Decision Tree Learning

CS4780/5780 – Machine Learning
Fall 2013
Thorsten Joachims
Cornell University

Reading: Mitchell Sections 2.1-2.3, 2.5-2.5.2, 2.7, Chapter 3

Supervised Learning

- Task:
  - Learn (to imitate) a function $f: X \rightarrow Y$

- Training Examples:
  - Learning algorithm is given the correct value of the function for particular inputs $\rightarrow$ training examples
  - An example is a pair $(x, f(x))$, where $x$ is the input and $f(x)$ is the output of the function applied to $x$.

- Goal:
  - Find a function $h: X \rightarrow Y$
    that approximates $f: X \rightarrow Y$
    as well as possible.

Hypothesis Space

- Instance Space $X$: Set of all possible objects described by attributes.
- Target Function $f$: Maps each instance $x \in X$ to target label $y \in Y$ (hidden).
- Hypothesis $h$: Function that approximates $f$.
- Hypothesis Space $H$: Set of functions we allow for approximating $f$.
- Training Data $S$: Set of instances labeled with target function $f$.

Inductive Learning Strategy

- Strategy and hope (for now, later theory):
  Any hypothesis $h$ found to approximate the target function $f$ well over a sufficiently large set of training examples $S$ will also approximate the target function well over other unobserved examples.

- Can compute:
  - A hypothesis $h \in H$ such that $h(x)=f(x)$ for all $x \in S$.

- Ultimate Goal:
  - A hypothesis $h \in H$ such that $h(x)=f(x)$ for all $x \in X$.

Consistency

Definition: A hypothesis $h$ is consistent with a set of training examples $S$ if and only if $h(x) = y$ for each training example $(x, y) \in S$.

$$\text{Consistent}(h, S) \Leftrightarrow \forall (x, y) \in S : h(x) = y$$

Version Space

Definition: The version space, $V_{S_{H,S}}$, with respect to hypothesis space $H$ and training examples $S$, is the set of hypotheses from $H$ consistent with all training examples in $S$.

$$V_{S_{H,S}} = \{ h \in H \mid \text{Consistent}(h, S) \}$$
List-Then-Eliminate Algorithm

- init VS ← H
- For each training example (x, y) ∈ S
  - remove from VS any hypothesis h for which h(x) ≠ y
- Output VS

Top-Down Induction of DT (simplified)

Training Data: \( S = ((x_1, y_1), \ldots, (x_n, y_n)) \)

\( \text{TID}T(S, y_{\text{def}}) \)
- IF(all examples in S have same y)
  - Return leaf with class y (or class y_{\text{def}}, if S is empty)
- ELSE
  - Pick A as the "best" decision attribute for next node
  - For each value \( v_i \) of A create a new descendent of node
    - \( S_i = ((x, y) \in S : \text{attr} A of x has value } v_i) \)
    - Subtree \( t_i \) for \( v_i \) is \( \text{TID}T(S_i, y_{\text{def}}) \)
  - RETURN tree with A as root and \( t_i \) as subtrees

• \text{Top-Down Induction of DT (simplified)}
• \text{Which Attribute is "Best"?}
• \text{Decision Tree Example: A* Homework}

<table>
<thead>
<tr>
<th>correct (true/false)</th>
<th>color (yes/no)</th>
<th>original (yes/no)</th>
<th>presentation (clear/unclear)</th>
<th>binder (yes/no)</th>
<th>A*</th>
</tr>
</thead>
<tbody>
<tr>
<td>complete yes</td>
<td>yes</td>
<td>clear</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>partial no</td>
<td>yes</td>
<td>clear</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>complete yes</td>
<td>no</td>
<td>unclear</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>complete yes</td>
<td>yes</td>
<td>clear</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Example: TDIDT

TDIDT(\( S, y_{\text{def}} \))
- IF(all ex in S have same y)
  - Return leaf with class y (or class y_{\text{def}}, if S is empty)
- ELSE
  - Pick A as the "best" decision attribute for next node
  - For each value \( v_i \) of A create a new descendent of node
    - \( S_i = ((x, y) \in S : \text{attr} A of x has value } v_i) \)
    - Subtree \( t_i \) for \( v_i \) is \( \text{TID}T(\( S_i, y_{\text{def}} \) \))
  - RETURN tree with A as root and \( t_i \) 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: Text Classification**

**Example: TDIDT**

**Example: Decision Tree Example: A* Homework**
Decision Tree for “Corporate Acq.”

Learned tree:
- has 299 nodes
- is consistent

Accuracy of learned tree:
- 11% error rate

Note: word stems expanded for improved readability.