Instance-Based Learning

CS4780/5780 – Machine Learning
Fall 2011

Thorsten Joachims
Cornell University

Reading: Mitchell Chapter 1 & Sections 8.1 - 8.2
• Definition:

Acquire an operational definition of a general category of objects given positive and negative training examples.
<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>(3)</td>
<td>(2)</td>
<td>(2)</td>
<td>(3)</td>
<td>(2)</td>
<td></td>
</tr>
<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>

**Instance Space X:** Set of all possible objects described by attributes (often called features).

**Concept c:** Subset of objects from X (c is unknown).

**Target Function f:** Characteristic function indicating membership in c based on attributes (i.e. label) (f is unknown).

**Training Data S:** Set of instances labeled with target function.
Concept Learning as Learning a Binary Function

• Task:
  – Learn (to imitate) a function $f: X \rightarrow \{+1,-1\}$

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

• Goal:
  – Find a function $h: X \rightarrow \{+1,-1\}$ that approximates $f: X \rightarrow \{+1,-1\}$ as well as possible.
• **Given:** Training data \((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\)
  
  – Attribute vectors: \(\vec{x}_i \in X\)
  
  – Label: \(y_i \in Y = \{-1, +1\}\)

• **Parameter:**
  
  – Similarity function: \(K : X \times X \rightarrow \mathbb{R}\)
  
  – Number of nearest neighbors to consider: \(k\)

• **Prediction rule**
  
  – New example \(x'\) with
  
  – K-nearest neighbors: \(k\) train examples with largest \(K(\vec{x}_i, \vec{x}')\)

\[
h(\vec{x}') = \arg \max_{y \in Y} \left\{ \sum_{i \in knn(\vec{x}')} 1[y_i = y] \right\}
\]
• How will new examples be classified?
  – Similarity function?
  – Value of $k$?

$$h(x') = \arg \max_{y \in Y} \left\{ \sum_{i \in \text{knn}(x')} 1[y_i = y] \right\}$$
Given: Training data \((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\)
- Attribute vectors: \(\vec{x}_i \in X\)
- Target attribute: \(y_i \in \{-1, +1\}\)

Parameter:
- Similarity function: \(K : X \times X \rightarrow \mathbb{R}\)
- Number of nearest neighbors to consider: \(k\)

Prediction rule
- New example \(x'\)
- K-nearest neighbors: \(k\) train examples with largest \(K(\vec{x}_i, \vec{x}')\)

\[
h(\vec{x}') = \arg \max_{y \in Y} \left\{ \sum_{i \in knn(\vec{x}')} 1_{[y_i = y]} K(\vec{x}_i, \vec{x}') \right\}
\]
• Symbolic (nominal)
  – *EyeColor* \{brown, blue, green\}

• Boolean
  – *alife* \{TRUE, FALSE\}

• Numeric
  – Integer: *age* \([0, 105]\)
  – Real: *length*

• Structural
  – Natural language sentence: parse tree
  – Protein: sequence of amino acids
Example: Expensive Housing (> $200 / sqft)
Example: Effect of $k$

Hastie, Tibshirani, Friedman 2001
• 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.
Given: Training data \((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\)
- Attribute vectors: \(\vec{x}_i \in X\)
- Target attribute: \(y_i \in \mathbb{R}\)

Parameter:
- Similarity function: \(K : X \times X \rightarrow \mathbb{R}\)
- Number of nearest neighbors to consider: \(k\)

Prediction rule
- New example \(x'\)
- K-nearest neighbors: \(k\) train examples with largest \(K(\vec{x}_i, \vec{x}')\)

\[
h(\vec{x}') = \frac{\sum_{i \in \text{knn}(\vec{x}')} y_i K(\vec{x}_i, \vec{x}')}{\sum_{i \in \text{knn}(\vec{x}')} K(\vec{x}_i, \vec{x}')} \]
Based on your rating, we think you'll enjoy these titles.

Want more suggestions? How often do you watch?

- Never
- Sometimes
- Often

Goofy

Raunchy

Recently Watched

Top 10 for Thorsten

- The Last Enemy
- George Gently
- MI-5
- Love the Beast

Collaborative Filtering