

Nearest-Neighbor Classifier

Instance-Based Classifiers

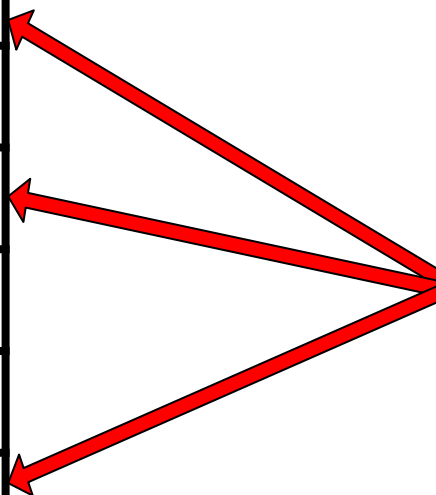
Set of Stored Cases

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

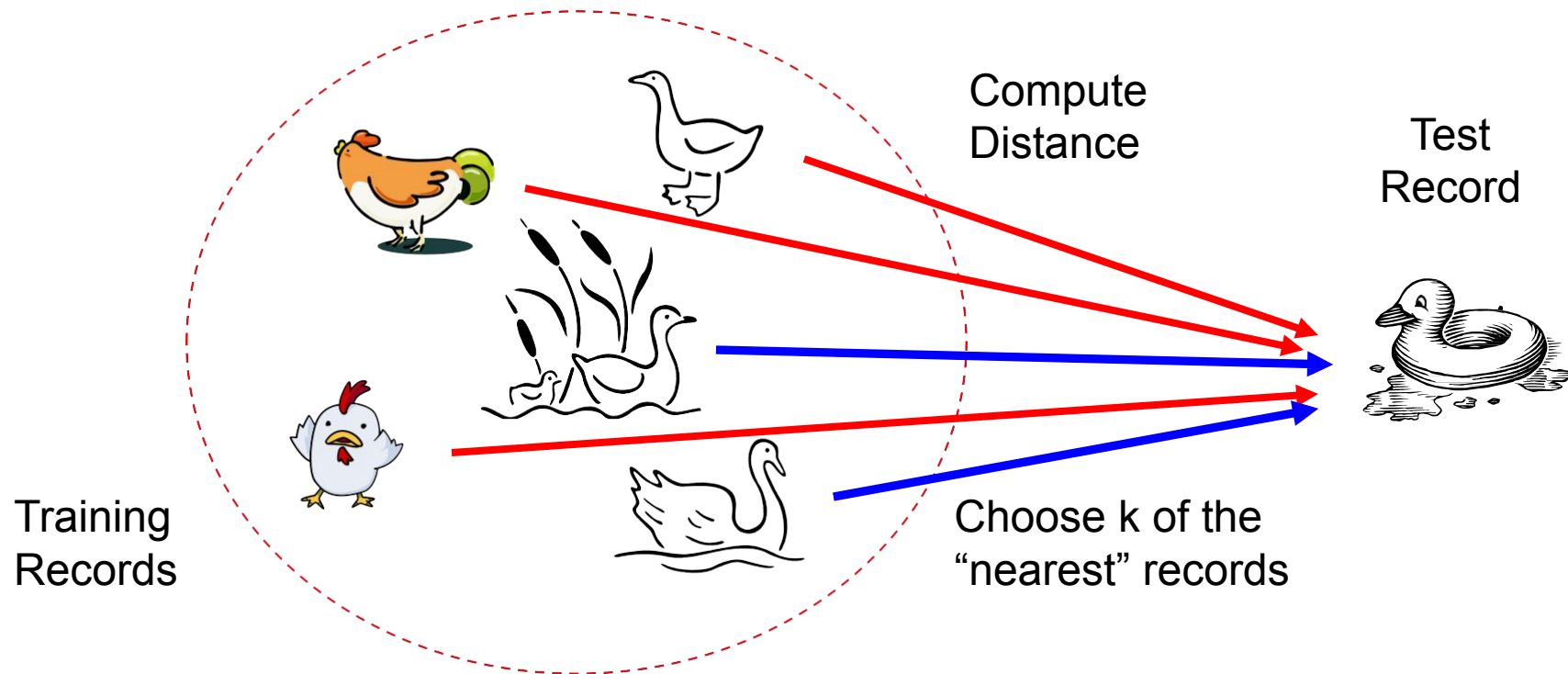


Instance Based Classifiers

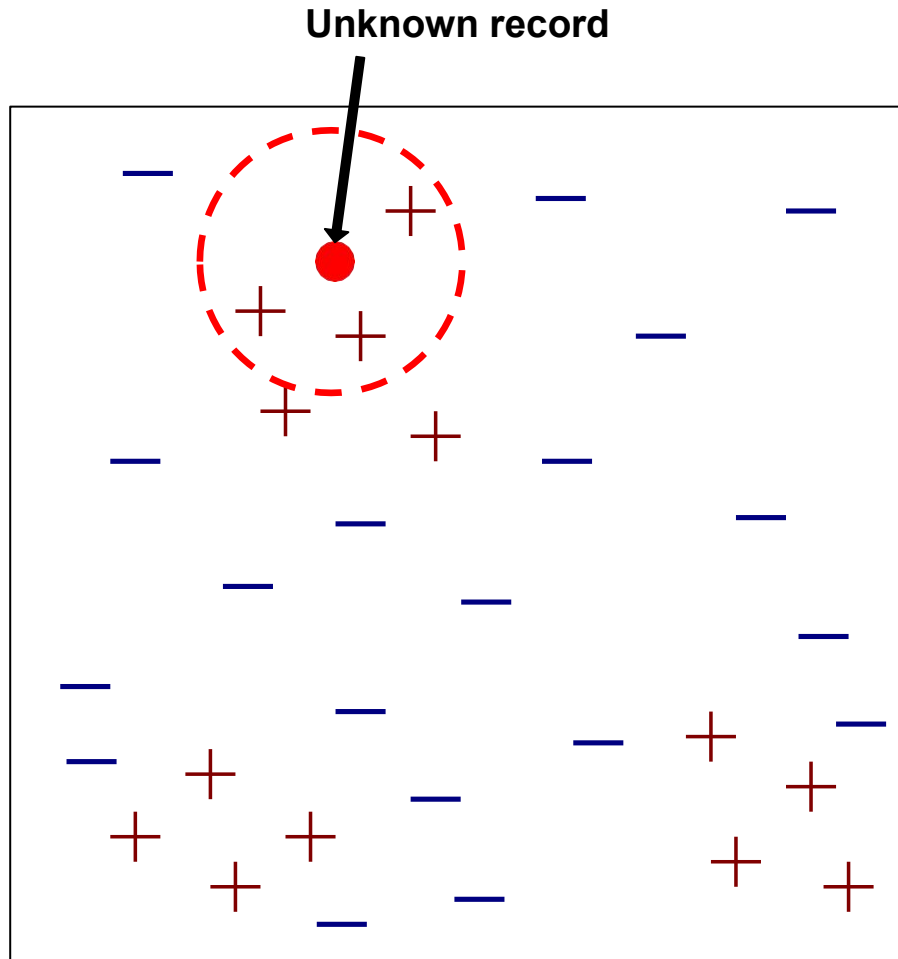
- Examples:
 - Rote-learner
 - Memorizes entire training data and performs classification only if attributes of record match one of the training examples exactly
 - Nearest neighbor
 - Uses k “closest” points (nearest neighbors) for performing classification

Nearest Neighbor Classifiers

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

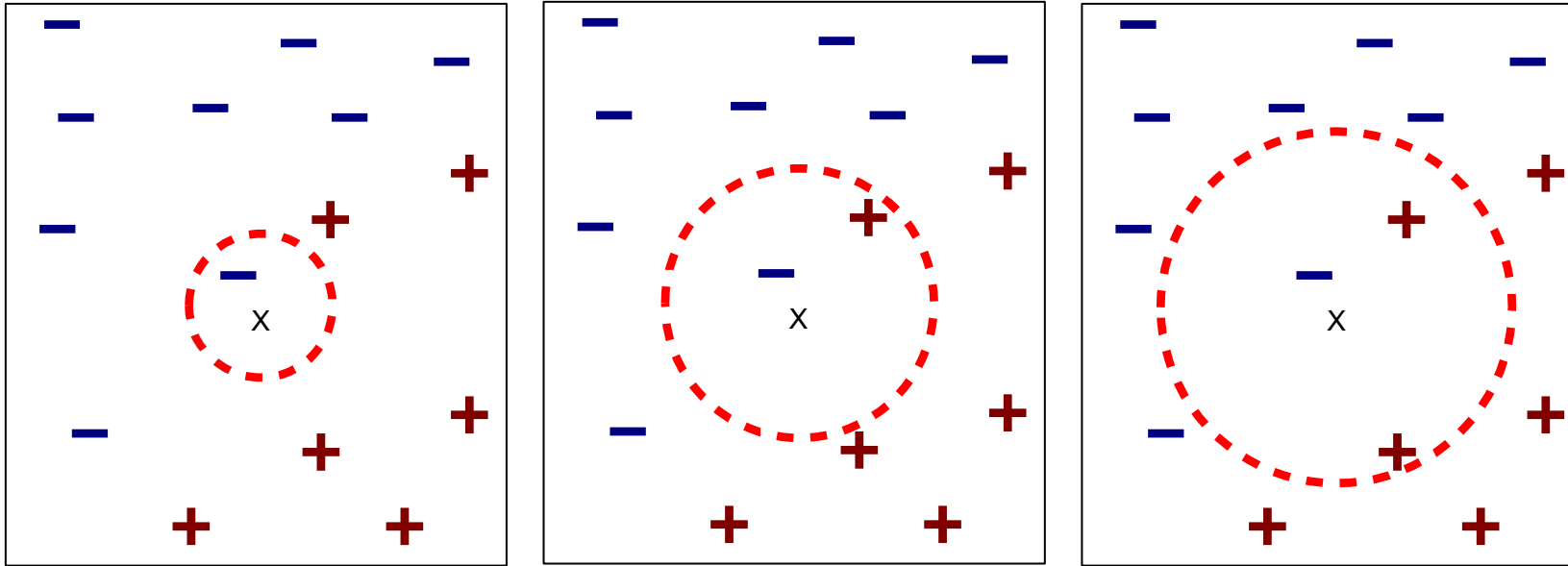


Nearest-Neighbor Classifiers



- | 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:
 - 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

- Compute distance between two points:

- Euclidean distance

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

- Manhattan distance

$$d(p, q) = \sum_i |p_i - q_i|$$

- q norm
distance

$$d(p, q) = \left(\sum_i |p_i - q_i|^q \right)^{1/q}$$

()

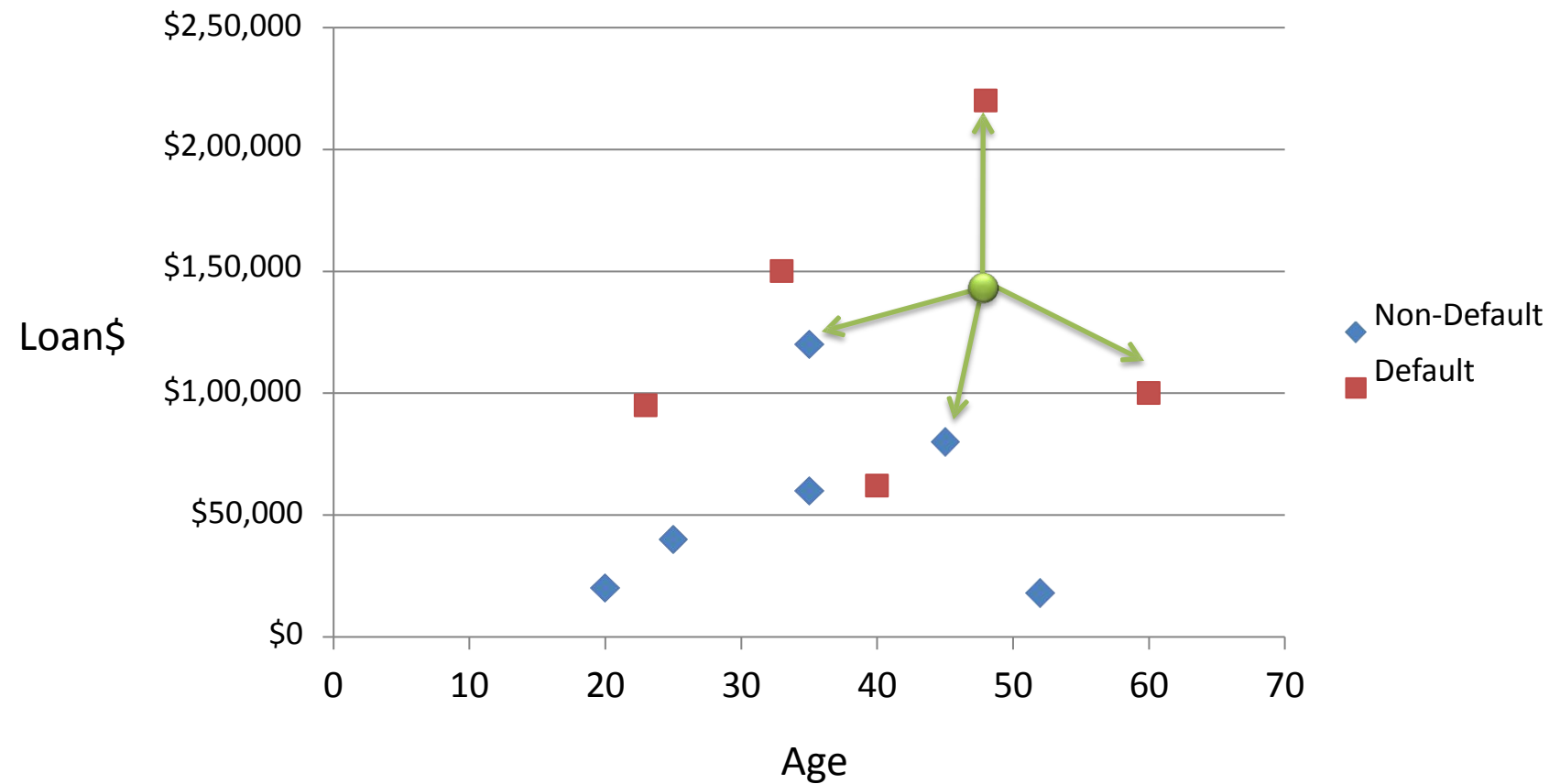
- Determine the class from nearest neighbor list
 - take the majority vote of class labels among the k-nearest neighbors where D_z is the set of k closest training examples to z.
 - Weigh the vote according to distance

The KNN classification algorithm

Let k be the number of nearest neighbors and D be the set of training examples.

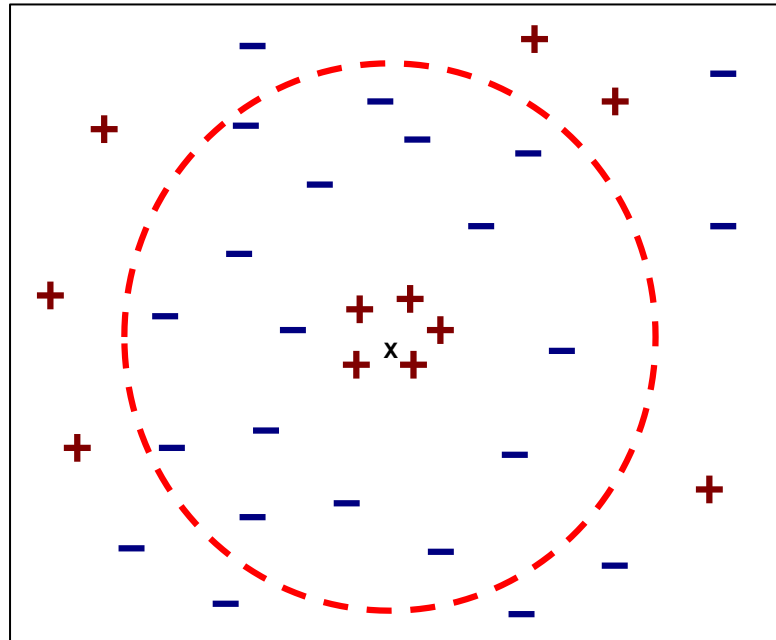
1. **for** each test example $z = (\mathbf{x}', y')$ **do**
2. Compute $d(\mathbf{x}', \mathbf{x})$, the distance between z and every example, $(\mathbf{x}, y) \in D$
3. Select $D_z \subseteq D$, the set of k closest training examples to z .
4. $y' = \operatorname{argmax}_v \sum_{(\mathbf{x}_i, y_i) \in D_z} I(v = y_i)$
5. **end for**

KNN Classification



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...

- 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 60 KG to 100KG
 - income of a person may vary from Rs10K to Rs 2 Lakh

Nearest Neighbor Classification...

- Problem with Euclidean measure:
 - High dimensional data
 - **curse of dimensionality**: all vectors are almost equidistant to the query vector

– Can produce undesirable results

0 1 1 1 1 1 1 1 1 1 1

$d = 1.4142$

vs

1 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 1

$d = 1.4142$

- ◆ 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 tree induction and rule-based systems
 - Classifying unknown records are relatively expensive

Thank
You