DATA MINING 2
Instance-based Classifiers

Riccardo Guidotti
a.a. 2020/2021

Slides edited from Tan, Steinbach, Kumar, Introduction to Data Mining
Instance-based Classifiers

• Instead of performing explicit generalization, compare new instances with instances seen in training, which have been stored in memory.

• Sometimes called *memory-based* learning.

**Advantages**

• Adapt its model to previously unseen data by storing a new instance or throwing an old instance away.

**Disadvantages**

• Lazy learner: it does not build a model explicitly.

• Classifying unknown records is relatively expensive: in the worst case, given *n* training items, the complexity of classifying a single instance is *O*(*n*).
Nearest-Neighbor Classifier (K-NN)

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

Requires three things

1. **Training set** of stored records
2. **Distance metric** to compute distance between records
3. **The value of k**, the number of nearest neighbors to retrieve
Nearest-Neighbor Classifier (K-NN)

Given a set of training records (memory), and a test record:

1. **Compute the distances** from the records in the training to the test.
2. **Identify the k “nearest” records**.
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

• $K$-nearest neighbors of a record $x$ are data points that have the $k$ smallest distance to $x$.
Choosing the Value of K

• If $k$ is too small, it is sensitive to noise points and it can lead to overfitting to the noise in the training set.

• If $k$ is too large, the neighborhood may include points from other classes.

• General practice $k = \sqrt{N}$ where $N$ is the number of samples in the training dataset.
Nearest Neighbor Classification

Compute distance between two points:
• Euclidean distance $d(p, q) = \sqrt{\sum_i (p_i - q_i)^2}$

Determine the class from nearest neighbors
• take the majority vote of class labels among the $k$ nearest neighbors
• weigh the vote according to distance (e.g. weight factor, $w = 1/d^2$)
Dimensionality and Scaling Issues

• Problem with Euclidean measure: high dimensional data can cause curse of dimensionality.
  • Solution: normalize the vectors to unit length

• 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 1.5m to 1.8m
  • weight of a person may vary from 10kg to 200kg
  • income of a person may vary from $10K to $1M
Parallel Exemplar-Based Learning System (PEBLS)

- PEBLS is a nearest-neighbor learning system (k=1) designed for applications where the instances have symbolic feature values.
- Works with both continuous and nominal features.
- For nominal features, the distance between two nominal values is computed using Modified Value Difference Metric (MVDM)
  \[ d(V_1, V_2) = \sum_i \left| \frac{n_{1i}}{n_1} - \frac{n_{2i}}{n_2} \right| \]
- Where \( n_1 \) is the number of records that consists of nominal attribute value \( V_1 \) and \( n_{1i} \) is the number of records whose target label is class \( i \).
Distance Between Nominal Attribute Values

- $d(\text{Status}=\text{Single}, \text{Status}=\text{Married}) = |2/4 - 0/4| + |2/4 - 4/4| = 1$
- $d(\text{Status}=\text{Single}, \text{Status}=\text{Divorced}) = |2/4 - 1/2| + |2/4 - 1/2| = 0$
- $d(\text{Status}=\text{Married}, \text{Status}=\text{Divorced}) = |0/4 - 1/2| + |4/4 - 1/2| = 1$
- $d(\text{Refund}=\text{Yes}, \text{Refund}=\text{No}) = |0/3 - 3/7| + |3/3 - 4/7| = 6/7$

<table>
<thead>
<tr>
<th>Class</th>
<th>marital status</th>
<th></th>
<th>Tid</th>
<th>Refund</th>
<th>Marital Status</th>
<th>Taxable Income</th>
<th>Cheat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>Yes</td>
<td>Single</td>
<td>125K</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>No</td>
<td>Married</td>
<td>100K</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Single</td>
<td>70K</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>Married</td>
<td>120K</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>Divorced</td>
<td>95K</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Divorced</td>
<td>60K</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>Married</td>
<td>220K</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>Single</td>
<td>85K</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>Married</td>
<td>75K</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>Single</td>
<td>90K</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Distance Between Records

\[ \delta(X, Y) = w_X w_Y \sum_{i=0}^{d} d(X_i, Y_i) \]

• Each record \( X \) is assigned a weight \( w_X = \frac{N_{X_{predict}}}{N_{X_{correct}}} \), which represents its reliability.

• \( N_{X_{predict}} \) is the number of times \( X \) is used for prediction.

• \( N_{X_{correct}} \) is the number of times the prediction using \( X \) is correct.

• If \( w_X \approx 1 \), \( X \) makes accurate prediction most of the time.

• If \( w_X > 1 \), then \( X \) is not reliable for making predictions. High \( w_X > 1 \) would result in high distance, which makes it less possible to use \( X \) to make predictions.
Characteristics of Nearest Neighbor Classifiers

• Instance-based learner: makes predictions without maintaining abstraction, i.e., building a model like decision trees.

• It is a lazy learner: classifying a test example can be expensive because need to compute the proximity values between test and training examples.

• In contrast eager learners spend time in building the model but then the classification is fast.

• Make their prediction on local information and for low $k$ they are susceptible to noise.

• Can produce wrong predictions if inappropriate distance functions and/or preprocessing steps are performed.
References

• Nearest Neighbor classifiers. Chapter 5.2. Introduction to Data Mining.
Exercises- kNN
b) k-NN (3 points)

Given the training set on the right, composed of elements numbered from 1 to 12, and labelled as circles and diamonds, use it to classify the remaining 3 elements (letters A, B and C) using a k-NN classifier with k=3. For each point to classify, list the points of the dataset that belong to its k-NN set.

Notice: A, B and C belong to the test set, not to the training set. Also, the Euclidean distance should be used.

Answer:
kNN(A) = \{3, 5, 12\} → CIRCLE
kNN(B) = \{3, 5, 7, 10\} → CIRCLE
kNN(C) = \{4, 6, 7\} → CIRCLE
Given the training set on the right, composed of elements numbered from 1 to 12, and labelled as circles and diamonds, use it to classify the remaining 3 elements (letters A, B and C) using a k-NN classifier with k=3. For each point to classify, list the points of the dataset that belong to its k-NN set. Notice: A, B and C belong to the test set, not to the training set. Also, the Euclidean distance should be used.
**k-Nearest Neighbor Classifier**

A medical expert is going to build up a case-based reasoning system for diagnosis tasks. Cases correspond to individual persons where the case problem parts are made up of a number of features describing possible symptoms and the solution parts represent the diagnosis (classification of disease). The case base contains the seven cases provided in the table below.

<table>
<thead>
<tr>
<th>Training</th>
<th>Fever</th>
<th>Vomiting</th>
<th>Diarrhea</th>
<th>Shivering</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_1$</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>healthy (H)</td>
</tr>
<tr>
<td>$c_2$</td>
<td>average</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>influenza (I)</td>
</tr>
<tr>
<td>$c_3$</td>
<td>high</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>influenza (I)</td>
</tr>
<tr>
<td>$c_4$</td>
<td>high</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>salmonella poisoning (S)</td>
</tr>
<tr>
<td>$c_5$</td>
<td>average</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>salmonella poisoning (S)</td>
</tr>
<tr>
<td>$c_6$</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>bowel inflammation (B)</td>
</tr>
<tr>
<td>$c_7$</td>
<td>average</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>bowel inflammation (B)</td>
</tr>
</tbody>
</table>

**Similarity provided by an expert**

\[
\begin{array}{ccc}
q & c & \text{no} & \text{avg} & \text{high} \\
\text{no} & 1.0 & 0.7 & 0.2 \\
\text{avg} & 0.5 & 1.0 & 0.8 \\
\text{high} & 0.0 & 0.3 & 1.0
\end{array}
\]

\[
\begin{array}{ccc}
q & \text{yes} & \text{no} \\
\text{yes} & 1.0 & 0.0 \\
\text{no} & 0.2 & 1.0
\end{array}
\]

Weights:
\[
\begin{align*}
W_F &= 0.3 \\
W_V &= 0.2 \\
W_D &= 0.2 \\
W_{Sh} &= 0.3
\end{align*}
\]
Classify the new instance $q = (\text{high}; \text{no}; \text{no}; \text{no})$ by applying the KNN algorithm with $K=1, 2, 3$
Calculate the similarity between all cases from the case base and the new instance $q = (\text{high}; \text{no}; \text{no}; \text{no})$

$c_1 = (\text{no}; \text{no}; \text{no}; \text{no})$:

$\text{Sim}(q; c_1) = 0.3*0.0 + 0.2*1.0 + 0.2*1.0 + 0.3*1.0 = 0.70$

$c_2 = (\text{average}; \text{no}; \text{no}; \text{no})$:

$\text{Sim}(q; c_2) = 0.3*0.3 + 0.2*1.0 + 0.2*1.0 + 0.3*1.0 = 0.79$

$c_3 = (\text{high}; \text{no}; \text{no}; \text{yes})$

$\text{Sim}(q; c_3) = 0.3*1.0 + 0.2*1.0 + 0.2*1.0 + 0.3*0.2 = 0.76$

$c_4 = (\text{high}; \text{yes}; \text{yes}; \text{no})$:

$\text{Sim}(q; c_4) = 0.3*1.0 + 0.2*0.2 + 0.2*0.2 + 0.3*1.0 = 0.68$

$c_5 = (\text{average}; \text{no}; \text{yes}; \text{no})$:

$\text{Sim}(q; c_5) = 0.3*0.3 + 0.2*1.0 + 0.2*0.2 + 0.3*1.0 = 0.63$

$c_6 = (\text{no}; \text{yes}; \text{yes}; \text{no})$:

$\text{Sim}(q; c_6) = 0.3*0.0 + 0.2*0.2 + 0.2*0.2 + 0.3*1.0 = 0.28$

$c_7 = (\text{average}; \text{yes}; \text{yes}; \text{no})$:

$\text{Sim}(q; c_7) = 0.3*0.3 + 0.2*0.2 + 0.2*0.2 + 0.3*1.0 = 0.47$
KNN Classification for K=1

c1 = (no; no; no; no):
Sim(q; c1) = 0.3*0.0 + 0.2 *1.0 + 0.2*1.0 + 0.3* 1.0 = 0.70

c2 = (average; no; no; no):
Sim(q; c2) = 0.3* 0.3 + 0.2 *1.0 + 0.2*1.0 + 0.3*1.0 = 0.79

c3 = (high; no; no; yes)
Sim(q; c3) = 0.3*1.0 + 0.2*1.0 + 0.2*1.0 + 0.3*0.2 = 0.76

c4 = (high; yes; yes; no):
Sim(q; c4) = 0.3*1.0 + 0.2*0.2 + 0.2*0.2 + 0.3*1.0 = 0.68

c5 = (average; no; yes; no):
Sim(q; c5) = 0.3*0.3 + 0.2*1.0 + 0.2*0.2 + 0.3*1.0 = 0.63

c6 = (no; yes; yes; no):
Sim(q; c6) = 0.3*0.0 + 0.2*0.2 + 0.2*0.2 + 0.3*1.0 = 0.28

c7 = (average; yes; yes; no):
Sim(q; c7) = 0.3*0.3 + 0.2*0.2 + 0.2*0.2 + 0.3*1.0 = 0.47

Class: Influenza
KNN Classification for K=2

\[ c_1 = \text{(no; no; no; no):} \]
\[ \text{Sim}(q; c_1) = 0.3 \times 0.0 + 0.2 \times 1.0 + 0.2 \times 1.0 + 0.3 \times 1.0 = 0.70 \]

\[ c_2 = \text{(average; no; no; no):} \]
\[ \text{Sim}(q; c_2) = 0.3 \times 0.3 + 0.2 \times 1.0 + 0.2 \times 1.0 + 0.3 \times 1.0 = 0.79 \]

\[ c_3 = \text{(high; no; no; yes):} \]
\[ \text{Sim}(q; c_3) = 0.3 \times 1.0 + 0.2 \times 1.0 + 0.2 \times 1.0 + 0.3 \times 0.2 = 0.76 \]

\[ c_4 = \text{(high; yes; yes; no):} \]
\[ \text{Sim}(q; c_4) = 0.3 \times 1.0 + 0.2 \times 0.2 + 0.2 \times 0.2 + 0.3 \times 1.0 = 0.68 \]

\[ c_5 = \text{(average; no; yes; no):} \]
\[ \text{Sim}(q; c_5) = 0.3 \times 0.3 + 0.2 \times 1.0 + 0.2 \times 0.2 + 0.3 \times 1.0 = 0.63 \]

\[ c_6 = \text{(no; yes; yes; no):} \]
\[ \text{Sim}(q; c_6) = 0.3 \times 0.0 + 0.2 \times 0.2 + 0.2 \times 0.2 + 0.3 \times 1.0 = 0.28 \]

\[ c_7 = \text{(average; yes; yes; no):} \]
\[ \text{Sim}(q; c_7) = 0.3 \times 0.3 + 0.2 \times 0.2 + 0.2 \times 0.2 + 0.3 \times 1.0 = 0.47 \]
KNN Classification for K=3

**c1 = (no; no; no; no):**
Sim(q; c1) = 0.3*0.0 + 0.2 *1.0 + 0.2*1.0 + 0.3* 1.0 = 0.70

**c2 = (average; no; no; no):**
Sim(q; c2) = 0.3 * 0.3 + 0.2 *1.0 + 0.2*1.0 + 0.3*1.0 = 0.79

**c3 = (high; no; no; yes):**
Sim(q; c3) = 0.3*1.0 + 0.2*1.0 + 0.2*1.0 + 0.3*0.2 = 0.76

**c4 = (high; yes; yes; no):**
Sim(q; c4) = 0.3*1.0 + 0.2*0.2 + 0.2*0.2 + 0.3*1.0 = 0.68

**c5 = (average; no; yes; no):**
Sim(q; c5) = 0.3*0.3 + 0.2*1.0 + 0.2*0.2 + 0.3*1.0 = 0.63

**c6 = (no; yes; yes; no):**
Sim(q; c6) = 0.3*0.0 + 0.2*0.2 + 0.2*0.2 + 0.3*1.0 = 0.28

**c7 = (average; yes; yes; no):**
Sim(q; c7) = 0.3*0.3 + 0.2*0.2 + 0.2*0.2 + 0.3*1.0 = 0.47

**Weights**
- W_F = 0.3
- W_V = 0.2
- W_D = 0.2
- W_Sh = 0.3

**Class:** Influenza