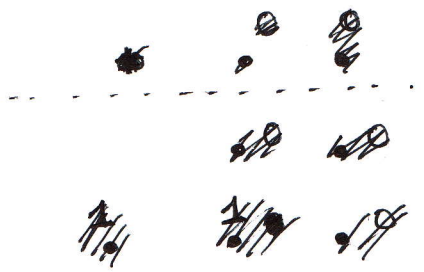
Example:

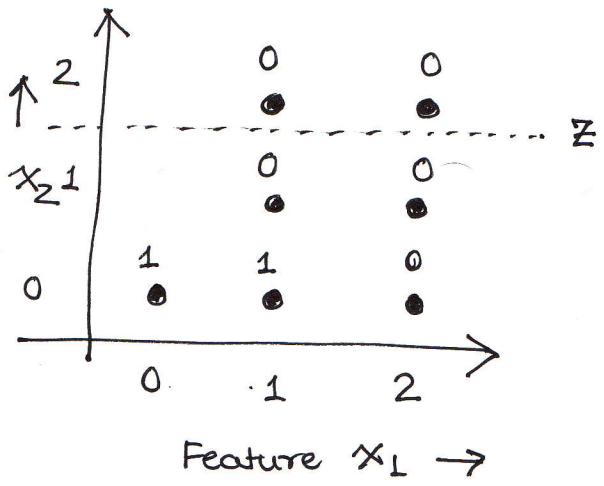
* . Suppose a node v has 5 points with label 0, 2 with label 1. Then, the corresponding X would be a r.v.

with: $Pr(X=0) = \frac{5}{7}$, $Pr(X=1) = \frac{2}{7}$.

So, $H(X) = -\frac{5}{7}\log\frac{5}{7} - \frac{2}{7}\log\frac{2}{7}$

* ~~Suppose, we split v among features f_1, thresholds t_1, use etc.~~





Suppose Figure shows the training data at a node v.

So, v has 5 label 0, 2 label 1 data points, and

$$H(X) = -\frac{5}{7}\log\frac{5}{7} - \frac{2}{7}\log\frac{2}{7}$$

Suppose we would like to split along Z. ($X_1 < 1.5$ or not).

Suppose $Z = 0$ corresponds to $X_1 \geq 1.5$ (1)

$Z = 1$ " " $X_1 < 1.5$ (2)

Then, $X|Z=0$ has 2 label 0s, 0 label 1s.

$$Pr(X=0|Z=0) = 1, \quad Pr(X=1|Z=0) = 0.$$

So, $H(X|Z=0) = -1\log 1 - 0\log 0 = 0$.

$X|Z=1$ has 3 label 0's and 2 label 1's.

$$Pr(X=0|Z=1) = 3/5, \quad Pr(X=1|Z=1) = 2/5.$$

$$H(X|Z=1) = -\frac{3}{5}\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5}$$

Also:

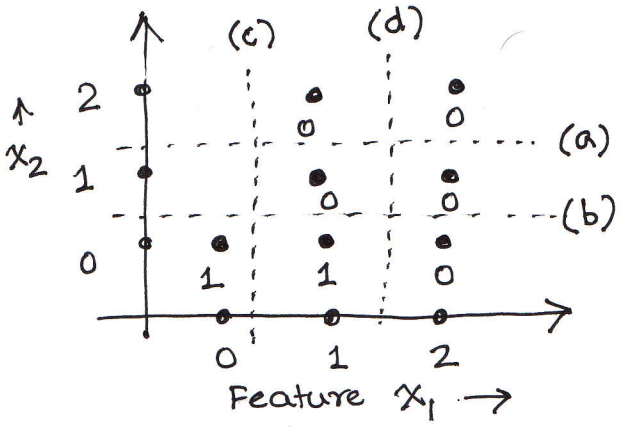
$Pr(Z=0) = 2/7$ (as 2 out of the total 7 points in X will lie on this branch)

and: $Pr(Z=1) = 5/7$ (as 5 " " " " 7 " " " " " " " " " ")

so: $H(X|Z) = Pr(Z=0)\,H(X|Z=0) + Pr(Z=1)\,H(X|Z=1)$

$= 0.48$

Example: How to Split a Node:



4 possible (feature, threshold) pairs to split along.

Call them (a), (b), (c), (d).

Initially: (5 0, 2 1)

(5 label 0's, 2 label 1's)

(a): Results: (2 0, 0 1); (3 0, 2 1)

$H(X|Z) = 0.48$

(b): Results: (4 0, 0 1); (1 0, 2 1)

$H(X|Z) = 0.27$

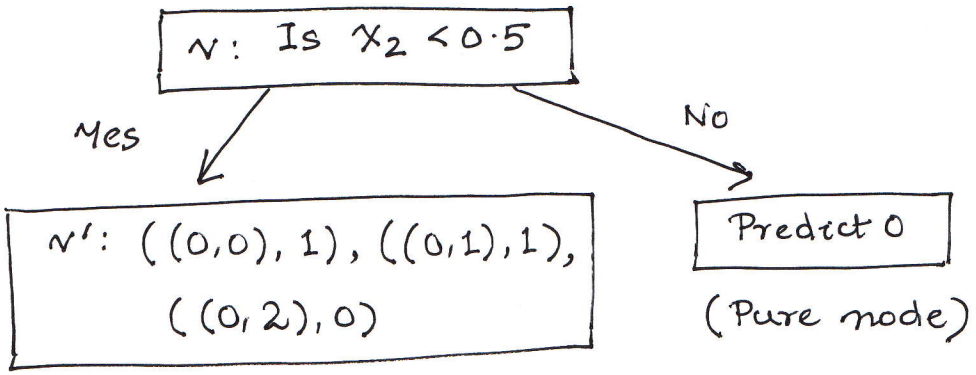
(c): Results: (5 0, 1 1); (0 0, 1 1)

$H(X|Z) = 0.38$

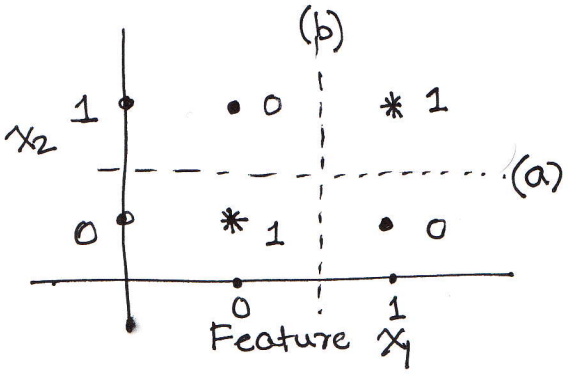
(d): Results: (2 0, 2 1); (3 0, 0 1)

$H(X|Z) = 0.39$

So, ID3 will select (b) to get:



Splitting a node: Example 2



Initially; ●(2 0, 2 1)

$H(X) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = \log 2$

$= 0.69.$

2 possible ways to split: (a) and (b).

For (a); Result: (1 0, 1 1), (1 0, 1 1)

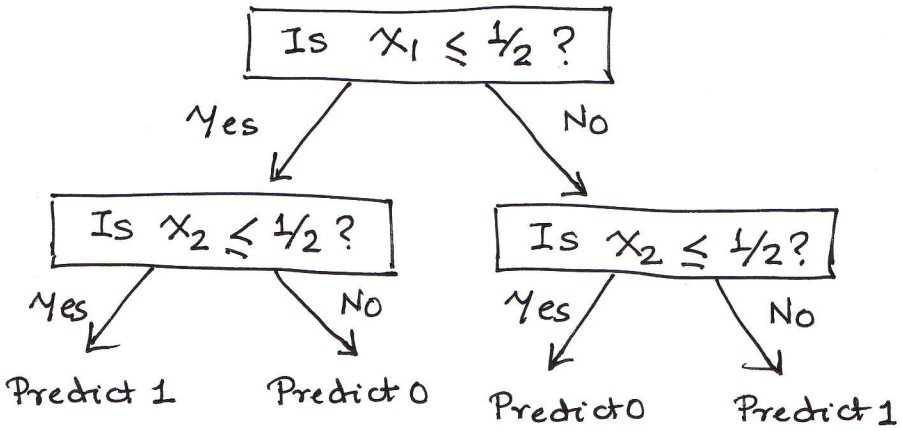
$H(X|Z) = 0.69$

For (b): Result: (1 0, 1 1); (1 0, 1 1)

$H(X|Z) = 0.69$

So, $IG(Z) = 0$.

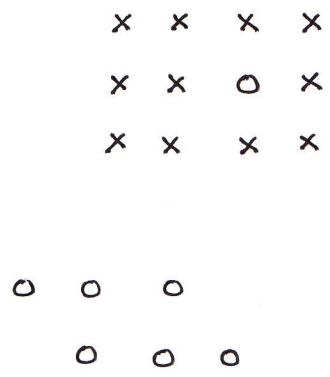
However, we shouldn't stop constructing the tree even if IG = 0. In this case, correct tree is:



ID3 Decision Trees:

Stopping Rule: Stop when every node is pure.
But this rule may overfit the data

Example:



The o may be an outlier or some noisy data.
If we build a decision tree that stops when all nodes are pure, then we may overfit the training data.

What is Overfitting?

Data comes from some underlying true distribution D.
Training data, test data → all samples from D.

Training error of a classifier:

$$\hat{err}(h) = \frac{1}{n}\sum_{i=1}^{n} 1(h(x_i) \neq y_i)$$

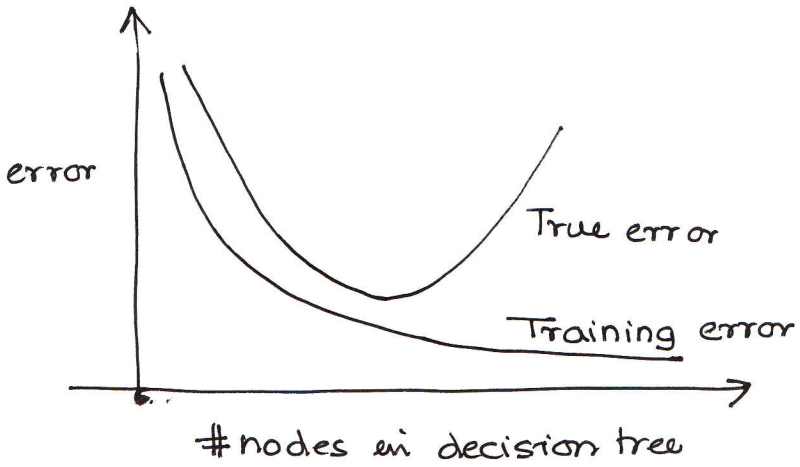
[$1(\cdot)$ is means an indicator variable which is 1 when the condition is true, 0 o/w.

True error of a classifier:

$$err(h) = \Pr_{(x,y)\sim D}(h(x) \neq y)$$

For a fixed classifier, with more data, training error should approach true error.

Overfitting happens when:



As we make the tree more and more complicated, training error decreases. At some point, true error stops improving and may get worse.

True data distribution has some structure, which we would like to capture. Training data captures some of this structure, but may have some "chance" structure of its own, which we should not model into a classifier.

How to Avoid Overfitting in Decision Trees? By Pruning.

- By pruning the tree using a validation dataset.

1. Split the training ~~data~~ data into training set S and validation set V.
2. Build ID3 tree T using training sets
3. Prune using V:

   Repeat:

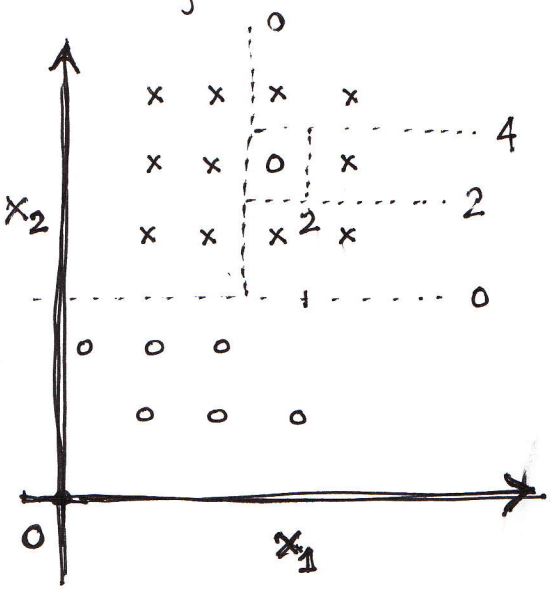
   For each node v in T:

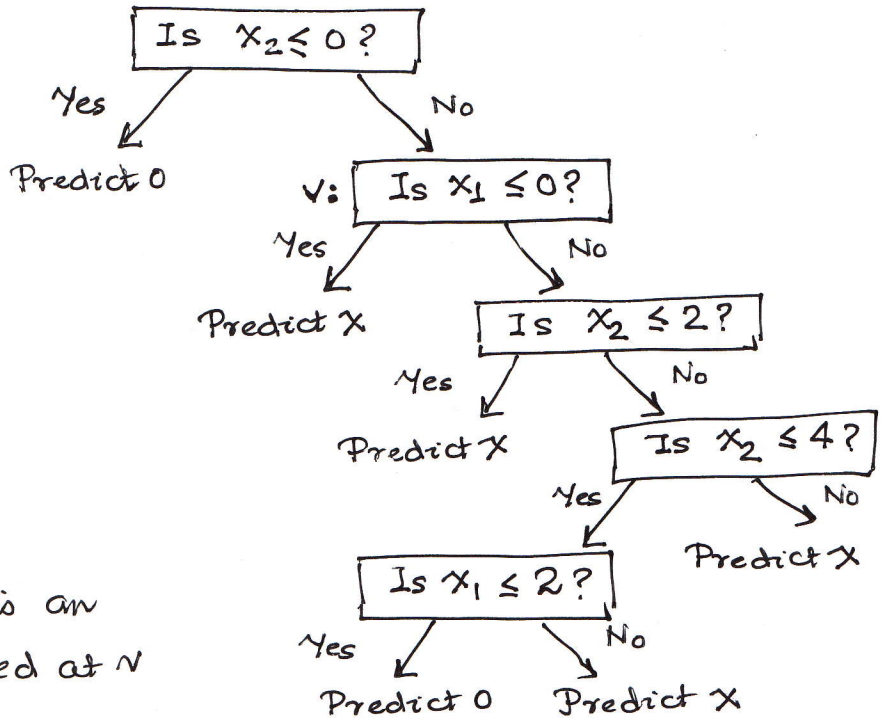
   Replace subtree rooted at v by single node that predicts majority label in v to get tree T'

   If error(T') on V $\leq$ error(T) on V, then T = T'

Continue till there is no such node v.

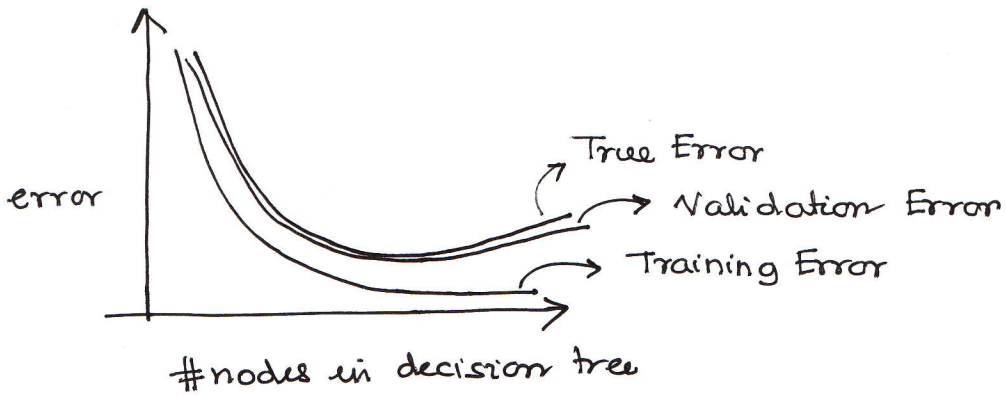
Example:





If the 0 point in between is an error, then the subtree rooted at N will have higher validation error than predicting X at N.

With Pruning:

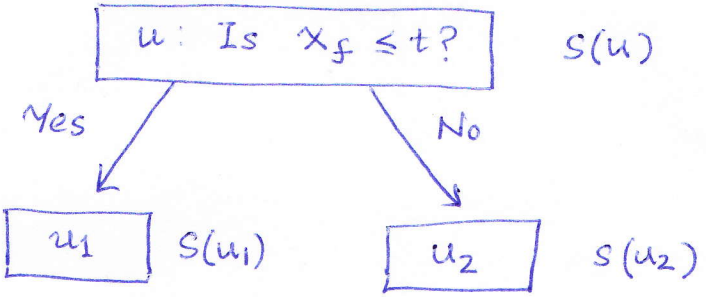


As tree grows more complicated, training error decreases.
At some point, true error stops improving and may even get worse. This will be reflected in the validation error, provided we have enough validation data.

Full Decision Tree Construction Algorithm (ID3):

* 1. Each node v: has a subset of training set $S(v)$
* Initially @ root; $S(\text{root})$ = full training set.

1. While $\exists$ an "impure" leaf u of the current tree:
   - Find a splitting rule for u. Say rule is: "Is $x_f \leq t$?"
   - Replace u with:



$S(u_1)$ = Subset of $S(u)$ for which $x_f \leq t$.
$S(u_2)$ = $S(u) \setminus S(u_1)$

   - If either u_1 or u_2 are "pure", make them a prediction node (that predicts the label of all of $S(u_1)$ or $S(u_2)$).

* "Pure" node: all data points associated with this node have the same label. Impure: Not pure.

Splitting Rule:

1. For all pairs of features f, thresholds t:
   * Compute Information Gain

$$IG = H(X) - H(X|Z)$$

   * Pick the pair that maximizes the Information Gain.