Instance-Based Learning

CS4780 – Machine Learning
Fall 2009

Thorsten Joachims
Cornell University

Reading: Mitchell Chapter 1 & Sections 8.1 - 8.2

Concept Learning

Definition:
Acquire an operational definition of a general category of objects given positive and negative training examples.

Concept Learning Example

<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>partial</td>
<td>yes</td>
<td>no</td>
<td>unclear</td>
<td>no</td>
<td>yes</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.

K-Nearest Neighbor (KNN)

• Given: Training data \( \{x_i, y_i\}_{i=1}^n \)
  
  \( x_i \in \mathcal{X} \)
  
  \( y_i \in \{-1, +1\} \)

• Parameter:
  
  Similarity function: \( R : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} \)

  Number of nearest neighbors to consider: \( k \)

• Prediction rule
  
  New example: \( x' \)

  K-nearest neighbors: \( k \) training examples with largest \( R(x_i, x') \)

  \[ h(x') = \arg \max_{y \in \{+1,-1\}} \left\{ \sum_{i \in \text{neighbors}(x')} y_i \right\} \]

KNN Example

<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>1</td>
<td>yes</td>
<td>yes</td>
<td>clear</td>
<td>no</td>
<td>yes / +1</td>
</tr>
<tr>
<td>2</td>
<td>yes</td>
<td>yes</td>
<td>clear</td>
<td>no</td>
<td>yes / +1</td>
</tr>
<tr>
<td>3</td>
<td>partial</td>
<td>yes</td>
<td>unclear</td>
<td>no</td>
<td>no / -1</td>
</tr>
<tr>
<td>4</td>
<td>yes</td>
<td>yes</td>
<td>clear</td>
<td>yes</td>
<td>yes / +1</td>
</tr>
</tbody>
</table>

• How will new examples be classified?
  
  Similarity function?

  Value of \( k \)?

  \[ h(x') = \arg \max_{y \in \{+1,-1\}} \left\{ \sum_{i \in \text{neighbors}(x')} y_i \right\} \]
**Weighted K-Nearest Neighbor**

- **Given:** Training data \((x_1, y_1), ..., (x_n, y_n)\)
  - Attribute vectors: \(x_i \in X\)
  - 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: \(k\) training examples with largest \(K(x_i, x')\)

\[
\hat{h}(x') = \arg \max_{y \in \mathbb{Y}} \left\{ \sum_{i=1}^{k} \frac{1}{k} K(x_i, x') \right\}
\]

**Types of Attributes**

- **Symbolic (nominal)**
  - EyeColor: {brown, blue, green}
- **Boolean**
  - anemic: {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)**

- **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.

---

**Example: Effect of k**

---

**Supervised Learning (Concept Learning, Classification, Regression, etc.)**

- **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-Nearest Neighbor for Regression**

- **Given:** Training data \((x_1, y_1), ..., (x_n, y_n)\)
  - Attribute vectors: \(x_i \in X\)
  - 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: \(k\) training examples with largest \(K(x_i, x')\)

\[
\hat{h}(x') = \frac{\sum_{i=1}^{k} \frac{1}{k} K(x_i, x')}{\sum_{i=1}^{k} K(x_i, x')}
\]
Collaborative Filtering