Fakultas Informatika
School of Computing
Telkom University

Machine Learning

Unsupervised Learning - Clustering

Adopted from ADF Slides

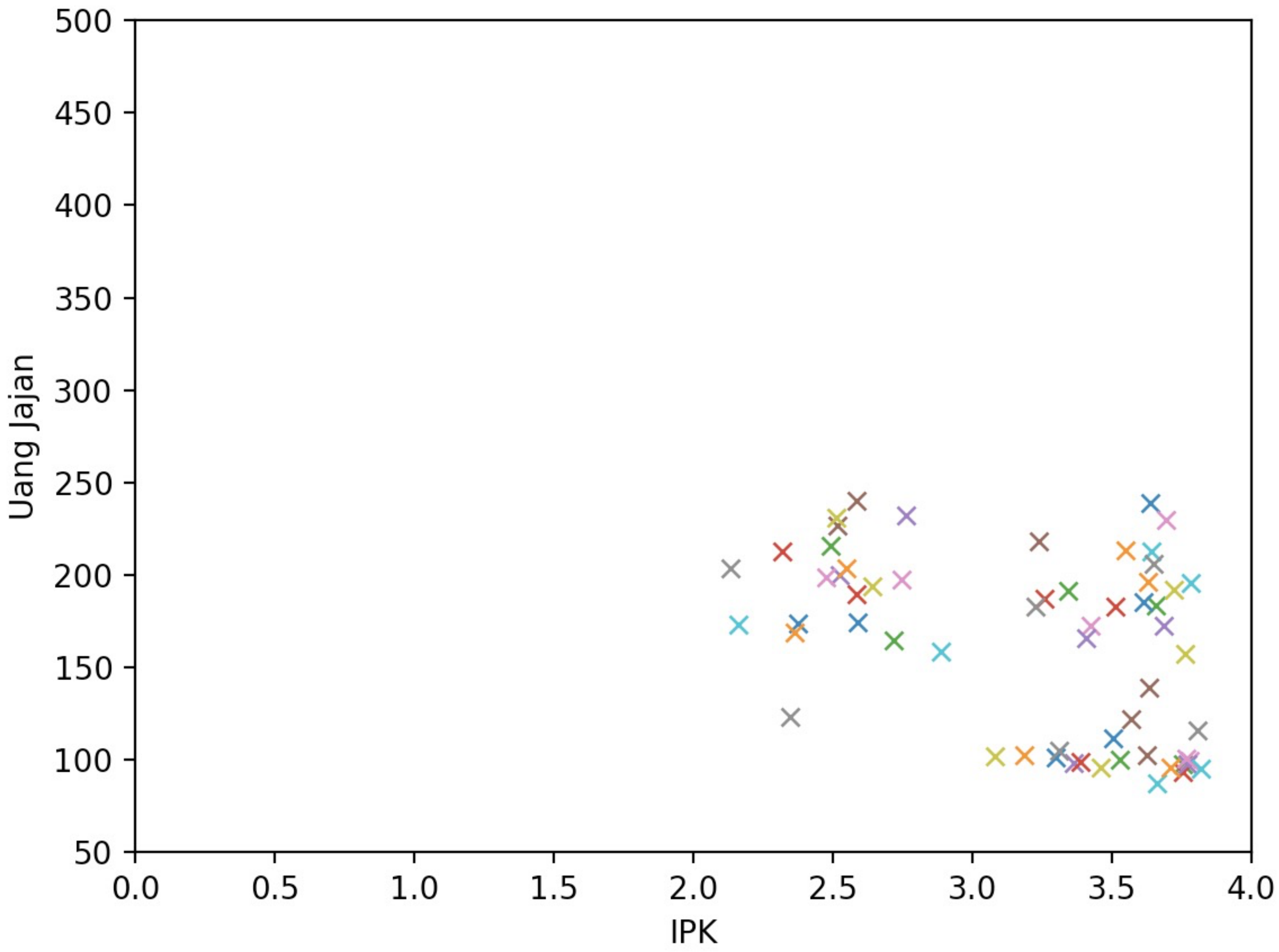
Outline

- Introduction:
  - Supervised vs Unsupervised
  - Clustering
- Partitional Clustering: K-Means
- Hierarchical Clustering: Agglomerative

Case

- Sebuah perusahaan sedang melakukan riset mengenai tipe mahasiswa yang ada di Tel-U.
- Aspek yang dilihat misalnya :
  - Uang Jajan
  - IPK
- Sebuah perusahaan ingin memasarkan produknya, namun sebelumnya ingin mengetahui segmen pasar yang ada di Bandung agar bisa tepat sasaran.
- Perusahaan Telekomunikasi ingin memasang BTS namun perlu mencari posisi yang optimal agar jumlah BTS sedikit, namun bisa mengcover sebagian besar pengguna

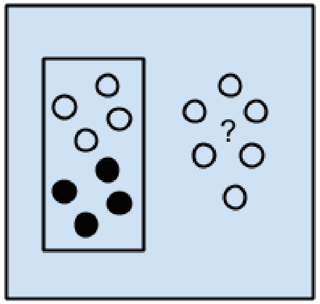
- Perusahaan celana perlu menentukan jenis ukuran celana (panjang dan lebar) agar bisa mengcover banyak pembeli namun juga tidak harus membuat terlalu banyak jenis ukuran.


500
450
400
350
300
250
200
150
100
50
Uang Jajan
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
IPK

Motivation

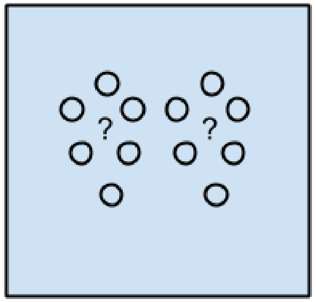
Supervised vs Unsupervised Learning

- Supervised
  - You have labeled data
  - Find a function which can map data to its label
  - discover patterns that relate data attributes with a target (class)



Supervised Learning Algorithms

- Unsupervised
  - You have unlabeled data
  - Discover the underlying structure of the data
  - Try to understand the data
  - Not predicting anything specific



Unsupervised Learning Algorithms

Clustering

- The process of grouping a set of objects into classes of similar object
  - Data within a cluster should be similar (or related) .
  - Data from different clusters should be dissimilar (or unrelated).

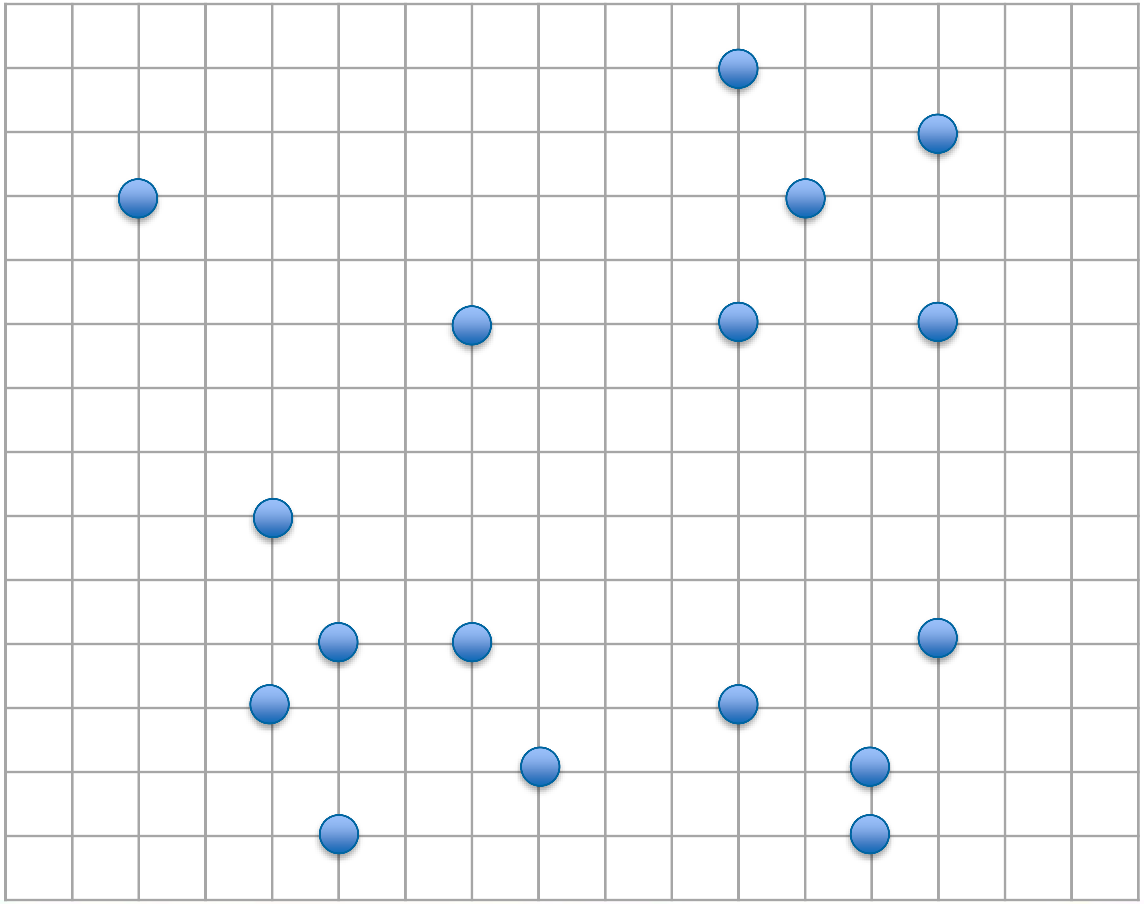
- Cluster: A collection/group of data objects/points

- Cluster analysis
  - find similarities between data according to characteristics underlying the data and grouping similar data objects into clusters

Clustering

- Try to answer:
  - How many sub-populations (groups) ?
  - What are their sizes?
  - Do elements in a sub-population have any common properties?
  - Are sub-populations cohesive? Can they be further split up?
  - Are there outliers?

Clustering



How many cluster?

What is a good cluster?

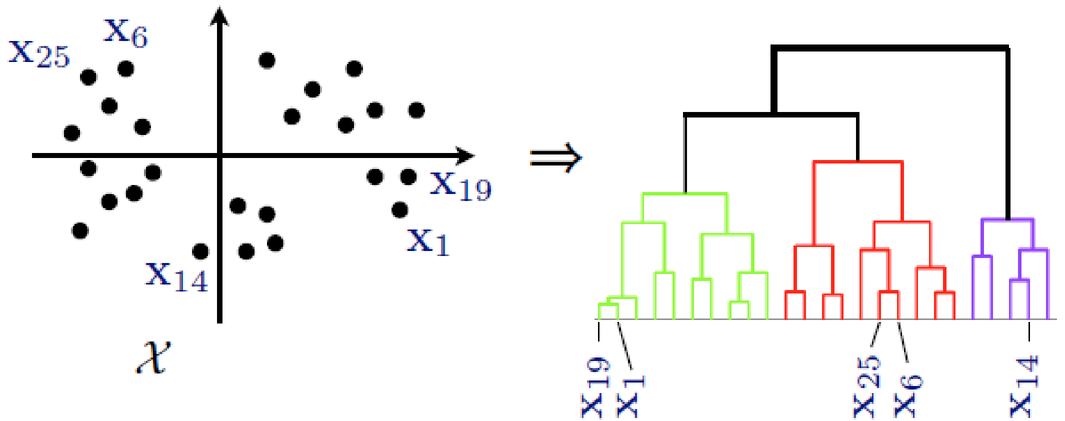
Define how to cluster

Types of Clustering

- Hierarchical (Connectivity-based)
  - Objects being more related to nearby objects than to objects farther away
- Partitional (Centroid-based)
  - Each cluster represented by a centroid
  - Determined by a proximity measurement
- Other types:
  - Distribution-based
  - Density-based
  - Etc.

Hierarchical (Connectivity-based)

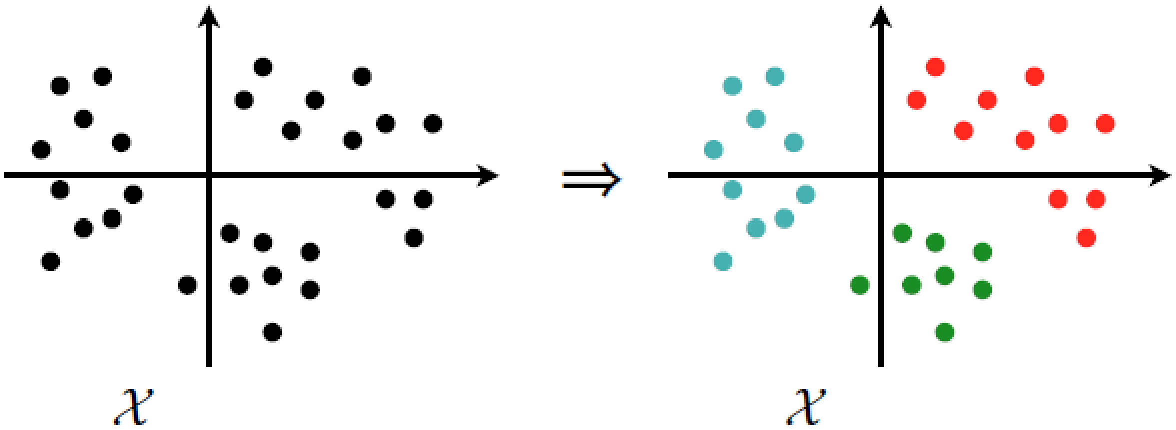
- Objects being more related to nearby objects than to objects farther away



  - In this approach, data vectors are arranged in a tree, where nearby ('similar') vectors x_i and x_j are placed close to each other in the tree
    - Any horizontal cut corresponds to a partitional clustering
    - In the example above, the 3 colors have been added manually for emphasis (they are not produced by the algorithm)

Partitional (Centroid-based)

- Each cluster represented by a centroid determined by a proximity measurement



  - Typically K and a proximity measure is selected by the user, while the chosen algorithm then learns the actual partitions

Types of Clustering

- Grouping criteria
  - Monothetic (using some common attribute)
  - Polythetic (using similarity/distance measure)
- Overlap criteria
  - Hard clustering
  - Soft clustering

Clustering Algorithms

- K-Means
  - Polythetic, Partitional, Hard Clustering
- K-D Trees
  - Monothetic, Hierarchical, Hard Clustering
- EM clustering
  - Polythetic, Partitional, Soft Clustering
- Fuzzy C-Means
  - Polythetic, Partitional, Soft Clustering
- Self-Organizing Map
- Quality Threshold
- Agglomerative
- Etc.

Usage

- Discover classes of unlabeled data
- Dimensionality reductions
- Graph coloring
- Color-based image segmentation
- Social network analysis
- Market segmentation
- Etc.

K-Means Algorithm

K-Means Algorithm

- A simple and often used partitional clustering method
- Hard, polythetic clustering
- Data partitioned into K cluster
  - Need to determine K
- Points in each cluster similar to a "centroid"

K-Means Algorithm

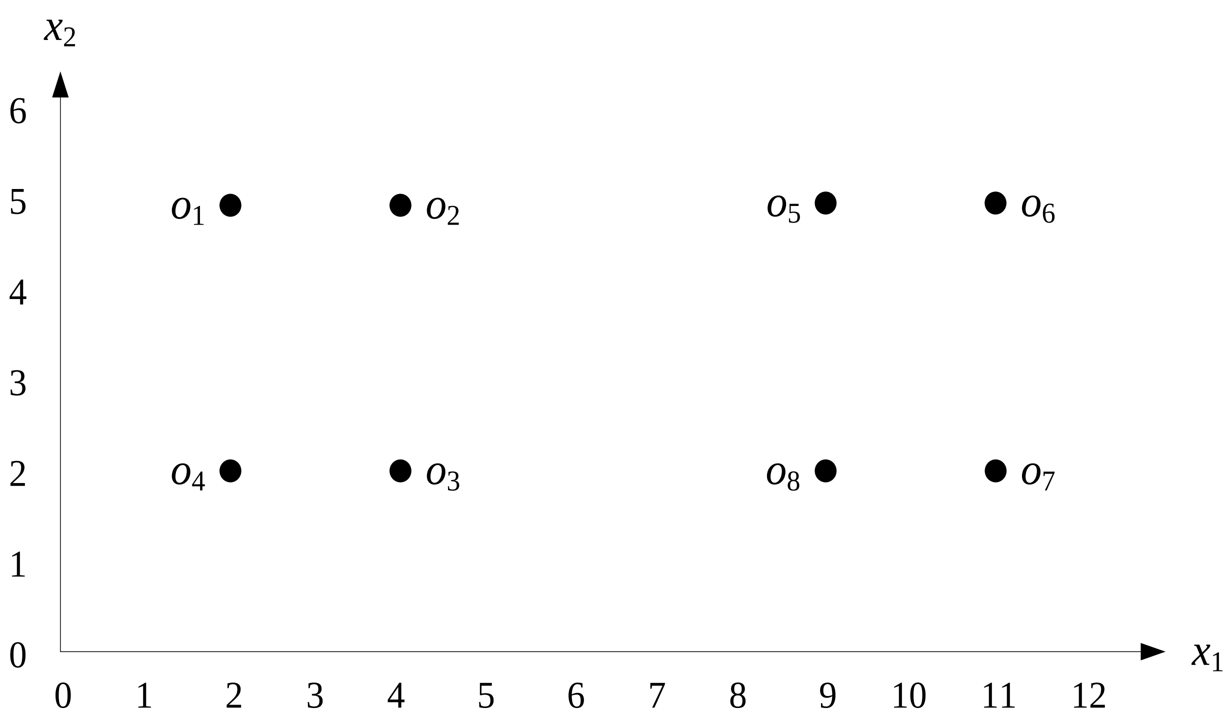
Algorithm 8.1 Basic K-means algorithm.

```
1: Select K points as initial centroids.
2: repeat
3:    Form K clusters by assigning each point to its closest centroid.
4:    Recompute the centroid of each cluster.
5: until Centroids do not change.
```

- Line 1: simplest solution is to initialize the c_j to equal K random vectors from the input data
- Line 3: For simplicity, use Euclidean
- Line 4: recalculate using $$\mathbf{c}_j = \frac{1}{|C_j|} \sum_{i \in C_j} \mathbf{x}_i$$

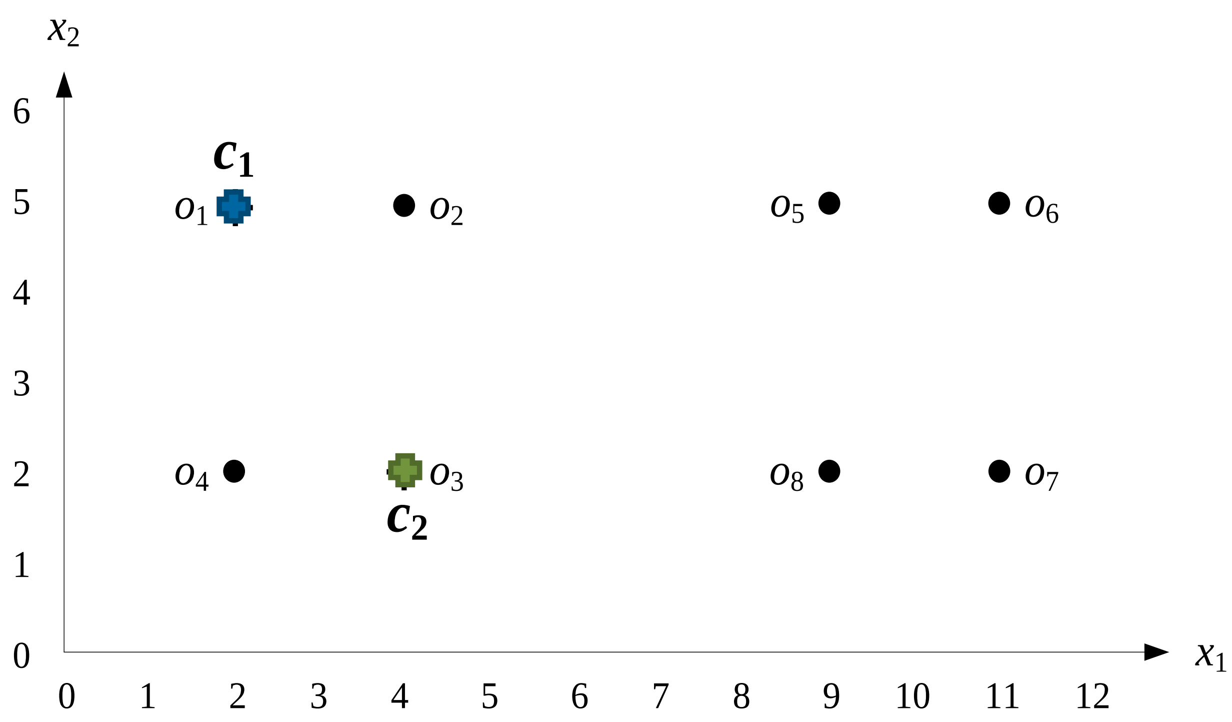
K-Means Example

K-Means - Example



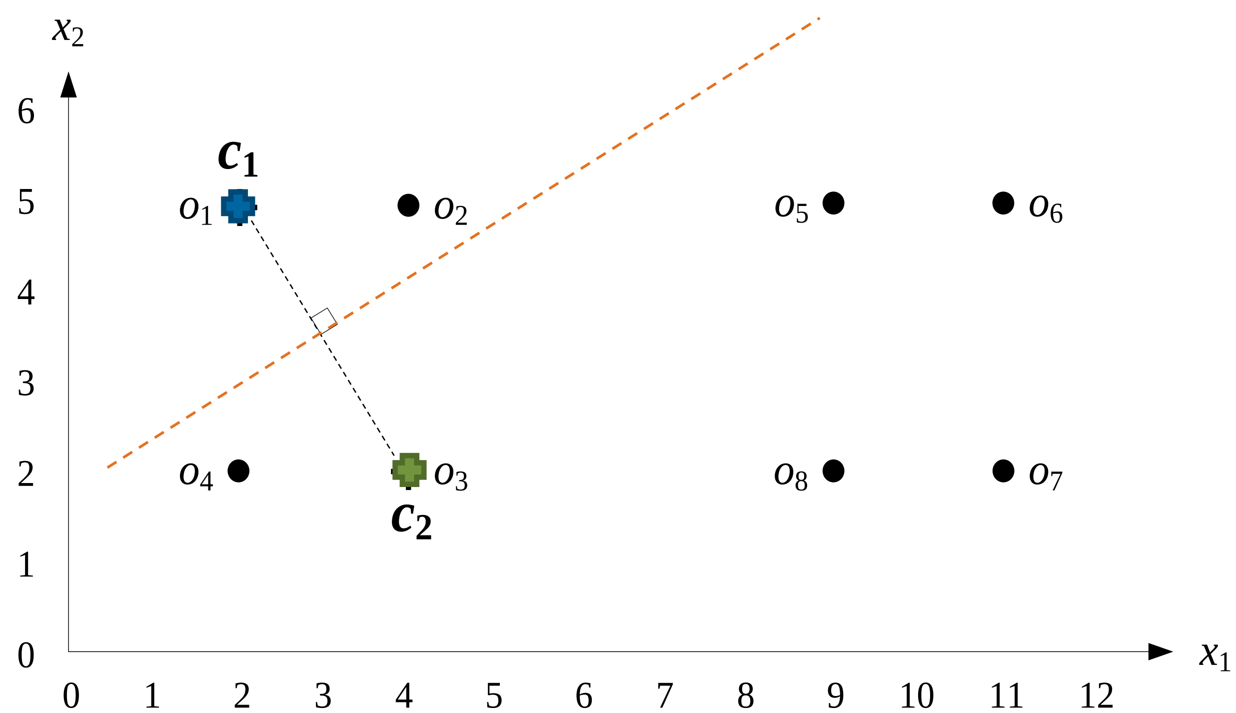
Set K = 2,
initialize 2 centroid randomly

K-Means - Example



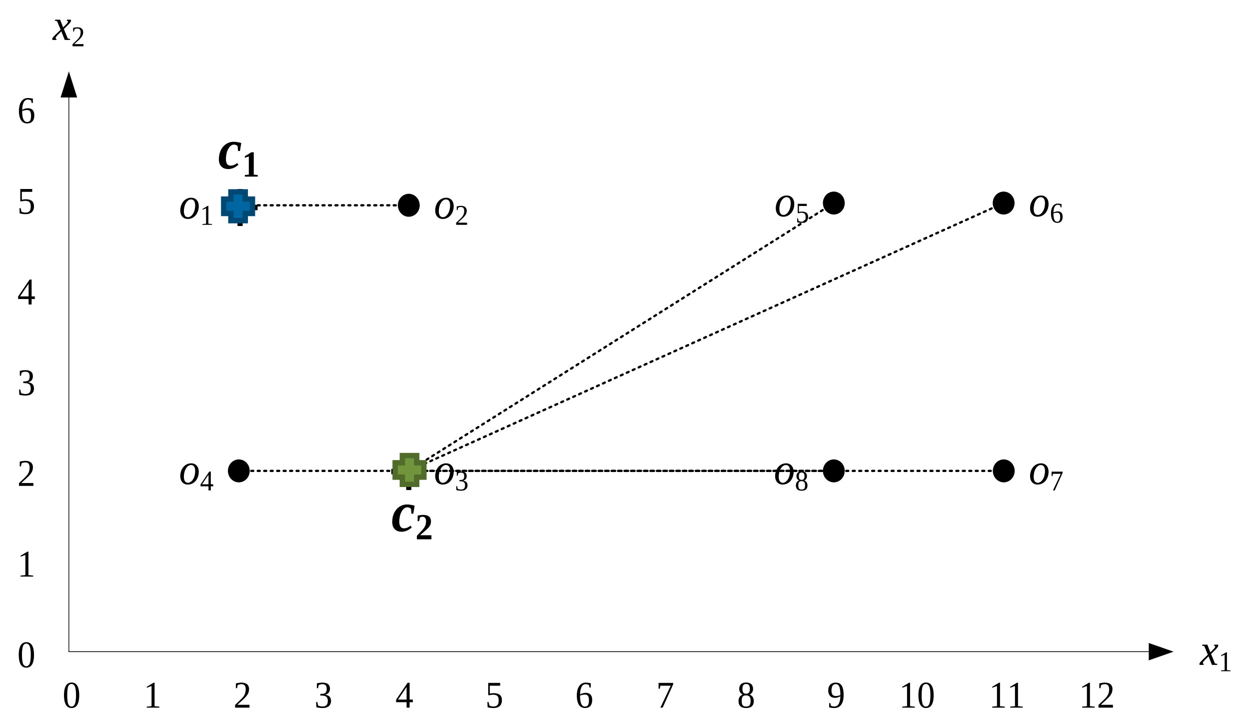
For each data point o, find nearest centroid c

K-Means - Example



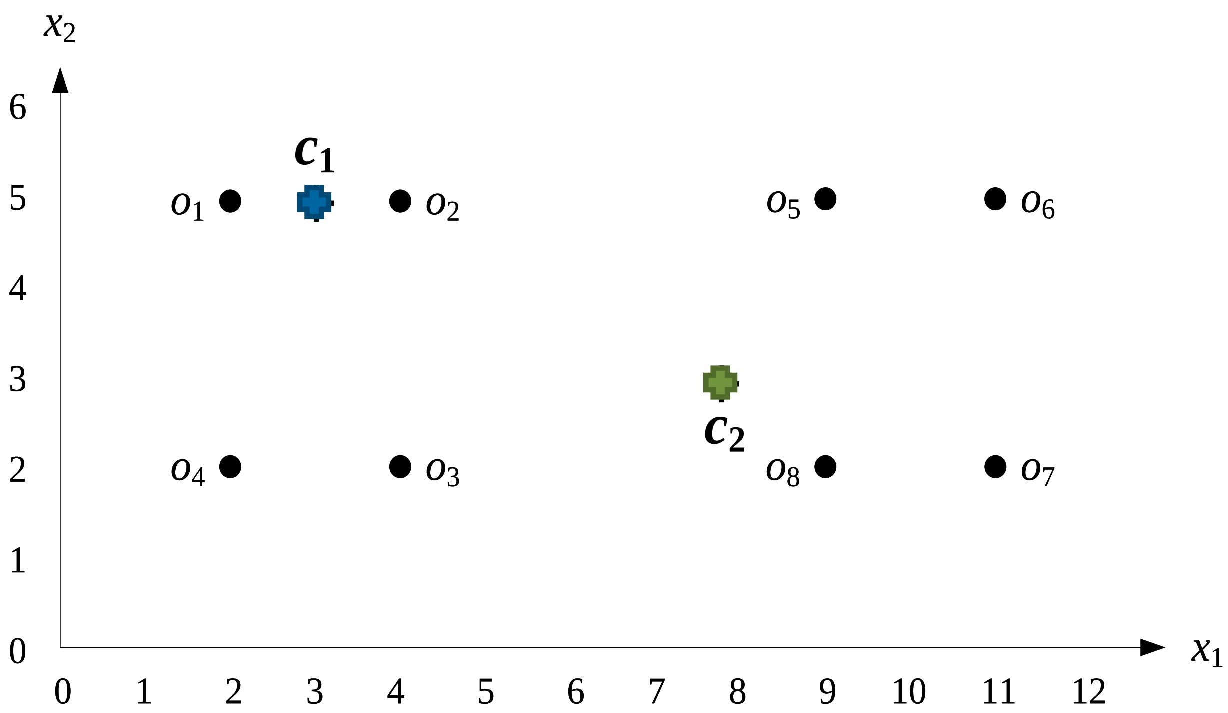
For each data point o, find nearest centroid c

K-Means - Example



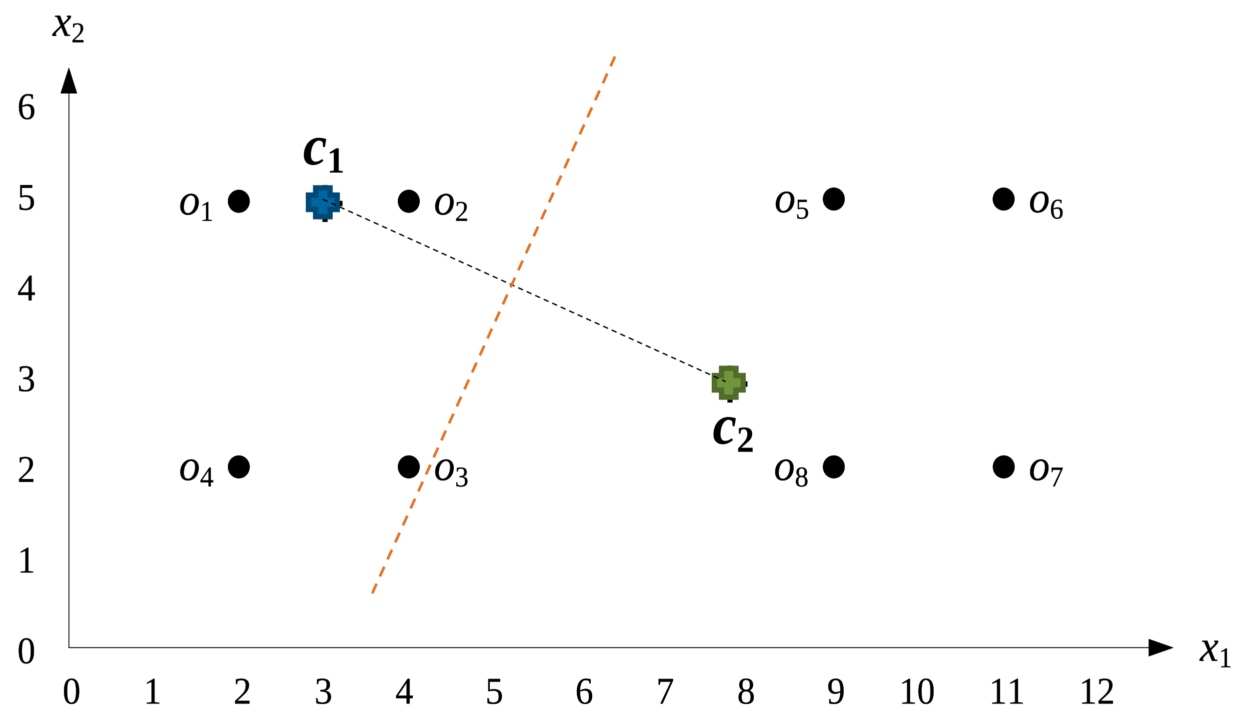
Calculate mean data of each cluster,
Update cluster

K-Means - Example



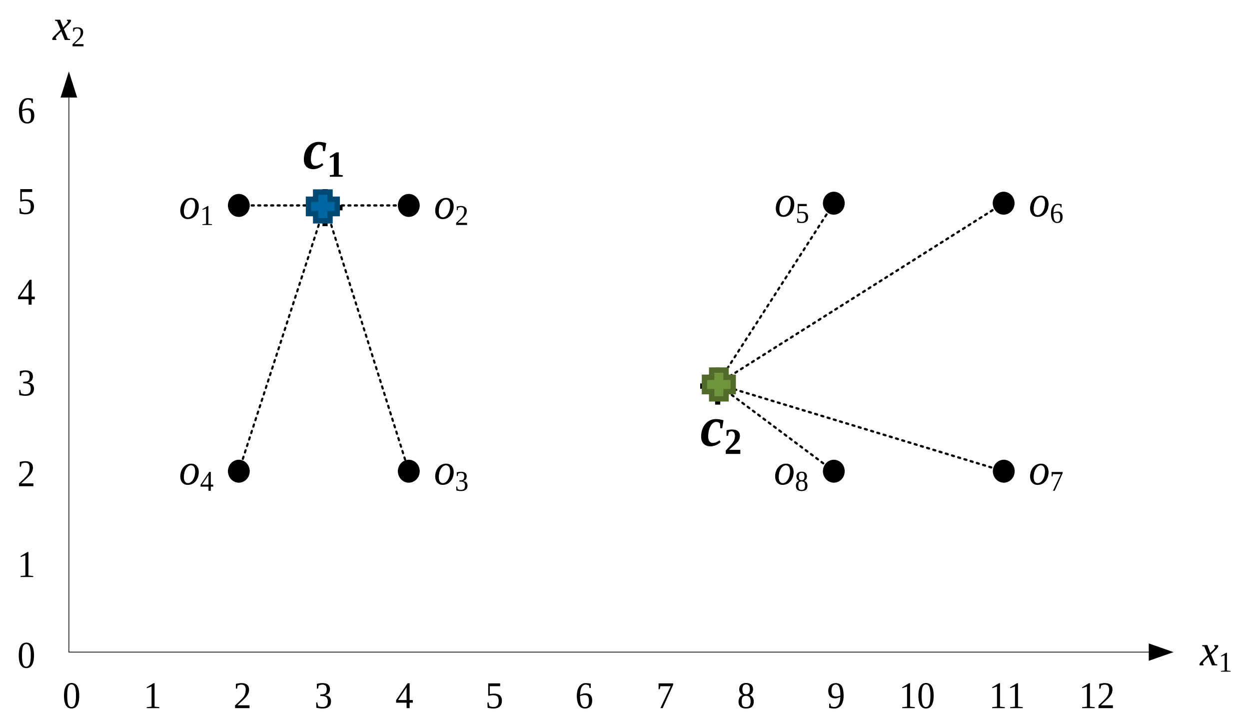
Iteration 2, new centroid
For each data point o, find nearest centroid c

K-Means - Example



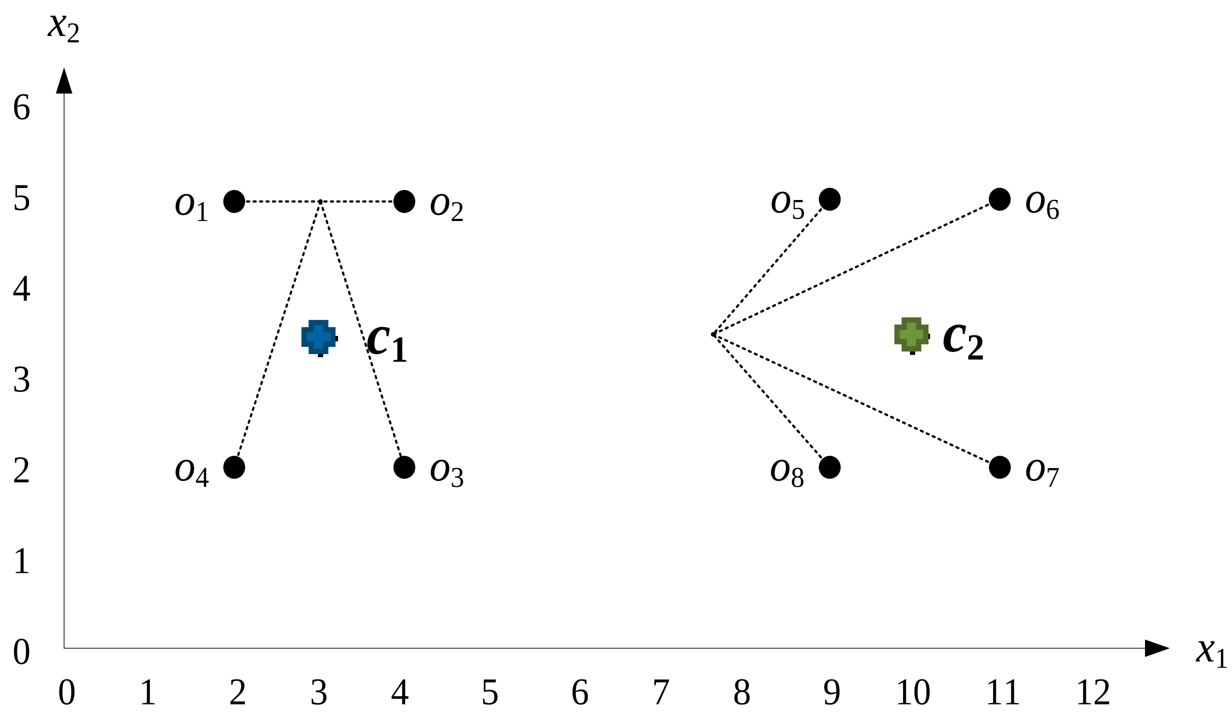
Iteration 2, new centroid
For each data point o, find nearest centroid c

K-Means - Example



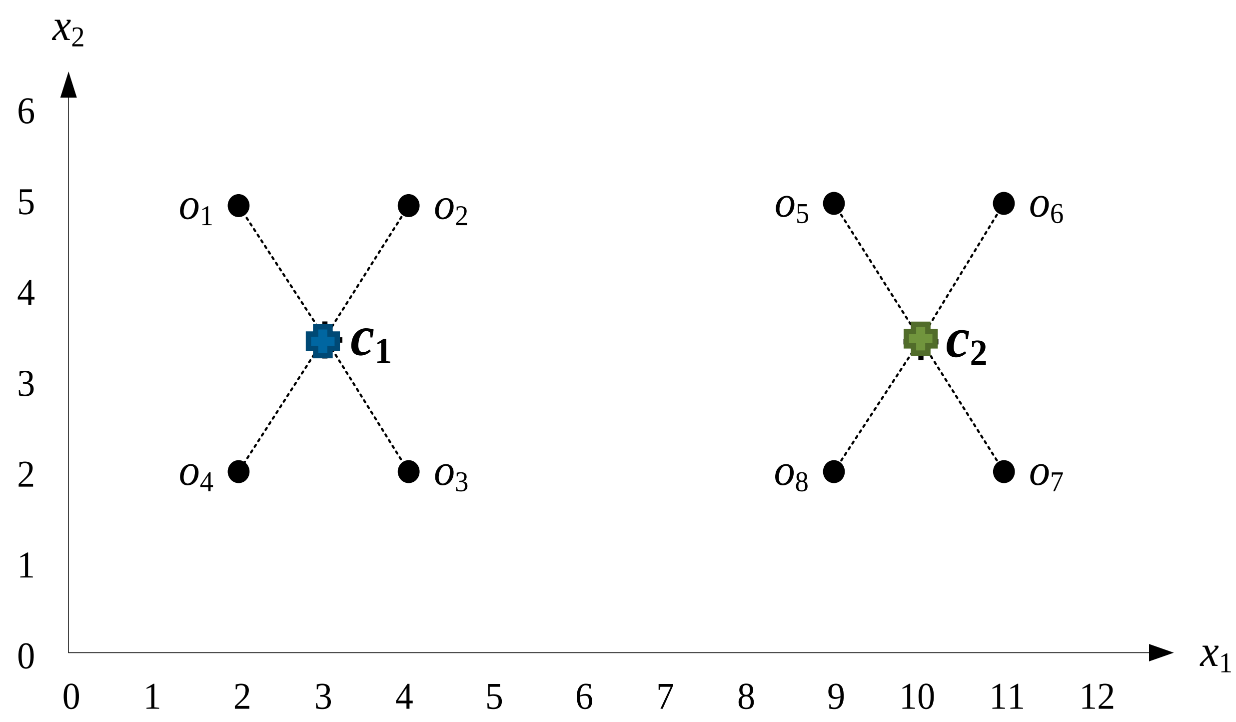
Calculate mean data of each cluster,
Update cluster

K-Means - Example



Iteration 3, new centroid
For each data point o, find nearest centroid c

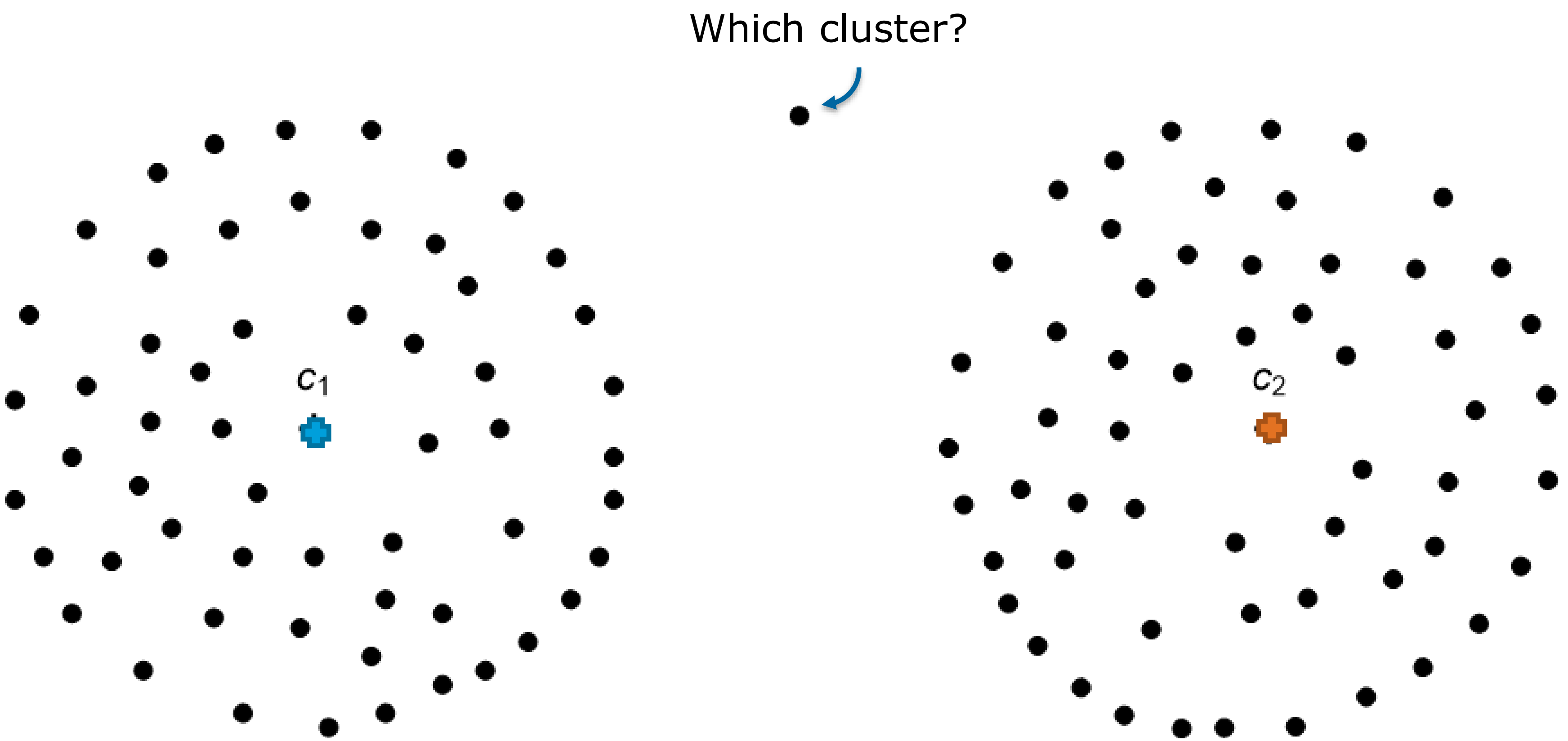
K-Means - Example



Calculate mean data of each cluster,
Cluster not updated, iteration stop

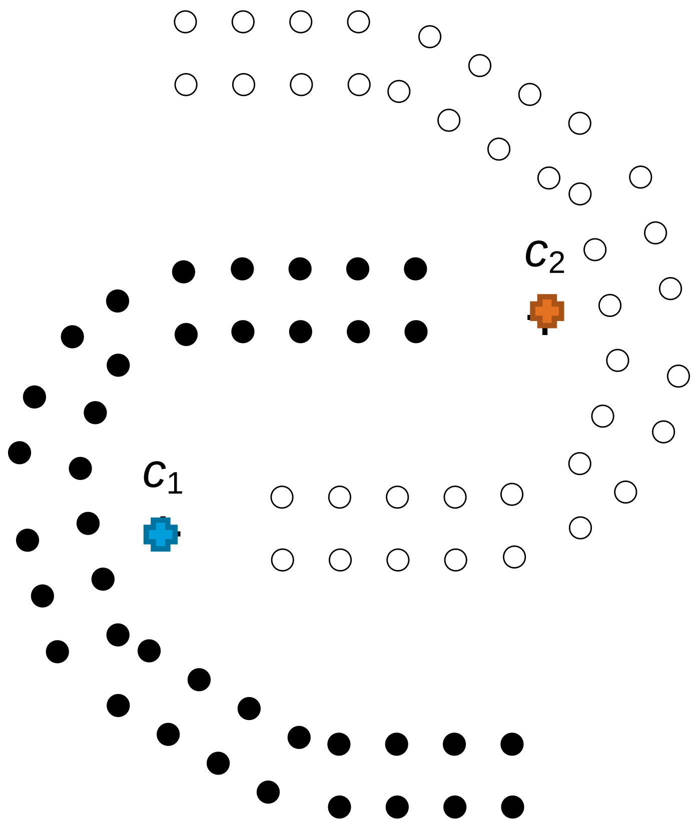
K-Means Problems

K-Means - Problem 1



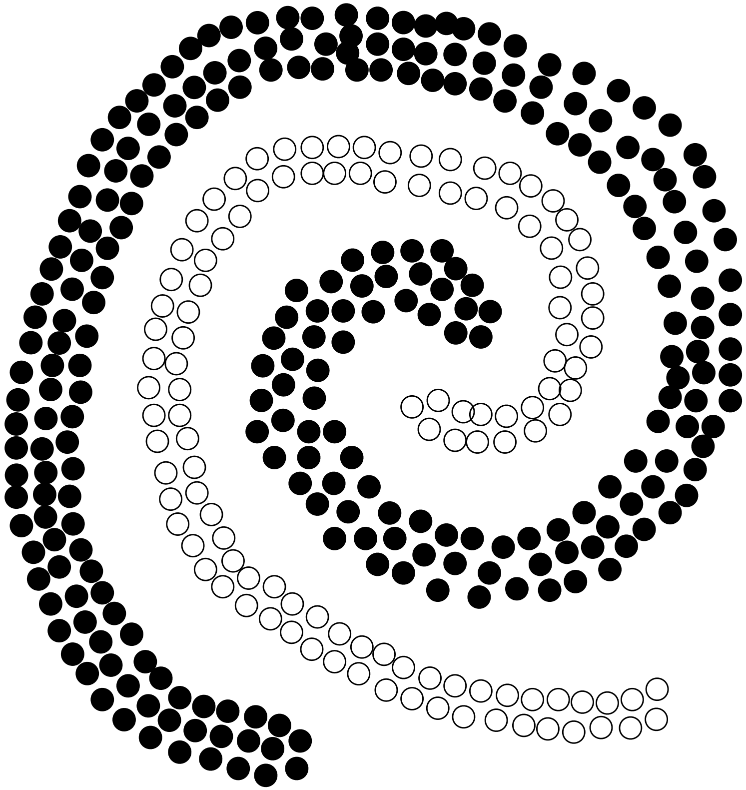
K-Means - Problem 2

Are these centroid good?
Are they accurate?

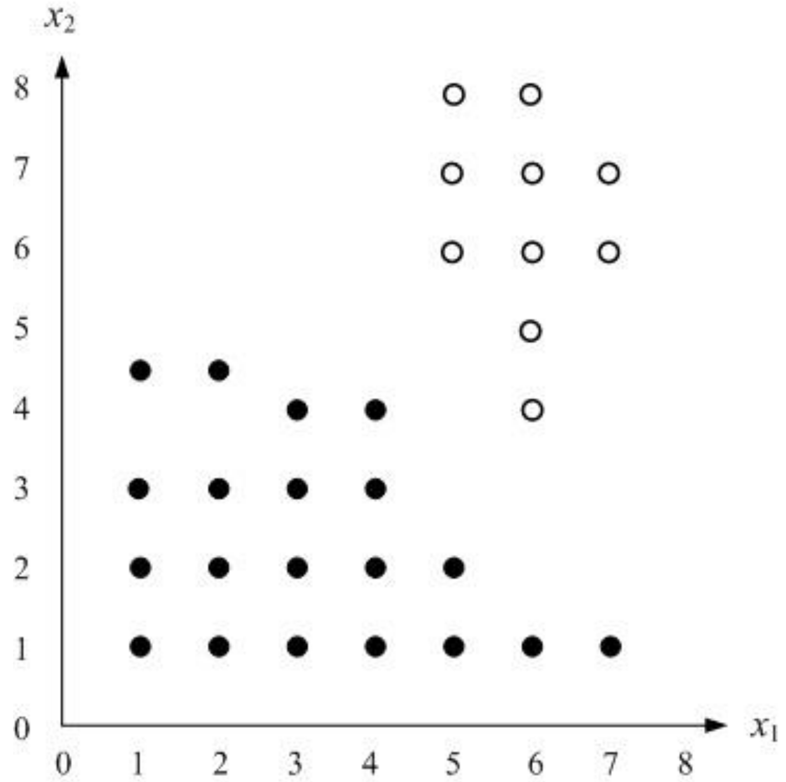


K-Means - Problem 2

Where should the centroids be?

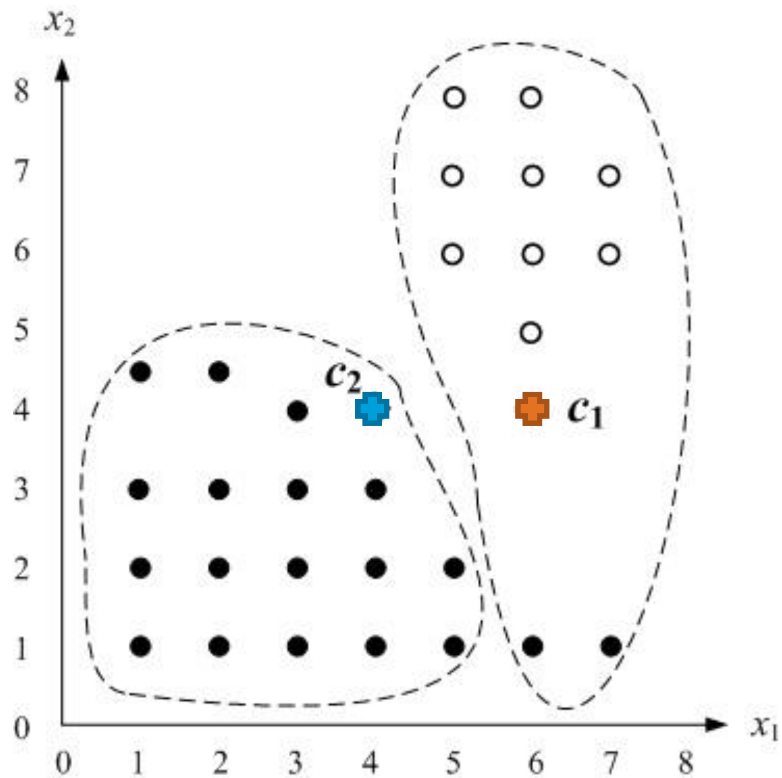


K-Means - Problem 3



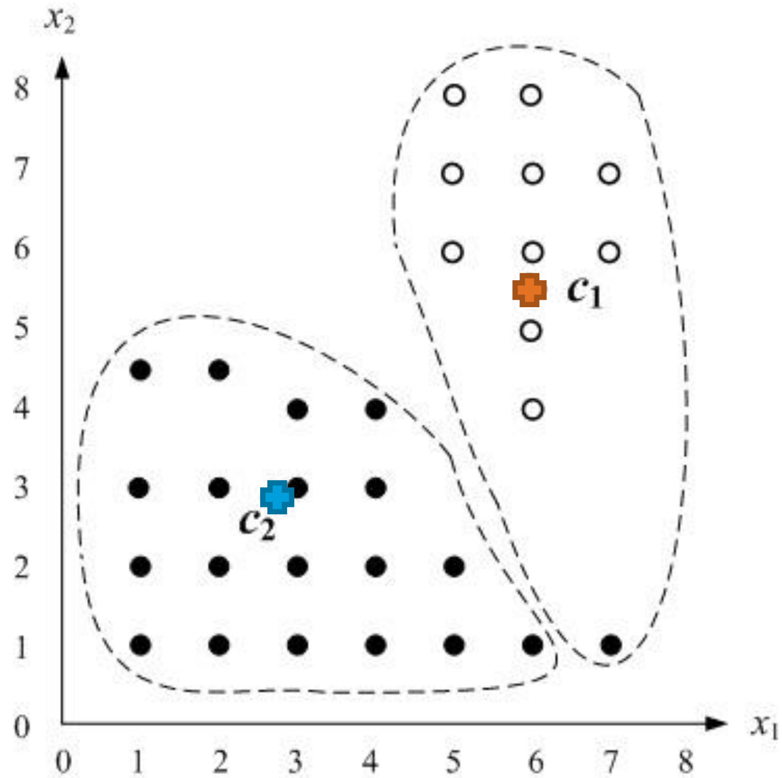
Watch the example,
we try to run K-means twice from 2 different initial centroid

K-Means - Problem 3



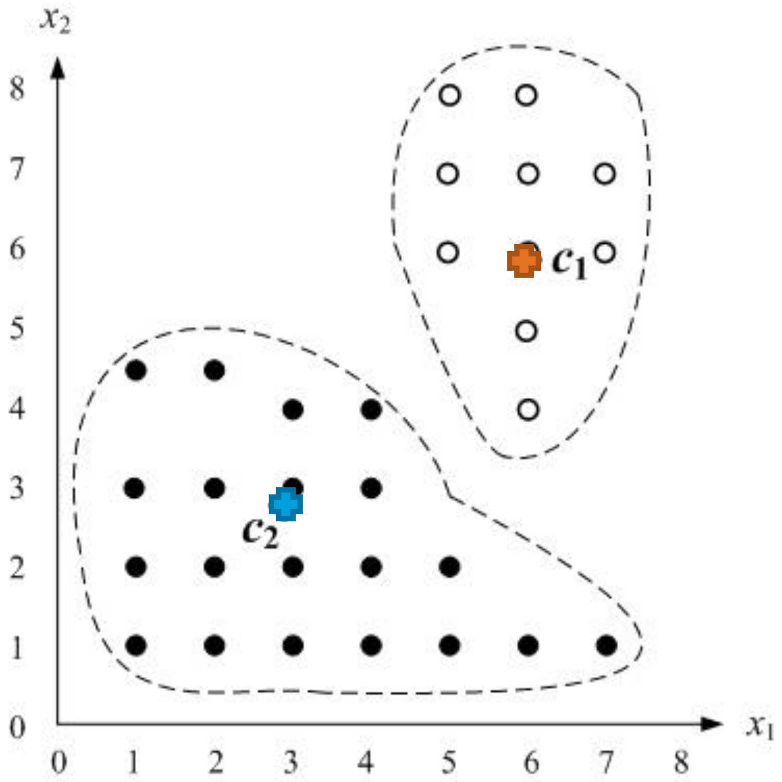
Try 1
Iteration 1

K-Means - Problem 3



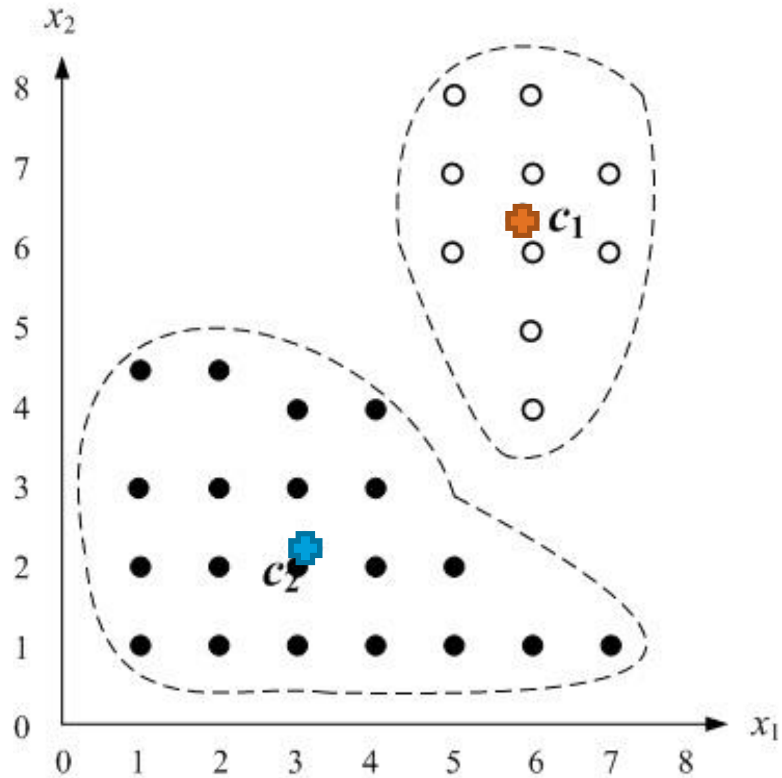
Try 1
Iteration 2

K-Means - Problem 3



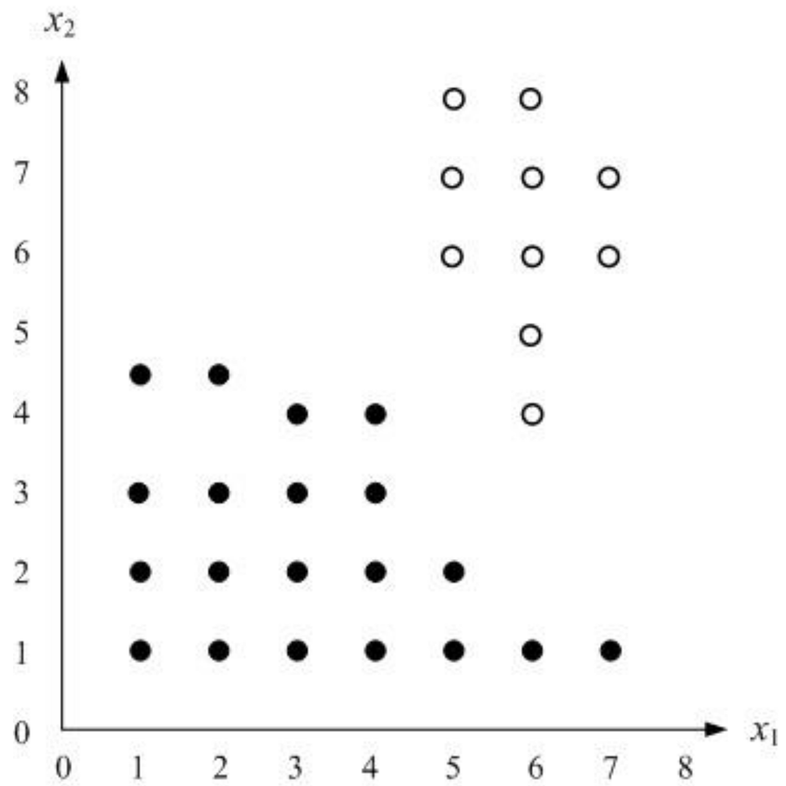
Try 1
Iteration 3

K-Means - Problem 3



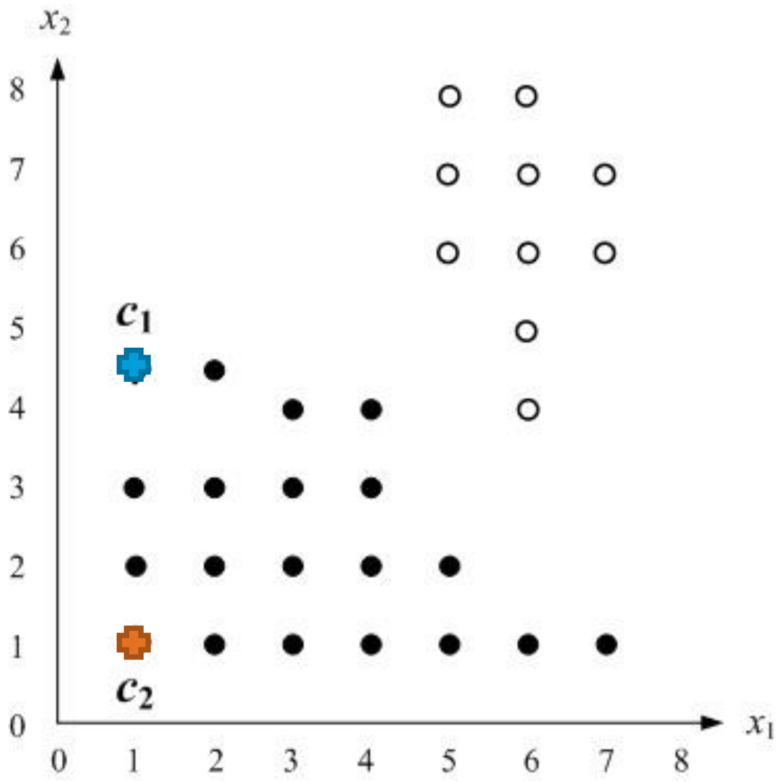
Try 1
Iteration 4, iteration stopped

K-Means - Problem 3



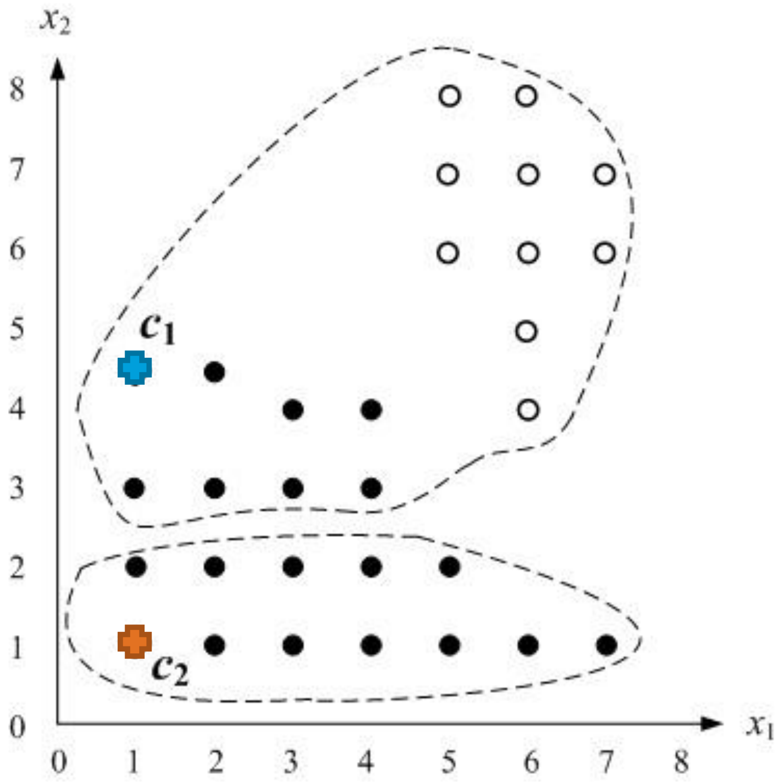
Let's try a different initial of centroids

K-Means - Problem 3



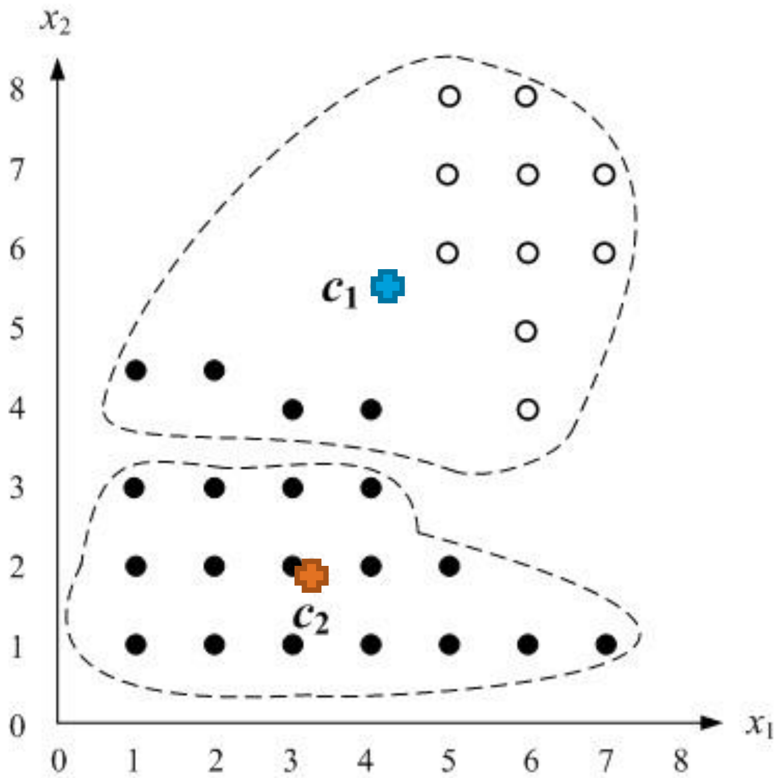
Try 2
Iteration 1

K-Means - Problem 3



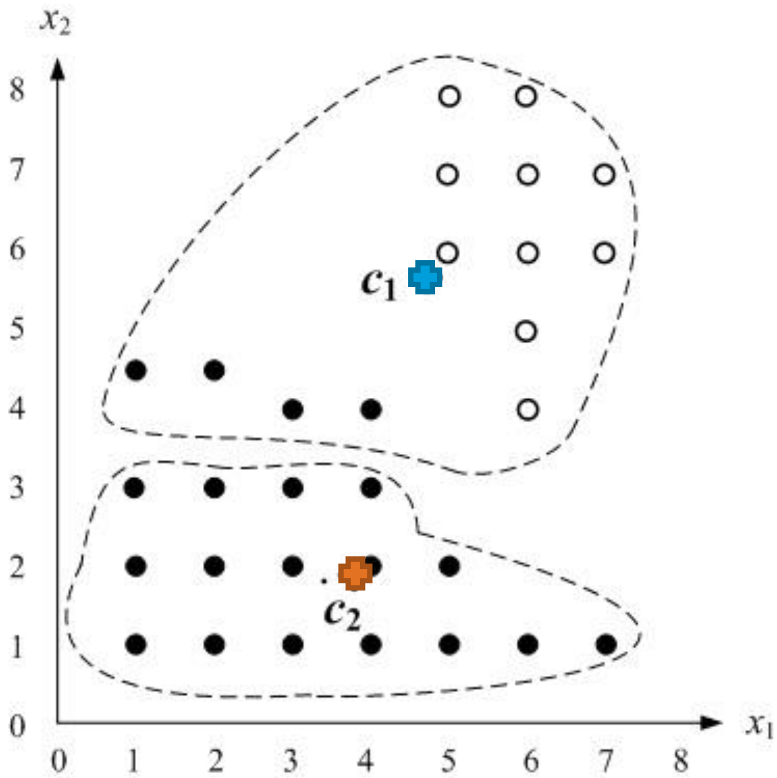
Try 2
Iteration 1

K-Means - Problem 3



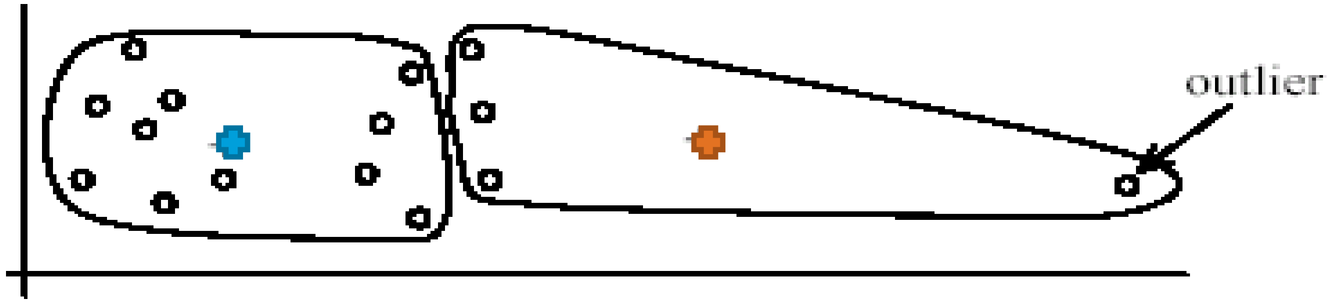
Try 2
Iteration 2

K-Means - Problem 3

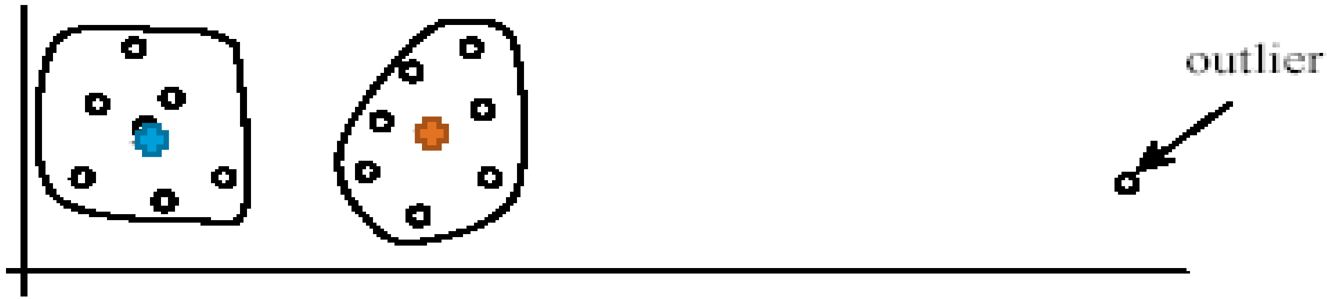


Try 2
Iteration 3, iteration stopped

K-Means - Problem 4

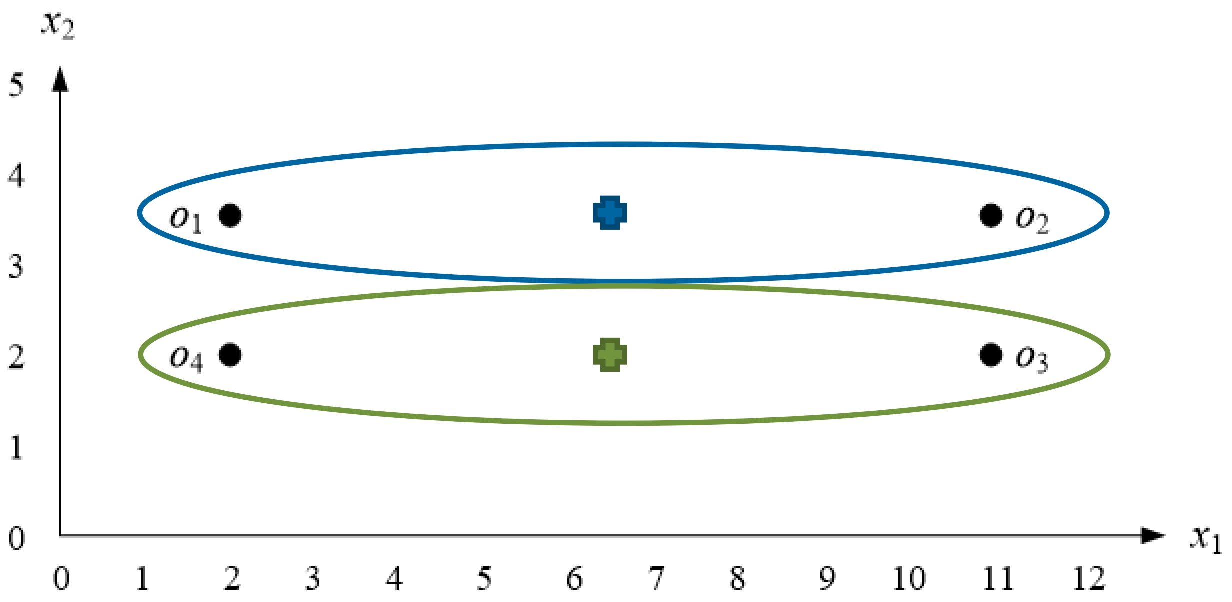


(A): Undesirable clusters



(B): Ideal clusters

K-Means - Problem 5



Nearby points may not end up in the same cluster

K-Means
Pros and Cons

Pros and Cons

- Pros:
  - Relatively simple to implement
  - Good for neat (rounded/convex shaped) data
  - Efficient, time complexity = O(#data * #cluster * #iteration)
- Cons
  - Necessity of specifying k
  - Sensitive to initial assignment of centroids
  - Sensitive to noise and outlier
  - Not suitable for discovering clusters with non-convex shapes
  - Non-deterministic, Can be inconsistent from one run to another
  - Need to define measurement to evaluate the performance

Cluster Quality

- Sum Square Error
  - Aggregate intra-cluster distance

$$SSE = \sum_{j=1}^{K} \sum_{\mathrm{x}_i \in C_j} \left\| \mathrm{c}_j - \mathrm{x}_i \right\|_2^2$$

- Silhouette coefficient

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

$a(i) =$ the average distance of i to all other data within the same cluster

$b(i) =$ the lowest average distance of i to all points in any other cluster

- Davies–Bouldin index, Dunn Index, Mutual Information
- Purity, F-Measure, Jaccard Index, Etc.

Things to try and observe

- Dealing with outliers
  - Try to remove some data points considered as noise
  - Perform random sampling
- Dealing with initial seeds
  - Try out multiple starting points,
    choose cluster result with the smallest SSE
    (or any other performance measure)

Things to try and observe

- Dealing with number of cluster
  - Hopkins Statistic
  - Elbow method
  - Cross-validation
  - Silhouette Analysis

K-Means
Determine the number of cluster

Hopkins Statistic

- To measure to what degree clusters exist in the data
- One way to do this is to compare the data against random data.
- On average, random data should not have clusters.
- Algorithm
  - Let D be a real dataset (data to be clustered)
  - Randomly sample *n* data from D → $(p_1, \ldots, p_n)$
  - Generate *n* random uniform data $randomD$ → $(q_1, \ldots, q_n)$ with the same variation with data D

Hopkins Statistic

- Algorithm
  - For each $p_i \in D$, find it's nearest neighbor $v_j \in D$; then compute the distance between p_i and v_j and denote it as $x_i = dist(p_i, v_j)$ or

    $$x_i = \min_{v \in D}\{dist(p_i, v)\}$$

  - For each $q_i \in randomD$, find it's nearest neighbor $v_j \in D$; then compute the distance between q_i and v_j and denote it as $y_i = dist(q_i, v_j)$ or

    $$y_i = \min_{v \in D, v \neq q_i}\{dist(q_i, v)\}$$

Hopkins Statistic

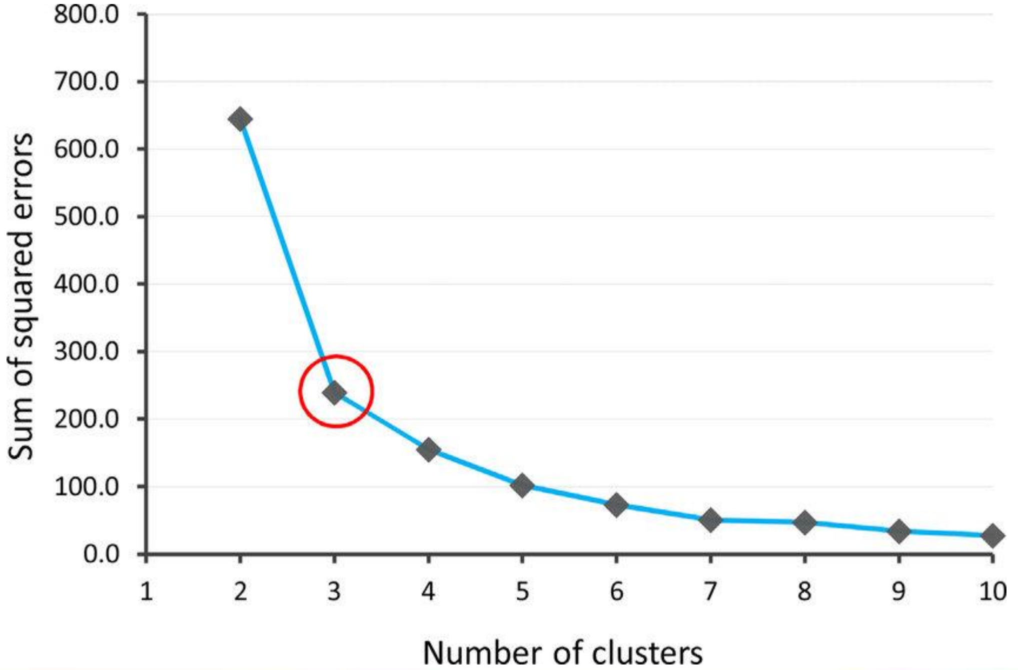
- Algorithm
  - Calculate Hopkins statistic H

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$

  - If the data is uniformly distributed (i.e. no meaningful clusters), then $\sum_{i=1}^{n} y_i$ would be close to $\sum_{i=1}^{n} x_i$, so H is around 0.5
  - But if clusters are present in D, then $\sum_{i=1}^{n} y_i$ would be substantially higher than $\sum_{i=1}^{n} x_i$, so H will larger toward 1.

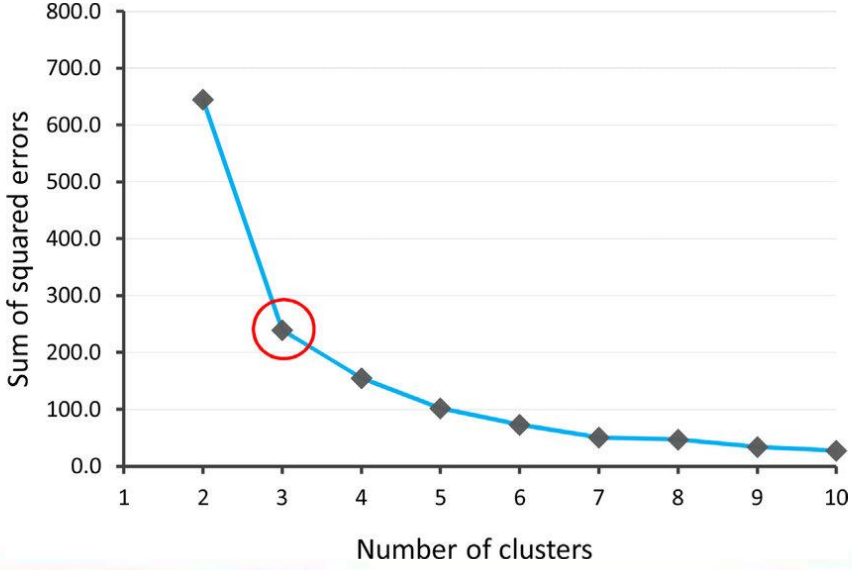
Elbow method

- Try different *k* from k=2 to k=*m*
- Plot Sum Squared Error (SSE) for each *k*



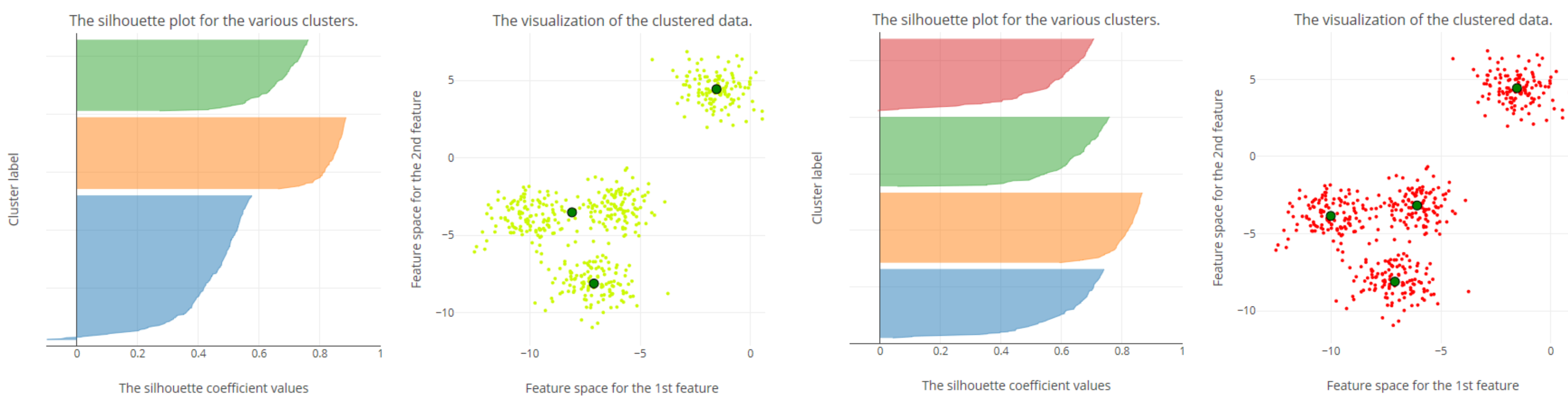
Elbow method

- The more clusters, the lower SSE. Why?
- Choose the minimum number of cluster when the SSE starts to level out
- Use cross validation and average of SSE of each fold



Silhouette Analysis

- A measure of how close each point in one cluster is to points in the neighboring clusters
- A measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation).



Question?


Fakultas Informatika
School of Computing
Telkom University

THANK YOU