Instance-Based Learning

CS4780/5780 – Machine Learning
Fall 2013

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.

Also called: binary classification, binary supervised learning,...
Concept Learning Example

Instance Space X: Set of all possible objects describable 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 $f : X \to \{+1,-1\}$

• Training Examples:
  – Learning algorithm is given the correct value of the function for particular inputs $\to$ 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 $h : X \to \{+1,-1\}$ that approximates $f : X \to \{+1,-1\}$ as well as possible.
K-Nearest Neighbor (KNN)

- Given: Training data \(((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n))\)
  - Attribute vectors: \(\vec{x}_i \in X\)
  - Labels: \(y_i \in Y\)
- 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 \(K(\vec{x}_i, \vec{x}')\)

\[
h(\vec{x}') = \arg \max_{y \in Y} \left\{ \sum_{i \in knn(\vec{x}')} 1[y_i = y] \right\}
\]
KNN Example

<table>
<thead>
<tr>
<th></th>
<th>correct (complete, partial, guessing)</th>
<th>color (yes, no)</th>
<th>original (yes, no)</th>
<th>presentation (clear, unclear, cryptic)</th>
<th>binder (yes, no)</th>
<th>A+</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>complete</td>
<td>yes</td>
<td>yes</td>
<td>clear</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>complete</td>
<td>no</td>
<td>yes</td>
<td>clear</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>partial</td>
<td>yes</td>
<td>no</td>
<td>unclear</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>4</td>
<td>complete</td>
<td>yes</td>
<td>yes</td>
<td>clear</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

- How will new examples be classified?
  - Similarity function?
  - Value of $k$?

$$h(x') = \arg \max_{y \in Y} \left\{ \sum_{i \in knn(x')} 1[y_i = y] \right\}$$
Weighted K-Nearest Neighbor

• Given: Training data \(((\tilde{x}_1, y_1), \ldots, (\tilde{x}_n, y_n))\)
  – Attribute vectors: \(\tilde{x}_i \in X\)
  – Target attribute: \(y_i \in Y\)

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

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

\[
h(\tilde{x}') = \arg \max_{y \in Y} \left\{ \sum_{i \in knn(\tilde{x}')} 1[y_i = y] K(\tilde{x}_i, \tilde{x}') \right\}
\]
Types of Attributes

• Symbolic (nominal)
  – EyeColor \{brown, blue, green\}
• Boolean
  – alive \{TRUE,FALSE\}
• Numeric
  – Integer: age [0, 105]
  – Real: height
• Structural
  – Natural language sentence: parse tree
  – Protein: sequence of amino acids
Example:
Expensive Housing (>\$200 / sqft)
Example: Effect of $k$

Hastie, Tibshirani, Friedman 2001
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.
Weighted K-NN for Regression

• Given: Training data \( (\tilde{x}_1, y_1), \ldots, (\tilde{x}_n, y_n) \)
  – Attribute vectors: \( \tilde{x}_i \in X \)
  – 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 \( K(\tilde{x}_i, \tilde{x}') \)

\[
h(\tilde{x}') = \frac{\sum_{i \in knn(\tilde{x}')} y_i K(\tilde{x}_i, \tilde{x}')}{\sum_{i \in knn(\tilde{x}')} K(\tilde{x}_i, \tilde{x}')}\]
### Collaborative Filtering

<table>
<thead>
<tr>
<th>Rating Matrix</th>
<th>( m_1 )</th>
<th>( m_2 )</th>
<th>( m_3 )</th>
<th>( m_4 )</th>
<th>( m_5 )</th>
<th>( m_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_1 )</td>
<td></td>
<td>1</td>
<td>5</td>
<td></td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>( u_2 )</td>
<td></td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>( u_3 )</td>
<td></td>
<td>2</td>
<td>4</td>
<td></td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>( u )</td>
<td>?</td>
<td>1</td>
<td>4</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

**Recently Watched**
- Trailer Park Boys

**Top 10 for Thorsten**
- The Last Enemy
- MI-5
- Love the Beast