SNS COLLEGE OF ENGINEERING

(Autonomous)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Artificial Intelligence & Machine Learning

K-Nearest Neighbor Algorithm

Prepared by,

P.Ramya

Assistant Professor/ECE

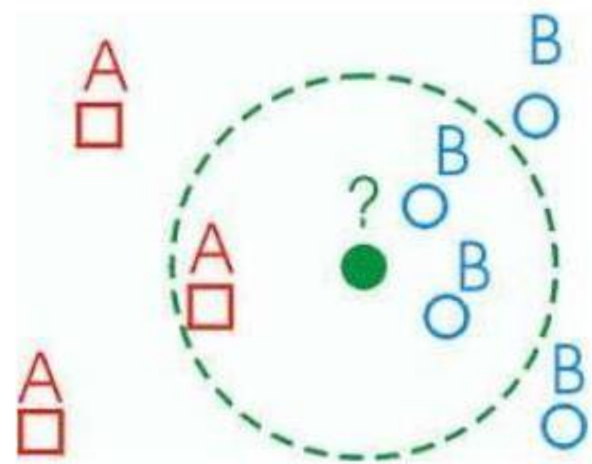
SNS College of Engineering

KNN

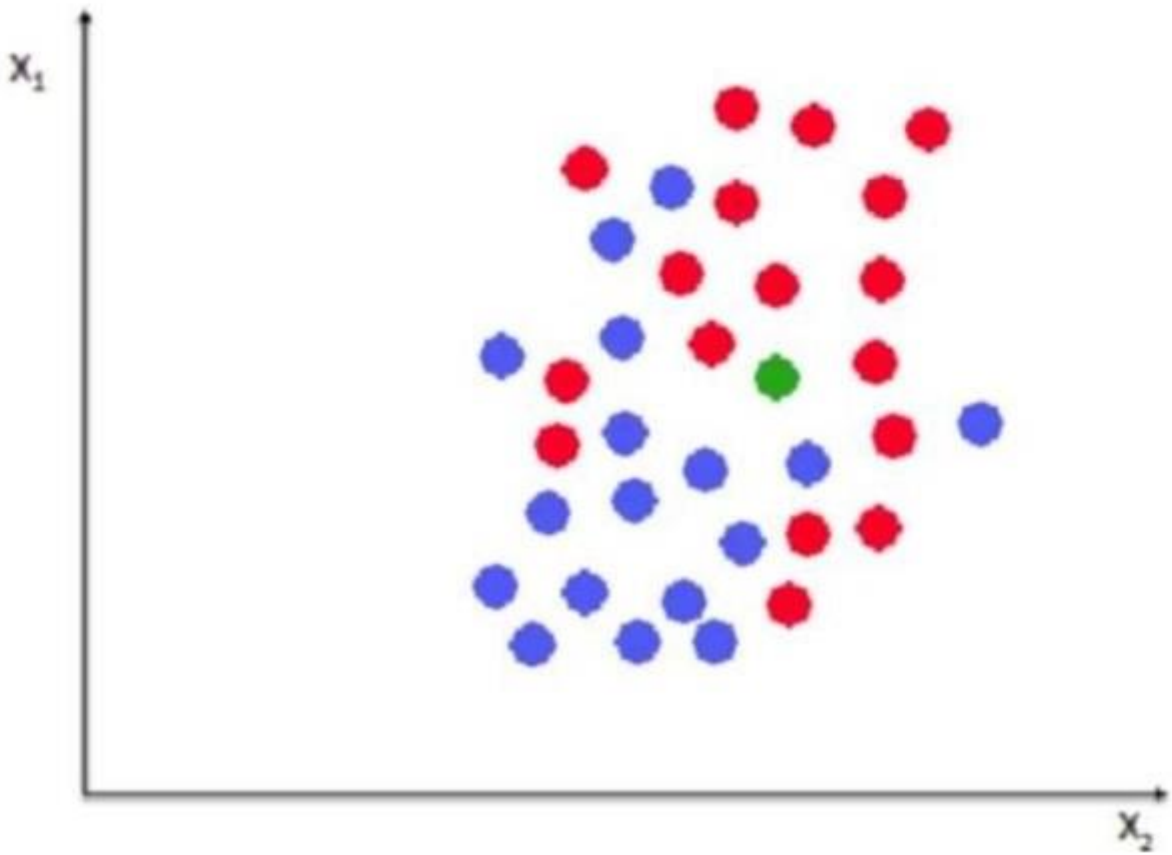
- K-Nearest Neighbors (KNN)
- Simple, but a very powerful classification algorithm
- Classifies based on a **similarity measure**
- Non-parametric
- Lazy learning
  - Does not “learn” until the test example is given
  - Whenever we have a new data to classify, we find its K-nearest neighbors from the training data

KNN: Classification Approach

- Classified by “**MAJORITY VOTES**” for its neighbor
classes
- Assigned to the most common class amongst its *K*-nearest neighbors (by measuring “distant” between data)



KNN: Example

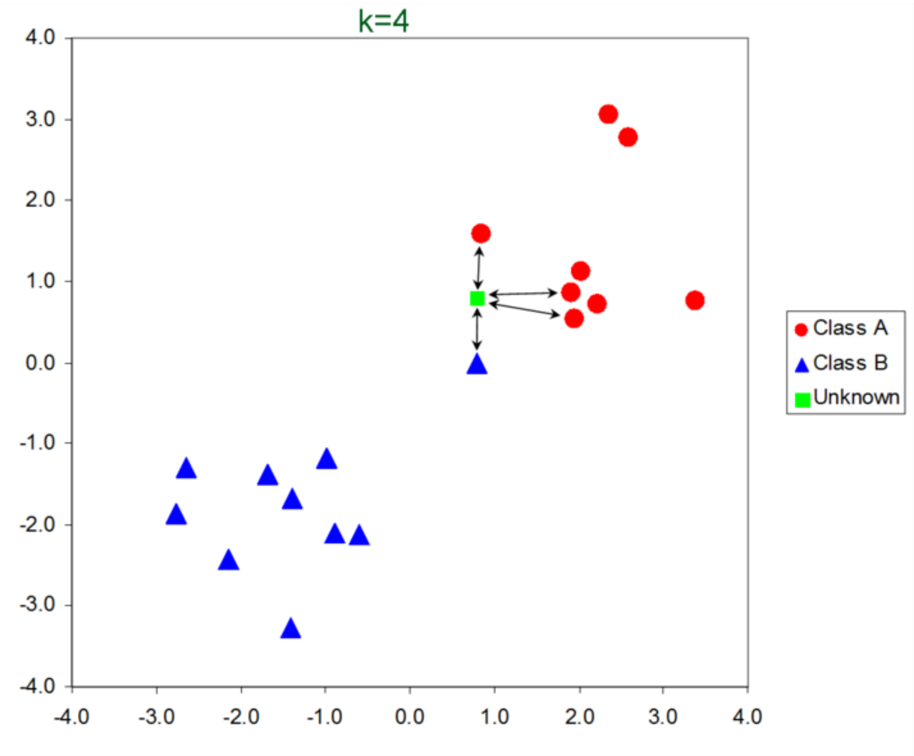


KNN: Pseudocode

- Step 1: Determine parameter K = number of nearest neighbors
- Step 2: Calculate the distance between the query-instance and all the training examples.
- Step 3: Sort the distance and determine nearest neighbors based on the k-th minimum distance.
- Step 4:Gather the category Y of the nearest neighbors.
- Step 5: Use simple majority of the category of nearest neighbors as the prediction value of the query instance.

Ref: https://www.slideshare.net/PhuongNguyen6/text-categorization

KNN: Example


k=4
4.0
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-4.0
-4.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
4.0
Class A
Class B
Unknown

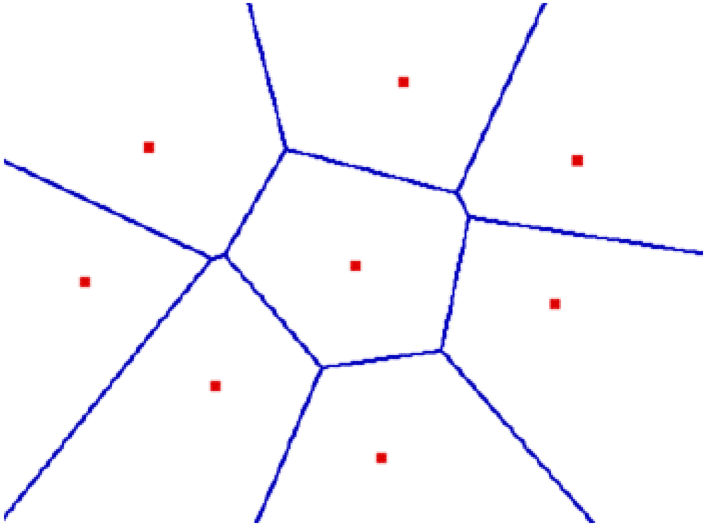
KNN: Euclidean distance matrix

Table 1. Euclidean distance matrix D listing all possible pairwise Euclidean distances between 19 samples.

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{x}_{15}$	$\mathbf{x}_{16}$	$\mathbf{x}_{17}$	$\mathbf{x}_{18}$
$\mathbf{x}_2$	1.5																	
$\mathbf{x}_3$	1.4	1.6																
$\mathbf{x}_4$	1.6	1.4	1.3															
$\mathbf{x}_5$	1.7	1.4	1.5	1.5														
$\mathbf{x}_6$	1.3	1.4	1.4	1.5	1.4													
$\mathbf{x}_7$	1.6	1.3	1.4	1.4	1.5	1.8												
$\mathbf{x}_8$	1.5	1.4	1.6	1.3	1.7	1.6	1.4											
$\mathbf{x}_9$	1.4	1.3	1.4	1.5	1.2	1.4	1.3	1.5										
$\mathbf{x}_{10}$	2.3	2.4	2.5	2.3	2.6	2.7	2.8	2.7	3.1									
$\mathbf{x}_{11}$	2.9	2.8	2.9	3.0	2.9	3.1	2.9	3.1	3.0	1.5								
$\mathbf{x}_{12}$	3.2	3.3	3.2	3.1	3.3	3.4	3.3	3.4	3.5	3.3	1.6							
$\mathbf{x}_{13}$	3.3	3.4	3.2	3.2	3.3	3.4	3.2	3.3	3.5	3.6	1.4	1.7						
$\mathbf{x}_{14}$	3.4	3.2	3.5	3.4	3.7	3.5	3.6	3.3	3.5	3.6	1.5	1.8	0.5					
$\mathbf{x}_{15}$	4.2	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	1.7	1.6	0.3	0.5				
$\mathbf{x}_{16}$	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	1.6	1.5	0.4	0.5	0.4			
$\mathbf{x}_{17}$	5.9	6.2	6.2	5.8	6.1	6.0	6.1	5.9	5.8	6.0	2.3	2.3	2.5	2.3	2.4	2.5		
$\mathbf{x}_{18}$	6.1	6.3	6.2	5.8	6.1	6.0	6.1	5.9	5.8	6.0	3.1	2.7	2.6	2.3	2.5	2.6	3.0	
$\mathbf{x}_{19}$	6.0	6.1	6.2	5.8	6.1	6.0	6.1	5.9	5.8	6.0	3.0	2.9	2.7	2.4	2.5	2.8	3.1	0.4

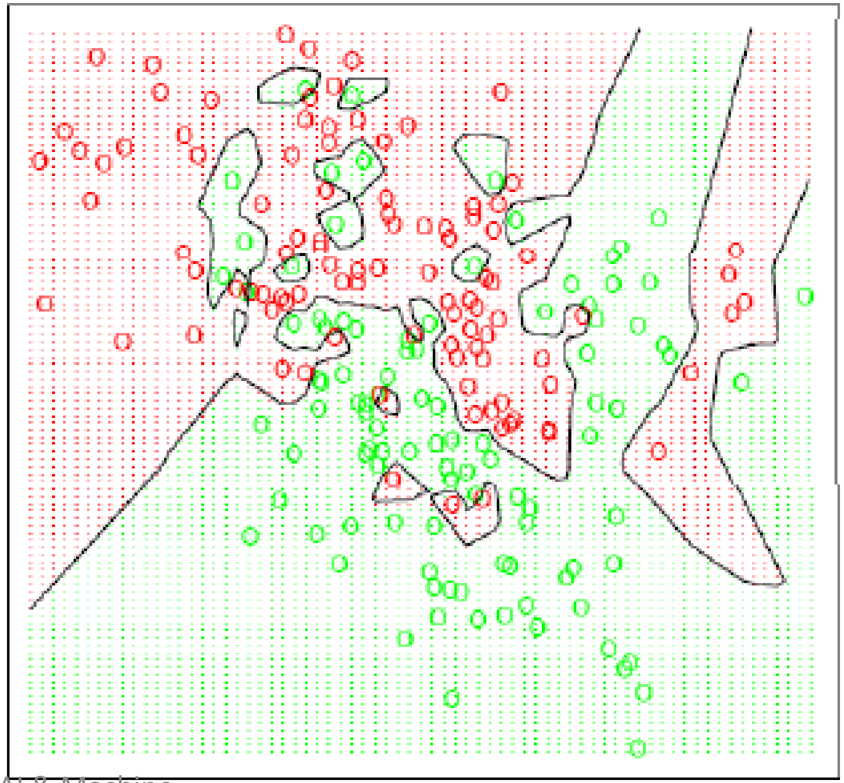
Decision Boundaries

- Voronoi diagram
- Describes the areas that are nearest to any given point, given a set of data.
- Each line segment is equidistant between two points of opposite class



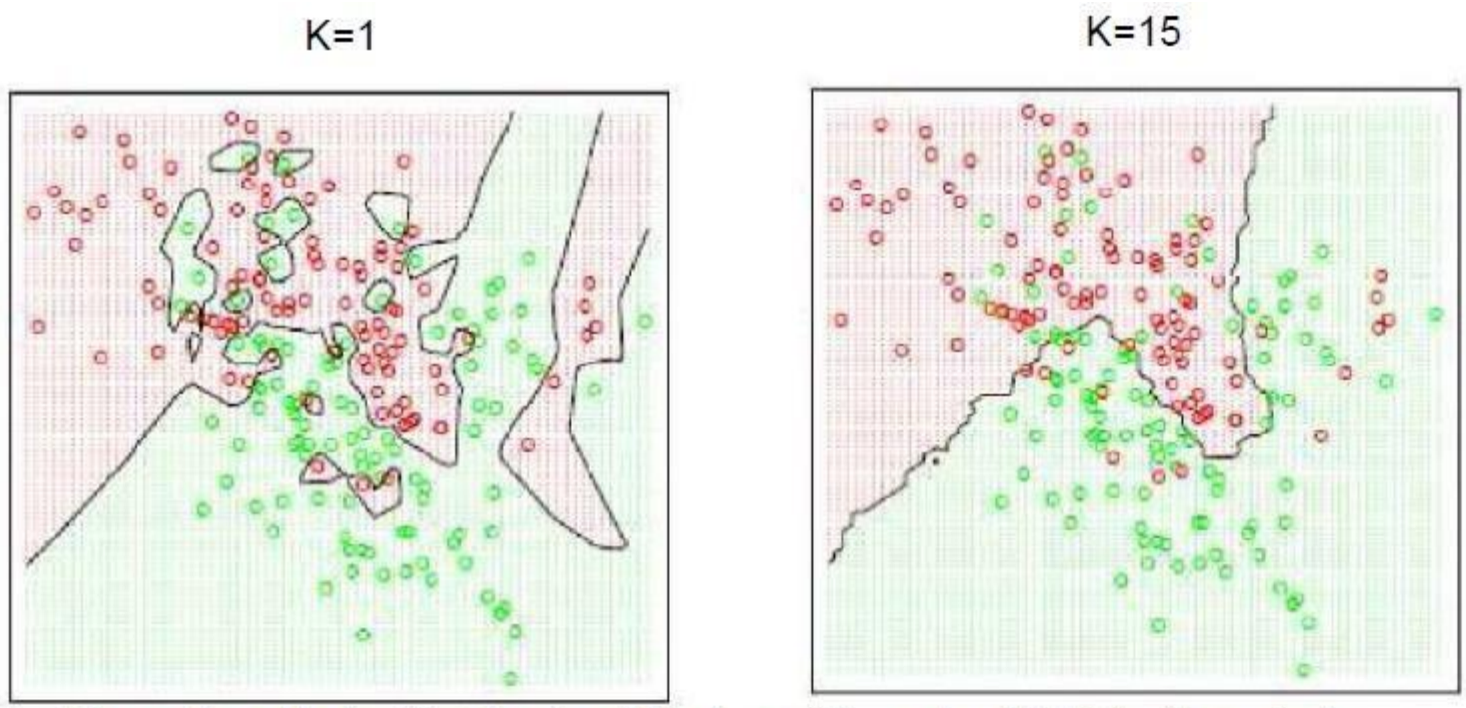
Decision Boundaries

- With large number of examples and possible noise in the labels, the decision boundary can become nasty!
- "Overfitting" problem



Effect of K

- Larger k produces smoother boundary effect
- When K==N, always predict the majority class



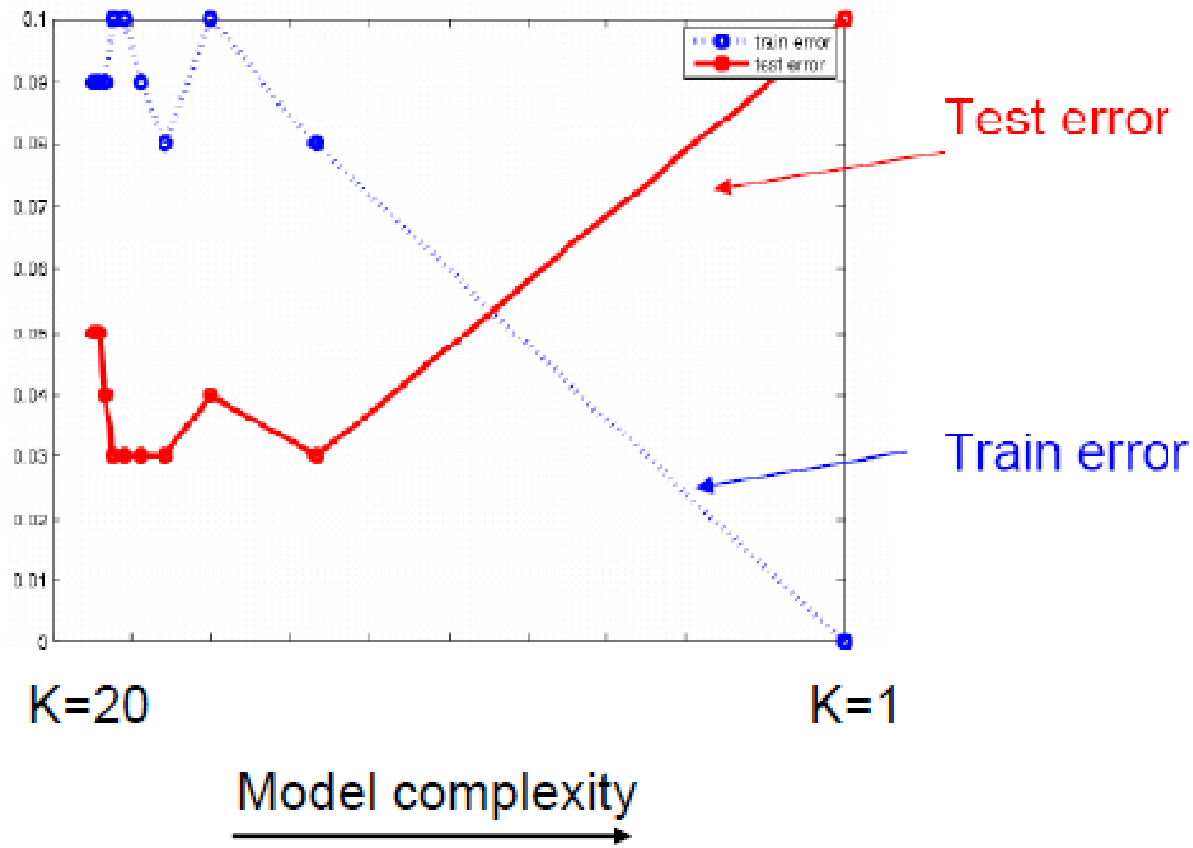
Figures from Hastie, Tibshirani and Friedman (Elements of Statistical Learning)

Discussion

- Which model is better between K=1 and K=15?
- Why?

How to choose k?

- Empirically optimal k?



Pros and Cons

- Pros
  - Learning and implementation is extremely simple and Intuitive
  - Flexible decision boundaries
- Cons
  - Irrelevant or correlated features have high impact and must be eliminated
  - Typically difficult to handle high dimensionality
  - Computational costs: memory and classification time computation

Similarity and Dissimilarity

- Similarity
  - Numerical measure of how alike two data objects are.
  - Is higher when objects are more alike.
  - Often falls in the range [0,1]
- Dissimilarity
  - Numerical measure of how different are two data objects
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0
  - Upper limit varies
- Proximity refers to a similarity or dissimilarity

Euclidean Distance

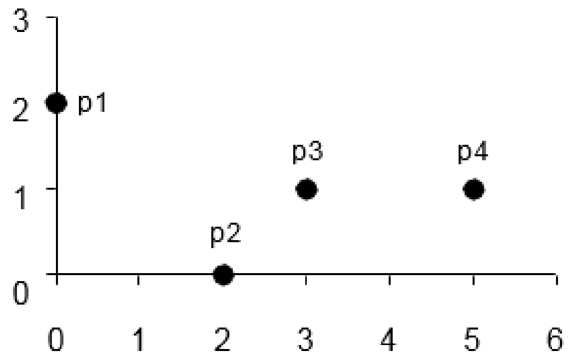
- Euclidean Distance

$$dist = \sqrt{\sum_{k=1}^{p} (a_k - b_k)^2}$$

Where p is the number of dimensions (attributes) and a_k and b_k are, respectively, the k-th attributes (components) or data objects a and b.

- Standardization is necessary, if scales differ.

Euclidean Distance



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Step by step process of Bayes Theorem

- Prior or State of Nature
- Class Conditional Probabilities
- Evidence
- Posterior Probabilities


thank
you