Machine Learning: Classification versus clustering

by Elena Battini Sönmez
İstanbul Bilgi University

The Classification problem:

- We start with a database of objects whose classes are already known
  - The database is known as the training database, since it trains us to know what the different types of things look like
- We take a new sample, and we want to know its class

Example of classification:

- Suppose we have a database storing info of different people, together with their credit rating

How much they earn, whether they own their house, how old they are, etc.

- We want to be able to use this database to give a credit to a new person

Intuitively, we want to give similar credit ratings to similar people

The k-Nearest Neighbours:

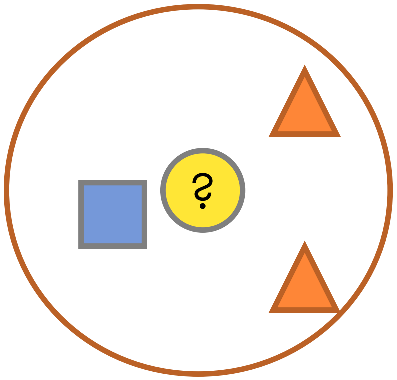
- The k-Nearest Neighbours (k-NN) classification algorithm considers the k-neighbours of the test sample and assigns it to the majority of the class
- Question: What makes two items count as similar, and how do we measure similarity?

Euclidean distance:

- The k-NN algorithm interprets each object in the database as a point in the space; that is, each attribute is a feature, a coordinate in the plane
- The similarity of two points is measured as the distance between them

$$\text{Euclidean\_dist}\ ((x,y),(a,b)) = \sqrt{(x-a)^2 + (y-b)^2}$$

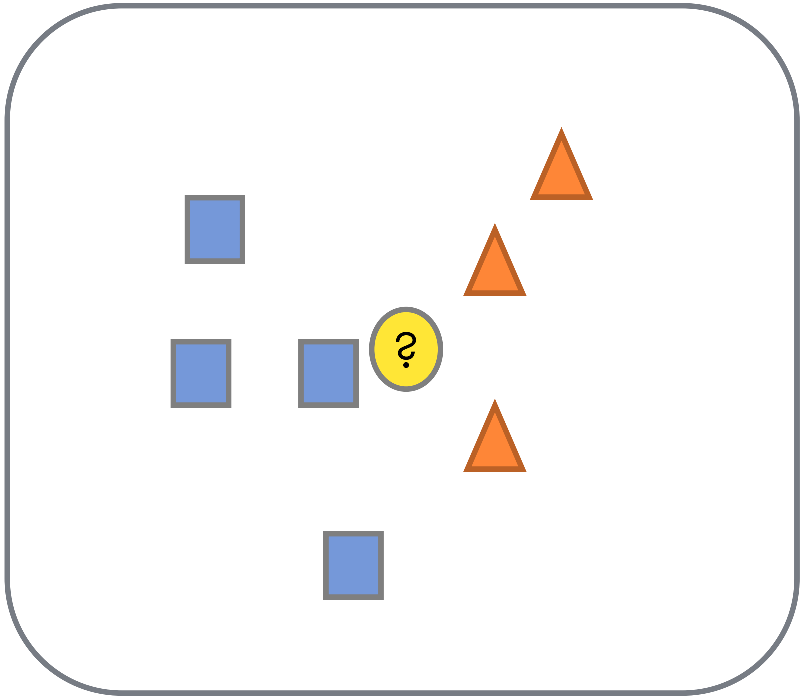
k-NN Algorithm:



- It requires:
  1. The set of stored labeled records (training set)
  2. A distance metric to compute the distance between records
  3. The value of *k*, the number of nearest neighbors to consider

- To classify an unknown record (test sample):
  - Compute distance to all other training records
  - Identify *k* nearest neighbors
  - Use class labels of nearest training samples to assign the class (e.g., by taking majority vote) to the test sample

Challenges of k-NN:



- Choosing the value of *k*:
  - If *k* is too small, sensitive to noise points
  - If *k* is too large, neighborhood may include points from other classes
  - Choose an odd value for *k*, to eliminate ties

Q: Give the class for K=1, 3, 5

Problems of k-NN:

- Computationally intensive, especially when the size of the training set grows
- High dimension
- Accuracy can be severely degraded by the presence of noisy or irrelevant features

Clustering:

- The process of organizing objects into groups whose members are similar in some way

A *cluster* is therefore a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters

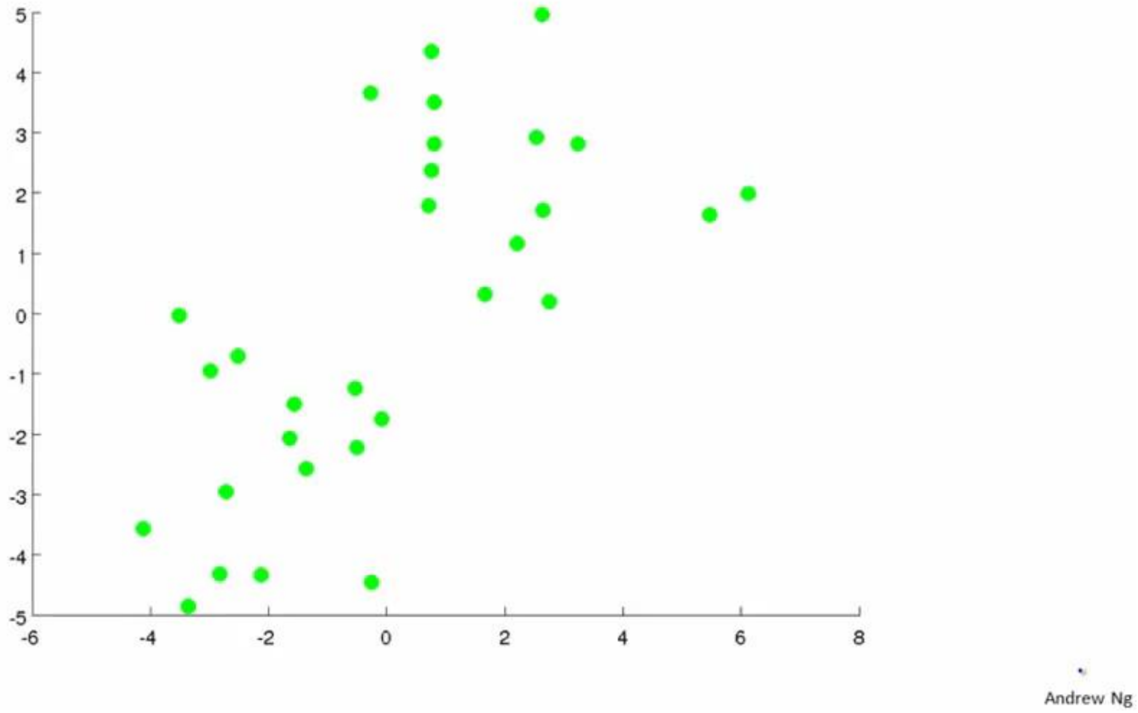
- The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data

Example of clustering:

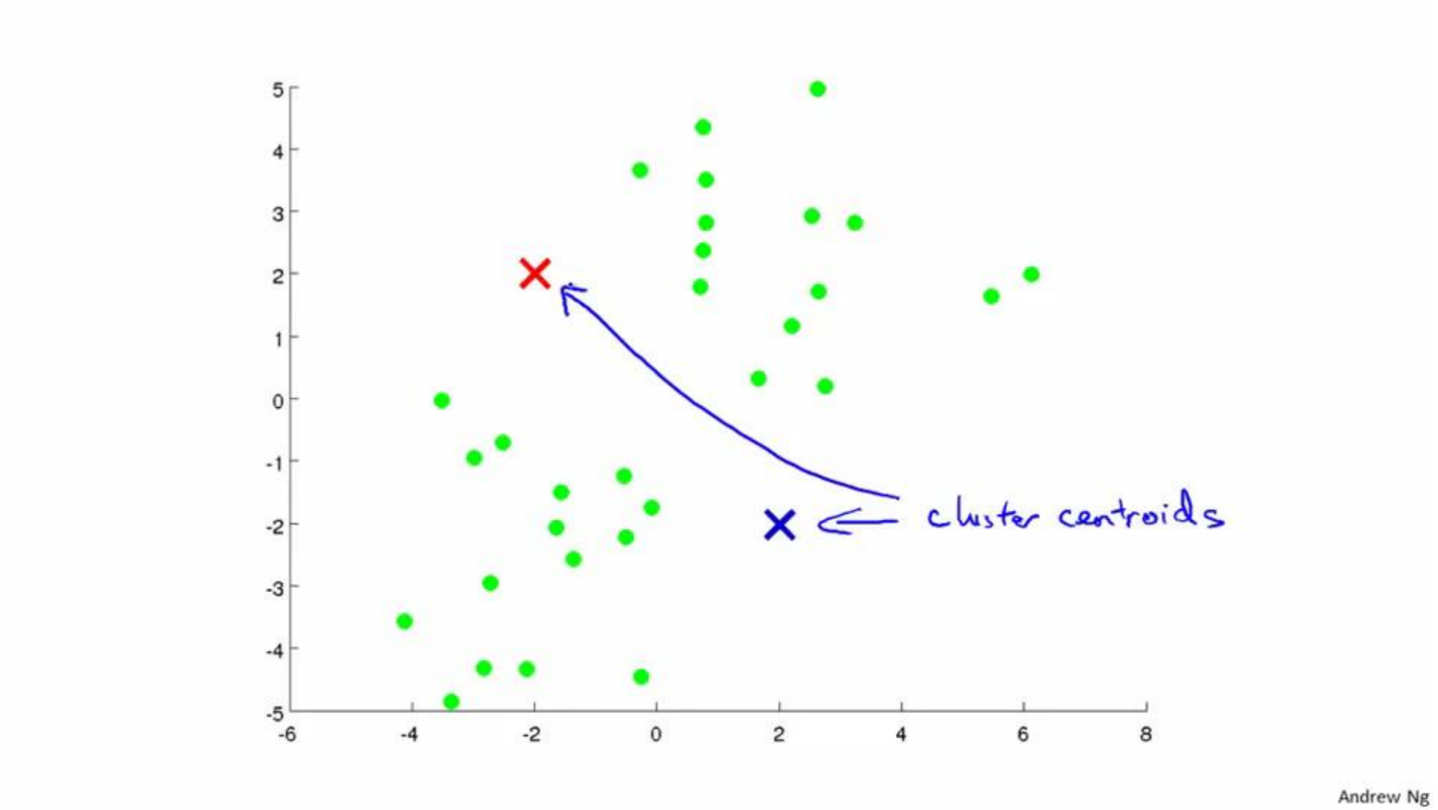
- *Marketing*: finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records;
- *Biology*: classification of plants and animals given their features;
- *Libraries*: book ordering;
- *Insurance*: identifying groups of motor insurance policy holders with a high average claim cost; identifying frauds;
- *WWW*: document classification; clustering weblog data to discover groups of similar access patterns.

K-means algorithm (1/6):

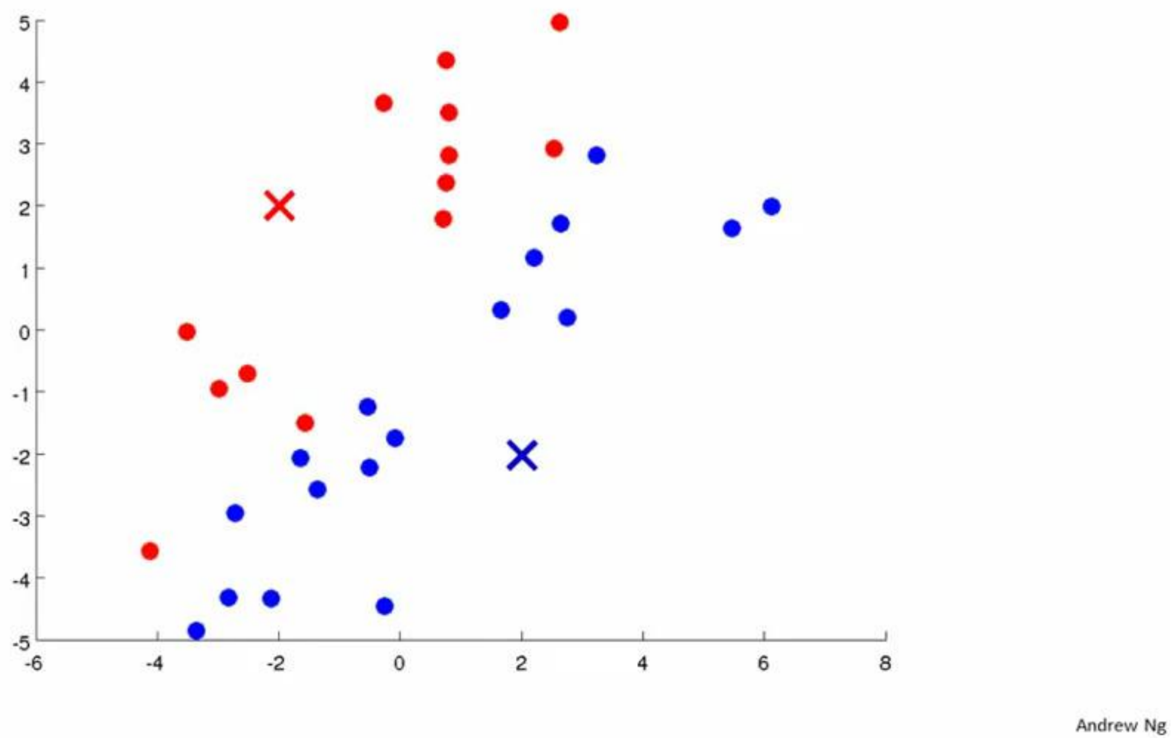
(https://class.coursera.org/ml-005/lecture/78)



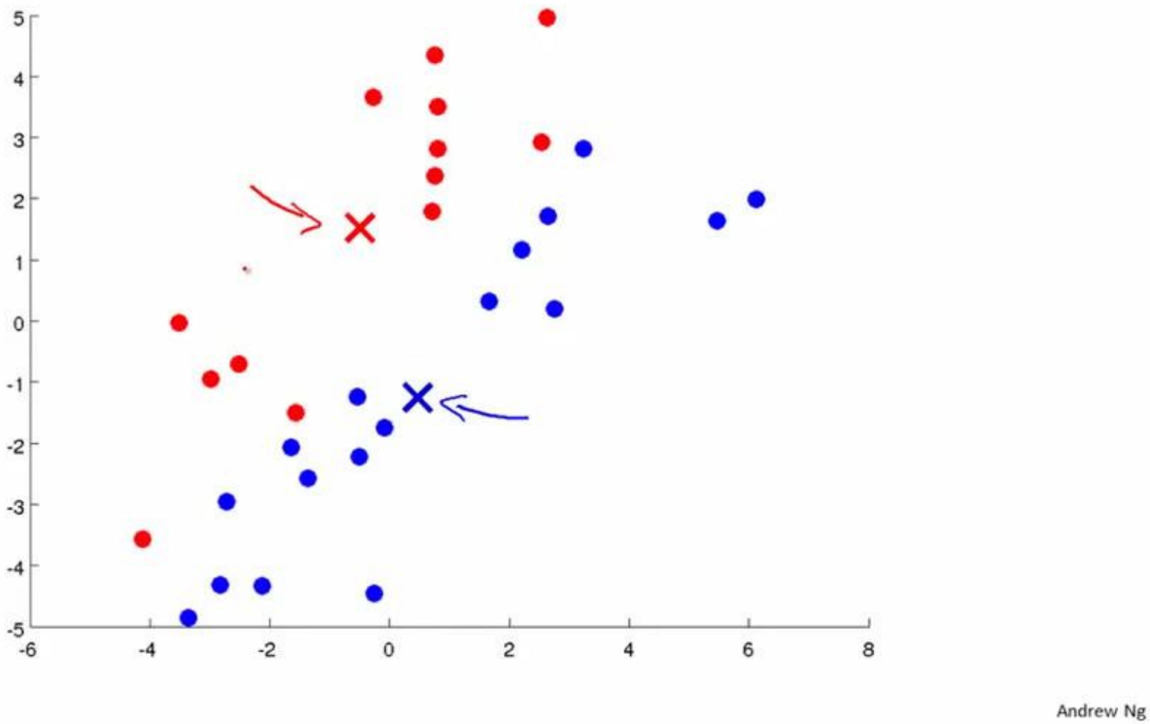
K-means algorithm (2/6):



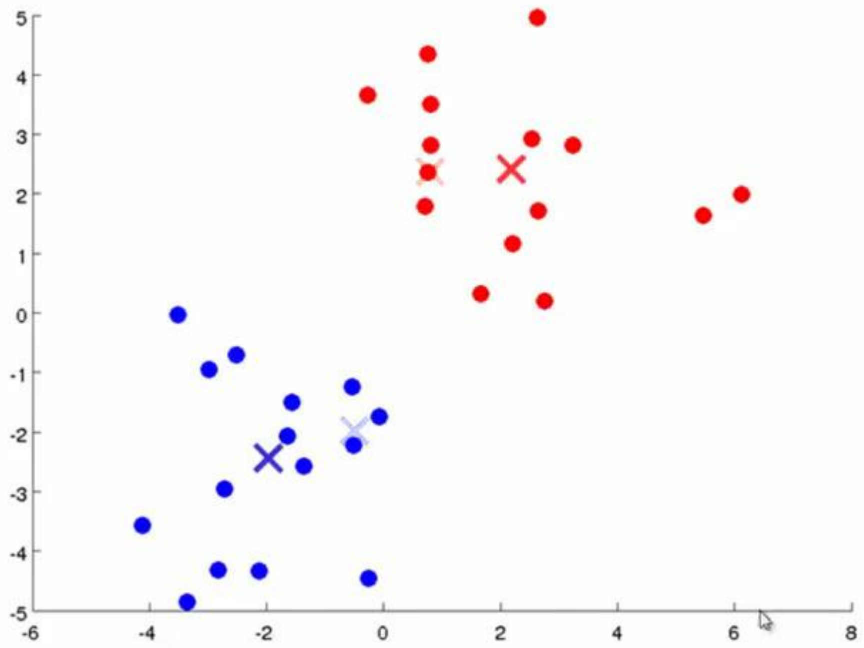
K-means algorithm (3/6):



K-means algorithm (4/6):



K-means algorithm (5/6):



Andrew Ng

K-means algorithm (6/6):

K-means algorithm

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

Repeat {

for i = 1 to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

for k = 1 to K

μ_k := average (mean) of points assigned to cluster k

}

Andrew Ng

Requirements of clustering algorithms:

- scalability
- dealing with different types of attributes
- discovering clusters with arbitrary shape
- ability to deal with noise and outliers
- high dimensionality