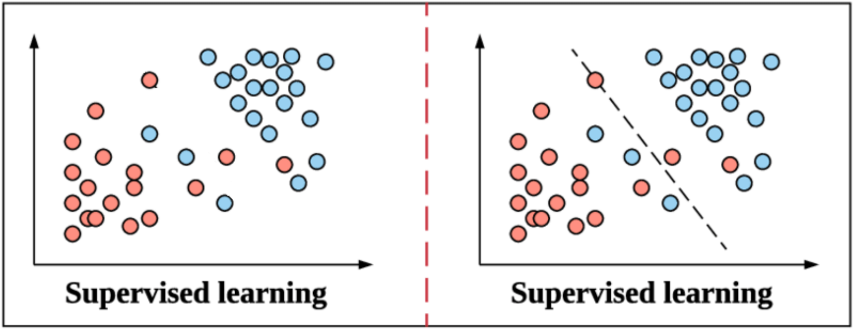
Machine Learning 2018 – Clustering

Kien C Nguyen

April 27, 2020

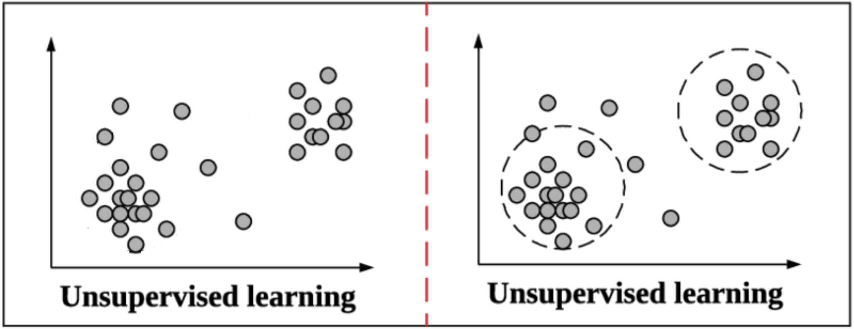
1. Supervised Learning
   - Input : data X and label Y
   - Goal : find parameters w that minimize the loss function
2. Why Supervised Learning?
   - predict outcomes from previous experiences

Figure: (Source: Orchestrating Development Lifecycle of Machine Learning Based IoT Applications: A Survey, Zhenyu Wen)

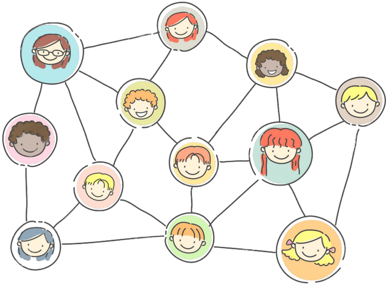


Introduction

1. Unsupervised Learning
   - Input : data X
   - Goal : group data by finding some commonality in the features
2. Why Supervised Learning?
   - find features which can be useful for categorization
   - find all kind of unknown patterns in data



1. Applications
   - Spam email filter
   - Marketing and Sales
   - Social Network



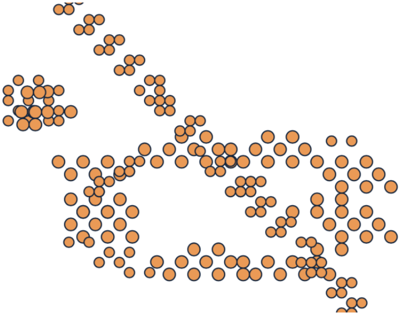
Introduction - Clustering

- Input : $S = \{x^{(i)}\}_{i=1}^{N}$ (N: number of samples), each sample (data point) is a D-dimensional vector

$$x^i = (x_1^i, x_2^i, \ldots, x_D^i)^T$$

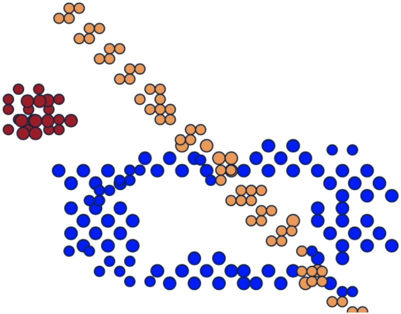
- Output: find structure in the data and organize them into groups.

Figure: Input samples. (Source: UIUC CS446 Lecture notes [1])



- A cluster is a set of samples that are alike
- Samples in different clusters are not alike

Figure: Clustered input samples. (Source: UIUC CS446 Lecture notes [1])



Distance Measures

- A distance measure (metric) is a function $d : R^D \times R^D \to R$ that satisfies
  1. $d(\mathbf{x}, \mathbf{y}) \geq 0, \quad d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$
  2. $d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{z})$ (Triangle inequality)
  3. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ (Symmetry)
- For the purpose of clustering, sometimes we can use distances that are not a metric (e.g. those that do not satisfy triangle inequality or symmetry.)

Distance Measures

- L^2 distance (Euclidean distance)

$$\begin{aligned} d(\mathbf{x}, \mathbf{y}) &= \|(\mathbf{x} - \mathbf{y})\|_2 = \sqrt{(\mathbf{x} - \mathbf{y})^2} \\ &= \sqrt{(\mathbf{x} - \mathbf{y})^T(\mathbf{x} - \mathbf{y})} = \sqrt{\sum_{i=1}^{D} (x_i - y_i)^2} \end{aligned}$$

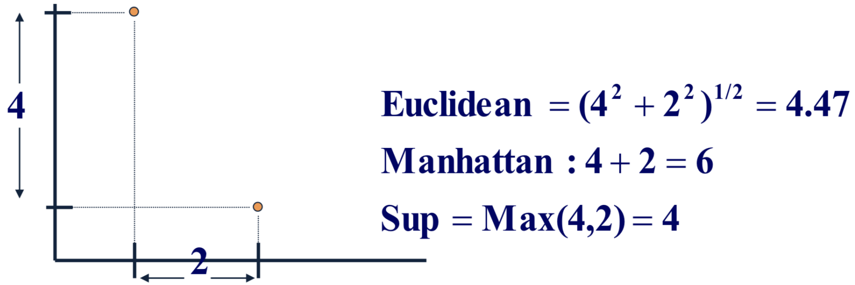
- L^1 distance (Manhattan distance)

$$d(\mathbf{x}, \mathbf{y}) = \|(\mathbf{x} - \mathbf{y})\|_1 = \sum_{i=1}^{D} |x_i - y_i|$$

- L^∞ distance (sup distance)

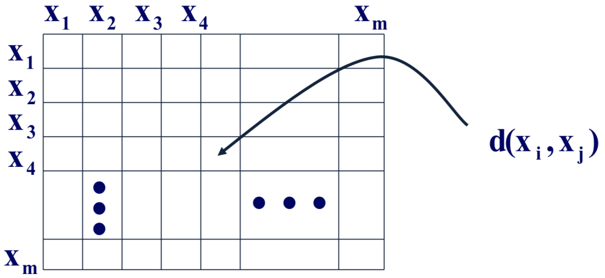
$$d(\mathbf{x}, \mathbf{y}) = \|(\mathbf{x} - \mathbf{y})\|_\infty = max_{1 \leq i \leq D} |x_i - y_i|$$

Figure: Different types of distance measures. (Source: UIUC CS446 Lecture notes [1])

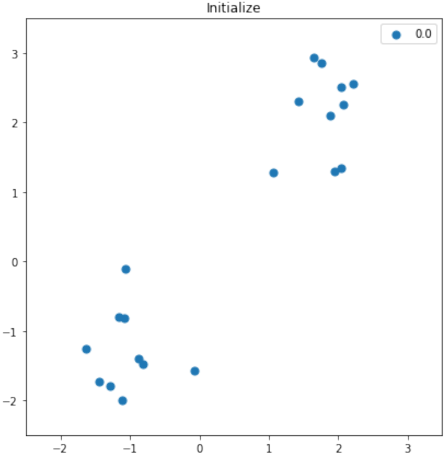


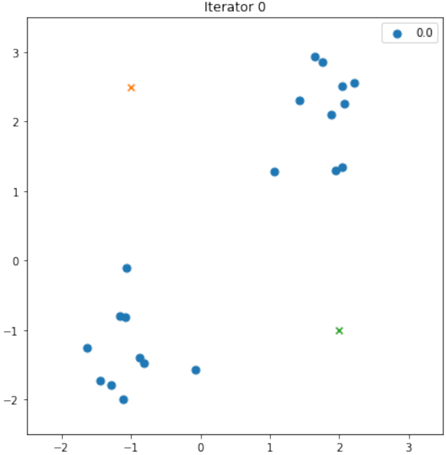
- We are given a matrix of distances between any pair of samples.

Figure: Matrix of distances. (Source: UIUC CS446 Lecture notes [1])

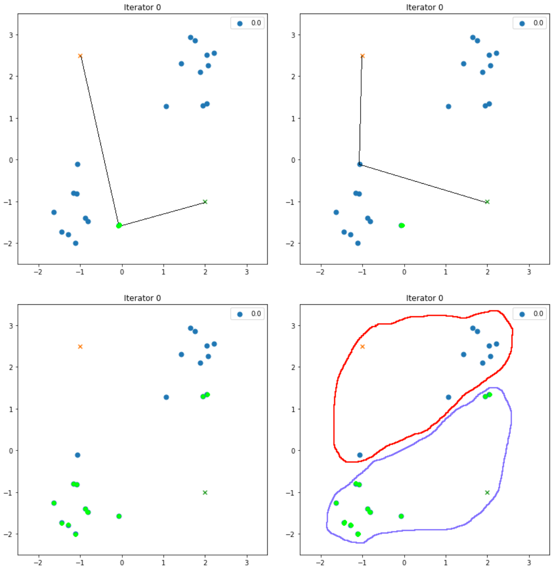


K-means Algorithm

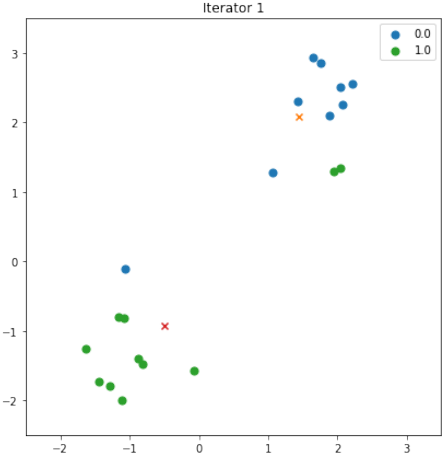




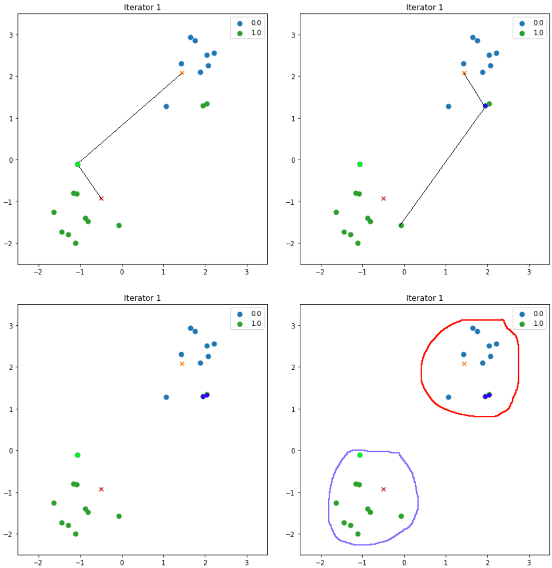
K-means Algorithm



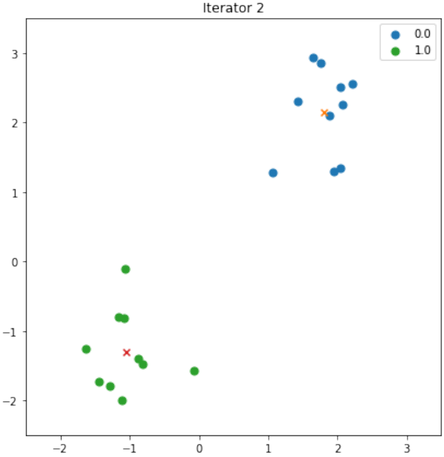
K-means Algorithm



K-means Algorithm



K-means Algorithm



Input:

- K (number of clusters)
- $\{x^{(i)}\}_{i=1}^{N}$

Initialization:

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

while *Assignment changes from the last iteration* **do**

  Assignment:

  for *$i = 1$ to N* **do**

    Assign $x^{(i)}$ to the cluster with the minimum distance $d(x^{(i)}, \mu_k)$

  end

  Update:

  for *j=1 to K* **do**

    μ_k = mean of all the points assigned to cluster k

  end

end

Algorithm 1: K-means Algorithm

Challenges of K-means

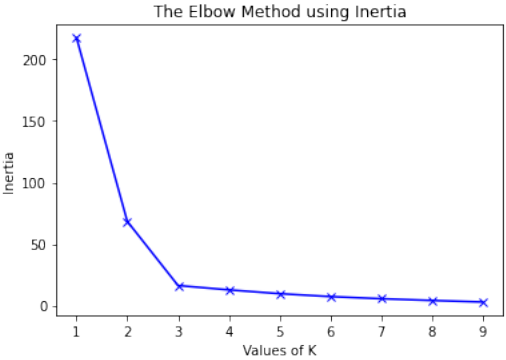
- Different K different outputs.
- With same K, the output won't be always the same because of the randomly initial centroids.
- Due to the nature of Euclidean distance, it is not a suitable algorithm when dealing with clusters that adopt non-spherical shapes.

How to choose right K

- Field knowledge
- Business decision
- Elbow Method

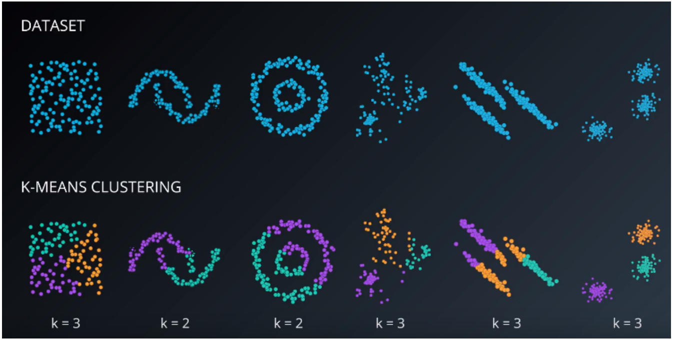
Elbow Method

- The elbow method is used for determining the correct number of clusters in a dataset.
- How it works? Plot the cost function against K and choose K using the "elbow" method.



K-means Limitations

- K-means clustering with spherical-shaped distributions



Another Unsupervised Learning

- Hierarchichal Clustering
- Density-Based Spatial Clustering of Applications with Noise (DBSCAN)
- Gaussian Mixture Models (GMM)
- Principal component analysis (PCA)

References

[1] UIUC CS 446 Machine Learning
[2] Andrew Ng – Coursera Machine Learning
[3] https://towardsdatascience.com/unsupervised-machine-learning-clustering-analysis-d40f2b34ae7e