Decision Trees

Keith Cheverst

Overview...

• Inductive Method
  – Generalises to learn rules from observed “training” data
    • IF the Wind is weak THEN PlayGolf is yes
    • IF the Outlook is overcast AND the wind is strong THEN PlayGolf is no
  – May turn out to be wrong when used on further data
    • Next Sunday the wind is strong and the outlook overcast but I do play golf...
  – DTs can be used for working out whether you get that cheap loan...

How do we produce Decision Trees?

• Basic Algorithm is recursive...
  – Determine which of the candidate attributes to use as the top most node of the decision tree (candidates being “Wind” or “Outlook” in the first step using this example)
  – do this by calculating the information gain for each candidate attribute
    – pick the attribute with the highest information gain.

Information Gain

• To understand Information Gain we firstly have to understand the concept of Entropy...

Entropy

“characterizes the impurity of an arbitrary collection of examples” [Mitchell, 1997]

• So if the impurity or randomness of a collection (with respect to the target classifier) is high then the entropy is high
• But if there is no randomness (complete uniformity with respect to the target classifier) then the entropy is zero

\[ \text{Entropy}(S) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_- \]

Where target classification is boolean
\[ p_+ \] the proportion of positive examples in collection S
\[ p_- \] the proportion of negative examples in collection S

\[ \text{Entropy}(2+,0-) = 0 \]
(Max uniformity)

\[ \text{Entropy}(1+,1-) = 1 \]
(Min uniformity)

\[ \text{Entropy}(2+,1-) = 0.92 \]
(Working to 2 decimal places)
How do we produce decision trees?

- Basic Algorithm is recursive...
  - Determine which attribute to use as the top most node of the decision tree (candidates being "Wind" or "Outlook" in the first step using this example)
  - do this by calculating the information gain for each candidate attribute
    - pick the attribute with the highest information gain.

Step 1.

- Determine which attribute to use as the top most node of the decision tree – do this by calculating the information gain for each candidate attribute – pick the attribute with the highest information gain.

\[
\text{Gain}(S, A) \equiv \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \left( \frac{|S_v|}{|S|} \right) \text{Entropy}(S_v)
\]

The information Gain of attribute \( A \) in collection \( S \) where \( \text{Values}(A) \) is the set of possible values for attribute \( A \) and \( S_v \) is the subset of \( S \) for which attribute \( A \) has the value \( v \).

\[
\text{Entropy}(S) \equiv -p \cdot \log_2 p - q \cdot \log_2 q
\]

Where target classification is boolean:
- \( p \): the proportion of positive examples in collection \( S \)
- \( q \): the proportion of negative examples in collection \( S \)
Calculating the IG of the Outlook Attribute...

\[
\text{Gain}(S, \text{Outlook}) = 0.98 - \sum_{v \in \text{Values}(\text{rain}, \text{sunny}, \text{overcast})} \left( \frac{|S_v|}{|S|} \times \text{Entropy}(S_v) \right)
\]

\[
\text{Gain}(S, \text{Outlook}) = 0.98 - \left( \frac{|S_{\text{rain}}|}{|S|} \times \text{Entropy}(S_{\text{rain}}) \right) - \left( \frac{|S_{\text{sunny}}|}{|S|} \times \text{Entropy}(S_{\text{sunny}}) \right) - \left( \frac{|S_{\text{overcast}}|}{|S|} \times \text{Entropy}(S_{\text{overcast}}) \right)
\]

\[
= 0.98 - (2/7) \text{Entropy}([1+,1]) - (2/7) \text{Entropy}([2+]) - (2/7) \text{Entropy}([1+,1])
\]

\[
= 0.98 - (4/7) \text{Entropy}([1+,2]) - (2/7) \text{Entropy}([2+]) - (2/7) \text{Entropy}([1+,1])
\]

\[
= 0.98 - (3/7) \text{Entropy}([1+,2]) - (4/7) \text{Entropy}([1+,3])
\]

\[
= 0.98 - (3/7) \times 0.0 - (4/7) \times 0.46 - 0.46
\]

\[
= 0.52
\]

Calculating the IG of the Wind Attribute...

\[
\text{Gain}(S, \text{Wind}) = 0.98 - \sum_{v \in \text{Values}(\text{rain}, \text{sunny}, \text{overcast})} \left( \frac{|S_v|}{|S|} \times \text{Entropy}(S_v) \right)
\]

\[
\text{Gain}(S, \text{Wind}) = 0.98 - \left( \frac{|S_{\text{rain}}|}{|S|} \times \text{Entropy}(S_{\text{rain}}) \right) - \left( \frac{|S_{\text{sunny}}|}{|S|} \times \text{Entropy}(S_{\text{sunny}}) \right) - \left( \frac{|S_{\text{overcast}}|}{|S|} \times \text{Entropy}(S_{\text{overcast}}) \right)
\]

\[
= 0.98 - (2/7) \text{Entropy}([1+,1]) - (2/7) \text{Entropy}([2+]) - (4/7) \text{Entropy}([1+,3])
\]

\[
= 0.98 - (3/7) \times 0.0 - (4/7) \times 0.46 - 0.46
\]

\[
= 0.52
\]

Let's consider another set of Training Data...

- This time with 15 rows and 4 attributes not including the target attribute ‘PlayTennis’
- Note that this is still an extremely small sample of training data – it’s not uncommon to run decision tree learning algorithms on census data!
  - See, UCI Machine Learning Repository for much larger set’s of training data that you can play with...
Step 1…

• Which attribute should we have as the top most node of our decision tree
• Determine the information gain for each candidate attribute…
  – Gain(S, Outlook) = 0.246
  – Gain(S, Humidity) = 0.151
  – Gain(S, Wind) = 0.048
  – Gain(S, Temperature) = 0.029
• So would have Outlook as top Node

After Step 1…

![Decision Tree Diagram]

Discussions…

Ubiquitous Computing…

Comprehensibility

• When discussing key challenges in the ubicomp domain, (Abowd and Mynatt, 2000) make comments concerning the need for comprehensibility :

  “One fear of users is the lack of knowledge of what some computing system is doing, or that something is being done ‘behind their backs’.”


Decision Trees and their potential role in Ubiquitous Computing

• Consider the following problem…
• We wish to learn behaviour patterns of a user around a given task so that we can build a system to provide proactive support for that task.
• Sensors can be used to build up ‘Context History’ Tables
• Because its proactive we want the user to be able to query the system with questions such as ‘why did you do that, what rule where you following?’

Example Proactive System: IOS…

The contexts considered in the experiment were: temperature, humidity, noise level, light level, the status of window, the status of fan and the location of a user.

How does the owner tend to cool his/her office?
Overall approach...

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Temp</th>
<th>Humidity</th>
<th>Light</th>
<th>Window</th>
<th>Fan</th>
<th>Home</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004-26-1</td>
<td>14:40</td>
<td>25</td>
<td>20</td>
<td>52</td>
<td>closed</td>
<td>off</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>2004-26-1</td>
<td>14:40</td>
<td>25</td>
<td>20</td>
<td>52</td>
<td>closed</td>
<td>off</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>2004-26-1</td>
<td>14:40</td>
<td>25</td>
<td>20</td>
<td>52</td>
<td>closed</td>
<td>off</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>2004-26-1</td>
<td>14:40</td>
<td>25</td>
<td>20</td>
<td>52</td>
<td>closed</td>
<td>off</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>2004-26-1</td>
<td>14:40</td>
<td>25</td>
<td>20</td>
<td>52</td>
<td>closed</td>
<td>off</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>2004-26-1</td>
<td>14:40</td>
<td>25</td>
<td>20</td>
<td>52</td>
<td>closed</td>
<td>off</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>2004-26-1</td>
<td>14:40</td>
<td>25</td>
<td>20</td>
<td>52</td>
<td>closed</td>
<td>off</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>2004-26-1</td>
<td>14:40</td>
<td>25</td>
<td>20</td>
<td>52</td>
<td>closed</td>
<td>off</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>2004-26-1</td>
<td>14:40</td>
<td>25</td>
<td>20</td>
<td>52</td>
<td>closed</td>
<td>off</td>
<td>off</td>
<td>on</td>
</tr>
</tbody>
</table>

Note that “raw” values would be converted into “symbolic” values, e.g., temperatures below 20°C would be classified as “cold.” Process called “Discretisation.”

When a Rule for Device Actuation is Triggered

- When a suggestion prompt is issued (which occurs if the user has indicated that a prompt rather than automatic action is required) it is displayed on the main control GUI.
  - If the system suggests that the fan should be turned off, then the UI changes to that shown below: the text on the ‘OFF’ button flashes black and white.

Some Useful Links...

- **Reading**: Machine Learning, Mitchell, McGraw-Hill
  - Pages 52 to 63
- **UCI Machine Learning Repository**
- **Tools for learning Computational Intelligence**
- **Log base2 table**
  - [http://usl.sis.pitt.edu/trurl/log-table.html](http://usl.sis.pitt.edu/trurl/log-table.html)