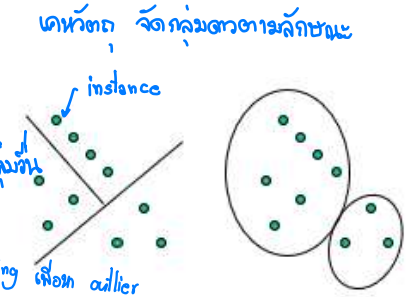
Chapter 4 Clustering

Associate Professor Yachai Limpiyakorn, Ph.D.
Department of Computer Engineering
Chulalongkorn University

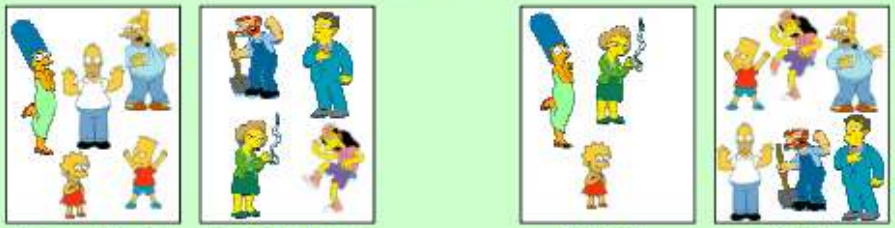
Natural Grouping

- Unsupervised learning
- Natural grouping:
  - High intra-class similarity
  - Low inter-class similarity
- As a tool for understanding data distribution or data preprocessing prior to DM modeling
- Clustering is subjective

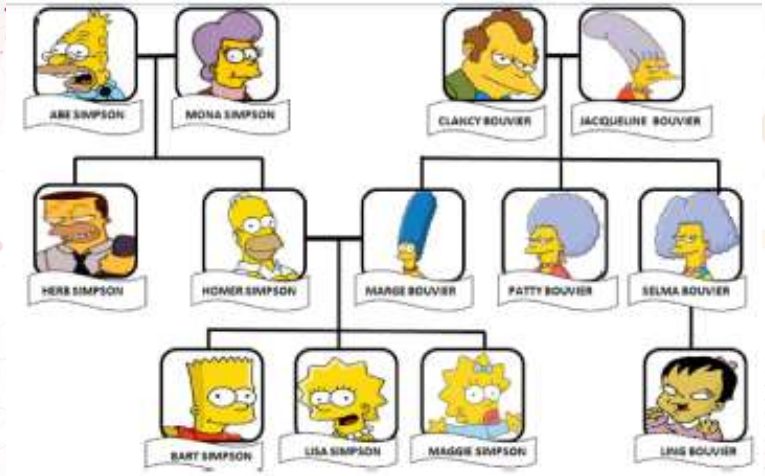


What is a natural grouping among these objects?

Clustering is subjective



Simpson Family



What is similarity?

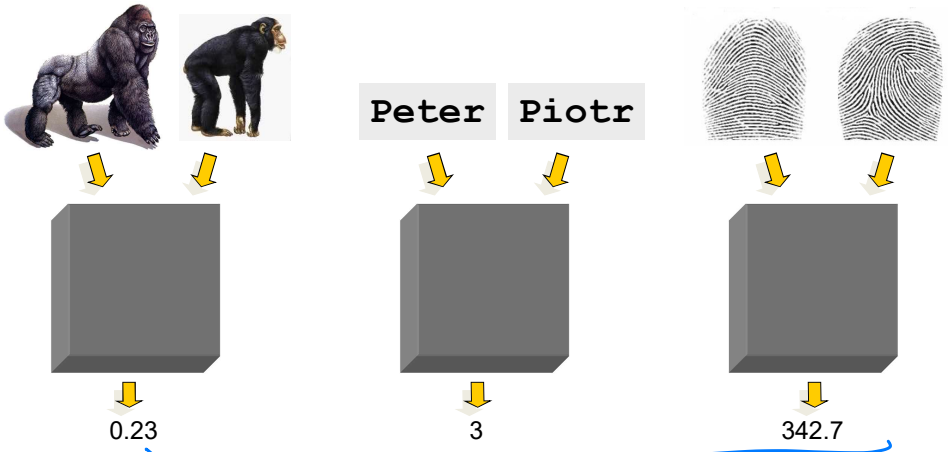
Before Clustering

1. Define data representation or data features
   - DNA, fingerprints, text, ...
2. Define distance measure (dissimilarity)
   - Euclidean distance, Edit distance, ...
3. Define clustering algorithm
   - K-means, Hierarchical, ...

distance $\propto \frac{1}{\text{similarity}}$

Defining distance measures

Definition: Let O_1 and O_2 be two objects from the universe of possible objects. The distance (dissimilarity) between O_1 and O_2 is a real number denoted by $D(O_1,O_2)$



Edit Distance Example

It is possible to transform any string Q into string C, using only *Substitution*, *Insertion* and *Deletion*. Assume that each of these operators has a cost associated with it.

The similarity between two strings can be defined as the cost of the cheapest transformation from Q to C.

Note that for now we have ignored the issue of how we can find this cheapest transformation.

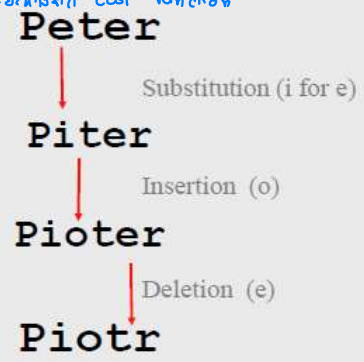
How similar are the names "Peter" and "Piotr"?

Assume the following cost function

Substitution	1 Unit
Insertion	1 Unit
Deletion	1 Unit

D(`Peter`, `Piotr`) is 3

วิธีการหาค่าจาก cost function



Distance Measures

- Edit/ Transformation distance
  - D(Patty,Selma) =4
    - Change dress color 1 pt.
    - Change shoe color 1 pt.
    - Change earring shape 1 pt.
    - Change hair part 1 pt.
- Minkowski distance
  - Manhattan
  - Euclidean

Minkowski Distance

- Some popular ones include: *Minkowski distance*:

$$d(i,j)=\sqrt[q]{(|x_{i1}-x_{j1}|^q+|x_{i2}-x_{j2}|^q+...+|x_{ip}-x_{jp}|^q)}$$

  - where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two p-dimensional data objects, and q is a positive integer
- Manhattan distance

$$d(i,j)=|x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+...+|x_{ip}-x_{jp}|$$

- Euclidean distance

$$d(i,j)=\sqrt{(|x_{i1}-x_{j1}|^2+|x_{i2}-x_{j2}|^2+...+|x_{ip}-x_{jp}|^2)}$$

Properties of Distance Measure

$D(A,B) \geq 0$	*Positivity*
$D(A,B) = D(B,A)$	*Symmetry*
$D(A,B) = 0$ iff $A = B$	*Reflexive*
$D(A,B) \leq D(A,C) + D(B,C)$	*Triangle Inequality*

Distance Matrix

ข้อมูลจะถูกจัดเก็บในเมทริกซ์ (two mode) ประกอบด้วยข้อมูล n ตัว แต่ละตัวมีคุณลักษณะ p คุณลักษณะ

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

มาตรวัดค่าความ(ไม่)คล้ายอยู่ในรูปแบบฟังก์ชันระยะห่าง *d(i,j)* ซึ่งจัดเก็บในเมทริกซ์ความไม่คล้าย นิยามฟังก์ชันระยะห่าง d(i,j) มักแตกต่างกันไปตามประเภทข้อมูล

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

ประเภทตัวแปร

- Numeric variables
- Binary variables
- Nominal
- Ordinal
- Variables of mixed types

Numeric Variable

Normalization or Standardization is suggested to transform to equivalent ranges
ทำให้ตัวแปรกว้างเท่ากัน

ratio - scaled

ตัวแปรที่ growth rate ไม่เป็นเชิงเส้น เช่น การเจริญเติบโตของแบคทีเรีย หรือการเสื่อมสลายของธาตุกัมมันตภาพรังสี ควร take log( ) prior to Normalization

Binary Variable

Symmetric

- Simple Matching Coefficient

$$d(i,j) = \frac{b+c}{a+b+c+d}$$

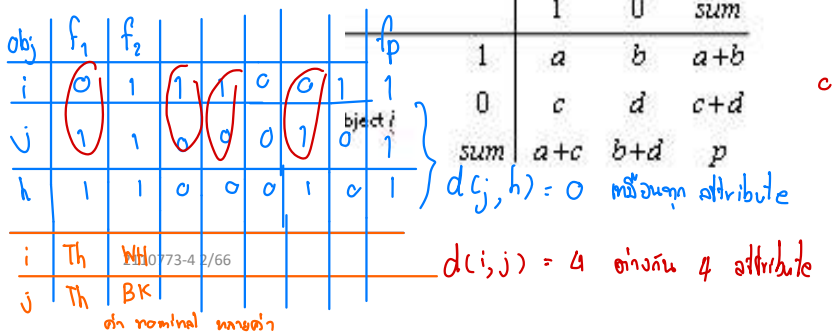
Asymmetric (0,0 ไม่สื่อความหมาย เช่น คุณตรวจเชื้อ 1 มีความหมายมากกว่า 0)

- Jaccard Coefficient

$$d(i,j) = \frac{b+c}{a+b+c}$$

ตัด d ทิ้งเพราะ (0,0 ไม่สื่อความหมาย)

หาทีละคู่



Object i \ Object j	1	0	sum
1	a	b	a+b
0	c	d	c+d
sum	a+c	b+d	p

c คือ attribute (0, 1)

d(j, k) = 0 เพื่อนทุก attribute

d(i, j) = 4 ต่างกัน 4 attribute

Nominal Variable

- วิธีการคำนวณความ(ไม่)คล้ายของตัวแปรไร้อันดับ อาจเลือกใช้ฟังก์ชันระยะห่าง
- วิธีที่1: Simple Matching

กำหนดให้ m เป็นจำนวนตัวแปรไร้อันดับที่วัตถุทั้งคู่มีค่าเหมือนกัน

p เป็นจำนวนตัวแปรทั้งหมด ระยะห่างระหว่างวัตถุทั้งสองวัดได้โดย

$$d(i,j) = \frac{p-m}{p}$$

(mismatched)

- วิธีที่2: แทนตัวแปรไร้อันดับที่มีค่าที่เป็นไปได้ M ค่า ด้วยตัวแปรทวิจำนวน M ตัว

ใช้ 7 bit กรณีมี 7 ค่า

↓

one hot

Ordinal Variable

- การคำนวณหาความ(ไม่)คล้ายของวัตถุที่มีตัวแปรแบบมีอันดับ สามารถใช้ฟังก์ชันระยะห่างที่ใช้สำหรับตัวแปร interval-scaled ได้ โดยการแทนค่าตัวแปร x_{if} ด้วยค่าอันดับของตัวแปร

$$r_{if} \in \{1,\ldots,M_f\}$$

- แปลงค่าตัวแปรมีอันดับที่ f^{th} ของวัตถุตัวที่ i^{th} ด้วยสมการข้างล่าง เพื่อแปลงช่วงค่าที่เป็นไปได้ของตัวแปรให้อยู่ภายในช่วง [0,1]

$$z_{if} = \frac{r_{if}-1}{M_f-1}$$

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}$$

สำหรับค่าน้ำหนัก $\delta_{ij}^{(f)} = 0$ ในกรณี

- ค่าข้อมูลของ x_{if} หรือ x_{jf} สูญหาย
- หรือ ตัวแปร f เป็น asymmetric binary และ $x_{if} = x_{jf} = 0$
- ไม่เช่นนั้น $\delta_{ij}^{(f)} = 1$

Variables of Mixed Types

- ถ้า f เป็นตัวแปร nominal หรือ binary จะได้ว่า $d_{ij}^{(f)} = 0$ ถ้า $x_{if} = x_{jf}$, หรือไม่เช่นนั้น $d_{ij}^{(f)} = 1$
- ถ้า f เป็นตัวแปร interval-scaled ให้ใช้ระยะห่างนอร์มอลไลซ์

$$d_{ij}^{(f)} = \frac{|x_{if}-x_{jf}|}{max_h x_{hf} - min_h x_{hf}}$$

- ถ้า f เป็นตัวแปร ordinal หรือ ratio-scaled
  - คำนวณค่าอันดับ r_{if} และนอร์มอลไลซ์ด้วยค่า $z_{if} = \frac{r_{if}-1}{M_f-1}$
  - มองว่าค่า z_{if} เสมือนตัวแปร interval-scaled

Table 7.3 A sample data table containing variables of mixed type.

object identifier	test-1 (categorical)	test-2 (ordinal)	test-3 (ratio-scaled)
1	code-A	excellent	445
2	code-B	fair	22
3	code-C	good	164
4	code-A	excellent	1,210

Dissimilarity Matrix of test-1 (categorical)

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ d(4,1) & d(4,2) & d(4,3) & 0 \end{bmatrix} \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & 1 & 0 & \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

test-2 (ordinal)

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 0.5 & 0.5 & 0 & \\ 0 & 1.0 & 0.5 & 0 \end{bmatrix}$$

test-3 (ratio-scale)

$$\begin{bmatrix} 0 & & & \\ 1.31 & 0 & & \\ 0.44 & 0.87 & 0 & \\ 0.43 & 1.74 & 0.87 & 0 \end{bmatrix}$$

we can use the dissimilarity matrices obtained for *test*-1 and *test*-2 later when we compute Equation (7.15). First, however, we need to complete some work for *test*-3 (which is ratio-scaled). We have already applied a logarithmic transformation to its values. Based on the transformed values of 2.65, 1.34, 2.21, and 3.08 obtained for the objects 1 to 4, respectively, we let $max_h = 3.08$ and $min_h = 1.34$. We then normalize the values in the dissimilarity matrix obtained in Example 7.5 by dividing each one by (3.08-1.34) = 1.74. This results in the following dissimilarity matrix for *test*-3:

$$\begin{bmatrix} 0 & & & \\ 0.75 & 0 & & \\ 0.25 & 0.50 & 0 & \\ 0.25 & 1.00 & 0.50 & 0 \end{bmatrix}$$

We can now use the dissimilarity matrices for the three variables in our computation of Equation (7.15). For example, we get $d(2,1) = \frac{1(1)+1(1)+1(0.75)}{3} = 0.92$. The resulting dissimilarity matrix obtained for the data described by the three variables of mixed types is:

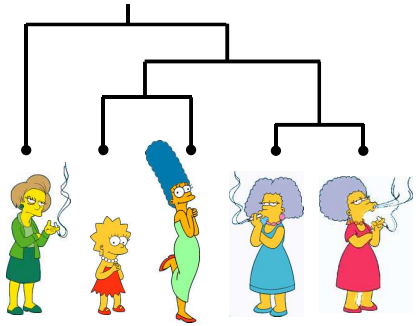
$$\begin{bmatrix} 0 & & & \\ 0.92 & 0 & & \\ 0.58 & 0.67 & 0 & \\ 0.08 & 1.00 & 0.67 & 0 \end{bmatrix}$$

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}$$

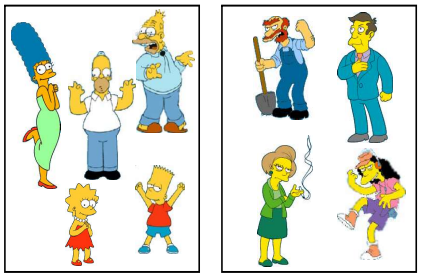
Two types of clustering

• **Partitional algorithms:** Construct various partitions and then evaluate them by some criterion
• **Hierarchical algorithms:** Create a hierarchical decomposition of the set of objects using some criterion

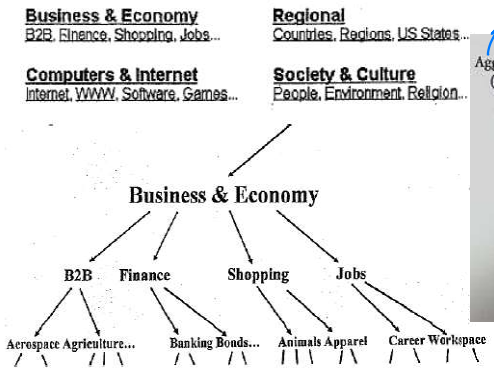
Hierarchical

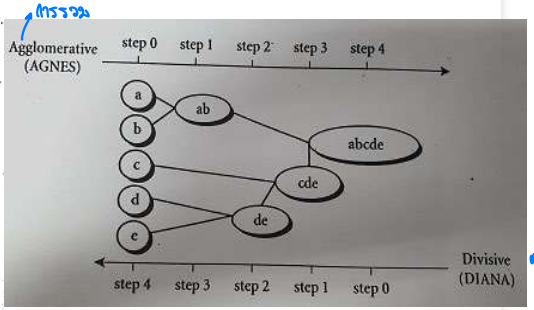


Partitional

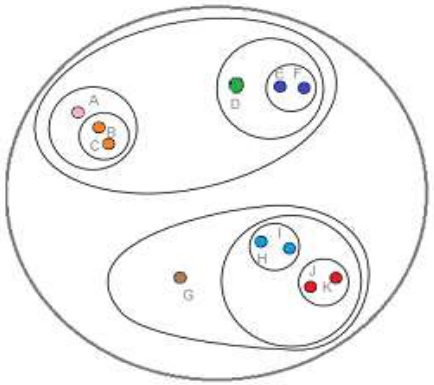


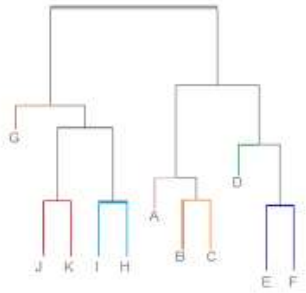
Top down vs. Bottom up





ตัวอย่างลำดับชั้นการจัดกลุ่มเว็บไซต์ไดเรกทอรี



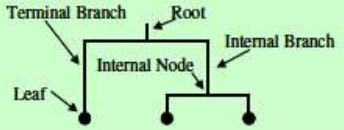


A dendrogram (right) representing nested clusters (left).

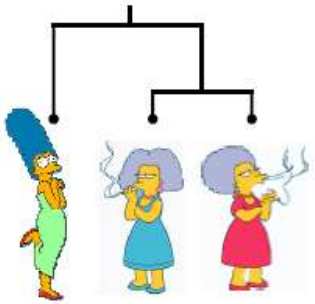
Dendrogram

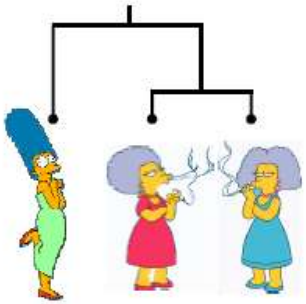
- tree diagram showing hierarchical clustering — relationships between similar sets of data.
- A tree that shows clustering process.

Parts of a Dendrogram



The similarity between two objects in a dendrogram is represented as the height of the lowest internal node they share.

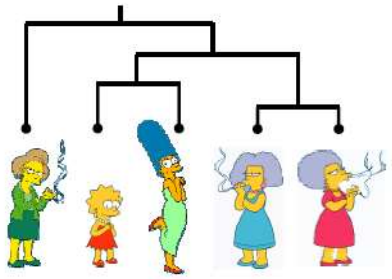




(How-to) Hierarchical Clustering

The number of dendrograms with n leafs $= (2n-3)!/[(2^{(n-2)})(n-2)!]$

Number of Leafs	Number of Possible Dendrograms
2	1
3	3
4	15
5	105
...	...
10	34,459,425

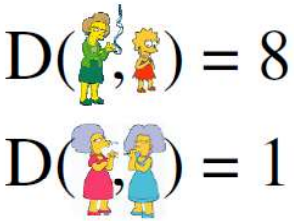


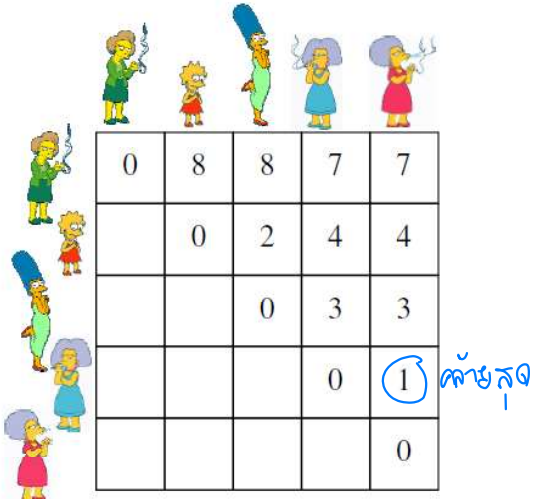
Since we cannot test all possible trees we will have to heuristic search of all possible trees. We could do this..

Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

Top-Down (divisive): Starting with all the data in a single cluster, consider every possible way to divide the cluster into two. Choose the best division and recursively operate on both sides.

We begin with a distance matrix which contains the distances between every pair of objects in our database.



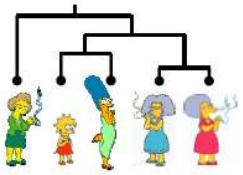


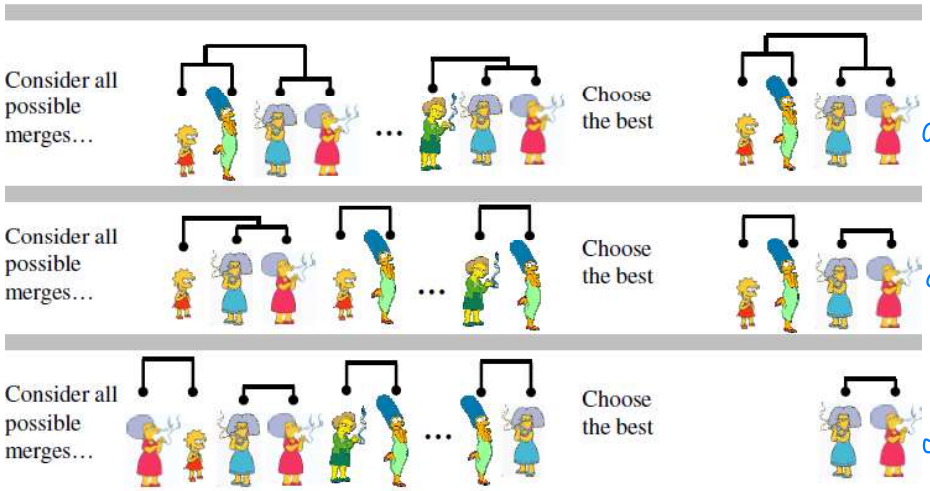
0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

คล้ายสุด

สร้างจากล่างขึ้นบน

Bottom-Up (agglomerative):
Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



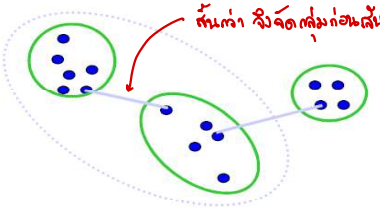


- **Single-linkage**: similarity of the closest pair. This can cause premature merging of groups with close pairs, even if those groups are quite dissimilar overall.

Similarity criteria: Single Link

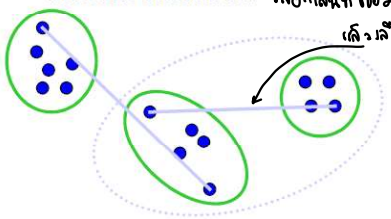
- cluster similarity = similarity of two most similar members

member ที่เหมือนสุดถูก group ก่อน



Hierarchical: Complete Link

- cluster similarity = similarity of two least similar members เลือกเส้นที่ เพื่อมที่น่วงมากที่สุด



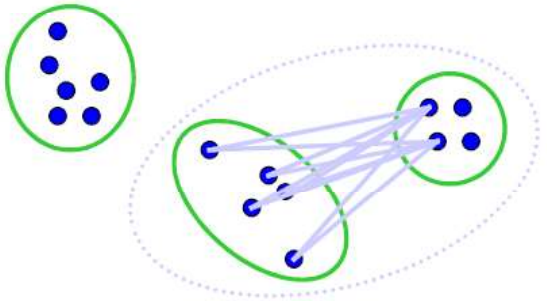
+ tight clusters

- **Complete linkage**: similarity of the furthest pair. One drawback is that outliers can cause merging of close groups later than is optimal.

ข้อด้อยคือ outlier ที่ห่างเสียเปรียบ ไม่มีโอกาสถูกจัดกลุ่มก่อน

Hierarchical: Average Link

- cluster similarity = average similarity of all pairs



the most widely used similarity measure

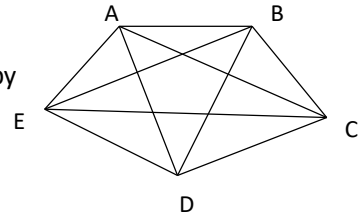
Robust against noise

- similarity between groups.

An example to show different Links

- Single link
  - Merge the nearest clusters measured by the shortest edge between the two
  - (((A B) (C D)) E)
- Complete link
  - Merge the nearest clusters measured by the longest edge between the two
  - (((A B) E) (C D))
- Average link
  - Merge the nearest clusters measured by the average edge length between the two
  - (((A B) (C D)) E)

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0



Example Merge w/ Complete Linkage (1)

samples	A	B	C	D	E	F	G
A	0	0.5000	0.4286	1.0000	0.2500	0.6250	0.3750
B	0.5000	0	0.7143	0.8333	0.6667	0.2000	0.7778
C	0.4286	0.7143	0	1.0000	0.4286	0.6667	0.3333
D	1.0000	0.8333	1.0000	0	1.0000	0.8000	0.8571
E	0.2500	0.6667	0.4286	1.0000	0	0.7778	0.3750
F	0.6250	0.2000	0.6667	0.8000	0.7778	0	0.7500
G	0.3750	0.7778	0.3333	0.8571	0.3750	0.7500	0

① ดูที่น้อยสุด

② กรณีเท่ากัน เลือกตัวไหนก็ได้

A→B กับ A→F 0.5 0.6250

samples	A	(B,F)	C	D	E	G
A	0	0.6250	0.4286	1.0000	0.2500	0.3750
(B,F)	0.6250	0	0.7143	0.8333	0.7778	0.7778
C	0.4286	0.7143	0	1.0000	0.4286	0.3333
D	1.0000	0.8333	1.0000	0	1.0000	0.8571
E	0.2500	0.7778	0.4286	1.0000	0	0.3750
G	0.3750	0.7778	0.3333	0.8571	0.3750	0

น้อยสุดก็วนไปทำงานต่อไป

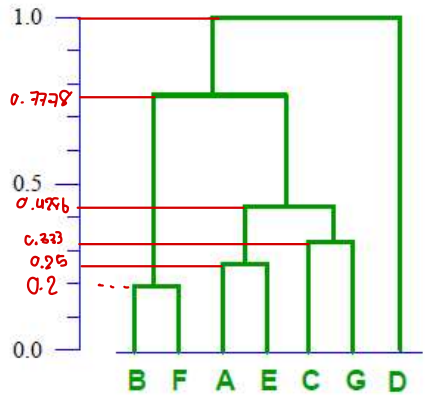
Example Merge w/ Complete Linkage (2)

samples	(A,E)	(B,F)	C	D	G
(A,E)	0	0.7778	0.4286	1.0000	0.3750
(B,F)	0.7778	0	0.7143	0.8333	0.7778
C	0.4286	0.7143	0	1.0000	0.3333
D	1.0000	0.8333	1.0000	0	0.8571
G	0.3750	0.7778	0.3333	0.8571	0

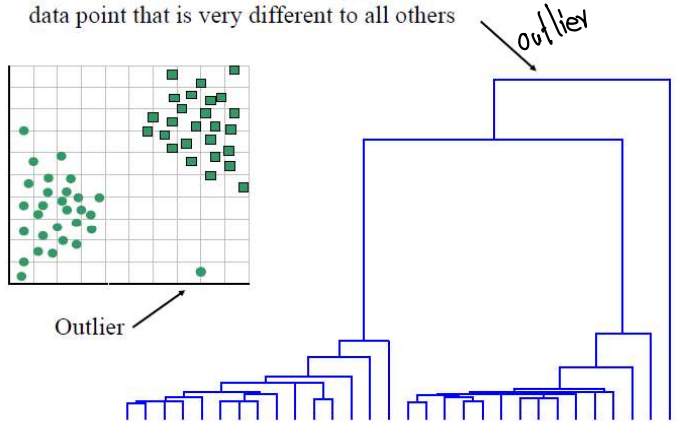
samples	(A,E)	(B,F)	(C,G)	D
(A,E)	0	0.7778	0.4286	1.0000
(B,F)	0.7778	0	0.7778	0.8333
(C,G)	0.4286	0.7778	0	1.0000
D	1.0000	0.8333	1.0000	0

samples	(A,E,C,G)	(B,F)	D
(A,E,C,G)	0	0.7778	1.0000
(B,F)	0.7778	0	0.8333
D	1.0000	0.8333	0

samples	(A,E,C,G,B,F)	D
(A,E,C,G,B,F)	0	1.0000
D	1.0000	0



Dendrogram for outlier detection

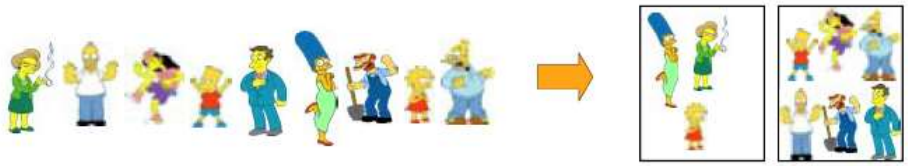


Summary of hierarchal clustering

- No need to specify the number of clusters in advance.
- Hierarchical structure maps nicely onto human intuition for some domains
- They do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects.
- Like any heuristic search algorithms, local optima are a problem.
- Interpretation of results is (very) subjective.
- Only a tree is returned

Partitional Clustering

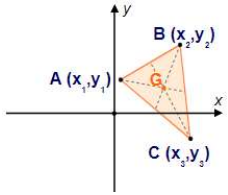
- Nonhierarchical, each instance is placed in exactly one of K nonoverlapping clusters.
- Since only one set of clusters is output, the user normally has to input the desired number of clusters K.



Centroid vs. Medoid

- Centroid, also called a geometric center, is the **center of mass of an object** (geometric balanced point)
- In general, a centroid is the *arithmetic mean* of all the points in the shape.

centroid เป็นแค่สมัคร ไม่ได้อยู่ใน dataset

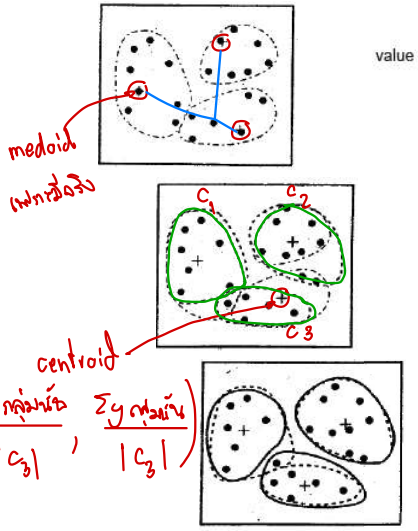


- Medoids are representative objects of a data set or a cluster within a data set whose sum of dissimilarities to all the objects in the cluster is minimal.
- Medoids are similar in concept to centroids, but medoids are always restricted to be members of the data set.

ค่าตัวจริงๆ ที่เป็นตัวแทนกลาง

medoid คือจุดกลางของ dataset

K-Means Algorithm



Algorithm: The *k*-Means algorithm for partitioning based on the mean value of objects in the cluster.

Input: The number of clusters *k* and a database containing *n* objects.

Output: A set of *k* clusters that minimizes the squared-error criterion.

Method:

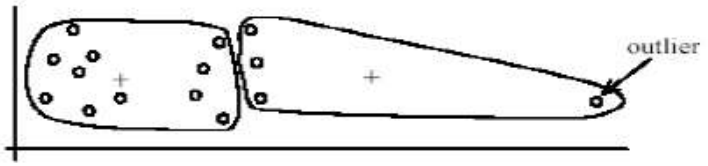
สุ่มเลือกจุดๆ k ตัวมาก่อน

(1) arbitrarily choose *k* objects as the initial cluster centers;
(2) repeat
(3) (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 cluster;
(4) update the cluster means i.e., calculate the mean value of the objects for each cluster;
(5) until no change;

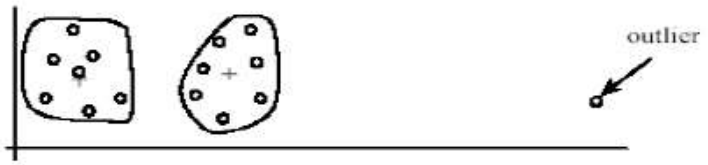
K-means Summary

- Strength
  - Simple, easy to implement and debug
  - Intuitive objective function: optimizes intra-cluster similarity
  - *Relatively efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t << n$.
- Weakness
  - Applicable only when *mean* is defined, then what about categorical data? การใช้ค่า mean จะใช้ categorical ไม่ได้
  - Often terminates at a *local optimum*. Initialization is important.
  - Need to specify K, the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers* จับ outlier ไม่ได้
  - Not suitable to discover clusters with *non-convex shapes* ไม่ค่อยเหมาะกับการจับ cluster ที่เป็นรูปร่างเลขสมน

Weaknesses of k-means: Problems with outliers



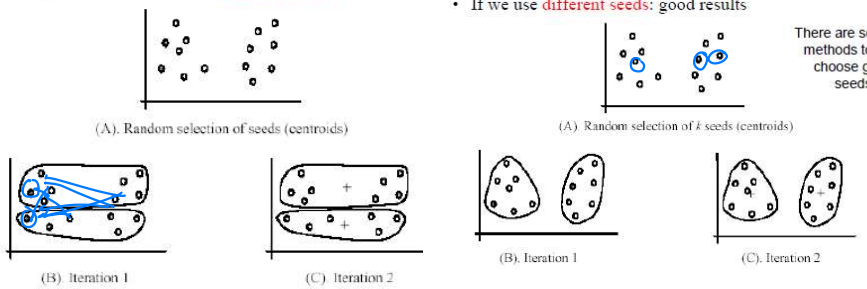
(A): Undesirable clusters



(B): Ideal clusters

Weaknesses of k-means (cont …)

- The algorithm is sensitive to initial seeds.

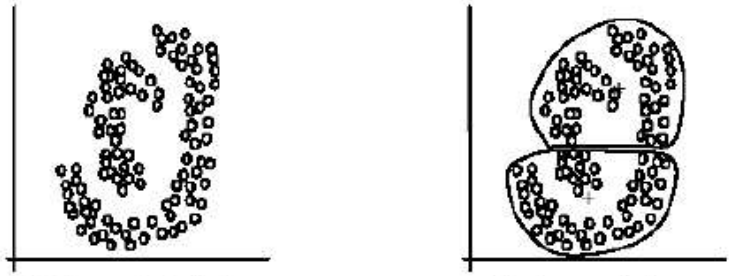


- If we use different seeds: good results

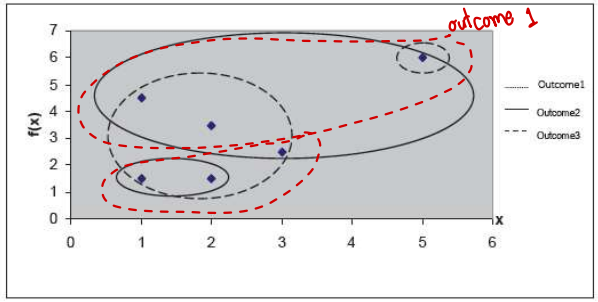
There are some methods to help choose good seeds

Weaknesses of k-means (cont …)

- The *k*-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).



คุณภาพการจัดกลุ่ม



วัตถุ	X	Y
1	1.0	1.5
2	1.0	4.5
3	2.0	1.5
4	2.0	3.5
5	3.0	2.5
6	5.0	6.0

Outcome	Cluster Centers	Cluster Points	Squared Error
1	(2.67,4.67) (2.00,1.83)	2, 4, 6 1, 3, 5	14.50
2	(1.5,1.5) (2.75,4.125)	1, 3 2, 4, 5, 6	15.94
3	(1.8,2.7) (5,6)	1, 2, 3, 4, 5 6	9.60

$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} |p - \overline{m}_i|^2$$

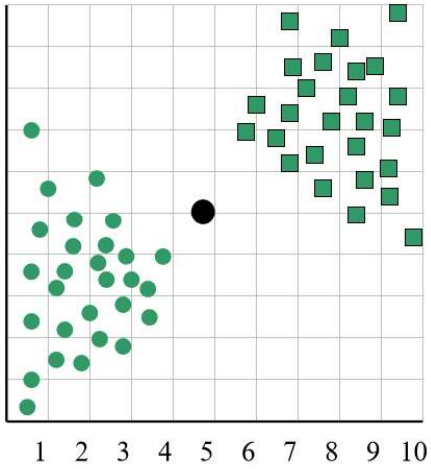
โดย ค่า E เป็นผลรวมค่าความผิดพลาดยกกำลังสองของวัตถุทั้งหมด

p คือ ค่าพิกัดในปริภูมิหลายมิติของวัตถุ

m_i คือ ค่าพิกัดเฉลี่ยในปริภูมิหลายมิติของกลุ่ม C_i

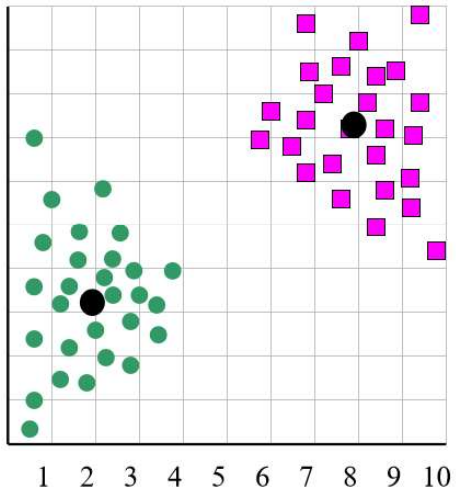
How can we tell the *right* number of clusters?

K=1



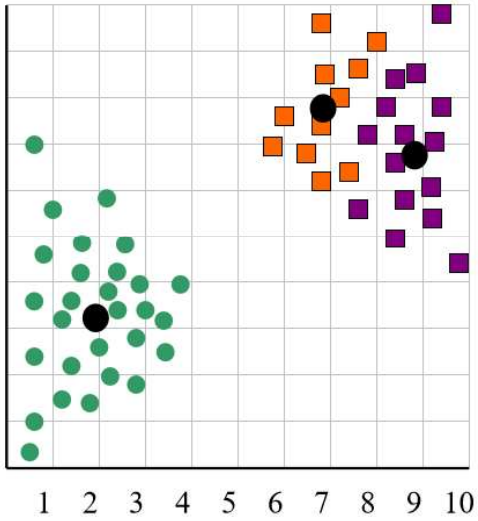
How can we tell the *right* number of clusters?

K=2



How can we tell the *right* number of clusters?

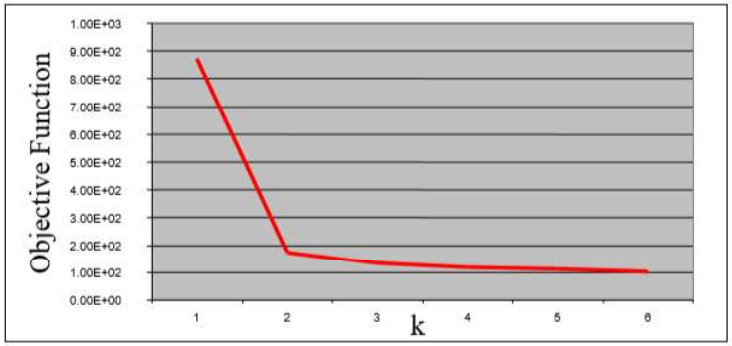
K=3



ชื่อเรียกอื่นๆ SSE

We can plot the objective function values for k equals 1 to 6…

The abrupt change at k = 2, is highly suggestive of two clusters in the data. This technique for determining the number of clusters is known as "knee finding" or "elbow finding".



Note that the results are not always as clear cut as in this toy example