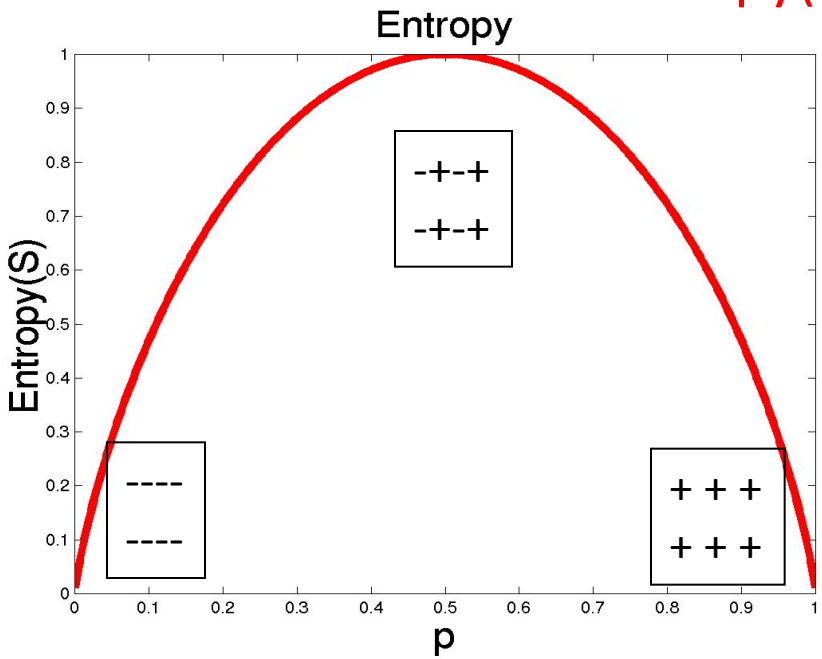
Decision Trees

- Decision tree representation
- ID3 Iterative Dichotomizer learning algorithm
- Entropy, information gain
- Overfitting

ID3:Entropy

$$Entropy(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$



- S is a sample of training examples taken from a population
- p_+ is the proportion of positive examples, very high or very low ⇒ low entropy.
- p_- is the proportion of negative examples
- Entropy measures the **impurity of S**

Top-Down Induction of Decision Trees -ID3 – A greedy BFS method

1. Start by calculating entropy of Decision node

2. A ← the "best" decision attribute for next *node*
Least Entropy, highest Information Gain

3. For each value of A create new descendant

4. If all training examples are perfectly classified (same value of target attribute) stop, else iterate over new leaf nodes.

Training Examples

$$Entropy(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Strong	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	Weak	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

Entropy of the Table's Decision Field

- $-(9/14) \times \log_2 (9/14) - (5/14) \times \log_2 (5/14) = .940$

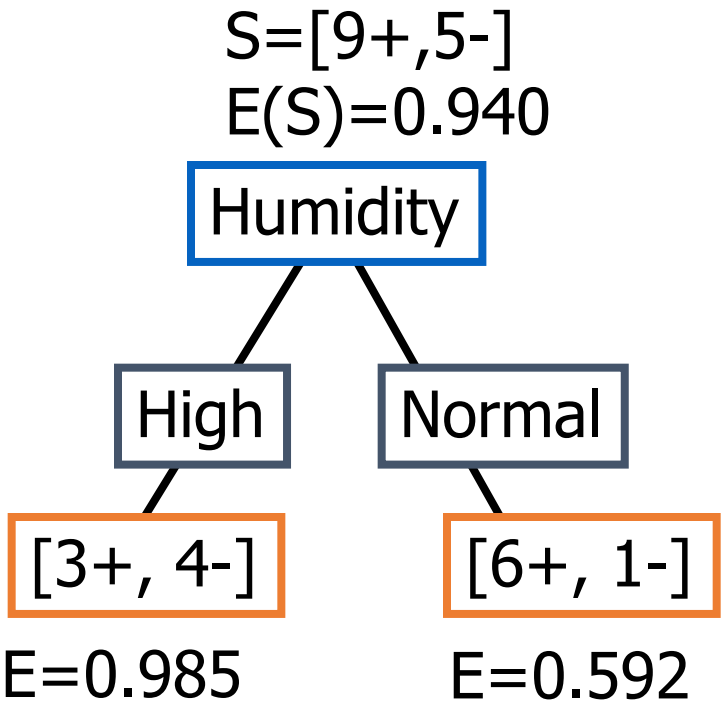
- Shows the amount of disorderliness

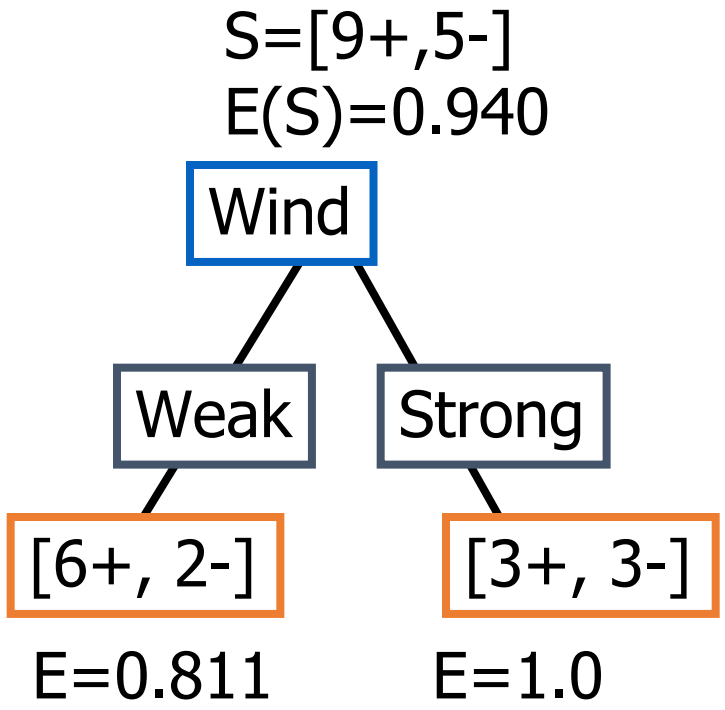
Information Gain

- Information Gain(S,A): expected reduction in entropy due to sorting S on attribute A

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in D_A} \frac{|S_v|}{|S|} Entropy(S_v)$$

Selecting the Next Attribute



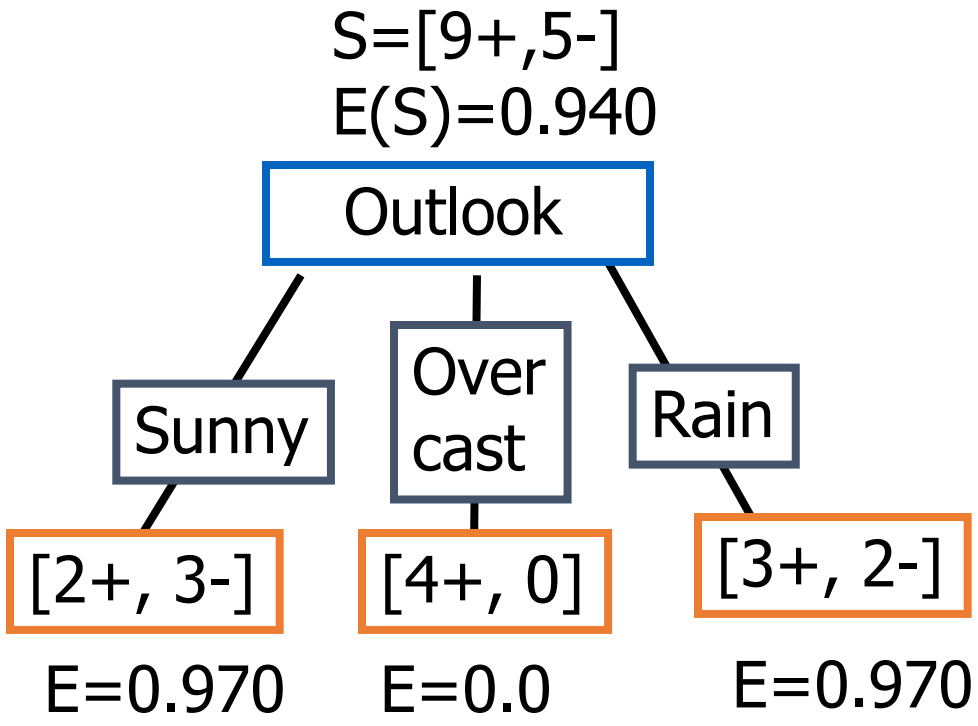


Gain(S,Humidity)
=0.940-(7/14)*0.985
– (7/14)*0.592
=0.151

Gain(S,Wind)
=0.940-(8/14)*0.811
– (6/14)*1.0
=0.048

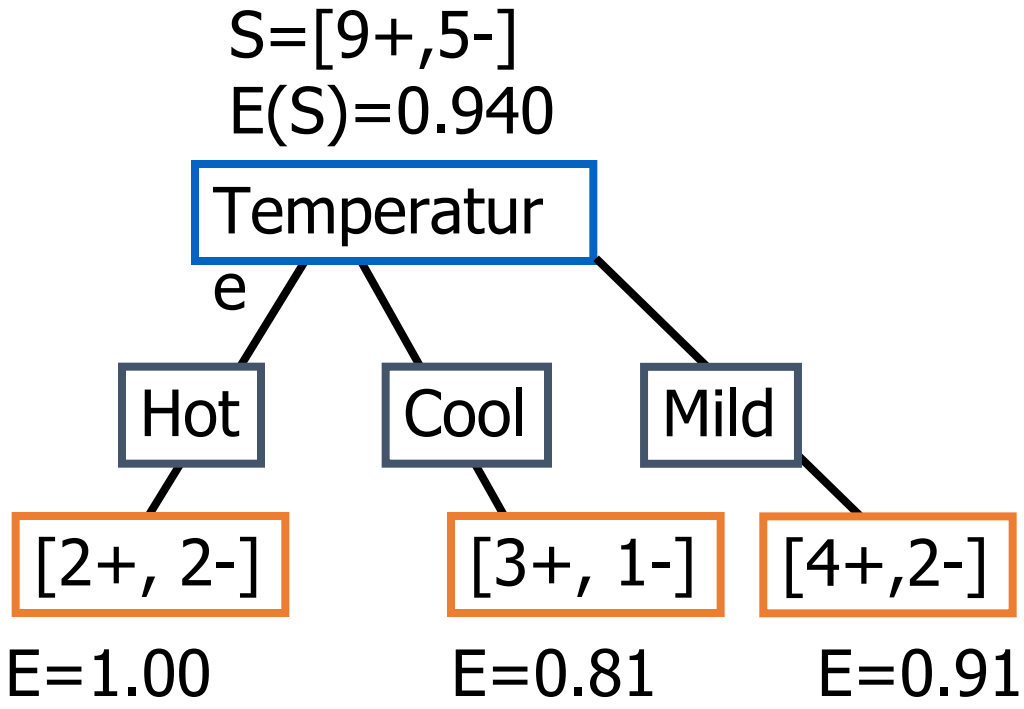
Humidity provides greater info. gain than Wind, w.r.t target classification.

Selecting the Next Attribute



Gain(S,Outlook)
=0.940-(5/14)*0.971
-(4/14)*0.0 – (5/14)*0.0971
=0.247

Temperature



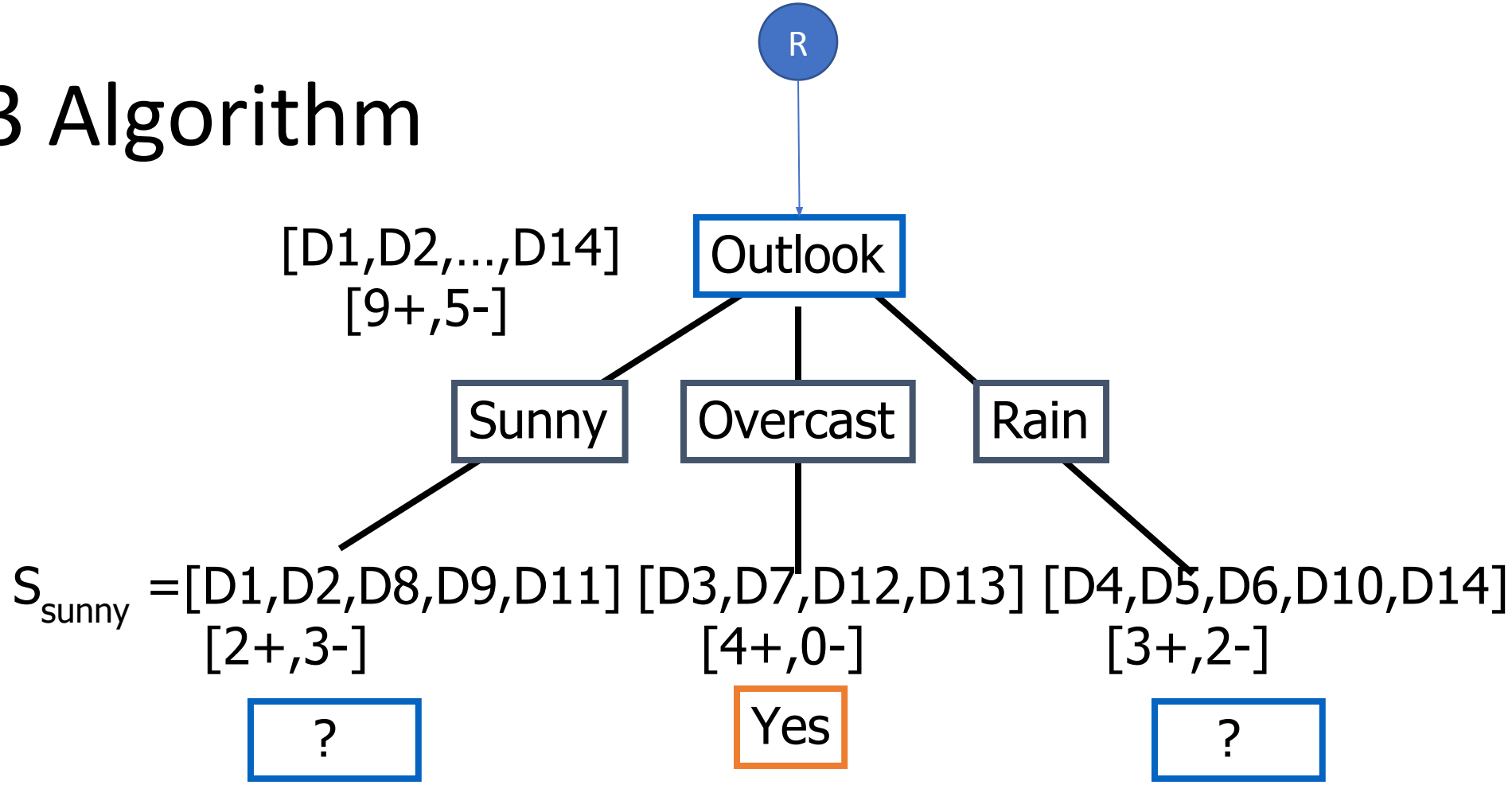
Selecting the Next Attribute

The information gain values for the 4 attributes are:

- Gain(S,Outlook) =0.247
- Gain(S,Humidity) =0.151
- Gain(S,Wind) =0.048
- Gain(S,Temperature) =0.029

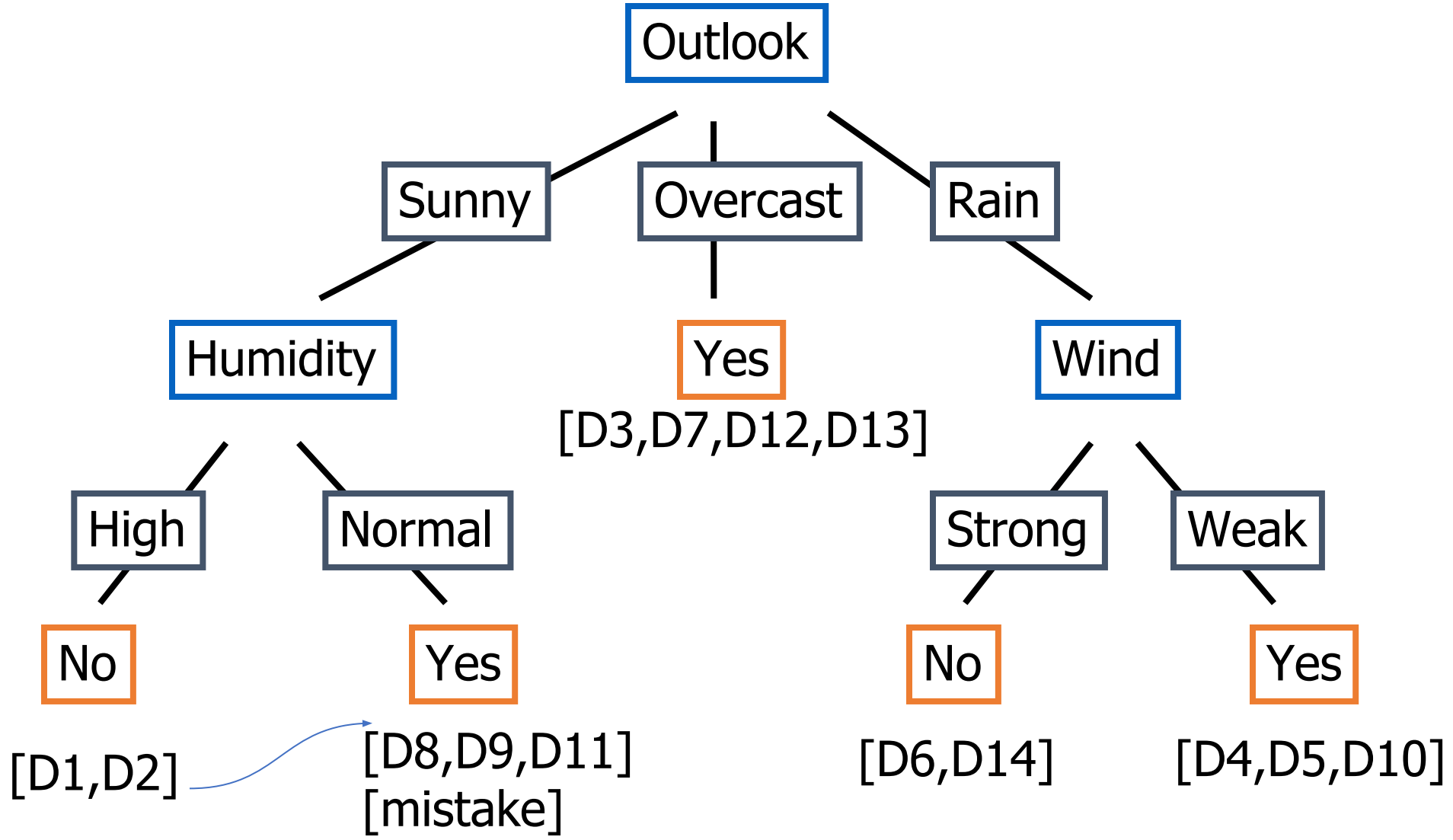
where S denotes the collection of training examples

ID3 Algorithm

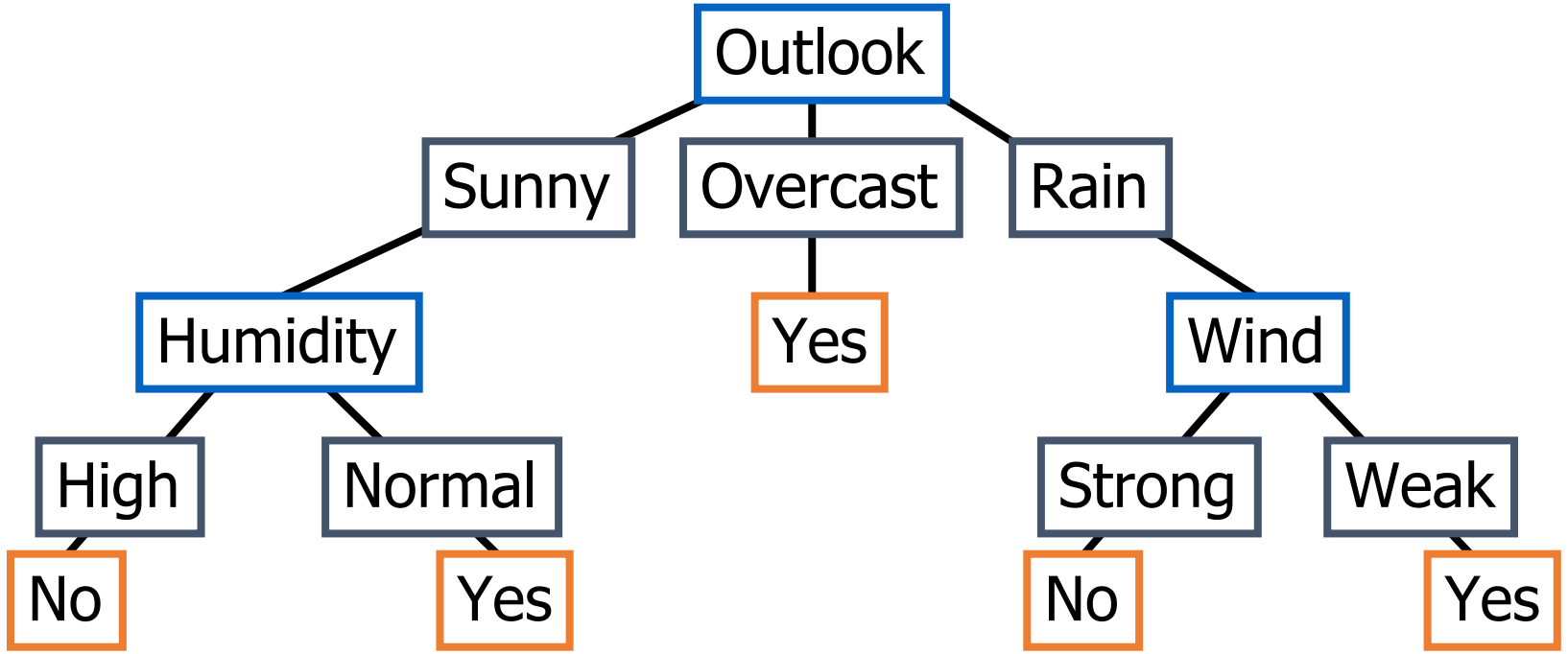


Gain(S_{sunny}, Humidity)=0.970-(3/5)0.0 – 2/5(0.0) = 0.970
Gain(S_{sunny}, Temp.)=0.970-(2/5)0.0 –2/5(1.0)-(1/5)0.0 = 0.570
Gain(S_{sunny}, Wind)=0.970= -(2/5)1.0 – 3/5(0.918) = 0.019

ID3 Algorithm



Converting a Tree to Rules



R_1: If (Outlook=Sunny) $\wedge$ (Humidity=High) Then PlayTennis=No
R_2: If (Outlook=Sunny) $\wedge$ (Humidity=Normal) Then PlayTennis=Yes
R_3: If (Outlook=Overcast) Then PlayTennis=Yes
R_4: If (Outlook=Rain) $\wedge$ (Wind=Strong) Then PlayTennis=No
R_5: If (Outlook=Rain) $\wedge$ (Wind=Weak) Then PlayTennis=Yes

Some Characteristics of Decision Trees

- Discrete values target function
- Suitable for discrete attribute values though continuous also possible
- Target function is expressed as disjunction of conjunction
- Searches the complete space of finite discrete valued function for correct hypothesis – decision trees
- However, results one single final hypothesis
- Greedy method for search, no backtrack. Solution may be sub-optimal
- Uses all training examples at each step
- Robust but may suffer overfiiting

Occam's Razor

”If two theories explain the facts equally well, then the simpler theory is to be preferred”

Arguments in favor:

- Fewer short hypotheses than long hypotheses
- A short hypothesis that fits the data is unlikely to be a coincidence
- A long hypothesis that fits the data might be a coincidence

Arguments opposed:

- There are many ways to define small sets of hypotheses

- Supports Inductive bias – after training, the learner justifies future classifications due to generalization, by choosing shortest tree