Concept Learning

- 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

- 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.

K-Nearest Neighbor (KNN)

- Given: Training data \( \langle \vec{x}_1, y_1 \rangle, \ldots, \langle \vec{x}_n, y_n \rangle \)
  - 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 \( \vec{x} \):
  - K-nearest neighbors: \( \vec{x} \) train examples with largest \( K(\vec{x}, \vec{x}^\prime) \)

\[
h(\vec{x}^\prime) = \arg \max_{\vec{y} \in Y} \left\{ \sum_{i \in kNN(\vec{x}^\prime)} 1_{\vec{y}_i = \vec{y}} \right\}
\]

KNN Example

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

\[
h(\vec{x}^\prime) = \arg \max_{\vec{y} \in Y} \left\{ \sum_{i \in kNN(\vec{x}^\prime)} 1_{\vec{y}_i = \vec{y}} \right\}
\]
Weighted K-Nearest Neighbor

- Given: Training data \( \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \)
  - Attribute vectors: \( \mathbf{x}_i \in \mathbb{X} \)
  - Target attribute: \( y_i \in \mathbb{Y} \)
- Parameter:
  - Similarity function: \( K: \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R} \)
  - Number of nearest neighbors to consider: \( k \)
- Prediction rule
  - New example \( x' \)
  - K-nearest neighbors: \( k \) train examples with largest \( K(\mathbf{x}_i, x') \)

\[
h(x') = \arg \max_{y' \in \mathbb{Y}} \left\{ \sum_{i \in kNN(x')} 1[y_i = y] K(\mathbf{x}_i, x') \right\}
\]

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)

Supervised Learning

- Task:
  - Learn (to imitate) a function \( f: \mathbb{X} \rightarrow \mathbb{Y} \)
- 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 \( h: \mathbb{X} \rightarrow \mathbb{Y} \)
    that approximates \( f: \mathbb{X} \rightarrow \mathbb{Y} \)
    as well as possible.

Example: Effect of \( k \)

Weighted K-NN for Regression

- Given: Training data \( \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \)
  - Attribute vectors: \( \mathbf{x}_i \in \mathbb{X} \)
  - Target attribute: \( y_i \in \mathbb{Y} \)
- Parameter:
  - Similarity function: \( K: \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R} \)
  - Number of nearest neighbors to consider: \( k \)
- Prediction rule
  - New example \( x' \)
  - K-nearest neighbors: \( k \) train examples with largest \( K(\mathbf{x}_i, x') \)

\[
h(x') = \frac{\sum_{i \in kNN(x')} y_i K(\mathbf{x}_i, x')}{\sum_{i \in kNN(x')} K(\mathbf{x}_i, x')}
\]
### 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>1</td>
<td>5</td>
<td></td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>$u_2$</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$u_3$</td>
<td>2</td>
<td>4</td>
<td></td>
<td>1</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>