Decision Trees

6601
# Classification

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A₁</td>
<td>A₂</td>
<td>...</td>
<td>Aₙ</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>v₁₁</td>
<td>v₁₂</td>
<td>...</td>
<td>v₁ₙ</td>
<td>c₁</td>
<td></td>
</tr>
<tr>
<td>v₂₁</td>
<td>v₂₂</td>
<td>...</td>
<td>v₂ₙ</td>
<td>c₂</td>
<td></td>
</tr>
</tbody>
</table>

...
Decision Tree

<table>
<thead>
<tr>
<th>H</th>
<th>W</th>
<th>O</th>
<th>P</th>
<th>Y/N?</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>S</td>
<td>S</td>
<td>Y/N?</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>S</td>
<td>O</td>
<td>Y/N?</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>W</td>
<td>R</td>
<td>Y/N?</td>
<td></td>
</tr>
</tbody>
</table>

CHAPTER 3 DECISION TREE

- **Outlook**
  - Sunny
  - Overcast
  - Rain

- **Humidity**
  - High
  - Normal

- **Wind**
  - Strong
  - Weak
Decision Tree

<table>
<thead>
<tr>
<th>H</th>
<th>W</th>
<th>O</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>S</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>S</td>
<td>O</td>
<td>Y/N?</td>
</tr>
<tr>
<td>H</td>
<td>W</td>
<td>R</td>
<td>Y/N?</td>
</tr>
</tbody>
</table>

Outlook

- Sunny
- Overcast
- Rain

Humidity

- High
- Normal

Wind

- Strong
- Weak
**Decision Tree**

<table>
<thead>
<tr>
<th>H</th>
<th>W</th>
<th>O</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>S</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>S</td>
<td>O</td>
<td>Y</td>
</tr>
<tr>
<td>H</td>
<td>W</td>
<td>R</td>
<td>Y/N?</td>
</tr>
</tbody>
</table>

- **Outlook**: Sunny, Overcast, Rain
- **Humidity**: High, Normal
- **Wind**: Strong, Weak

---

Thursday, October 3, 13
Decision Tree
Continuous Attributes

\[ x > \text{th} \]

Yes \quad No

\[ \ldots \quad \ldots \]
Continues Attributes

Guillotine Cut

What is the tree?
Continues Attributes

\[
x > \text{th}_1
\]

\[
y > \text{th}_2
\]

\[
\text{yes} \rightarrow \text{yes}
\]

\[
\text{no}
\]

\[
\text{yes} \rightarrow \text{yes}
\]

\[
\text{no}
\]
Decision Tree

Patrons?
- None: F
- Some: T
- Full: WaitEstimate?
  - >60: F
  - 30-60: Alternate?
    - No: F
    - Yes: Hungry?
      - No: Reservation?
        - No: Bar?
          - No: F
          - Yes: T
      - Yes: Fri/Sat?
        - No: Raining?
          - No: F
          - Yes: T
        - Yes: Alternate?
          - No: T
          - Yes: T
Which Tree is better if the classification accuracy is the same?
Decision Tree Learning

Idea: a good attribute splits the examples into subsets that are (ideally) “all positive” or “all negative”

*Patrons?* is a better choice—gives information about the classification
Entropy

Measure Of Uncertainty / Unpredictability in a Random Variable

Claude Shannon

Quantifies Information in a Message

\[ H(X) = - \sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \]
Entropy

\[ H(X) = - \sum_{i=1}^{n} p(x_i) \log_b p(x_i) \]

\[
\left( -\frac{6}{12} \ast \log_2 \frac{6}{12} \right) - \left( -\frac{6}{12} \ast \log_2 \frac{6}{12} \right) = 1
\]

Idea: a good attribute splits the examples into subsets that are (ideally) “all positive” or “all negative”

*Patrons?* is a better choice—gives **information** about the classification
Information Gain

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

Idea: a good attribute splits the examples into subsets that are (ideally) “all positive” or “all negative”

_Patrons?_ is a better choice—gives _information_ about the classification.
Calculate

\[ H(X) = - \sum_{i=1}^{n} p(x_i) \log_b p(x_i) \]

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

<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>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Result

\[ \text{Gain}(S, \text{Outlook}) = 0.246 \]
\[ \text{Gain}(S, \text{Humidity}) = 0.151 \]
\[ \text{Gain}(S, \text{Wind}) = 0.048 \]
\[ \text{Gain}(S, \text{Temperature}) = 0.029 \]
function DTL(examples, attributes, default) returns a decision tree

if examples is empty then return default
else if all examples have the same classification then return the classification
else if attributes is empty then return MODE(examples)
else
    \textit{best} \leftarrow \text{Choose-Attribute}(attributes, examples)
    \textit{tree} \leftarrow \text{a new decision tree with root test } \textit{best}
    \textbf{for each} value \textit{v}_i \textbf{of } \textit{best} \textbf{do}
        \textit{examples}_i \leftarrow \{\text{elements of } \textit{examples} \text{ with } \textit{best} = \textit{v}_i\}
        \textit{subtree} \leftarrow \text{DTL}(\textit{examples}_i, \text{attributes} – \textit{best}, \text{MODE}(\textit{examples}))
        add a branch to \textit{tree} with label \textit{v}_i \text{ and subtree } \textit{subtree}

return \textit{tree}
Greedy Algorithm

- In search terms: A greedy algorithm with the Information Gain as a Heuristic
- Could we do better?
@relation 'gatech_admission'
@attribute 'recommendation' {'strong', 'weak'}
@attribute 'gpa' real
@attribute 'gre_math' real
@attribute 'gre_verbal' real
@attribute 'admitted' {'yes', 'no'}

@data
'strong', 4, 800, 800, 'yes'
'weak', 3.4, 600, 500, 'yes'
'strong', 3.6, 800, 550, 'yes'
'strong', 3, 700, 650, 'yes'
'weak', 3.2, 800, 800, 'yes'
'strong', 4, 550, 500, 'yes'
'strong', 3.7, 700, 750, 'yes'
'weak', 4, 800, 200, 'yes'
'strong', 4, 200, 800, 'yes'
'strong', 3.4, 600, 500, 'yes'
'strong', 3.6, 800, 550, 'yes'
'weak', 3, 700, 650, 'yes'
'strong', 4, 550, 500, 'yes'
'strong', 3.7, 700, 750, 'yes'
'weak', 2.8, 800, 800, 'no'
'strong', 2, 500, 200, 'no'
'strong', 3.5, 200, 800, 'no'
'weak', 2, 800, 800, 'no'
'weak', 1.7, 100, 100, 'no'
'strong', 3.7, 50, 0, 'no'
'weak', 2.8, 100, 800, 'no'
'weak', 4, 200, 200, 'no'
'strong', 3.5, 200, 800, 'no'
'weak', 3.7, 50, 0, 'no'

What is the best?
Weka Demo
Use Case: Mobile Text Entry
Rollon (E, W)
Rolloff (E ,W)
Use Case: Fat Thumbs

```
prob ≤ 0
   | prevcuradjacent.nom = False
   | curfutadjacent.nom = False
   | dropprobdiffabs ≤ 0.000934: repeat (104.0/45.0)
   | dropprobdiffabs > 0.000934
   | futneighborprob ≤ 0: nonobo (552.0/11.0)
   | futneighborprob > 0
   | neighborprob ≤ 0.011765: nonobo (143.0/21.0)
   | neighborprob > 0.011765
   | ud.subl ≤ -140: nonobo (172.0/80.0)
   | ud.subl > -140: obosubstitute (527.0/109.0)
   | curfutadjacent.nom = True: rollon (323.0/73.0)
   | prevcuradjacent.nom = True: rolloff (555.0/46.0)
prob > 0
   | dt.ud.0.p1 ≤ 124
   | dropprobdiffsign ≤ -1: rolloff (155.0/36.0)
   | dropprobdiffsign > -1: nonerror (111.0/49.0)
```

Use Case: Fat Thumbs

No Dictionaries were harmed in the making of this Decision Tree.
What are the drawbacks?

Decision Trees are known to over fit the data
Ensemble Learning

Mixture of Experts: Have we seen one before?
Random Forests

Winning!
Random Forests

Bagging: Bootstrap AGGregation

- **INPUT**: Data Set of size $N$ with $M$ dimensions
- 1) **SAMPLE** $n$ times from Data
- 2) **SAMPLE** $m$ times from Attributes
- 3) **LEARN TREE** on sampled Data and Attributes
- **REPEAT UNTIL** $k$ trees
Use Case: Kinect

depth image $\rightarrow$ body parts $\rightarrow$ 3D joint proposals
Use Case: Kinect
Use Case: Kinect

Pixel to classify Offset

(a) $\theta_1$ $\theta_2$

(b) $\theta_1$ $\theta_2$
Use Case: Kinect

Training 3 trees to depth 20 from 1 million images takes about a day on a 1000 core cluster