1 What is a Decision Tree?

An alternative to traditional rule based systems, in which a tree contains the actual rules. An example is shown in Figure 1.

![Decision Tree Diagram]

Figure 1: A decision tree used to determine whether it is suitable to play tennis.

2 Decision Tree Representation

Decision trees classify instances by following the tree from the root to a leaf which provides the classification.
<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 1: The training set for the decision tree of the PlayTennis example.

- A node specifies a test for some attribute.
- Branches starting from a node, correspond to the possible values of the attribute.
- Leaves correspond to the classification result.

3 Decision Trees and Rules

A decision tree represents a disjunction of conjunctions of constraints on the attribute values of instances.

The decision tree of Figure 1 is equivalent to the rule:

(Outlook = sunny AND Humidity = normal) OR
(Outlook = overcast) OR
(Outlook = rain AND Wind = weak)

4 Learning Decision Trees

Decision trees are built based on a given training set, as that shown in Table 1.

5 The ID3 Algorithm

The ID3 algorithm can be used to build (learn) a decision tree.

The following pseudocode illustrates how the ID3 algorithm works:

ID3(Examples, Target_attribute, Attributes)
• Create a Root node for the tree.
• If all Examples are positive, return the single node Root with label = +.
• If all Examples are negative, return the single node Root with label = −.
• If all Attributes are empty, return the single node Root with label = (the most common value of Target_attribute in Examples).
• Otherwise Begin
  – \( A \) = the attribute that best classifies Examples (the attribute with the highest information gain).
  – Root = \( A \).
  – For each possible value \( v_i \) of \( A \):
    1. Add a new tree branch under Root, corresponding to the test \( A = v_i \)
    2. Let \( \text{Examples}_{v_i} \) be the subset of Examples that have value \( v_i \) for \( A \).
    3. If \( \text{Examples}_{v_i} \) is empty
      * Then below this new branch add a leaf node with label = (the most common value of Target_attribute in Examples).
      * Else below this new branch add the subtree ID3(\( \text{Examples}_{v_i} \), Target_attribute, Attributes - \{A\})
• End
• Return Root

6 The Entropy Measure

Entropy is a measure in information theory which characterises the (im)purity of a collection of examples.

Given a set \( S \) containing positive and negative examples, the entropy is defined as:

\[
\text{Entropy}(S) = -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus
\]

where \( p_\oplus, p_\ominus \) the probabilities of finding a positive or a negative example in \( S \) respectively.

Generally, if the target attribute can take \( n \) different values then:

\[
\text{Entropy}(S) = \sum_{i=1}^{n} -p_i \log_2 p_i
\]

Entropy Example:

Assume a set \( S \) of 14 examples which includes 9 positive and 5 negative examples.

Then:

\[
\text{Entropy}(9+, 5-) = -\frac{9}{14} \cdot \log_2 \left(\frac{9}{14}\right) - \frac{5}{14} \cdot \log_2 \left(\frac{5}{14}\right) = 0.94
\]

Note that the entropy is 0, is all the examples in a set belong to the same class (either all of them are positive or all of them are negative).
One interpretation of entropy in information theory is that it determines the minimum number of bits which encodes the classification of a member of $S$. If $p_\Theta = 1$, the receiver knows that the example will be positive so no message needs to be sent.

7 Information Gain

The information gain measures the expected reduction in entropy, or otherwise the effectiveness of an attribute in classifying the training data.

The information gain of an attribute $A$ in a collection of examples $S$ is defined as:

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

where $|S_v|$ the number of examples in $S$ for which the $A = v$.

Example:

Consider the set $S$ of Table 1 where from the 14 examples, 6 of the positive and 2 of the negative examples have $\text{Wind} = \text{weak}$, and the remainder have $\text{Wind} = \text{strong}$.

Then:

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

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

$$S_{\text{Weak}} ← [6+, 2-]$$

$$S_{\text{Strong}} ← [3+, 3-]$$

$$\text{Gain}(S, \text{Wind}) = \text{Entropy}(S) - \sum_{v \in \text{Weak, Strong}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$= \text{Entropy}(S) - \frac{8}{14} \cdot \text{Entropy}(S_{\text{Weak}}) - \frac{6}{14} \cdot \text{Entropy}(S_{\text{Strong}})$$

$$= 0.94 - \frac{8}{14} \cdot 0.811 - \frac{6}{14} \cdot 1.0$$

$$= 0.048 \quad (5)$$

Converting the base of a logarithm

The following formula can be used to calculate the $\log_2$ of a number based on the $\log_{10}$ of the number:

$$\log_2 x = \frac{\log_{10} x}{\log_{10} 2} \quad (6)$$

or more generally:

$$\log_a x = \frac{\log_b x}{\log_b a} \quad (7)$$

where $a \neq 1$, $b \neq 1$ and $x > 0$. 
8 ID3 Example

To build the decision tree for the training data of Table 1, the information gain of all the attributes must be calculated, in order to determine the root of the tree.

\[
\begin{align*}
\text{Gain}(S, \text{Outlook}) &= 0.246 \\
\text{Gain}(S, \text{Humidity}) &= 0.151 \\
\text{Gain}(S, \text{Wind}) &= 0.048 \\
\text{Gain}(S, \text{Temperature}) &= 0.029
\end{align*}
\] (8)

Therefore \textit{Outlook} is the root of the tree as it has the highest information gain, and the first level of the decision tree is shown in Figure 2.

Figure 2: The decision tree for the \textit{PlayTennis} example after the application of the first step of ID3.

The process of selecting a new attribute and partitioning the training examples is now repeated for each nonterminal descendant node, using only the examples associated with the node. For example, for the left node of Figure 2, only examples 1, 2, 8, 9, 11 are considered. This process is repeated until:

1. every attribute has been included along this path of the tree, or
2. The training examples associated with the leaf node all have the same target attribute value (i.e. they are either all positive or all negative)

9 Continuous valued Attributes

The attributes of a decision tree can be continuous (non-discrete).
Continuous values of the decision node attributes can be discretised by defining new discrete-values attributes which partition the continuous attribute values into a discrete set of intervals.

For example, for a continuous valued attribute $A$, the ID3 algorithm can define a new boolean attribute $A_c$ that is true if $A < c$ and false otherwise.