Information Gain

- We want to determine which attribute in a given set of training feature vectors is most useful for discriminating between the classes to be learned.

- Information gain tells us how important a given attribute of the feature vectors is.

- We will use it to decide the ordering of attributes in the nodes of a decision tree.
Calculating Information Gain

**Information Gain** = entropy(parent) − [average entropy(children)]

**Parent**

\[
\text{entropy} = -\left(\frac{14}{30} \cdot \log_2 \frac{14}{30}\right) - \left(\frac{16}{30} \cdot \log_2 \frac{16}{30}\right) = 0.996
\]

**17 instances**

\[
\text{entropy} = -\left(\frac{4}{17} \cdot \log_2 \frac{4}{17}\right) = 0.787
\]

**13 instances**

\[
\text{entropy} = -\left(\frac{12}{13} \cdot \log_2 \frac{12}{13}\right) = 0.391
\]

**Entire population (30 instances)**

\[
\text{entropy} = -\left(\frac{13}{17} \cdot \log_2 \frac{13}{17}\right) - \left(\frac{4}{17} \cdot \log_2 \frac{4}{17}\right) = 0.787
\]

\[
\text{entropy} = -\left(\frac{1}{13} \cdot \log_2 \frac{1}{13}\right) - \left(\frac{12}{13} \cdot \log_2 \frac{12}{13}\right) = 0.391
\]

*(Weighted) Average Entropy of Children* = \[
\left(\frac{17}{30} \cdot 0.787\right) + \left(\frac{13}{30} \cdot 0.391\right) = 0.615
\]

**Information Gain** = 0.996 - 0.615 = 0.38 *for this split*
Entropy-Based Automatic Decision Tree Construction

Training Set $S$

$x_1 = (f_{11}, f_{12}, \ldots, f_{1m})$

$x_2 = (f_{21}, f_{22}, f_{2m})$

$\vdots$

$\vdots$

$x_n = (f_{n1}, f_{n2}, f_{2m})$

Node 1

What feature should be used?

What values?

Quinlan suggested information gain in his ID3 system and later the gain ratio, both based on entropy.
Using Information Gain to Construct a Decision Tree

1. Choose the attribute A with highest information gain for the full training set at the root of the tree.

2. Construct child nodes for each value of A. Each has an associated subset of vectors in which A has a particular value.

3. Repeat recursively till when?

\[ S' = \{ s \in S \mid \text{value}(A) = v_1 \} \]
Simple Example

Training Set: 3 features and 2 classes

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>I</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>I</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>II</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>II</td>
</tr>
</tbody>
</table>

How would you distinguish class I from class II?
<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>I</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>I</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>II</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>II</td>
</tr>
</tbody>
</table>

Split on attribute X

If X is the best attribute, this node would be further split.

\[
\begin{align*}
E_{\text{parent}} &= 1 \\
\text{GAIN} &= 1 - \left( \frac{3}{4} \right) \times 0.9184 - \left( \frac{1}{4} \right) \times 0 = 0.3112
\end{align*}
\]
<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>I</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>I</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>II</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>II</td>
</tr>
</tbody>
</table>

Split on attribute Y

\[
\begin{align*}
E_{\text{parent}} &= 1 \\
\text{GAIN} &= 1 - \left(\frac{1}{2}\right) 0 - \left(\frac{1}{2}\right) 0 = 1; \text{ BEST ONE}
\end{align*}
\]
<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>I</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>I</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>II</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>II</td>
</tr>
</tbody>
</table>

Split on attribute Z

**Split on attribute Z**

\[
E_{\text{parent}} = 1
\]

\[
GAIN = 1 - \left( \frac{1}{2}\right)(1) - \left( \frac{1}{2}\right)(1) = 0 \quad \text{ie. NO GAIN; WORST}
\]