Decision Tree Learning

CS4780/5780 – Machine Learning
Fall 2011
Thorsten Joachims
Cornell University

Reading: Mitchell Sections 2.1, 2.2, 2.5-2.5.2, 2.7, Chapter 3
- Hypothesis space
- Version space
- Inductive learning hypothesis
- List-then-eliminate algorithm
- Decision tree representation
- Classifying with a decision tree
- ID3 decision tree learning algorithm
- Entropy, Information gain
- Overfitting
<table>
<thead>
<tr>
<th>correct</th>
<th>color</th>
<th>original</th>
<th>presentation</th>
<th>binder</th>
<th>A+Homework</th>
</tr>
</thead>
<tbody>
<tr>
<td>complete</td>
<td>yes</td>
<td>yes</td>
<td>clear</td>
<td>no</td>
<td>yes / +1</td>
</tr>
<tr>
<td>complete</td>
<td>no</td>
<td>yes</td>
<td>clear</td>
<td>no</td>
<td>yes / +1</td>
</tr>
<tr>
<td>partial</td>
<td>yes</td>
<td>no</td>
<td>unclear</td>
<td>no</td>
<td>no / -1</td>
</tr>
<tr>
<td>complete</td>
<td>yes</td>
<td>yes</td>
<td>clear</td>
<td>yes</td>
<td>yes / +1</td>
</tr>
</tbody>
</table>

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

$$Consistent(h, S) \equiv \forall (x, y) \in S : h(x) = y$$

<table>
<thead>
<tr>
<th>correct</th>
<th>color</th>
<th>original</th>
<th>presentation</th>
<th>binder</th>
<th>A+Homework</th>
</tr>
</thead>
<tbody>
<tr>
<td>complete</td>
<td>yes</td>
<td>yes</td>
<td>clear</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>complete</td>
<td>no</td>
<td>yes</td>
<td>clear</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>partial</td>
<td>yes</td>
<td>no</td>
<td>unclear</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>complete</td>
<td>yes</td>
<td>yes</td>
<td>clear</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
Definition: The version space, \( V_{H,S} \), with respect to hypothesis space \( H \) and training examples \( S \), is the subset of hypotheses from \( H \) consistent with all training examples in \( S \).

\[
V_{H,S} \equiv \{ h \in H | \text{Consistent}(h, S) \}
\]
**Decision Tree Example: A+ Homework**

<table>
<thead>
<tr>
<th>correct</th>
<th>color</th>
<th>original</th>
<th>presentation</th>
<th>binder</th>
<th>A+ Homework</th>
</tr>
</thead>
<tbody>
<tr>
<td>complete</td>
<td>yes</td>
<td>yes</td>
<td>clear</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>complete</td>
<td>no</td>
<td>yes</td>
<td>clear</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>partial</td>
<td>yes</td>
<td>no</td>
<td>unclear</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>complete</td>
<td>yes</td>
<td>yes</td>
<td>clear</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
Training Data: \( S = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\} \)

\textbf{TDIDT}(S, y_{def})

- IF(all examples in \( S \) have same class \( y \))
  - Return leaf with class \( y \) (or class \( y_{def} \), 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
    - \( S_i = \{(\vec{x}, y) \in S : \text{attribute } A \text{ of } \vec{x} \text{ has value } v_i\} \)
    - Subtree \( t_i \) for \( v_i \) is TDIDT(\( S_i, y_{def} \))
  - RETURN tree with \( A \) as root and \( t_i \) as subtrees
**Example: TDIDT**

\[
\text{TDIDT}(S, y_{\text{def}})
\]

- **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
    - \(S_i = \{ (\vec x, y) \in S : \text{attrib } A \text{ of } \vec x \text{ is } v_i \}\)
    - Subtree \(t_i\) for \(v_i\) is \(\text{TDIDT}(S_i, y_{\text{def}})\)
  - RETURN tree with \(A\) as root and \(t_i\) as subtrees

**Example Data S:**

<table>
<thead>
<tr>
<th>(\vec{x})</th>
<th>(y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_1)</td>
<td>((c, y, c))</td>
</tr>
<tr>
<td>(x_2)</td>
<td>((c, n, u))</td>
</tr>
<tr>
<td>(x_3)</td>
<td>((c, y, u))</td>
</tr>
<tr>
<td>(x_4)</td>
<td>((c, n, c))</td>
</tr>
<tr>
<td>(x_5)</td>
<td>((p, y, c))</td>
</tr>
<tr>
<td>(x_6)</td>
<td>((g, y, c))</td>
</tr>
<tr>
<td>(x_7)</td>
<td>((c, y, c))</td>
</tr>
<tr>
<td>(x_8)</td>
<td>((c, y, u))</td>
</tr>
<tr>
<td>(x_9)</td>
<td>((p, y, c))</td>
</tr>
<tr>
<td>(x_{10})</td>
<td>((c, y, c))</td>
</tr>
</tbody>
</table>
Which Attribute is “Best”?

A₁

[29+, 35-]

true

[21+, 5-]

false

[8+, 30-]

A₂

[29+, 35-]

true

[18+, 33-]

false

[11+, 2-]
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”)

**LAROCHE STARTS BID FOR NECO SHARES**
Investor David F. La Roche of North Kingstown, R.I., said he is offering to purchase 170,000 common shares of NECO Enterprises Inc at 26 dlr each. He said the successful completion of the offer, plus shares he already owns, would give him 50.5 pct of NECO's 962,016 common shares. La Roche said he may buy more, and possible all NECO shares. He said the offer and withdrawal rights will expire at 1630 EST/2130 gmt, March 30, 1987.

**SALANT CORP 1ST QTR FEB 28 NET**
Oper shr profit seven cts vs loss 12 cts. Oper net profit 216,000 vs loss 401,000. Sales 21.4 mln vs 24.9 mln. NOTE: Current year net excludes 142,000 dlr tax credit. Company operating in Chapter 11 bankruptcy.
Decision Tree for “Corporate Acq.”

Total size of tree: 299 nodes

Note: word stems expanded for improved readability.
Definition: A particular instance of a learning problem is described by a probability distribution $P(X, Y)$.

Definition: A sample $S = ((\tilde{x}_1, y_1), ..., (\tilde{x}_n, y_n))$ is independently identically distributed (i.i.d.) according to $P(X, Y)$.
Sample Error and Generalization Error

Definition: The error on sample $S$ $\text{Err}_S(h)$ of a hypothesis $h$ is $\text{Err}_S(h) = \frac{1}{n} \sum_{i=1}^{n} \Delta(h(\bar{x}_i), y_i)$.

Definition: $\Delta(a, b)$ is the 0/1-loss function

$$\Delta(a, b) = \begin{cases} 0 & \text{if } a = b \\ 1 & \text{else} \end{cases}$$

Definition: The prediction/generalization/true error $\text{Err}_P(h)$ of a hypothesis $h$ for a learning task $P(X, Y)$ is

$$\text{Err}_P(h) = \sum_{\bar{x} \in X, y \in Y} \Delta(h(\bar{x}), y) P(X = \bar{x}, Y = y).$$
Learning as Prediction
Overview

- **Goal**: Find $h$ with small prediction error $Err_P(h)$ over $P(X,Y)$.
- **Strategy**: Find (any?) $h$ with small error $Err_{S\text{train}}(h)$ on training sample $S_{\text{train}}$.

- **Training Error**: Error $Err_{S\text{train}}(h)$ on training sample.
- **Test Error**: Error $Err_{S\text{test}}(h)$ on test sample is an estimate of $Err_P(h)$.
Overfitting

![Graph showing accuracy versus size of tree (number of nodes)]

- Note: Accuracy = 1.0 - Error
Reduced-Error Pruning

Accuracy vs. Size of tree (number of nodes)

On training data
On test data
On test data (during pruning)
Text Classification Example

Results

- **Unpruned Tree:**
  - Size: 437 nodes  Training Error: 0.0%  Test Error: 11.0%

- **Early Stopping Tree:**
  - Size: 299 nodes  Training Error: 2.6%  Test Error: 9.8%

- **Post-Pruned Tree:**
  - Size: 167 nodes  Training Error: 4.0%  Test Error: 10.8%

- **Rule Post-Pruning:**
  - Size: 164 tests  Training Error: 3.1%  Test Error: 10.3%
  - Examples of rules:
    - IF vs = 1 THEN - [99.4%]
    - IF vs = 0 & export = 0 & takeover = 1 THEN + [93.6%]