Instance Based Classifiers

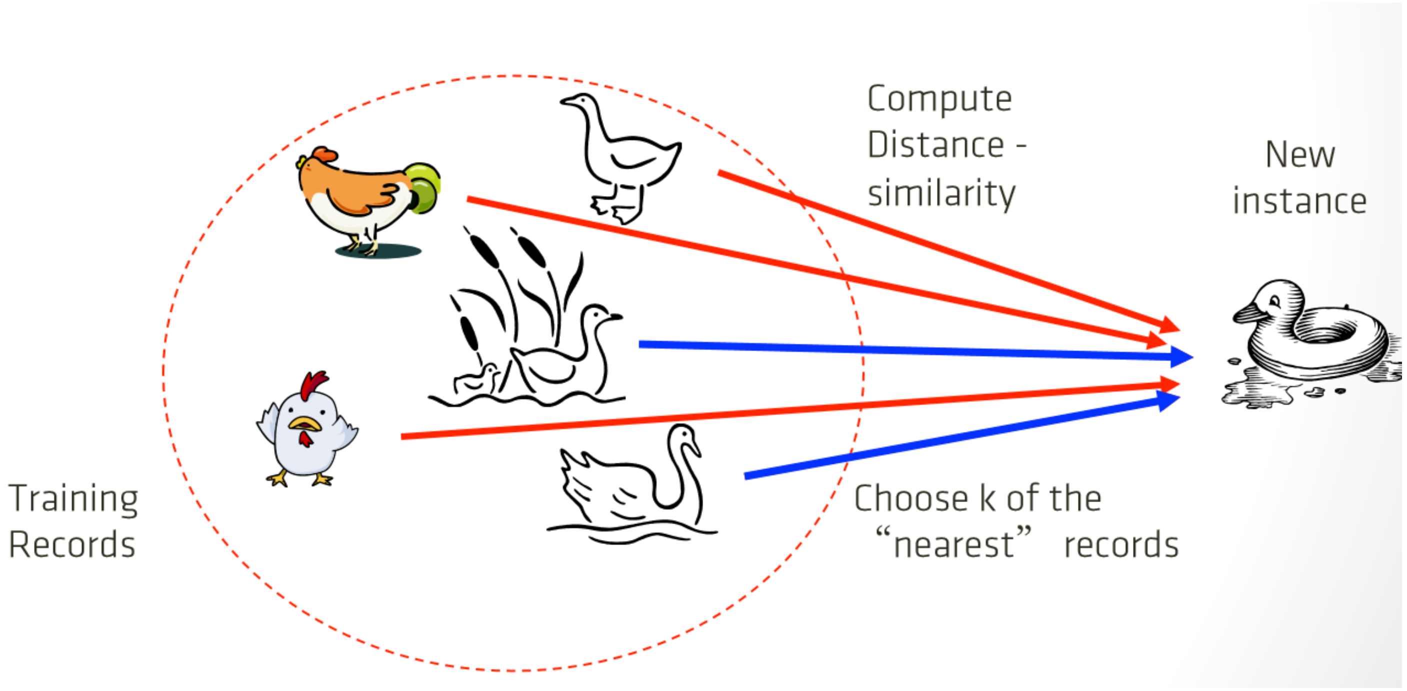
MSC. BUI QUOC KHANH
KHANHBQ@HANU.EDU.VN

Instance Based Classifiers

- Instance-based classifiers
  - do not induce a model from training data
  - use a set of pre-classified instances to predict "on the fly" the class label of unseen cases
  - Called lazy classifiers
- 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

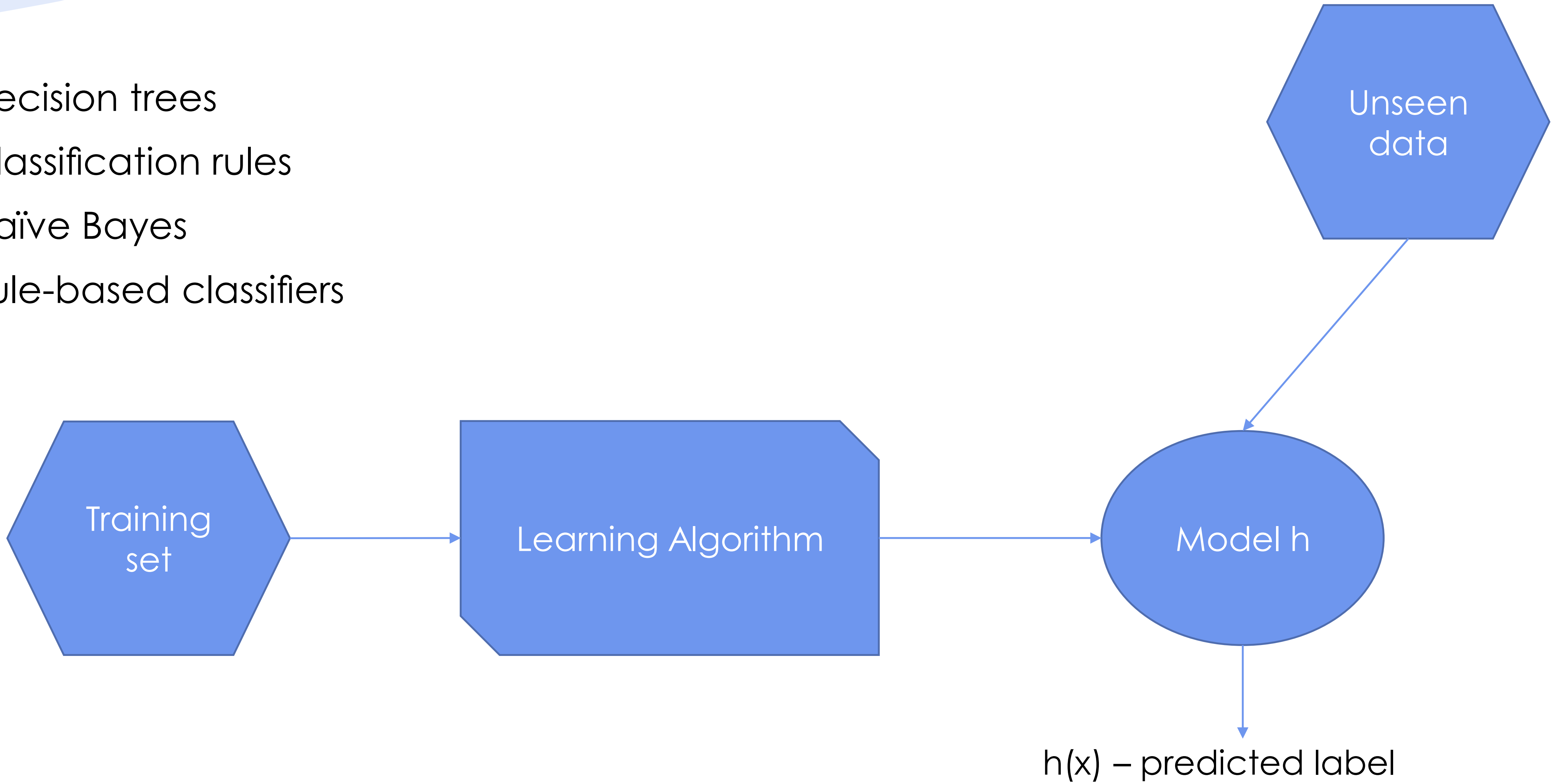


Eager vs Lazy Lernears

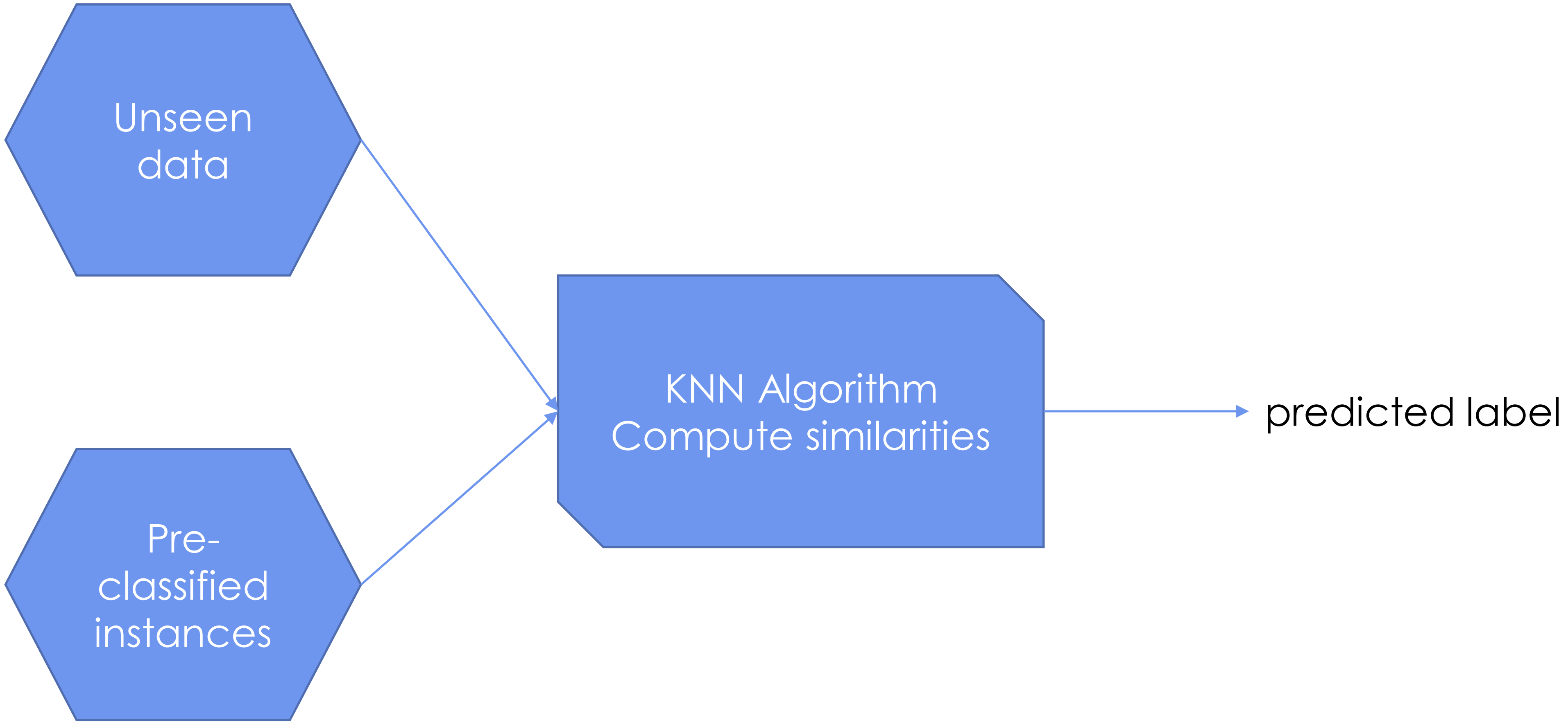
- Eager learner: induce a model fitting the training set – decision trees, rule-based classifiers, Naïve Bayes, etc
- Lazy learners: do not require model induction from data – they need to compute similarity of the unseen instance w.r.t. a set of pre-classified examples

Eager classifiers

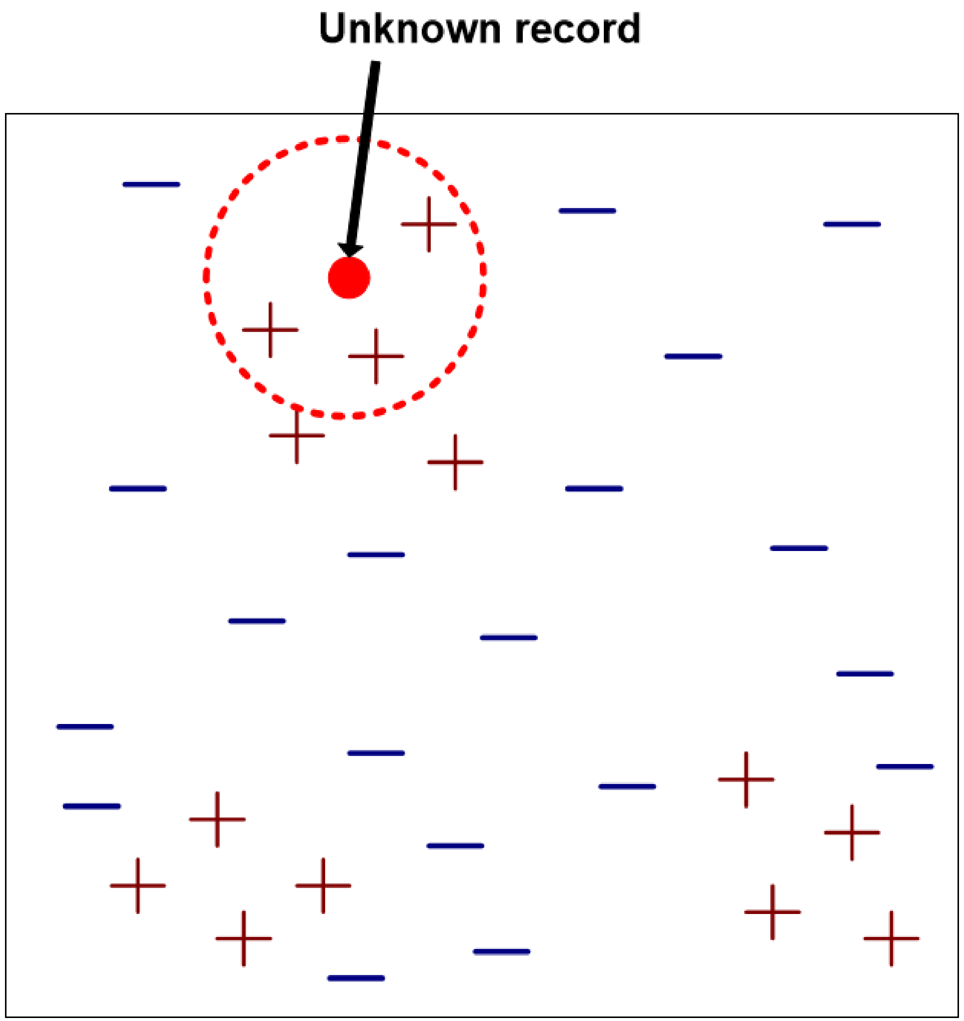
- Decision trees
- Classification rules
- Naïve Bayes
- Rule-based classifiers
- …



Lazy classifiers

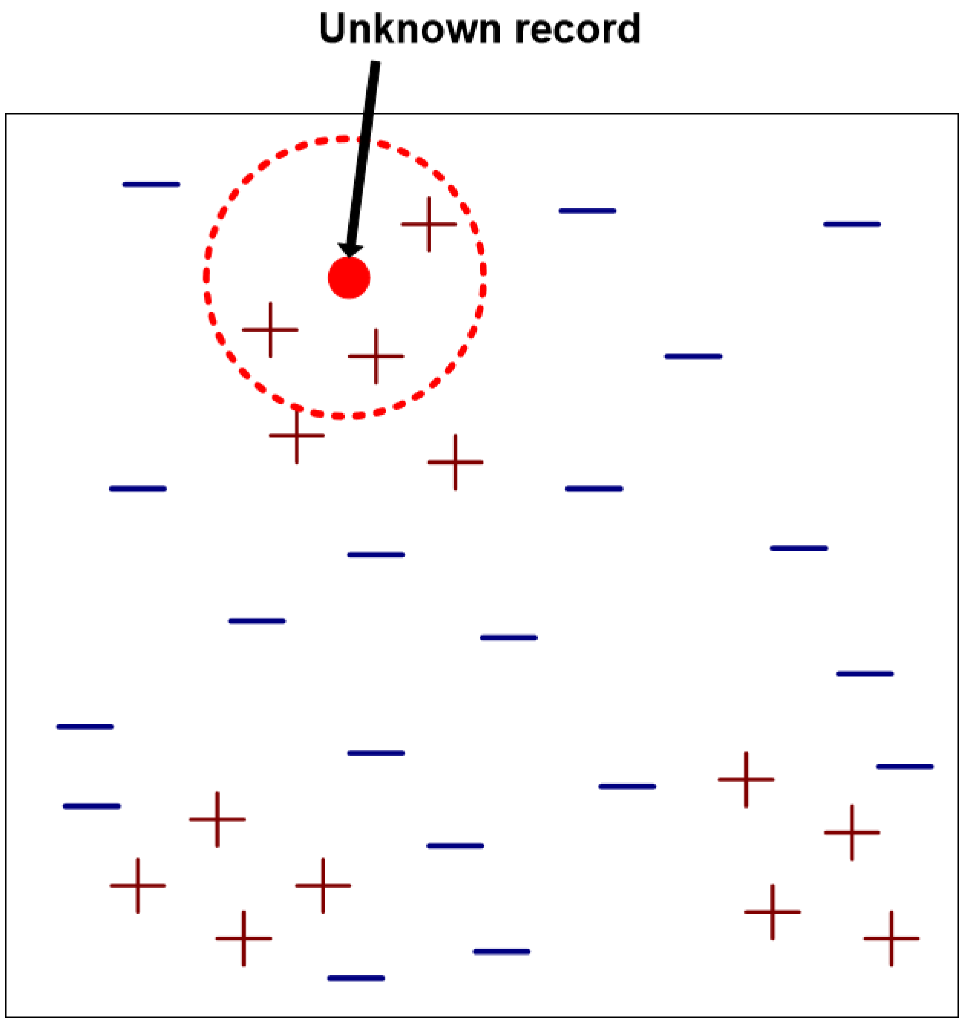


K-Nearest-Neighbor Classifiers



- Requires three things
  - The set of pre-classified instances
  - Distance Metric to compute distance between instances
  - 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 instances
  - 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

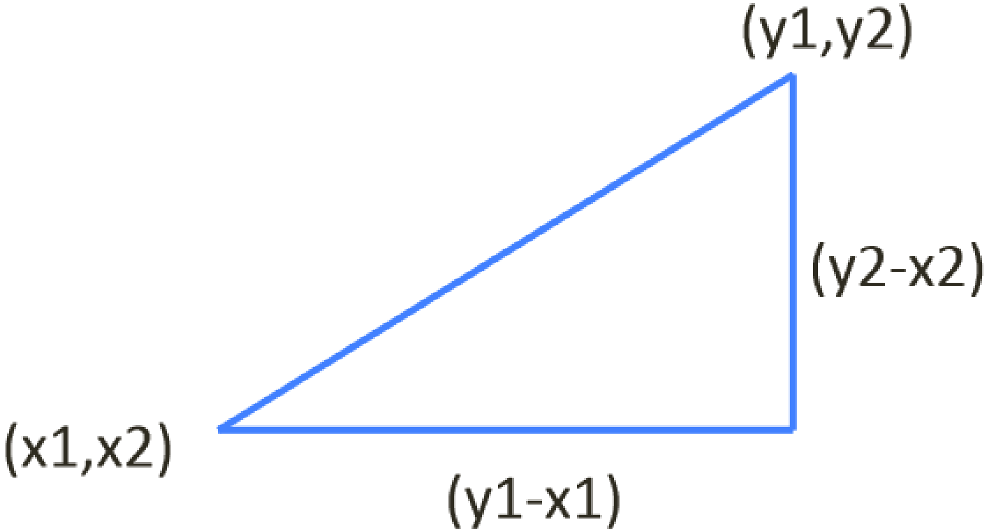
- Compute distance between two points:
  - Euclidean distance

$$\mathrm{dist}(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

  - where x=<x_1,…,x_n> and y=<y_1,…,y_n> are two examples, n is the number of their attributes, and x_i and y_i the values of the i-th attributes of x and y
  - Euclidean distances apply only to numerical attributes

Compute distance of X to other instances - Euclidean distance

$$\mathrm{dist}(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$



Compute distance of X to other instances - SMC

- Simple Matching Coefficient:
  - $SMC = \frac{number\ of\ matching\ attribute\ values}{Number\ of\ attributes}$
- Given
  - X1= <15,rome,yellow>
  - X2= <20,paris,yellow>
- SMC(X1,X2) = 1/3= 0.33

Compute distance of X to other instances - Cosine

- Documents are represented as a document – word matrix
- Given two vectors of attributes (documents), A and B, the cosine similarity cos(θ) is represented using a dot product and magnitude as

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

- Since term frequencies are positive, cos(θ) ranges from 0 to 1, with 1 meaning exactly the same documents

Compute distance of X to other instances - Cosine

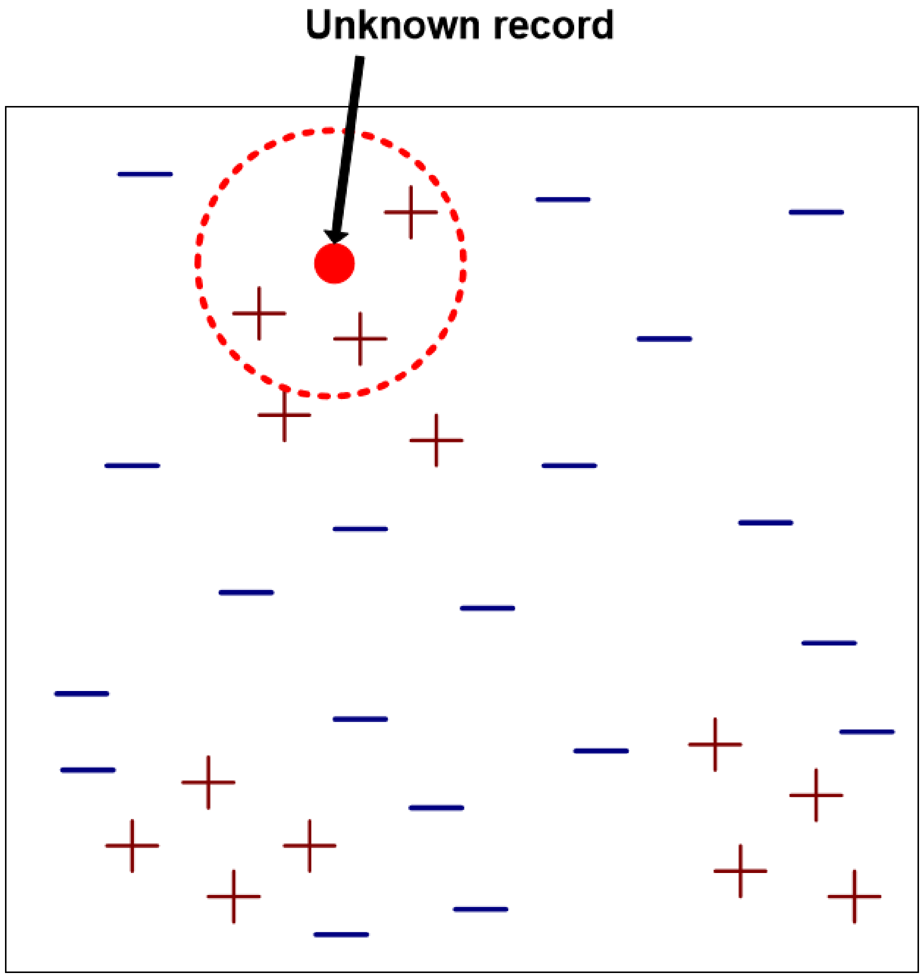
	w_1	w_2	w_3	w_4	w_5	Class
d1	0	1	1	1	0	Sport, politics
d2	0	0	1	1	1	gossip
d3	1	0	0	1	0	Sport, gossip
d4	1	0	0	1	0	politics

- $Cos(d_1, d_2) = \frac{\sum_i d_1(i) \times d_2(i)}{\sqrt{\sum_i d_1(i)^2} \times \sqrt{\sum_i d_2(i)^2}}$

Cos(d1,d2) = $\frac{2}{\sqrt{3} \times \sqrt{3}}$ = 0.66

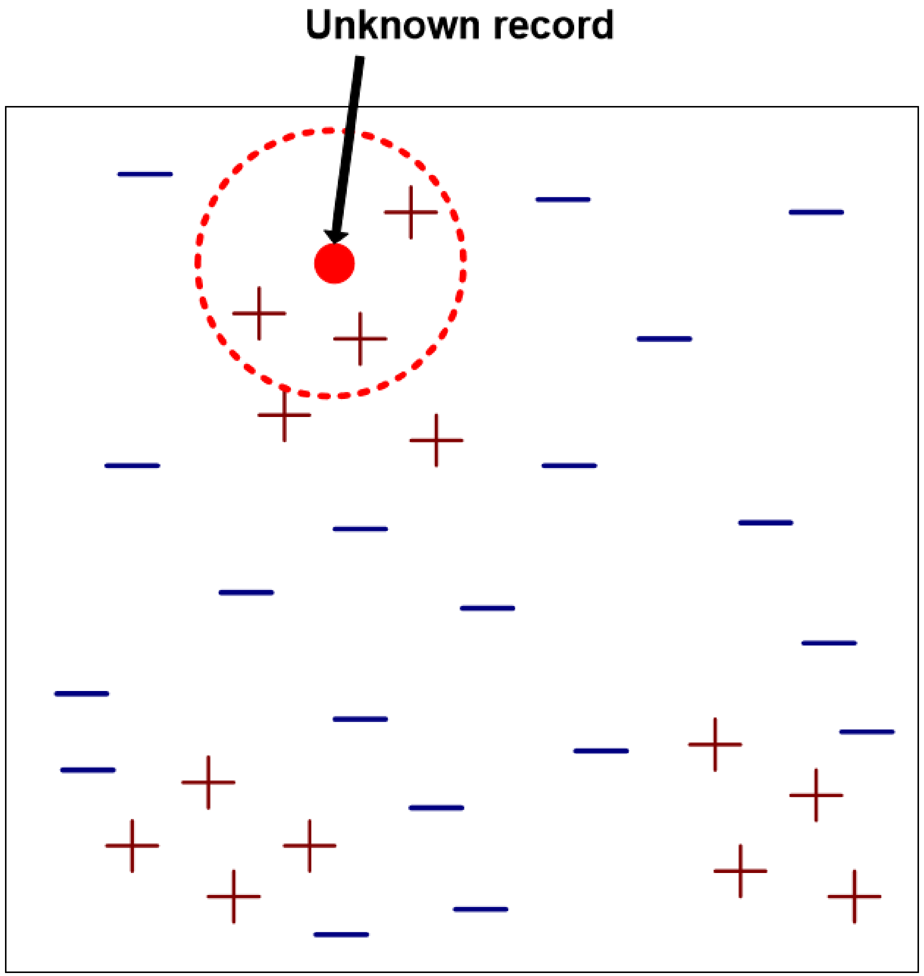
Cos(d3,d4) = $\frac{2}{\sqrt{2} \times \sqrt{2}}$ = 1 (d3 and d4 are identical)

K-Nearest Neighbor Classifiers



- Requires three things
  - The set of pre-classified instances
  - Distance Metric to compute distance between instances
  - 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)

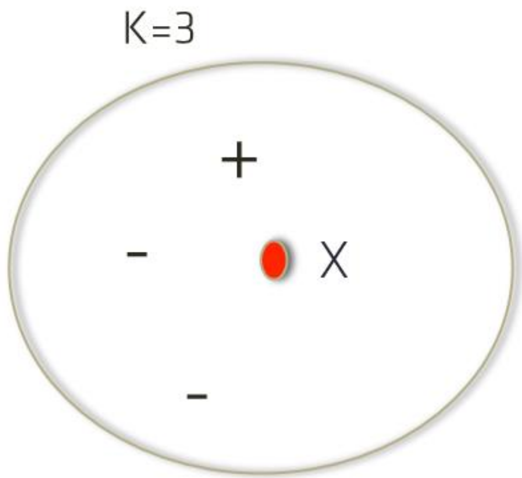
K-Nearest Neighbor Classifiers



- Requires three things
  - The set of pre-classified instances
  - Distance Metric to compute distance between instances
  - 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

- 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?



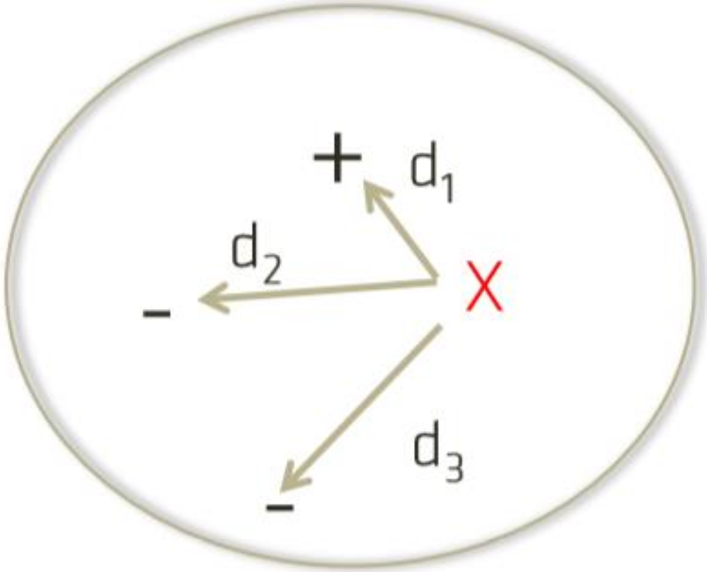
- K=3
- 1 positive and 2 negative examples
- What is the class of X?

Determining the class of a new instance

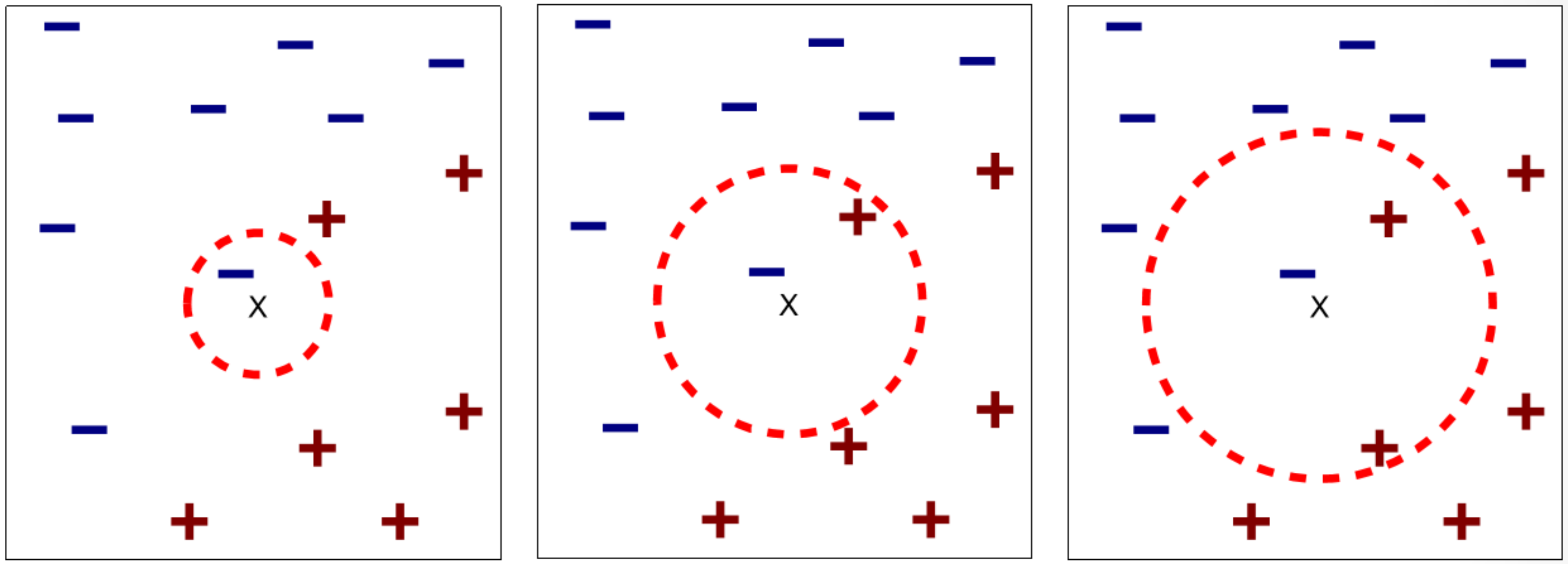
- Determining the class of a new instance X from the k nearest neighbors :
  - Each neighbor Y has associated a weight $w(Y) = 1/d^2$, where d is the distance of Y from X
  - Distant examples will have little effect on the class of X
  - Take the majority weighted vote of class labels among the k-nearest neighbors
  - NOTE: if the distance of X from Y is 0 (the two instances coincide), then class(X)=class(Y)

Determining the class of a new instance

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



K-NN The choice of K

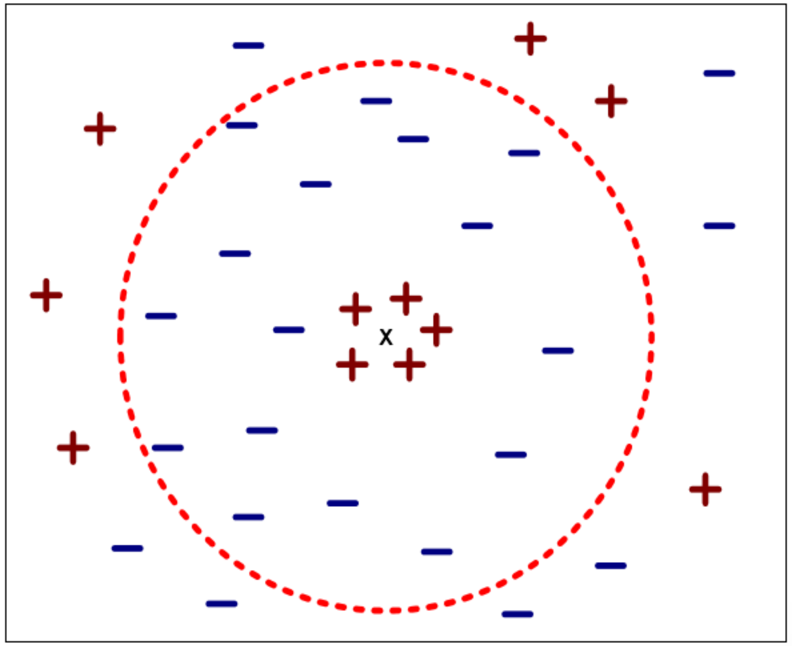


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

- K-nearest neighbors of an instance X are data points (instances) that have the k smallest distances to x

K-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



Issues with K-NN Classifiers

- The quality of classification strongly depends on the proximity metrics
- Suppose we want to classify persons based on their height and weight
  - Height has a low variability – from 1.5 to 1.9 meters
  - Weight has a higher variability – from 50 to 150 kg
  - The proximity measure is dominated by the height, unless the scale of the attributes is not taken into consideration
- Suppose each example is described in terms of 50 attributes, but only 2 are relevant to classification; examples having identical values for the 2 attributes may nevertheless be distant –proximity is dominated by not relevant attributes

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
  - Classifying a test instance X may be expensive as the similarity of X to all training examples is to be computed