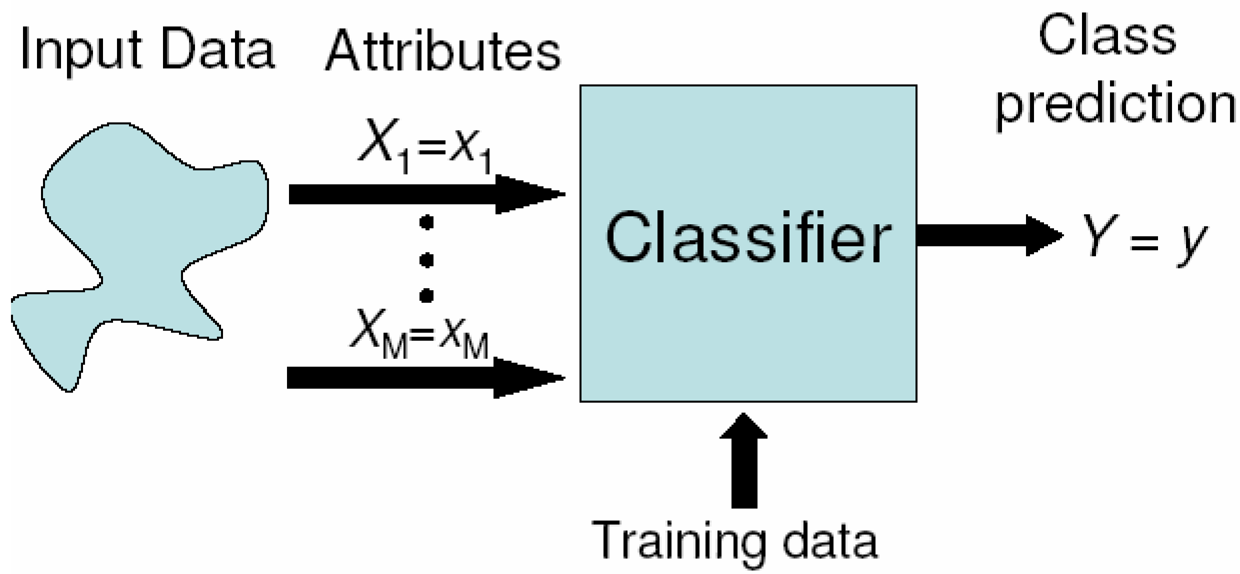
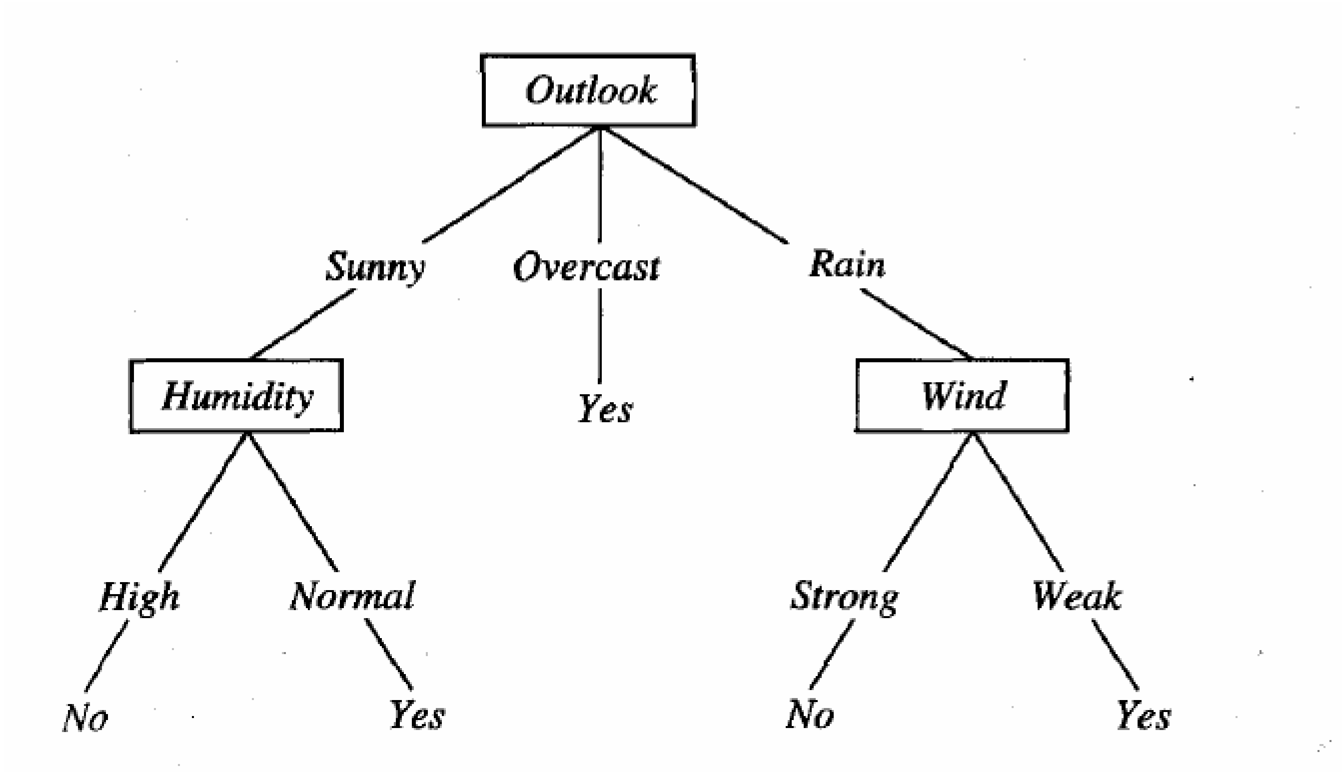
Decision Tree


Input Data
Attributes
Class prediction
$X_1=x_1$
$X_M=x_M$
Classifier
$Y = y$
Training data

Example 1

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
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

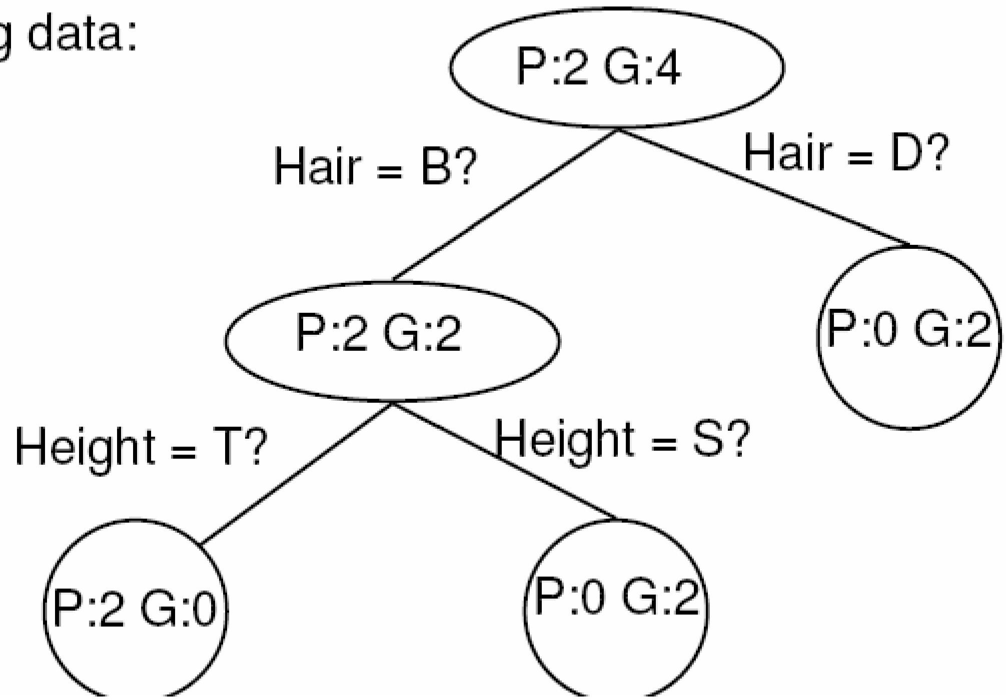
Example 1 – Decision Tree

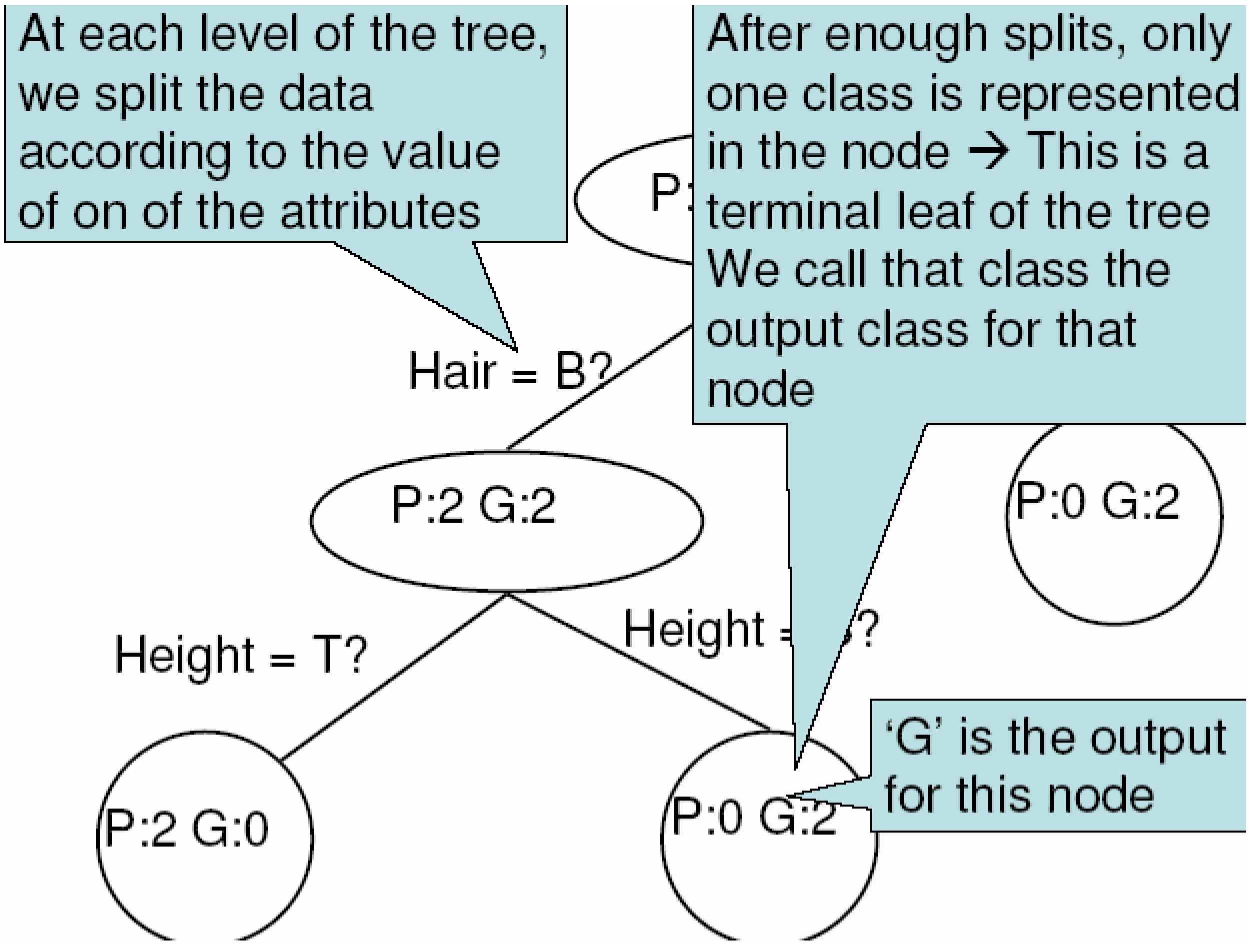


Decision Tree Example

- Three variables:
  - Hair = {blond, dark}
  - Height = {tall,short}
  - Country = {Gromland, Polvia}

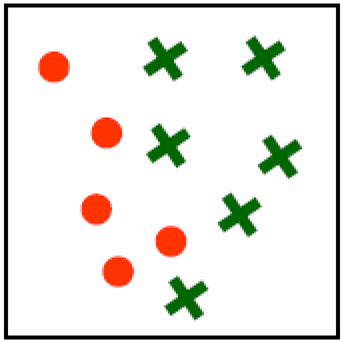
Training data:
(B,T,P)
(B,T,P)
(B,S,G)
(D,S,G)
(D,T,G)
(B,S,G)




At each level of the tree, we split the data according to the value of on of the attributes
After enough splits, only one class is represented in the node → This is a terminal leaf of the tree We call that class the output class for that node
Hair = B?
P:2 G:2
P:0 G:2
Height = T?
P:2 G:0
P:0 G:2
'G' is the output for this node

Example 3

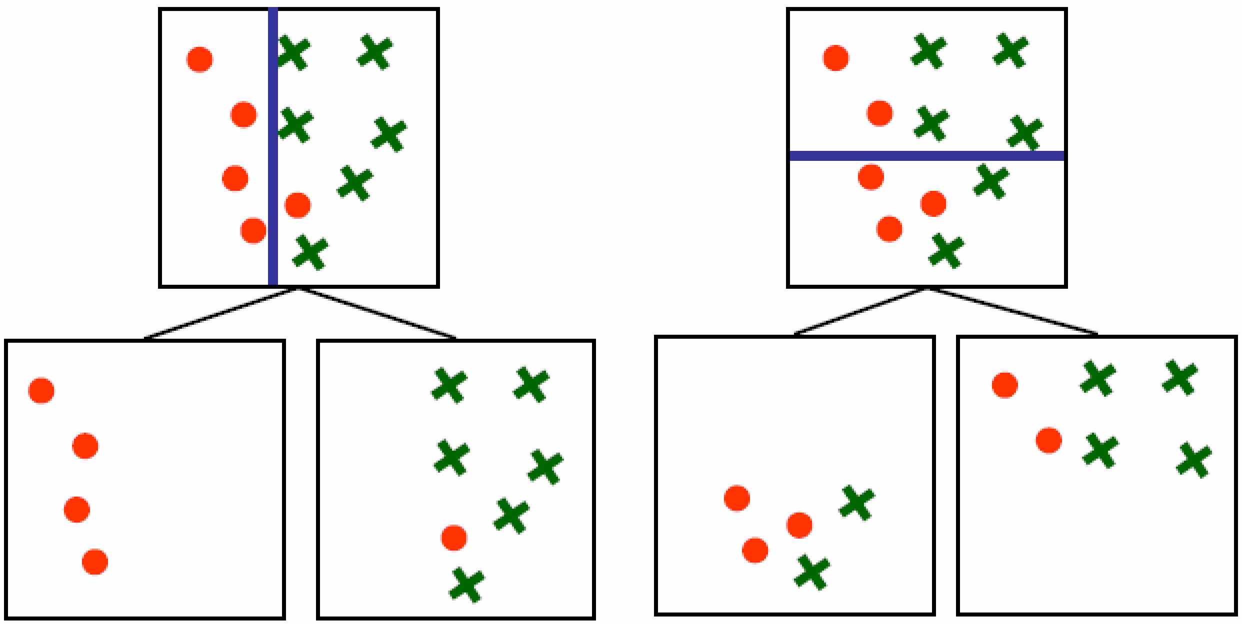
How to choose the attribute/value to split on at each level of the tree?



- Two classes (red circles/green crosses)
- Two attributes: X_1 and X_2
- 11 points in training data
- Idea → Construct a decision tree such that the leaf nodes predict correctly the class for all the training examples

Example 2 – Attribute Selection

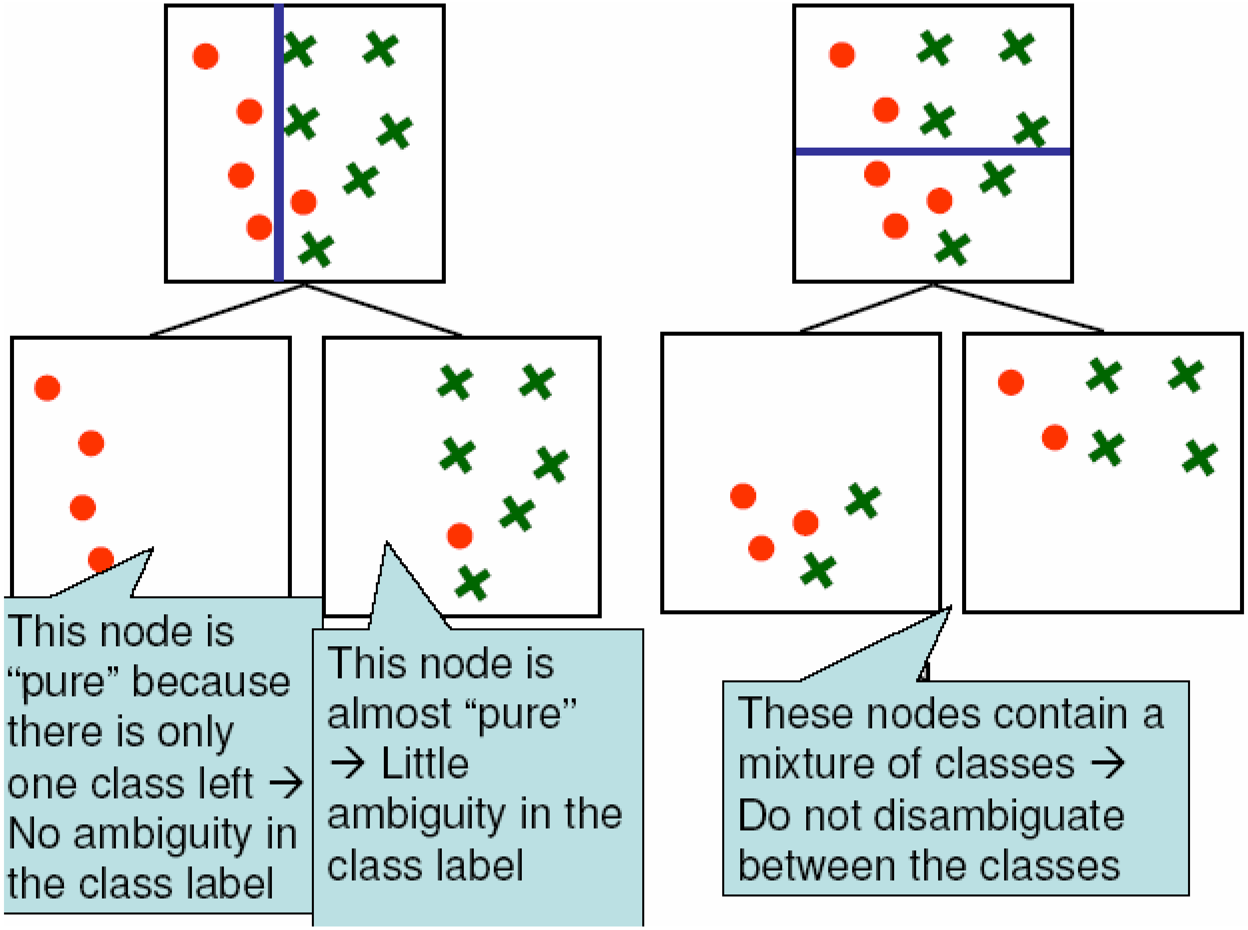
How to choose the attribute/value to split on
at each level of the tree?



Good

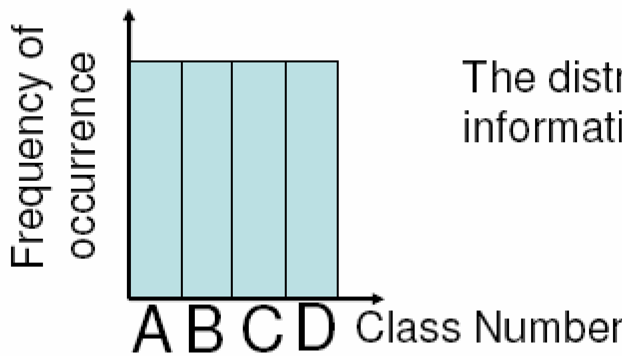
Bad

Example 2 – Attribute Selection



Digression: Information Content

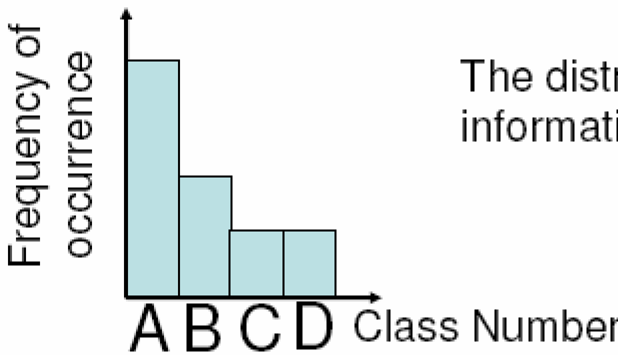
- Suppose that we are dealing with data which can come from four possible values (A, B, C, D)
- Each class may appear with some probability
- Suppose P(A) = P(B) = P(C) = P(D) = 1/4
- What is the average number of bits necessary to encode each class?
- In this case: average = 2 = 2xP(A)+2xP(B)+2xP(C)+2xP(D)
  - A → 00 B → 01 C → 10 D → 11



The distribution is not very informative → impure

Information Content

- Suppose now P(A) = 1/2 P(B) = 1/4 P(C) = 1/8 P(D) = 1/8
- What is the average number of bits necessary to encode each class?
- In this case, the classes can be encoded by using 1.75 bits on average
- A → 0 B → 10 C → 110 D → 111
- Average
  = 1xP(A)+2xP(B)+3xP(C)+3xP(D) = 1.75



The distribution is more informative → higher purity

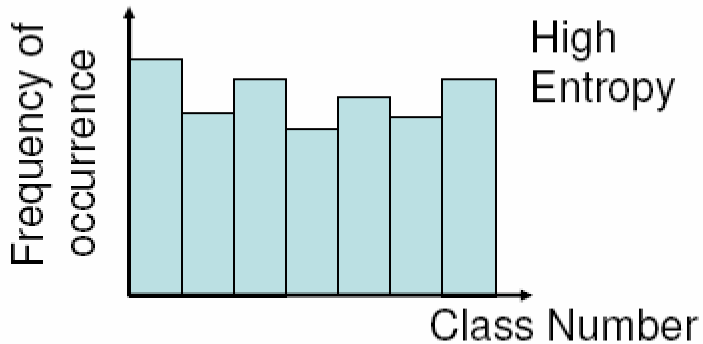
Entropy

- In general, the average number of bits necessary to encode *n* values is the entropy:

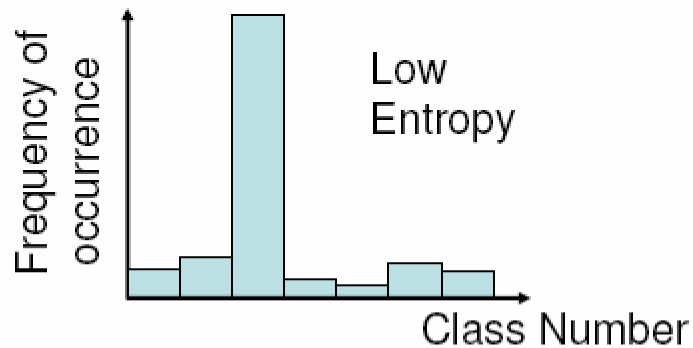
$$H = -\sum_{i=1}^{n} -P_i \log_2 P_i$$

- P_i = probability of occurrence of value *i*
  - High entropy → All the classes are (nearly) equally likely
  - Low entropy → A few classes are likely; most of the classes are rarely observed

Entropy

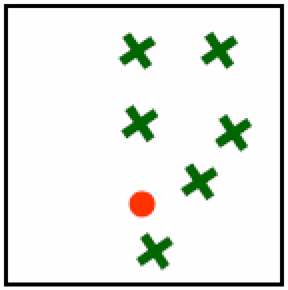


The entropy captures the degree of “purity” of the distribution



Example Entropy Calculation

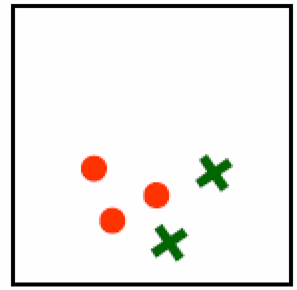
(1)



$N_A = 1$
$N_B = 6$
$p_A = N_A/(N_A+N_B) = 1/7$
$p_B = N_B/(N_A+N_B) = 6/7$

$H_1 = -p_A \log_2 p_A - p_B \log_2 p_B$
$= 0.59$

(2)

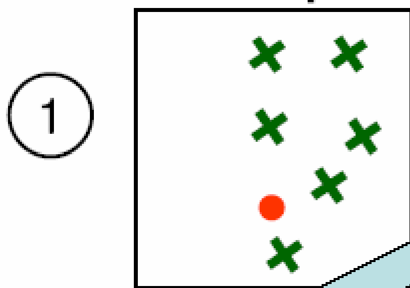


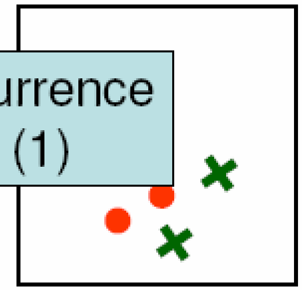
$N_A = 3$
$N_B = 2$
$p_A = N_A/(N_A+N_B) = 3/5$
$p_B = N_B/(N_A+N_B) = 2/5$

$H_2 = -p_A \log_2 p_A - p_B \log_2 p_B$
$= 0.97$

$H_1 < H_2$ => (2) less pure than (1)

Example Entropy Calculation





Frequency of occurrence of class A in node (1)

Frequency of occurrence of class B in node (1)

Entropy of node (1)

$N_A = 1$

$N_B = 6$

$p_A = N_A/(N_A+N_B) = 1/7$

$p_B = N_B/(N_A+N_B) = 6/7$

$H_1 = -p_A \log_2 p_A - p_B \log_2 p_B$

$= 0.59$

$p_A = N_A/(N_A+N_B) = 3/5$

$p_B =$

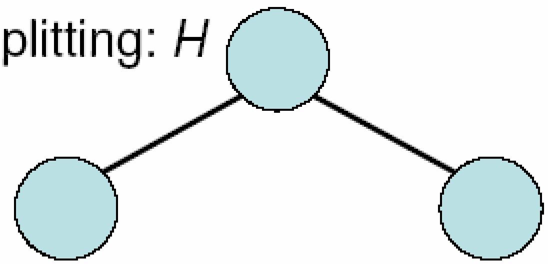
$H_2 = -p_A \log_2 p_A - p_B \log_2 p_B$

$= 0.97$

$H_1 < H_2$ => (2) less pure than (1)

Conditional Entropy

Entropy before splitting: H



After splitting, a fraction P_L of the data goes to the left node, which has entropy H_L

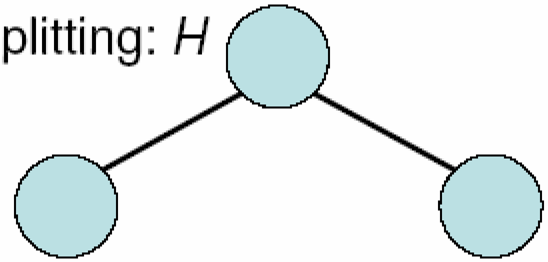
After splitting, a fraction P_R of the data goes to the left node, which has entropy H_R

The average entropy after splitting is:

$$H_L \times P_L + H_R \times P_R$$

Conditional Entropy

Entropy before splitting: H



After splitting, a fraction P_L of the data goes to th... has e...

...ropy H_R

Entropy of left node

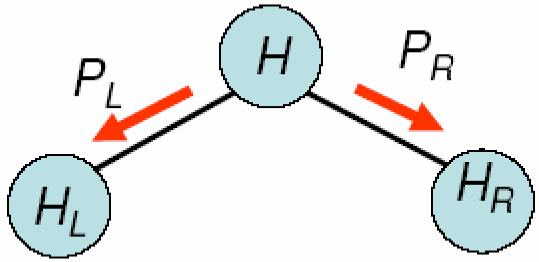
Probability that a random input is directed to the left node

The av...ge entro... after splitting is:

$$H_L \times P_L + H_R \times P_R$$

“Conditional Entropy”

Information Gain



We want nodes as pure as possible

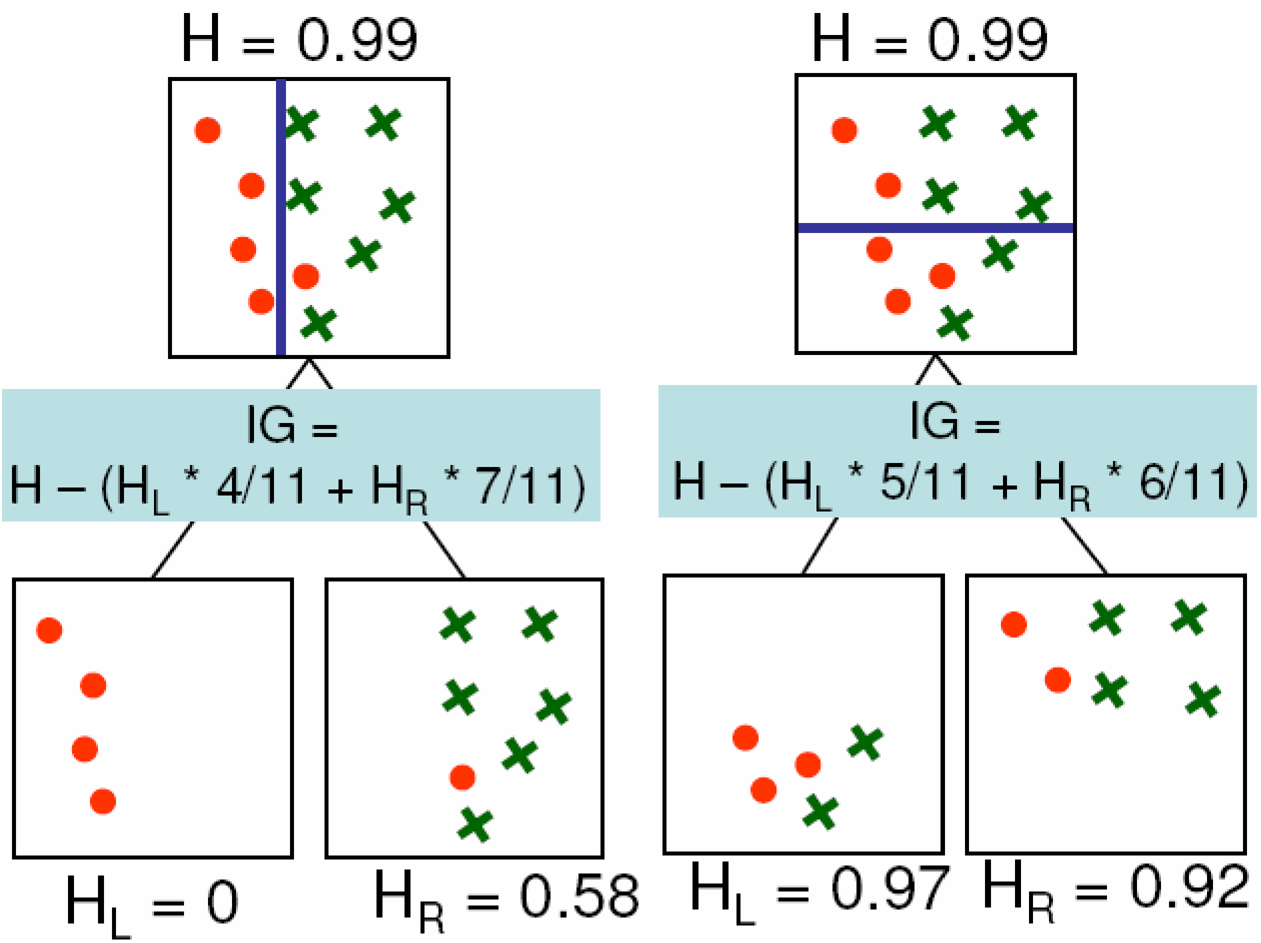
→We want to reduce the entropy as much as possible

→ We want to maximize the difference between the entropy of the parent node and the expected entropy of the children

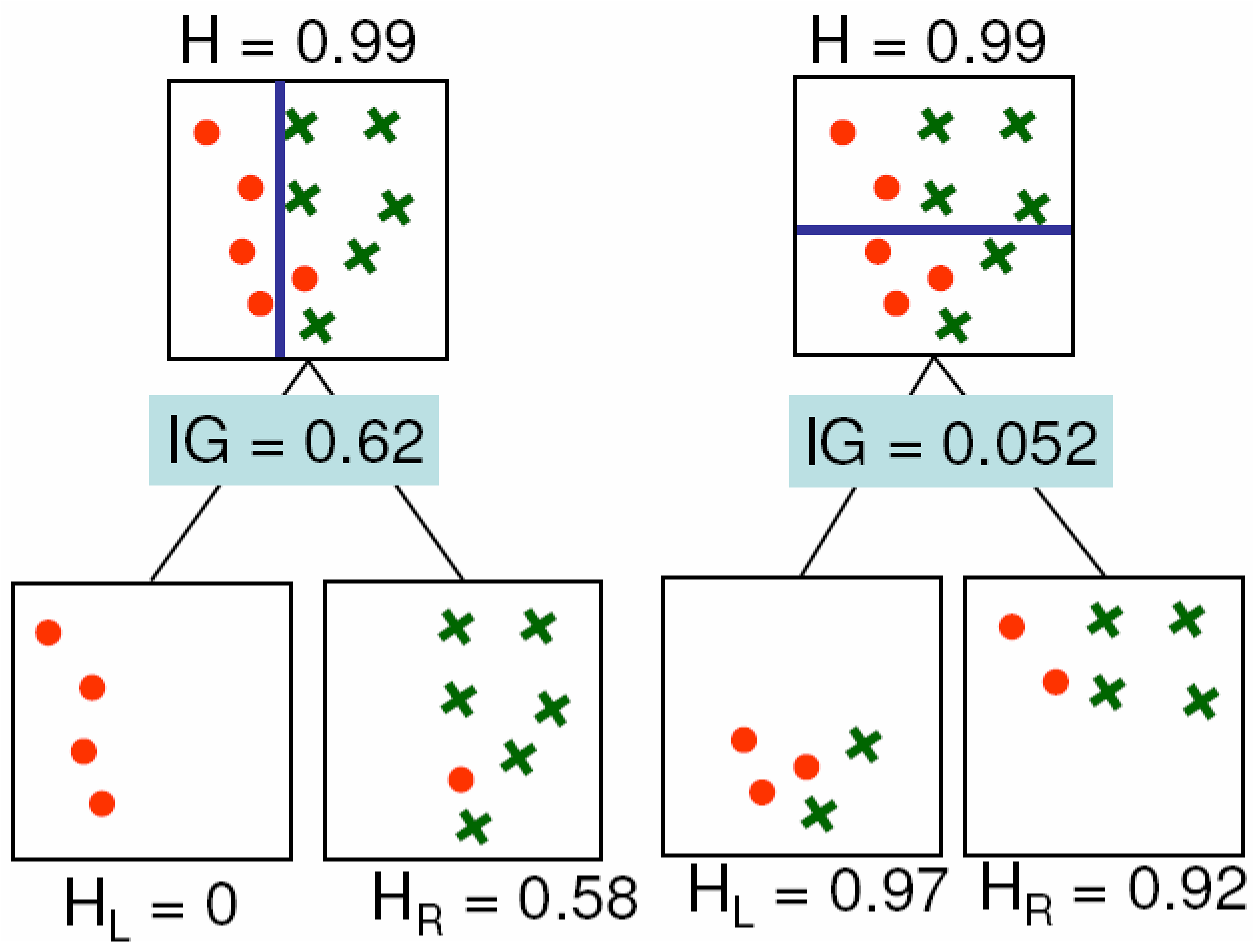
Maximize:

$$IG = H - (H_L \times P_L + H_R \times P_R)$$

Example 2 – Attribute Selection



Example 2 – Information Gain



Example 1 – Revisited

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
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

Example 2 – Information Gain

$$Values(Wind) = Weak, Strong$$

$$S = [9+, 5-]$$

$$S_{Weak} \leftarrow [6+, 2-]$$

$$S_{Strong} \leftarrow [3+, 3-]$$

$$\begin{aligned} Gain(S, Wind) &= Entropy(S) - (8/14)Entropy(S_{Weak}) \\ &\quad - (6/14)Entropy(S_{Strong}) \\ &= 0.940 - (8/14)0.811 - (6/14)1.00 \\ &= 0.048 \end{aligned}$$

Example 2 – Information Gain