Machine Learning

Lecture 11: k-Nearest Neighbors

COURSE CODE: CSE451

2021

Course Teacher

Dr. Mrinal Kanti Baowaly

Associate Professor

Department of Computer Science and Engineering, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Bangladesh.

Email: baowaly@gmail.com

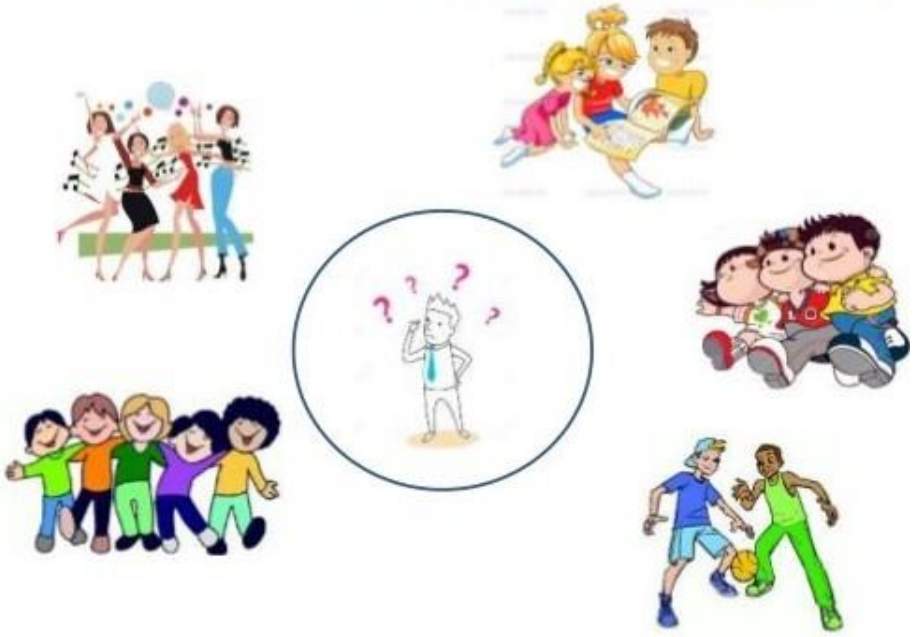
Instance Based Learning

- No model is learned
- It does not learn from the training set immediately instead it stores the dataset and uses them at the time of prediction(hence also called lazy learning).
- It classifies/predicts the test data based on its similarity to the stored training data.
- Example: KNN algorithm

What is k-Nearest Neighbors (kNN) learning?

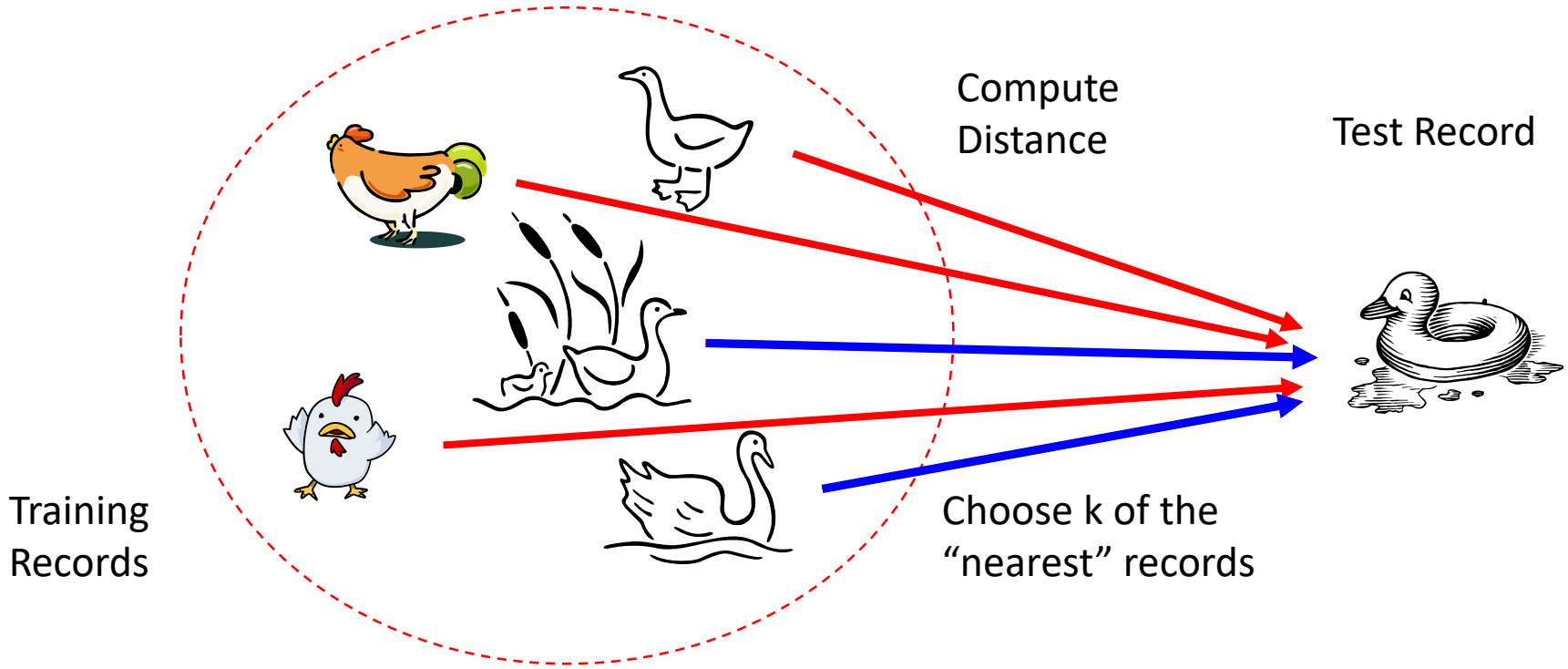
- A type of instance-based learning in which an unknown object is classified with the most common class among its k-closet objects

Basic Idea:
Analogy for kNN

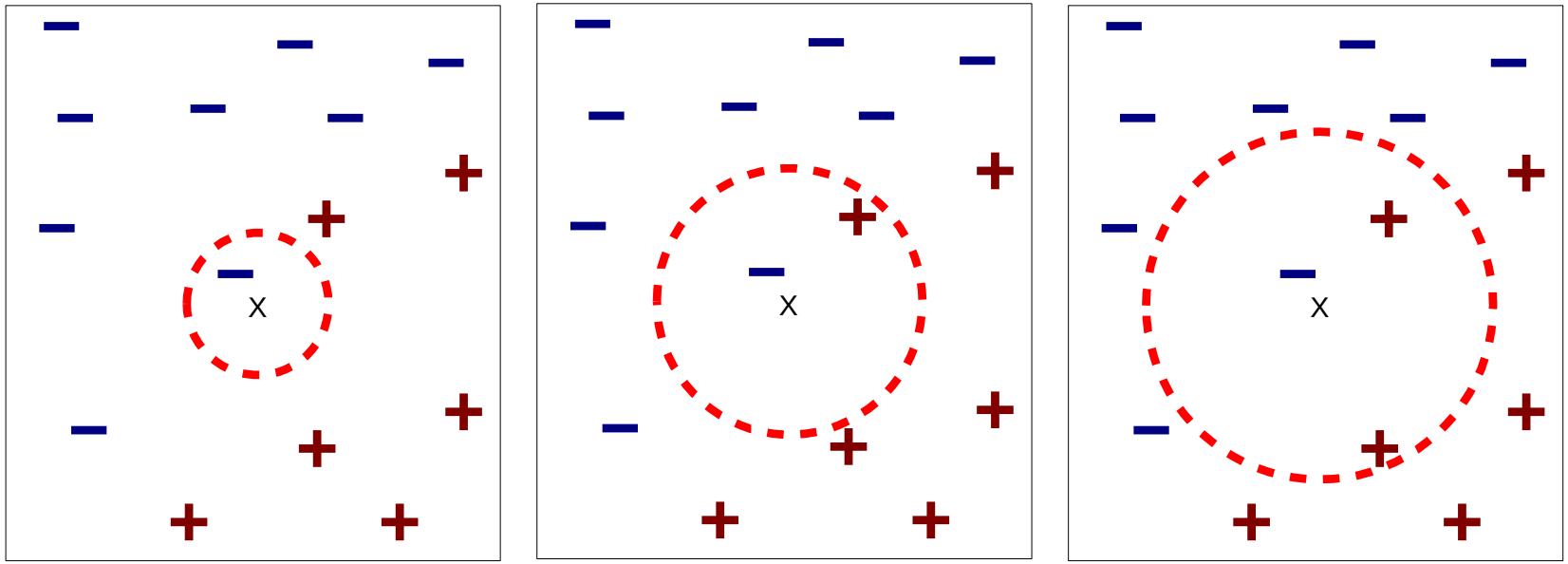


Another Analogy for kNN

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



What are k-Nearest Neighbors?



(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

kNN algorithm

- To classify an unknown record
  - Compute distance to all other training records
  - Identify k-nearest records (neighbors)
  - Find the most common class of the nearest k-neighbors and assign the class for the unknown record

Distance measures

- Euclidean distance: It is useful in low dimensions, it doesn’t work well in high dimensions and for categorical variables.
- Hamming distance: Calculate the distance between binary vectors.
- Manhattan distance: Calculate the distance between real vectors using the sum of their absolute difference. Also called City Block Distance.
- Minkowski distance: Generalization of Euclidean and Manhattan distance.

Detail: Distance Metrics in Machine Learning

How to choose the value of k?

- Choice of k is very critical – A small value of k means that noise will have a higher influence on the result. A large value make it computationally expensive and may defeat the basic philosophy behind kNN (that objects that are near might have similar classes).
- A simple approach to select $k = \sqrt{n}$, where n is the number of samples in the training data.
- Sometimes it is best to run through each possible value for k (e.g., start with k=1 and then increase it) and then decide the value of k that outputs the best performance with respect to training and test data
- Choose an odd number for the binary classification

How to decide the class label?

- Take the majority vote of class labels from the k-Nearest Neighbors
- Weigh the vote according to distance weight factor, $w = 1/d^2$

Example of kNN

- Suppose you have height, weight and T-shirt size of some customers
- You need to predict the T-shirt size of a new customer named ‘Monica’ who has height 161cm and weight 61kg.

Detail: ListenData

Characteristics of kNN

- Non-parametric (i.e. it does not make any assumption on underlying data)
- Lazy learner/instance-based
  (Find what is Eager Vs. Lazy Learners? Source: Datacamp)
- Very simple and easy to implement
- Minimal training but expensive testing
- Choosing the value of k is crucial
- Variables should be normalized/standardized else higher range variables can bias it (source: ListenData)
- Susceptible of high number of independent variables

Some Learning Materials

Datacamp: KNN Classification using Scikit-learn

Javatpoint: K-Nearest Neighbor(KNN) Algorithm for Machine Learning

ListenData: K NEAREST NEIGHBOR : STEP BY STEP TUTORIAL

AnalyticsVidhya: Introduction to k-Nearest Neighbors