Inductive learning hypothesis: any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over any other unobserved examples.

Assumptions for Inductive Learning Algorithms:
- The training sample represents the population
- The input features permit discrimination

The hypothesis produced is sometimes called the concept description — essentially a program that can be used to classify subsequent instances.

\[ k \text{-nearest neighbor} \]
Also called instance-based Learning; case-based learning.

A: set of features/attributes, \( A_1, \ldots, A_n \) that describe the problem

\[ x = x_{a_1} x_{a_2} \ldots x_{a_n}, \text{ where } x_{a_i} \text{ is the value of feature } A_i \text{ in example } x \]

\[ f(x) : x \rightarrow c \in C = \{ c_1, \ldots, c_m \} \]

The case base is the set of training examples \( (x_1, f(x_1)), (x_2, f(x_2)), \ldots \)

\[ k \text{-nearest neighbor algorithm for computing } f(x): \]
1. Compare new example, \( x \), to each case, \( y_i \), in the case base and calculate for each pair:
   \[ \text{sim}(x, y) = \sum_{i=1}^{n} \text{match}(x_{a_i}, y_{a_i}) \]
   where \( \text{match}(a, b) \) is a function that returns 1 if \( a \) and \( b \) are equal and 0 otherwise.
2. Let \( R = \) the top \( k \) cases ranked according to \( \text{sim} \)
3. Return as \( f(x) \) the class, \( c \), that wins the majority vote among \( f(R_1), f(R_2), \ldots, f(R_k) \). Handle ties randomly.
Types of Attributes

1. Symbolic (nominal) – $\text{EyeColor} \in \{\text{brown, blue, green}\}$
2. Boolean – $\text{anemic} \in \{\text{TRUE, FALSE}\}$
3. Numeric (Integer, Real) – $\text{age} \in [0, 105]$

How do we compute the similarity between $\text{EyeColor} = \text{brown}$ and $\text{EyeColor} = \text{green}$?

Example of case retrieval for k-nn

<table>
<thead>
<tr>
<th>outlook</th>
<th>temp</th>
<th>humidity</th>
<th>windy</th>
<th>plan</th>
<th>sim</th>
</tr>
</thead>
<tbody>
<tr>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>cs472</td>
<td></td>
</tr>
<tr>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>true</td>
<td>cs472</td>
<td></td>
</tr>
<tr>
<td>overcast</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>soccer</td>
<td></td>
</tr>
<tr>
<td>overcast</td>
<td>mild</td>
<td>normal</td>
<td>true</td>
<td>football</td>
<td></td>
</tr>
<tr>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>false</td>
<td>soccer</td>
<td></td>
</tr>
<tr>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>false</td>
<td>soccer</td>
<td></td>
</tr>
</tbody>
</table>

$A$: outlook, temp, humidity, windy

$k = 1$, $C = \{\text{soccer, cs472 football}\}$

test case: $X = \text{sunny cool high false}$

$k$-Nearest Neighbor Algorithm

1. Memorizes all observed instances and their class
2. Is this rote learning?
3. Is this really learning?
4. When does the induction take place?

Advantages and Disadvantages

What constitutes the concept description?