Engaging Content
Engaging People

Decision Trees / NLP Introduction

Dr. Kevin Koidl
School of Computer Science and Statistic Trinity College Dublin
ADAPT Research Centre

Ireland's European Structural and Investment Funds Programmes 2014-2020
Co-funded by the Irish Government and the European Union
European Union
European Regional Development Fund
SFI Science Foundation Ireland

The ADAPT Centre is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

What is Decision Tree learning?

Decision tree learning is widely used for inductive inference.

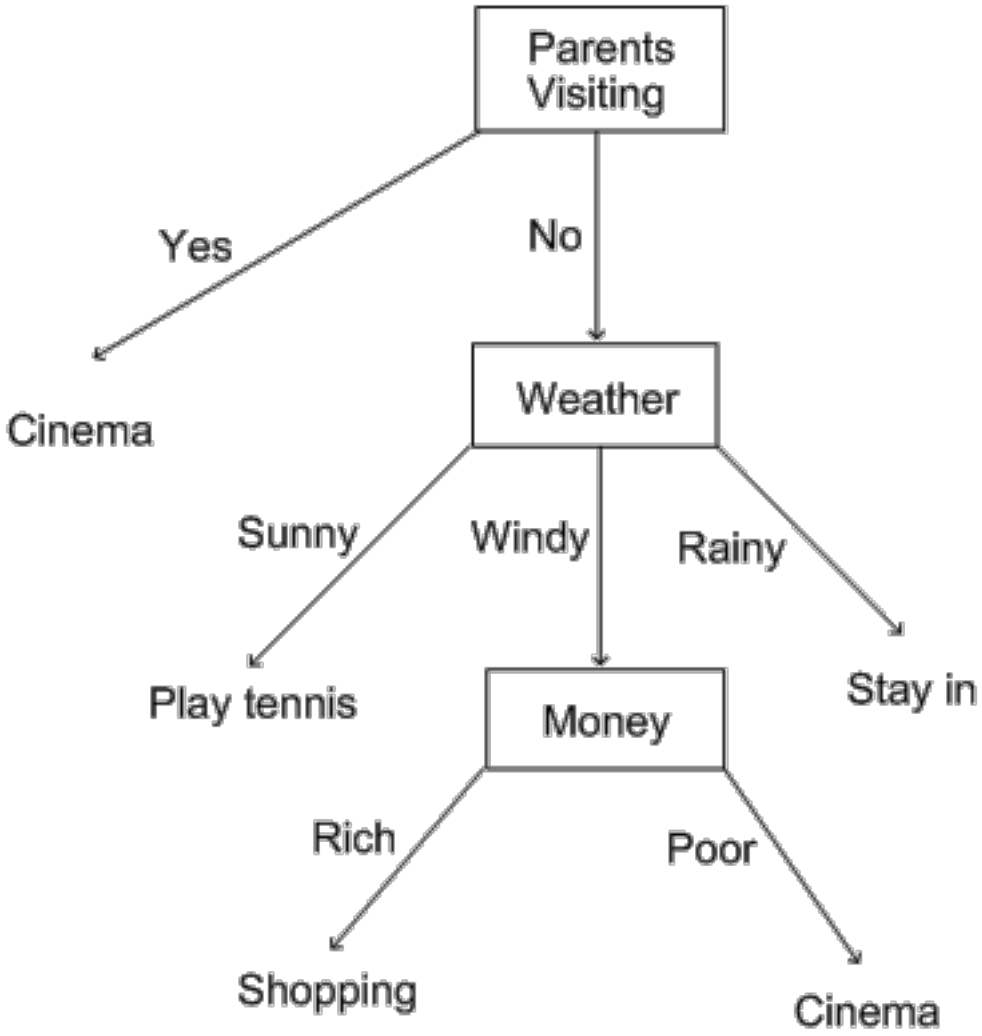
Used for approximating discrete-valued functions.

Every (terminating) algorithmic decision process can be modelled as a tree.

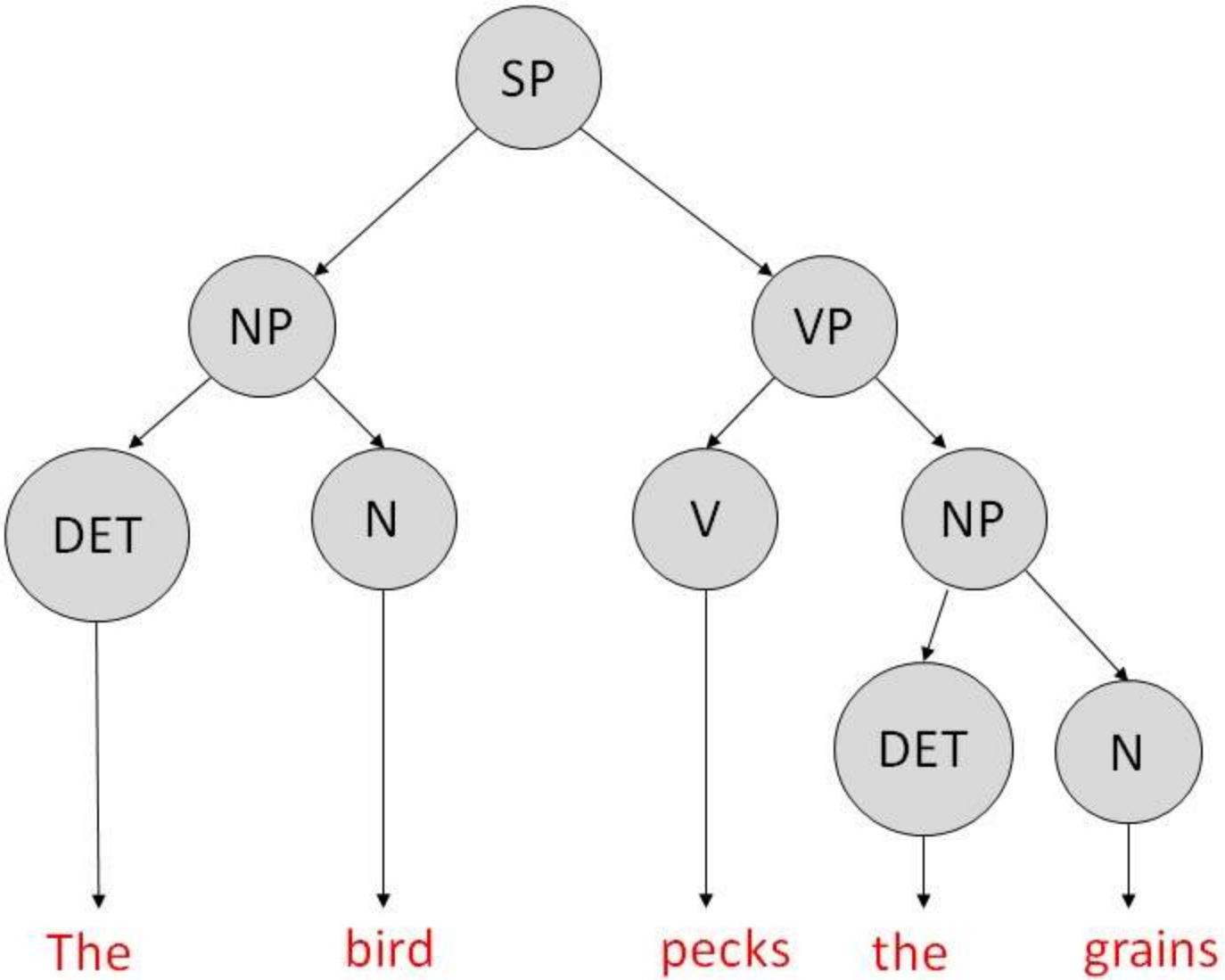
Typical decision tree learning algorithms includes ID3, ASSISTANT, and C4.5.

Each methods searches a completely expressive hypothesis space.

Example of a Tree



Example of a Tree



Typical Problems Solved with Decision Trees

Most Rule based systems (note: Knowledge Acquisition Bottleneck)

- Equipment diagnosis
- Medical diagnosis
- Credit card risk analysis
- Robot movement
- Pattern Recognition
- Face recognition
- Missing word problem (chat bots)
- Others?

Decision Tree Representation

Decision trees **classify instances** by sorting top down.

A **leaf** provides the classification of the instance.

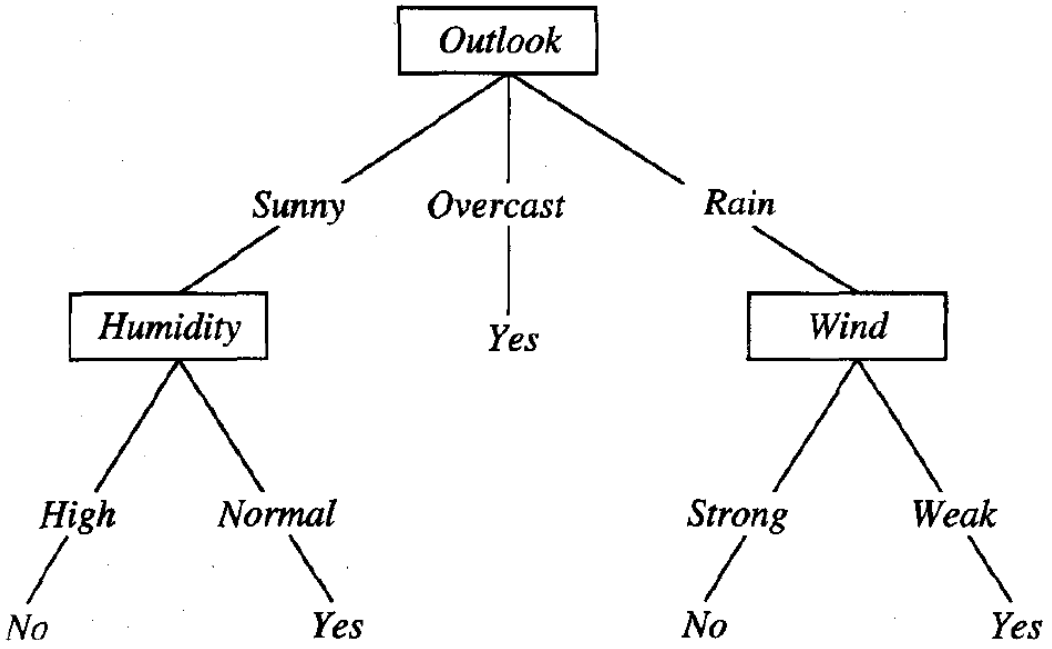
A **node** specifies a test of some attribute of the instance.

A **branch** corresponds to a possible values an attribute.

An **instance** is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute in the given example.

This **process** is then repeated for the subtree rooted at the new node.

Decision Tree Representation (Classification)



This tree classifies Saturday mornings according to whether or not they are suitable for playing tennis.

Prediction: Play Tennis?

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

Decision Tree Representation

This decision tree classifies Saturday mornings according to whether they are suitable for playing tennis.

$$\langle Outlook = Sunny,\ Temperature = Hot,\ Humidity = High,\ Wind = Strong \rangle$$

Example instance gets sorted down the leftmost branch of this decision tree and classified a a negative instance (i.e., the tree predicts that PlayTennis = no).

Follows logical AND / OR Structure.

$$\begin{aligned} & (Outlook = Sunny \ \wedge \ Humidity = Normal) \\ \vee \quad & (Outlook = Overcast) \\ \vee \quad & (Outlook = Rain \ \wedge \ Wind = Weak) \end{aligned}$$

ID3 Algorithm

Most algorithms that have been developed for **learning decision trees** are variations on a core algorithm that employs a **top-down, greedy search** through the space of possible **decision trees** (Quinlan 1986) and its successor C4.5 (Quinlan 1993).

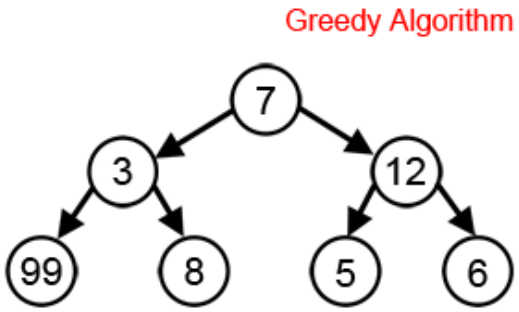
ID3, learns decision trees by constructing them **top-down**, beginning with the question "which attribute should be tested at the root of the tree?”

What is Greedy Search?

At each step, make decision which makes greatest improvement in whatever you are trying optimize.

Does not backtrack (unless you hit a dead end)

This type of search is likely not to be a globally optimum solution, but generally works well.



At each node of tree, make decision on which attribute best classifies training data at that point.

End result will be tree structure representing a hypothesis, which works best for the training data.

ID3 Algorithm

1. Each instance attribute is evaluated using a statistical test to determine how well it alone classifies the training examples.
2. The best attribute is selected and used as the test at the root node of the tree.
3. A descendant of the root node is then created for each possible value of this attribute.
4. Training examples are sorted to the appropriate descendant node.
5. The entire process is then repeated using the training examples associated with each descendant node to select the best attribute to test at that point in the tree.
6. This forms a greedy search for an acceptable decision tree, in which the algorithm never backtracks to reconsider earlier choices.

Which Attribute Is the Best Classifier?

The ID3 algorithm selects, which attribute to test at each node in the tree.

We would like to select the attribute that is most useful for classifying examples.

What is a good quantitative measure of the worth of an attribute?

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

ID3 uses this **information gain** measure to select among the candidate attributes at each step while growing the tree.

Information Gain is based on Entropy

Given a collection S, containing positive and negative examples of some target concept, the entropy of S relative to this Boolean classification is:

$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Where p(+), is the proportion of positive examples in S and p(-), is the proportion of negative examples in S. In all calculations involving entropy we define 0 log 0 to be 0.

Example - Entropy

S is a collection of 14 examples of a Boolean concept, including 9 positive and 5 negative examples [9+, 5].

Then the entropy of S relative to this Boolean classification is:

$$Entropy([9+, 5-]) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$$
$$= 0.940$$

Entropy is 0 if all members of S belong to the same class.

If all members positive p(+) = 1, then p(-) = 0, and

Entropy(S) = -1 * log2(1) - 0 * log2 0 = -1 * 0 - 0 * log2 0 = 0.

Prediction: Play Tennis?

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

Entropy

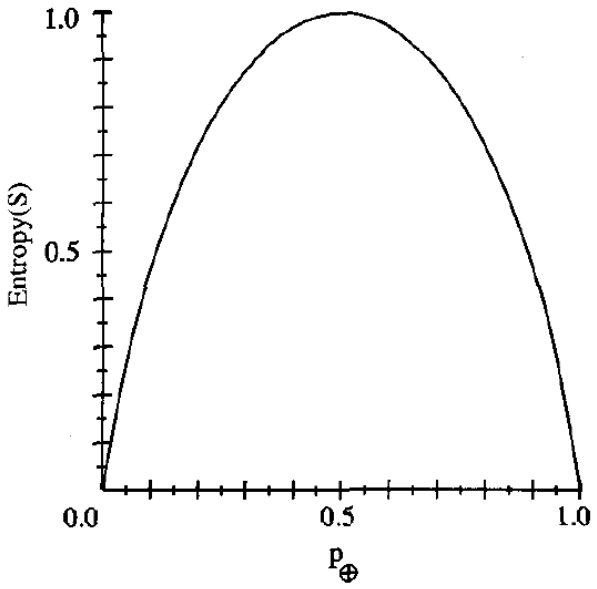


FIGURE 3.2
The entropy function relative to a boolean classification, as the proportion, $p_\oplus$, of positive examples varies between 0 and 1.

$$Entropy(S) \equiv -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$

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

Information Gain – Expected reduction in Entropy

Expected reduction in entropy caused by partitioning the examples according to this attribute.

Gain (S,A) of an attribute A relative to a collection of examples S is defined as:

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

Values(A) is the set of positive values for attribute A.

S_v is the subset of S for which attribute A has value v.

Information Gain Explained

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

The expected entropy described by the second term is the sum of the entropies of each subset S, weighted by the fraction of examples that belong to S.

Gain(S, A) returns the expected reduction in entropy caused by knowing the value of attribute A.

Gain(S, A) is the information provided about the target function value, given the value of some other attribute A.

Prediction: Play Tennis?

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 - Entropy

S is a collection of 14 examples of a Boolean concept, including 9 positive and 5 negative examples [9+, 5].

Then the entropy of S relative to this Boolean classification is:

$$Entropy([9+, 5-]) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$$
$$= 0.940$$

Entropy is 0 if all members of S belong to the same class.

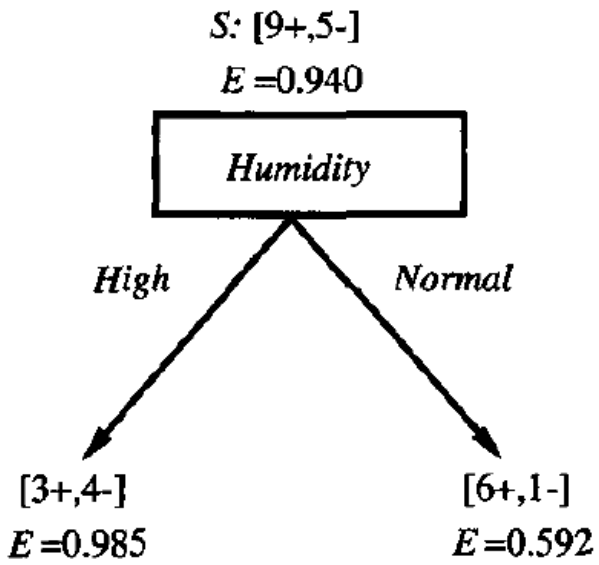
If all members positive p(+) = 1, then p(-) = 0, and

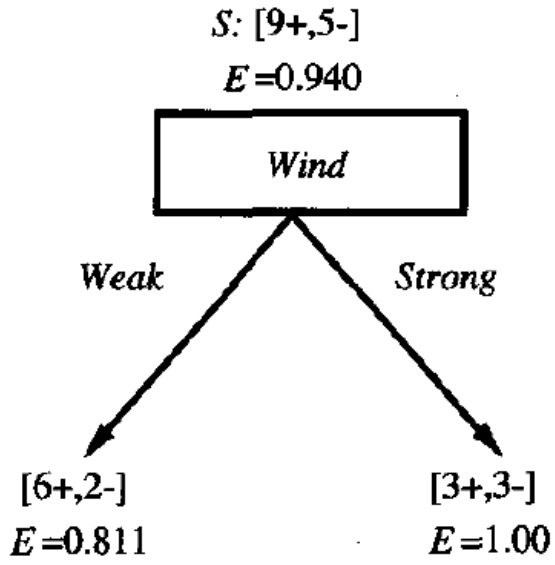
Entropy(S) = -1 * log2(1) - 0 * log2 0 = -1 * 0 - 0 * log2 0 = 0.

Example Calculation

$$
\begin{aligned}
Values(Wind) &= Weak, Strong \\
S &= [9+, 5-] \\
S_{Weak} &\leftarrow [6+, 2-] \\
S_{Strong} &\leftarrow [3+, 3-] \\
Gain(S, Wind) &= Entropy(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} Entropy(S_v) \\
&= Entropy(S) - (8/14) Entropy(S_{Weak}) \\
&\quad - (6/14) Entropy(S_{Strong}) \\
&= 0.940 - (8/14)0.811 - (6/14)1.00 \\
&= 0.048
\end{aligned}
$$

Selection of attribute as best classifier





Gain (S, Humidity)

= .940 - (7/14).985 - (7/14).592
= .151

Gain (S, Wind)

= .940 - (8/14).811 - (6/14)1.0
= .048

Which attribute should be tested first in the tree?

ID3 determines the information gain for each candidate attribute (i.e., Outlook, Temperature, Humidity, and Wind), then selects the one with highest information gain. S denotes the collection of training examples.

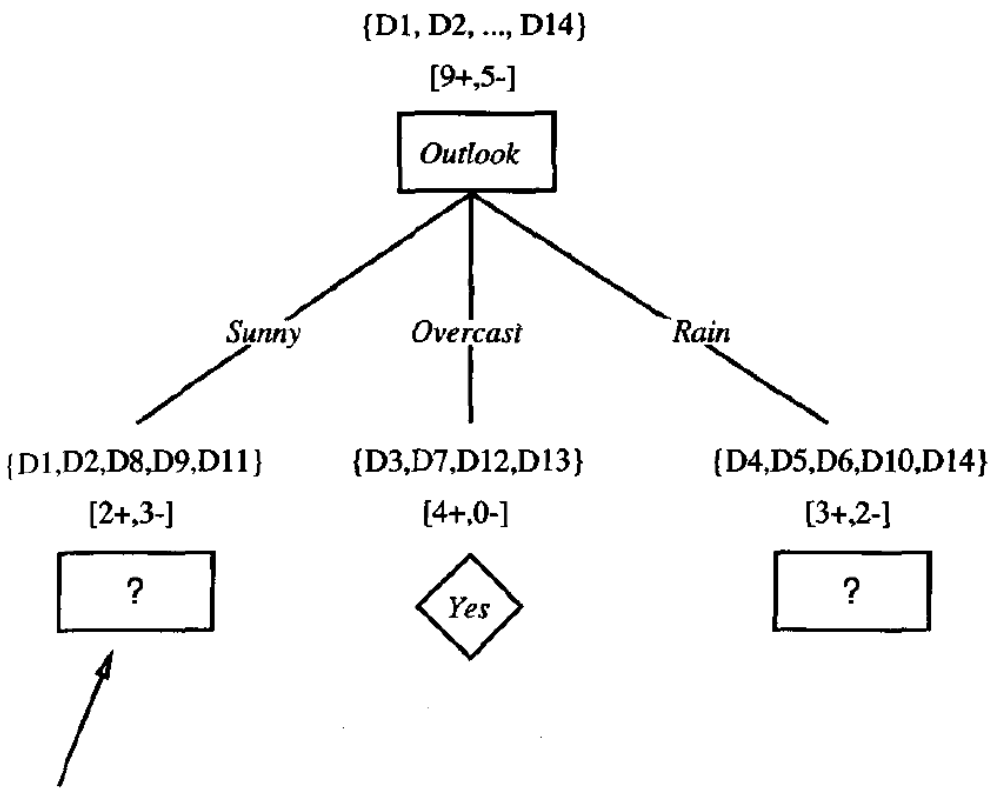
$$Gain(S, Outlook) = 0.246$$
$$Gain(S, Humidity) = 0.151$$
$$Gain(S, Wind) = 0.048$$
$$Gain(S, Temperature) = 0.029$$

Outlook is selected as the decision attribute for the root node, and branches are created below the root for each of its possible values (i.e., Sunny, Overcast, and Rain).

Prediction: Play Tennis?

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

Result



Which attribute should be tested here?

S_{sunny} = {D1,D2,D8,D9,D11}

Gain (S_{sunny}, *Humidity*) = .970 - (3/5) 0.0 - (2/5) 0.0 = .970

Gain (S_{sunny}, *Temperature*) = .970 - (2/5) 0.0 - (2/5) 1.0 - (1/5) 0.0 = .570

Gain (S_{sunny}, *Wind*) = .970 - (2/5) 1.0 - (3/5) .918 = .019

ID3 Algorithm

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. Target_attribute is the attribute whose value is to be 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*