Introduction to Data Analytics

ITE 5201

Lecture9-K Nearest Neighbor Classification

Instructor: Parisa Pouladzadeh
Email: parisa.pouladzadeh@humber.ca

www.udemy. /course/python-for-data-science-and-machine-learning.com

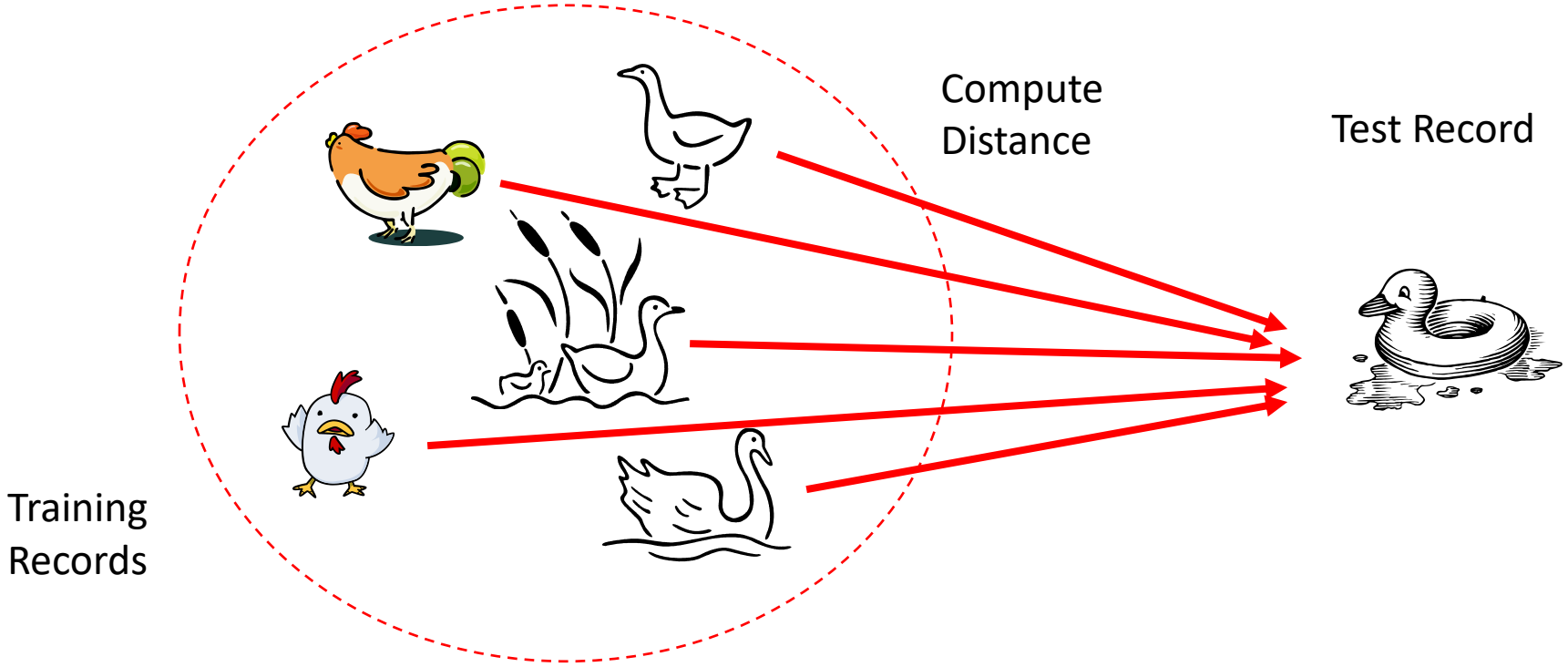
Nearest Neighbor Classifiers

- KNN algorithm is one of the simplest classification algorithm
- non-parametric
  - it does not make any assumptions on the underlying data distribution
- lazy learning algorithm.
  - there is *no explicit training phase* or it is very minimal.
  - also means that the training phase is pretty fast .
  - Lack of generalization means that KNN keeps all the training data.
- Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point.

Nearest Neighbor Classifiers

Basic idea:

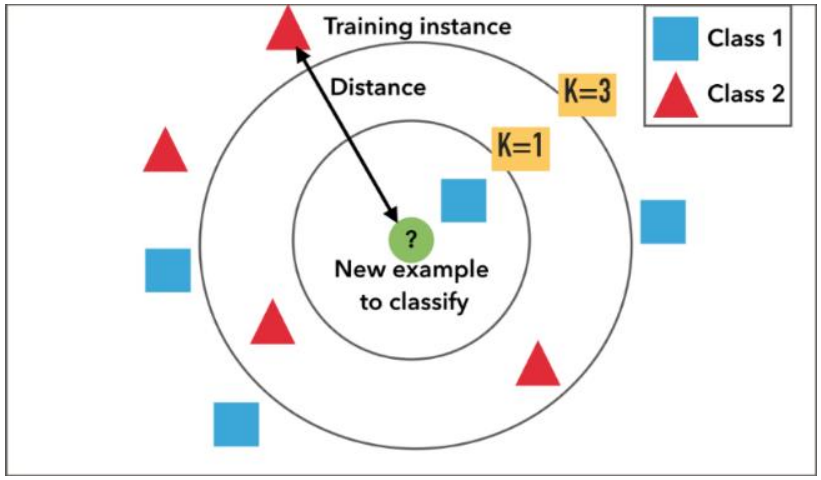
- If it walks like a duck, quacks like a duck, then it's probably a duck



Nearest Neighbor Classifiers

KNN Algorithm is based on feature similarity

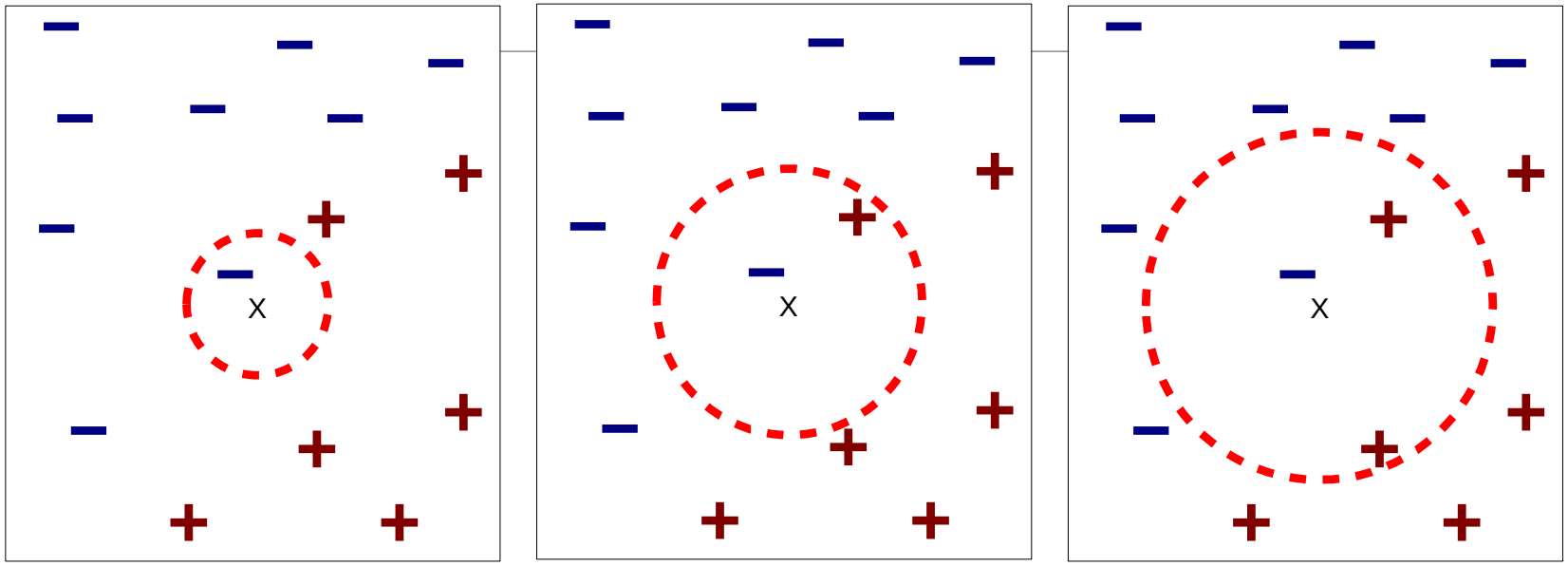
How closely out-of-sample features resemble our training set determines how we classify a given data point



Basic Idea

- *k*-NN classification rule is to assign to a test sample the majority category label of its *k* nearest training samples
- In practice, *k* is usually chosen to be odd, so as to avoid ties
- The *k* = 1 rule is generally called the nearest-neighbor classification rule

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

Classification steps

- Training phase: a model is constructed from the training instances.
  - classification algorithm finds relationships between predictors and targets
  - relationships are summarised in a model
- Testing phase: test the model on a test sample whose class labels are known but not used for training the model
- Usage phase: use the model for classification on new data whose class labels are unknown

K-Nearest Neighbor

Features

- All instances correspond to points in an n-dimensional Euclidean space
- Classification is delayed till a new instance arrives
- Classification done by comparing feature vectors of the different points
- Target function may be discrete or real-valued

K-Nearest Neighbor

- An arbitrary instance is represented by

  $(a_1(x), a_2(x), a_3(x),.., a_n(x))$
  - $a_i(x)$ denotes features
- Euclidean distance between two instances

  $$d(x_i, x_j)=\text{sqrt (sum for r=1 to n } (a_r(x_i) - a_r(x_j))^2)$$
- Continuous valued target function
  - mean value of the k nearest training examples

Euclidean Distance

- K-nearest neighbours uses the local neighborhood to obtain a prediction
- The K memorized examples more similar to the one that is being classified are retrieved
- A distance function is needed to compare the examples similarity
- This means that if we change the distance function, we change how examples are classified

Formula

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

p, q = two points in Euclidean n-space

q_i, p_i = Euclidean vectors, starting from the origin of the space (initial point)

n = n-space

Normalization

If the ranges of the features differ, feaures with bigger values will dominate decision

In general feature values are normalized prior to distance calculation

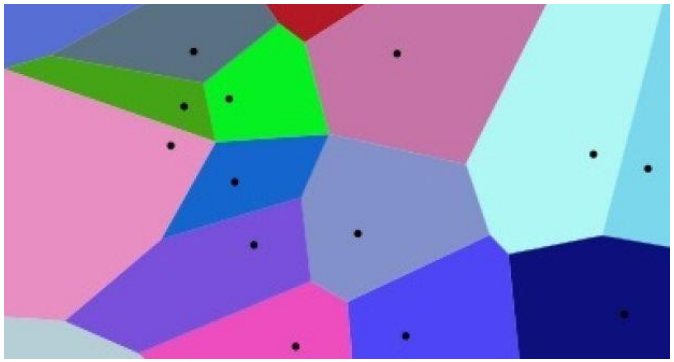
$$Xs = \frac{X - mean}{s.d.}$$

$$Xs = \frac{X - mean}{max - min}$$

$$Xs = \frac{X - min}{\max - min}$$

Voronoi diagram

- We frequently need to find the nearest hospital, surgery or supermarket.
- A map divided into cells, each cell covering the region closest to a particular centre, can assist us in our quest.



Voronoi diagram

Another practical problem is to choose a location for a new service, such as a school, which is as far as possible from existing schools while still serving the maximum number of families.

A Voronoi diagram can be used to find the largest empty circle amid a collection of points, giving the ideal location for the new school. Of course, numerous parameters other than distance must be considered, but access time is often the critical factor.

Numerical Example

Steps:

1. Determine parameter K = number of nearest neighbors

2. Calculate the distance between the query-instance and all the training samples

3. Sort the distance and determine nearest neighbors based on the K-th minimum distance

4. Gather the category of the nearest neighbors

5. Use simple majority of the category of nearest neighbors as the prediction value of the query instance

Example

We have data from the questionnaires survey (to ask people opinion) and objective testing with two attributes (acid durability and strength) to classify whether a special paper tissue is good or not. Here is four training samples

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Y = Classification
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

Now the factory produces a new paper tissue that pass laboratory test with X1 = 3 and X2 = 7. Without another expensive survey, can we guess what the classification of this new tissue is?

1. Determine parameter K = number of nearest neighbors

Suppose use K = 3

2. Calculate the distance between the query-instance and all the training samples

Coordinate of query instance is (3, 7), instead of calculating the distance we compute square distance which is faster to calculate (without square root)

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	**Square Distance to query instance (3, 7)**
7	7	$(7-3)^2+(7-7)^2=16$
7	4	$(7-3)^2+(4-7)^2=25$
3	4	$(3-3)^2+(4-7)^2=9$
1	4	$(1-3)^2+(4-7)^2=13$

3. *Sort the distance and determine nearest neighbors based on the K-th minimum distance*

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Square Distance to query instance (3, 7)	**Rank minimum distance**	**Is it included in 3-Nearest neighbors?**
7	7	$(7-3)^2+(7-7)^2=16$	**3**	**Yes**
7	4	$(7-3)^2+(4-7)^2=25$	**4**	**No**
3	4	$(3-3)^2+(4-7)^2=9$	**1**	**Yes**
1	4	$(1-3)^2+(4-7)^2=13$	**2**	**Yes**

4. Gather the category Y *of the nearest neighbors.* Notice in the second row last column that the category of nearest neighbor (Y) is not included because the rank of this data is more than 3 (=K).

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Square Distance to query instance (3, 7)	Rank minimum distance	Is it included in 3-Nearest neighbors?	**Y = Category of nearest Neighbor**
7	7	$(7-3)^2+(7-7)^2=16$	3	Yes	**Bad**
7	4	$(7-3)^2+(4-7)^2=25$	4	No	-
3	4	$(3-3)^2+(4-7)^2=9$	1	Yes	**Good**
1	4	$(1-3)^2+(4-7)^2=13$	2	Yes	**Good**

5. *Use simple majority of the category of nearest neighbors as the prediction value of the query instance*

We have 2 good and 1 bad, since 2>1 then we conclude that a new paper tissue that pass laboratory test with X1 = 3 and X2 = 7 is included in **Good** category.