



COMP7/8118 M50

Data Mining

Nearest Neighbor Classifier

Xiaofei Zhang

Slides compiled from Jiawei Han and Raymond C.W. Wong's work

THE UNIVERSITY OF
MEMPHIS

Instance-Based Classifiers

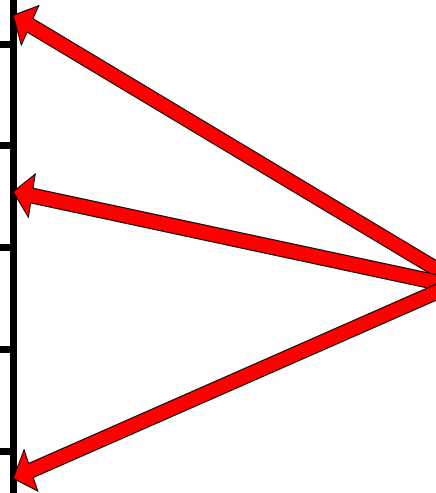
Set of Stored Cases

Atr1	AtrN	Class
			A
			B
			B
			C
			A
			C
			B

- Store the training records
- Use training records to predict the class label of unseen cases

Unseen Case

Atr1	AtrN

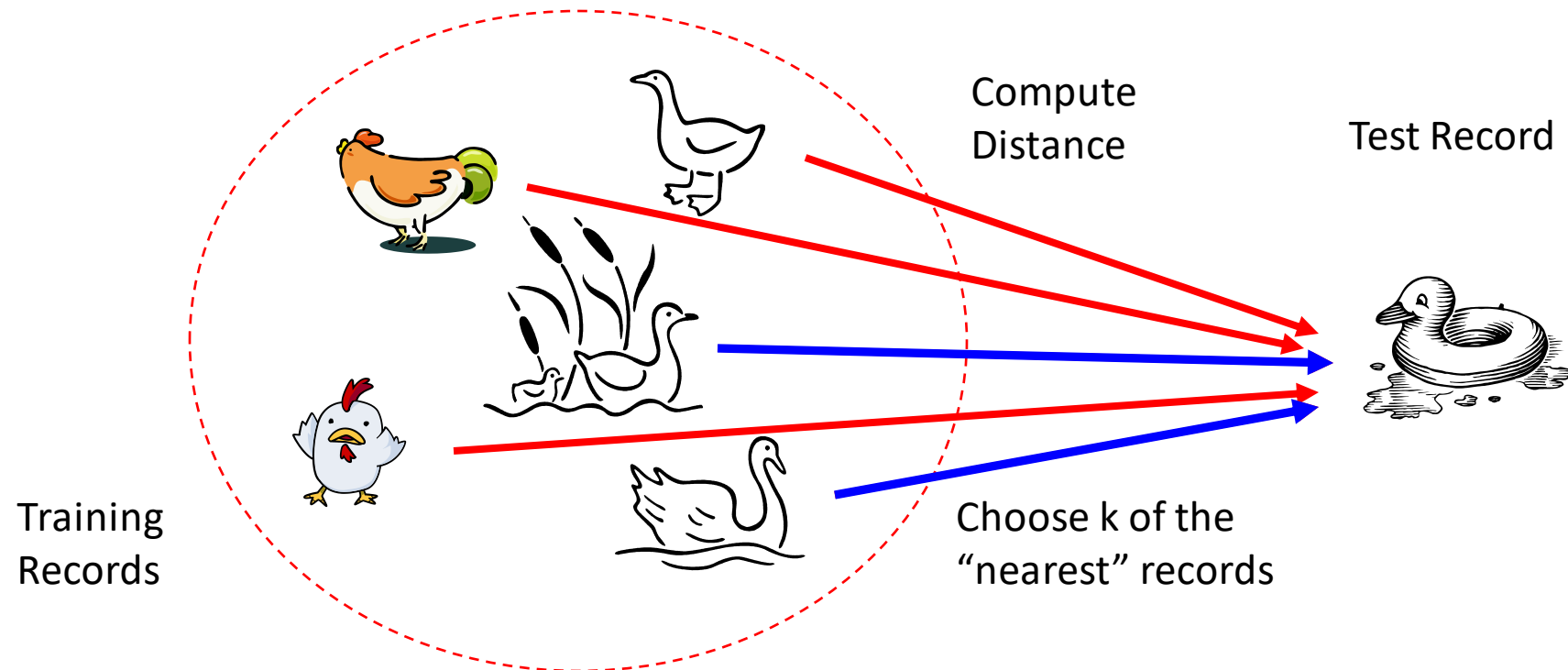


Instance Based Classifiers

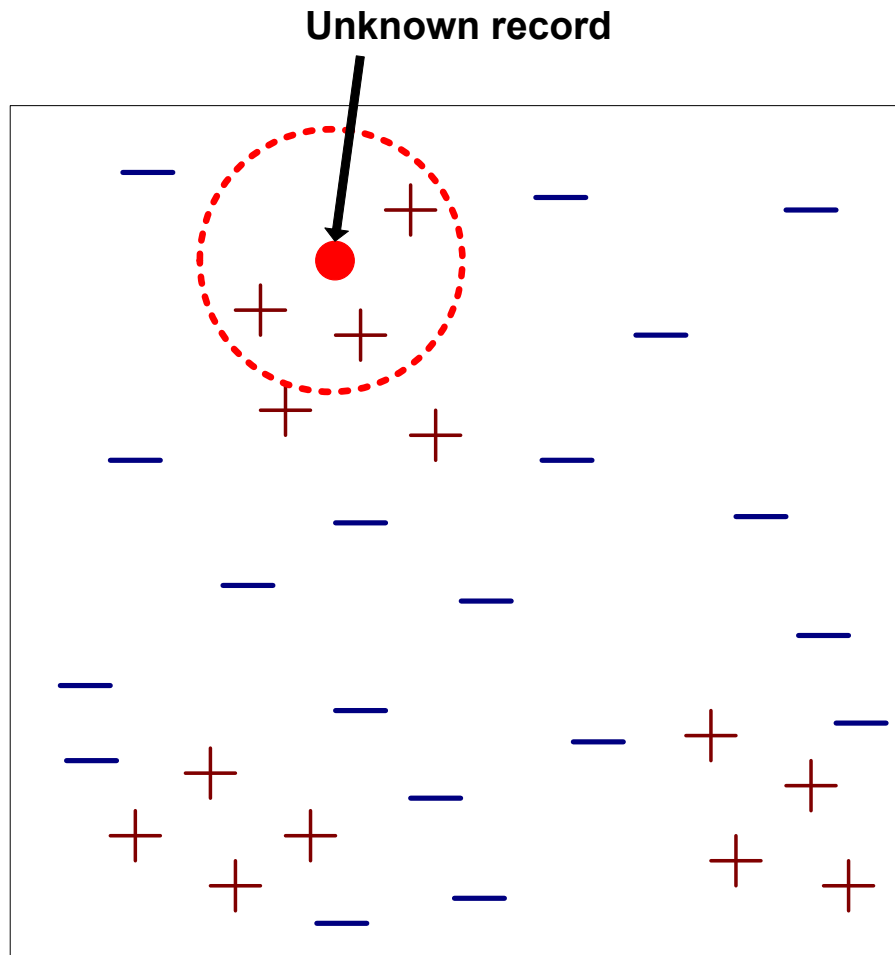
- Examples:
 - Rote-learner
 - Memorizes entire training data and performs classification only if attributes of record match one of the training examples exactly
 - Nearest neighbor classifier
 - Uses k “closest” points (nearest neighbors) for performing classification

Nearest Neighbor Classifiers

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

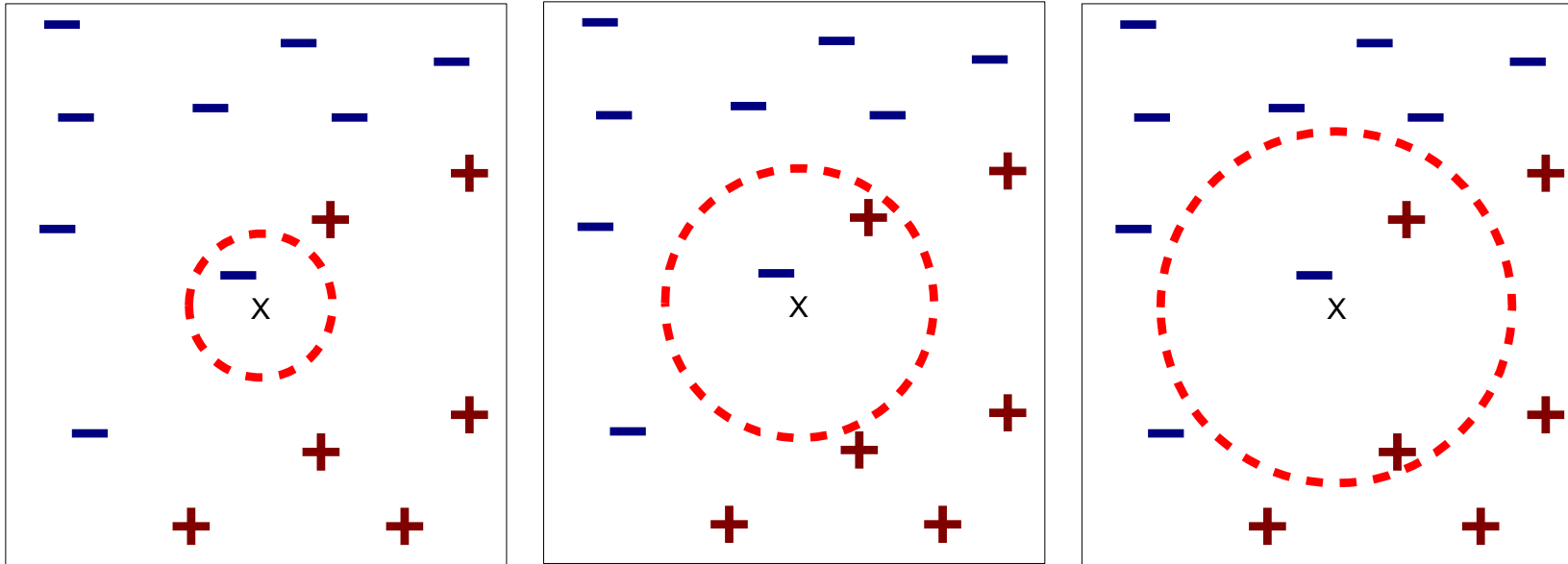


Nearest-Neighbor Classifiers



- 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:
 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

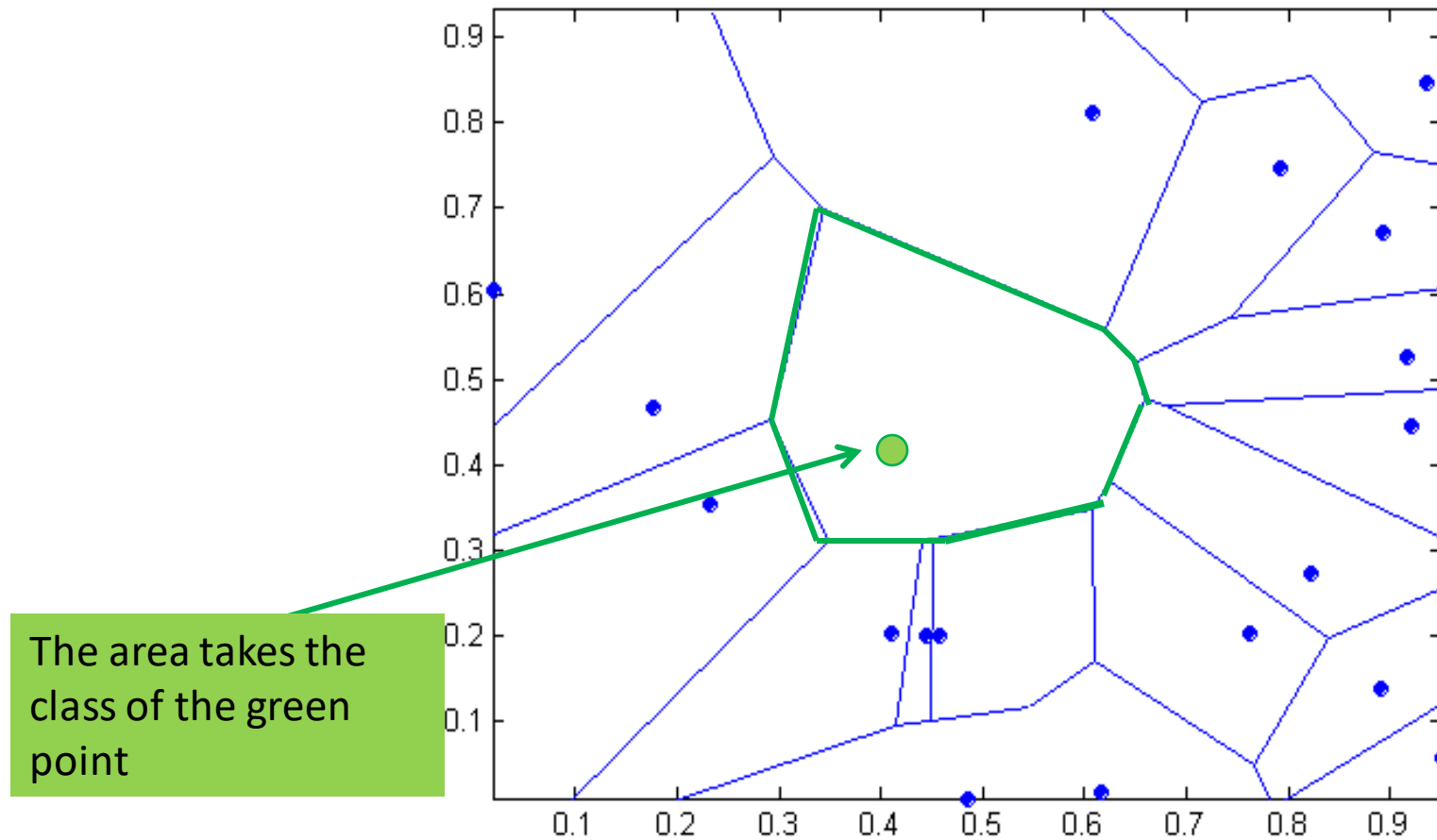
(b) 2-nearest neighbor

(c) 3-nearest neighbor

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

1 nearest-neighbor

Voronoi Diagram defines the classification boundary



Nearest Neighbor Classification

- Compute distance between two points:
 - Euclidean distance

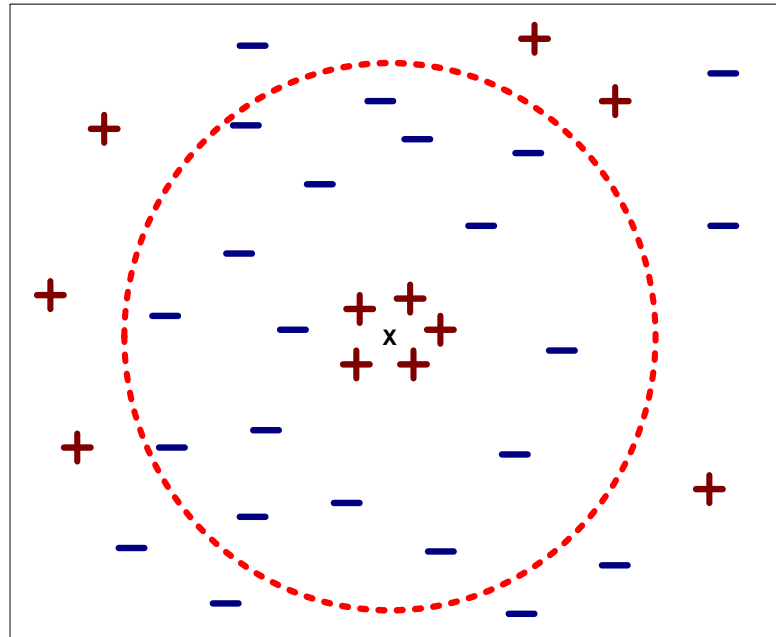
$$d(p, q) = \sqrt{\sum_i (p_i - q_i)^2}$$

- Determine the class from nearest neighbor list
 - take the majority vote of class labels among the k-nearest neighbors
 - Weigh the vote according to distance
 - weight factor, $w = 1/d^2$

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 value of k is the
complexity of the model



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

vs

1 0 0 0 0 0 0 0 0 0 0 0

0 1 1 1 1 1 1 1 1 1 1 1

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

Nearest neighbor Classification...

- k-NN classifiers are **lazy learners**
 - It does not build models explicitly
 - Unlike **eager learners** such as decision trees
- Classifying unknown records are relatively expensive
 - Naïve algorithm: $O(n)$
 - Need for **structures** to retrieve nearest neighbors fast.
 - The **Nearest Neighbor Search** problem.
 - Also, **Approximate Nearest Neighbor Search**