K-means Clustering

Ke Chen

Department of Computer Science, The University of Manchester

Ke.Chen@manchester.ac.uk

Outline

Introduction

Partitioning clustering and history of K-means algorithm

Distance and Similarity Metric

Minkowski distance and cosine similarity/distance metrics

K-means Algorithm

Algorithmic description of K-means clustering

Illustrative Example

Step-by-step K-means clustering demo on synthetic datasets

Relevant Issue

How to partition with K-means, limitation and extension, scatter-based cluster validation

- Partitioning clustering: clustering via iteratively dividing a given dataset into several non-empty and mutually exclusive clusters, which forms a partition of the dataset.
- The number of clusters in dataset, K, is assumed to be known or given in advance.
- A partitioning method would find out an optimal partition, $P^* = \{C_1^*, \cdots, C_K^*\} \in \mathbb{P}_X$, for dataset, X, via minimising sum of squared distance of data items in each cluster to its "representative point" in each cluster:

$$P^* = \operatorname*{argmin}_{P \in \mathbb{P}_X} \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in C_k} d^2(\boldsymbol{x}, \boldsymbol{m}_k), \quad P = \{C_1, \cdots, C_K\},$$

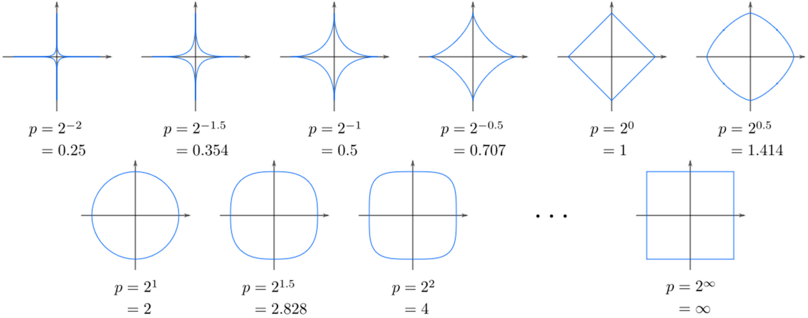
  where $\mathbb{P}_X$ is the set of all possible partitions of K clusters on X, C_k is the kth cluster and $\boldsymbol{m}_k$ is its "representative" point in P and $d(\cdot, \cdot)$ is a distance measure.
- When the "representative" point is set to mean of cluster, it is K-means clustering.

- K-means clustering: finding out a global optimal solution is very hard and computationally expensive in general.
- Hugo Steinhauts (1887-1972) proposed an idea that efficiently find a local optimal solution to the K-means clustering problem in 1956.
- The current version of K-means algorithm carrying out Steinhauts' idea appeared in James MacQueen's paper regarding analysis of multivariate observations published in 1967.
- The K-means algorithm is among the simplest yet the most commonly used clustering algorithms (one of top 10 popular ML algorithms recently voted by ML and data science practitioners).

- **Minkowski distance**
  - For two data points, $\boldsymbol{a} = [a_1 \ a_2 \ \cdots \ a_n]^T \in \mathbb{R}^n$ and $\boldsymbol{b} = [b_1 \ b_2 \ \cdots \ b_n]^T \in \mathbb{R}^n$, Minknowski distance (family) for metric data is defined as follows:

$$d(\boldsymbol{a}, \boldsymbol{b}) = \Big(\sum_{i=1}^{n} |a_i - b_i|^p\Big)^{\frac{1}{p}} = \Big(|a_1 - b_1|^p + |a_2 - b_2|^p + \cdots + |a_n - b_n|^p\Big)^{\frac{1}{p}}.$$

  - Manhattan (city block) distance ($p = 1$): $d(\boldsymbol{a}, \boldsymbol{b}) = \sum_{i=1}^{n} |a_i - b_i|$.
  - Euclidean distance ($p = 2$): $d(\boldsymbol{a}, \boldsymbol{b}) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$.



- **Cosine similarity**
  - For two data points, $\boldsymbol{a} = [a_1 \ a_2 \ \cdots \ a_n]^T \in \mathbb{R}^n$ and $\boldsymbol{b} = [b_1 \ b_2 \ \cdots \ b_n]^T \in \mathbb{R}^n$, Cosine similarity for non-metric data is defined as follows:

$$s(\boldsymbol{a}, \boldsymbol{b}) = \cos(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a}^T \boldsymbol{b}}{||\boldsymbol{a}||||\boldsymbol{b}||} = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2}\sqrt{\sum_{i=1}^n b_i^2}}.$$

  - $\boxed{\text{Property}}$: $-1 \leq s(\boldsymbol{a}, \boldsymbol{b}) \leq 1$.
- **Cosine distance**
  - A similarity can be converted into the corresponding distance and vice versa.
  - Cosine distance for non-metric data is defined as follows:

$$d(\boldsymbol{a}, \boldsymbol{b}) = 1 - \cos(\boldsymbol{a}, \boldsymbol{b}) = 1 - \frac{\boldsymbol{a}^T \boldsymbol{b}}{||\boldsymbol{a}||||\boldsymbol{b}||} = 1 - \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2}\sqrt{\sum_{i=1}^n b_i^2}}.$$

  - $\boxed{\text{Property}}$: $0 \leq d(\boldsymbol{a}, \boldsymbol{b}) \leq 2$.
- Nonmetric data: frequency of words in documents, genes in micro-arrays, $\cdots$

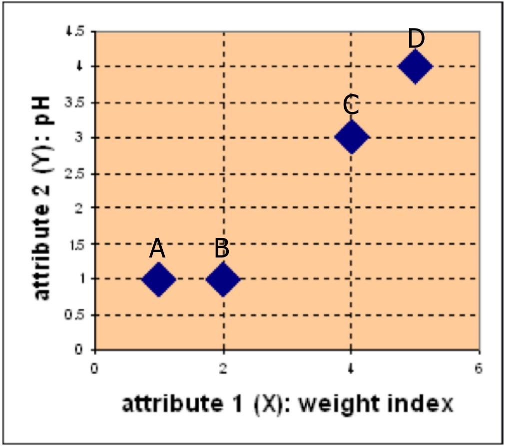
Input: Data set, X, number of clusters, K, and an appropriate distance (similarity) measure reflecting the nature of data in X

- **Initialisation**: randomly choose K points as cluster centres (means)
- **Step 1**: calculate distances (similarities) between all the points in X and K cluster centres
- **Step 2**: find out the closest cluster centre for each data point in X and assign the data point to this cluster
- **Step 3**: update its cluster centre for every cluster changed in the last step by averaging all the new member points in this cluster
- **Step 4**: output K clusters if memberships in all K clusters do not change. Otherwise, go to **Step 1**.

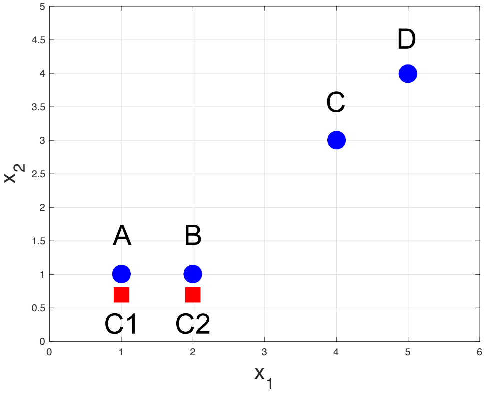
Fact: K-means algorithm always converges (i.e., memberships of all K clusters no longer change) in a finite number of iterations but could end up with an unwanted partition.

- **Dataset 1**: Medicine clustering analysis ($K = 2$)

Medicine	Weight	pH-Index
A	1	1
B	2	1
C	4	3
D	5	4



- **Dataset 1**: Medicine clustering analysis ($K = 2$)



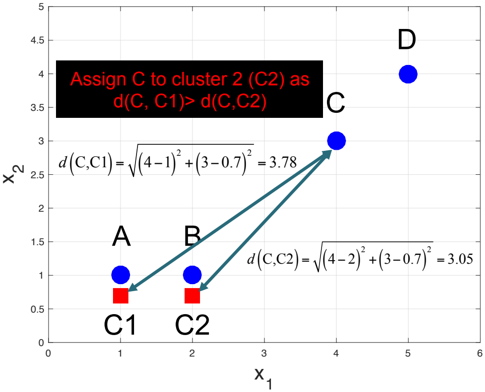
Determine in advance:

- Group to K=2 clusters.
- Use Euclidean distance to measure the dissimilarity between data points.
- Set initial cluster centers. For instance,

  C1: (1, 0.7)

  C2: (2, 0.7)

- **Dataset 1**: Medicine clustering analysis ($K = 2$)



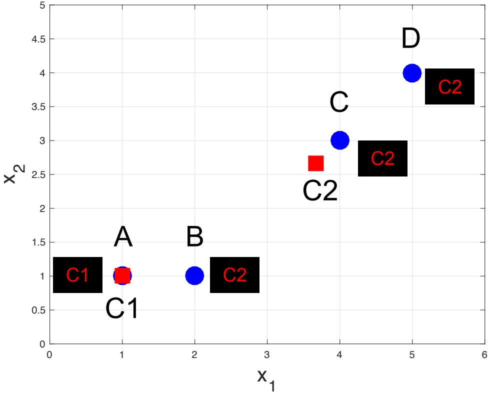
A: (1, 1)
B: (2, 1)
C: (4, 3)
D: (5, 4)

Determine in advance:

- Group to K=2 clusters.
- Use Euclidean distance to measure the (dis)similarity between data points.
- Set initial cluster centers. For instance,
  C1: (1, 0.7)
  C2: (2, 0.7)

- Step 1: Calculate distances (or similarities) between the data points and the cluster center points.
- Step 2: Find the nearest cluster center to each data point, and assign the data point to that cluster.

- **Dataset 1**: Medicine clustering analysis ($K = 2$)



Determine in advance:

- Group to K=2 clusters.
- Use Euclidean distance to measure the (dis)similarity between data points.
- Set initial cluster centers. For instance,

  C1: (1, 0.7)

  C2: (2, 0.7)

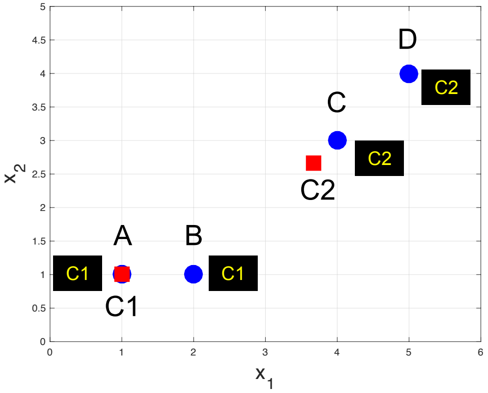
- Step 3: Calculate the new cluster center for each cluster, by **averaging** its member points.

A: (1, 1)
B: (2, 1)
C: (4, 3)
D: (5, 4)

$$\mathrm{C1} = \mathrm{A} = (1,1)$$

$$\mathrm{C2} = \frac{\mathrm{B}+\mathrm{C}+\mathrm{D}}{3} = \frac{(2,1)+(4,3)+(5,4)}{3} = \left(\frac{2+4+5}{3}, \frac{1+3+4}{3}\right) = (3.67, 2.67)$$

- **Dataset 1**: Medicine clustering analysis ($K = 2$)

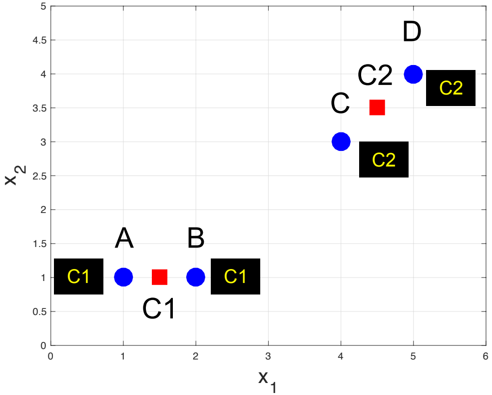


Determine in advance:

- Group to K=2 clusters.
- Use Euclidean distance to measure the (dis)similarity between data points.
- Set initial cluster centers. For instance,
  C1: (1, 0.7)
  C2: (2, 0.7)

- Repeat Steps1-2 to update cluster membership.

- **Dataset 1**: Medicine clustering analysis ($K = 2$)



Determine in advance:

- Group to K=2 clusters.
- Use Euclidean distance to measure the (dis)similarity between data points.
- Set initial cluster centers. For instance,

  C1: (1, 0.7)

  C2: (2, 0.7)

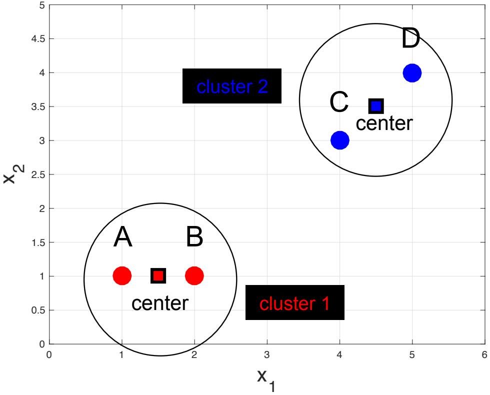
- Repeat Step 3 to update cluster center.

A: (1, 1)
B: (2, 1)
C: (4, 3)
D: (5, 4)

$$C1 = \frac{A+B}{2} = \frac{(1,1)+(2,1)}{2} = (1, 1.5)$$

$$C2 = \frac{C+D}{2} = \frac{(4,3)+(5,4)}{2} = (4.5, 3.5)$$

- **Dataset 1**: Medicine clustering analysis ($K = 2$)

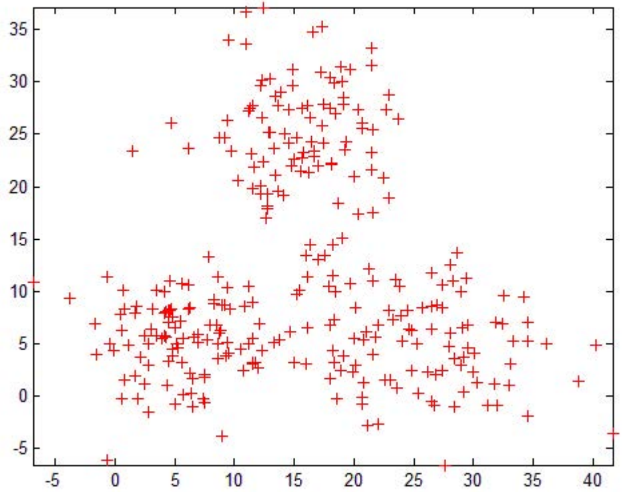


Determine in advance:

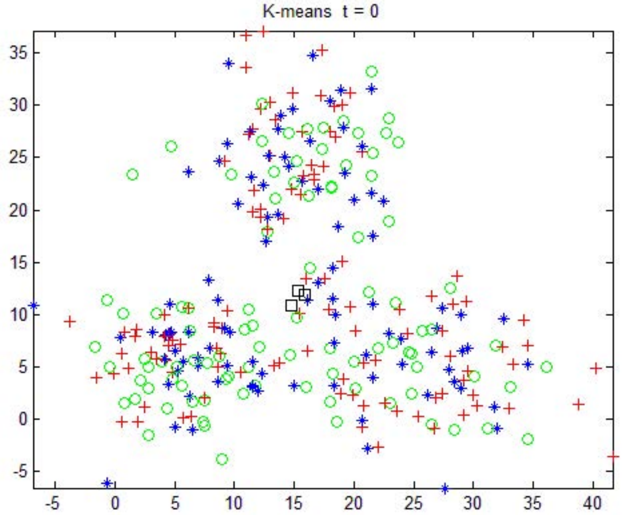
- Group to K=2 clusters.
- Use Euclidean distance to measure the (dis)similarity between data points.
- Set initial cluster centers. For instance,
  C1: (1, 0.7)
  C2: (2, 0.7)

- Stop repeating when there is no change in the membership of each cluster.

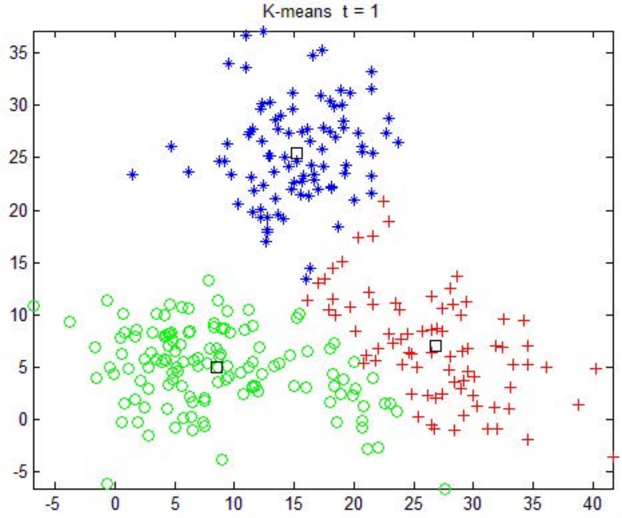
- **Dataset 2**: Synthetic data ($K = 3$), Euclidean Distance



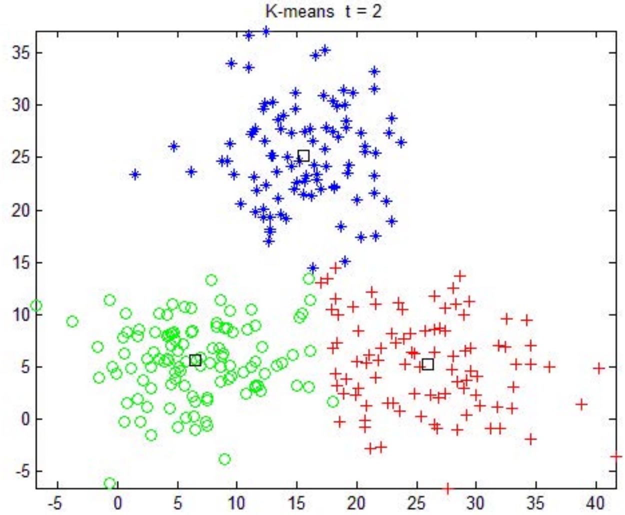
- **Dataset 2**: Synthetic data ($K = 3$), Euclidean Distance



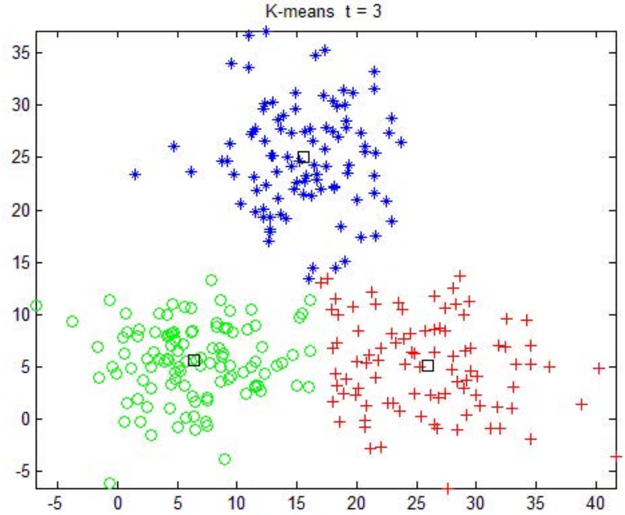
- **Dataset 2**: Synthetic data ($K = 3$), Euclidean Distance



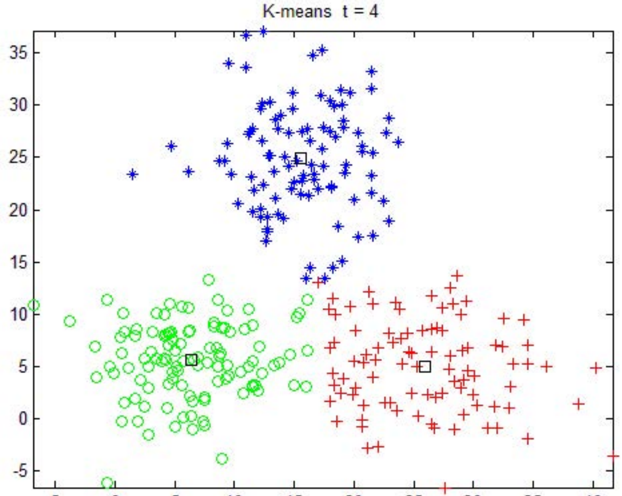
- **Dataset 2**: Synthetic data ($K = 3$), Euclidean Distance



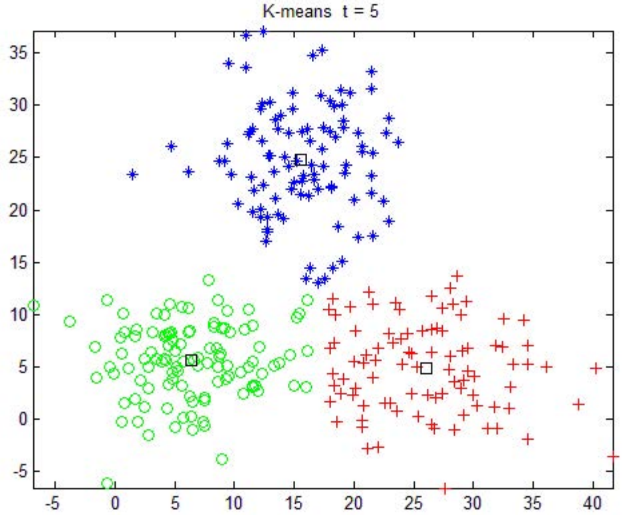
- **Dataset 2**: Synthetic data ($K = 3$), Euclidean Distance



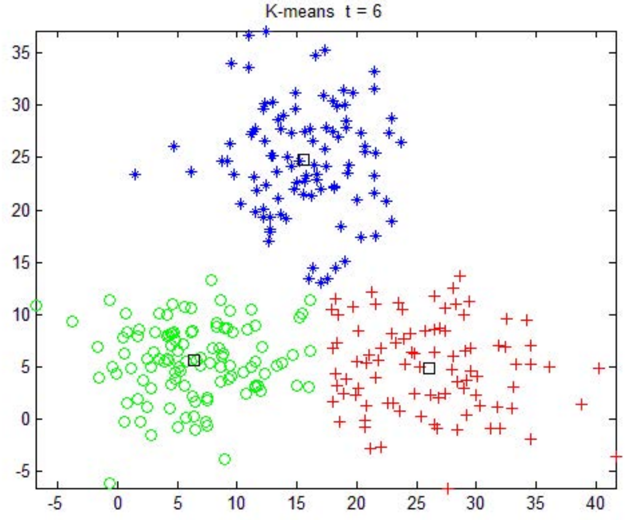
- **Dataset 2**: Synthetic data ($K = 3$), Euclidean Distance



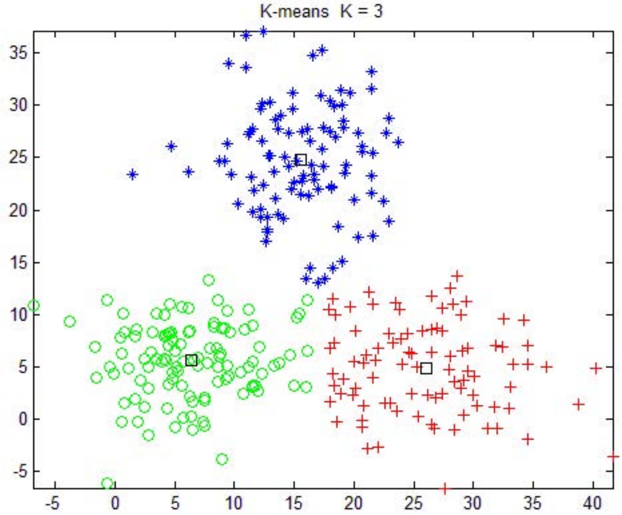
- **Dataset 2**: Synthetic data ($K = 3$), Euclidean Distance



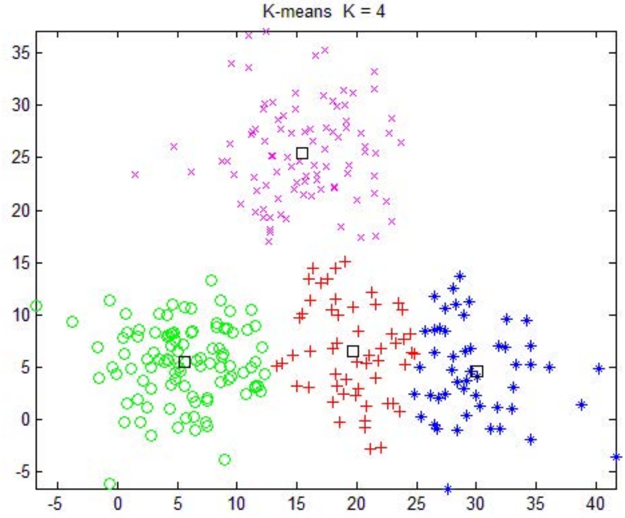
- **Dataset 2**: Synthetic data ($K = 3$), Euclidean Distance



- **Dataset 2**: Synthetic data ($K = 3$), Euclidean Distance

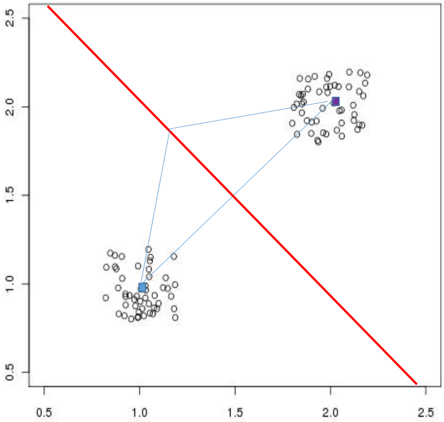


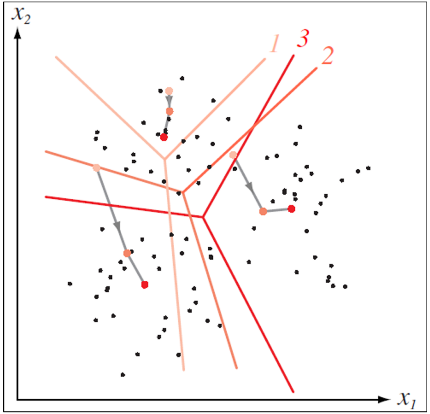
- **Dataset 2**: Synthetic data ($K = 4$), Euclidean Distance



How K-means partition the data space?

- Once K cluster centres are set, they divide the entire data space into K mutually exclusive regions (clusters) to form a partition collectively.
- Boundary between two clusters passes the mid-points between their cluster centres.

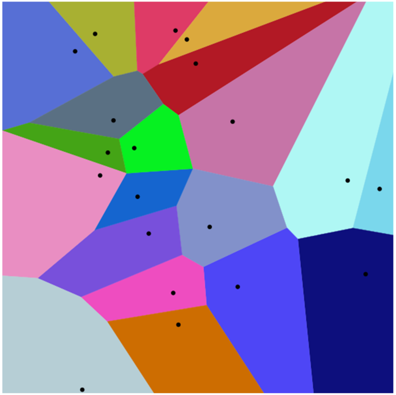




How K-means partition the data space?

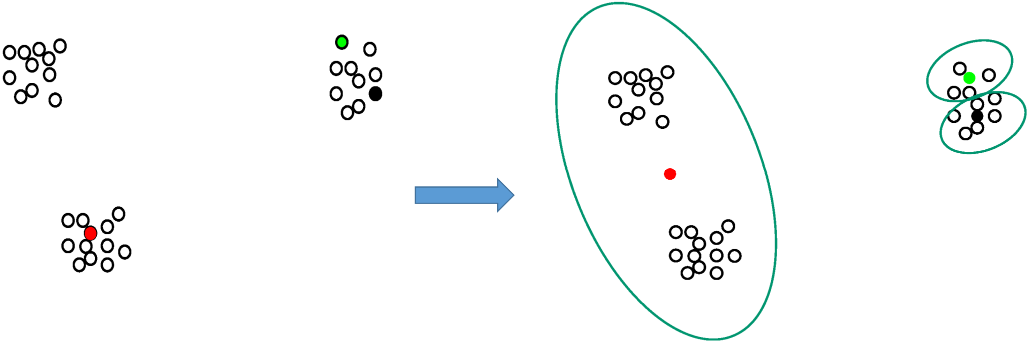
- Once K cluster centres are set, they divide the entire data space into K mutually exclusive regions (clusters) to form a partition collectively.
- Partition is a distance-dependent Voronoi diagram (named after Georgy Voronoy).

Manhattan Distance

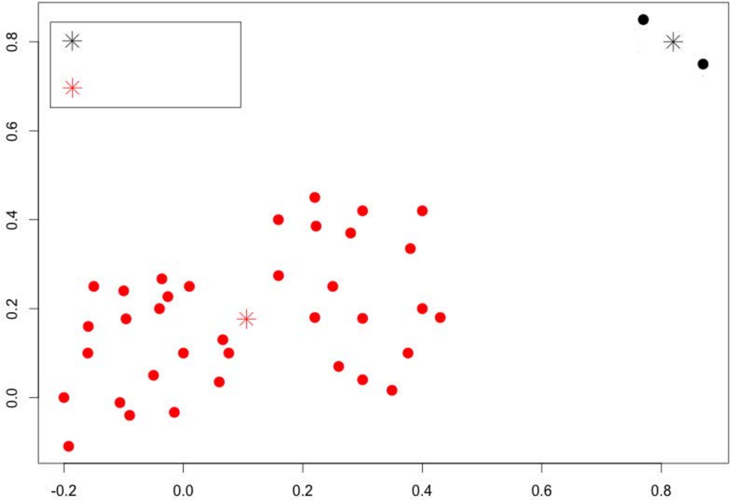


Euclidean Distance

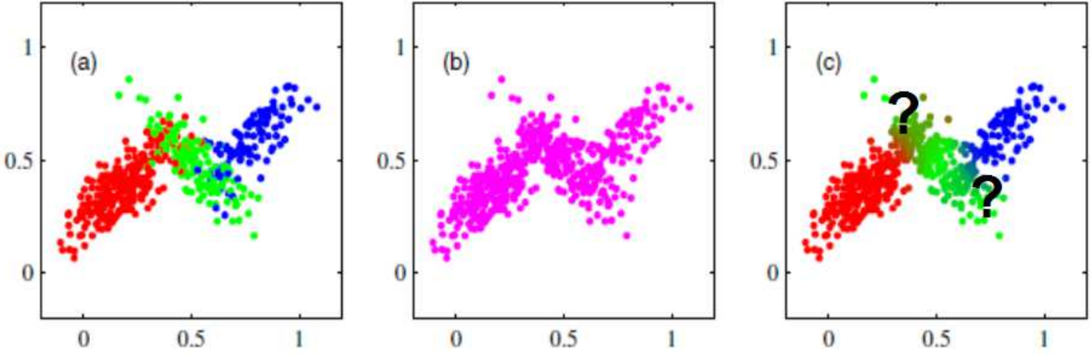
- Limitation: sensitive to initial cluster centres
- Extension: K-medoids, K-means++, $\cdots$



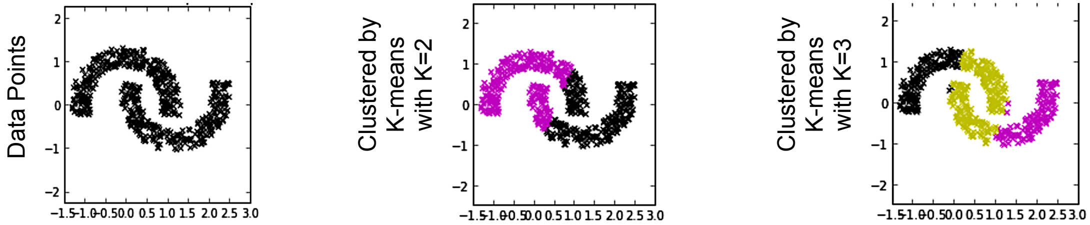
- Limitation: sensitive to outliers and noisy data
- Extension: K-median, K-means++, $\cdots$



- Limitation: unable to deal with "overlapping" clusters properly
- Extension: Probabilistic generative model, e.g., GMM, $\cdots$



- Limitation: unable to discover non-convex clusters underlying data
- Extension: spectral clustering, density-based clustering, $\cdots$

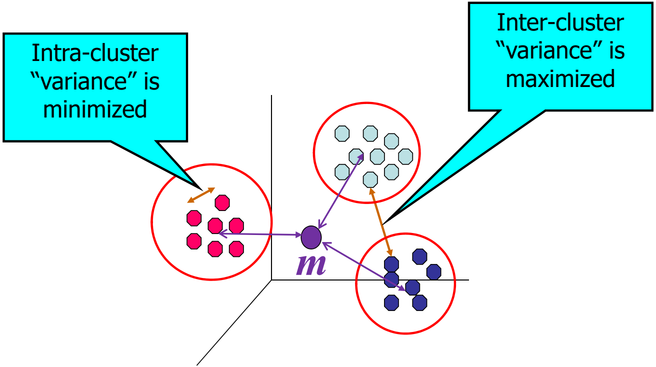


Scatter-based Cluster Validation

- Motivation: evaluate clustering quality and help finding clusters K if unknown
- Within-cluster-scatter (SSW) versus Between-cluster-scatter (SSB)

$$\mathrm{SSW}(K) = \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in C_k} d^2(\boldsymbol{x}, \boldsymbol{m}_k), \quad \mathrm{SSB}(K) = \sum_{k=1}^{K} |C_k| d^2(\boldsymbol{m}, \boldsymbol{m}_k)$$

where $|C_k|$ is number of data points in C_k and $\boldsymbol{m}$ is global mean of entire dataset.



Scatter-based Cluster Validation

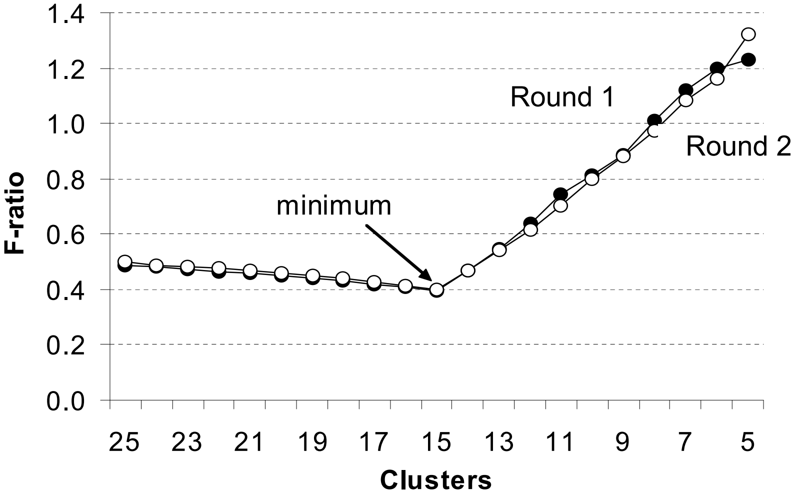
- F-ratio (W-B) index: measure ratio of the within-cluster-scatter (SSW) against the between-cluster-scatter (SSB)
- For a partition of K ($K > 1$) clusters on dataset X, F-ratio index is defined by

$$\mathrm{F}(K) = \frac{K * \mathrm{SSW}(K)}{\mathrm{SSB}(K)} = \frac{K \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in C_k} d^2(\boldsymbol{x}, \boldsymbol{m}_k)}{\sum_{k=1}^{K} |C_k| d^2(\boldsymbol{m}, \boldsymbol{m}_k)}$$

  where the mean of cluster k is $\boldsymbol{m}_k = \frac{1}{|C_k|} \sum_{\boldsymbol{x} \in C_k} \boldsymbol{x}$ and the global mean of entire dataset X is $\boldsymbol{m} = \frac{1}{|X|} \sum_{\boldsymbol{x} \in X} \boldsymbol{x}$. $d(\cdot, \cdot)$ is distance measure.
- $\boxed{\text{Property}}$: promoting a partition of compactness, being well-separated, small number of clusters (K) and large cluster size ($|C_k|$); i.e., the smaller F-ratio index, the better clustering quality

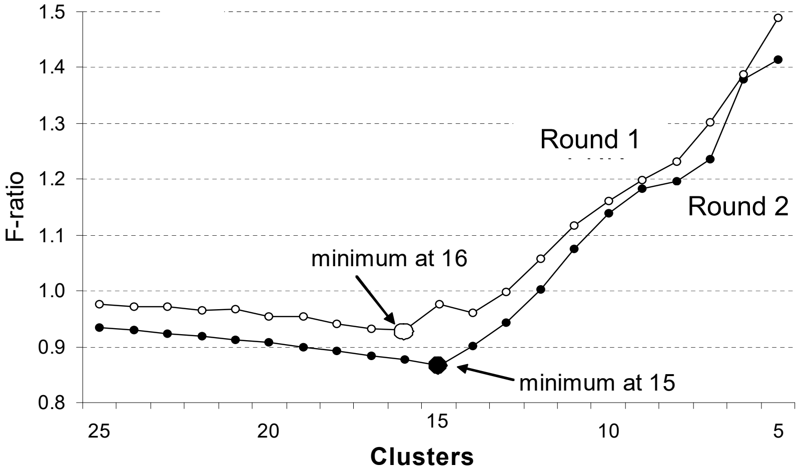
Scatter-based Cluster Validation

- Example 1: find out an optimal number of clusters with F-ratio index



Scatter-based Cluster Validation

- Example 2: find out an optimal number of clusters with F-ratio index



Reference

If you want to deepen your understanding and learn something beyond this lecture, you can self-study the optional references below.

[Alpaydin, 2014] Alpaydin E. (2014): *Introduction to Machine Learning* (3rd Ed.), MIT Press. (Sections 7.1-7.3 & 7.9)

[Goodfellow et al., 2016] Goodfellow I., Bengio Y., and Courville A. (2016): *Deep Learning*, MIT Press. (Section 5.8.2)

[Barber, 2012] Barber D. (2012): *Bayesian Reasoning and Machine Learning*, Cambridge University Press. (Sections 20.3)

[Jain et al., 1999] Jain A.K., Murty M.N. and Flynn P.J. (1999): Data clustering: A review. *ACM Computing Survey*, Vol. 31, No. 3, pp. 264-323.