



SHIRAZ UNIVERSITY
Computer Science and Engineering Department
Machine Learning Lab

Machine Learning

Decision Trees

Based on:

Tom M. Mitchell, *Machine learning* ----- 3

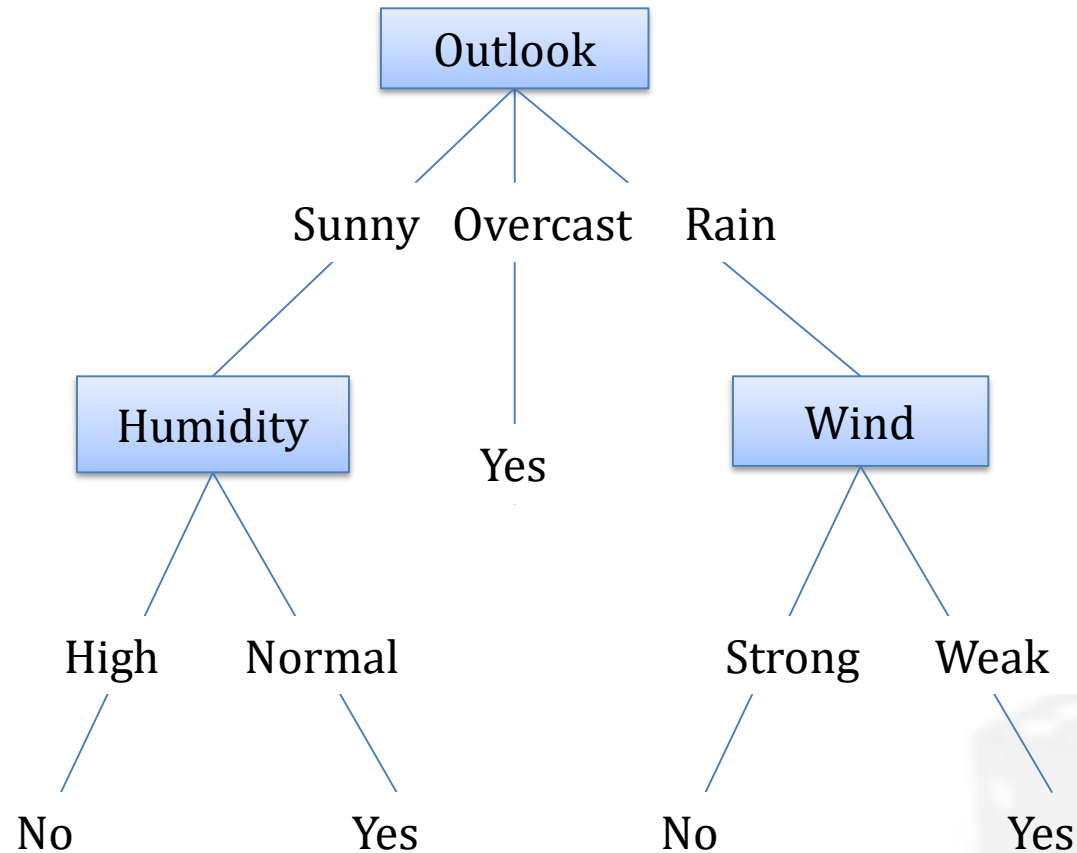
Fayyad, Usama, and Keki Irani. "Multi-interval discretization of continuous-valued attributes for classification learning." (1993).

Decision tree for play tennis



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

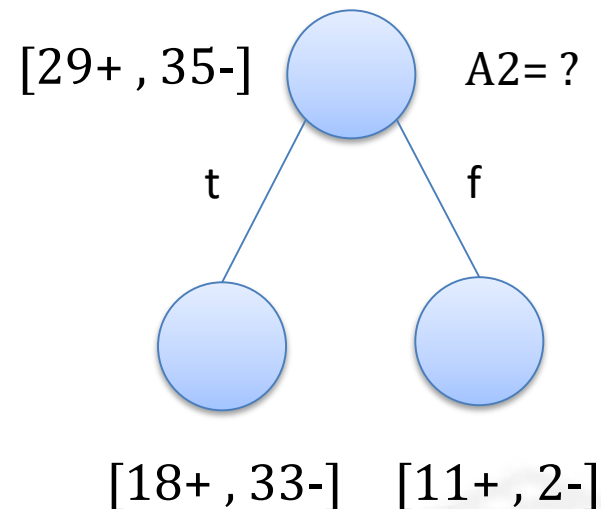
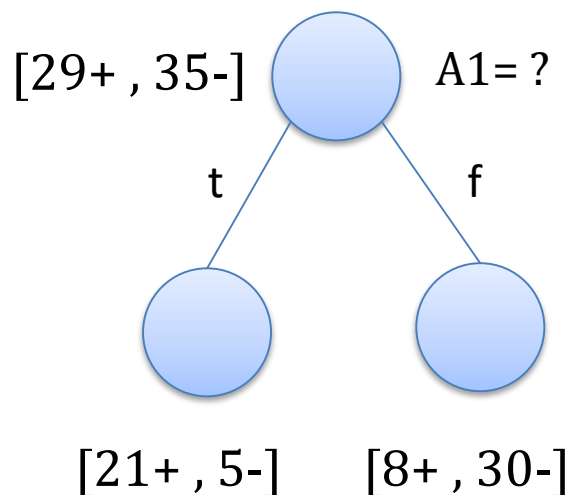
Decision tree for play tennis



Which attribute?



Which attribute is the best?



How to compare attribute?



➤ Entropy

- ❖ Entropy $H(X)$ of a random variable X

$$H(X) = - \sum_{i=1}^N P(x = i) \log_2 P(x = i)$$

- ❖ $H(X)$ is the expected number of bits needed to encode a randomly drawn value of X (under the most efficient code)
- ❖ Why?

Information theory:

Most efficient code assigns $-\log_2 P(X = i)$ bits to encode the message $X=i$,

So expected number of bits to code one random X is:

$$- \sum_{i=1}^N P(x = i) \log_2 P(x = i)$$

Entropy – another example



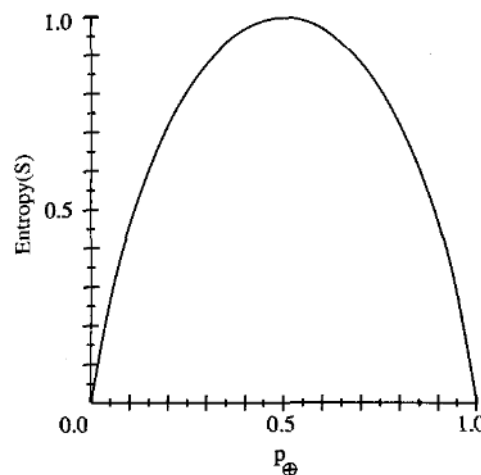
- ❖ *Example 1.1.2 (Cover):* Suppose we had a horse race with eight horses taking part, Assume their respective odds of winning are:

$$\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}\right)$$

- ❖ We can calculate the entropy as

$$H(X) = -\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{4}\log\left(\frac{1}{4}\right) - \frac{1}{8}\log\left(\frac{1}{8}\right) - \frac{1}{16}\log\left(\frac{1}{16}\right) - \\ 4 * \frac{1}{64}\log\left(\frac{1}{64}\right) = 2 \text{ bits}$$

Sample entropy



- ❖ Y is a class label
- ❖ P_+ is the proportion of label that is equal to 1
- ❖ P_- is the proportion of label that is equal to -1
- ❖ Entropy measure the impurity of Y

$$H(Y) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_-$$

Examples for computing entropy



$$H(X) = - \sum_{i=1}^N P(x = i) \log_2 P(x = i)$$

C1	0
C2	6

$$P(c1) = \frac{0}{6} = 0 \quad P(c2) = \frac{6}{6} = 1$$
$$\text{Entropy} = -0 \log 0 - 1 \log 1 = -0 - 0 = 0$$

C1	1
C2	5

$$P(c1) = \frac{1}{6} \quad P(c2) = \frac{5}{6}$$
$$\text{Entropy} = -\left(\frac{1}{6}\right) \log_2 \left(\frac{1}{6}\right) - \left(\frac{5}{6}\right) \log_2 \left(\frac{5}{6}\right) = 0.65$$

C1	2
C2	4

$$P(c1) = \frac{2}{6} \quad P(c2) = \frac{4}{6}$$
$$\text{Entropy} = -\left(\frac{2}{6}\right) \log_2 \left(\frac{2}{6}\right) - \left(\frac{4}{6}\right) \log_2 \left(\frac{4}{6}\right) = 0.92$$

Information gain



➤ Information Gain:

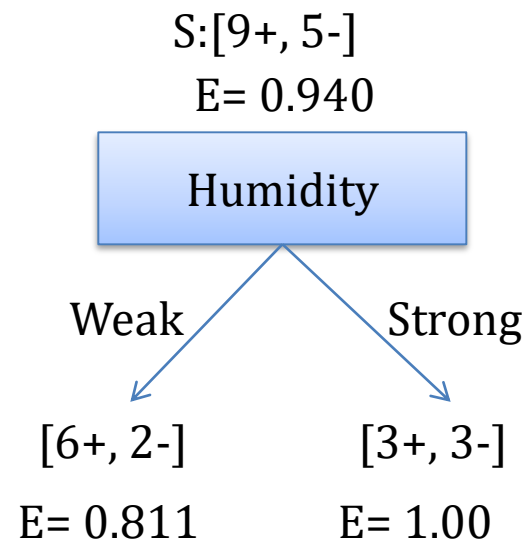
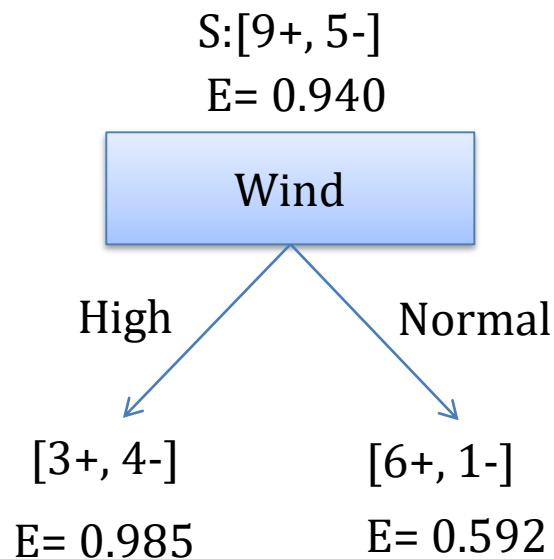
$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^k \frac{n_i}{n} Entropy(i) \right)$$

Parent Node, p is split into k partitions; n_i is number of records in portion i

- ❖ Measures Reduction in Entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN).
- ❖ Used in ID3 and C4.5.
- ❖ Disadvantage: Tends to prefer that in large number of partitions, each being small but pure.

Gain(S,A) = mutual information between A and target class variable over sample S.

Selecting the root attribute



$$\begin{aligned}
 \text{Gain}(S, \text{Humidity}) &= 0.940 - (7/14)0.985 - \\
 &\quad (7/14)0.592 \\
 &= 0.151
 \end{aligned}$$

$$\begin{aligned}
 \text{Gain}(S, \text{Wind}) &= 0.940 - (8/14)0.811 - (6/14)1.0 \\
 &= 0.048
 \end{aligned}$$

$$\begin{aligned}
 \text{Gain}(S, \text{Outlook}) &= 0.246 \\
 \text{Gain}(S, \text{Humidity}) &= 0.151 \\
 \text{Gain}(S, \text{wind}) &= 0.048 \\
 \text{Gain}(S, \text{Temperature}) &= 0.029
 \end{aligned}$$

Splitting based on INFO...



➤ Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{split}}{SplitINFO} \quad SplitINFO = - \sum_{i=1}^k \frac{n_i}{n} \log \frac{n_i}{n}$$

Parent Nodes, p is split into k partitions.
 n_i is the number of records in partition i .

- ❖ Adjust information Gain by the entropy of the partitioning (SplitINFO).
Higher entropy partitioning (large number of small partitions) is penalized!
- ❖ Used in C4.5.
- ❖ Designed to overcome the disadvantage of Information Gain.



Question

➤ Why use Information Gain to select attributes in decision trees? What other criteria seem reasonable?

❖ Gini $\sum \sum_{j \neq i} P_i P_j = (\sum P_i)^2 - \sum P_i^2 = 1 - \sum P_i^2$

❖ Chi square $\chi^2 = \sum \sum \frac{(x_{ij} - E_{ij})^2}{E_{ij}}$

❖ Marsh correction

❖ ...

Expected count example



House Style	Location		Total
	Urban Obs.	Rural Obs.	
Split- Level	63	49	112
Ranch	15	33	48
Total	78	82	160

Expected count example



House Style	Location		Total
	Urban Obs.	Rural Obs.	
Split- Level	63	49	112
Ranch	15	33	48
Total	78	82	160

Marginal probability = $\frac{112}{160}$

Expected count example



Marginal probability = $\frac{112}{160}$

House Style	Location		Total
	Urban Obs.	Rural Obs.	
Split- Level	63	49	112
Ranch	15	33	48
Total	78	82	160

Marginal probability = $\frac{78}{160}$

Expected count example



Joint probability = $\frac{78}{160} \frac{112}{160}$

Marginal probability = $\frac{112}{160}$

House Style	Location		Total
	Urban Obs.	Rural Obs.	
Split- Level	63	49	112
Ranch	15	33	48
Total	78	82	160

Marginal probability = $\frac{78}{160}$

Expected count example



$$\text{Joint probability} = \frac{78}{160} \frac{112}{160}$$

$$\text{Marginal probability} = \frac{112}{160}$$

House Style	Location		Total
	Urban Obs.	Rural Obs.	
Split- Level	63	49	112
Ranch	15	33	48
Total	78	82	160

$$\text{Marginal probability} = \frac{78}{160}$$

$$\begin{aligned}\text{Expected count} &= 160 \cdot \frac{112}{160} \frac{78}{160} \\ &= 54.6\end{aligned}$$



Function Approximation and Decision tree Learning



Function Approximation



Problem Setting:

Set of possible instances X

Unknown target function $f: X \rightarrow Y$

Set of function hypothesis $H = \{h|h: X \rightarrow Y\}$

Input:

Training examples $\{(x^i, y^i)\}$ of unknown target function f

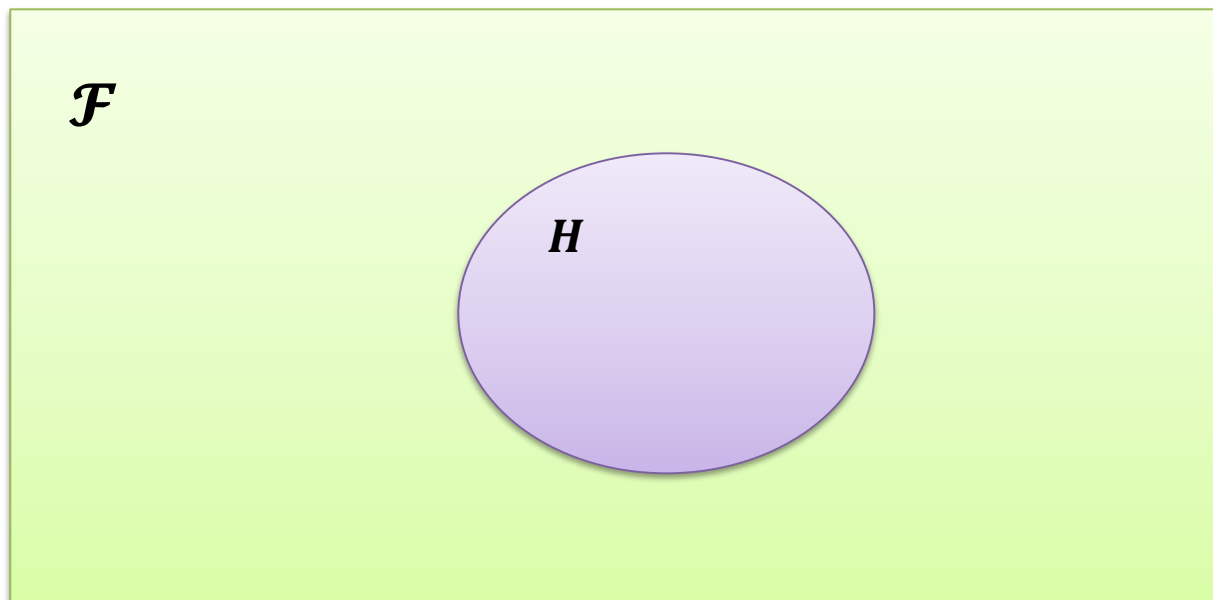
Output

Hypothesis $h \in H$ that best approximation target function f

Function Approximation



Unknown target function $f: X \rightarrow Y$
 $H \subset \mathcal{F}$



Hypothesis Representer

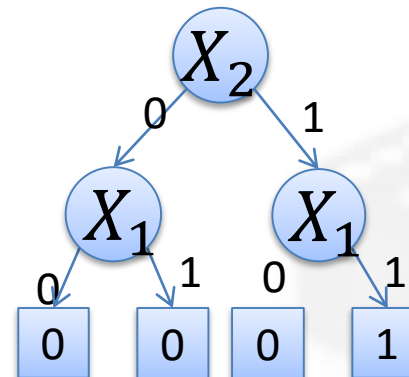
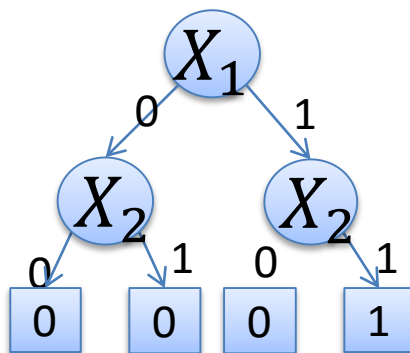


We use decision tree for representing functions that we want to approximate.

We might use decision tree or neural network or some other kind of representor to represent the approximated functions.

Is there only one decision tree can represent this function?

$$Y = X_1 \wedge X_2$$



Hypothesis Space of D.T.



$X_i, Y \in \{0,1\}$ and we have n features.

Unknown target function $f: X \rightarrow Y$

\mathcal{F}

$|\mathcal{F}| = ?$

Hypothesis Space of D.T.



$X_i, Y \in \{0,1\}$ and we have n features.

$$|\mathcal{F}| = 2^{2^n}$$

$$|H| = ?$$

We want number of functions that represent by decision trees, not number of decision trees?



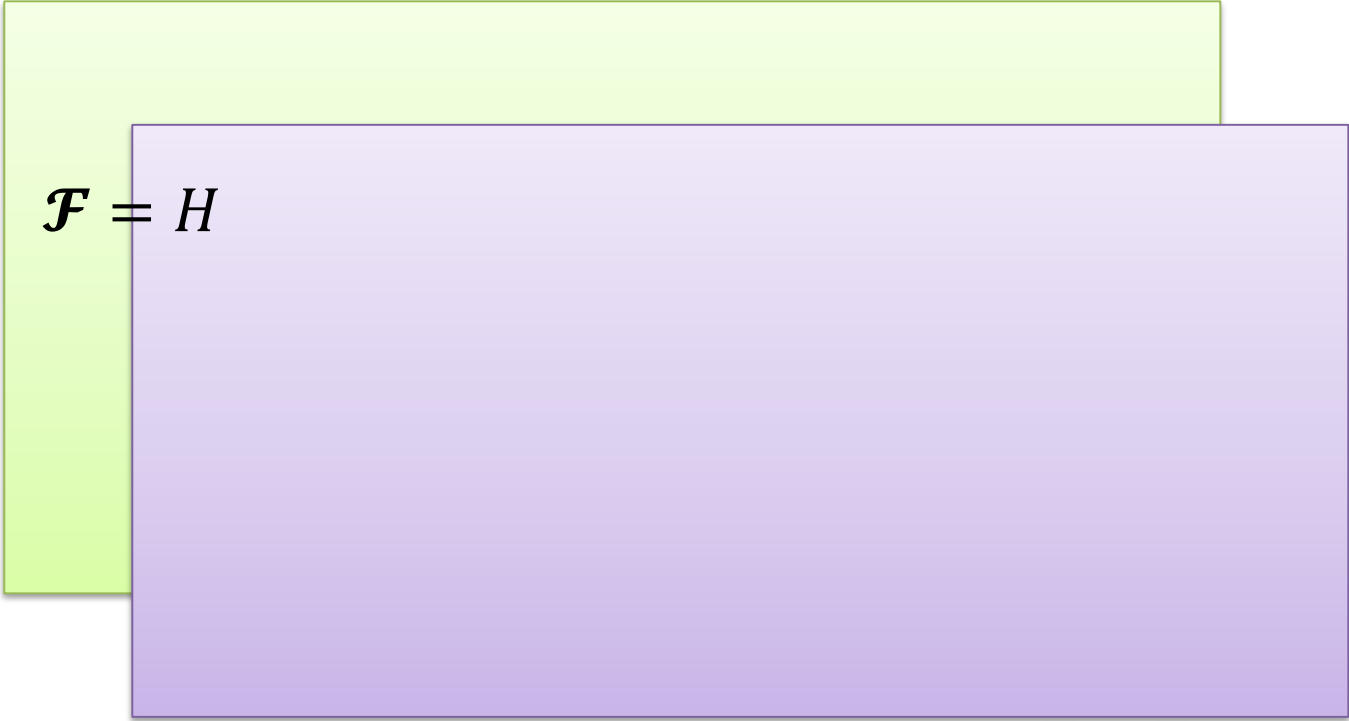
Hypothesis Space of D.T.



$$X_i, Y \in \{0,1\}$$

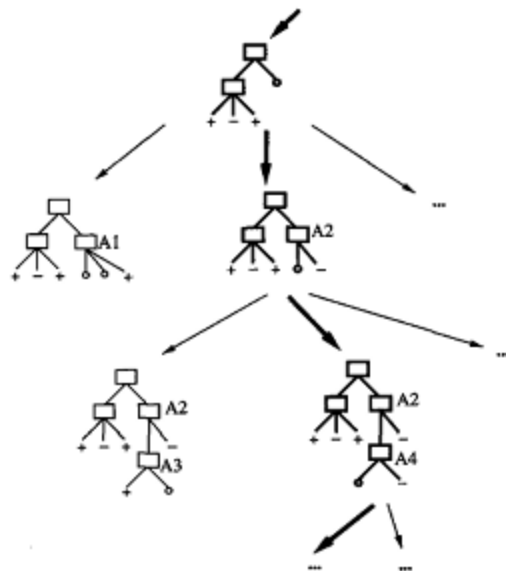
$$|H| = 2^{2^n}$$

Decision tree can represent all function $f: X \rightarrow Y$


$$\mathcal{F} = H$$

Inductive bias in decision trees

- ❖ 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.





Continuous variable and discretization



Quiz

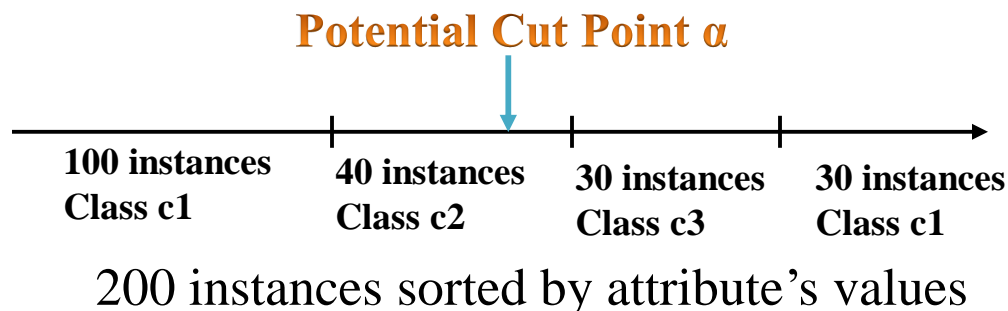


➤ What if some of the attributes are continues? (e.g. temperature in the pervious example.)

❖ 40, 48, 60, 72, 80, 90,...



Where is cut point α ?



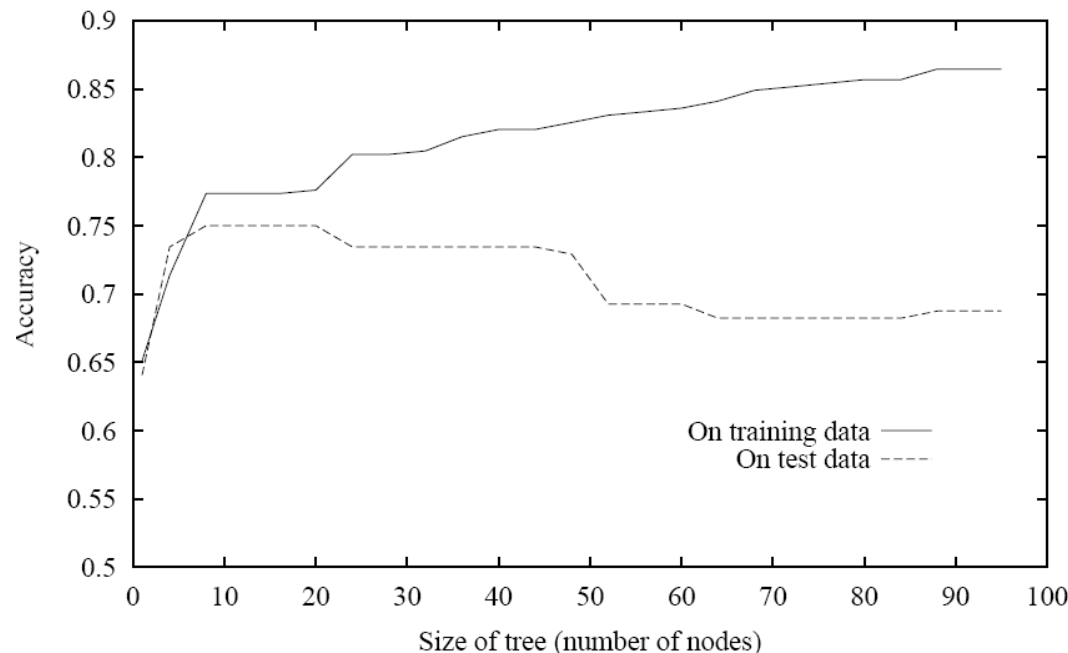
➤ Entropy Calculation

- ❖ The entropy of the left and right partitions (S_L, S_R) induced by every probable cut point α :

➤ Minimum Entropy

- ❖ The cut point α is selected whose $E(x_j, \alpha, S)$ is minimal amongst all candidates

Avoiding over-fitting the data



Given a hypothesis space H , a hypothesis $h \in H$ is said to overfit the training data if there exists some alternative hypothesis $h' \in H$, such that h has smaller error than h' over the training examples, but h' has a smaller error than h over the entire distribution of instances.

Pruning decision trees



- ❖ Approaches that stop growing the tree earlier, before it reaches the point where it perfectly classifies the training data.
- ❖ Approaches that allow the tree to overfit the data, and then post-prune the tree.

