# INTRODUCTION TO MACHINE LEARNING

## K-NEAREST NEIGHBOR ALGORITHM

Sai Chandra Kosaraju
Department of Computer Science @ UNLV

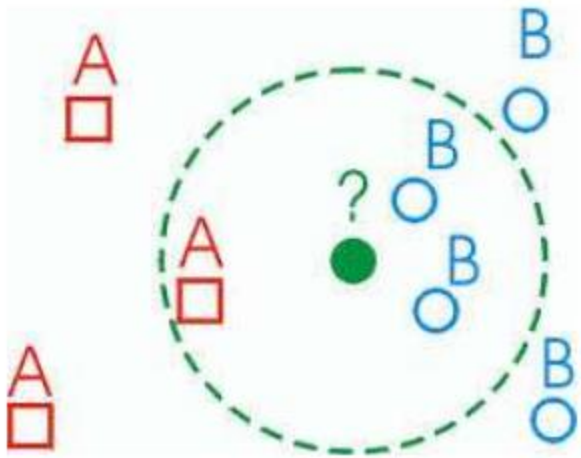
# KNN

- K-Nearest Neighbors (KNN)
- Simple, but a very powerful classification algorithm
- Classifies based on a **similarity measure**
- 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

Ref: https://www.slideshare.net/tilanigunawardena/k-nearest-neighbors

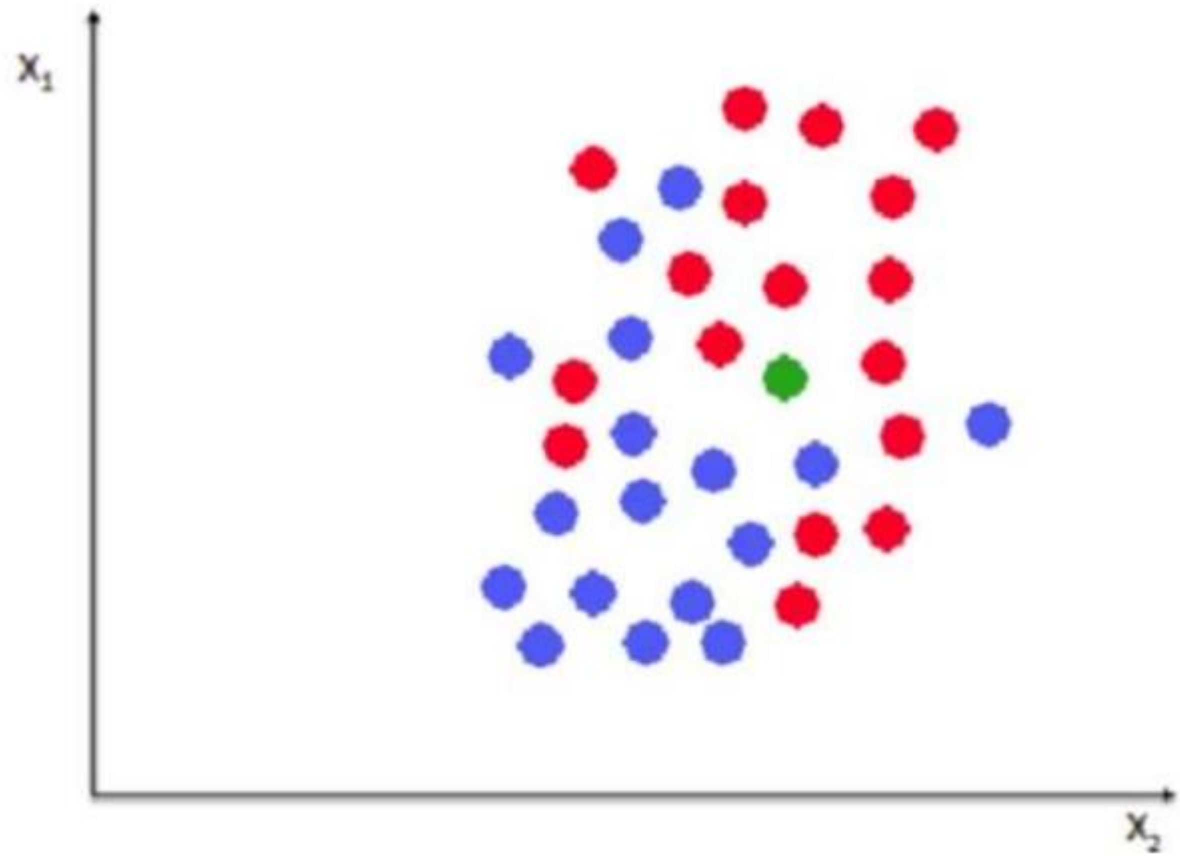
# KNN: Classification Approach

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




Ref: https://www.slideshare.net/tilanigunawardena/k-nearest-neighbors

# KNN: Example



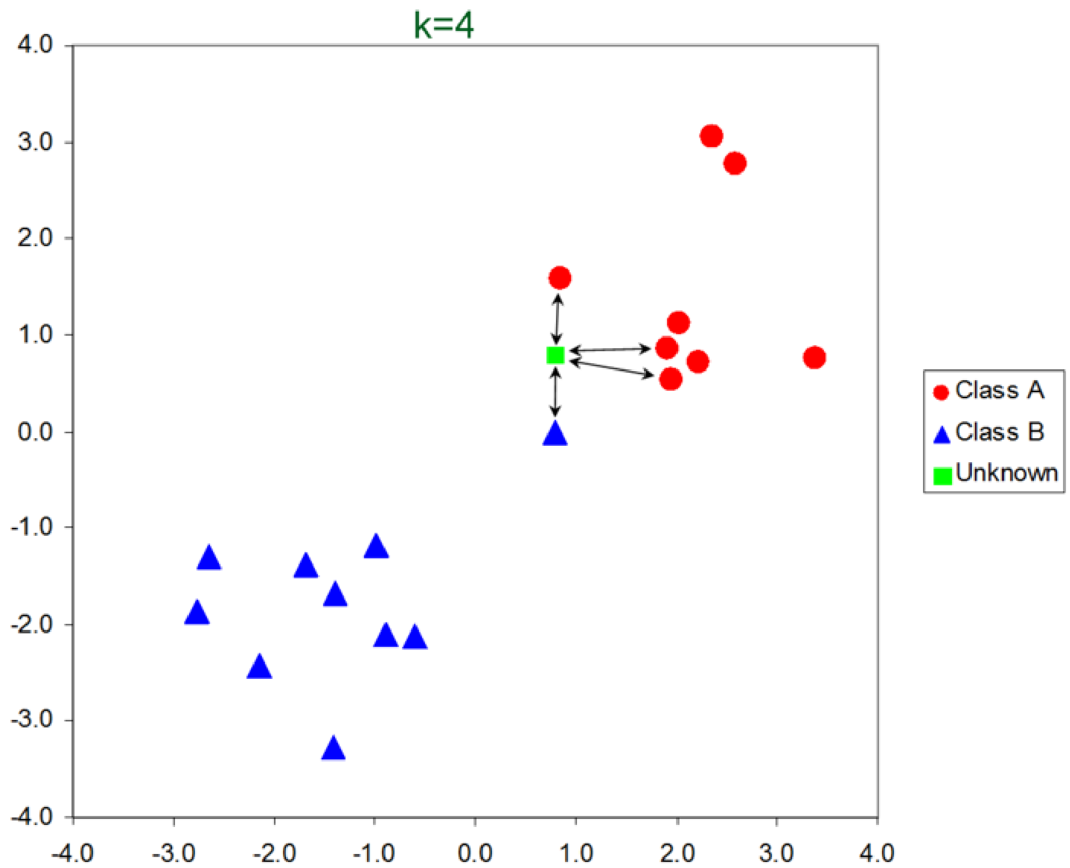

Ref: https://www.slideshare.net/tilanigunawardena/k-nearest-neighbors

# 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




Ref: http://www.scholarpedia.org/article/K-nearest_neighbor

# 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) of data objects a and b.

- Standardization is necessary, if scales differ.

# KNN: Euclidean distance matrix

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

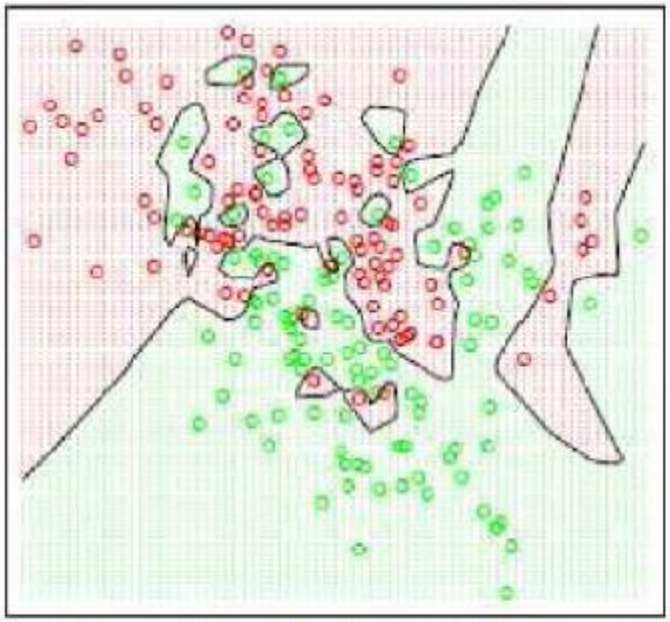
| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | $x_{10}$ | $x_{11}$ | $x_{12}$ | $x_{13}$ | $x_{14}$ | $x_{15}$ | $x_{16}$ | $x_{17}$ | $x_{18}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_2$ | 1.5 | | | | | | | | | | | | | | | | | |
| $x_3$ | 1.4 | 1.6 | | | | | | | | | | | | | | | | |
| $x_4$ | 1.6 | 1.4 | 1.3 | | | | | | | | | | | | | | | |
| $x_5$ | 1.7 | 1.4 | 1.5 | 1.5 | | | | | | | | | | | | | | |
| $x_6$ | 1.3 | 1.4 | 1.4 | 1.5 | 1.4 | | | | | | | | | | | | | |
| $x_7$ | 1.6 | 1.3 | 1.4 | 1.4 | 1.5 | 1.8 | | | | | | | | | | | | |
| $x_8$ | 1.5 | 1.4 | 1.6 | 1.3 | 1.7 | 1.6 | 1.4 | | | | | | | | | | | |
| $x_9$ | 1.4 | 1.3 | 1.4 | 1.5 | 1.2 | 1.4 | 1.3 | 1.5 | | | | | | | | | | |
| $x_{10}$ | 2.3 | 2.4 | 2.5 | 2.3 | 2.6 | 2.7 | 2.8 | 2.7 | 3.1 | | | | | | | | | |
| $x_{11}$ | 2.9 | 2.8 | 2.9 | 3.0 | 2.9 | 3.1 | 2.9 | 3.1 | 3.0 | 1.5 | | | | | | | | |
| $x_{12}$ | 3.2 | 3.3 | 3.2 | 3.1 | 3.3 | 3.4 | 3.3 | 3.4 | 3.5 | 3.3 | 1.6 | | | | | | | |
| $x_{13}$ | 3.3 | 3.4 | 3.2 | 3.2 | 3.3 | 3.4 | 3.2 | 3.3 | 3.5 | 3.6 | 1.4 | 1.7 | | | | | | |
| $x_{14}$ | 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 | | | | | |
| $x_{15}$ | 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 | | | | |
| $x_{16}$ | 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 | | | |
| $x_{17}$ | 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 | | |
| $x_{18}$ | 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 | |
| $x_{19}$ | 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 |

Ref: http://www.scholarpedia.org/article/K-nearest_neighbor

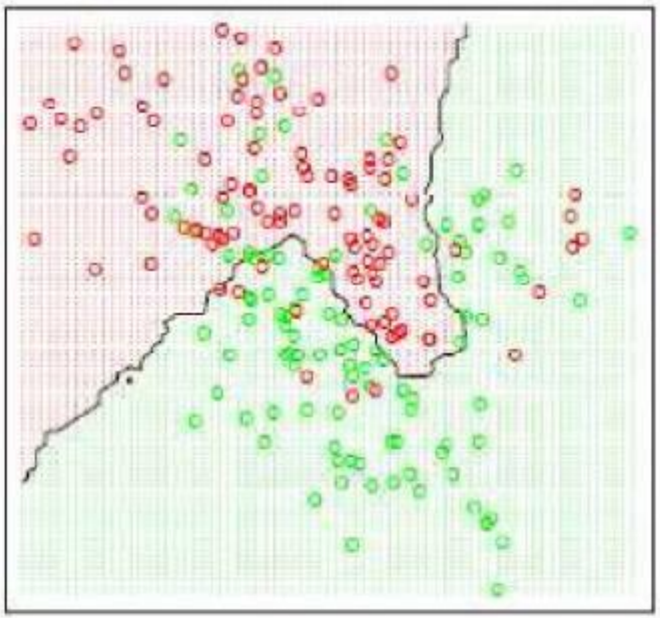
# Effect of K

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

K=1



K=15



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

Ref: https://www.slideshare.net/tilanigunawardena/k-nearest-neighbors

# Discussion

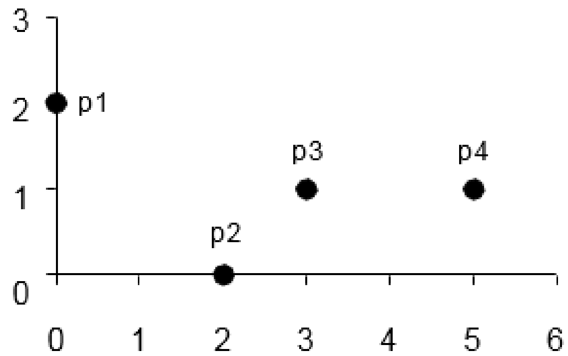
- Which model is better between K=1 and K=15?
- Why?

# 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

Ref: https://www.slideshare.net/tilanigunawardena/k-nearest-neighbors

# 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 |

# Discussion

- Can we use KNN for regression problems?