Classification

Dr Ahmed Rafea
Decision Trees

• The Decision Tree is one of the most popular classification algorithms in current use in Data Mining and Machine Learning.

• A Decision Tree is a tree-structured plan of a set of attributes to test in order to predict the output.
Example: “Play/Don’t Play” Training Set

<table>
<thead>
<tr>
<th>No.</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>N</td>
</tr>
<tr>
<td>2</td>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>true</td>
<td>N</td>
</tr>
<tr>
<td>3</td>
<td>overcast</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>P</td>
</tr>
<tr>
<td>4</td>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>false</td>
<td>P</td>
</tr>
<tr>
<td>5</td>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>false</td>
<td>P</td>
</tr>
<tr>
<td>6</td>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>true</td>
<td>N</td>
</tr>
<tr>
<td>7</td>
<td>overcast</td>
<td>cool</td>
<td>normal</td>
<td>true</td>
<td>P</td>
</tr>
<tr>
<td>8</td>
<td>sunny</td>
<td>mild</td>
<td>high</td>
<td>false</td>
<td>N</td>
</tr>
<tr>
<td>9</td>
<td>sunny</td>
<td>cool</td>
<td>normal</td>
<td>false</td>
<td>P</td>
</tr>
<tr>
<td>10</td>
<td>rain</td>
<td>mild</td>
<td>normal</td>
<td>false</td>
<td>P</td>
</tr>
<tr>
<td>11</td>
<td>sunny</td>
<td>mild</td>
<td>normal</td>
<td>true</td>
<td>P</td>
</tr>
<tr>
<td>12</td>
<td>overcast</td>
<td>mild</td>
<td>high</td>
<td>true</td>
<td>P</td>
</tr>
<tr>
<td>13</td>
<td>overcast</td>
<td>hot</td>
<td>normal</td>
<td>false</td>
<td>P</td>
</tr>
<tr>
<td>14</td>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>true</td>
<td>N</td>
</tr>
</tbody>
</table>
Decision Tree Derived from Training Set

```
Decision Tree Derived from Training Set

outlook

sunny overcast rain

humidity

high normal

P N P

windy

true false

N P
```
ID3 Algorithm

• Given (1) a set of disjoint target classes \((C_1, C_2, \ldots, C_k)\), and (2) a set of training data, \(S\), containing objects of more than one class.

• ID3 uses a series of tests to refine \(S\) into subsets that contain objects of only one class.

• ID3 builds a decision tree, where non-terminal nodes correspond to tests on a single attribute of the data, and terminal nodes correspond to classified subsets of the data.

• Let \(T\) be any test on a single attribute. Thus \(T\) produces a partition \(\{S_1, S_2, \ldots, S_n\}\) based on outcome \(O_1, O_2, \ldots, O_n\).
Tree Structure of Partitioned Objects

$S$

$O_1$ $O_2$ $\ldots$ $O_n$

$S_1$ $S_2$ $\ldots$ $S_n$
Information Theory

• Consider a set of message \( M = \{m_1, m_2, \ldots, m_n\} \)
• Each message \( m_i \) has probability \( p(m_i) \) of being received and contains an amount of information \( I(m_i) \) as follows:
  \[
  I(m_i) = -\log_2 p(m_i)
  \]
• The uncertainty (or entropy) of a message set \( U(M) \) is the sum of information in the possible messages weighted by their probabilities:
  \[
  U(M) = -\sum_i p(m_i) \log_2 p(m_i) \quad \text{for} \ i = 1 \text{ to } n
  \]
Building Decision Trees in ID3

• Let $N_i$ stand for the number of cases in $S$ that belong to class $C_i$. Then the probability that a random case $c$ belongs to class $C_i$ is estimated to be:

$$p(c \in C_i) = \frac{N_i}{|S|}$$

• Thus the amount of information in a message of class $C_i$ is:

$$I(c \in C_i) = -\log_2 p(c \in C_i) \text{ bits}$$

• Consider the set of target classes as a message set $\{C_1, C_2, \ldots, C_k\}$. The uncertainty $U(S)$ measures the average amount of information need to determine the class of a random case, $c \in S$, prior to partitioning by any test. Thus:

$$U(S) = \sum_{i=1 \text{ to } k} p(c \in C_i) \times I(c \in C_i) \text{ bits}$$
• Consider a similar uncertainty measure after S has been partitioned into \{S_1, S_2, \ldots, S_n\} by a test \(T\):

\[
U_T(S) = \sum_{i=1}^{n} \left( \frac{|S_i|}{|S|} \times U_i(S_i) \right)
\]

• \(U_T(S)\) measures how much information is needed for the partitioning. Thus ID3 decides what attribute to branch on next by selecting the test \(T\) that gains the most information, i.e. maximum \(G_S(T)\) given below:

\[
G_S(T) = U(S) - U_T(S)
\]
“Play/Don’t Play” Example

\[ S = \{P, N\} \]

\[
U(S) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}
\]

\[
= -(9/14) \log_2 (9/14) - (6/14) \log_2 (6/14) = 0.9338
\]

For \( T = \text{Outlook} \), \( \{S_1, S_2, S_3\} = \{\text{sunny, overcast, rain}\} \)

\[
U(\text{sunny}) = -(2/5) \log_2 (2/5) - (3/5) \log_2 (3/5) = 0.971
\]

\[
U(\text{overcast}) = -(4/4) \log_2 (4/4) - (0/4) \log_2 (0/4) = 0
\]

\[
U(\text{rain}) = -(3/5) \log_2 (3/5) - (2/5) \log_2 (2/5) = 0.971
\]

\[
U_{\text{Outlook}}(S) = (5/14) \times 0.971 + (4/14) \times 0 + (5/14) \times 0.971 = 0.6936
\]

\[
G_S(\text{Outlook}) = U(S) - U_{\text{Outlook}}(S) = 0.9338 - 0.694 = 0.2402
\]
Similarly,

\[ U_{Temperature}(S) = 0.9226 \]
\[ U_{Humidity}(S) = 0.9177 \]
\[ U_{Windy}(S) = 0.8922 \]
\[ G_S(Temperature) = U(S) - U_{Temperature}(S) = 0.9338 - 0.9226 = 0.0112 \]
\[ G_S(Humidity) = U(S) - U_{Humidity}(S) = 0.9338 - 0.9177 = 0.0161 \]
\[ G_S(Windy) = U(S) - U_{Windy}(S) = 0.9338 - 0.8922 = 0.0416 \]

Thus \( T = Outlook \) has the highest information gain and is thus chosen as the root.