Decision Tree Learning
CS472/CS473 – Fall 2005

Decision Tree Example: BigTip

Food
Price
Speedy

yes
no

yes
no

yes
no

1 great yes yes adequate no yes
2 great no yes adequate no yes
3 mediocre yes no high no no
4 great yes yes adequate yes yes

Top-Down Induction of DT (simplified)

Training Data: D = {(x₁,y₁),…,(xₙ,yₙ)}

FDIDF(D,cₐₐₜ)
• IF(all examples in D have same class c)
  – Return leaf with class c (or class cₐₜₜ, if D=Ø)
• ELSE IF(no attributes left to test)
  – Return leaf with class c of majority in D
• ELSE
  – Pick A as the “best” decision attribute for next node
  – FOR each value vᵢ of A create a new descendent of node
  • Dᵢ = {(x,y) ∈ D : attribute A of x has value vᵢ}
  • Subtree tᵢ for vᵢ is TDIDF(Dᵢ,cₐₜₜ)
  – RETURN tree with A as root and tᵢ as subtrees

Example: Text Classification

• Task: Learn rule that classifies Reuters Business News
  – Class +: “Corporate Acquisitions”
  – Class -: Other articles
  – 2000 training instances

• Representation:
  – Boolean attributes, indicating presence of a keyword in article
  – 9947 such keywords (more accurately, word “stems”)

Example: TDIDT

Training Data D:

<table>
<thead>
<tr>
<th>F</th>
<th>S</th>
<th>P</th>
<th>BigTip</th>
</tr>
</thead>
<tbody>
<tr>
<td>x₁ = (g, y, n)</td>
<td>f(x₁) = 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x₂ = (g, u, h)</td>
<td>f(x₂) = 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x₃ = (g, y, n)</td>
<td>f(x₃) = 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x₄ = (g, y, a)</td>
<td>f(x₄) = 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x₅ = (m, y, a)</td>
<td>f(x₅) = 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x₆ = (y, y, n)</td>
<td>f(x₆) = 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x₇ = (g, y, a)</td>
<td>f(x₇) = 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x₈ = (g, y, h)</td>
<td>f(x₈) = 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x₉ = (m, y, a)</td>
<td>f(x₉) = 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x₁₀ = (g, y, n)</td>
<td>f(x₁₀) = 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Picking the Best Attribute to Split

• Ockham’s Razor:
  – All other things being equal, choose the simplest explanation

• Decision Tree Induction:
  – Find the smallest tree that classifies the training data correctly

• Problem
  – Finding the smallest tree is computationally hard

• Approach
  – Use heuristic search (greedy search)
**Which Attribute is “Best”?**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>[29+, 35-]</td>
</tr>
<tr>
<td>A2</td>
<td>[21+, 5-]</td>
</tr>
<tr>
<td></td>
<td>[8+, 30-]</td>
</tr>
<tr>
<td></td>
<td>[18+, 33-]</td>
</tr>
<tr>
<td></td>
<td>[11+, 2-]</td>
</tr>
</tbody>
</table>

- **Heuristics**
  - Pick split that decreases training error the most
  - Pick split that maximizes information (Information Gain)
  - Other statistical tests

**Information Gain**
- Idea: Measure how much information an attribute conveys
- Entropy: Number of bits to transmit one label (~disorder)
  \[
  Entropy(D) = \frac{n}{p} \log_2 p - \frac{n}{q} \log_2 q
  \]
- Information Gain: Reduction in entropy, if attribute value known
  \[
  Gain(D, A) = Entropy(D) - \sum_{e \in Values(A)} \frac{|D_e|}{|D|} \cdot Entropy(D_e)
  \]

**Decision Tree for “Corporate Acq.”**

- \( vs = 1: - \)
- \( vs = 0: \)
  - \( export = 1: \)
    - \( rate = 1: + \)
    - \( stake = 1: + \)
    - \( debenture = 1: + \)
    - \( takeover = 1: + \)
    - \( takeover = 0: \)
      - \( file = 0: - \)
      - \( file = 1: \)
        - \( share = 1: + \)
        - \( share = 0: - \)

**Total size of tree:**
- 299 nodes

**How Expressive are Decision Trees?**

- **What functions \( h: X \rightarrow Y \) can a decision tree represent?**
  - Assume that \( X \) is finite (only finite number of instances)
    - Decision trees can represent any function over a finite instance space \( X \).
  - What if \( X \) is not finite (e.g. integer-valued attributes)?
  - What if \( X \) is not discrete (e.g. real-valued attributes)?
  - What if the data contains noise?
    - In the most extreme case, examples can have the same attribute values, but different labels.

**TDIDT Extensions**

- **Numerical (continuous) attributes**
  - Use \( > \) and \( < \) in attribute tests
  - Example: \( age < 40 \) and \( age \geq 40 \)

- **Finite attributes with many values**
  - Example:
    - Target concept is “brakes defect”
    - Instances: all cars in the US
    - Attributes: Manufacturer (3 values), VIN (100,000,000 values)
  - Which attribute will Information Gain select? \( \rightarrow \) GainRatio

- **Numerical (continuous) target attribute (regression)**
  - E.g. pick attribute test so that target values become more similar
  - E.g. predict mean value of examples in each leaf

- **Early stopping and Pruning**