DT ~~Is~~ learning is a method of approximating discrete-valued target functions, in which learned function is represented by a decision Tree.

⇒ Can also be represented as if-then rules to improve human readability.

⇒ ———"——— as disjunction (or) of conjunctions (and) on the attribute/feature values of instances.

⇒ Each node in the tree specifies a test of some attribute of the instance and each branch descending from that node corresponds to the one possible values of the attribute.

⇒ Appropriate problems for decision tree learning:

→ Instances are represented by attribute-value (~~[illegible]~~) pair. i.e. attribute 'temprature' can have small # of disjoint possible values. eg. ('Hot', 'mild', 'cold'),

- Extension of 'DT' allows real valued attributes too.

→ The target function has discrete output values.

  → also handles real-values.

→ The training data may contain errors or missing attribute values.

  → DT learning are robust to errors in labelling to training examples or attribute values.

  → DT learning can handle instances with unknow attribute values.

The Basic DT learning Algo:-

The core algos employs a Top-Down Greedy ~~Approach~~ search through the space of possible DTs. (Hypotheses)

→ ID3 Algo

→ C4.5 Algo.

ID3 in it's pure form performs no backtracking in its search. Once it selects a particular attribute to test at a particular level in the tree it never backtracks.
So ID3 can converge to locally optimal solution.
There is an extension that calls post-pruning that adds a form of backtracking.

Inductive Bias : Set of assumptions that, together with the training data, deductively justifies the classification assigned by the learner to future instances.

Inductive v/s Deductive Reasoning :

→ aims a developing a theory
→ aims at testing an existing theory.
→ starting from a specific premises and forming a general conclusions.
→ using general premises to form a specific conclusion.

Inductive Bias of ID3 :

Shorter trees are preferred over longer trees. Trees that place high information gain attributes close to the root are preferred over those that do not.

BFS-ID3 ⇒ ID3 can be viewed as an efficient approximation to BFS-ID3, using a greedy heuristic search to attempt to find the shortest tree without conducting the entire Breadth first search through the hypothesis space.

Preference Bias for ID3 v/s restriction bias for linear function.

Thus far, we had discussed the entropy for special case where target classification is boolean.

if target attribute can take 'c' different values.

$$\text{Entropy}(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

where p_i = proportion of S belongs to class i.

How Information GAIN is related to Entropy?

IG ~~gain~~ Measures the expected reduction in entropy. Caused by partitioning the examples according to this attribute.

$$\cancel{\text{GAIN}(S,A) = \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)}$$

~~for collection S & Attribute A~~

$$\text{GAIN}(S,A) = \underbrace{\text{Entropy}(S)}_{\text{Entropy of original collection}} - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

for collection S & attribute A

↳ The weighted average of ~~all~~ entropies associated with all subsets partitioned on ~~attribute A~~ values of attribute A.

$$S_v = \{ s \in S \mid A(s) = v \}$$

~~Values(A,s)~~

Values of $A(S) = v$

in nutshell, GAIN(S,A) tells us what amount of entropy/impurity has reduces when we used Attribute A at a node.

Basic Algorithm. ID3.

~~Learth~~ learns DT by constructing them Top-Down beginning with the question " which attribute should be tested at the root of the tree?". ~~The~~ To answer the question, each instance attribute is evaluated ~~wis~~ using a stastical test to ~~dect~~ determine how well it alone classifies the training examples.

The Best attribute is selected and used as the test at the root node of the tree.

A descendent of the root node is then created for each possible value of this tree & the training examples are sorted to the appropriate descendent node.

The entire process is repeated using the training examples ~~as~~ associated with ~~the~~ each descendent nodes to select the best attribute to test at that point in tree.

This forms a greedy search for an acceptable DT, in which algorithm never ~~to~~ backtracks to reconsider earlier choices.

Which attribute is the Best choice at any node?

We need a quatitative measure called "Information gain" measures how well a given attribute seperates the training examples according to their target classification.

Entropy : Characterizes the ~~(im)purity~~ impurity of an arbitrary collection of examples.

→ A collection S, containing +ve & -ve examples of some target concept.

$$\boxed{\text{the entropy of } S = -p_{+ve} \log_2 p_{+ve} - p_{-ve} \log_2 p_{-ve}}$$

e.g. S = a collection of 14 examples of some boolean concept. 9 +ve & 5 -ve examples.

$$\text{Entropy}(S) = \text{Entropy}(9+, 5-) = -(9/14)\log_2(9/14) - \left(\frac{5}{15}\right)\log_2\left(\frac{5}{14}\right)$$

$$= 0.940$$

What are the extremes on entropy scale?

→ if all members belong ~~of~~ to ~~class~~ the same class

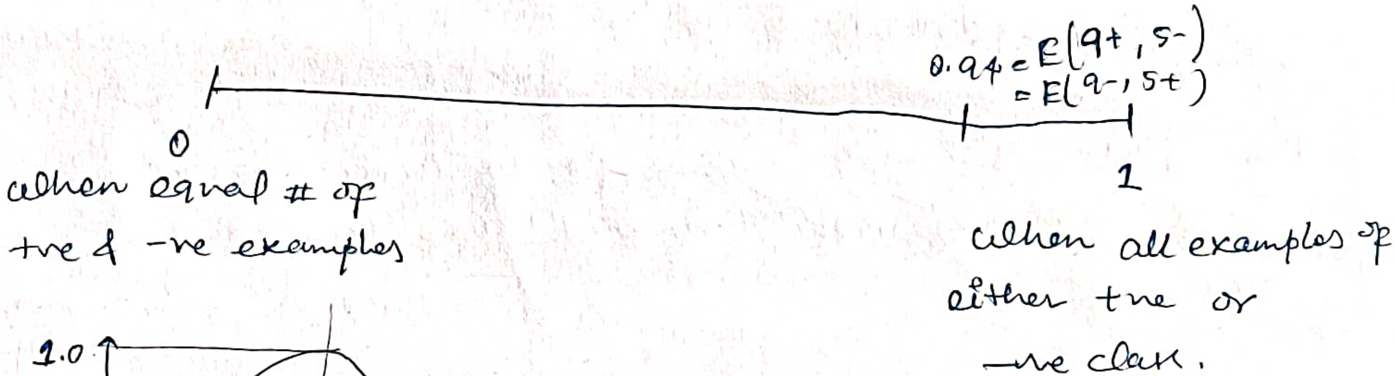
$$E = -\left(\frac{0}{14}\right)\log_2\left(\frac{0}{14}\right) - \frac{14}{14}\log_2\left(\frac{14}{14}\right)$$

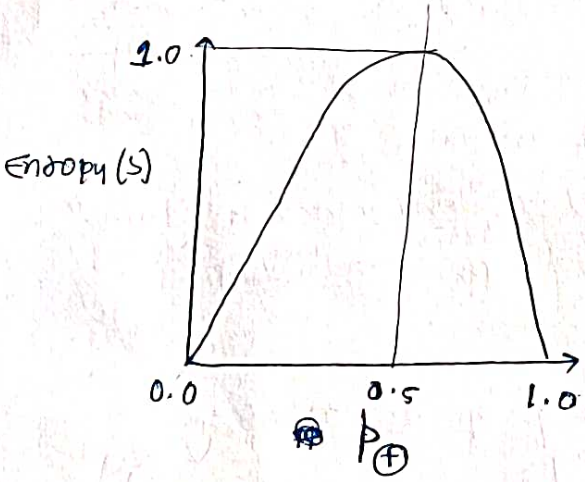
$$= -0 - 1\log_2 1 \Rightarrow -0-0 \Rightarrow 0$$

→ if equal # of +ve & -ve class examples.

$$E = -\left(\frac{7}{14}\right)\log_2\left(\frac{7}{14}\right) - \frac{7}{14}\log_2\frac{7}{14}$$

$$= -2\times\frac{1}{2}\log_2\frac{1}{2} \Rightarrow -2\times\frac{1}{2}\times -1 \Rightarrow 1$$





Overfitting: A hypothesis overfits the training examples if some other hypothesis that fits the training data less well actually performs better over the entire distribution of instances (i.e. including instances beyond the training set)

It can lead to overfitting when;

→ There is noise in the data.

→ # of training examples is too small to produce a representative sample of the true target function.

~~Th~~ In DT there are several approaches to avoid overfitting

→ approaches that stop growing the tree earlier ~~before it reaches the point~~

→ — " — that allows the tree to overfit the data and then post-prune the tree.

→ These are more successful technique, due to the difficulty in the first approach of estimating precisely when to stop growing the tree

→ Train-Validation set paradigm to avoid overfitting.

→ Separate data into Train set & validation set (usually in 2:1 ratio).

→ o Training set is used to form the learned hypothesis

o validation set is used to evaluate the accuracy of this hypothesis & evaluate the impact of pruning this hypothesis.

→ when the data is very less to divide, then this approach has its limits.