Nearest Neighbor Classifiers

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

- Requires the following:
  - A set of labeled records
  - Proximity metric to compute distance/similarity between a pair of records (e.g., Euclidean distance)
  - The value of $k$, the number of nearest neighbors to retrieve
  - A method for using class labels of $K$ nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

How to Determine the class label of a Test Sample?

- Take the majority vote of class labels among the $k$-nearest neighbors
- Weight the vote according to distance
  - weight factor, $w = 1/d^2$
Choice of proximity measure matters

- For documents, cosine is better than correlation or Euclidean

| 1 1 1 1 1 1 1 1 1 0 | vs | 0 0 0 0 0 0 0 0 0 1 |
| 0 1 1 1 1 1 1 1 1 1 |   | 1 0 0 0 0 0 0 0 0 0 |

Euclidean distance = 1.4142 for both pairs, but the cosine similarity measure has different values for these pairs.

Nearest Neighbor Classification...

- 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...

- **Data preprocessing is often required**
  - 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 90lb to 300lb
      - income of a person may vary from $10K to $1M
  - Time series are often standardized to have 0 means a standard deviation of 1

Nearest-neighbor classifiers

- Nearest neighbor classifiers are local classifiers
- They can produce decision boundaries of arbitrary shapes.
Nearest Neighbor Classification...

- **How to handle missing values in training and test sets?**
  - Proximity computations normally require the presence of all attributes
  - Some approaches use the subset of attributes present in two instances
    - This may not produce good results since it effectively uses different proximity measures for each pair of instances
    - Thus, proximities are not comparable

- **Handling irrelevant and redundant attributes**
  - Irrelevant attributes add noise to the proximity measure
  - Redundant attributes bias the proximity measure towards certain attributes
  - Can use variable selection or dimensionality reduction to address irrelevant and redundant attributes
Improving KNN Efficiency

- Avoid having to compute distance to all objects in the training set
  - Multi-dimensional access methods (k-d trees)
  - Fast approximate similarity search
  - Locality Sensitive Hashing (LSH)
- Condensing
  - Determine a smaller set of objects that give the same performance
- Editing
  - Remove objects to improve efficiency