K - Nearest Neighbor (KNN)

Dr. M. Sridevi

Definition

- K-Nearest Neighbor is considered a lazy learning algorithm

that classifies data sets based on their similarity with neighbors.

- “K” stands for number of data set items that are considered for the classification.

Mathematically

- For the given attributes A={X1, X2….. XD} Where D is the dimension of the data, we need to predict the corresponding classification group G={Y1,Y2…Yn} using the proximity metric over K items in D dimension that defines the closeness of association such that X € RD and Yp € G.

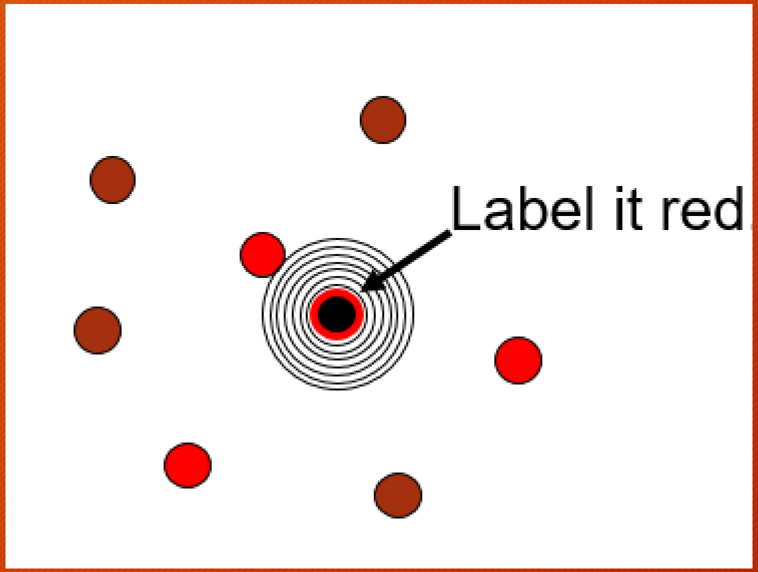
- Idea:
    - Similar examples have similar label.
    - Classify new examples like similar training examples.
- Algorithm:
    - Given some new example x for which we need to predict its class y
    - Find most similar training examples
    - Classify *x* “like” these most similar examples
- Questions:
    - How to determine similarity?
    - How many similar training examples to consider?
    - How to resolve inconsistencies among the training examples?

KNN Algorithm

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

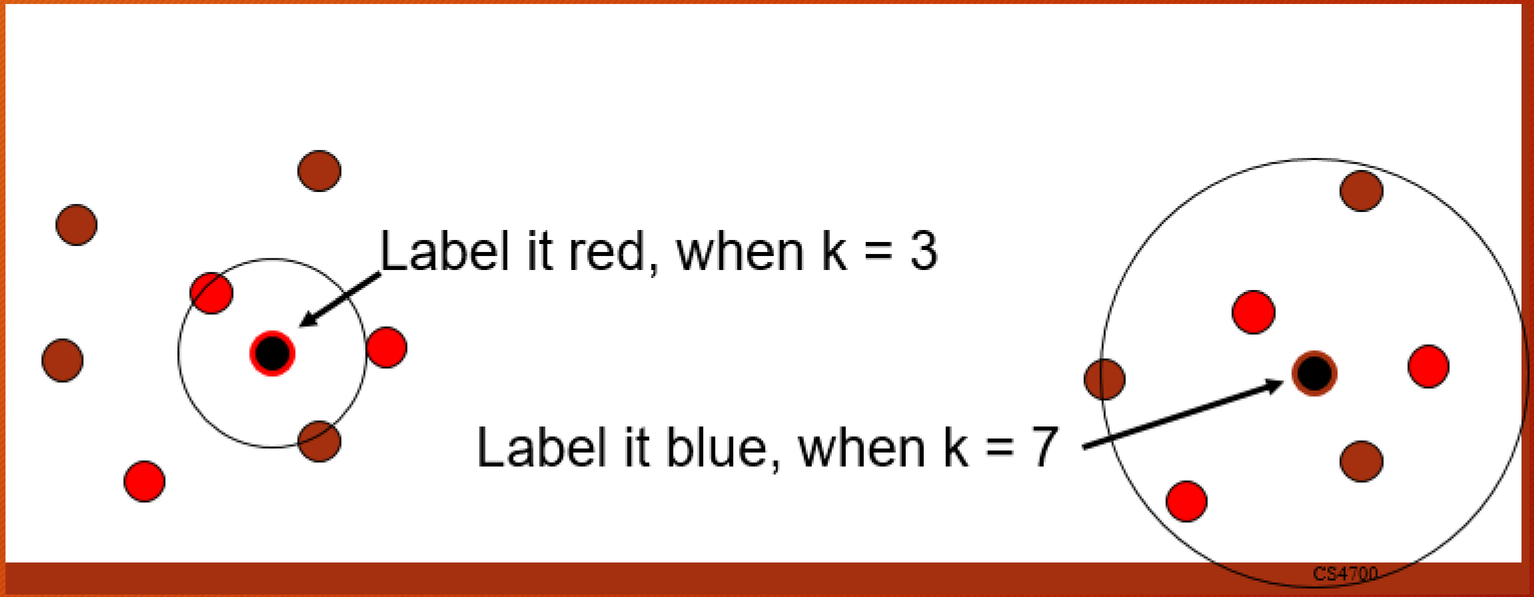
1-Nearest Neighbor

- One of the simplest of all machine learning classifiers
- Simple idea: label a new point the same as the closest known point



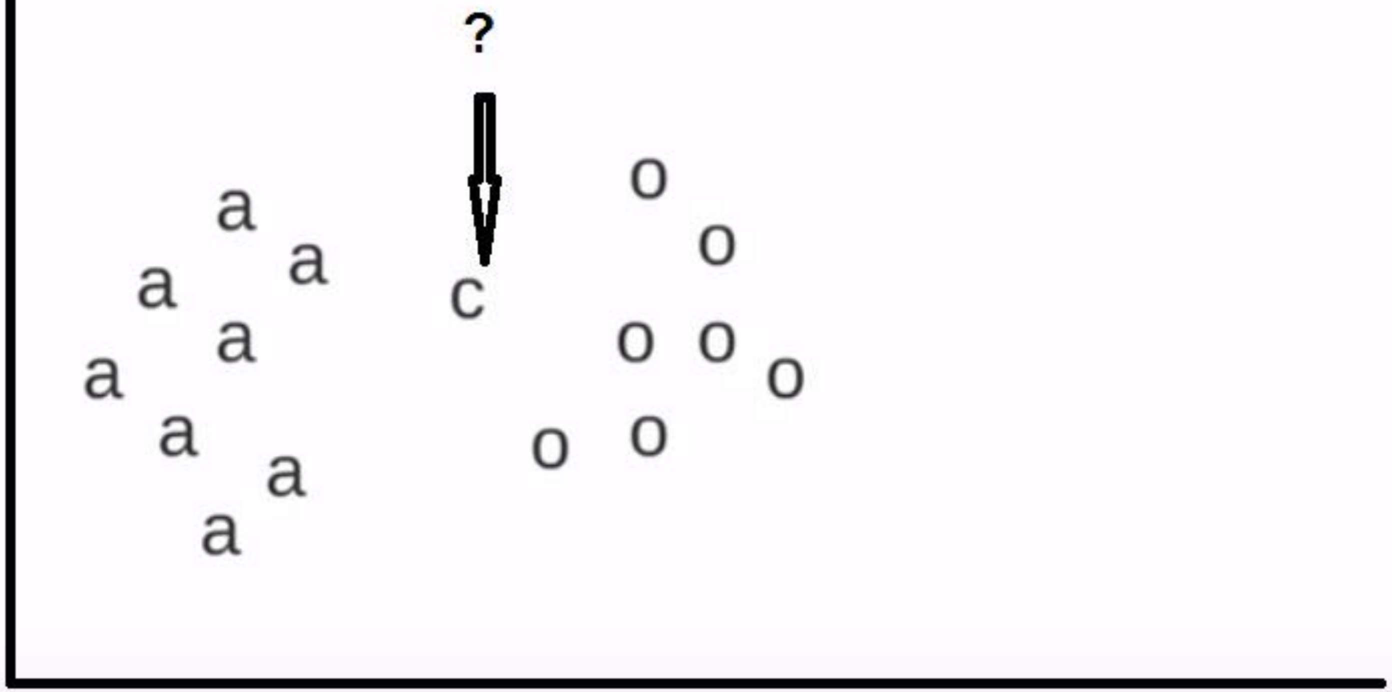
k – Nearest Neighbor

- Generalizes 1-NN to smooth away noise in the labels
- A new point is now assigned the most frequent label of its *k* nearest neighbors

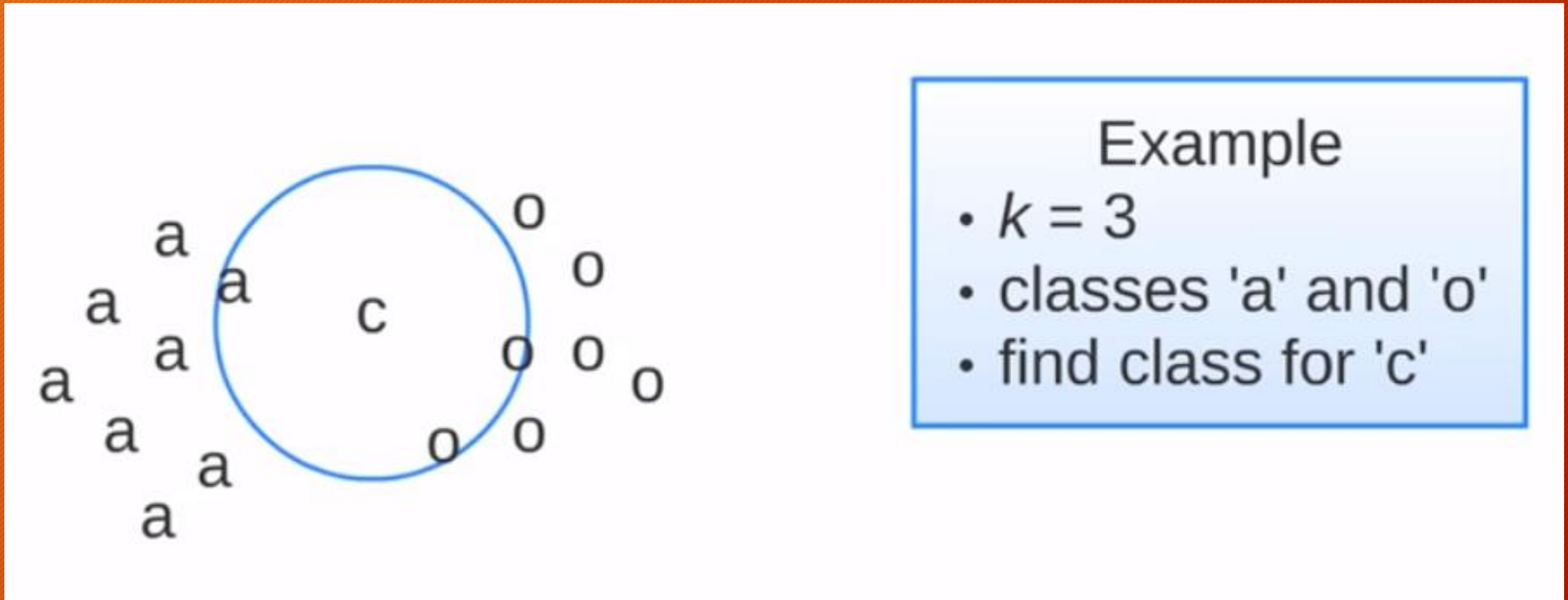


Simple example

Given N training vectors, *k*NN algorithm identifies the *k* nearest neighbors of 'c', regardless of labels



C ?


a
a
a
a
a
a
a
a
a
c
o
o
o
o
o
o
o
o
Example
• k = 3
• classes 'a' and 'o'
• find class for 'c'

Numerical Example 1

Customer	Age	Income	No. credit cards	Class	Distance from John
George	35	35K	3	No	$\text{sqrt}\ [(35-37)^2+(35-50)^2+(3-2)^2]=15.16$
Rachel	22	50K	2	Yes	$\text{sqrt}\ [(22-37)^2+(50-50)^2+(2-2)^2]=15$
Steve	63	200K	1	No	$\text{sqrt}\ [(63-37)^2+(200-50)^2+(1-2)^2]=152.23$
Tom	59	170K	1	No	$\text{sqrt}\ [(59-37)^2+(170-50)^2+(1-2)^2]=122$
Anne	25	40K	4	Yes	$\text{sqrt}\ [(25-37)^2+(40-50)^2+(4-2)^2]=15.74$
John	37	50K	2	?	

Numerical Example 2

Age	Loan	Default	Distance
25	$40,000	N	102000
35	$60,000	N	82000
45	$80,000	N	62000
20	$20,000	N	122000
35	$120,000	N	22000
52	$18,000	N	124000
23	$95,000	Y	47000
40	$62,000	Y	80000
60	$100,000	Y	42000
48	$220,000	Y	78000
33	$150,000	Y	8000
48	**$142,000**	**?**	

Euclidean Distance

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Numerical Example 3

Age	Loan	Default	Distance
0.125	0.11	N	0.7652
0.375	0.21	N	0.5200
0.625	0.31	N	0.3160
0	0.01	N	0.9245
0.375	0.50	N	0.3428
0.8	0.00	N	0.6220
0.075	0.38	Y	0.6669
0.5	0.22	Y	0.4437
1	0.41	Y	0.3650
0.7	1.00	Y	0.3861
0.325	0.65	Y	0.3771
0.7	0.61	?	

Standardized Variable

$$X_s = \frac{X - Min}{Max - Min}$$

Similarity Metrics

Similarity Measure	Data Format
Contingency Table, Jaccard coefficient, Distance Measure	Binary
Z-Score, Min-Max Normalization, Distance Measures	Numeric
Cosine Similarity, Dot Product	Vectors

Distance Measure

Euclidean Distance:

$X = \langle x_1, x_2, \cdots, x_n \rangle$ $Y = \langle y_1, y_2, \cdots, y_n \rangle$

$dist(X,Y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}$

Ex: Given X = {-2,2} & Y = {2,5}

Euclidean Distance = dist(X,Y) = [(-2-2)^2 + (2-5)^2]^(1/2)

= dist(X,Y) = (16 + 9)^(1/2)

= dist(X,Y) = 5

Distance Measure

- For the numeric data let us consider some distance measures:
  - Manhattan Distance:

$$X = \langle x_1, x_2, \cdots, x_n \rangle \quad Y = \langle y_1, y_2, \cdots, y_n \rangle$$

$$dist(X,Y) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|$$

  - Ex: Given X = {1,2} & Y = {2,5}
    Manhattan Distance = dist(X,Y) = |1-2|+|2-5|
    = 1+3
    = 4

Selecting the Number of Neighbors

- Increase k:
  - Makes KNN less sensitive to noise
- Decrease k:
  - Allows capturing finer structure of space
- **Pick k not too large, but not too small (depends on data)**

Curse-of-Dimensionality

- Prediction accuracy can quickly degrade when number of attributes grows.
  - Irrelevant attributes easily “swamp” information from relevant attributes
  - When many irrelevant attributes, similarity/distance measure becomes less reliable
- Remedy
  - Try to remove irrelevant attributes in pre-processing step
  - Weight attributes differently
  - Increase k (but not too much)

Advantages and Disadvantages of KNN

- Need distance/similarity measure and attributes that “match” target function.

- For large training sets,
- → Must make a pass through the entire dataset for each classification. This can be prohibitive for large data sets.

- Prediction accuracy can quickly degrade when number of attributes grows.