Classification

Nearest Neighbor
### Instance based classifiers

#### Set of Stored Cases

<table>
<thead>
<tr>
<th>Atr1</th>
<th>........</th>
<th>AtrN</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B</td>
</tr>
</tbody>
</table>

- Store the training samples
- Use training samples to predict the class label of unseen samples

#### Unseen Case

<table>
<thead>
<tr>
<th>Atr1</th>
<th>........</th>
<th>AtrN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Instance based classifiers

- Examples:
  - Rote learner
    - Memorize entire training data
    - Perform classification only if attributes of test sample match one of the training samples exactly
  - Nearest neighbor
    - Use $k$ “closest” samples (nearest neighbors) to perform classification
Nearest neighbor classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it’s probably a duck

- compute distance
- test sample
- choose k of the “nearest” samples
- training samples
Nearest neighbor classifiers

Requires three inputs:

1. The set of stored samples
2. Distance metric to compute distance between samples
3. The value of $k$, the number of nearest neighbors to retrieve
Nearest neighbor classifiers

To classify unknown record:
1. Compute distance to other training records
2. Identify $k$ nearest neighbors
3. Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)
Definition of nearest neighbor

(a) 1-nearest neighbor
(b) 2-nearest neighbor
(c) 3-nearest neighbor

\( k \)-nearest neighbors of a sample \( x \) are datapoints that have the \( k \) smallest distances to \( x \)
1-nearest neighbor

Voronoï diagram
Nearest neighbor classification

- Compute distance between two points:
  - Euclidean distance
    \[ d(x, y) = \sqrt{\sum_i (x_i - y_i)^2} \]

- Options for determining the class from nearest neighbor list
  - Take majority vote of class labels among the \( k \)-nearest neighbors
  - Weight the votes according to distance
    - example: weight factor \( w = 1 / d^2 \)
Choosing the value of $k$:

- If $k$ is too small, sensitive to noise points.
- If $k$ is too large, neighborhood may include points from other classes.
Nearest neighbor classification

- Scaling issues
  - Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
  - Example:
    - height of a person may vary from 1.5 m to 1.8 m
    - weight of a person may vary from 90 lb to 300 lb
    - income of a person may vary from $10K to $1M
Nearest neighbor classification...

- Problem with Euclidean measure:
  - High dimensional data
    - curse of dimensionality
  - Can produce counter-intuitive results

\[
\begin{align*}
\text{1 1 1 1 1 1 1 1 1 1 0} & \quad \text{vs} \quad \text{1 0 0 0 0 0 0 0 0 0 0} \\
\text{0 1 1 1 1 1 1 1 1 1 1} & \quad \text{d = 1.4142} \\
\text{0 0 0 0 0 0 0 0 0 0 1} & \quad \text{d = 1.4142}
\end{align*}
\]

- one solution: normalize the vectors to unit length
Nearest neighbor classification

- \( k \)-Nearest neighbor classifier is a lazy learner
  - Does not build model explicitly.
  - Unlike eager learners such as decision tree induction and rule-based systems.
  - Classifying unknown samples is relatively expensive.
- \( k \)-Nearest neighbor classifier is a local model, vs. global model of linear classifiers.
Example: PEBLS

- **PEBLS: Parallel Examplar-Based Learning System (Cost & Salzberg)**
  - Works with both continuous and nominal features
    - For nominal features, distance between two nominal values is computed using modified value difference metric (MVDM)
  - Each sample is assigned a weight factor
  - Number of nearest neighbor, \( k = 1 \)
Example: PEBLS

<table>
<thead>
<tr>
<th>Tid</th>
<th>Refund</th>
<th>Marital Status</th>
<th>Taxable Income</th>
<th>Cheat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>Single</td>
<td>125K</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>Married</td>
<td>100K</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
<td>Single</td>
<td>70K</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Married</td>
<td>120K</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>No</td>
<td>Divorced</td>
<td>95K</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>No</td>
<td>Married</td>
<td>60K</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Yes</td>
<td>Divorced</td>
<td>220K</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>No</td>
<td>Single</td>
<td>85K</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>No</td>
<td>Married</td>
<td>75K</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>No</td>
<td>Single</td>
<td>90K</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Distance between nominal attribute values:

\[ d(\text{Single, Married}) = |2/4 - 0/4| + |2/4 - 4/4| = 1 \]

\[ d(\text{Single, Divorced}) = |2/4 - 1/2| + |2/4 - 1/2| = 0 \]

\[ d(\text{Married, Divorced}) = |0/4 - 1/2| + |4/4 - 1/2| = 1 \]

\[ d(\text{Refund=Yes, Refund=No}) = |0/3 - 3/7| + |3/3 - 4/7| = 6/7 \]

\[ d(V_1, V_2) = \sum_i \left| \frac{n_{1i}}{n_1} - \frac{n_{2i}}{n_2} \right| \]

<table>
<thead>
<tr>
<th>Class</th>
<th>Marital Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Single  Married Divorced</td>
</tr>
<tr>
<td>Yes</td>
<td>2       0             1</td>
</tr>
<tr>
<td>No</td>
<td>2       4             1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Refund</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>0      3</td>
</tr>
<tr>
<td>No</td>
<td>3      4</td>
</tr>
</tbody>
</table>
Example: PEBLS

<table>
<thead>
<tr>
<th>Tid</th>
<th>Refund</th>
<th>Marital Status</th>
<th>Taxable Income</th>
<th>Cheat</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Yes</td>
<td>Single</td>
<td>125K</td>
<td>No</td>
</tr>
<tr>
<td>Y</td>
<td>No</td>
<td>Married</td>
<td>100K</td>
<td>No</td>
</tr>
</tbody>
</table>

Distance between record X and record Y:

\[ \Delta(X, Y) = w_X w_Y \sum_{i=1}^{d} d(X_i, Y_i)^2 \]

where:

\[ w_X = \frac{\text{Number of times X is used for prediction}}{\text{Number of times X predicts correctly}} \]

\[ w_X \approx 1 \text{ if X makes accurate prediction most of the time} \]

\[ w_X > 1 \text{ if X is not reliable for making predictions} \]
Decision boundaries in global vs. local models

- **linear regression**
  - global
  - stable
  - can be inaccurate

- **15-nearest neighbor**
  - local
  - accurate
  - unstable

- **1-nearest neighbor**

What ultimately matters: **GENERALIZATION**