Introduction to Machine Learning

Lecture 05: Decision Tree

April 16, 2021

Jie Wang

Machine Intelligence Research and Applications Lab

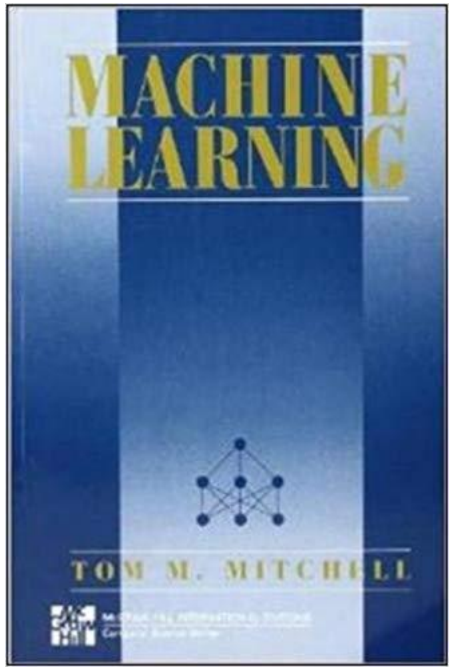
Department of Electronic Engineering and Information Science (EEIS)

http://staff.ustc.edu.cn/~jwangx/

jiewangx@ustc.edu.cn

Contents

- **Example**
- **ID3**
- **Extensions of ID3**



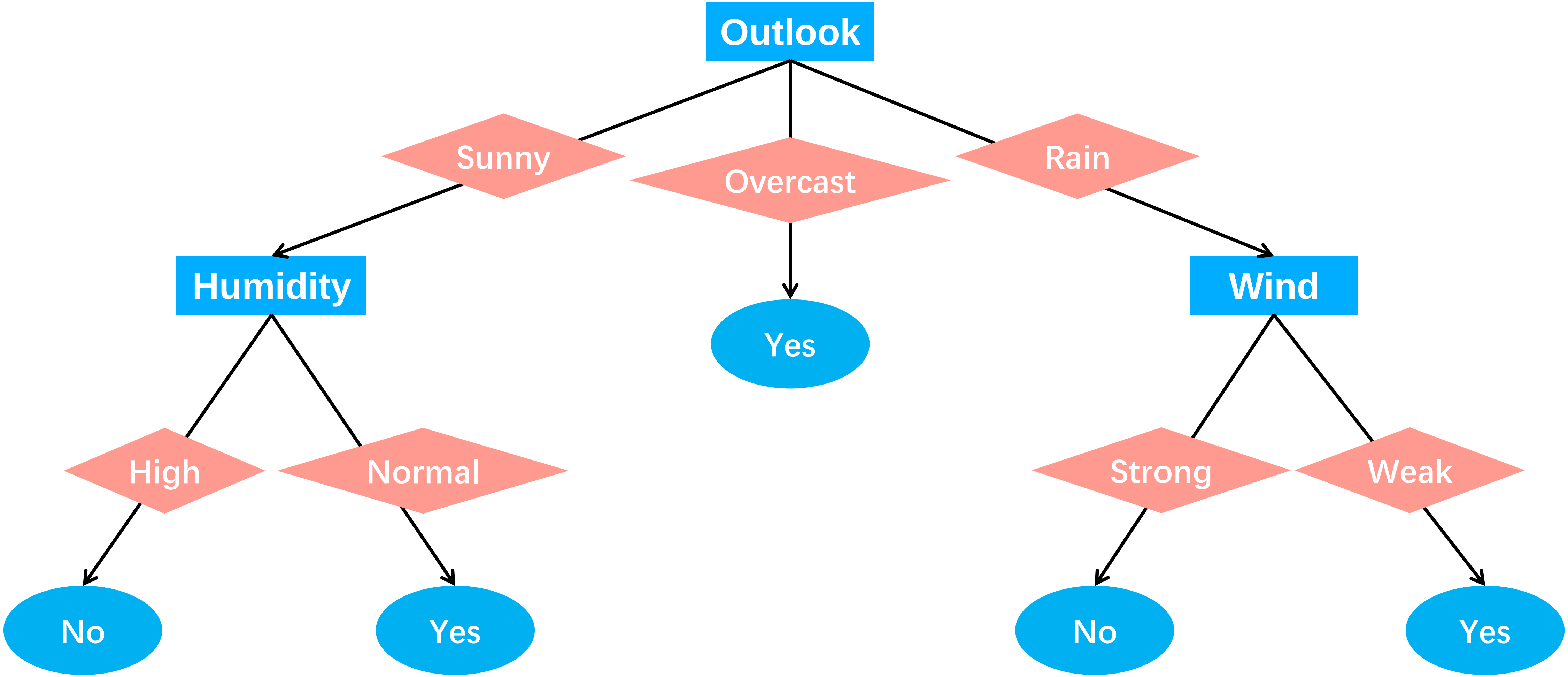
Chapter 3

- **Example**

Example

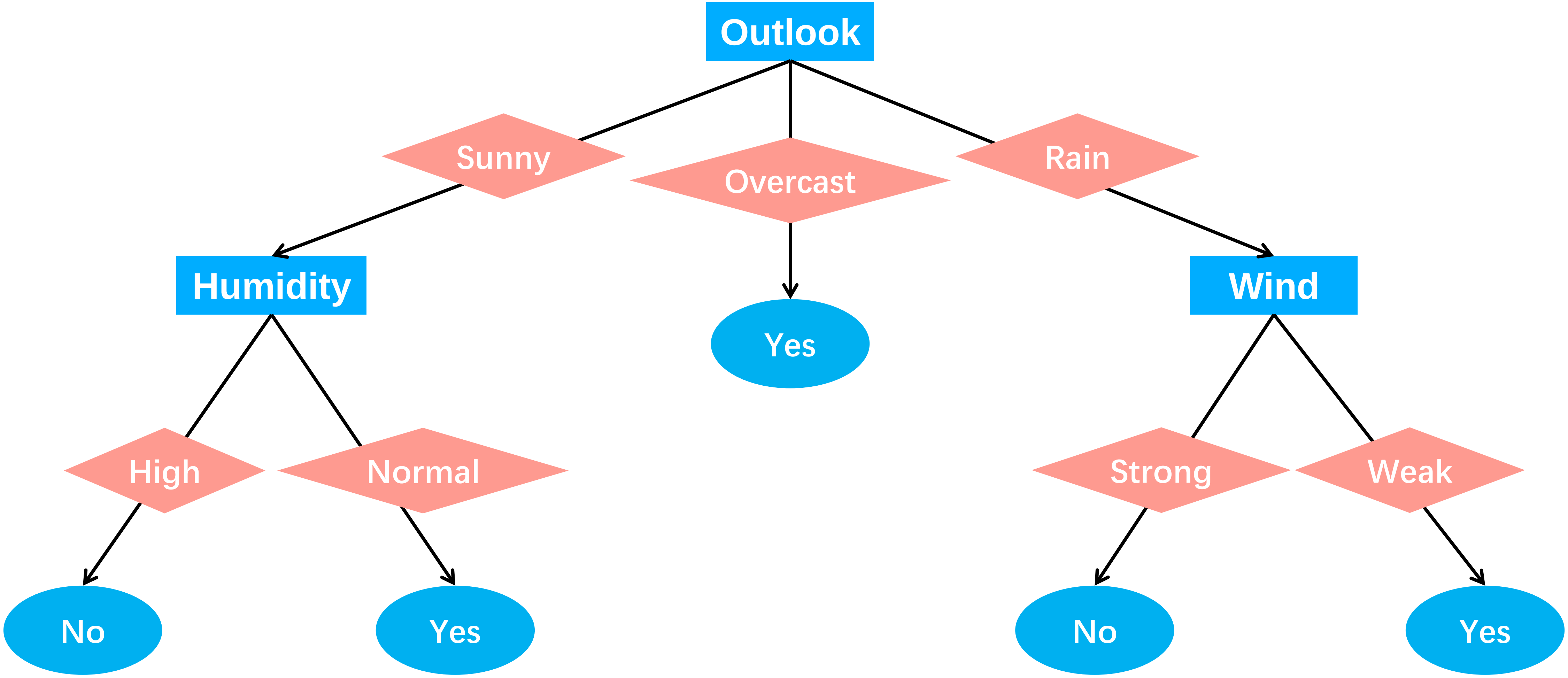
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	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


Outlook
Sunny
Overcast
Rain
Humidity
Yes
Wind
High
Normal
Strong
Weak
No
Yes
No
Yes

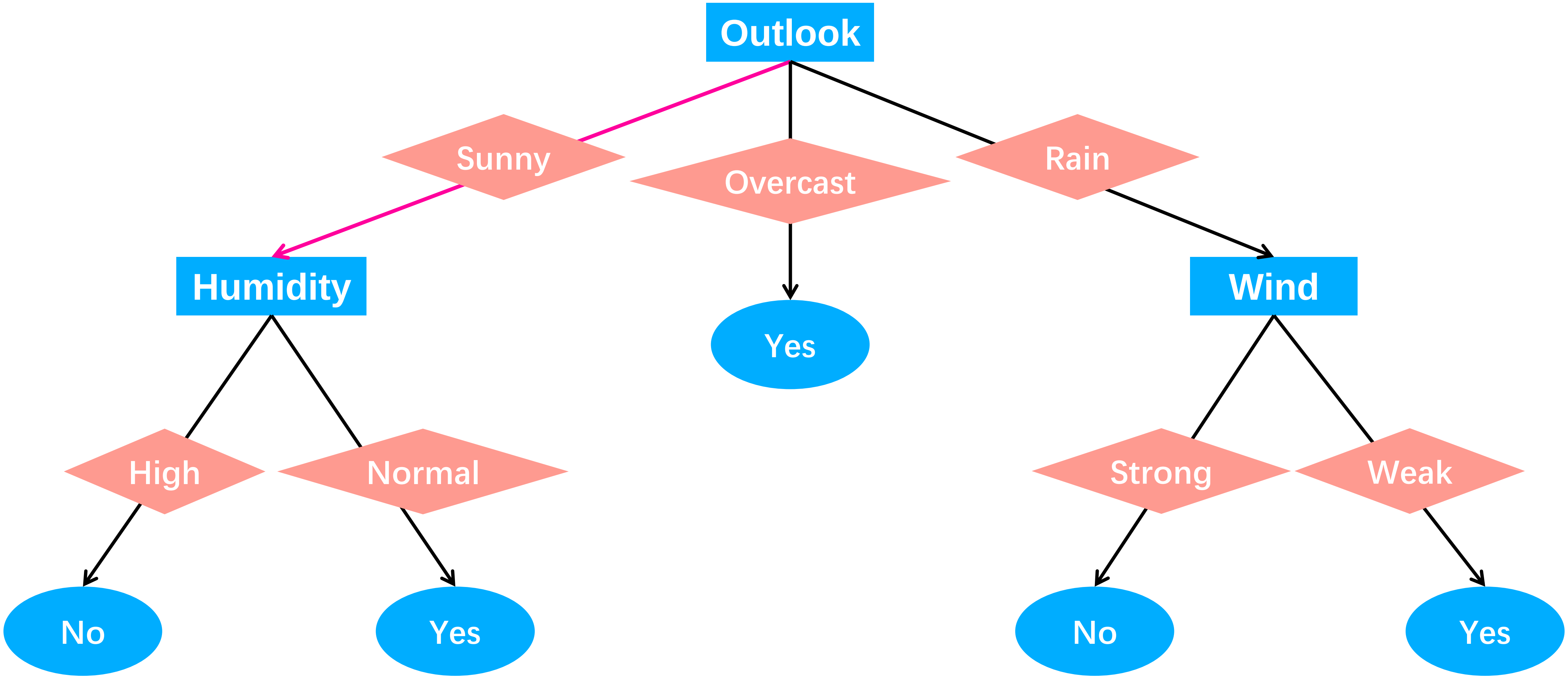
Example

{Outlook=Sunny, Temperature=Hot, Humidity=High, Wind=Strong}



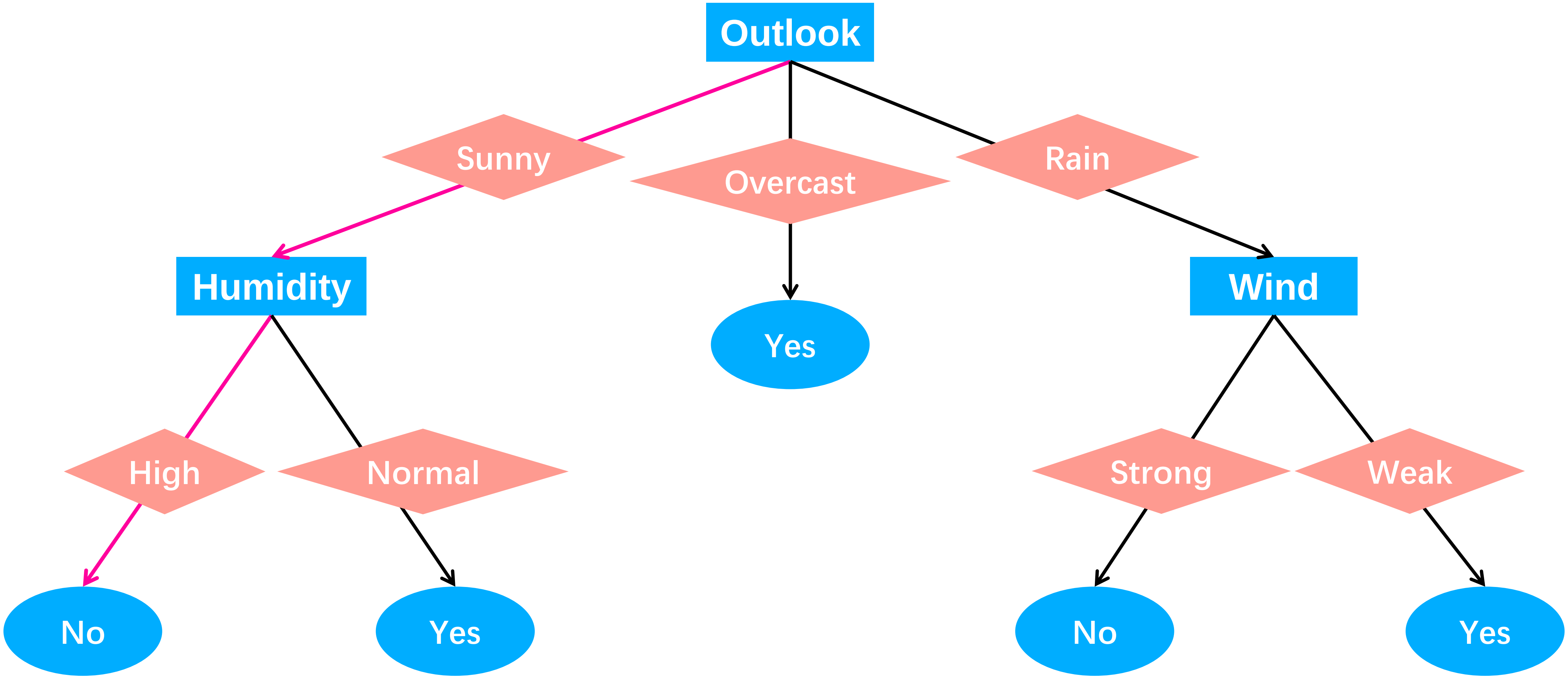
Example

{Outlook=Sunny, Temperature=Hot, Humidity=High, Wind=Strong}



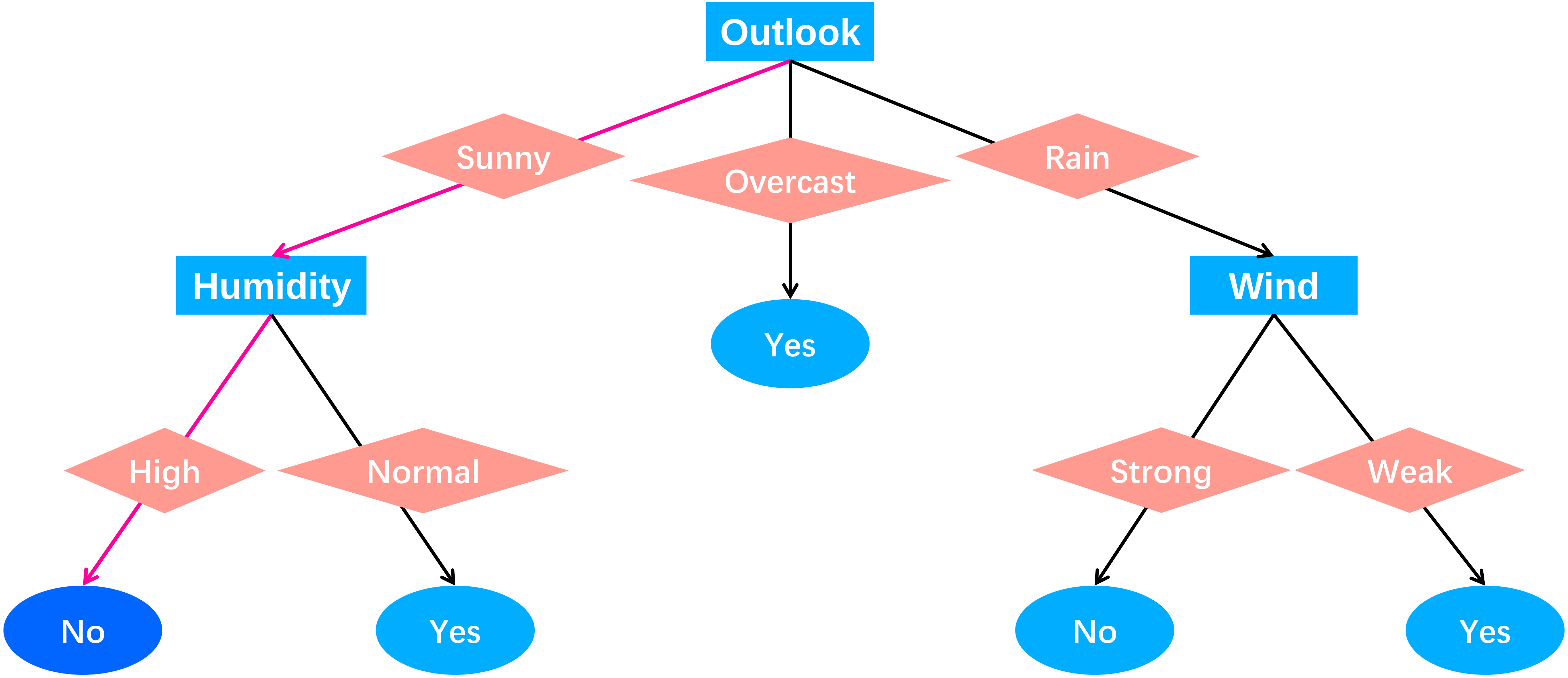
Example

{Outlook=Sunny, Temperature=Hot, Humidity=High, Wind=Strong}



Example

{Outlook=Sunny, Temperature=Hot, Humidity=High, Wind=Strong}



Appropriate Problems

- Each attribute takes on a small number of disjoint possible values.
- The target function has discrete output values (classification).
- The training data may contain missing attribute values.
- ……

- **ID3**

ID3

Which Attribute is the best classifier?

ID3

Which Attribute is the best classifier?

Information gain

measures how well a given attribute separates the training examples according to their target classification

ID3

- Which Attribute is the best classifier?

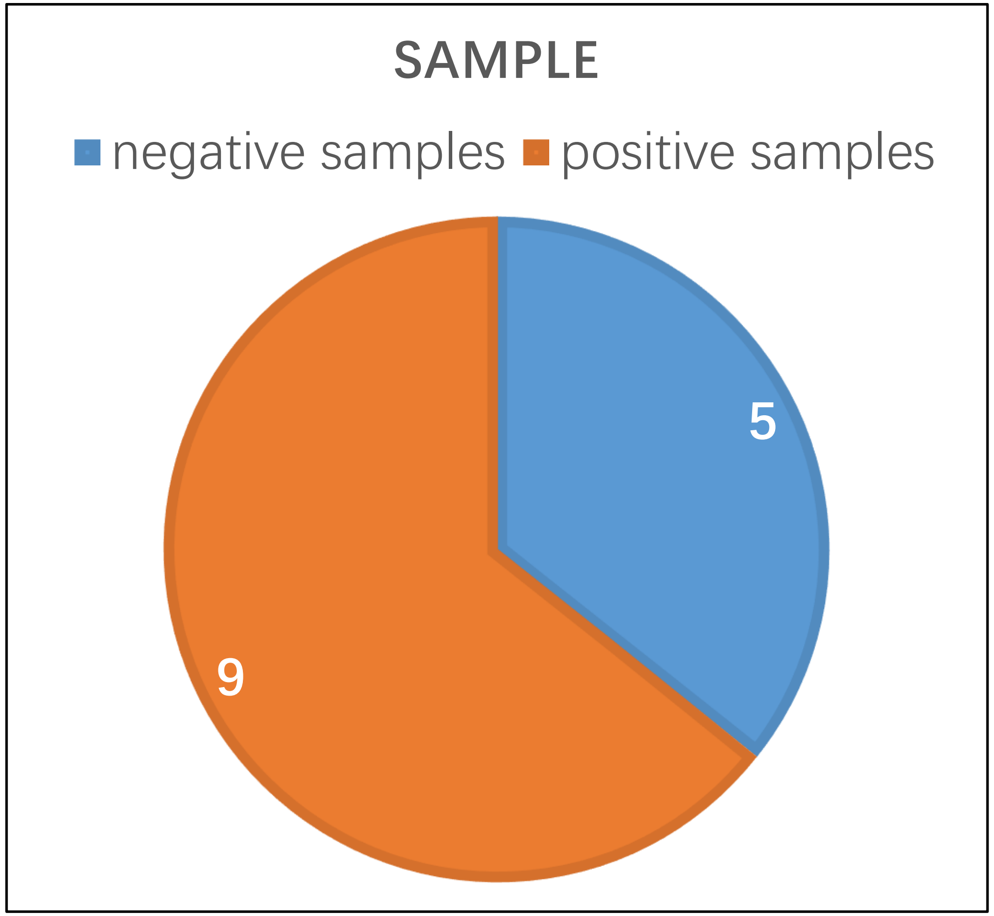
Information gain: measures how well a given attribute separates the training examples according to their target classification

↑

Entropy: measures the impurity of an arbitrary collection of data instances

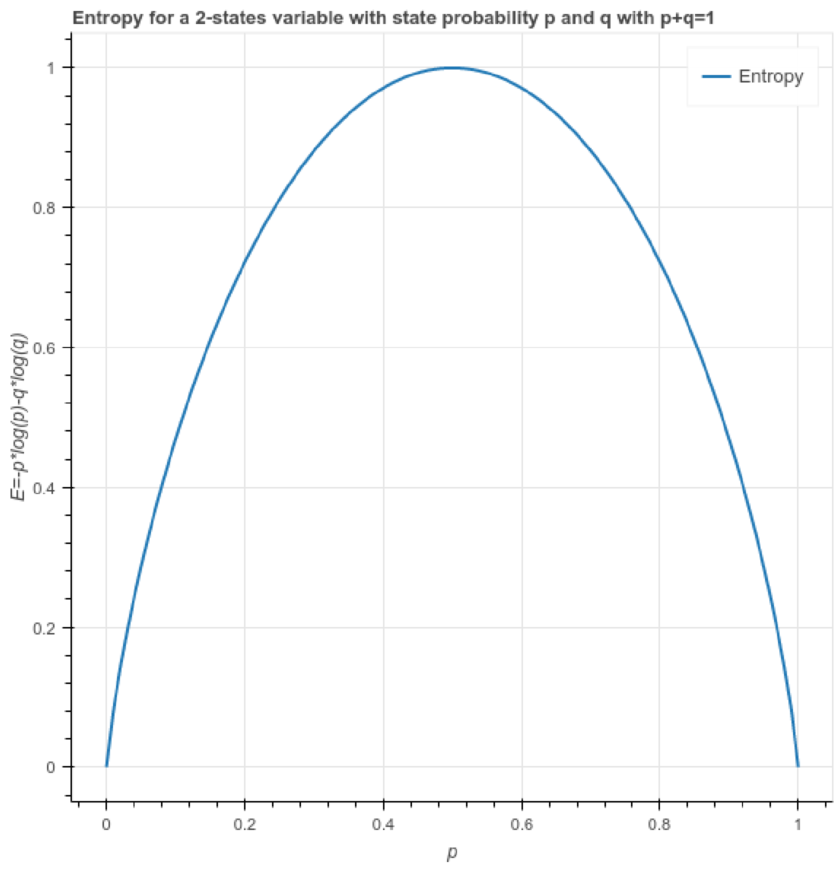
Entropy

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



$$Entropy([9+, 5-])$$
$$= -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$$
$$= 0.94$$

Entropy



- The entropy is 0 if all members of S belong to the same class.
- The entropy is 1 when S contains an equal number of positive and negative examples.

https://bricaud.github.io/personal-blog/entropy-in-decision-trees/

Information Gain

$$Gain(S, A) := Entropy(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

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

Information Gain

$$Gain(S, A) := Entropy(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

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

$$Values(Wind) = \{Weak, Strong\}$$

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

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

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

Information Gain

$$Gain(S, A) := Entropy(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

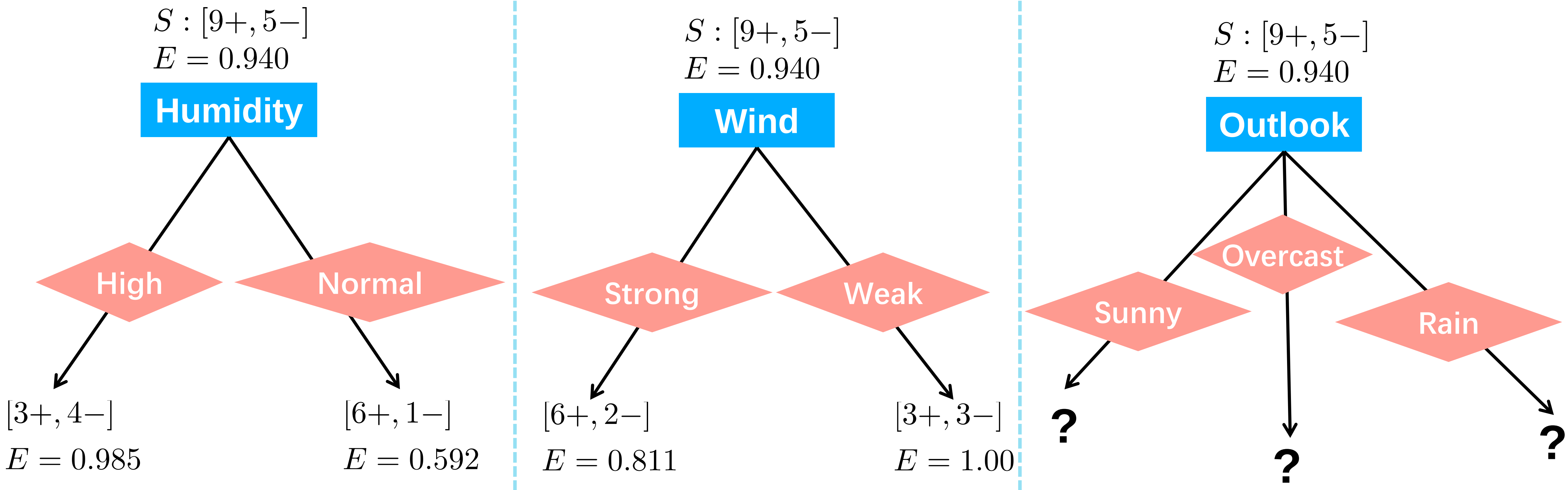
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	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

$$\begin{aligned} Values(Wind) &= \{Weak, Strong\} \\ S &= [9+, 5-] \\ S_{Weak} &\leftarrow [6+, 2-] \\ S_{Strong} &\leftarrow [3+, 3-] \end{aligned}$$

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

Information Gain

- Which Attribute is the best classifier?



$Gain(S, Humidity)$
$=0.94 - (7/14)0.985 - (7/14)0.592$
$=0.151$

$Gain(S, Wind)$
$=0.94 - (8/14)0.811 - (6/14)1.0$
$=0.048$

$Gain(S, Outlook) =$?

Information Gain

predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples.

- Create a *Root* node for the tree
- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = −
- If *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target_attribute* in *Examples*
- Otherwise Begin
    - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
    - The decision attribute for *Root* $\leftarrow A$
    - For each possible value, v_i, of A,
        - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
        - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A
        - If $Examples_{v_i}$ is empty
            - Then below this new branch add a leaf node with label = most common value of *Target_attribute* in *Examples*
            - Else below this new branch add the subtree ID3($Examples_{v_i}$, *Target_attribute*, *Attributes* − {A}))
- End
- Return *Root*

* The best attribute is the one with highest *information gain*, as defined in Equation (3.4).

TABLE 3.1

Summary of the ID3 algorithm specialized to learning boolean-valued functions. ID3 is a greedy

Pruning

- Overfitting

reasonable strategy, in fact it can lead to difficulties when there
or when the number of training examples is too small to prodı
sample of the true target function. In either of these cases, th
can produce trees that *overfit* the training examples.

We will say that a hypothesis overfits the training exan
hypothesis that fits the training examples less well actually perf
entire distribution of instances (i.e., including instances beyor

Definition: Given a hypothesis space H, a hypothesis $h \in H$
training data if there exists some alternative hypothesis $h' \in$
smaller error than h' over the training examples, but h' has a

Pruning

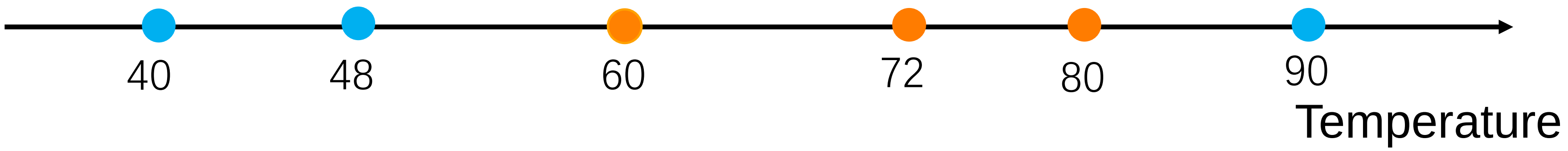
- Post-pruning
  - Split the data into a training set and a validation set
  - Train the decision tree on the training set
  - **While** pruning improves the accuracy of the tree on the validation set
    - Scan the nodes one by one
    - **If** removing the nodes (and all its descendants) improves the accuracy of the tree on the validation set
      - Remove the node and all its descendants
    - **Endif**

original over the validation set. This has the effect that any l
to coincidental regularities in the training set is likely to be p
same coincidences are unlikely to occur in the validation set
iteratively, always choosing the node whose removal most in
tree accuracy over the validation set. Pruning of nodes cor
pruning is harmful (i.e., decreases accuracy of the tree over t

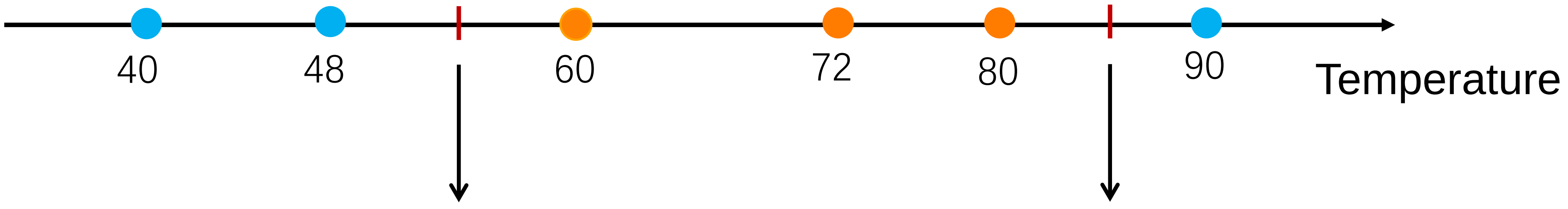
The impact of reduced-error pruning on the accuracy
is illustrated in Figure 3.7. As in Figure 3.6, the accuracy c
measured over both training examples and test examples. Th
Figure 3.7 shows accuracy over the test examples as the tre
pruning begins, the tree is at its maximum size and lowest ac

- **Extensions of ID3**

Continuous-Valued Attributes



Continuous-Valued Attributes



$$\frac{48+60}{2}$$

$$\frac{80+90}{2}$$

Missing Attribute Values

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	?	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

- Approach 1
  - Assign the common value to the missing attribute value

Missing Attribute Values

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	**?**	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

- Approach 1
  - Assign the common value to the missing attribute value
- Approach 2
  - Weight the instance by the frequencies of the attribute values

Missing Attribute Values

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	?	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

- Approach 1
  - Assign the common value to the missing attribute value
- Approach 2
  - Weight the instance by the frequencies of the attribute values

D6	?	Cool	Normal	Strong	No

Missing Attribute Values

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	?	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

- Approach 1
  - Assign the common value to the missing attribute value
- Approach 2
  - Weight the instance by the frequencies of the attribute values

D6	?	Cool	Normal	Strong	No

5/13	D6-1	Sunny	Cool	Normal	Strong	No
4/13	D6-2	Overcast	Cool	Normal	Strong	No
4/13	D6-3	Rain	Cool	Normal	Strong	No

Questions