Instance Based Classifiers

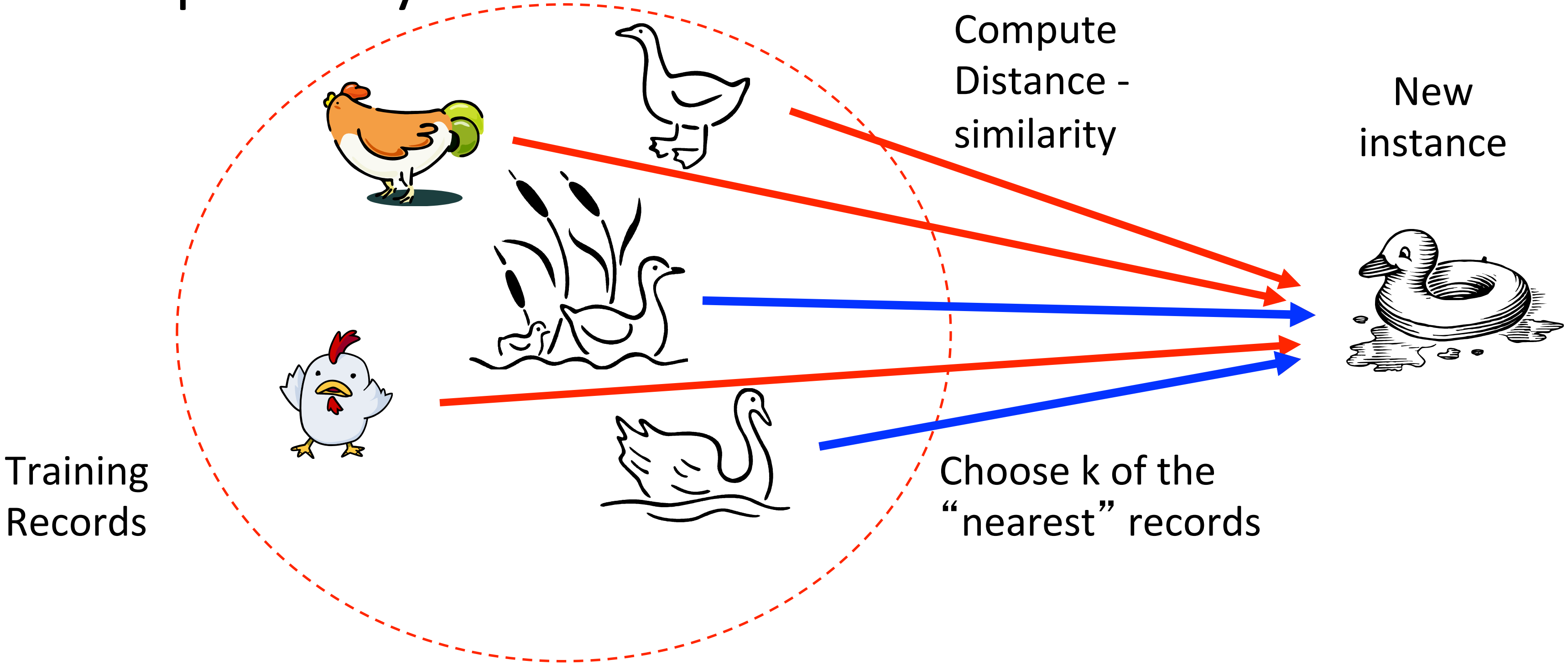
Pasquale Rullo
rullo@mat.unical.it

Instance Based Classifiers

- Instance-based classifiers do not induce a model from training data
- On the contrary, they use a set of pre-classified instances to predict “on the fly” the class label of unseen cases
- K-Nearest Neighbors (KNN)

Nearest Neighbor Classifiers

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



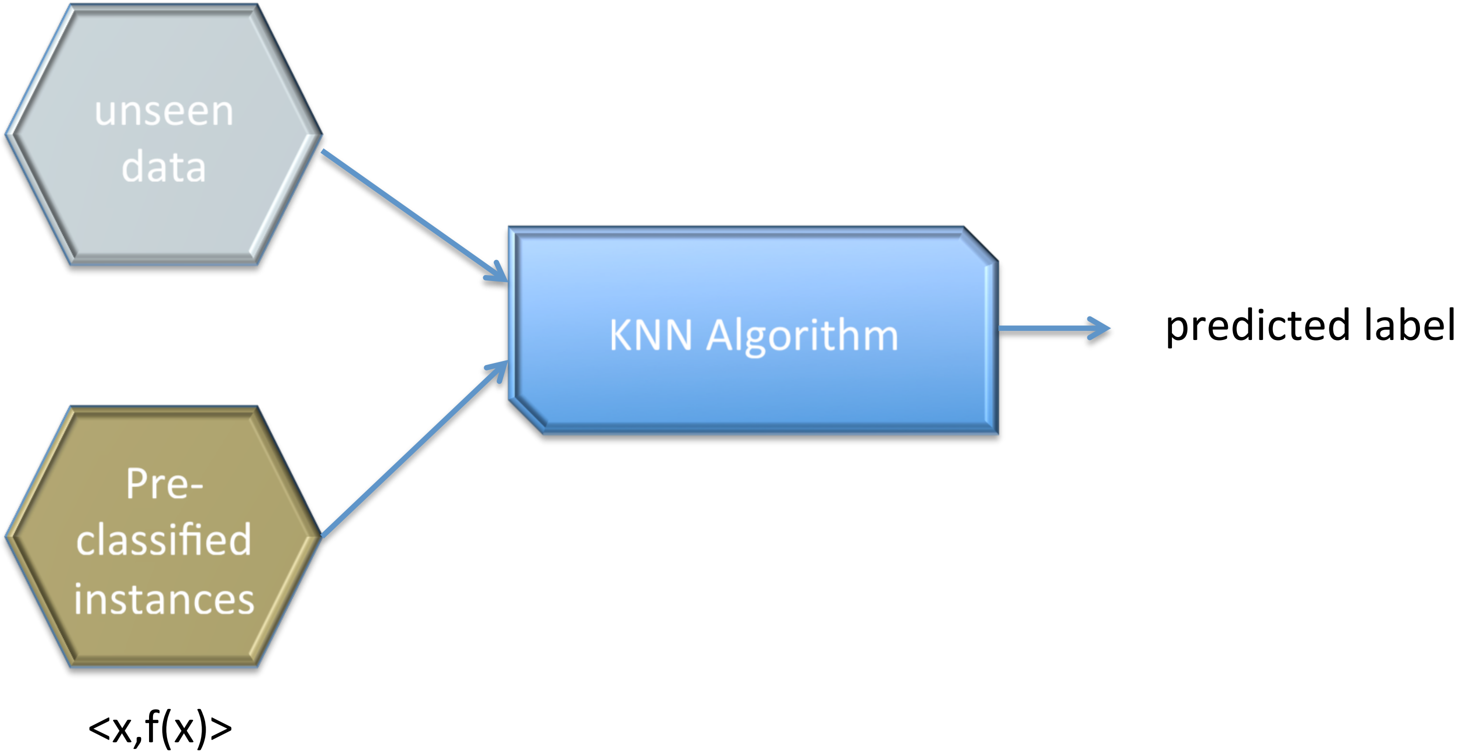
Compute Distance - similarity

New instance

Training Records

Choose k of the “nearest” records

Lazy classifiers



unseen data

Pre-classified instances

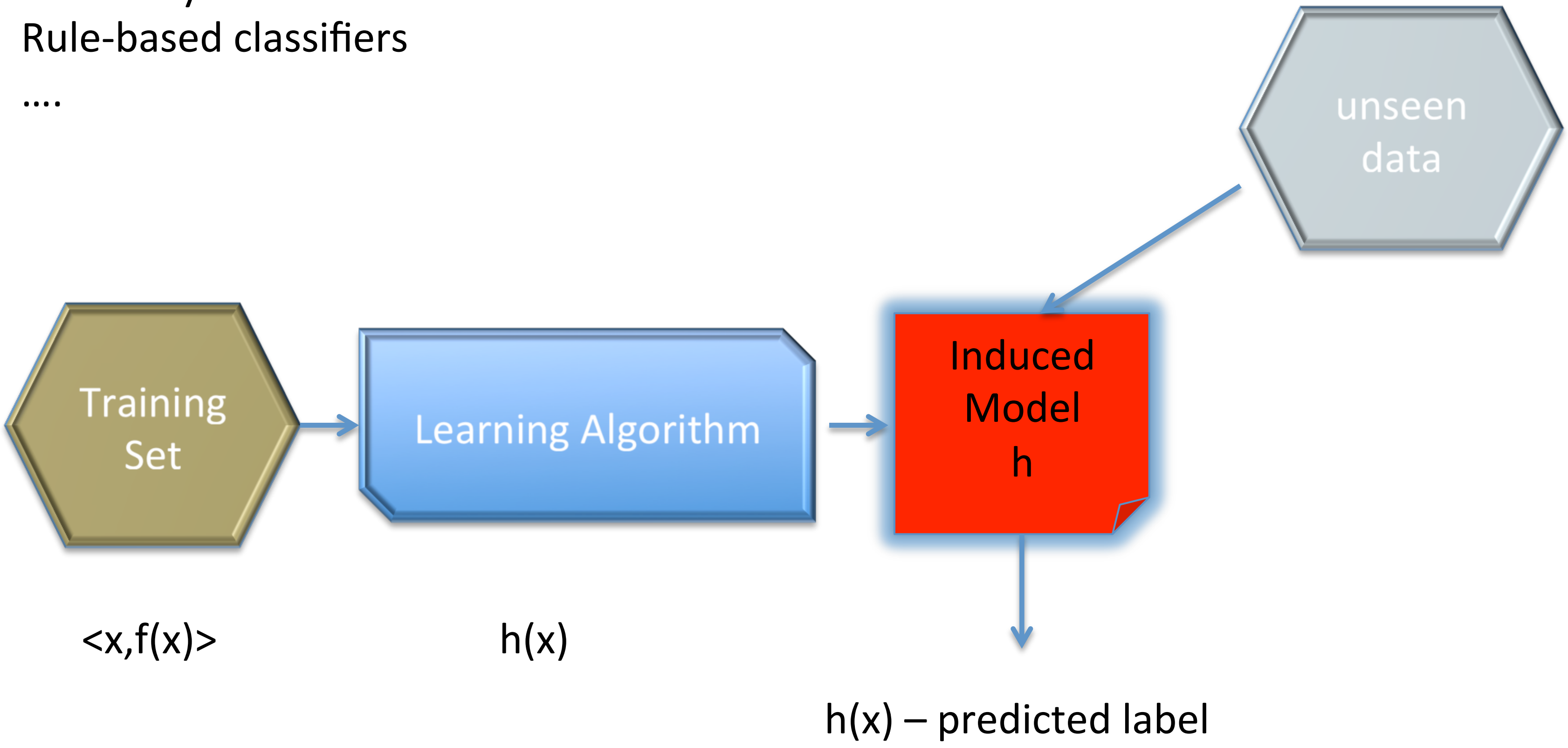
<x,f(x)>

KNN Algorithm

predicted label

Eager classifiers

- Decision trees
- Classification rules
- Naïve Bayes
- Rule-based classifiers
- ….



unseen data

Training Set

Learning Algorithm

Induced Model h

<x,f(x)>

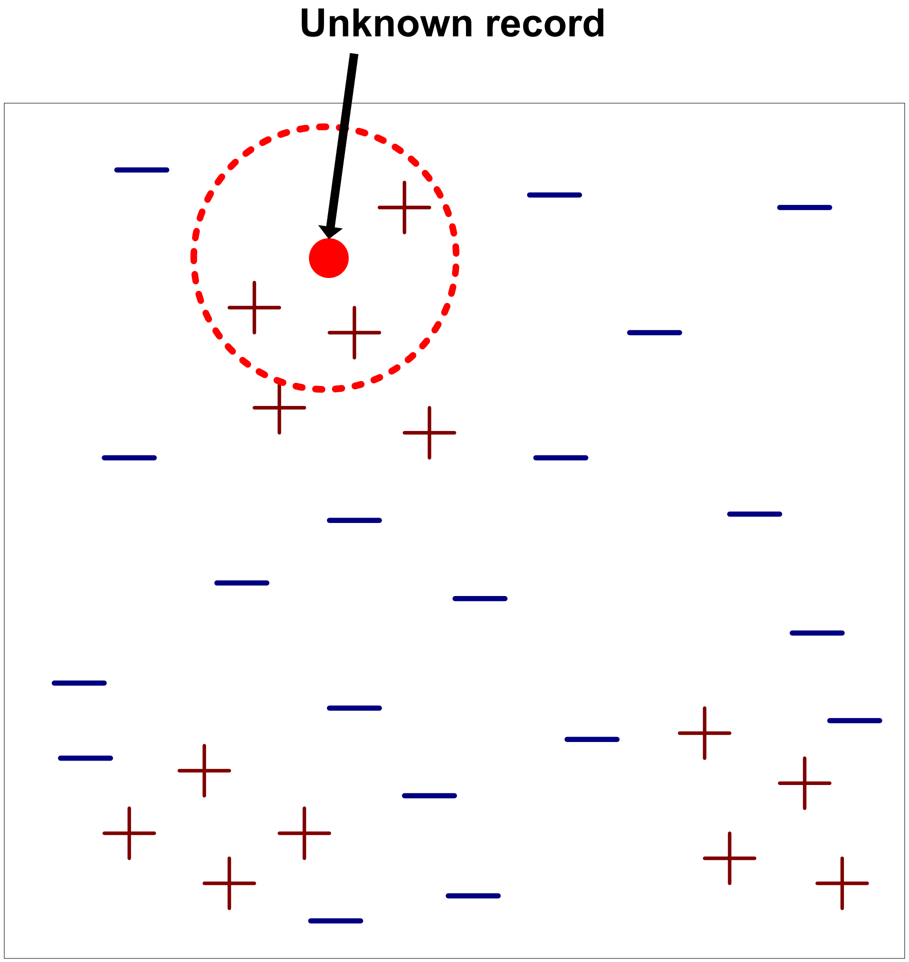
h(x)

h(x) – predicted label

K-Nearest Neighbors (kNN)

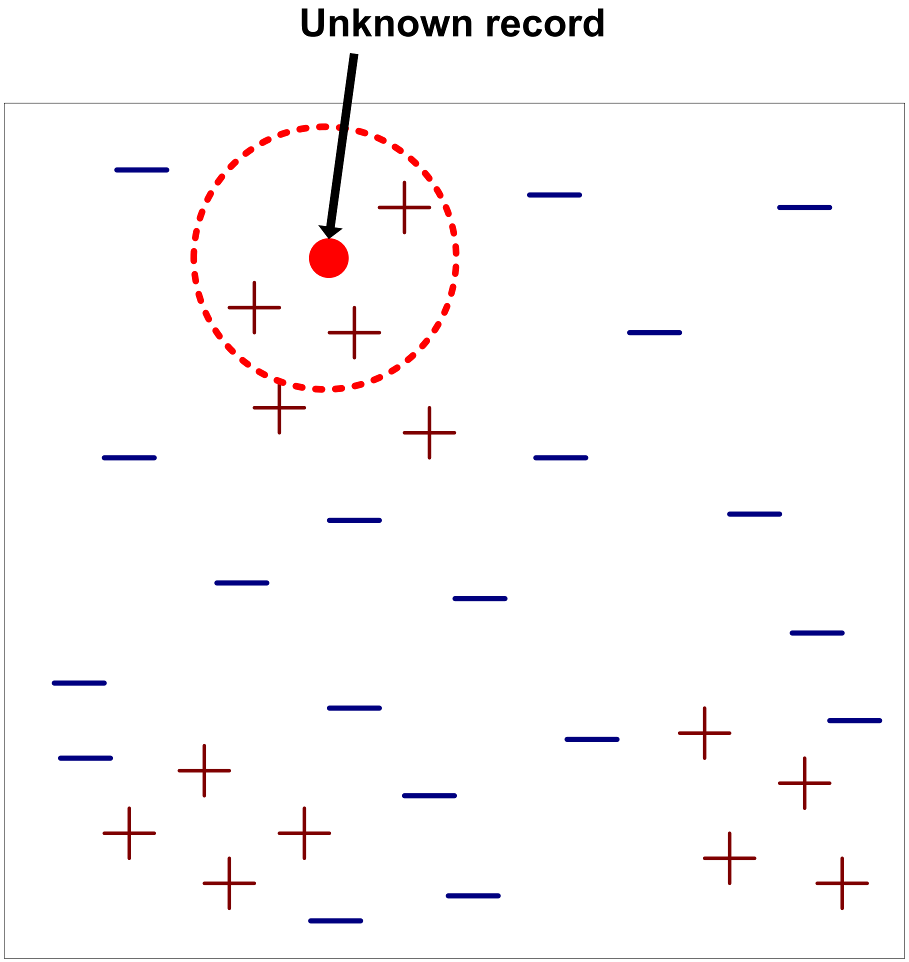
- K-Nearest Neighbors (kNN)
  - Uses k “closest” examples (nearest neighbors) in the training set for performing classification
  - “Closeness” is computed by using a Similarity function (or Distance function)

K-Nearest-Neighbor Classifiers



- Requires three things
  - The set of pre-classified instances
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve
- To classify an unseen instance X:
  - Compute distance of X to other instances
  - Identify *k* nearest neighbors (smallest distance, highest similarity)
  - Use class labels of k nearest neighbors to determine the class label of unseen instance(e.g., by taking majority vote)

K-Nearest-Neighbor Classifiers



- Requires three things
  - The set of pre-classified instances
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve
- To classify an unseen instance X:
  - Compute distance of X to other instances
  - Identify *k* nearest neighbors (smallest distance, highest similarity)
  - Use class labels of k nearest neighbors to determine the class label of unseen instance(e.g., by taking majority vote)

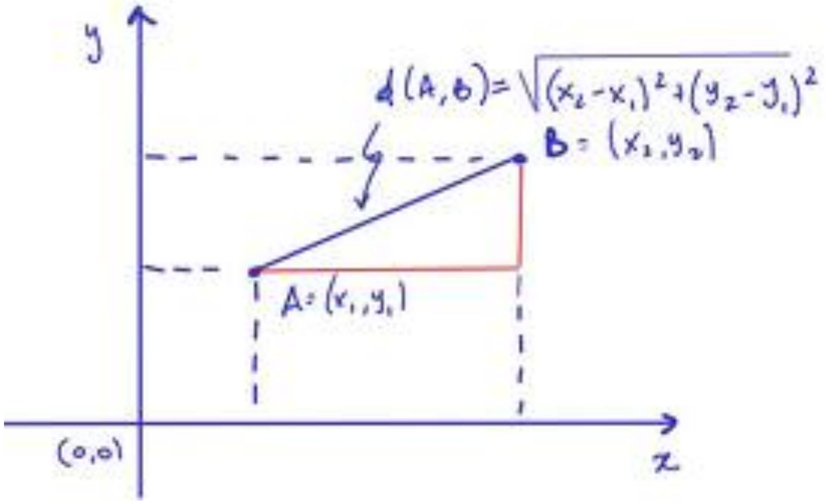
Compute distance of X to other instances - Euclidean distance

- Compute distance between two points:
  - Euclidean distance

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

  - where p and q are two instances, n is the number of their attributes, and p_i and q_i the values of the i-th attributes of p and q. Euclidean distances apply only to numerical attributes

Compute distance of X to other instances - Euclidean distance



$$\text{2-distanza} = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$$

Compute distance of X to other instances - SMC

- Simple Matching Coefficient:
  - $$\text{SMC} = \frac{\text{number of matching attribute values}}{\text{Number of attributes}}$$
- Given

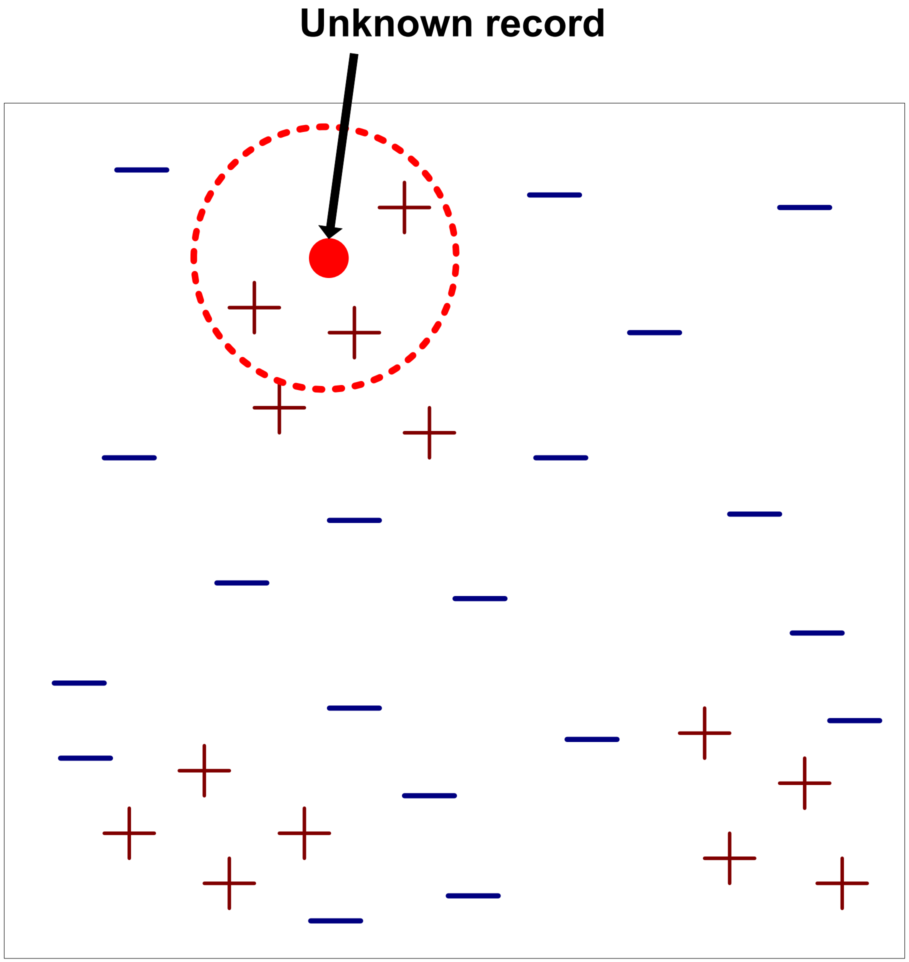
X_1= <15,rome,yellow>

X_2= <20,paris,yellow>

SMC(X_1,X_2) = 1/3= 0.33

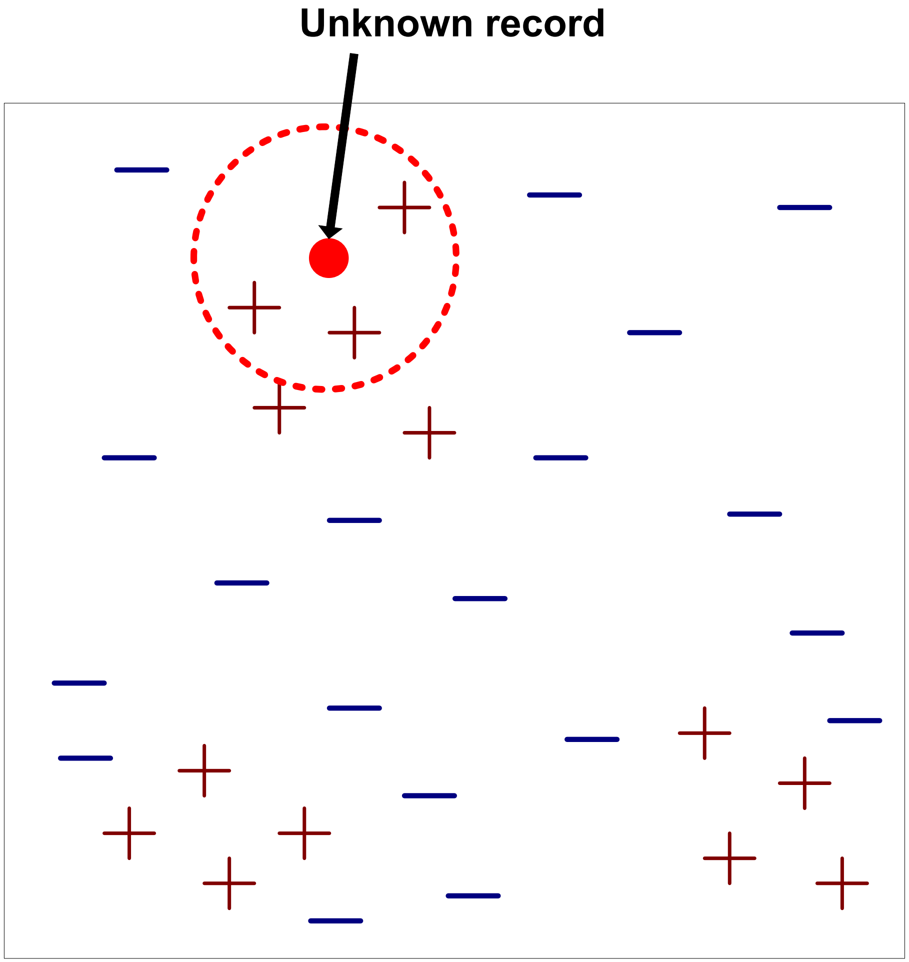
- Other metrics: Jaccard coefficient, Cosine similarity (for documents), etc.

K-Nearest-Neighbor Classifiers



- Requires three things
  - The set of pre-classified instances
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve
- To classify an unseen instance X:
  - Compute distance of X to other instances
  - Identify *k* nearest neighbors (smallest distance, highest similarity)
  - Use class labels of k nearest neighbors to determine the class label of unseen instance(e.g., by taking majority vote)

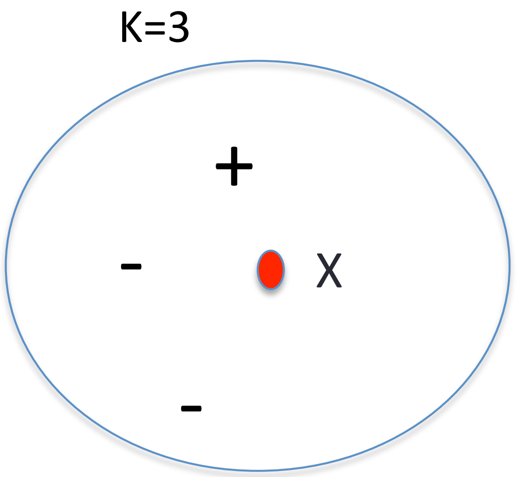
K-Nearest-Neighbor Classifiers



- Requires three things
  - The set of pre-classified instances
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve
- To classify an unseen instance X:
  - Compute distance of X to other instances
  - Identify *k* nearest neighbors (smallest distance, highest similarity)
  - Use class labels of k nearest neighbors to determine the class label of unseen instance(e.g., by taking majority vote)

Determining the class of a new instance X

- K-nearest neighbors of an instance X are data points (instances in the training set) that have the k smallest distances from X (the k most similar instances)
- What if the K-nearest neighbors have different class labels?



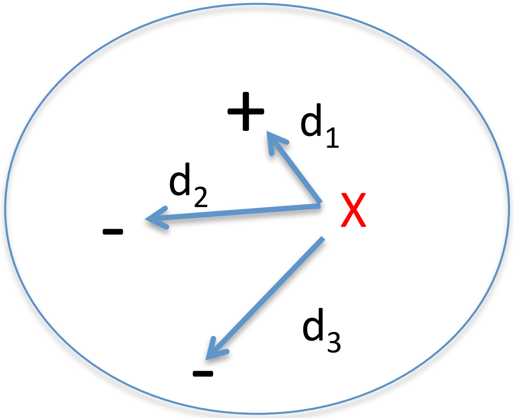
- K=3
- 1 positive and 2 negative examples
- What is the class of X?

Determining the class of a new instance X

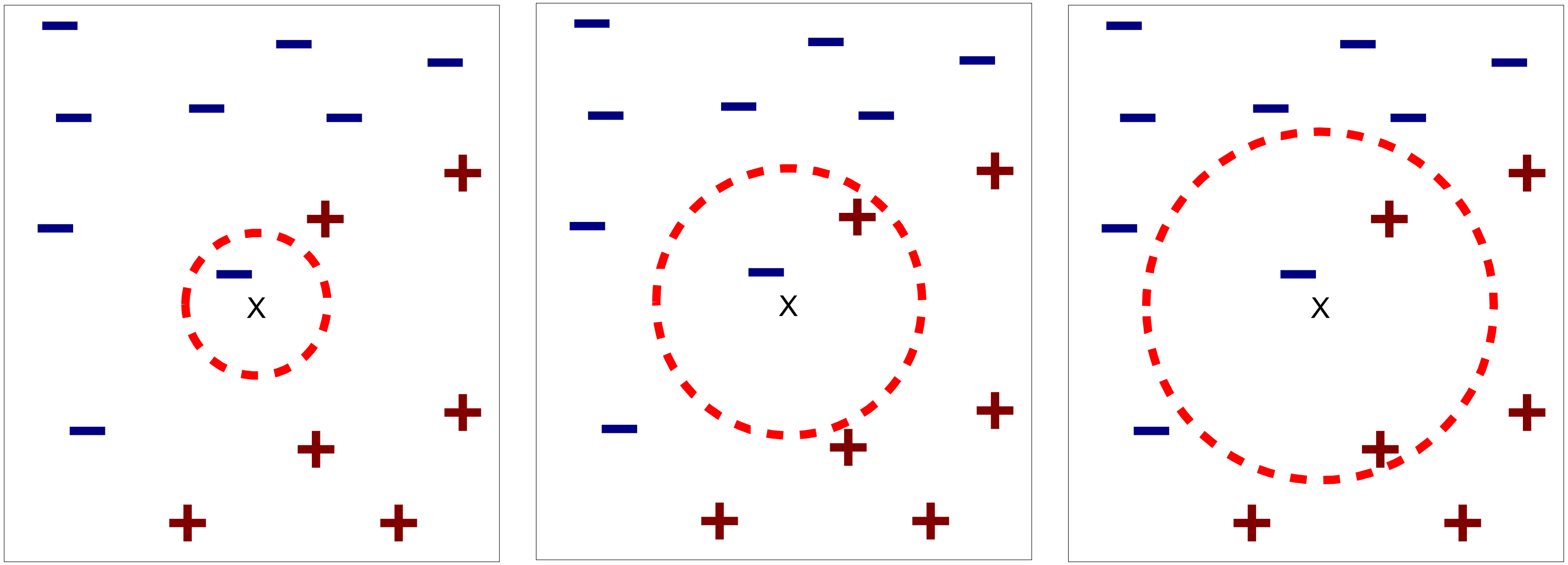
- Determining the class of a new instance X from the k nearest neighbors :
  - Each neighbor Y has associated a weight $w(Y) = 1/d^2$, where d is the distance of Y from X
  - take the majority weighted vote of class labels among the k-nearest neighbors

Nearest-Neighbor Classifiers

- Example: k=3; 1 positive example with distance d_1=2, and 2 negative ones, with distances d_2=3 and d_3=5.
  - w+ = 1/4= 0.25
  - w- = 1/9+1/25= 0.15
  - Vote = 0.25-0.15 >0
- The new instance is classified positive



Nearest-Neighbor Classifiers

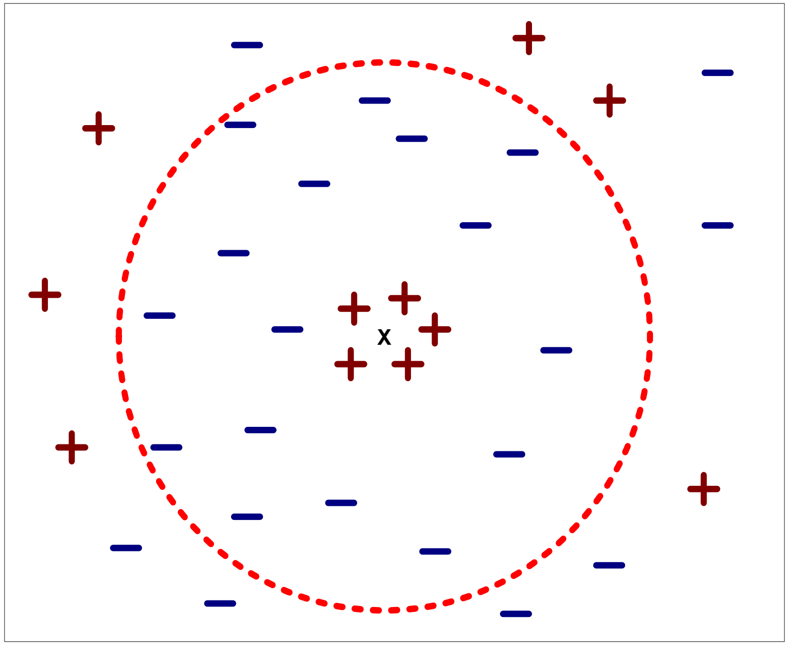


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

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

Nearest-Neighbor Classifiers

- 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



Conclusions

- k-NN classifiers are lazy learners that
  - do not build models explicitly (unlike eager learners such as decision tree induction and rule-based systems)
  - use a set of pre-classified instances along with similarity metrics for classifying unseen data