Clustering: Unsupervised Learning

IF-3270 Pembelajaran Mesin

Teknik Informatika ITB

Modul 7: Clustering

Fariska Z. Ruskanda, S.T., M.T.
(fariska@informatika.org)

KK IF -Teknik Informatika - STEI ITB

02 Partitional Clustering

IF3270 - Pembelajaran Mesin
(Machine Learning)

Outline

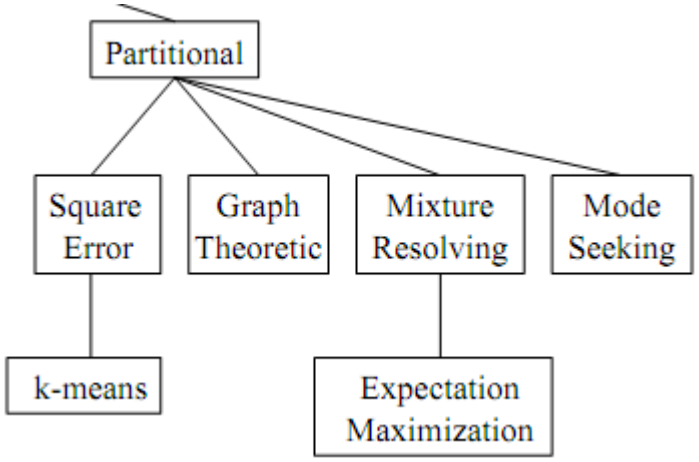
Partinioning Method

Squared Error Clustering

K-Means

K-Medoids

Graph-Theoretic



Partitioning Methods

- Relocation Algorithms
- Probabilistic Clustering
- *K*-medoids Methods
- *K*-means Methods
- Density-Based Algorithms
  - Density-Based Connectivity Clustering
  - Density Functions Clustering

Partitional clustering

Squared-Error/Relocation Clustering, Graph Theoretic, Density-based

Metode *Partitioning*

- Kriteria objektif: objek di dalam suatu cluster memiliki kemiripan yang lebih besar dibanding objek yang berada di cluster yang lain.
  - Minimize square-error function.
- Relokasi iteratif:
  - proses iteratif menempatkan objek ke kluster untuk memperbaiki partisi.
- Contoh: Squared-Error Clustering (k-Means), k-medoids, Graph theoretic

Squared Error Clustering
(Jain dkk, 1999)

- Objective: To obtain a partition which, for a fixed number of **clusters**, minimizes the **square-error**

- **Square-error** is the sum of the Euclidean distances between each pattern and its **cluster** center.
  - Squared error untuk clustering L of a pattern set H (containing k clusters):

$$e^2(\mathscr{X}, \mathscr{L}) = \sum_{j=1}^{K} \sum_{i=1}^{n_j} \left\| \mathbf{x}_i^{(j)} - \mathbf{c}_j \right\|^2$$

  - H: jumlah pattern / objek
  - x: pattern/objek, c: centroid, n: jumlah object dalam cluster K, K: cluster

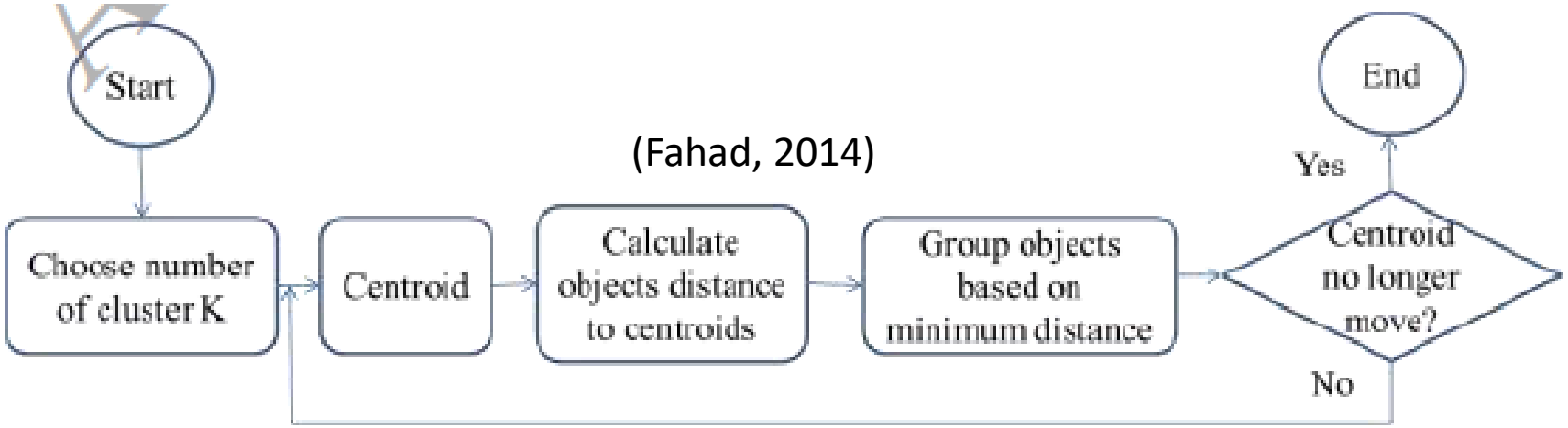
Squared Error Clustering
(Jain dkk, 1999)

(1) Select an initial partition of the patterns with a fixed number of clusters and cluster centers.

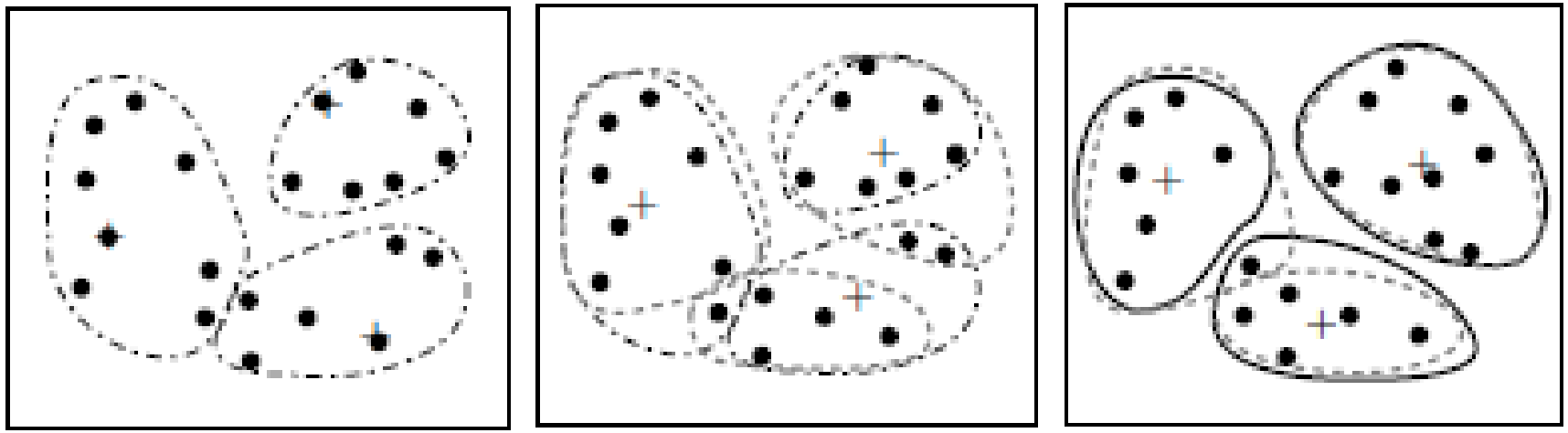
(2) Assign each pattern to its closest cluster center and compute the new cluster centers as the centroids of the clusters. Repeat this step until convergence is achieved, i.e., until the cluster membership is stable.

(3) Merge and split clusters based on some heuristic information, optionally repeating step 2.

Review: K-Means



(Fahad, 2014)



(a) (b) (c)

Algoritma K-Means
(Han & Kamber, 2006)

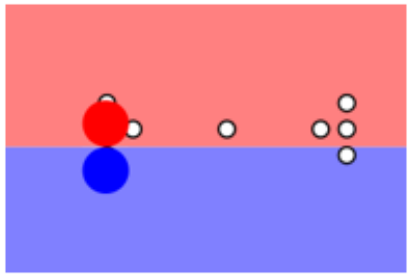
- The k-means algorithm for partitioning, where each cluster's center is represented by mean value of the objects in the cluster.
- Input: **k** (number of clusters), D (data set containing **n** objects)
- Output: A set of k clusters
- Kompleksitas: **O(nkt)**; t: jumlah iterasi; k<<n~~; t<<n~~
- Method:
  1. Arbitrarily choose k objects from D as the initial cluster centers
  2. Repeat
     - (re) assign each object to the cluster to which the object is the most similar based on the mean value of the objects in the clusters;
     - Update the cluster means, i.e. calculate the mean value of the objects for each cluster;
  3. Until no change (convergent)

Notes

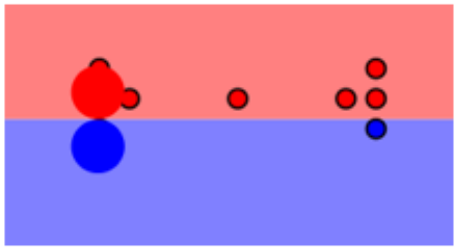
- *K*-means can be applied only when mean of a cluster is defined
  - Variants: *k*-modes
- Konvergen: square-error minimum

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2$$

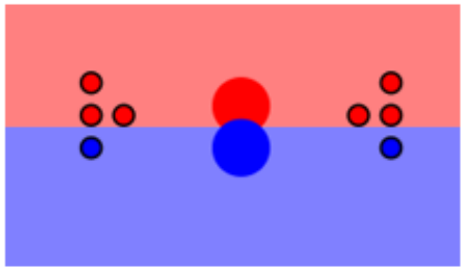
K-Means: Pengaruh Cluster Awal



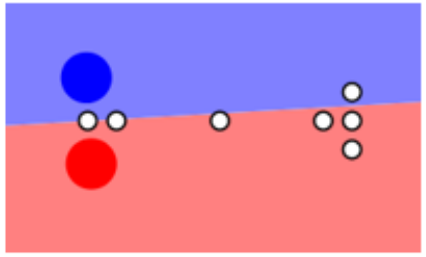
i=0
{A,**B**,D,E,F,G,H}, {**C**,I}



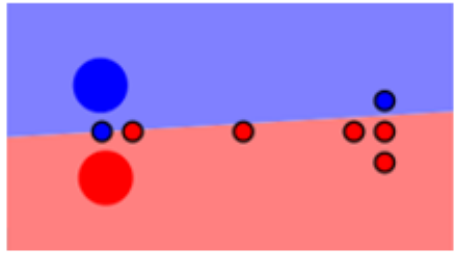
i=1
{A,B,D,E,F,G,H}, {C,I}



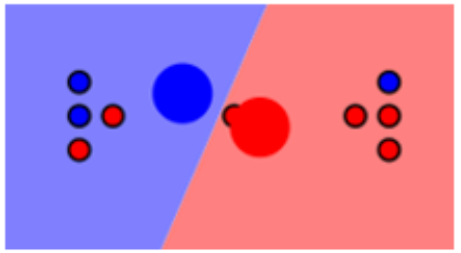
i=2
{A,B,D,E,F,G,H}, {C,I}



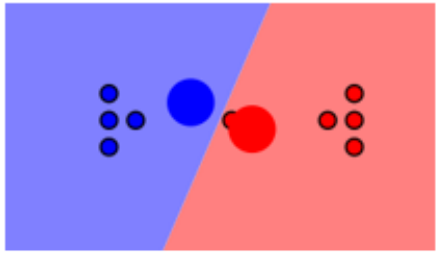
i=0
{**A**,G}, {B,**C**,D,E,F,H,I}



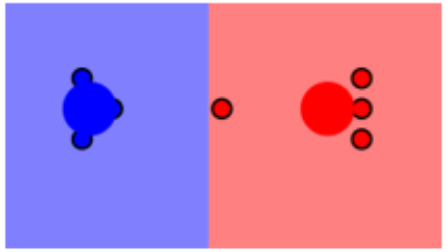
i=1
{A,B,G}, {C,D,E,F,H,I}



i=2
{A,B,G}, {C,D,E,F,H,I}



i=3
{A,B,C,D}, {E,F,G,H,I}



i=4
{A,B,C,D}, {E,F,G,H,I}

Sumber: https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

Kelemahan K-Means (Berkhin, 2006)

- Tidak ada panduan penentuan nilai k yang baik
- Hasil sangat dipengaruhi oleh inisialisasi centroid
  - Sering berhenti pada optimum lokal
  - Hasil akhir tidak stabil
- Algoritma tidak *scalable*
- Mean hanya terdefinisi untuk atribut numerik
  - Atribut nominal → metode k-modes
- Prosesnya sensitif terhadap outliers
  - Outlier dgn nilai yang ekstrim besar mempengaruhi mean pada centroid

K-Medoids

- K-medoids lebih handal dibanding k-means dalam menangani noise atau outlier (Han & Kamber, 2006)
- Absolute-error criterion:

$$E = \sum_{j=1}^{k} \sum_{p \in C_j} |p - o_j|,$$

Partitioning Around Medoids (PAM)

PAM, a *k*-medoids algorithm for partitioning based on medoid or central objects.

Input: *k* (number of clusters), *D* (data set containing *n* objects)

Output: A set of *k* clusters

Kompleksitas: $O(k(n-k)^2)$ untuk 1 iterasi → $O(k^3*n^2)$ high complexity

Method:

(1) arbitrarily choose *k* objects in *D* as the initial representative objects or seeds;

(2) repeat

(3) assign each remaining object to the cluster with the nearest representative object;

(4) randomly select a nonrepresentative object, o_{random};

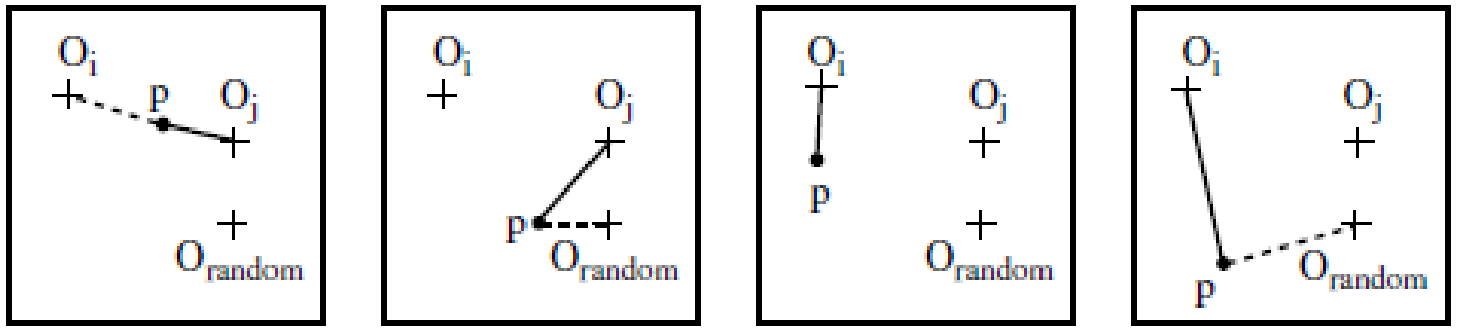
(5) compute the total cost, *S*, of swapping representative object, o_j, with o_{random};

(6) if $S < 0$ then swap o_j with o_{random} to form the new set of *k* representative objects;

(7) until no change;

Penjelasan (5)

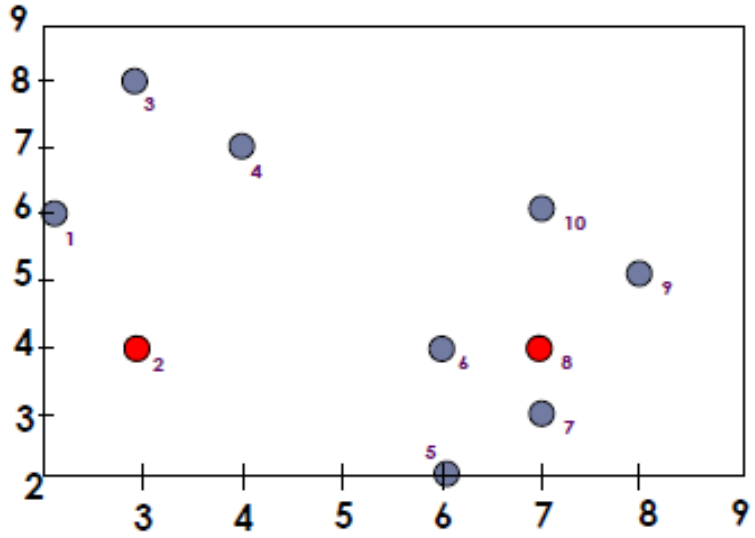
- Compute the total cost, *S*, of swapping representative object, o_j, with o_{random}
- S = difference of absolute error
- Oj is representative object of o_{random}
- S is calculated after temporary assignment of each nonrepresentative object based on swapping cases below.



K-Medoids Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



Goal: create two clusters

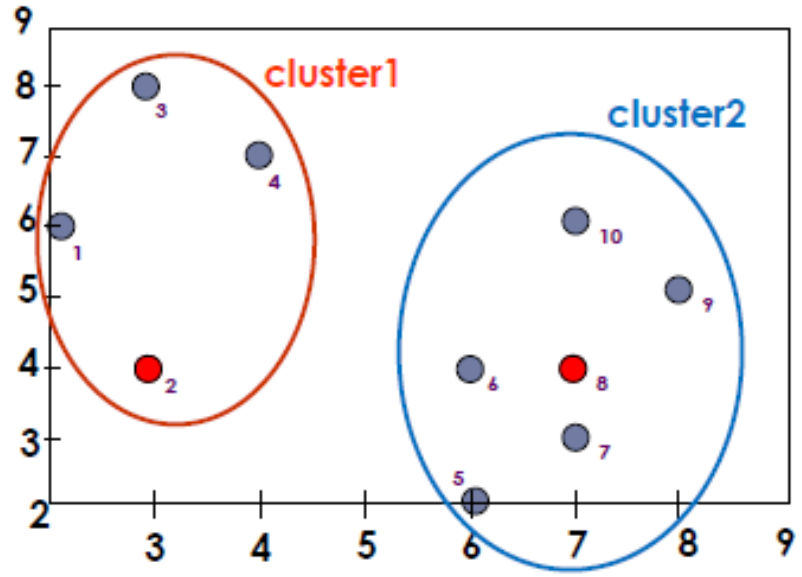
Choose randmly two medoids

$O_2 = (3,4)$
$O_8 = (7,4)$

K-Medoids Example (2)

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Assign each object to the closest representative object

→Using L1 Metric (Manhattan), we form the following clusters

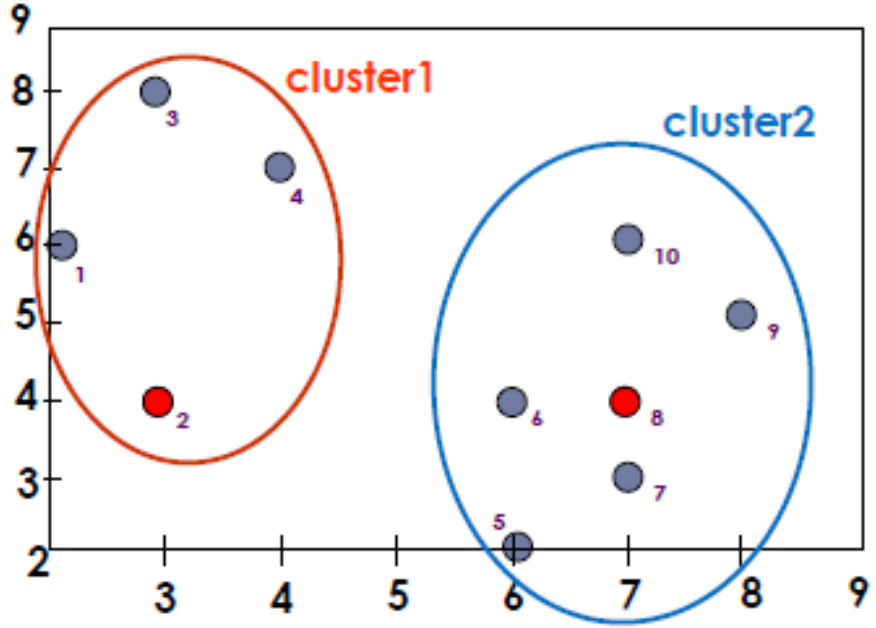
Cluster1 = {O_1, O_2, O_3, O_4}

Cluster2 = {O_5, O_6, O_7, O_8, O_9, O_{10}}

K-Medoids Example (3)

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Compute the absolute error criterion **[for the set of Medoids (O2,O8)]**

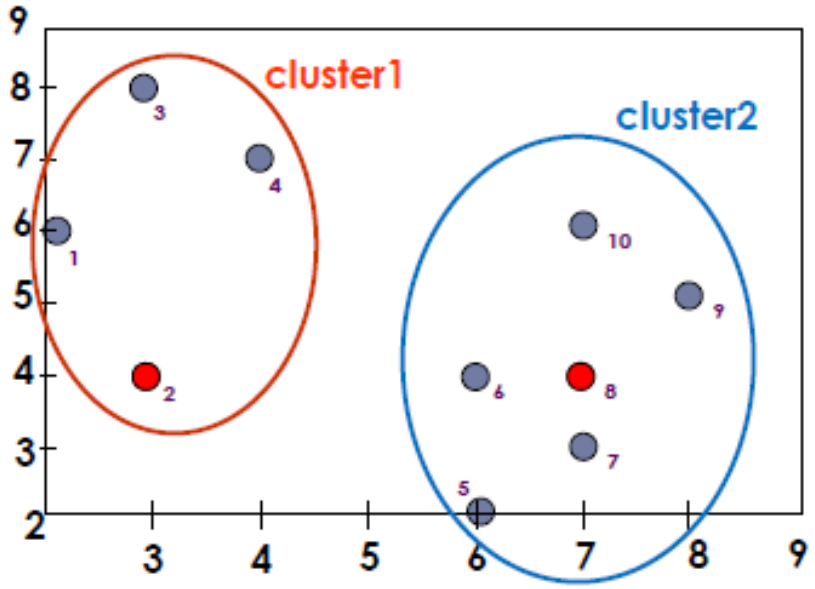
$$E=\sum_{i-1}^{k}\sum_{p\in C_i}|p-o_i|=|o_1-o_2|+|o_3-o_2|+|o_4-o_2|$$

$$+|o_5-o_8|+|o_6-o_8|+|o_7-o_8|+|o_9-o_8|+|o_{10}-o_8|$$

K-Medoids Example (4)

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



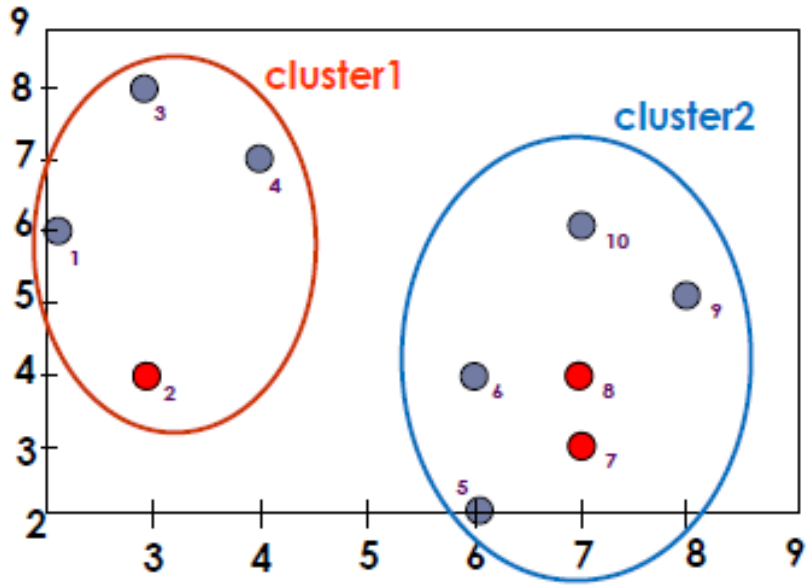
→The absolute error criterion **[for the set of Medoids (O2,O8)]**

$$E = (3+4+4)+(3+1+1+2+2) = 20$$

K-Medoids Example (5)

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Choose a random object O_7

→Swap O8 and O7

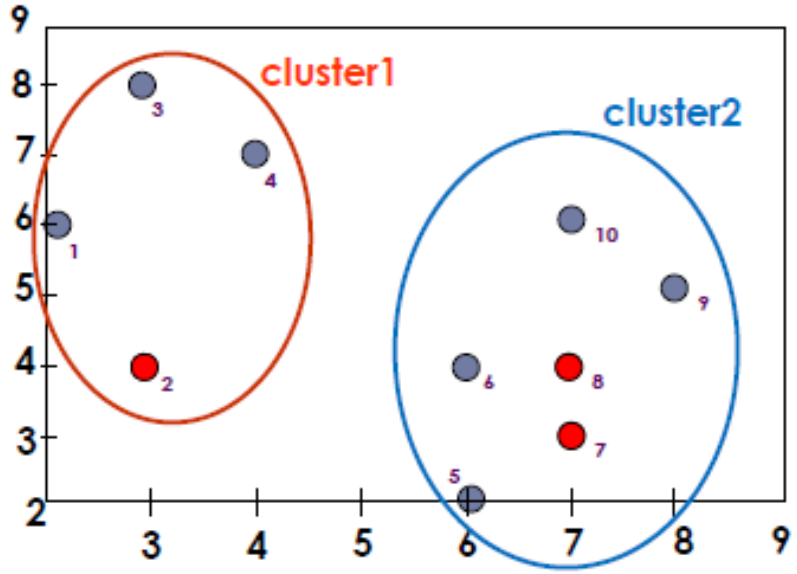
→Compute the absolute error criterion **[for the set of Medoids (O2,O7)]**

$$E = (3+4+4)+(2+2+1+3+3) = 22$$

K-Medoids Example (6)

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6

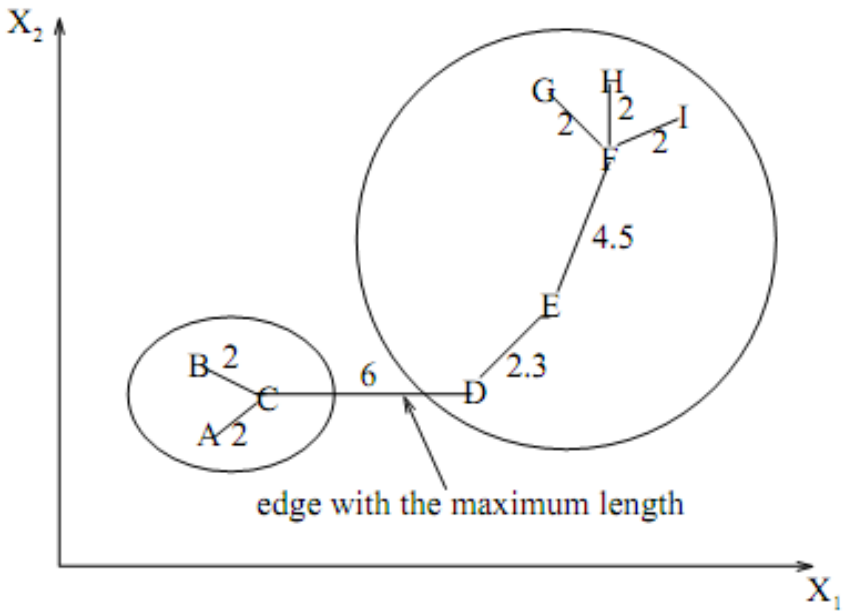


- In this example, changing the medoid of cluster 2 did not change the assignments of objects to clusters.
- What are the possible cases when we replace a medoid by another object?

Analisis K-Medoids

- K-medoids lebih robust terhadap outlier karena medoid tidak sensitive terhadap nilai outlier seperti k-means.
- K-medoids prosesnya membutuhkan waktu lebih banyak

Graph-Theoretic Clustering

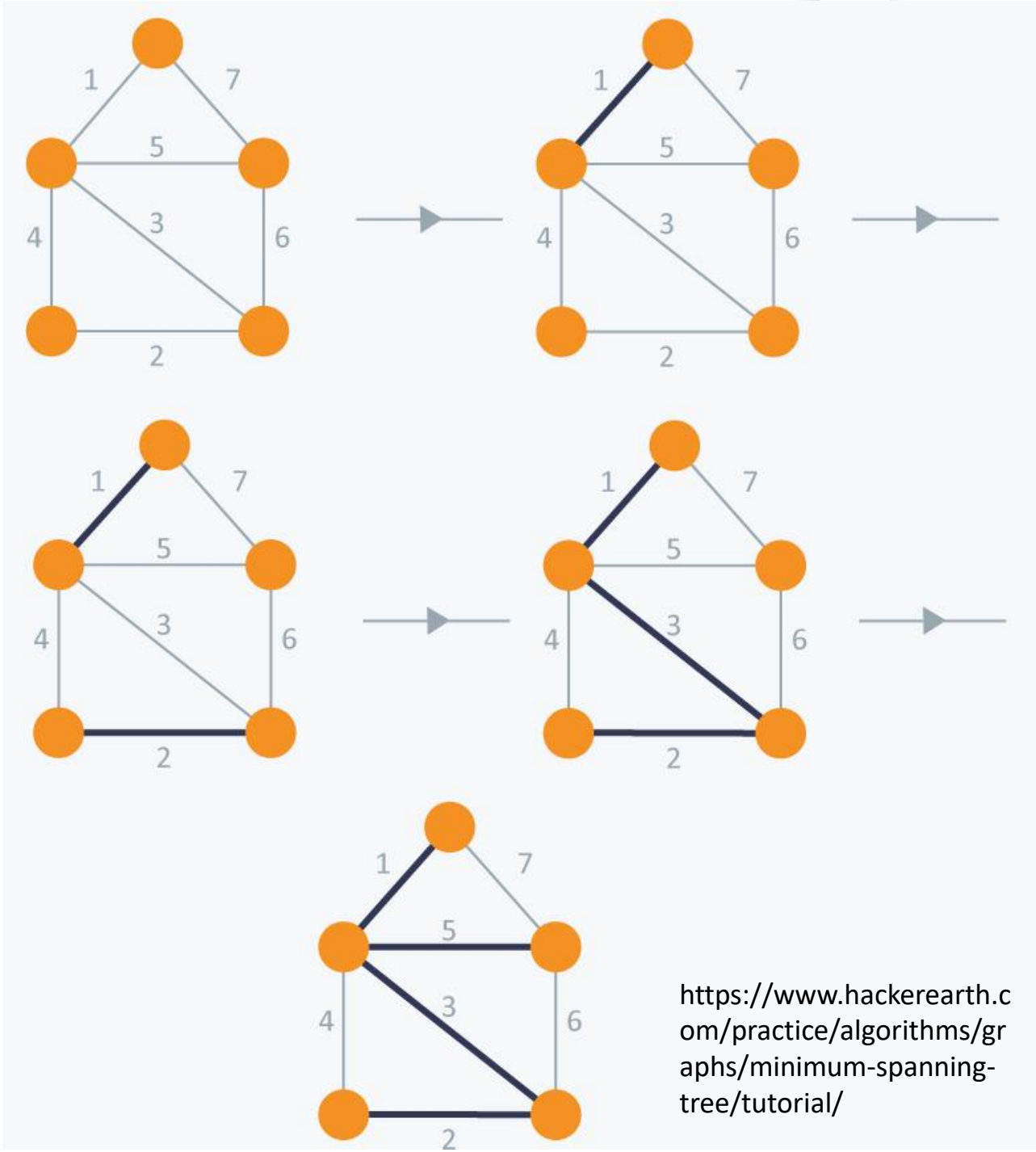


- MST-based clustering construct the minimal spanning tree (MST) of the data
- Delete the MST edges with the largest lengths to generate clusters:

  C-D (2 cluster), E-F (3 cluster)

Sumber: Jain dkk (1999)

Review: Kruskal's Algorithm



https://www.hackerearth.com/practice/algorithms/graphs/minimum-spanning-tree/tutorial/

MST-based Clustering

- Time complexity: O(|E|*log |V|) ; v: vertices, e: edges
- Advantages: clustering in high efficiency, the clustering result with high accuracy
- Disadvantages: the time complexity increasing dramatically with the increasing of graph complexity;

Q & A: K-means

- Q: Kompleksitas algoritma K-means O(nkt), tetapi disebutkan tidak scalable.
- A: Jumlah iterasi untuk kasus terburuknya bisa mencapai eksponensial / superpolynomial.
  - Xu & Tian (2017): Kompleksitas algoritma K-means O(nkt) dikategorikan kompleksitas rendah.
    - Dimensi diabaikan → O(nktd) ; d dimensi data
  - Berkhin (2006): Scalability both in terms of computing time and memory requirements.
  - Arthur & Vassilvitskii (2006): The worst-case running time of k-means is superpolynomial number of iteration($2^{\Omega(n)}$ iteration).

Q & A: K-Medoid

- Q: Total cost, *S*, of swapping representative object < 0 ?
- A: S = *difference in absolute-error* value (Han & Kamber, 2006)
- The cost function calculates the *difference in absolute-error* value if a current representative object is replaced by a nonrepresentative object.
- If the total cost is negative, then ***oj is replaced or swapped with orandom since the actual absolute*** error *E would be reduced.*
- *If the total cost is positive, the current representative object,* ***oj, is considered acceptable, and nothing is changed in the iteration.***

Q & A: K-Medoid

- Q: bisa hanya 1 iterasi berhenti jika Orandom tidak lebih baik ?
- A: Ya

New k-medoids

Step 1: (Select initial medoids)

1-1. Calculate the distance between every pair of all objects based on the chosen dissimilarity measure (Euclidean distance in our case).

1-2. Calculate v_j for object j as follows:

$$v_j = \sum_{i=1}^{n} \frac{d_{ij}}{\sum_{l=1}^{n} d_{il}}, \quad j = 1,\ldots,n \tag{2}$$

1-3. Sort v_j's in ascending order. Select k objects having the first k smallest values as initial medoids.

1-4. Obtain the initial cluster result by assigning each object to the nearest medoid.

1-5. Calculate the sum of distances from all objects to their medoids.

Step 2: (Update medoids)

Find a new medoid of each cluster, which is the object minimizing the total distance to other objects in its cluster. Update the current medoid in each cluster by replacing with the new medoid.

Step 3: (Assign objects to medoids)

3-1. Assign each object to the nearest medoid and obtain the cluster result.

3-2. Calculate the sum of distance from all objects to their medoids. If the sum is equal to the previous one, then stop the algorithm. Otherwise, go back to the Step 2.

Park, H. S., & Jun, C. H. (2009). A simple and fast algorithm for K-medoids clustering. *Expert systems with applications*, *36*(2), 3336-3341.

Q & A: Square-error vs Absolute error

- Q: Perbedaan penggunaan square-error dan abs error pada clustering?
- A: **Square-error** is the sum of the Euclidean distances between each pattern and its **cluster** center.
- Minimize sum of square error: find mean
- Minimize sum of absolute error: find median (median ignores outliers)
- Squared error penalizes large errors more than does absolute error and is more forgiving of small errors than absolute error is.


03 Density-based Clustering

IF3270 Pembelajaran Mesin