Module A) Supervised Learning

- Introduction to supervised learning
- Decision trees
  - Knowledge representation
  - Introduction to inducing trees
  - Ensemble of trees: boosting, bagging
- Neural networks
  - Extensions
    - for multiple dependent variables
    - Sequence analysis

Supervised Learning

Functional Overview – Learning Stage

Examples

- + + - → -

Age, … income, [Default | NoDefault]
Case A. 30, ..., $110K, Default
Case B. 50, ..., $130K, NoDefault
Case C. 45, ..., $90K, NoDefault
Case A. 32, ..., $105K, Default
Case B. 49, ..., $82K, NoDefault
Case C. 29, ..., $50K, NoDefault

Model/ Theory

Supervised Learning

Functional Overview – Application Stage

Unlabeled Examples

Age, … income, [Default | NoDefault]
Case zx. 29, ..., $113K, ?
Case zy. 42, ..., $81K, ?
Case zz. 41, ..., $92K, ?

Predictions

zx. Default
zy. NoDefault
zz. NoDefault

Model/ Theory

Supervised Learning Terminology

- Attribute, instances
- Training set, test set
- Training set accuracy
- Test set accuracy
- Confusion Matrix
- False positive
- False negative
- Cost sensitive learning
how and when to use decision trees

- how
  - classification
    - gain insights into why customers default on loans
  - prediction
    - screen customer loan applications
    - feature/column/attribute selection
    - to explain other learning techniques
- when
  - instances are attribute value pairs
  - discrete target function
  - noisy training data or missing values

another perspective of a decision tree model

top-down tree induction

main loop:
1. A ← the “best” decision attribute for next node
2. Assign A as decision attribute for node
3. For each value of A, create new descendant of node
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, then STOP. Else iterate over new leaf nodes

which attribute is best?
Which Column and Split Point?

- Multitude of techniques:
  - Entropy/Information gain
  - Chi square test (CHAID)
  - Test of independence
  - GINI index

Information Gain

\[ Gain(S, A) = \text{expected reduction in entropy due to sorting on } A \]

\[ Gain(S, A) \equiv Entropy(S) - \sum_{x \in \text{values}(A)} \frac{|S_x|}{|S|} \text{Entropy}(S_x) \]

Entropy

\[ Entropy(S) = \text{expected number of bits needed to encode class } \oplus \text{ or } \ominus \text{ of randomly drawn member of } S \text{ (under optimal, shortest-length code)} \]

\[ \text{Information theory: optimal length code assigns } \log_2 p \text{ bits to message having probability } p. \]

So, expected number of bits to encode \( \oplus \) or \( \ominus \) of random member of \( S \):

\[ p_0 (-\log_2 p_0) + p_1 (-\log_2 p_1) \]

\[ \text{Entropy}(S) = -p_0 \log_2 p_0 - p_1 \log_2 p_1. \]

Data Set

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>Play/Tennis</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>
Choosing the Next Attribute - 1

Which attribute is the best classifier?

Choosing the Next Attribute - 2

Representational and Search Bias

Note $H$ is the power set of instances $X$

Not really...

- Preference for short trees, and for those with high information gain attributes near the root
- Bias is a preference for some hypotheses, rather than a restriction of hypothesis space $H$
- Occam's razor: prefer the shortest hypothesis that fits the data

Occam’s Razor

- 14th Century Franciscan friar; William of Occam.
- The principle states that “Entities should not be multiplied unnecessarily.”
- People often reinvented Occam’s Razor
  - Newton - “We are to admit no more causes of natural things than such are both true and sufficient to explain their appearances.”
  - To most scientist the razor is:
    - “when you have two competing theories which make exactly the same predictions, the one that is simpler is the better.”

Lecture 2 - CS635

Lecture 2 - CS635

Lecture 2 - CS635

Lecture 2 - CS635
Additional Reading

- Cost sensitive learning

- Occam’s Razor