IF4071 Pembelajaran Mesin

Clustering: Unsupervised Learning

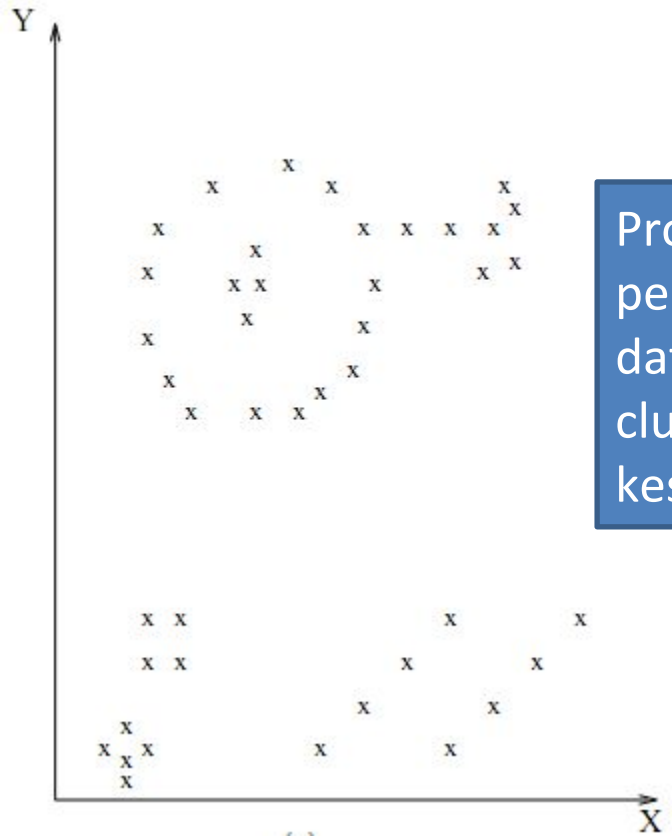
Dessi Puji Lestari/Masayu Leylia Khodra

Semester Ganjil 2018/2019

Referensi

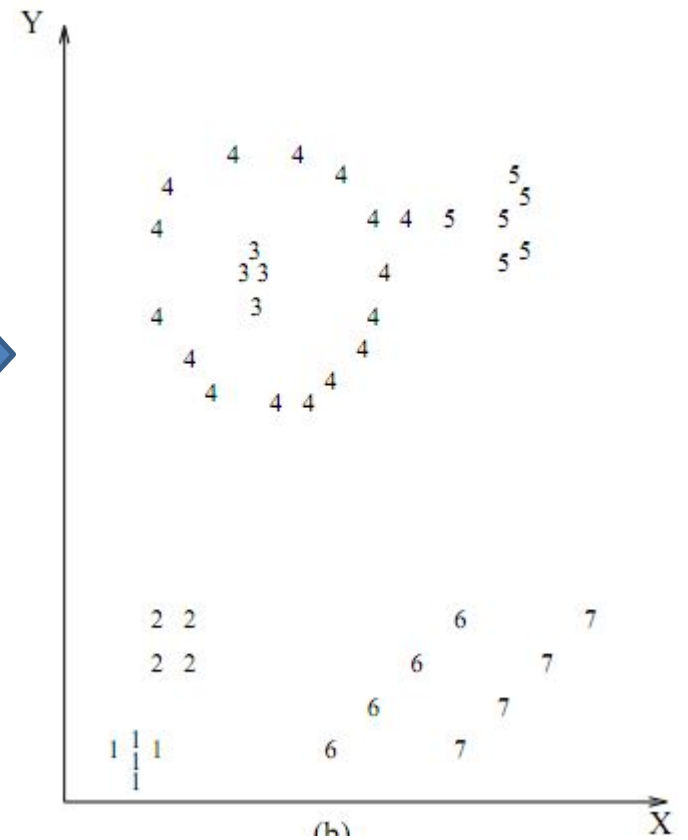
- Jiawei Han , Micheline Kamber (2006), Data Mining: Concepts and Techniques.
- Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, *2*(2), 165-193.
- Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., ... & Bouras, A. (2014). A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE transactions on emerging topics in computing*, *2*(3), 267-279.
- Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., ... & Lin, C. T. (2017). A review of clustering techniques and developments. *Neurocomputing*, *267*, 664-681.
- A.K. Jain, M.N. Murty, P.J. FLYNN (1999), Data Clustering: A Review. ACM computing surveys
- Pengyu Hong (2005), Introduction to Hierarchical Clustering Analysis
- Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data* (pp. 25-71). Springer, Berlin, Heidelberg.
- Jiawei Han & Micheline Kamber (2006), Data Mining: Concepts and Technique
- Rui Xu, Donald Wunsch (2005), Survey of Clustering Algorithm
- DBSCAN http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=72F8D8F33A502FAB448D4C13809D83C3?doi=10.1.1.71.1980&rep=rep1&type=pdf
- http://www.cse.buffalo.edu/~jing/cse601/fa12/materials/clustering_density.pdf

Clustering: What ?



Input Data

Proses pengelompokan data menjadi clusters berbasis kesamaan data



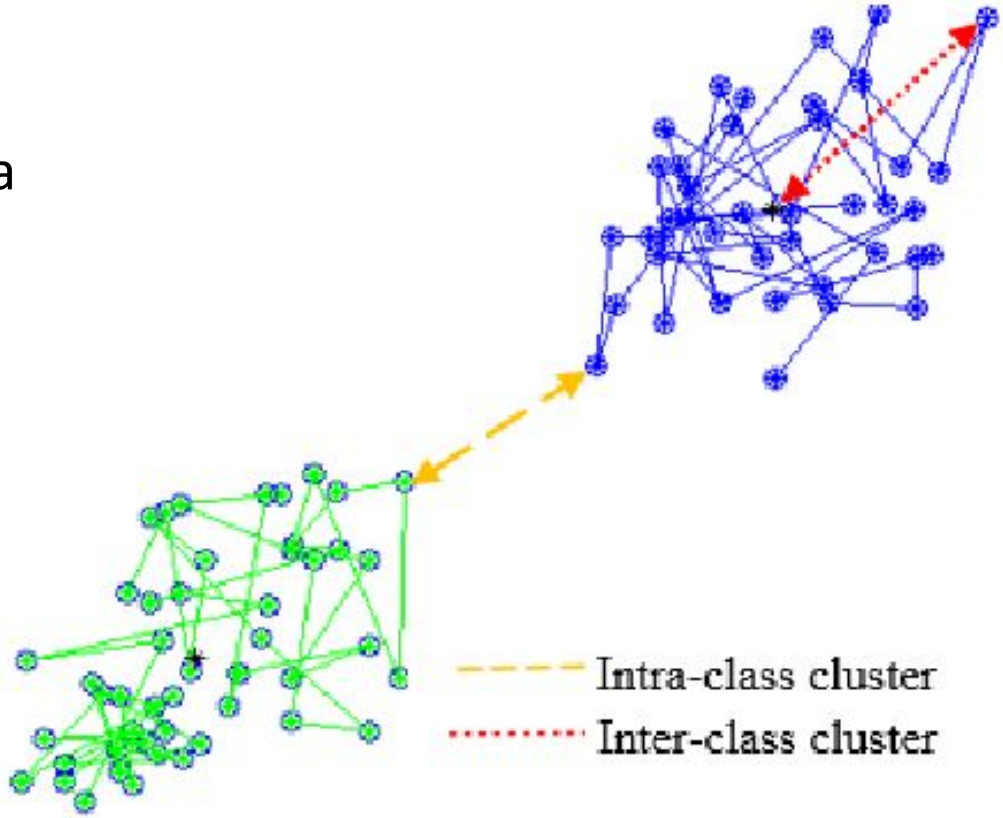
Desired cluster

Unsupervised learning = learning from raw data

Sumber: Jain dkk (1999)

Clustering: Finding natural groups

- High intra-cluster similarity
  - Data pd cluster yang sama harus semirip mungkin
- Low inter-cluster similarity
  - Data pd cluster yang berbeda harus sejauh mungkin
- Pengukuran kemiripan dan jarak harus jelas dan punya semantik praktikal (sesuai domain)

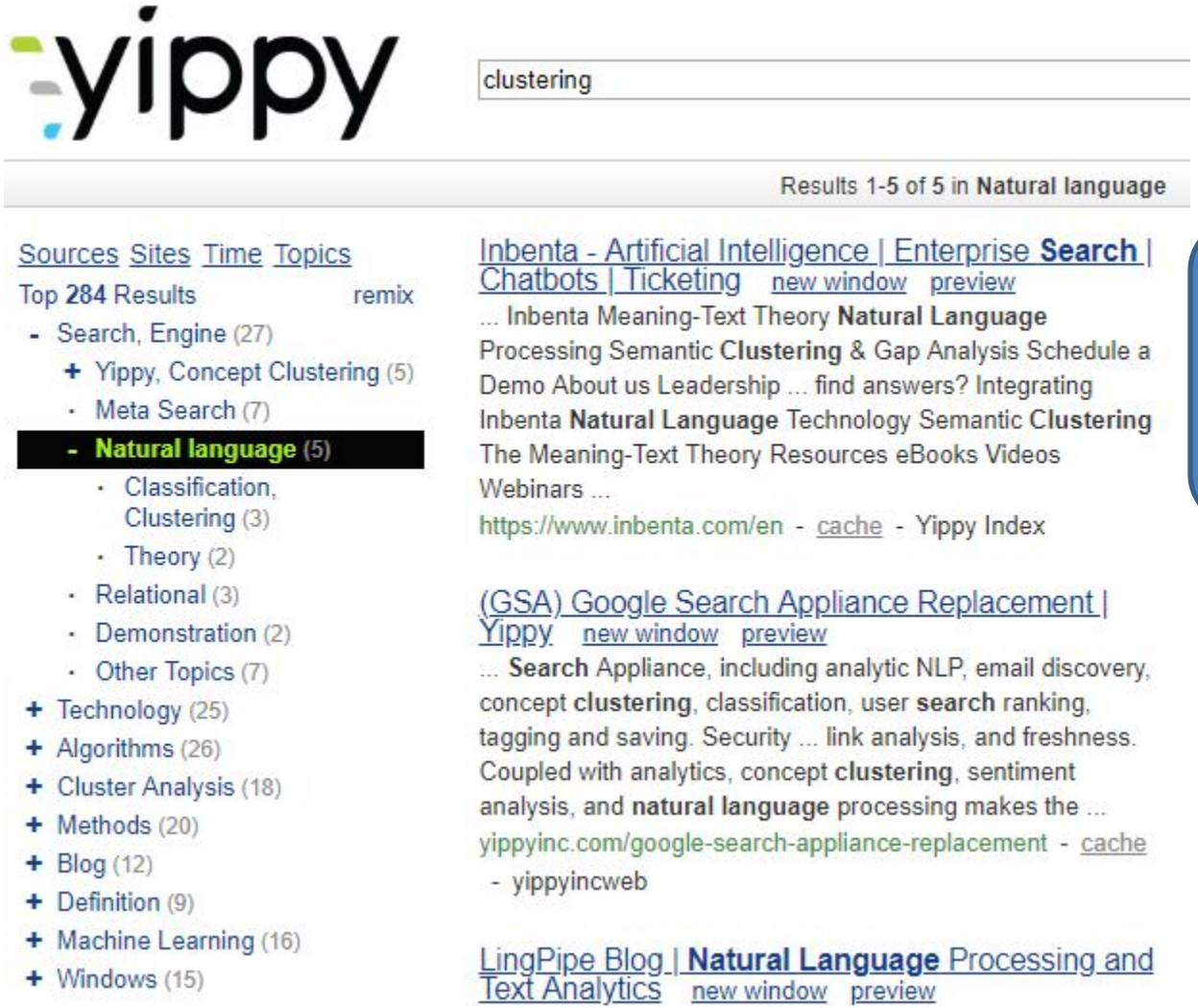


https://www.researchgate.net/profile/Sharifah_Sakinah_Syed_Ahmad/publication/280627665/

Clustering: Why ?

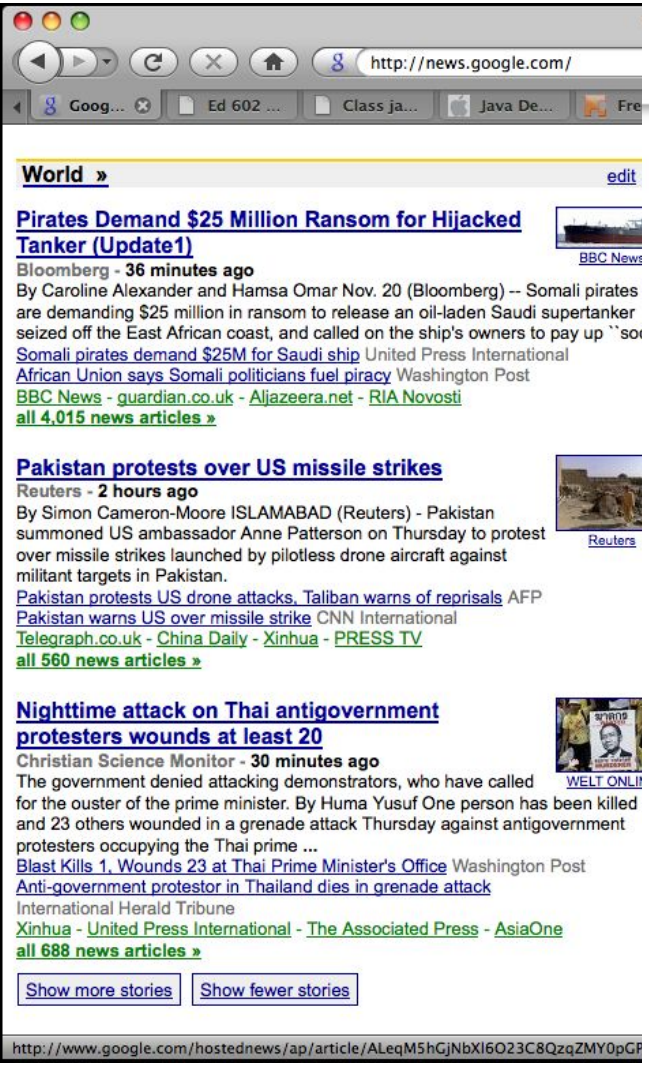
- Data discovery (cluster = struktur internal data)
  - Contoh: search engine, news aggregator, gen
- Tujuan awalnya partisi / pengelompokan
  - Contoh: segmentasi pasar
- Bagian dari teknik lainnya
  - Contoh: peringkasan berbasis clustering

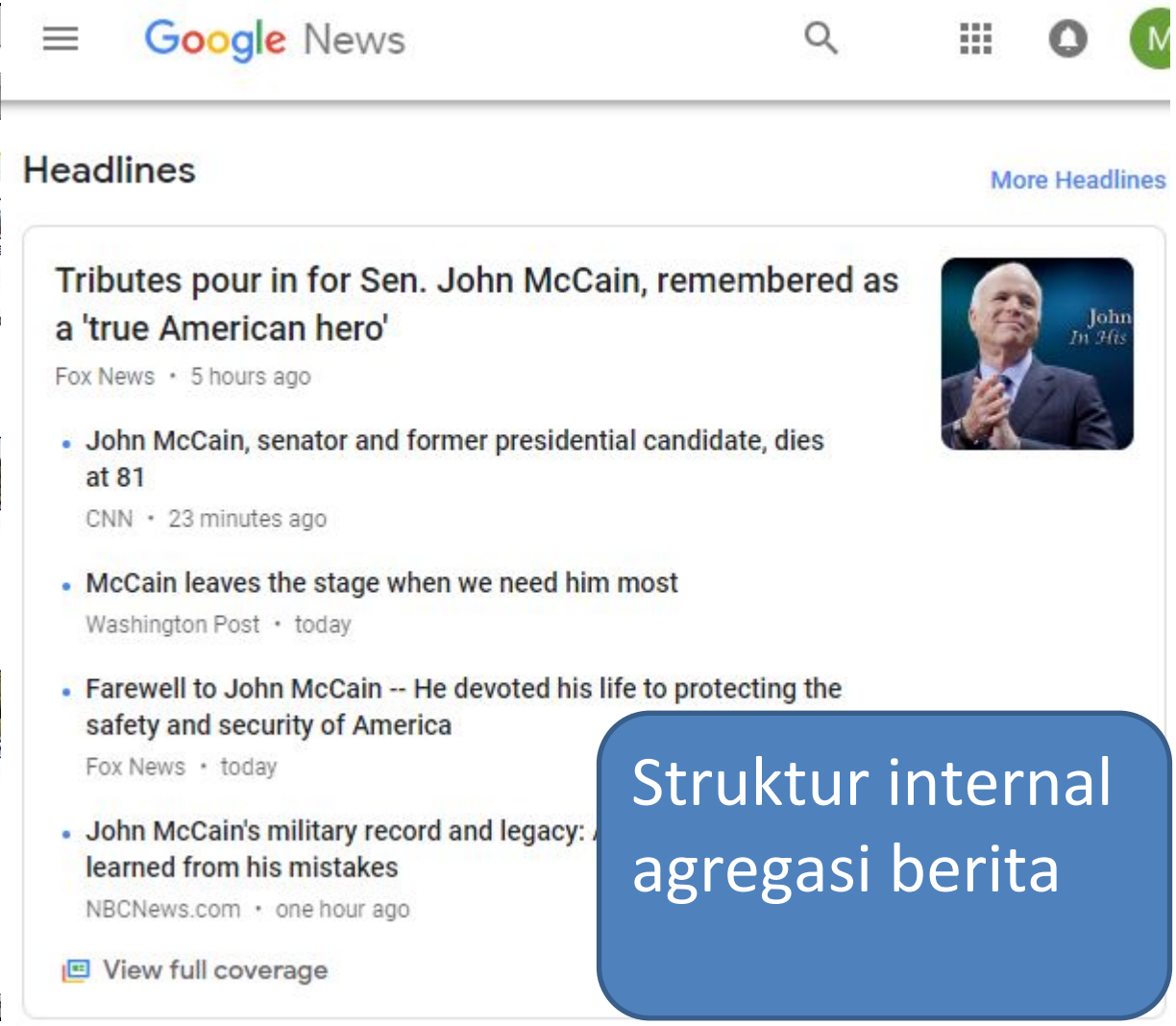
Why: Clustering pada Search Engine



Struktur internal hasil pencarian

Why: Clustering pada News Aggregator

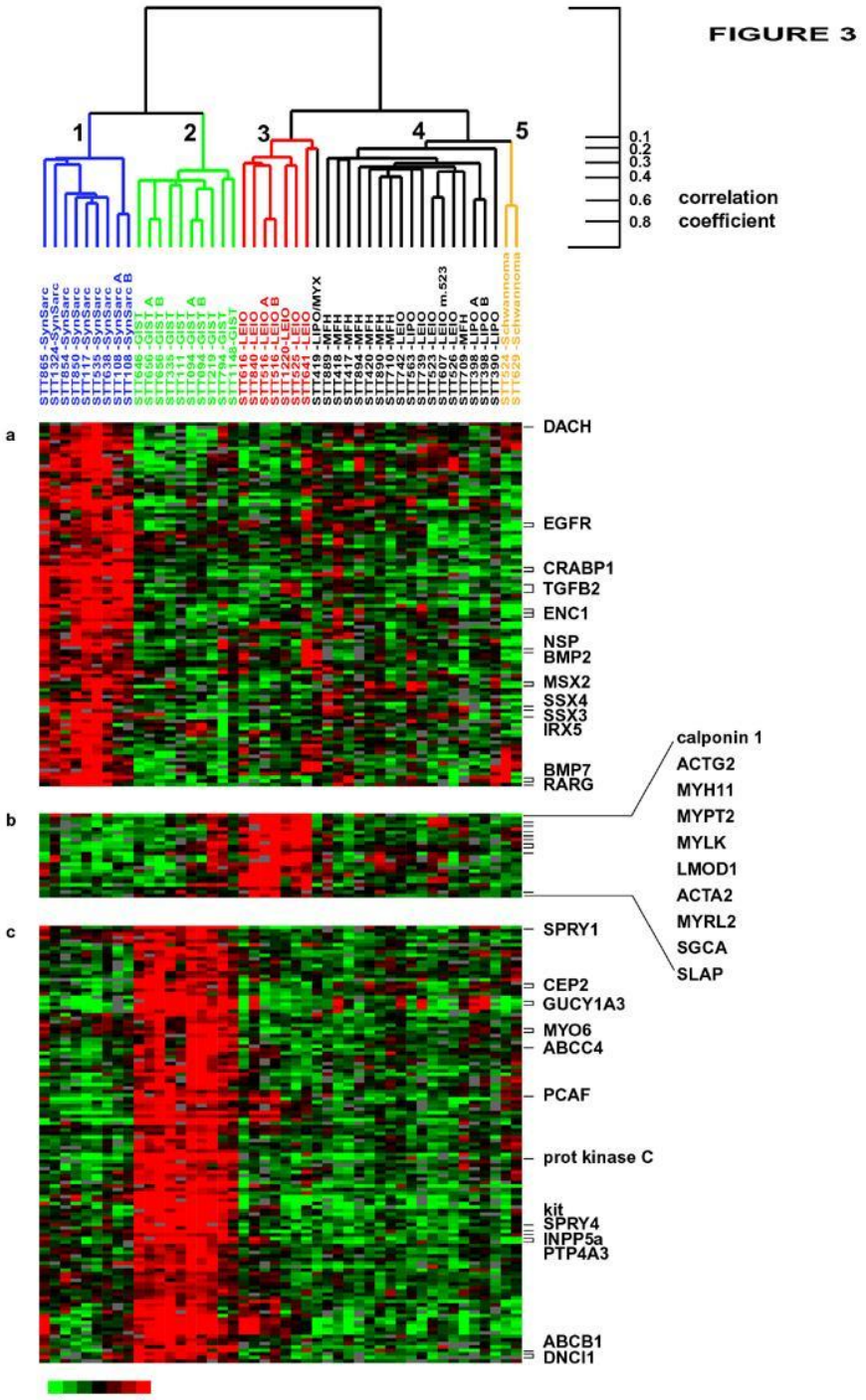




Struktur internal agregasi berita

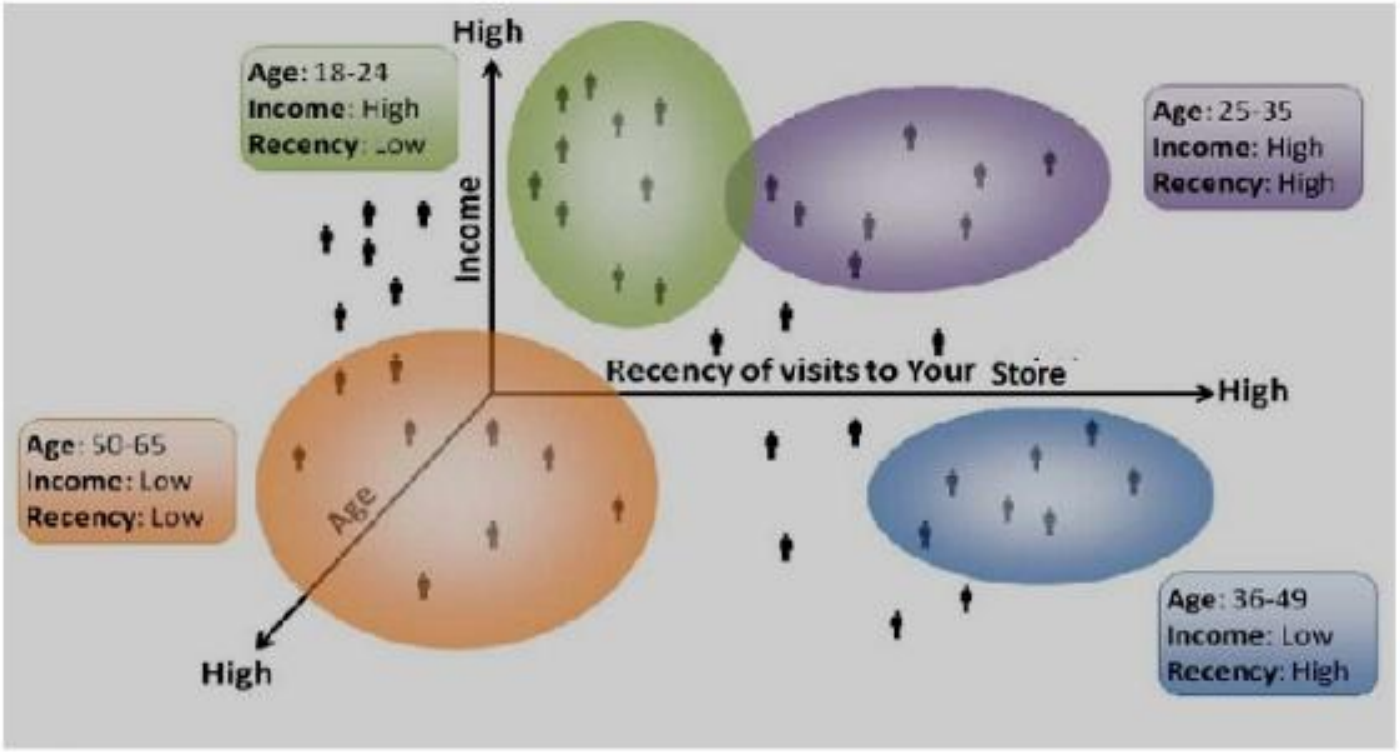
Why: Clustering pada Gen

http://genome-www.stanford.edu/sarcoma/supplemental_data.html



Why: Clustering untuk Segmentasi

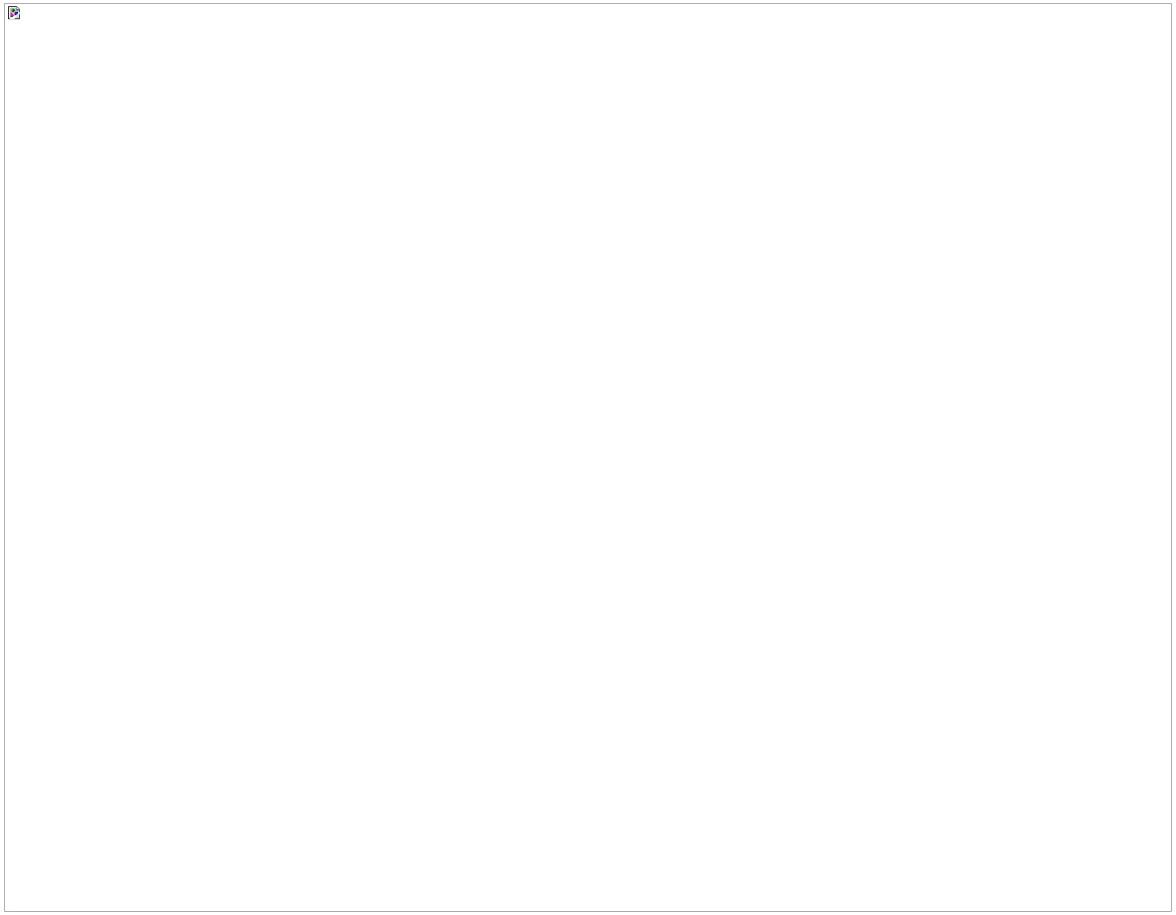
Example - Clusters using Age, Income & Recency



Canvass

https://www.slideshare.net/Canvass/marketing-data-analytics

Why: Clustering-based Approach

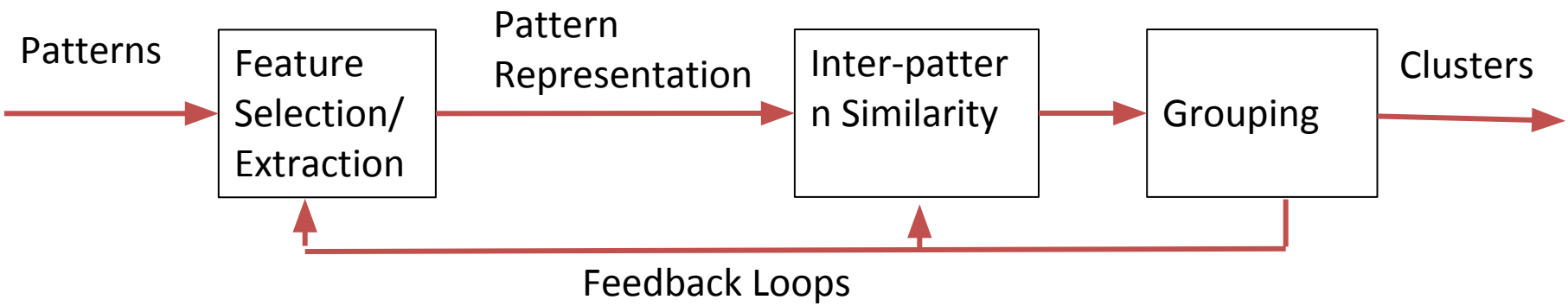


Clustering-based summarization

Clustering-based outlier detection

Clustering-based analysis

Tahapan Clustering



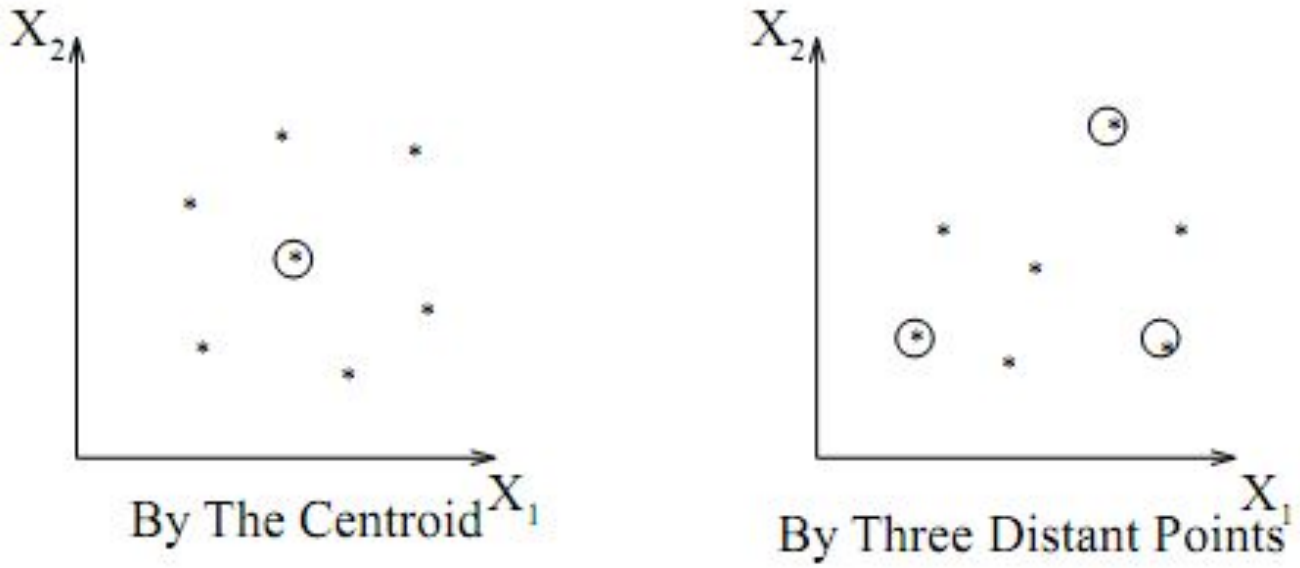
- Tahapan utama:
  1) Feature selection: original features subset of features
     Feature extraction: transformation into new features
  2) pattern proximity/similarity measure
  3) Grouping
- Clustering output: hard atau soft (membership degree)

Tahapan Clustering (lanjutan)

- Tahapan opsional:
  4) data abstraction
  5) assessment of output (good or poor)

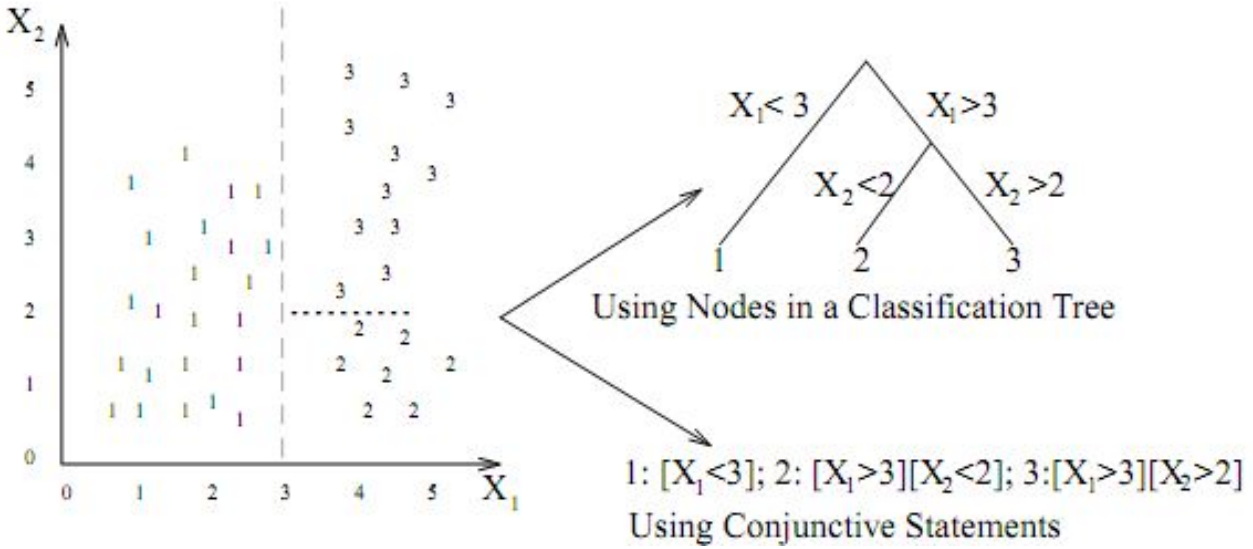
Representasi Cluster (1)

- Centroid atau set of distant point



Sumber: Jain dkk (1999)

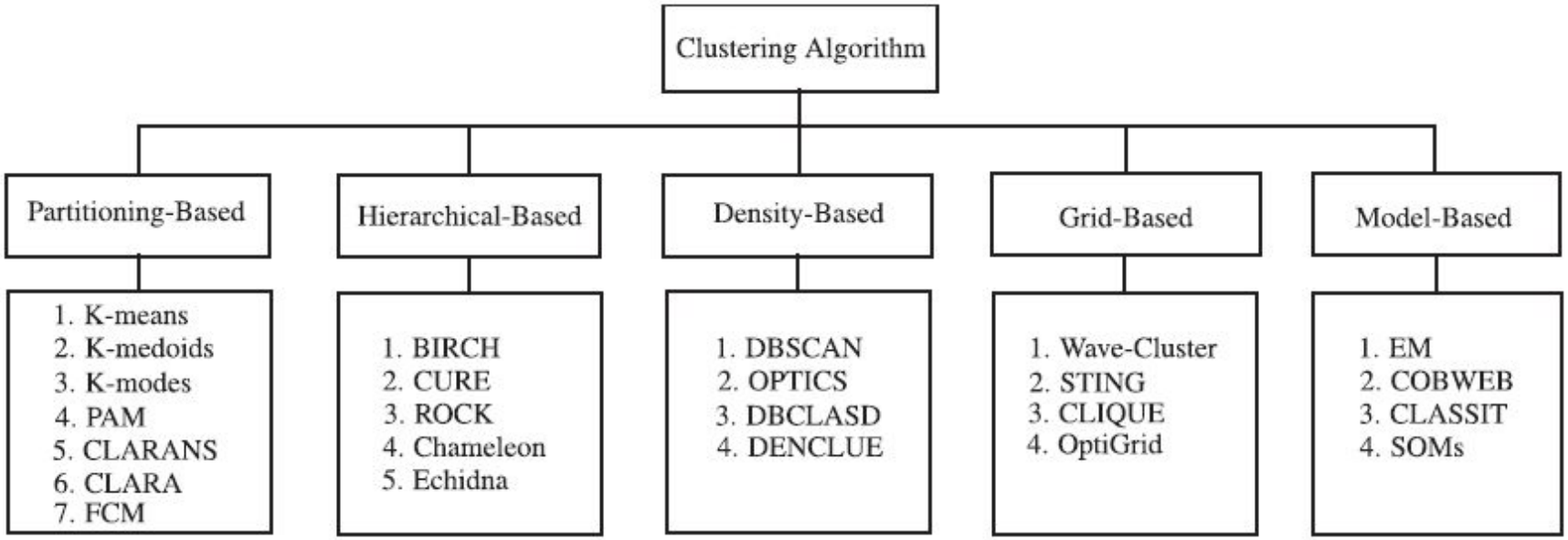
Representasi Cluster (2)



- Pohon klasifikasi
- Conjunctive statements

Sumber: Jain dkk (1999)

Kategori Metode Clustering (Fahad, 2014)



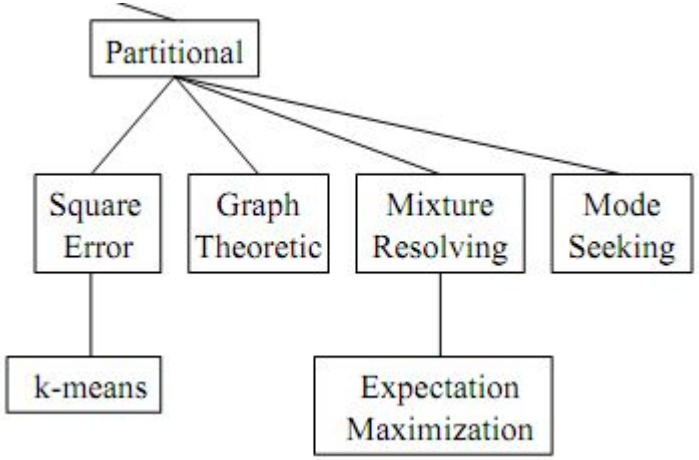
Kategori Metode Clustering
(Han & Kamber, 2006)

1. Metode *partitioning*
   - mengidentifikasi partisi yang mengoptimalkan kriteria pengelompokan (squared error, absolute error)
   - Konstruksi k-partisi data (partisi ~ cluster); k ≤ jumlah data
   - Contoh: K-means, k-medoids
2. Metode *hierarchical*

   menghasilkan rangkaian partisi bersarang
   - Agglomerative (bottom-up, merge):
     1 object ~ 1 cluster 1 cluster n-object
   - Divisive (top-down, split):
     1 cluster n-object 1 object ~ 1 cluster

Kategori Metode Clustering (lanj) (Han & Kamber, 2006)

3. Metode berbasis density
   - Densitas: jumlah objek
   - Contoh: DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
4. Metode berbasis grid
   - Struktur grid, cepat, bergantung jumlah sel, tidak dipengaruhi jumlah objek, perhitungan bisa dilakukan secara paralel
   - Contoh: STING (STatistical INformation Grid)
5. Metode berbasis model
   - Contoh: EM (Expectation-Maximization), SOM (self-organizing map)



Partitioning Methods

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

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

PARTITIONAL CLUSTERING

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)

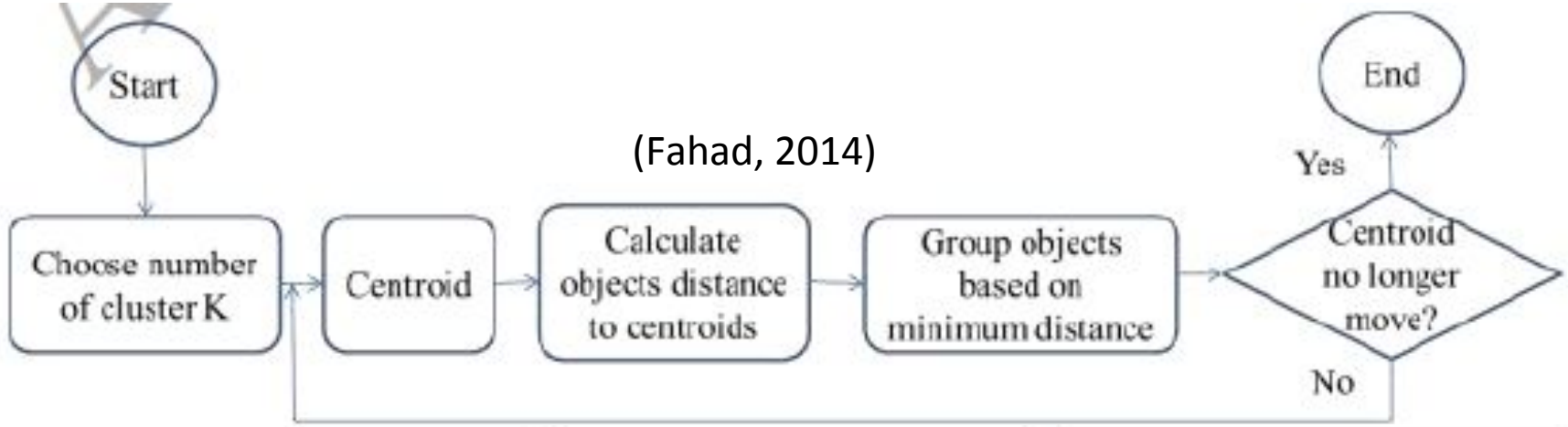
(1) Select an initial partition of the patterns with a fixed number of clusters and cluster centers.

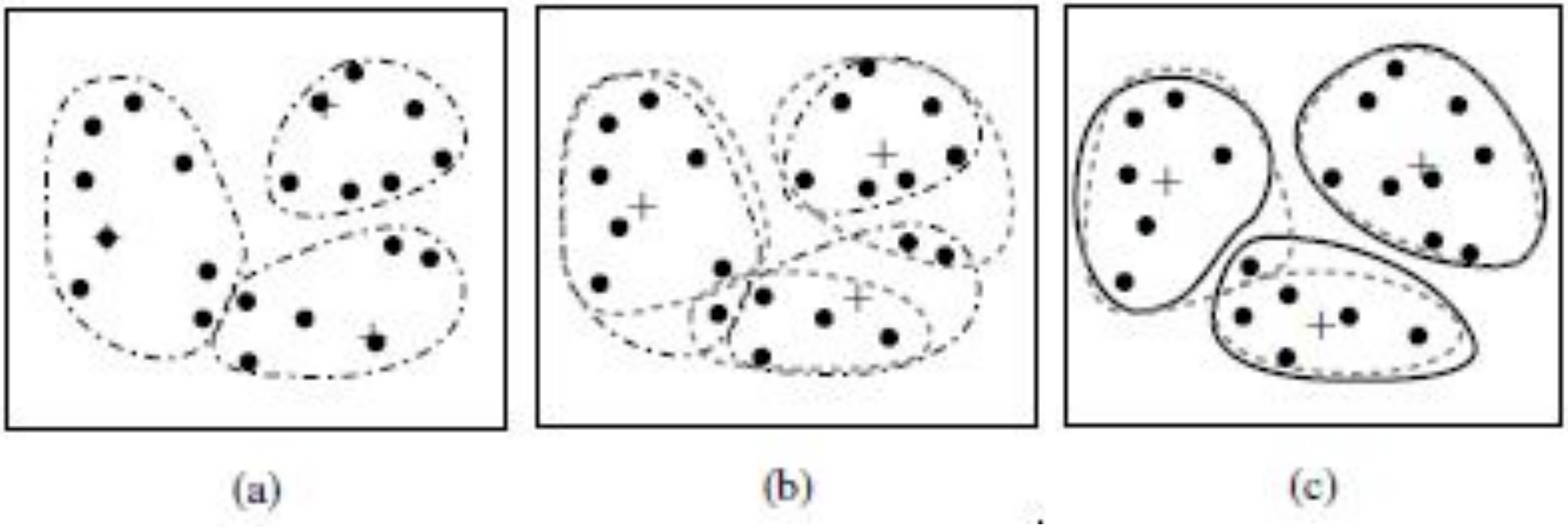
(2) Assign each pattern to its closest cluster center and compute the new cluster centers as the centroids of the clusters. Repeat this step until convergence is achieved, i.e., until the cluster membership is stable.

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

Review: K-Means

(Fahad, 2014)





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~~

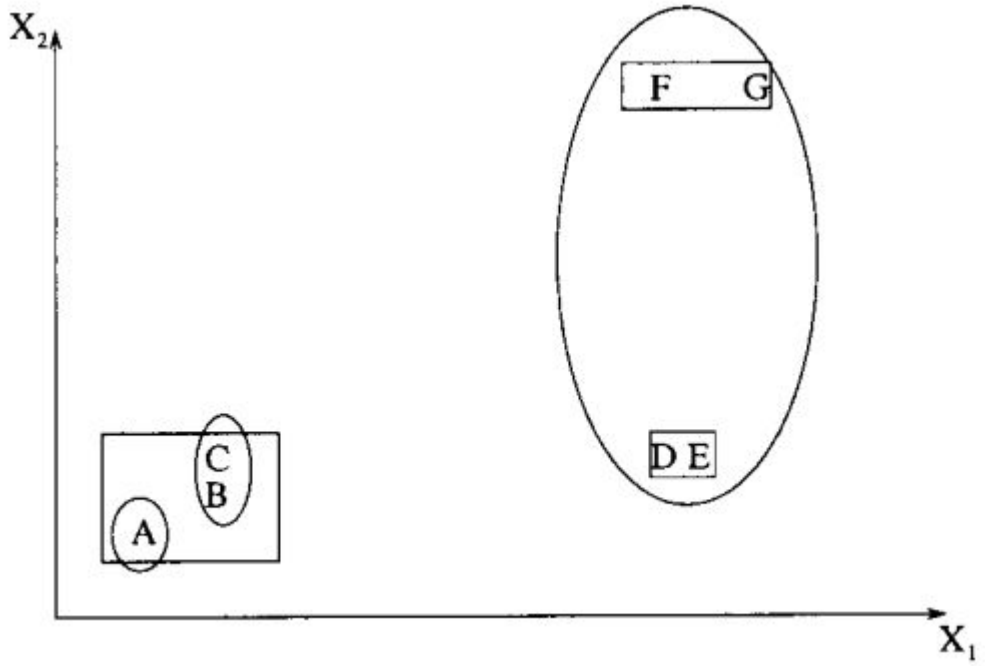
- Method:
  1. Arbitrarily choose k objects from D as the initial cluster centers
  2. Repeat
     - (re) assign each object to the cluster to which the object is the most similar based on the mean value of the objects in the clusters;
     - Update the cluster means, i.e. calculate the mean value of the objects for each cluster;

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



- Cluster awal: A,B,C □ {{A}, {B,C}, {D, E, F, G}}
- Cluster awal: A, D, F □{{A, B, C}, {D, E}, {F, G}}

Sumber: Jain dkk (1999)

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, $\boldsymbol{o}_{random}$;

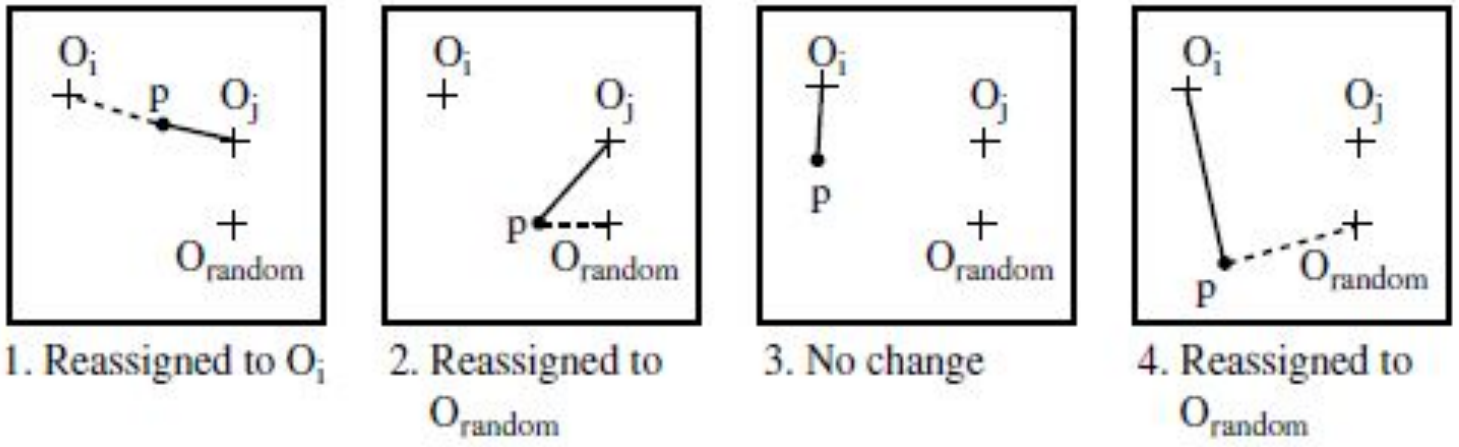
(5) compute the total cost, *S*, of swapping representative object, $\boldsymbol{o}_j$, with $\boldsymbol{o}_{random}$;

(6) if $S < 0$ then swap $\boldsymbol{o}_j$ with $\boldsymbol{o}_{random}$ to form the new set of *k* representative objects;

(7) until no change;

Penjelasan (5)

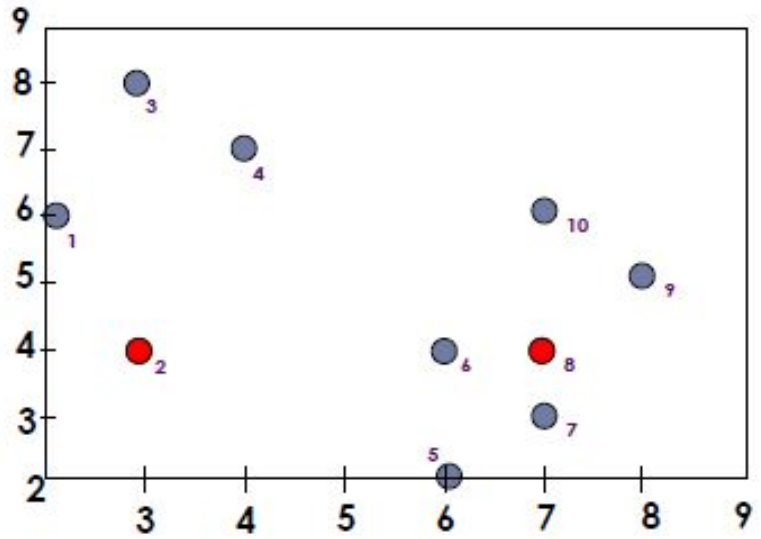
- Compute the total cost, *S*, of swapping representative object, $\boldsymbol{o}_j$, with $\boldsymbol{o}_{\text{random}}$
- S = difference of absolute error
- Oj is representative object of $\boldsymbol{o}_{\text{random}}$
- S is calculated after temporary assignment of each nonrepresentative object 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

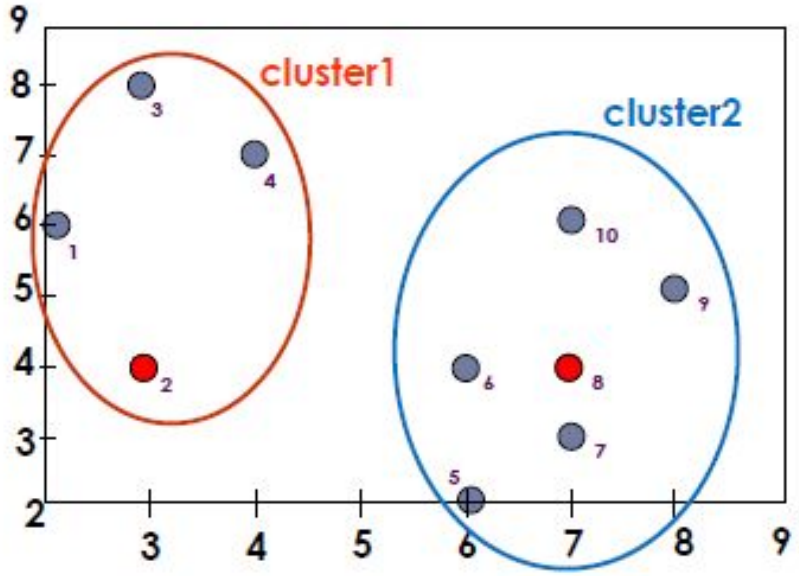
Choose randmly two medoids

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

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

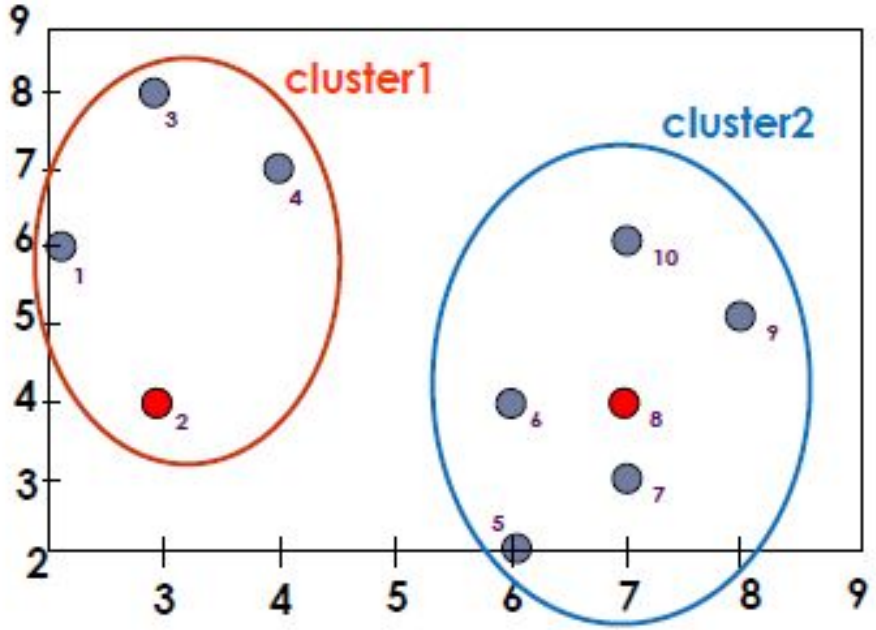
Cluster1 = $\{O_1, O_2, O_3, O_4\}$

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)]**

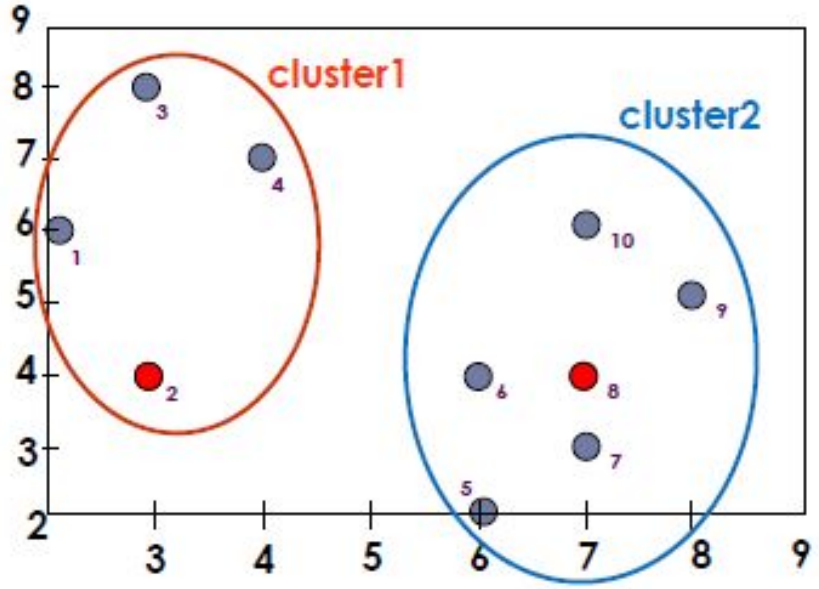
$$E=\sum_{i-1}^{k}\sum_{p\in C_i}|p-o_i|=|o_1-o_2|+|o_3-o_2|+|o_4-o_2|$$

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

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



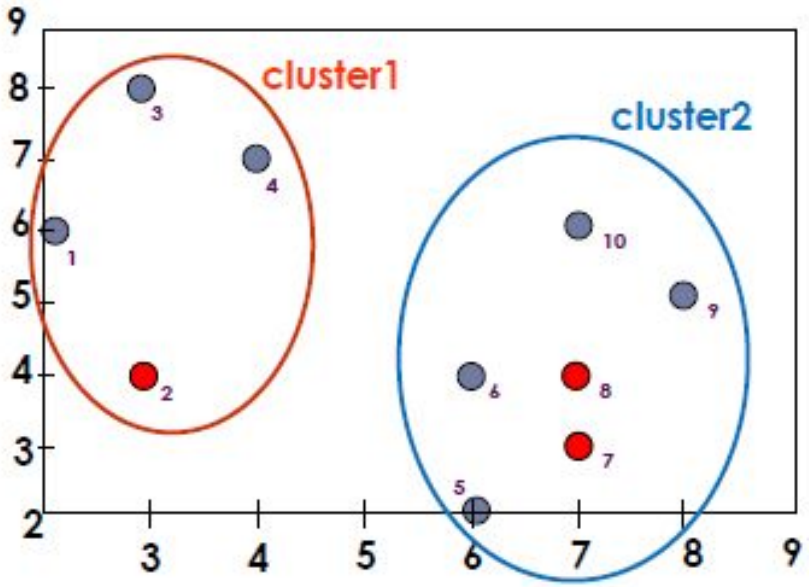
→The absolute error criterion **[for the set of Medoids (O2,O8)]**

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

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

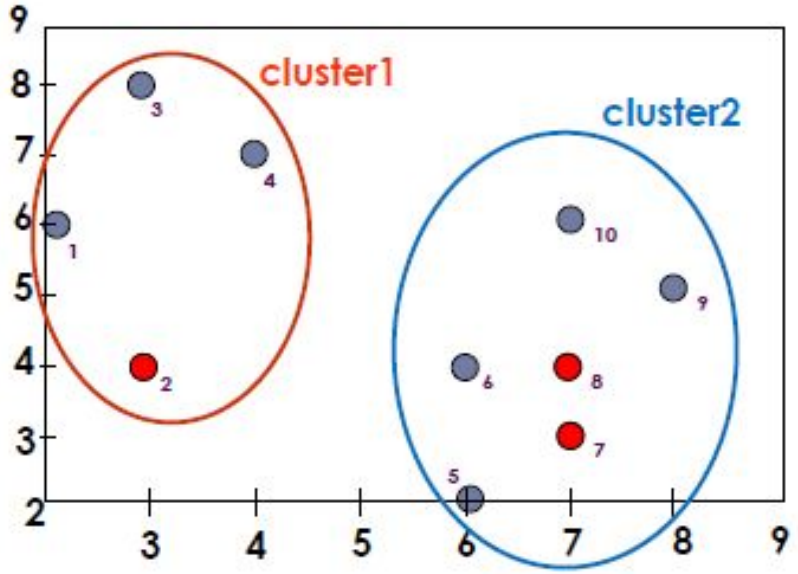
→Compute the absolute error criterion **[for the set of Medoids (O2,O7)]**

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

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

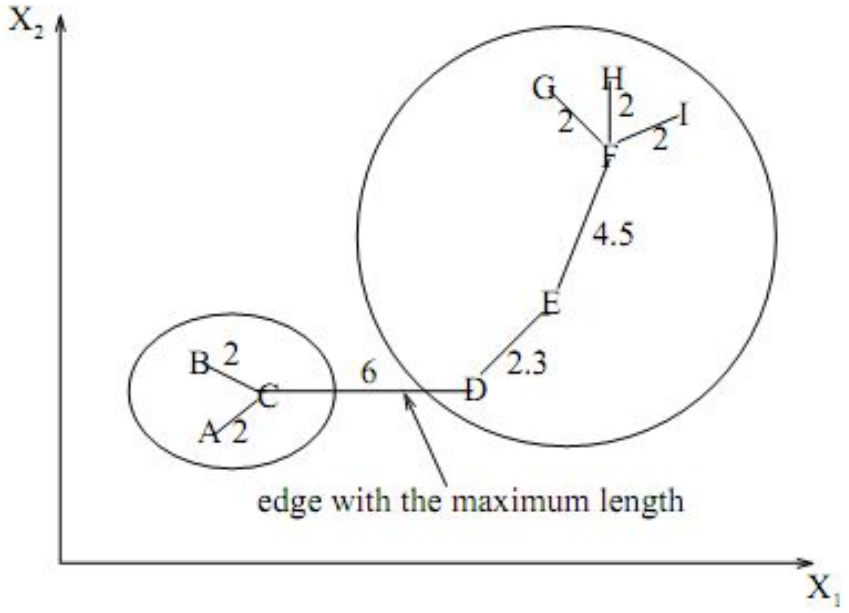


- In this example, changing the medoid of cluster 2 did not change the assignments of objects to clusters.
- What are the possible cases when we replace a medoid by another object?

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

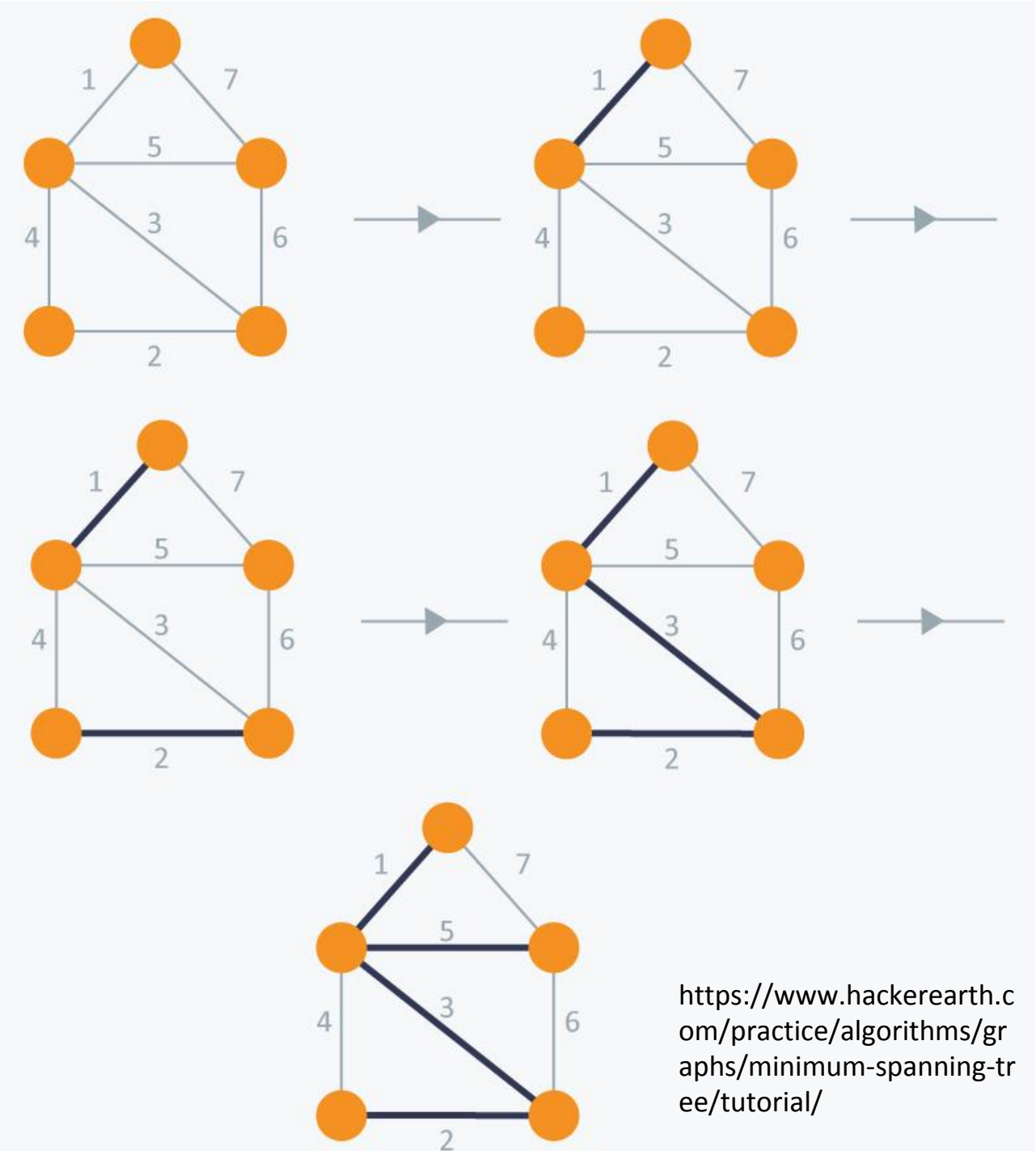


- MST-based clustering construct the minimal spanning tree (MST) of the data
- Delete the MST edges with the largest lengths to generate clusters:

  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).

Referensi Tambahan

- Arthur, D., & Vassilvitskii, S. (2006, June). How slow is the k-means method?. In *Proceedings of the twenty-second annual symposium on Computational geometry* (pp. 144-153). ACM.
- Vattani, A. (2011). K-means requires exponentially many iterations even in the plane. *Discrete & Computational Geometry*, *45*(4), 596-616.
- Bahmani, B., Moseley, B., Vattani, A., Kumar, R., & Vassilvitskii, S. (2012). Scalable k-means++. *Proceedings of the VLDB Endowment*, *5*(7), 622-633.

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.

Minimum square-error: mean

$$g(c) = \sum_{i=1}^{n} (X_i - c)^2$$

g(c) minimum yaitu g’(c)=0 c=mean(X)

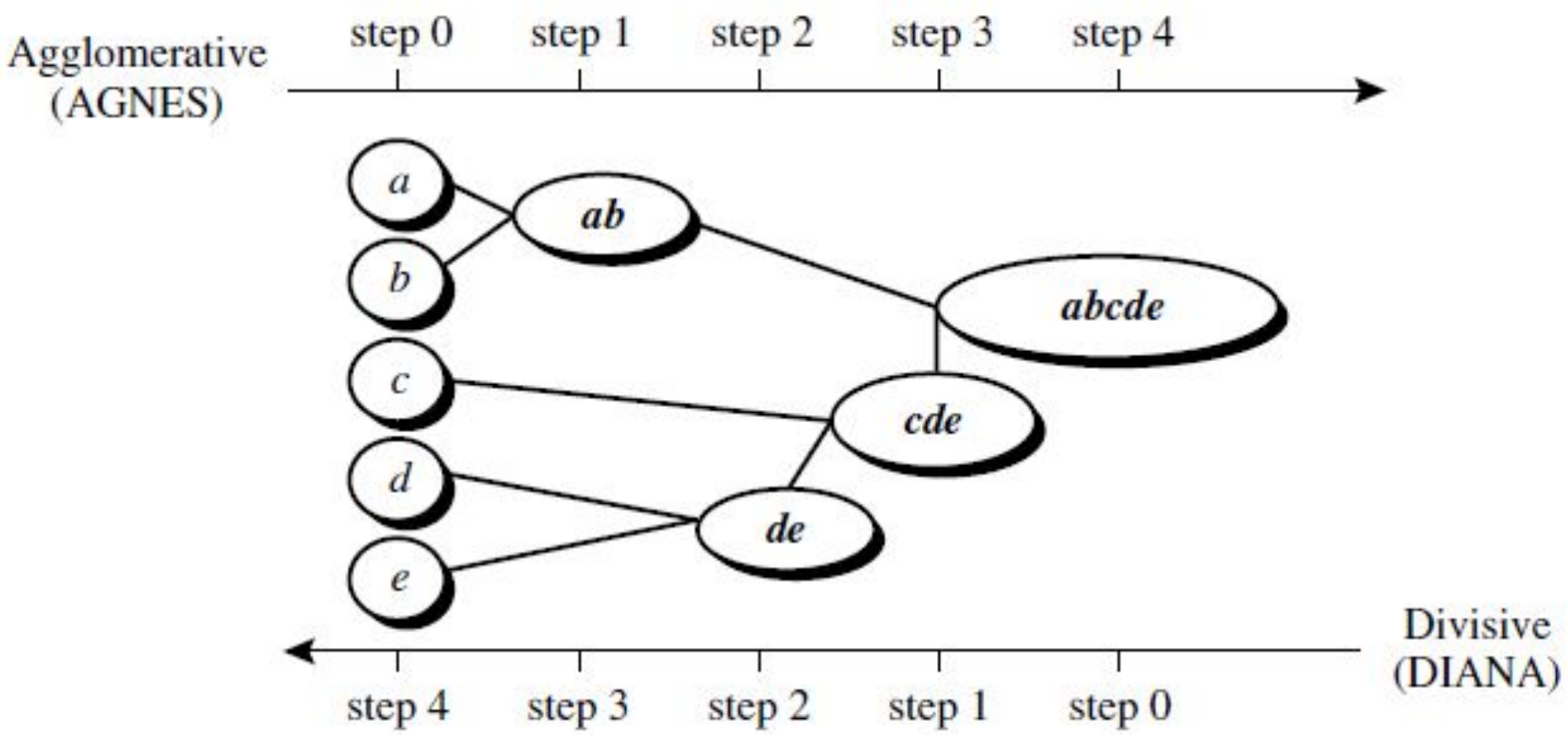
$$0 = g'(c) = \sum_{i=1}^{n} 2(X_i - c)(-1) = 2\left(nc - \sum_{i=1}^{n} X_i\right) \Rightarrow$$

$$c = \frac{1}{n} \sum_{i=1}^{n} X_i = \bar{X}$$

Agglomerative, Divisive

HIERARCHICAL CLUSTERING

Hierarchical Clustering



Han & Kamber (2006)

Agglomerative Clustering Algorithm

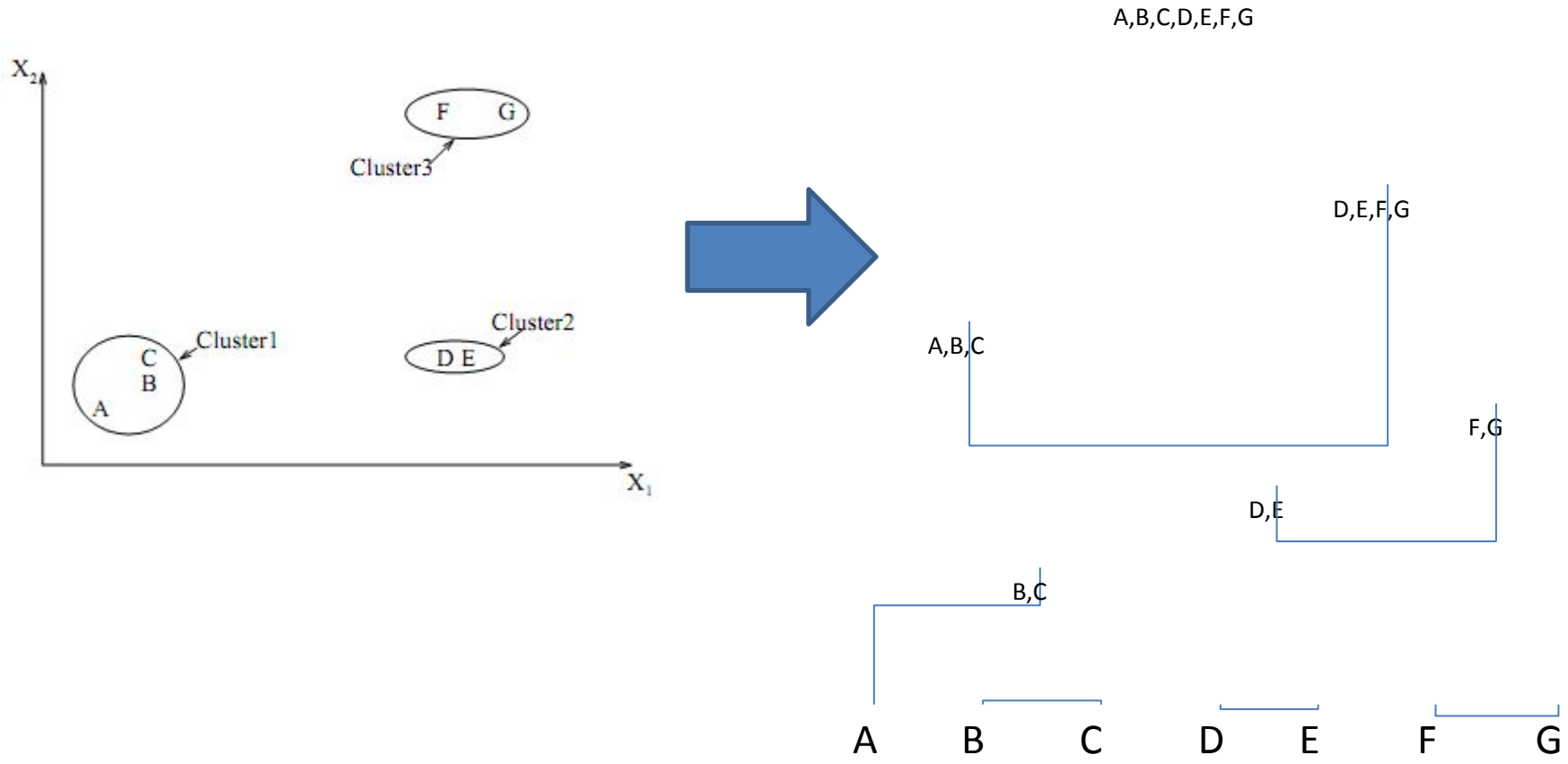
1) Start with N singleton clusters. Calculate the proximity matrix for the N clusters.
2) Search the minimal distance

$$D(C_i, C_j) = \min_{\substack{1 \le m, l \le N \\ m \ne l}} D(C_m, C_l)$$

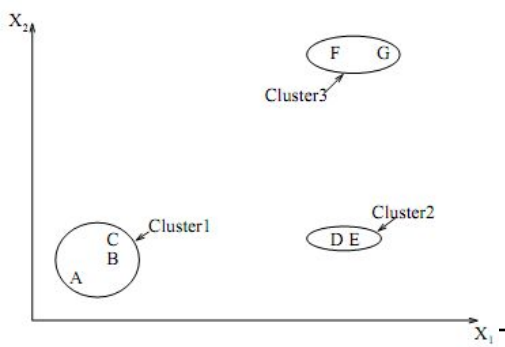
where $D(*, *)$ is the distance function discussed before, in the proximity matrix, and combine cluster C_i and C_j to form a new cluster.
3) Update the proximity matrix by computing the distances between the new cluster and the other clusters.
4) Repeat steps 2)–3) until all objects are in the same cluster.

Sumber: Xu & Wunsch (2005); Jain dkk (1999)

Ilustrasi Agglomerative HC



Ilustrasi Agglomerative HC (lanj)



	A	B	C	D	E	F	G
A							
B			■				
C							
D							
E							
F							
G							

	A	B,C	D	E	F	G
A						
B,C						
D				■		
E						
F						
G						

	A	B,C	D,E	F	G
A					
B,C					
D,E					
F					■
G					

	A	B,C	D,E	F,G
A		■		
B,C				
D,E				
F,G				

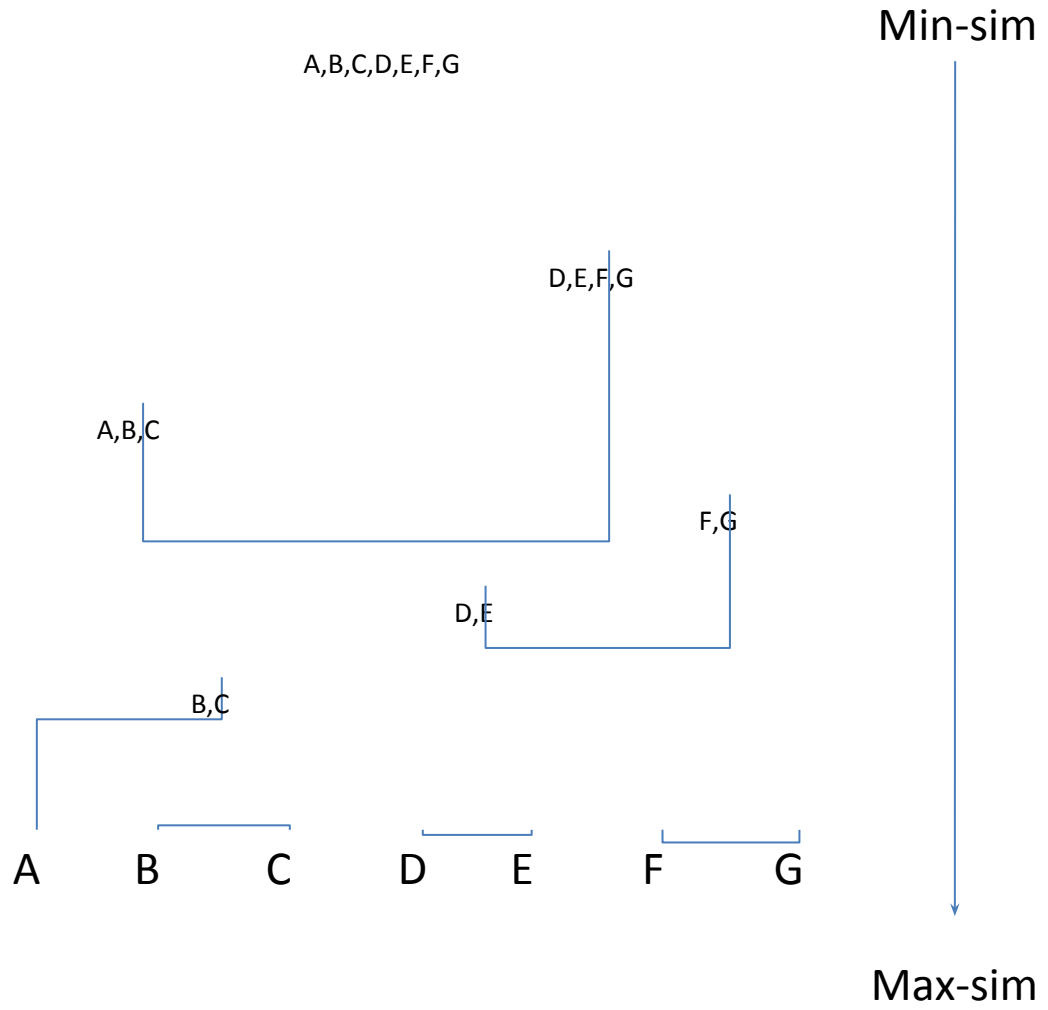
- Iterasi 0: (A),(B),(C),(D),(E),(F),(G)
- Iterasi 1 : (A),(B,C),(D),(E),(F),(G)
- Iterasi 2 : (A),(B,C),(D,E),(F),(G)
- Iterasi 3 : (A),(B,C),(D,E),(F,G)
- Iterasi 4 : (A,(B,C)),(D,E),(F,G)
- Iterasi 5 : (A,(B,C)),((D,E),(F,G))
- Iterasi 6 : ((A,(B,C)),((D,E),(F,G)))

	A, (B,C)	D,E	F,G
A, (B,C)			
D,E			■
F,G			

	A, (B,C)	(D,E), (F,G)
A, (B,C)		■
(D,E), (F,G)		

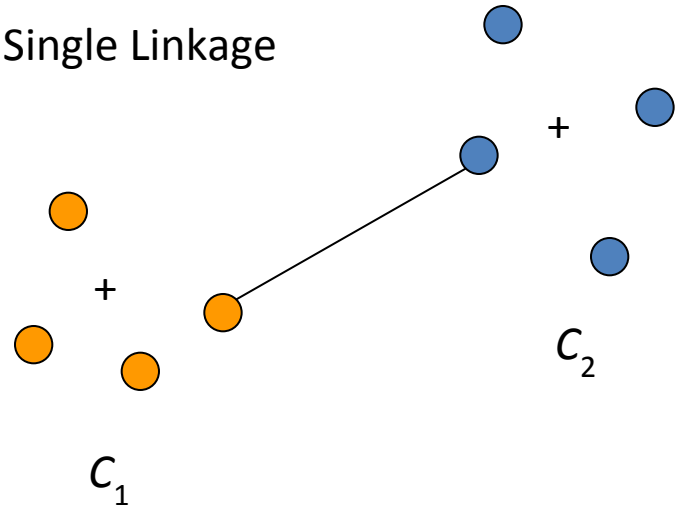
Dendogram

Iterasi 0: (A),(B),(C),(D),(E),(F),(G)
Iterasi 1 : (A),(B,C),(D),(E),(F),(G)
Iterasi 2 : (A),(B,C),(D,E),(F),(G)
Iterasi 3 : (A),(B,C),(D,E),(F,G)
Iterasi 4 : (A,(B,C)),(D,E),(F,G)
Iterasi 5 : (A,(B,C)),((D,E),(F,G))
Iterasi 6 : ((A,(B,C)),((D,E),(F,G)))



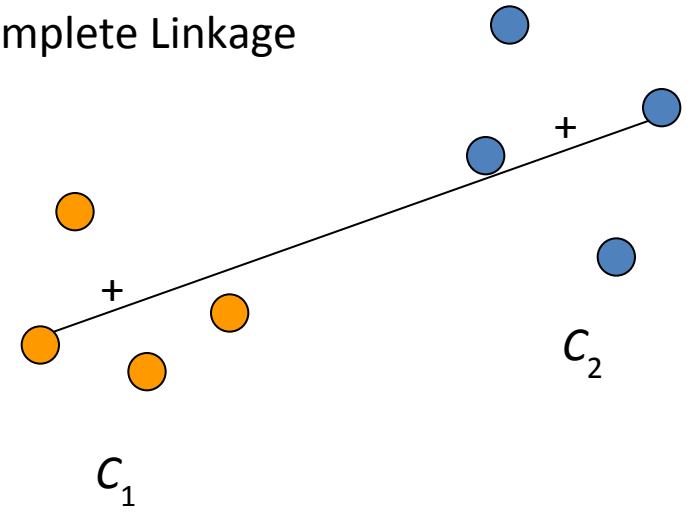
Linkage: Single, Complete, Average, Average Group

Single Linkage



Dissimilarity between two clusters = Minimum dissimilarity between the members of two clusters

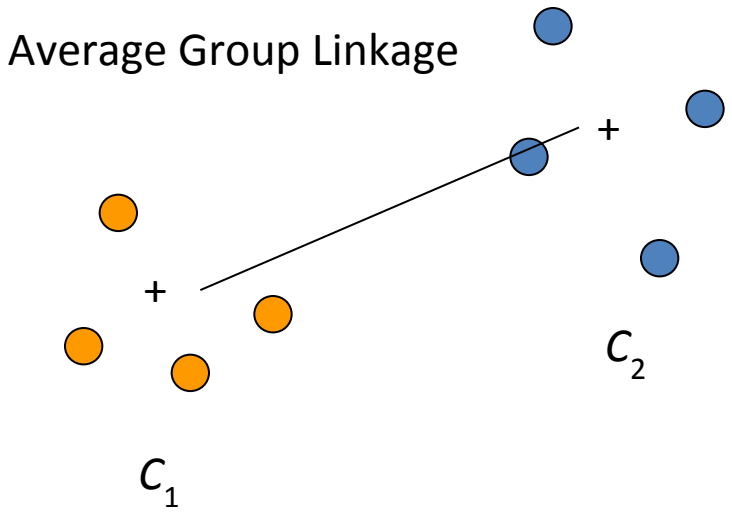
Complete Linkage



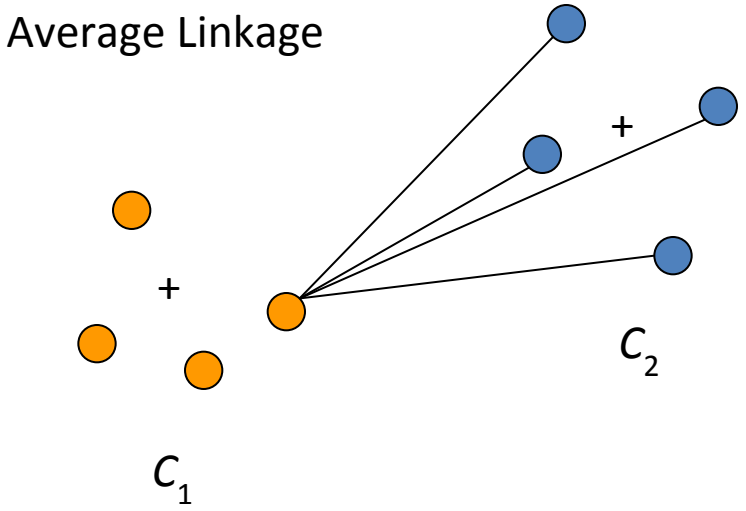
Dissimilarity between two clusters = Maximum dissimilarity between the members of two clusters

Sumber: Hong (2005)

Linkage: Single, Complete, Average, Average Group (lanjutan)



Dissimilarity between two clusters = Distance between two cluster means.



Dissimilarity between two clusters = Averaged distances of all pairs of objects (one from each cluster).

Sumber: Hong (2005)

Single vs Complete-Link

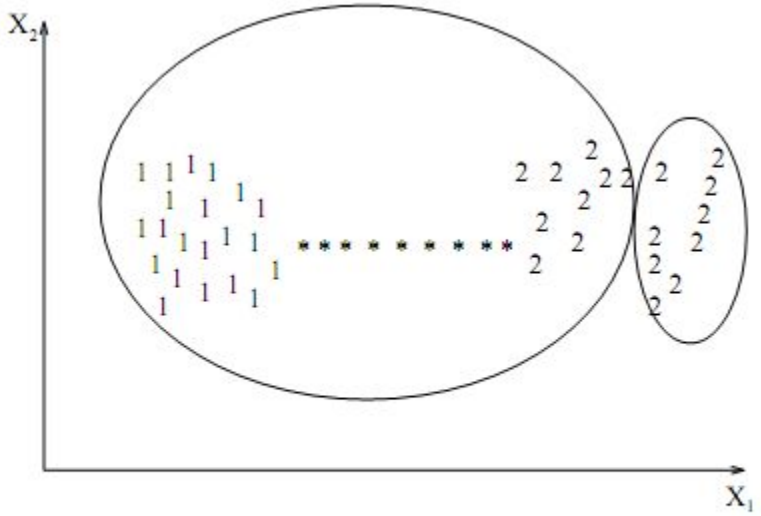


Figure 12. A single-link clustering of a pattern set containing two classes (1 and 2) connected by a chain of noisy patterns (*).

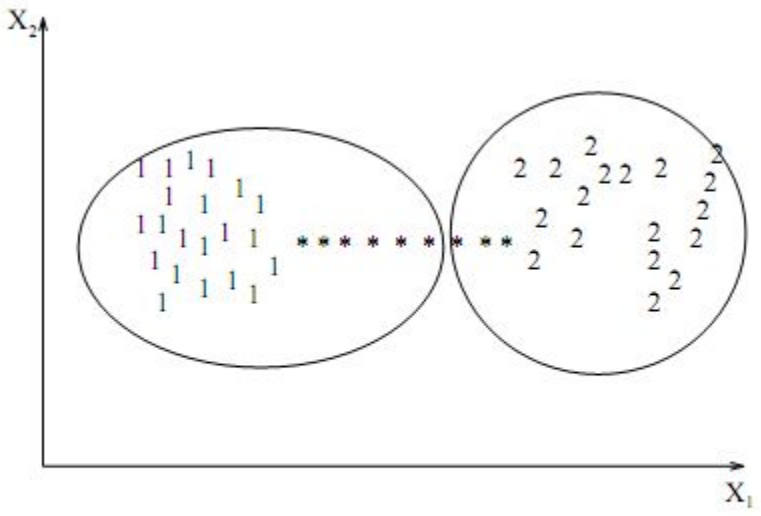


Figure 13. A complete-link clustering of a pattern set containing two classes (1 and 2) connected by a chain of noisy patterns (*).

- Single link clustering suffers from a chaining effect.
- From a pragmatic viewpoint, it has been observed that the complete-link algorithm produces more useful hierarchies in many applications that single-link alg.

Sumber: Jain dkk (1999)

Divisive Clustering

- In the beginning, the entire data set belongs to a cluster and a procedure successively divides it until all clusters are singleton clusters.
- Divisive clustering is not commonly used in practice:
  - For a cluster with N objects, there are $2^{N-1}-1$ possible two-subset divisions, which is very expensive in computation (Xu & Wunsch, 2005).

DBSCAN

DENSITY BASED CLUSTERING

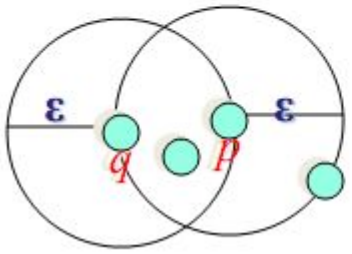
Density-based Clustering

- Clusters are dense regions in the data space, separated by regions of lower object density
- A cluster is defined as a maximal set of density-connected points

ε-Neighborhood – Objects within a radius of ε from an object.

$$N_\varepsilon(p) : \{q \mid d(p,q) \le \varepsilon\}$$

"High density" - ε-Neighborhood of an object contains at least *MinPts* of objects.



ε-Neighborhood of p
ε-Neighborhood of q
Density of p is "high" (MinPts = 4)
Density of q is "low" (MinPts = 4)

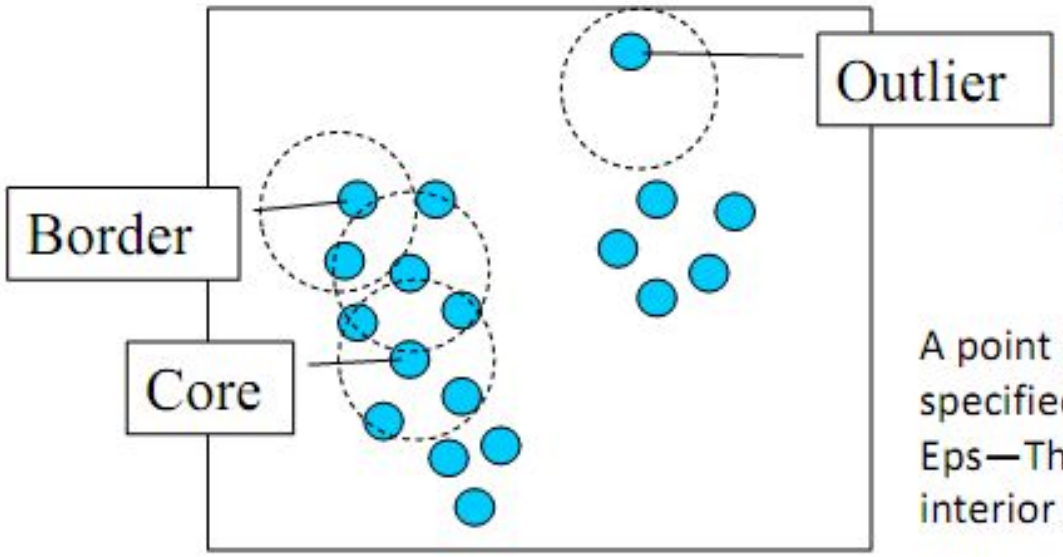
DBSCAN (Density Based Spatial Clustering of Applications with Noise)

- DBSCAN starts with an arbitrary point p and retrieves all points density-reachable from p wrt. Eps and MinPts.
  - If p is a core point, this procedure yields a cluster wrt. Eps and MinPts.
  - If p is a border point, no points are density-reachable from p and DBSCAN visits the next point of the database.

DBSCAN: Object Type

- The neighborhood within a radius ε of a given object is called the **ε-neighborhood** of the object.
- If the ε-neighborhood of an object contains at least a minimum number, *MinPts*, of objects, then the object is called a **core object**.
- Given a set of objects, D, we say that an object $\boldsymbol{p}$ is **directly density-reachable** from object $\boldsymbol{q}$ if $\boldsymbol{p}$ is within the ε-neighborhood of $\boldsymbol{q}$, and $\boldsymbol{q}$ is a core object.
- An object $\boldsymbol{p}$ is **density-reachable** from object $\boldsymbol{q}$ with respect to ε and *MinPts* in a set of objects, D, if there is a chain of objects $\boldsymbol{p_1}, \ldots, \boldsymbol{p_n}$, where $\boldsymbol{p_1} = \boldsymbol{q}$ and $\boldsymbol{p_n} = \boldsymbol{p}$ such that $\boldsymbol{p_{i+1}}$ is directly density-reachable from $\boldsymbol{p_i}$ with respect to ε and *MinPts*, for $1 \leq i \leq n$, $\boldsymbol{p_i} \in D$.
- An object $\boldsymbol{p}$ is **density-connected** to object $\boldsymbol{q}$ with respect to ε and *MinPts* in a set of objects, D, if there is an object $\mathbf{o} \in D$ such that both $\boldsymbol{p}$ and $\boldsymbol{q}$ are density-reachable from $\boldsymbol{o}$ with respect to ε and *MinPts*.

Core, Border, Outlier



Given ε and *MinPts*, categorize the objects into three exclusive groups.

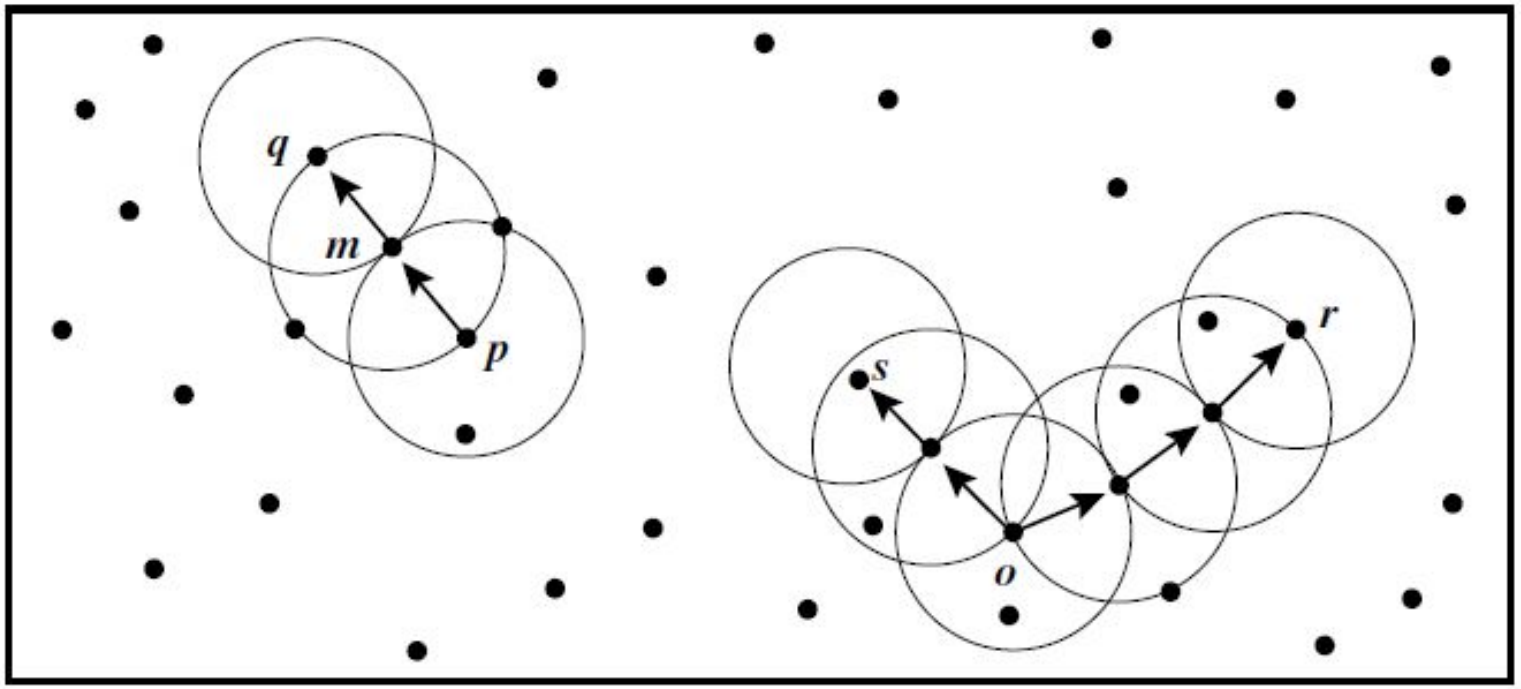
A point is a core point if it has more than a specified number of points (MinPts) within Eps—These are points that are at the interior of a cluster.

$\varepsilon = 1\text{unit}, \text{MinPts} = 5$

A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point.

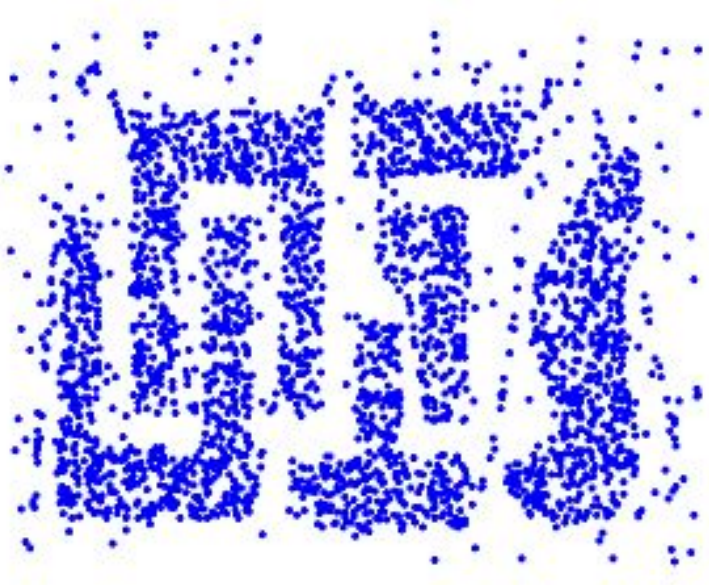
A noise point is any point that is not a core point nor a border point.

DBSCAN find clusters

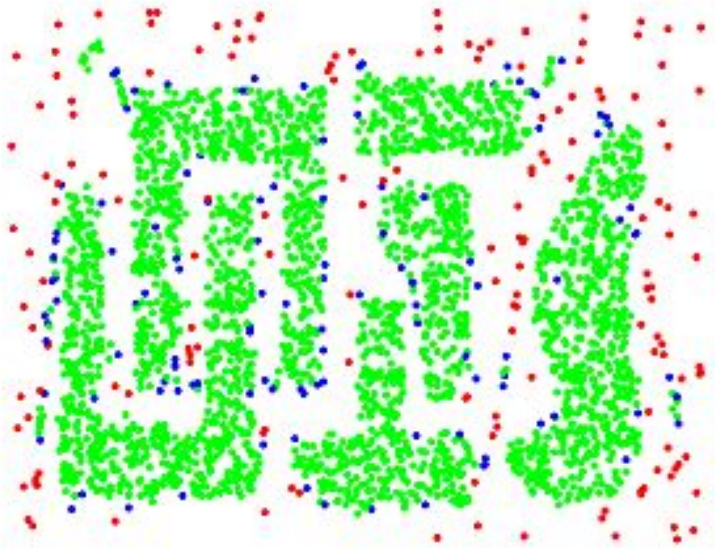


DBSCAN searches for clusters by checking the e-neighborhood of each point in the database. If the e-neighborhood of a point ***p contains*** more than *MinPts, a new cluster with* ***p as a core object is created. DBSCAN then*** iteratively collects directly density-reachable objects from these core objects, which may involve the merge of a few density-reachable clusters. The process terminates when no new point can be added to any cluster.

DBSCAN Example



Original Points



Point types: core, border and outliers

ε = 10, MinPts = 4

DBScan – Example (1)

- If Epsilon is 2 and minpoint is 2, what are the clusters that DBScan would discover with the following 8 examples:

  A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9).

  Matriks jarak (kuadrat):

	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	25	72	13	50	52	65	5
A2	25	0	37	18	25	17	10	20
A3	72	37	0	25	2	4	53	41
A4	13	18	25	0	13	17	52	2
A5	50	25	2	13	0	2	45	25
A6	52	17	4	17	2	0	29	29
A7	65	10	53	52	45	29	0	58
A8	5	20	41	2	25	29	58	0

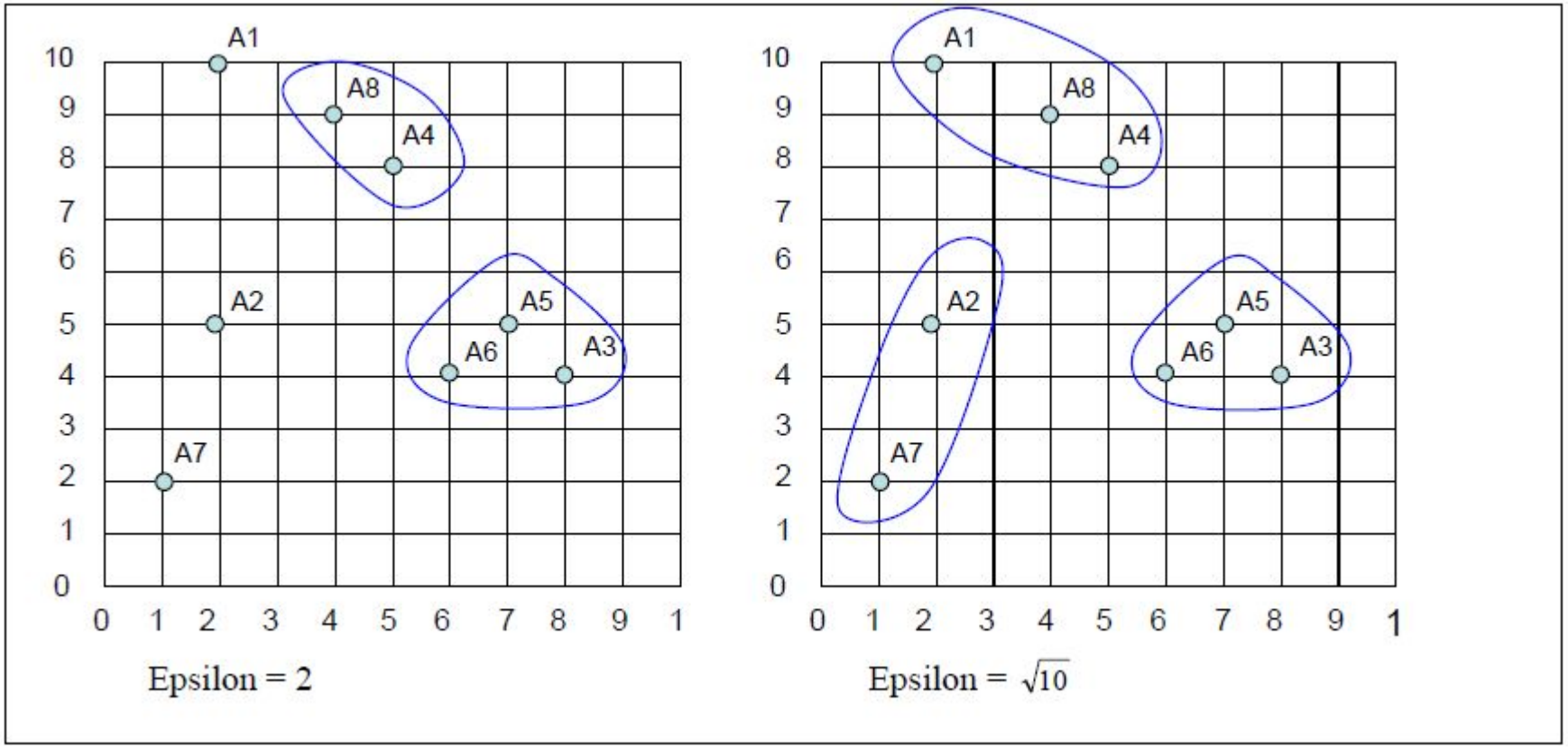
DBScan – Example (2)

- Solutions:
- Epsilon neighborhood of each point
  - N2(A1)={};
  - N2(A2)={};
  - N2(A3)={A5, A6};
  - N2(A4)={A8};
  - N2(A5)={A3, A6};
  - N2(A6)={A3, A5};
  - N2(A7)={};
  - N2(A8)={A4}
  - So A1, A2, and A7 are outliers, while we have two clusters C1={A4, A8} and C2={A3, A5, A6}

	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	25	72	13	50	52	65	5
A2	25	0	37	18	25	17	10	20
A3	72	37	0	25	2	4	53	41
A4	13	18	25	0	13	17	52	2
A5	50	25	2	13	0	2	45	25
A6	52	17	4	17	2	0	29	29
A7	65	10	53	52	45	29	0	58
A8	5	20	41	2	25	29	58	0

- If Epsilon is $10^{1/2}$ then the neighborhood of some points will increase:
  - A1 would join the cluster C1 and A2 would joint with A7 to form cluster C3={A2, A7}.

DBScan – Example (3)

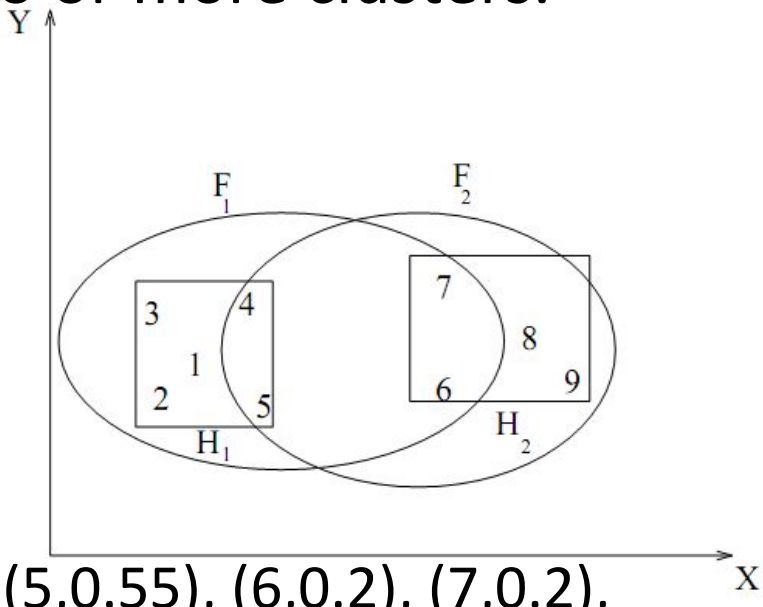


Fuzzy C-Mens

FUZZY CLUSTERING

Fuzzy Clustering

- Fuzzy Clustering is a method of clustering which allows one piece of data to belong to two or more clusters.



- Hard cluster (disjoint):
  - H1={1,2,3,4,5};
  - H2={6,7,8,9}

- Fuzzy Cluster:
  - F1={(1,0.9), (2,0.8), (3,0.7), (4,0.6), (5,0.55), (6,0.2), (7,0.2), (8,0.0), (9,0.0)};
  - F2={(1,0.1), (2,0.2), (3,0.3), (4,0.4), (5,0.45), (6,0.8), (7,0.8), (8,1.0), (9,1.0)}

Sumber: Jain dkk (1999)

Matriks Membership

- F1={(1,0.9), (2,0.8), (3,0.7), (4,0.6), (5,0.55), (6,0.2), (7,0.2), (8,0.0), (9,0.0)};
- F2={(1,0.1), (2,0.2), (3,0.3), (4,0.4), (5,0.45), (6,0.8), (7,0.8), (8,1.0), (9,1.0)}
- $$\begin{bmatrix} 0.9 & 0.8 & 0.7 & 0.6 & 0.55 & 0.2 & 0.2 & 0 & 0 \\ 0.1 & 0.2 & 0.3 & 0.4 & 0.45 & 0.8 & 0.8 & 1 & 1 \end{bmatrix}$$
- $\sum_{i=1}^{c} u_{ik} = 1$ untuk semua data ke-k

Fuzzy c-means (FCM)

- It is based on minimization of the following objective function:

$$J_m(U, v) = \sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik})^m \|y_k - v_i\|_A^2 \qquad \widehat{v_i}^s = \frac{\sum_{k=1}^{N} (u_{ik})^m . y_k}{\sum_{k=1}^{N} (u_{ik})^m}$$

Where:

- N is number of measured data
- c is number of cluster
- m is degree of fuzzy overlap (m > 1)
- u_{ik} is the degree of membership of x_k in the cluster i
- x_k is the ith of d-dimensional measured data
- c_i is the d-dimension center of the cluster
- $||*||_A$ is any A-norm expressing the similarity between any measured data and the center. A: positive definite

Algoritma Fuzzy c-means (Bezdek dkk, 1984)

1. Tentukan c (number of cluster); m (matrix exponent for controlling the degree of fuzzy overlap), dan initial matrix $U^{(0)}$.
2. Hitung means setiap cluster i:

$$\widehat{v_i}^{\,s} = \frac{\sum_{k=1}^{N} (u_{ik})^m \cdot y_k}{\sum_{k=1}^{N} (u_{ik})^m}$$

3. Update membership matrix setiap data k pada setiap cluster i:

$$\widehat{u_{ik}}^{\,s+1} = \left(\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m-1}}\right)^{-1}; \; d_{ik} = ||y_k - v_i||$$

4. Bandingkan $\widehat{u_{ik}}^{\,s+1}$ dengan $\widehat{u_{ik}}^{\,s}$.

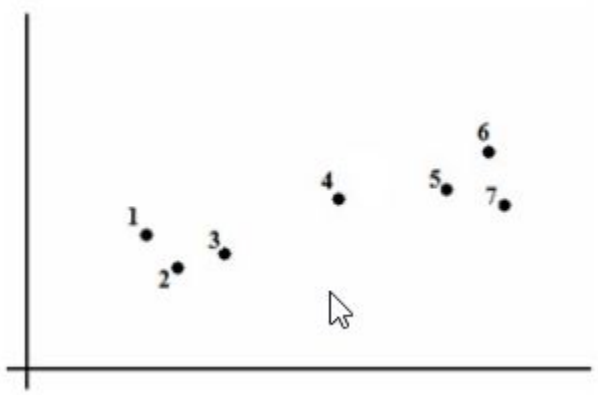
Jika $|\widehat{u_{ik}}^{\,s+1} - \widehat{u_{ik}}^{\,s}| <$ eps, stop.

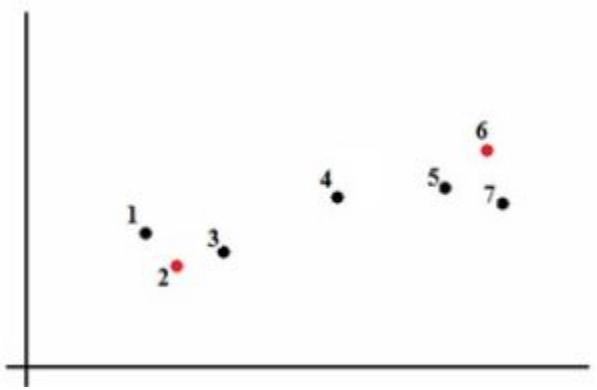
Jika tidak, set $\widehat{u_{ik}}^{\,s} = \widehat{u_{ik}}^{\,s+1}$ dan kembali ke (2).

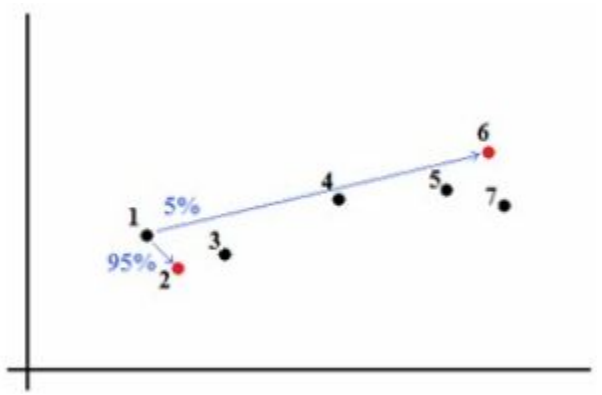
m: Fuzzification Parameter

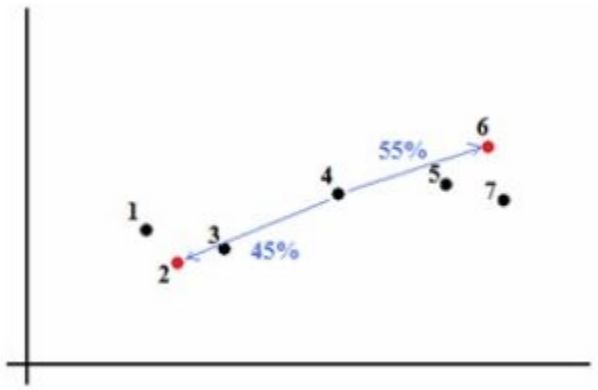
- Range m [1,∞]
- Weighting exponent m controls the relative weights placed on each of the squared errors d^2_{ik}.
- As m -> 1 partitions that minimize Jm become increasingly hard clustering
  - at m = 1, are necessarily hard.
- Each entry of optimal Us for J_m approaches (1/c) as m -> ∞
  - It will blur (defocus) membership towards the fuzziest state.
- No theoretical or computational evidence distinguishes an optimal m.
  - Range of useful values [1, 30].
  - If a test set is available, the best strategy for selecting m at present seems to be experimental.
  - For most data, 1.5 ~< m ~< 3.0 gives good results.

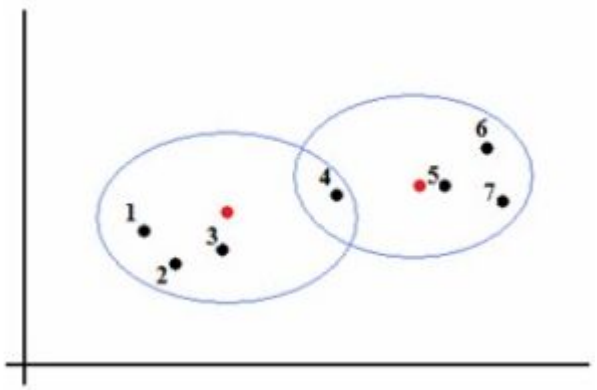
FCM: Contoh 1











FCM: Contoh 2

- X={2,3,4,7,9,10,11}; k=|X|=7; m=2; c=2; eps=0.01
- Inisialisasi $U^{(0)}$ $\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$
- Means: $\widehat{v_i}^{\,s} = \frac{\sum_{k=1}^{N}(u_{ik})^m . y_k}{\sum_{k=1}^{N}(u_{ik})^m}$

- $\widehat{v_{c1}}^{\,0} = \frac{1^2.2+1^2.3+1^2.4+0^2.7+0^2.9+0^2.10+0^2.11}{1^2+1^2+1^2+0^2+0^2+0^2+0^2} = 3$
- $\widehat{v_{c2}}^{\,0} = \frac{0^2.2+0^2.3+0^2.4+1^2.7+1^2.9+1^2.10+1^2.11}{0^2+0^2+0^2+1^2+1^2+1^2+1^2} = \frac{37}{4} = 9.25$

Update membership: $\widehat{u_{ik}}^{s+1} = \left(\sum_{j=1}^{c}\left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m-1}}\right)^{-1}$

- $\widehat{u_{c11}}^{(1)} = \frac{1}{\left(\frac{2-3}{2-3}\right)^2+\left(\frac{2-3}{2-9.25}\right)^2} = \frac{1}{\left(\frac{-1}{-1}\right)^2+\left(\frac{-1}{-7.25}\right)^2} = 0.98$
- $\widehat{u_{c12}}^{(1)} = \frac{1}{\left(\frac{3-3}{3-3}\right)^2+\left(\frac{3-3}{3-9.25}\right)^2} = \frac{1}{\left(\frac{0}{0}\right)^2+\left(\frac{0}{-6.25}\right)^2} = 1$
- $\widehat{u_{c13}}^{(1)} = \frac{1}{\left(\frac{4-3}{4-3}\right)^2+\left(\frac{4-3}{4-9.25}\right)^2} = \frac{1}{\left(\frac{1}{1}\right)^2+\left(\frac{1}{-5.25}\right)^2} = 0.96$
- $\widehat{u_{c14}}^{(1)} = \frac{1}{\left(\frac{7-3}{7-3}\right)^2+\left(\frac{7-3}{7-9.25}\right)^2} = \frac{1}{\left(\frac{4}{4}\right)^2+\left(\frac{4}{-2.25}\right)^2} = 0.24$
- $\widehat{u_{c15}}^{(1)} = \frac{1}{\left(\frac{9-3}{9-3}\right)^2+\left(\frac{9-3}{9-9.25}\right)^2} = \frac{1}{\left(\frac{6}{6}\right)^2+\left(\frac{6}{-0.25}\right)^2} = 0.00$
- $\widehat{u_{c16}}^{(1)} = \frac{1}{\left(\frac{10-3}{10-3}\right)^2+\left(\frac{10-3}{10-9.25}\right)^2} = \frac{1}{\left(\frac{7}{7}\right)^2+\left(\frac{7}{0.75}\right)^2} = 0.01$

Update membership: $\widehat{u_{ik}}^{s+1} = \left(\sum_{j=1}^{c}\left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m-1}}\right)^{-1}$

- $\widehat{u_{c21}}^{(1)} = \frac{1}{\left(\frac{2-9.25}{2-3}\right)^2+\left(\frac{2-9.25}{2-9.25}\right)^2} = \frac{1}{\left(\frac{-7.25}{-1}\right)^2+\left(\frac{-7.25}{-7.25}\right)^2} = 0.02$
- $\widehat{u_{c22}}^{(1)} = \frac{1}{\left(\frac{3-9.25}{3-3}\right)^2+\left(\frac{3-9.25}{3-9.25}\right)^2} = \frac{1}{\left(\frac{-6.25}{0}\right)^2+\left(\frac{-6.25}{-6.25}\right)^2} = 0.00$
- $\widehat{u_{c23}}^{(1)} = \frac{1}{\left(\frac{4-9.25}{4-3}\right)^2+\left(\frac{4-9.25}{4-9.25}\right)^2} = \frac{1}{\left(\frac{-5.25}{1}\right)^2+\left(\frac{-5.25}{-5.25}\right)^2} = 0.04$
- $\widehat{u_{c24}}^{(1)} = \frac{1}{\left(\frac{7-9.25}{7-3}\right)^2+\left(\frac{7-9.25}{7-9.25}\right)^2} = \frac{1}{\left(\frac{-2.25}{4}\right)^2+\left(\frac{-2.25}{-2.25}\right)^2} = 0.76$
- $\widehat{u_{c25}}^{(1)} = \frac{1}{\left(\frac{9-9.25}{9-3}\right)^2+\left(\frac{9-9.25}{9-9.25}\right)^2} = \frac{1}{\left(\frac{-0.25}{6}\right)^2+\left(\frac{-0.25}{-0.25}\right)^2} = 1.00$
- $\widehat{u_{c26}}^{(1)} = \frac{1}{\left(\frac{10-9.25}{10-3}\right)^2+\left(\frac{10-9.25}{10-9.25}\right)^2} = \frac{1}{\left(\frac{0.75}{7}\right)^2+\left(\frac{0.75}{0.75}\right)^2} = 0.99$

Compare $\widehat{u_{ik}}^{s+1}$ dengan $\widehat{u_{ik}}^{s}$

- $U^{(0)} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$
- $U^{(1)} \begin{bmatrix} 0.98 & 1 & 0.96 & 0.24 & 0 & 0.01 & 0.05 \\ 0.02 & 0 & 0.04 & 0.76 & 1 & 0.99 & 0.95 \end{bmatrix}$
- Kembali menghitung $\widehat{v_i}^{(1)}$ dan $\widehat{u_{ik}}^{(2)}$ sampai $|\widehat{u_{ik}}^{s+1} - \widehat{u_{ik}}^{s}| < eps$

FCM: Contoh 3 (m=2)

x	y	c1	c2
1	6	0.8	0.2
2	5	0.9	0.1
3	8	0.7	0.3
4	4	0.3	0.7
5	7	0.5	0.5
6	9	0.2	0.8

FCM: Contoh 3 (m=2)

x	y	c1	c2
1	6	0.8	0.2
2	5	0.9	0.1
3	8	0.7	0.3
4	4	0.3	0.7
5	7	0.5	0.5
6	9	0.2	0.8

step 1

x	y	c1	c2
1	6	0.89	0.11
2	5	0.89	0.11
3	8	0.55	0.45
4	4	0.56	0.44
5	7	0.00	1.00
6	9	0.21	0.79

Perhitungan centroid v

x	y	c1	c2	u1k^2	u2k^2	x1'	y1'	x2'	y2'
1	6	0.8	0.2	0.64	0.04	0.64	3.84	0.04	0.24
2	5	0.9	0.1	0.81	0.01	1.62	4.05	0.02	0.05
3	8	0.7	0.3	0.49	0.09	1.47	3.92	0.27	0.72
4	4	0.3	0.7	0.09	0.49	0.36	0.36	1.96	1.96
5	7	0.5	0.5	0.25	0.25	1.25	1.75	1.25	1.75
6	9	0.2	0.8	0.04	0.64	0.24	0.36	3.84	5.76
				2.32	1.52	5.58	14.28	7.38	10.5
				centroid v		**2.41**	**6.16**	**4.86**	**6.89**

Hitung Jarak dan Update uik(c1,c2)

x	y	c1	c2	d1k	d2k
1	6	0.89	0.11	1.41	3.96
2	5	0.89	0.11	1.22	3.43
3	8	0.55	0.45	1.94	2.16
4	4	0.56	0.44	2.68	3.02
5	7	0.00	1.00	2.73	0.18
6	9	0.21	0.79	4.58	2.40

EM

MODEL-BASED CLUSTERING

Preliminaries

- We assume that the dataset *X* has been generated by a *parametric* distribution *p(X)*.
- Estimation of the parameters of *p* is known as *density estimation*.
- We consider Gaussian distribution.

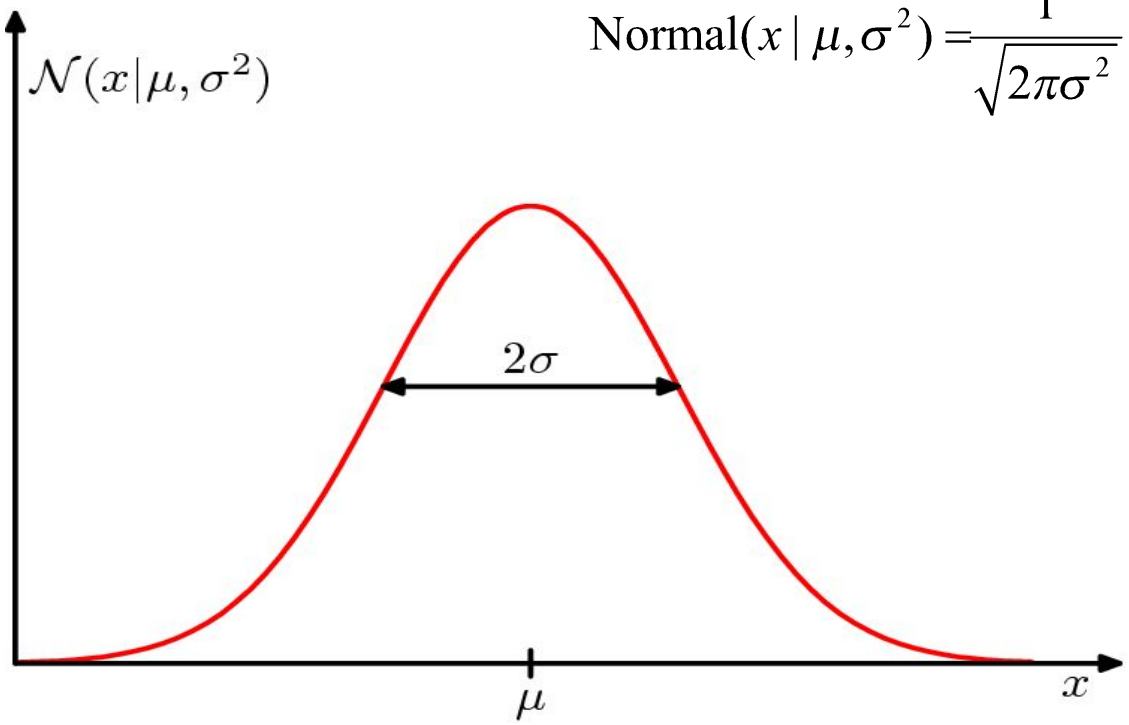
Figures taken from:
http://research.microsoft.com/~cmbishop/PRML/

Typical parameters

- *Mean* (μ): average value of $p(X)$, also called expectation.
- *Variance* (σ): provides a measure of variability in $p(X)$ around the mean.
- *Covariance*: measures how much two variables vary together.
- *Covariance matrix*: collection of co-variances between all dimensions.
  - Diagonal of the covariance matrix contains the variances of each attribute.

One-dimensional Gaussian

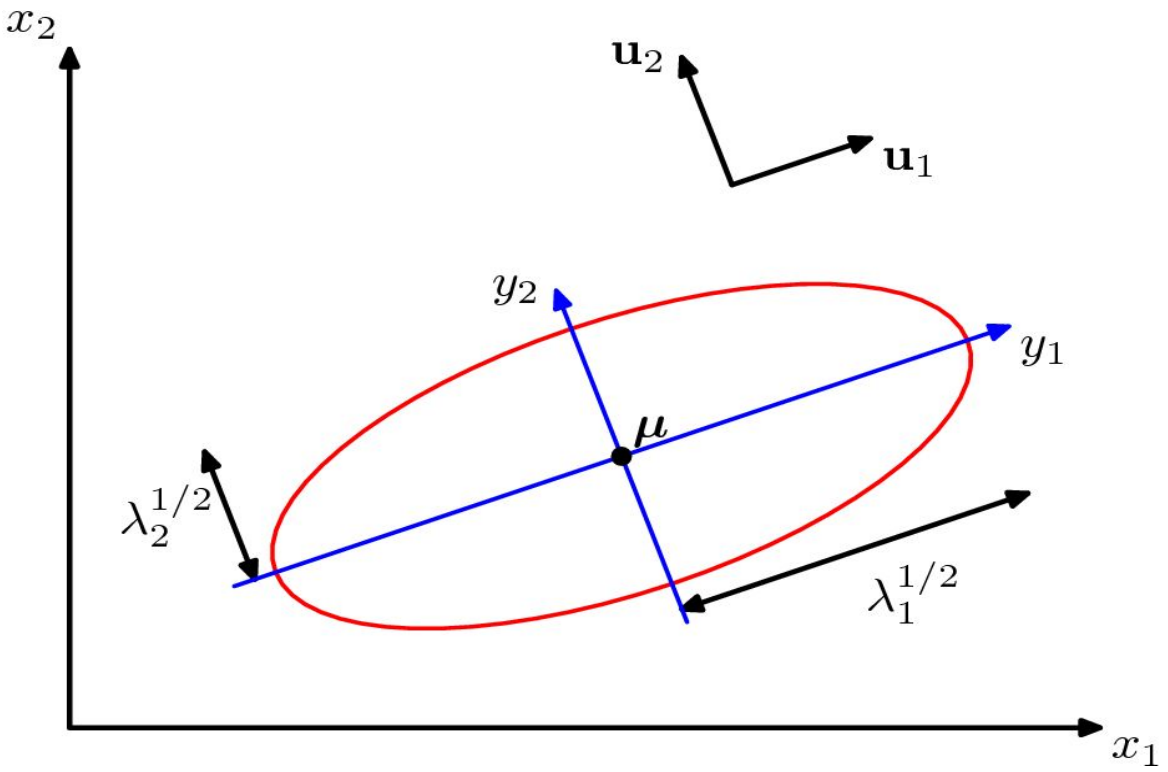
$$\text{Normal}(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2}(x-\mu)^2 \right\}$$



- Parameters to be estimated are the mean (μ) and variance (σ)

Multivariate Gaussian (1)

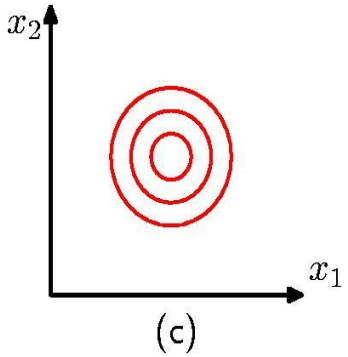
$$\text{Normal}(\mathbf{x} \mid \mu, \Sigma) = \frac{1}{(2\pi)^2} \frac{1}{\det(\Sigma)^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma (\mathbf{x}-\mu)\right\}$$



- In multivariate case we have covariance matrix instead of variance

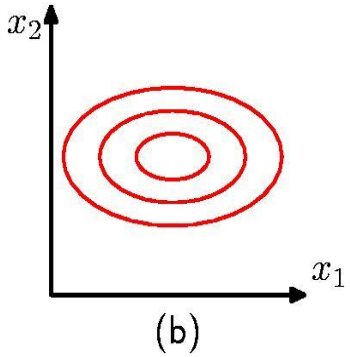
Multivariate Gaussian (2)

Single



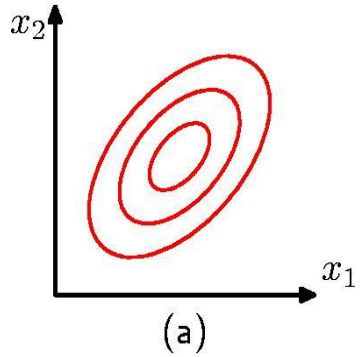
$$\Sigma = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$$

Diagonal



$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$$

Full covariance



$$\Sigma = \begin{pmatrix} \sigma_{11}^2 & \sigma_{12}^2 \\ \sigma_{12}^2 & \sigma_{22}^2 \end{pmatrix}$$

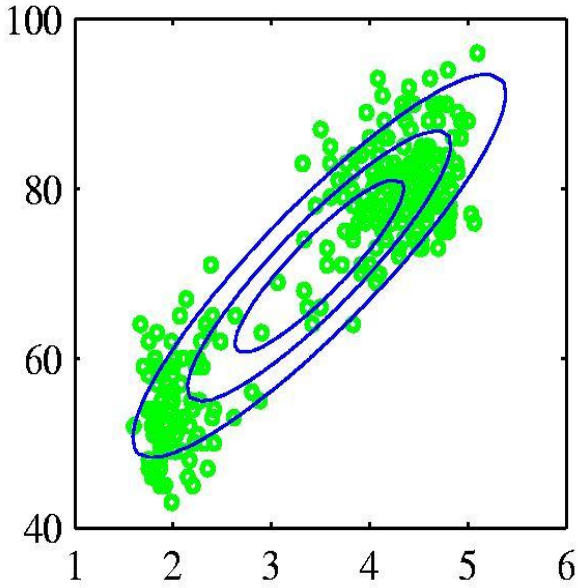
Complete data log likelihood:

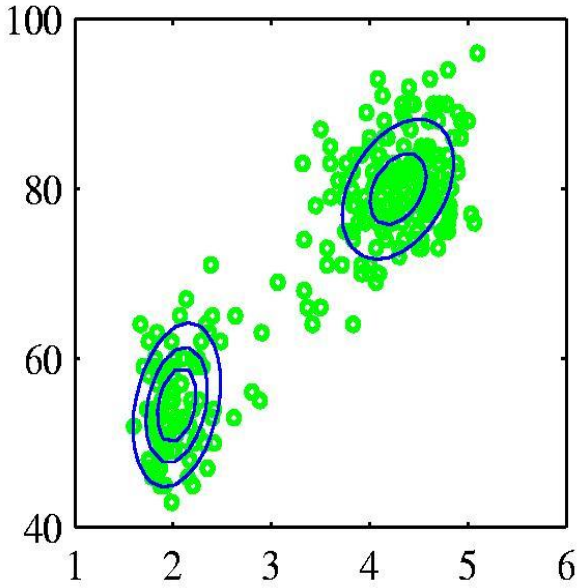
$$\ln p(X) = \ln \prod_{n=1}^{N} \text{Normal}(\mathbf{x}_n \mid \mu, \Sigma)$$

Maximum Likelihood (ML) parameter estimation

- Maximize the log likelihood formulation
- Setting the gradient of the complete data log likelihood to zero we can find the closed form solution.
  - Which in the case of mean, is the sample average.

When one Gaussian is not enough

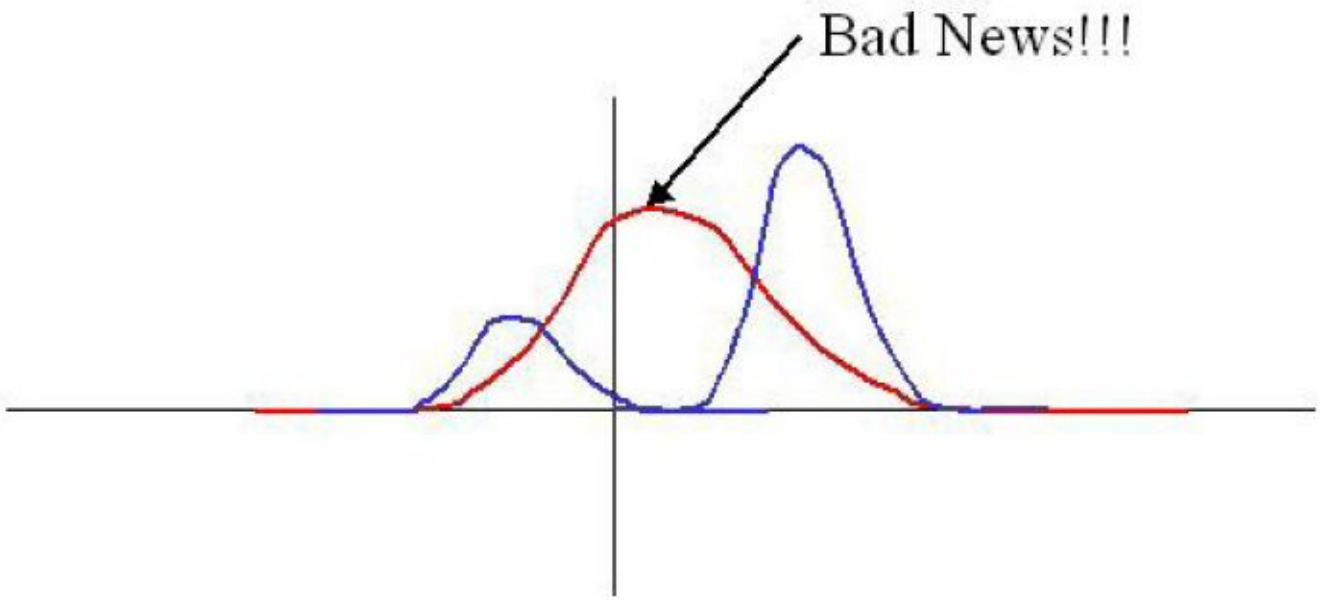




- Real world datasets are rarely unimodal!

Is it good enough model?

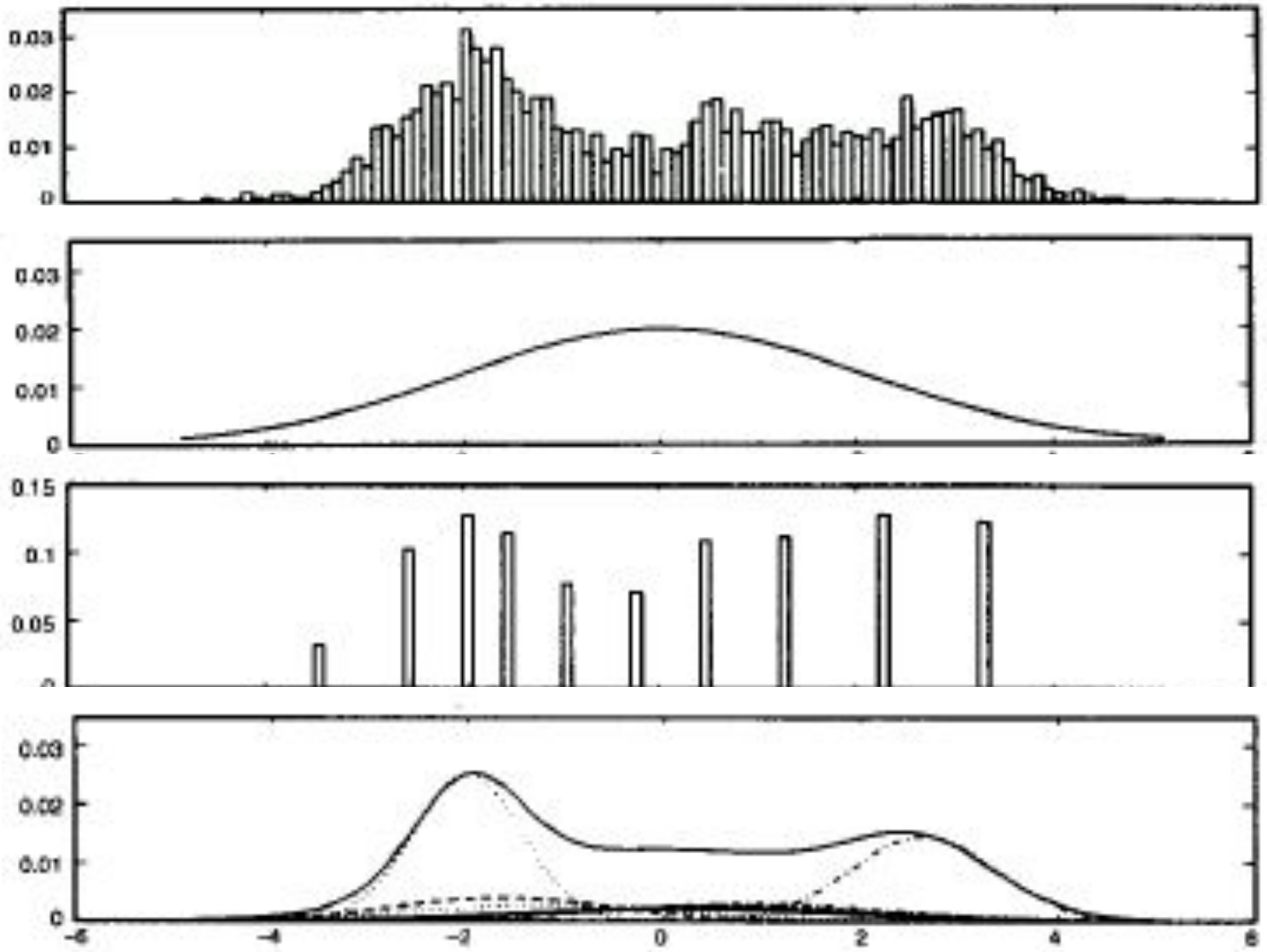
- Single Gaussian may do a bad job of modeling distribution in any dimension:



- Solution: Mixtures of Gaussians

Figure from Chen, Picheney et al slides

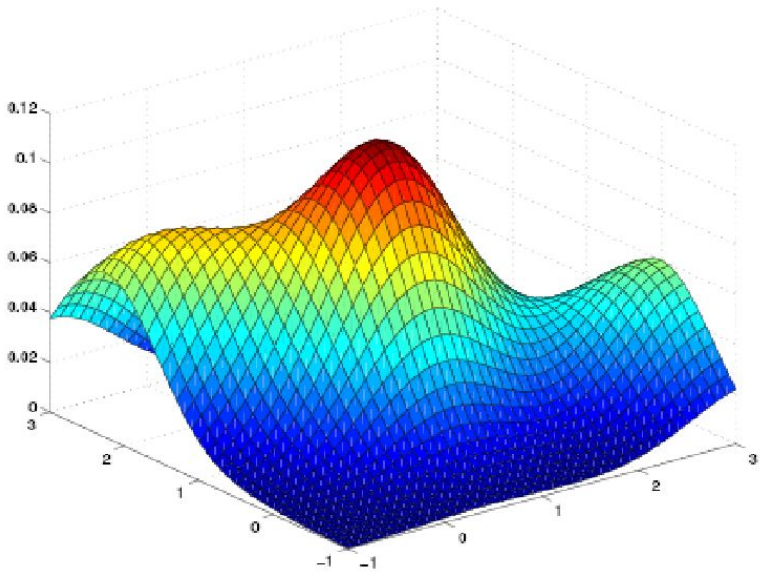
GMM gives the arbitrarily-shaped densities a better approximation.



GMM : Why

- Pros :
  - probabilistic framework (robust)
  - computationally efficient
  - easily to be implemented

Gaussian Mixture Models (GMM)

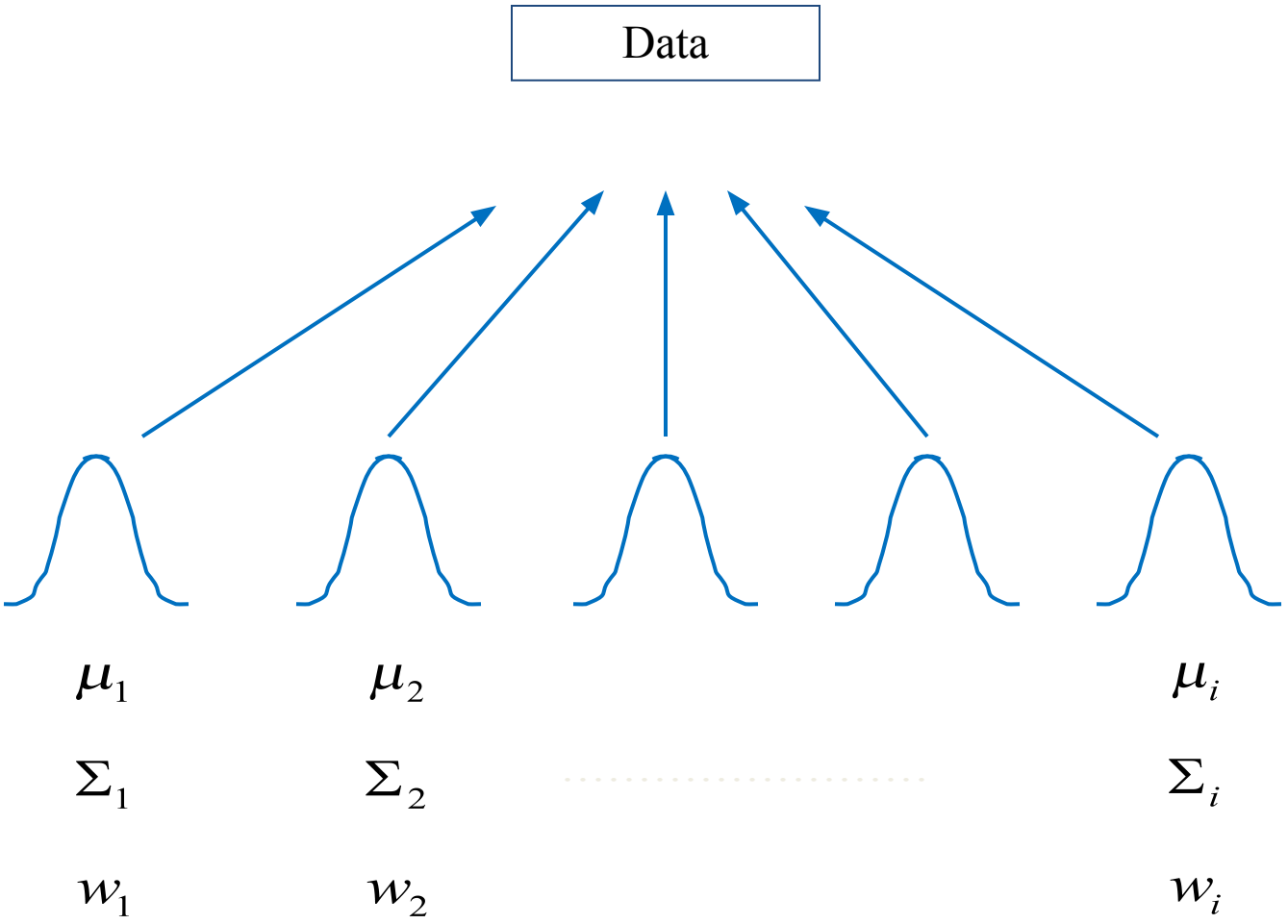


- Weighted sum of N Gaussians:

$$p(x) = \sum_{i=1}^{N} w_i \mathcal{N}(x, \mu_i, \Sigma_i)$$

- Can model arbitrary densities.
- Complexity increases *linearly* with N

Each Gaussian component models a cluster



ML Parameter Estimation : Basic Idea

Step:

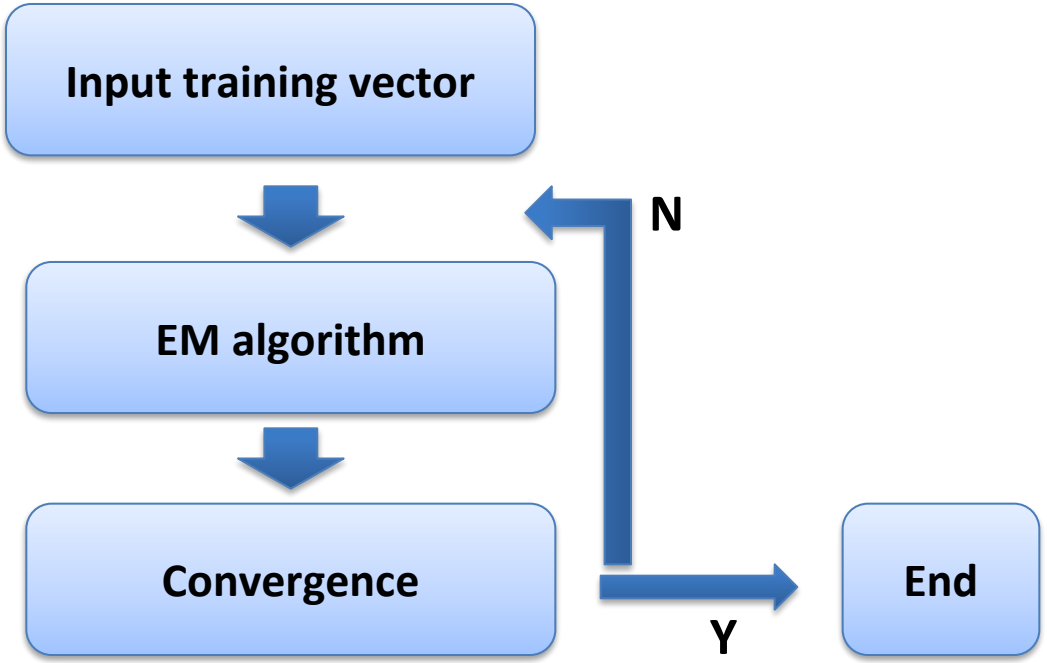
1. Beginning with an initial model λ
2. Estimate a new model $\bar{\lambda}$ such that

$$p(X \mid \bar{\lambda}) \geq p(X \mid \lambda)$$

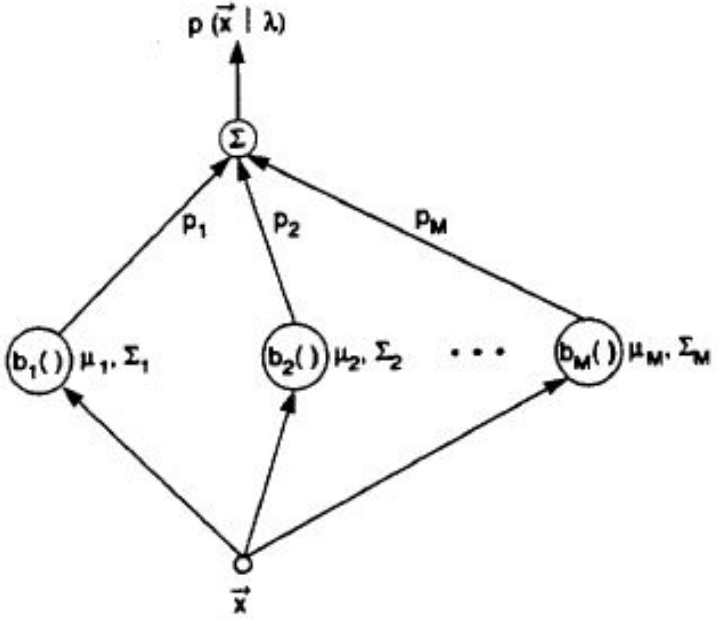
3. Repeated 2. until convergence is reached.

Parameter Estimation

- Conventional GMM training process



Model Description



Gaussian Mixture Density

$$p(\vec{x}|\lambda) = \sum_{i=1}^{M} w_i N_i(\vec{x})$$

Where $\vec{x}$ D-dimensional random vector

Nodal, Grand,Global

Nodal, diagonal (this)

$$N_i(\vec{x}) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(\vec{x}-\vec{\mu}_i)'\Sigma_i^{-1}(\vec{x}-\vec{\mu}_i)\right\}$$

$$\lambda = \{w_i, \vec{\mu}_i, \Sigma_i\} \qquad i = 1, \ldots, M$$

ML Parameter Estimation : Basic Algorithm

- Cluster the training vectors to the mixture component with the highest likelihood

$$C_i = \arg\max_{1 \le i \le M} N_i(\underline{\underline{x}})$$

- Re-estimate parameters of each component

$\overline{w_i}$ number of vectors classified in cluster i / total number of training vectors (T)

$\underline{\underline{\overline{\mu_i}}}$ sample mean of vectors classified in cluster i.

$\overline{\overline{\sum_i}}$ sample covariance matrix of vectors classified in cluster i

EM Algorithm

1. Estimate the GMM parameters via maximum likelihood (ML) estimation

$$p(X \mid \lambda) = \prod_{t=1}^{T} p(\vec{x}_t \mid \lambda) \qquad p(\vec{x}_t \mid \lambda) = \sum_{i=1}^{M} w_i \mathcal{N}_i(\vec{x}) = \sum_{i=1}^{M} w_i p(i \mid \vec{x}, \lambda$$

$$p(i \mid \vec{x}, \lambda) = N_i(\vec{x}) = \frac{1}{(2\pi)^{D/2} \left|\Sigma_i\right|^{1/2}} \exp\left\{ -\frac{1}{2} (\vec{x} - \vec{\mu}_i)' \Sigma_i^{-1} (\vec{x} - \vec{\mu}_i) \right\}$$

2. Expectation-maximization (EM) algorithm

$$\overline{w}_i = \frac{1}{T} \sum_{t=1}^{T} p(i \mid \vec{x}_t, \lambda) \qquad \vec{\overline{\mu}}_i = \frac{\sum_{t=1}^{T} p(i \mid \vec{x}_t, \lambda) \vec{x}_t}{\sum_{t=1}^{T} p(i \mid \vec{x}_t, \lambda)}$$

$$\overline{\sigma}_i^2 = \frac{\sum_{t=1}^{T} p(i \mid \vec{x}_t, \lambda) x_t^2}{\sum_{t=1}^{T} p(i \mid \vec{x}_t, \lambda)} - \overline{\mu}_i^2$$

$$p(X \mid \overline{\lambda}) \geq p(X \mid \lambda)$$

3. Konvergence: $p(X \mid \overline{\lambda}) < p(X \mid \lambda)$

Jika belum konvergen, kembali ke 2.

Convergence

The EM Algorithm will converge because:

- During E step, we make $F(Q^{k+1}, \theta^k) = \log P(D|\theta^k)$.
- During M step, we choose θ^{k+1} that increases F.
- Recall that F is a lower bound,

$$F(Q^{k+1}, theta^{k+1}) \leq \log P(D|\theta^{k+1}).$$

- Implies

$$\log P(D|\theta^k) \leq \log P(D|\theta^{k+1})$$

- Implies convergence! (Why?)

Relation to K-Means

Similarities

K-Means used GMM with:

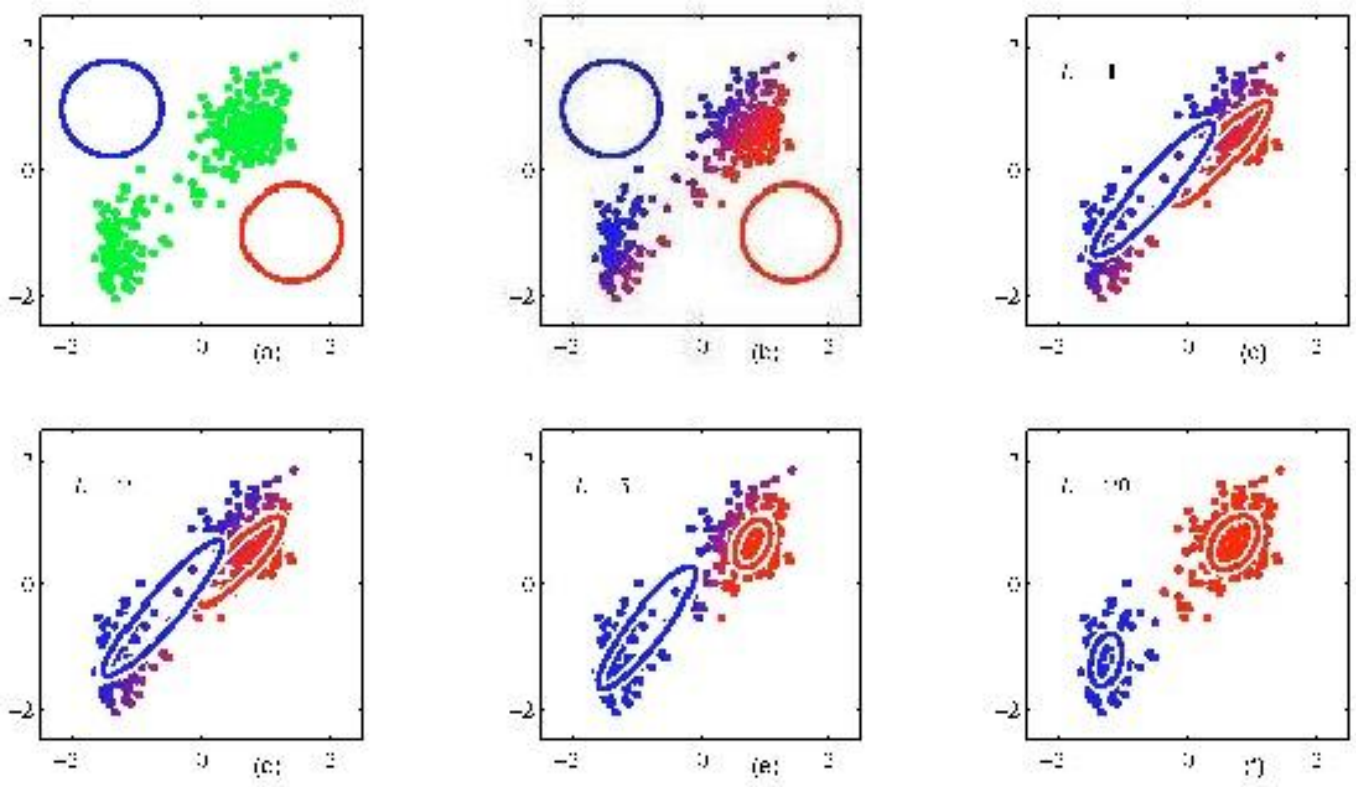
- covariance $\Sigma = I$ (fixed)
- uniform $P(Z_k)$ (fixed)
- unknown means

Alternated estimating labels and recomputing unknown model parameters.

Difference

Makes "hard" assignment to cluster during E step.

EM Visualization



STING

GRID-BASED CLUSTERING

GRID-BASED CLUSTERING

- Popular for mining clusters in a large multidimensional space
- Clusters are regarded as denser regions than their surroundings.
- Differs from the conventional clustering algorithms:
  - concerned not with the data points but with the value space that surrounds the data points.

Advantages

- Fast (complexity):
  - No distance computations
  - Clustering is performed on summaries and not individual objects; complexity is usually O(#-populated-grid-cells) and not O(#objects)
  - Easy to determine which clusters are neighboring
- Shapes are limited to union of grid-cells

Algorithm

Typical grid-based clustering algorithm consists of the following basic steps (Grabusts and Borisov, 2002):

1. Creating the grid structure, i.e., partitioning the data space into a finite number of cells.
2. Calculating the cell density for each cell.
3. Sorting of the cells according to their densities. Eliminate cells, whose density is below a certain threshold τ.
4. Identifying cluster centers. Form clusters from contiguous (adjacent) groups of dense cells (usually minimizing a given objective function)

Grid-Based Clustering Methods

- Using multi-resolution grid data structure
- Clustering complexity depends on the number of populated grid cells and not on the number of objects in the dataset
- Several interesting methods (in addition to the basic grid-based algorithm)
  - STING (a STatistical INformation Grid approach) by Wang, Yang and Muntz (1997)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

STING

- Wang et al. (1997) proposed a STatistical INformation Grid-based clustering method (STING) to cluster spatial databases.
- The algorithm can be used to facilitate several kinds of spatial queries.
- The spatial area is divided into rectangle cells, which are represented by a hierarchical structure.
  - Let the root of the hierarchy be at level 1, its children at level 2, etc.
  - The number of layers could be obtained by changing the number of cells that form a higher-level cell.
  - A cell in level *i* corresponds to the union of the areas of its children in level *i* + 1.
  - Each cell has 4 children and each child corresponds to one quadrant of the parent cell.
  - Only two-dimensional spatial space is considered in this algorithm.

STING (2)

- Each cell at a high level is partitioned into a number of smaller cells in the next lower level
- Statistical info of each cell is calculated and stored beforehand and is used to answer queries
- Parameters of higher level cells can be easily calculated from parameters of lower level cell
  - *count*, *mean*, *s*, *min*, *max*
  - type of distribution—normal, *uniform*, etc.
- Use a top-down approach to answer spatial data queries

STING (3)

Use a top-down approach to answer spatial data queries

1. Start from the root and proceed to the next lower level using STING Index
2. For each cell in the current level, compute the confidence interval indicating a cell's relevance to a given query;
   - If it is relevant, include the cell in a cluster
   - If it irrelevant, remove cell from further consideration
   - otherwise, look for relevant cells at the next lower layer
   - Repeat this process until the bottom layer is reached

PERTANYAAN ?