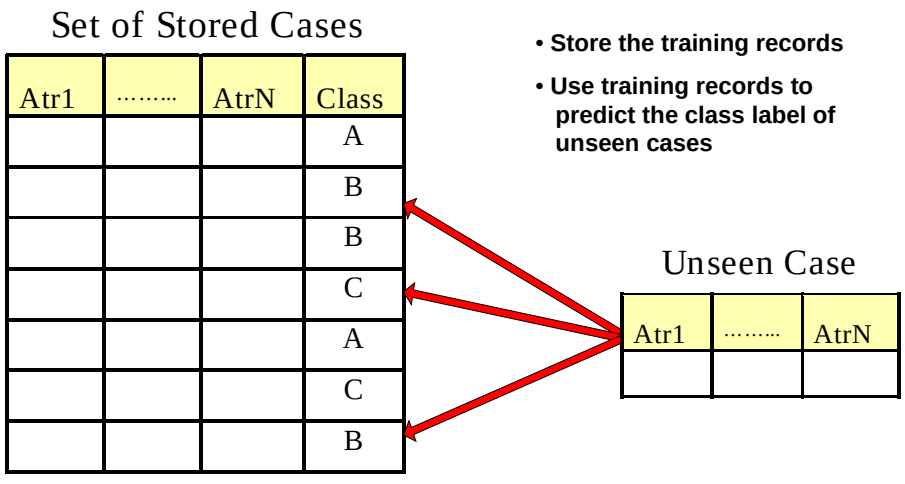
Instance-based classification

- Function-based classification
  - Decision trees, Bayes nets, neural nets provide parameterized models of the data
  - One can discard training data once model in hand
- Instance-based classification
  - Use the data (or a subset) as its own model
  - A "field theory" of data

Instance-Based Classifiers



Eager versus lazy classification

- Eager: construct model before classifying
  - Choose global approximation independent of instance to be classified
  - Can require considerable time to construct approximation
  - A single hypothesis covering all of the space of instances can have poor accuracy if data do not fit the form of the possible hypotheses
- Lazy: construct "model" while classifying
  - Less time training but more time predicting
  - Increase accuracy by combining many local functions to form global approximation over a richer hypothesis space

Instance-based classification

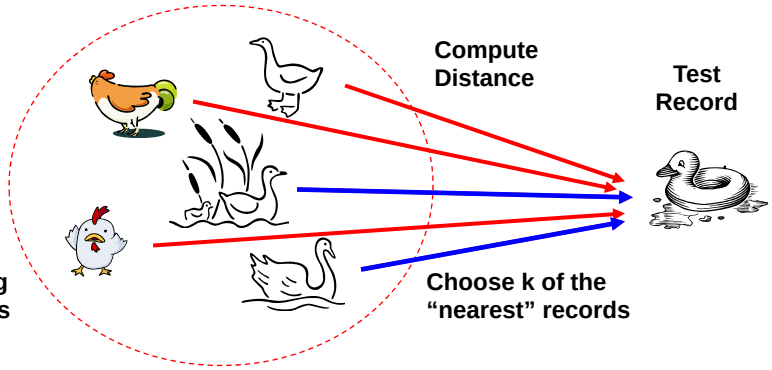
- The model of data consists of "prototypical" instances plus classification procedure
  - A set of stored instances summarize the training data
  - A classification procedure classifies new instances with respect to the "basis" of stored instances
- Instead of choosing new model in response to new training data, choose new stored instances
- Two approaches
  - Instances determine new global model (support-vector machines)
  - Instances themselves constitute only model

Instance-based methods

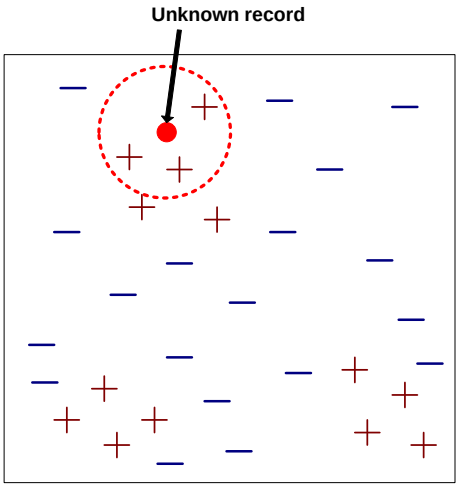
- k-nearest neighbor (k-NN) approach
  - Instances form points in a Euclidean space
  - Compare distance of instance to prototypes
- Locally weighted regression
  - Constructs local approximation
- Case-based reasoning
  - Instances form points in a non-Euclidean space of symbolic representations
  - Compare similarity of instance to prototypes

Nearest Neighbor Classifiers

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

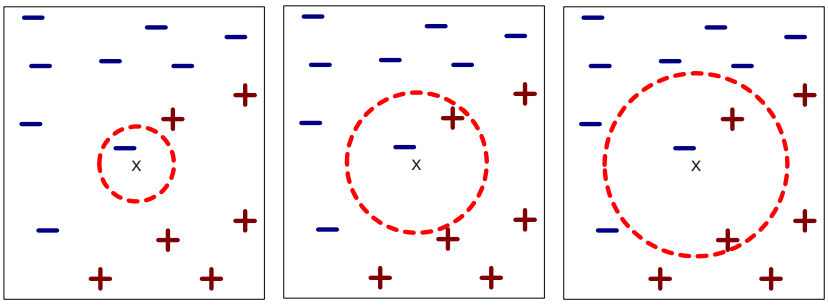


Nearest-Neighbor Classifiers



- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve
- To classify an unknown record:
  - Compute distance to other training records
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Definition of Nearest Neighbor

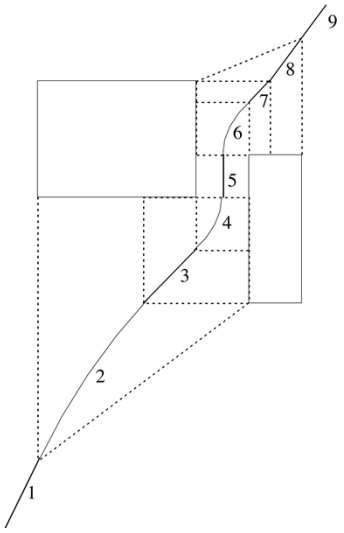


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

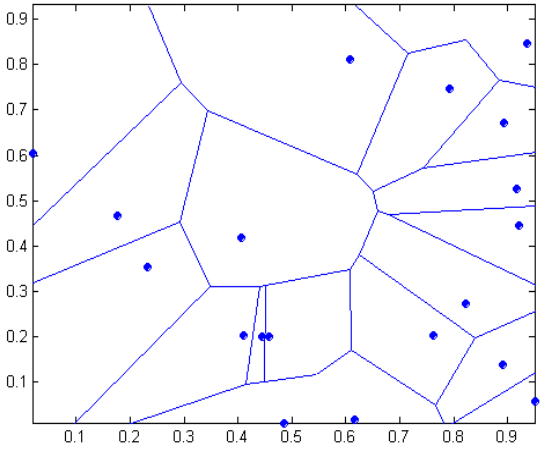
Nearest-neighbor classification

- Produces nonlinear class boundaries
  - For example, from hyperrectangles that cover instances
- Sometimes piecewise linear (as in Voronoi diagram)



1 nearest-neighbor

Voronoi Diagram



The k-nearest neighbor algorithm

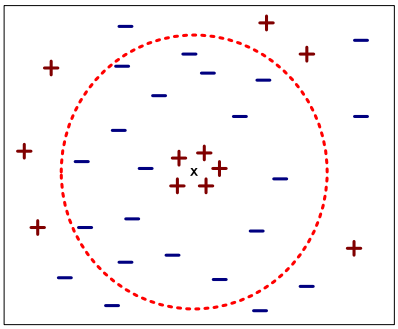
- Instances inhabit points in n-dimensional space
- Metric measures distances from new instances to training exemplars or sets
  - Discrete or continuous metric values
  - Euclidean or other metric function
    - Manhattan distance: $x_0 + \ldots + x_n$
    - Euclidean distance: $x_0^2 + \ldots + x_n^2$
    - Minkowski metrics: $x_0^p + \ldots + x_n^p$
- Nearest neighbor defined as the training instance or set closest to instance in distance

The k-nearest neighbor algorithm

- The continuous-metric k-NN algorithm predicts the mean values of the k nearest neighbors
- Distance-weighted nearest neighbor algorithm
  - Weight the contribution of each of the k neighbors according to their distance to the query point x_q
    - giving greater weight to closer neighbors
    - $w \equiv (d(x_q, x_i))^{-2}$
  - Similarly, for real-valued target functions

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



The k-nearest neighbor algorithm

- 1-NN fairly sensitive to noise
- k ≥ 3 reduces sensitivity significantly
- For large k and numbers of prototype instances, accuracy resembles that of Bayesian classification

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 1.5m to 1.8m
    - weight of a person may vary from 90lb to 300lb
    - 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

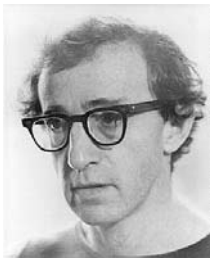
1 1 1 1 1 1 1 1 1 1 1 0		1 0 0 0 0 0 0 0 0 0 0 0	
0 1 1 1 1 1 1 1 1 1 1 1	vs	0 0 0 0 0 0 0 0 0 0 0 1	
d = 1.4142		d = 1.4142	

- Solution: Normalize the vectors to unit length

The curse of dimensionality

- In high dimensional spaces, contributions from irrelevant attributes can dominate the distance between neighbors
- Overcome the curse by stretching relevant axes or eliminating least relevant attributes
  - For example, project data onto principal components

Nearness in Euclidean Distance?






Case-based classification

- Instances represented by rich symbolic descriptions (e.g., function graphs)
  - Moves instance space out of Euclidean space
  - Indexing based on syntactic similarity measure
- Methodology
  - Tight coupling between case retrieval, knowledge-based reasoning, and problem solving
  - Multiple retrieved cases may be combined
  - Backtracking and adaptation to additional cases on match failure

Case-based classification

- Trade metric spaces for comparative similarity relations over complex representations
  - Can avoid some problems of high dimensionality
- Preorder $x \preceq_z y$ based at each point z of space
  - $x \preceq_z x$
  - $x \preceq_z y$ and $y \preceq_z w$ imply $x \preceq_z w$
  - Not necessarily antisymmetric
  - Not necessarily total orders
  - No comparisons of "distances" from distinct origins