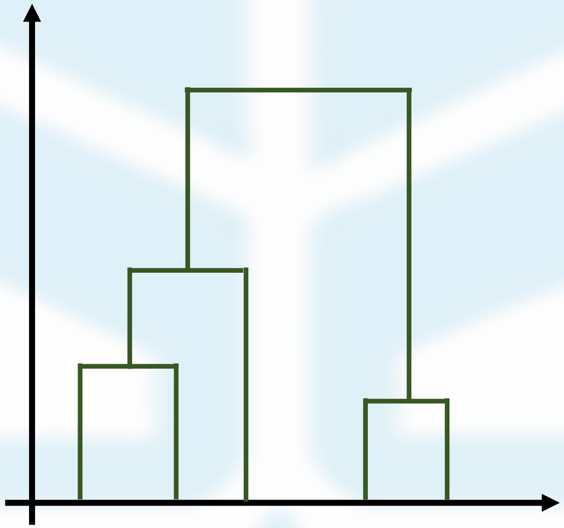
به نام او، به یاد او، برای او

Part 2

درس داده‌کاوی

خوشه‌بندی سلسله‌مراتبی



Dr. *Ali Valinejad*

valinejad.ir
valinejad@umz.ac.ir

Supervised learning vs. Unsupervised learning

❖ Hierarchical Clustering Approach:

- These find successive clusters using previously established clusters. A set of nested clusters organized as a hierarchical tree.
- A typical clustering analysis approach via ***partitioning*** data set ***sequentially.***
- Construct nested partitions layer by layer via grouping objects into a tree of clusters.
- Do not need to know the number of clusters in advance.
- Use distance matrix as clustering criteria.

❖ Agglomerative vs. Divisive:

Agglomerative: a bottom-up strategy

- Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters.

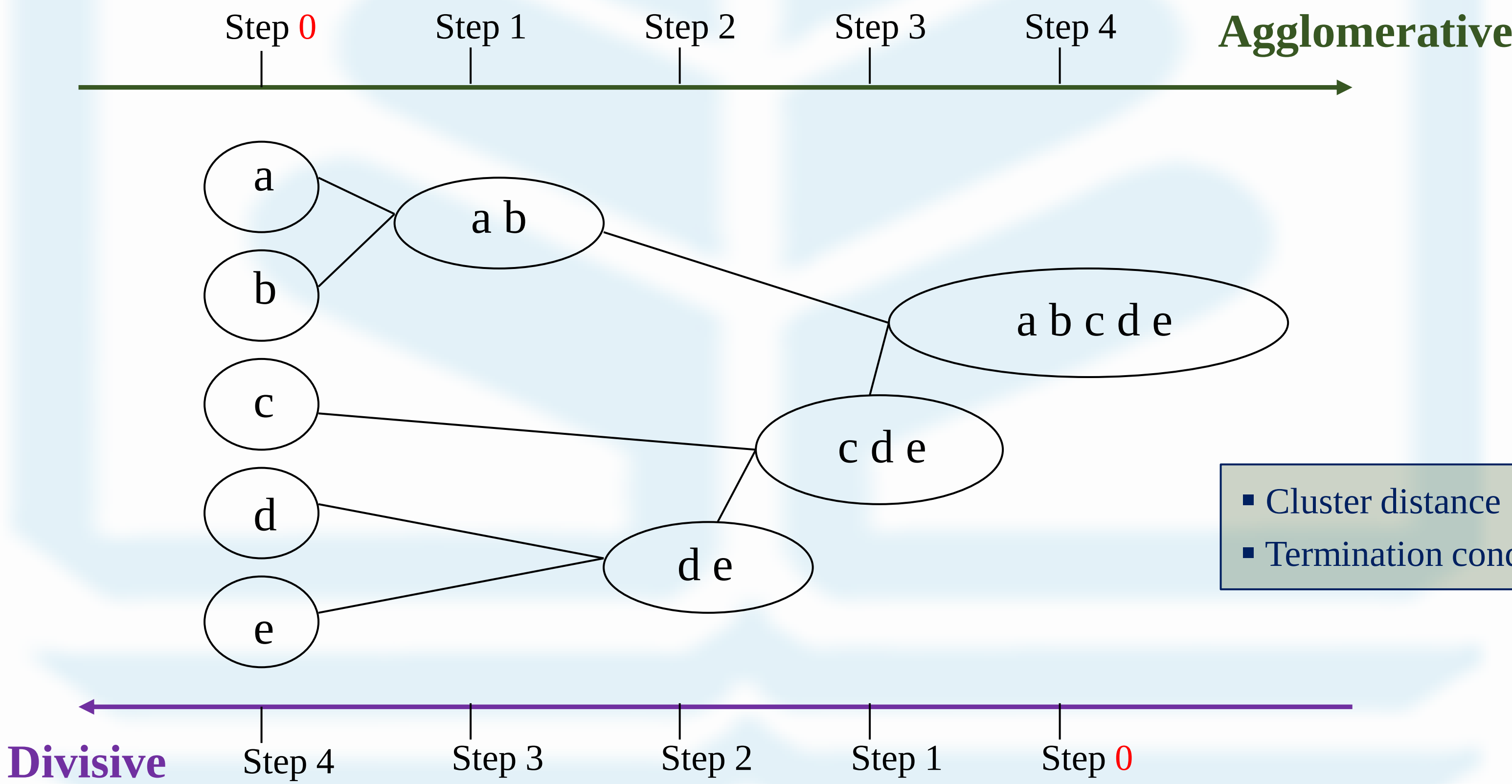
Divisive: a top-down strategy

- Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

Agglomerative vs. Divisive clustering

❖ **Example:**

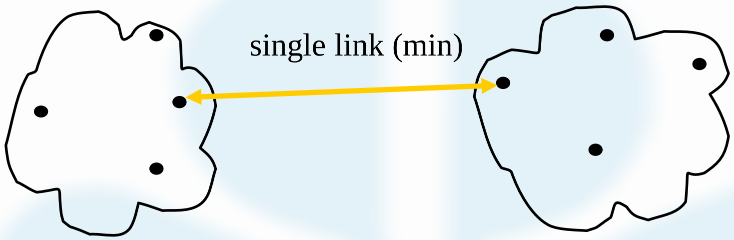
Agglomerative and *divisive* clustering on the data set {a, b, c, d, e }



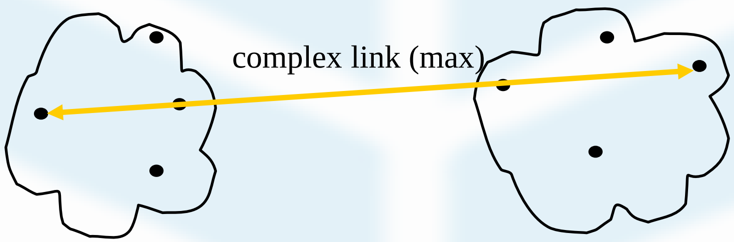
- Cluster distance
- Termination condition

Cluster Distance Measures

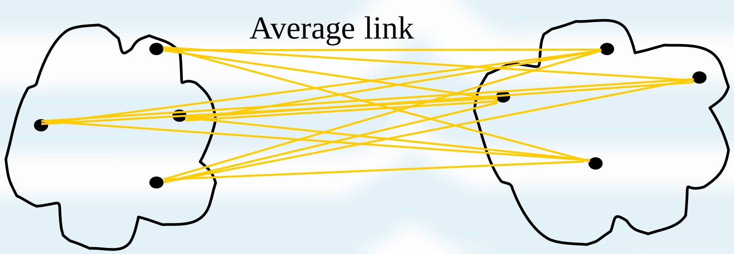
- **Single link:** smallest distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \min\{d(x_{ip}, x_{jq})\}$



- **Complete link:** largest distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \max\{d(x_{ip}, x_{jq})\}$



- **Average:** avg distance between elements in one cluster and elements in the other, i.e.,
  $d(C_i, C_j) = \mathrm{avg}\{d(x_{ip}, x_{jq})\}$



Cluster Distance Measures

Example: Given a data set of five objects characterized by a single continuous feature, assume that there are two clusters: C1: {a, b} and C2: {c, d, e}.

	a	**b**	**c**	**d**	**e**
Feature	1	2	4	5	6

1. Calculate the distance matrix .
2. Calculate three cluster distances between C1 and C2.

Cluster Distance Measures

Example: Given a data set of five objects characterized by a single continuous feature, assume that there are two clusters: C1: {a, b} and C2: {c, d, e}.

	a	b	c	d	e
Feature	1	2	4	5	6

1. Calculate the distance matrix .
2. Calculate three cluster distances between C1 and C2.

$$\begin{array}{c|c} & \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{bmatrix} 0 & 1 & 3 & 4 & 5 \\ 1 & 0 & 2 & 3 & 4 \\ 3 & 2 & 0 & 1 & 2 \\ 4 & 3 & 1 & 0 & 1 \\ 5 & 4 & 2 & 1 & 0 \end{bmatrix} \end{array}$$

Single link

$$\mathrm{dist}(C_1, C_2) = \min\{d(a,c), d(a,d), d(a,e), d(b,c), d(b,d), d(b,e)\}$$
$$= \min\{3, 4, 5, 2, 3, 4\} = 2$$

Complete link

$$\mathrm{dist}(C_1, C_2) = \max\{d(a,c), d(a,d), d(a,e), d(b,c), d(b,d), d(b,e)\}$$
$$= \max\{3, 4, 5, 2, 3, 4\} = 5$$

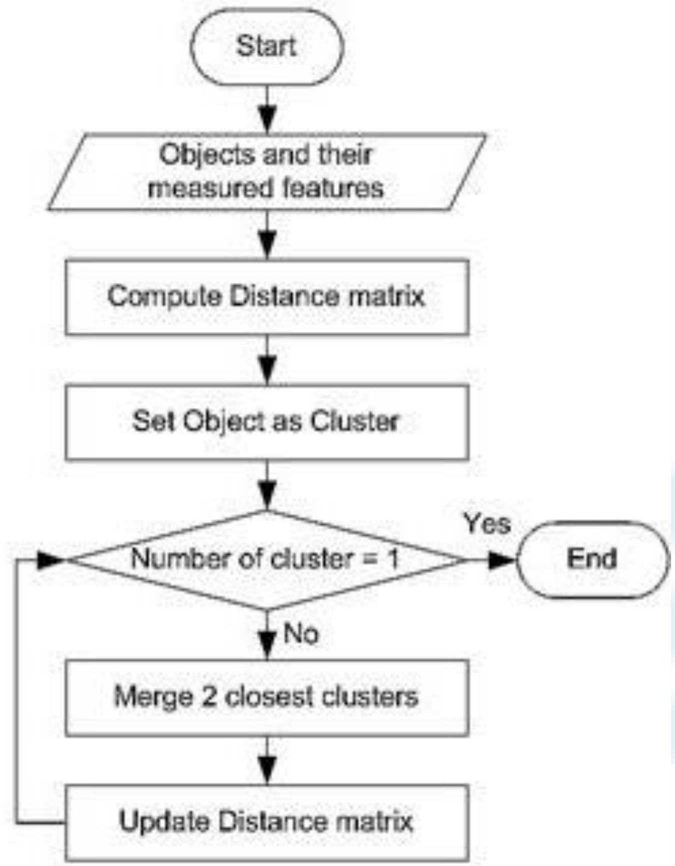
Average

$$\mathrm{dist}(C_1, C_2) = \frac{d(a,c) + d(a,d) + d(a,e) + d(b,c) + d(b,d) + d(b,e)}{6}$$
$$= \frac{3+4+5+2+3+4}{6} = \frac{21}{6} = 3.5$$

Agglomerative Algorithm

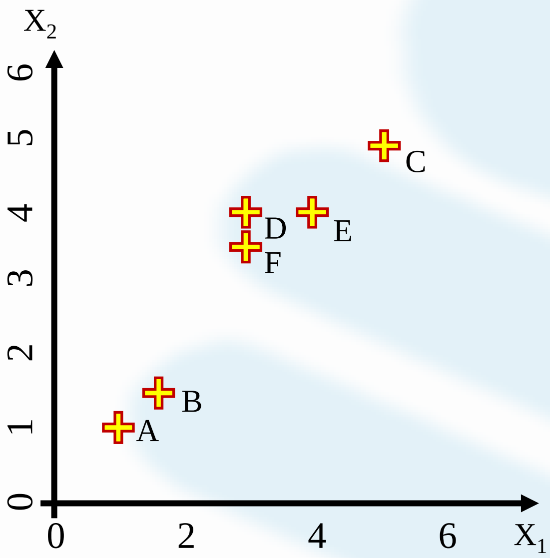
The ***Agglomerative*** algorithm is carried out in three steps:

1) Convert all object features into a distance matrix.
2) Set each object as a cluster (thus if we have *N* objects, we will have *N* clusters at the beginning).
3) Repeat until number of cluster is one (or known # of clusters):
   - Merge two closest clusters
   - Update "distance matrix"



Agglomerative Algorithm (Example)

Problem: clustering analysis with agglomerative algorithm

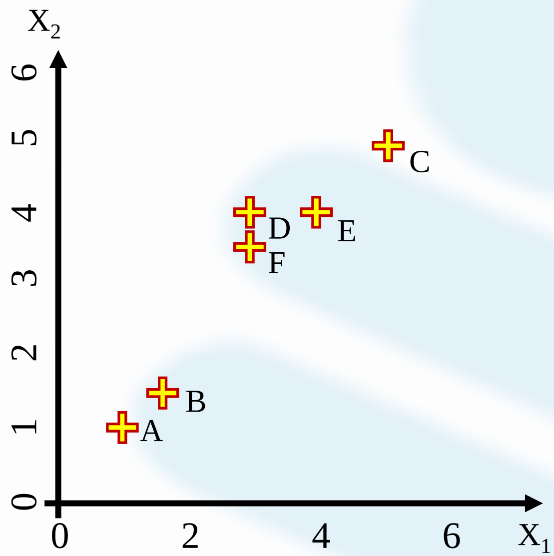


	X_1	X_2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

data matrix

Agglomerative Algorithm (Example)

Problem: clustering analysis with agglomerative algorithm



	X_1	X_2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

$$d_{AB} = \left((1-1.5)^2 + (1-1.5)^2\right)^{\frac{1}{2}} = \sqrt{\tfrac{1}{2}} = 0.7071$$

$$d_{DF} = \left((3-3)^2 + (4-3.5)^2\right)^{\frac{1}{2}} = 0.5$$

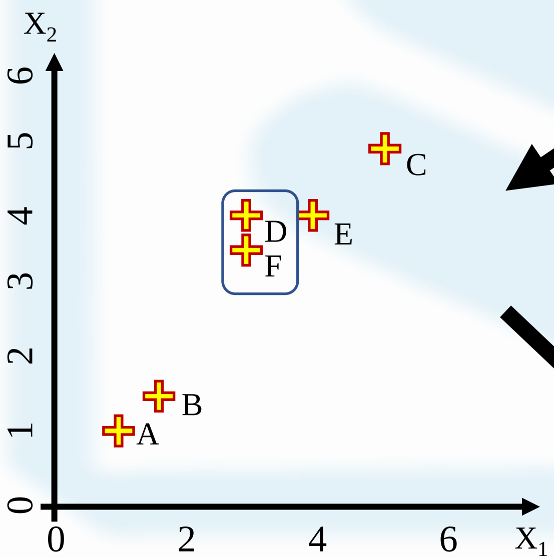
Euclidean distance

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

distance matrix

Agglomerative Algorithm (Example)

Iteration 1: Merge two closest clusters



Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

Agglomerative Algorithm (Example)

Iteration 1: Update distance matrix

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

$$d_{(D,F)\mapsto A} = \min\left(d_{DA}, d_{FA}\right) = \min\left(3.61, 3.20\right) = 3.20$$

$$d_{(D,F)\mapsto B} = \min\left(d_{DB}, d_{FB}\right) = \min\left(2.92, 2.50\right) = 2.50$$

$$d_{(D,F)\mapsto C} = \min\left(d_{DC}, d_{FC}\right) = \min\left(2.24, 2.50\right) = 2.24$$

$$d_{E\to(D,F)} = \min\left(d_{ED}, d_{EF}\right) = \min\left(1.00, 1.12\right) = 1.00$$

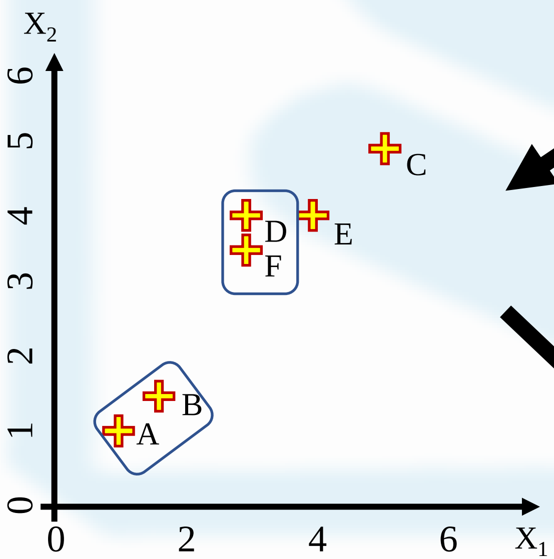
Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

Agglomerative Algorithm (Example)

Iteration 2: Merge two closest clusters



Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	**?**	0	2.24	1.41
(D, F)	**?**	2.24	0	1.00
E	**?**	1.41	1.00	0

Agglomerative Algorithm (Example)

Iteration 2: Update distance matrix

Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

$$d_{C\to(A,B)} = \min\left(d_{CA}, d_{CB}\right) = \min\left(5.66, 4.95\right) = 4.95$$

$$d_{(D,F)\to(A,B)} = \min\left(d_{DA}, d_{DB}, d_{FA}, d_{FB}\right) = \min\left(3.61, 2.92, 3.20, 2.50\right) = 2.50$$

$$d_{E\to(A,B)} = \min\left(d_{EA}, d_{EB}\right) = \min\left(4.24, 3.54\right) = 3.54$$

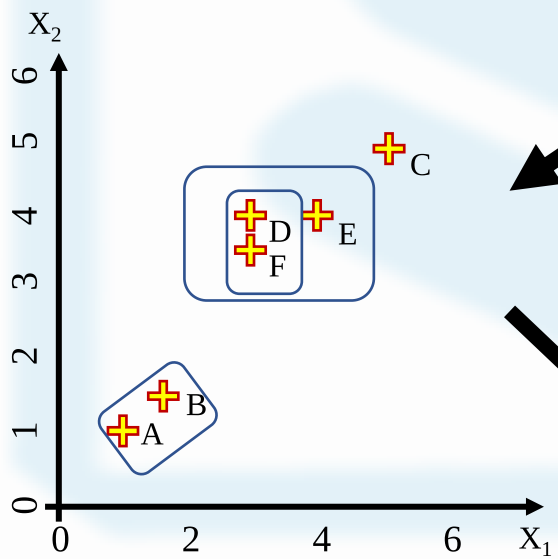
Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	**?**	0	2.24	1.41
(D, F)	**?**	2.24	0	1.00
E	**?**	1.41	1.00	0

Min Distance (Single Linkage)

Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

Agglomerative Algorithm (Example)

Iteration 3: Merge two closest clusters & Update distance matrix



Min Distance (Single Linkage)

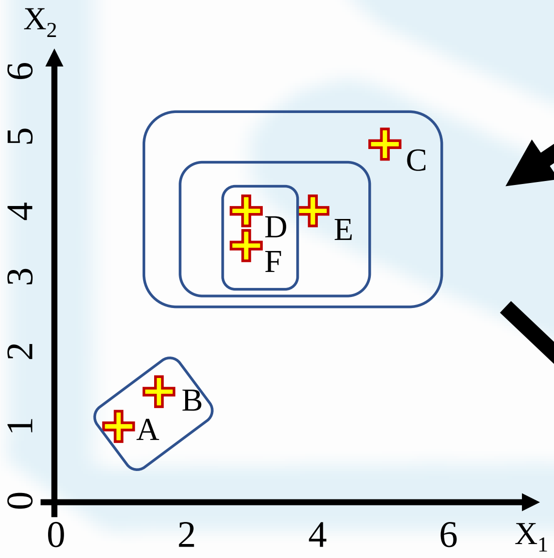
Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	4.95	0	2.24	1.41
(D, F)	2.50	2.24	0	1.00
E	3.54	1.41	1.00	0

Min Distance (Single Linkage)

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	2.50	1.41	0.00

Agglomerative Algorithm (Example)

Iteration 4: Merge two closest clusters & Update distance matrix



Min Distance (Single Linkage)

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	2.50	1.41	0.00

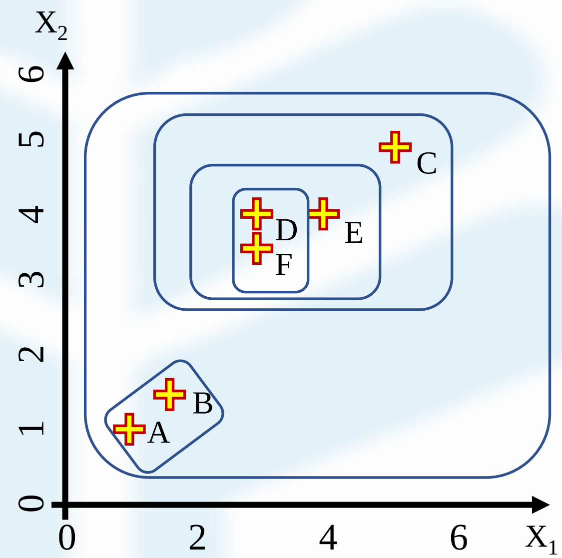
Min Distance (Single Linkage)

Dist	(A,B)	((D, F), E),C
(A,B)	0.00	2.50
((D, F), E),C	2.50	0.00

Agglomerative Algorithm
(Example)

Final Result (meeting termination condition)

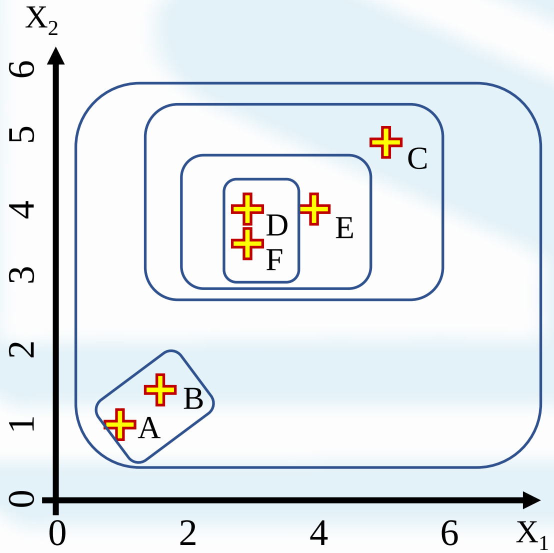
	X_1	X_2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

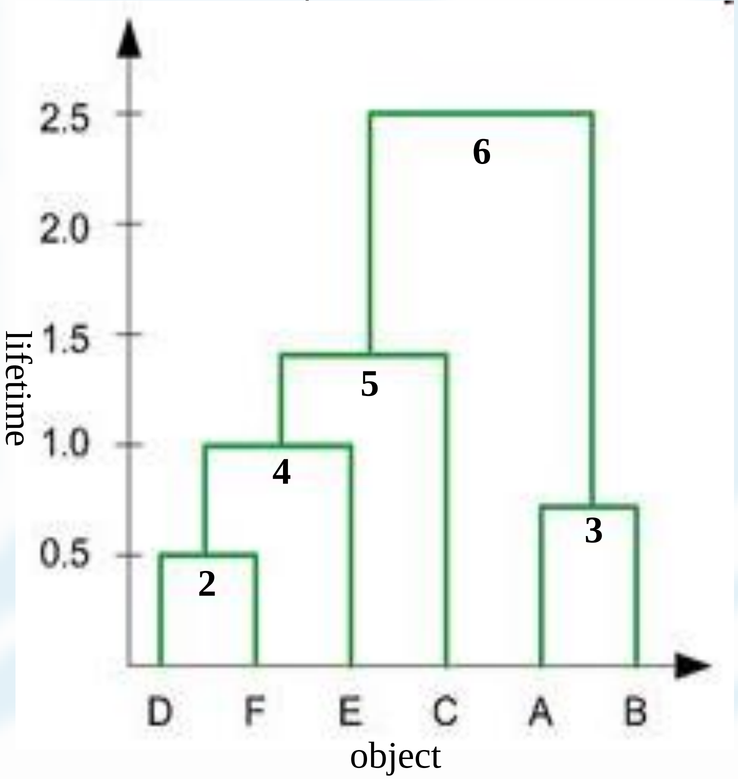


Key Concepts in Hierarchal Clustering

❖ Dendrogram Tree Representation

1. In the beginning we have 6 clusters: A, B, C, D, E and F
2. We merge clusters D and F into cluster (D, F) at distance 0.50
3. We merge cluster A and cluster B into (A, B) at distance 0.71
4. We merge clusters E and (D, F) into ((D, F), E) at distance 1.00
5. We merge clusters ((D, F), E) and C into (((D, F), E), C) at distance 1.41
6. We merge clusters (((D, F), E), C) and (A, B) into ((((D, F), E), C), (A, B)) at distance 2.50
7. The last cluster contain all the objects, thus conclude the computation





Key Concepts in Hierarchical Clustering

❖ Lifetime vs K-cluster Lifetime

- **Lifetime**

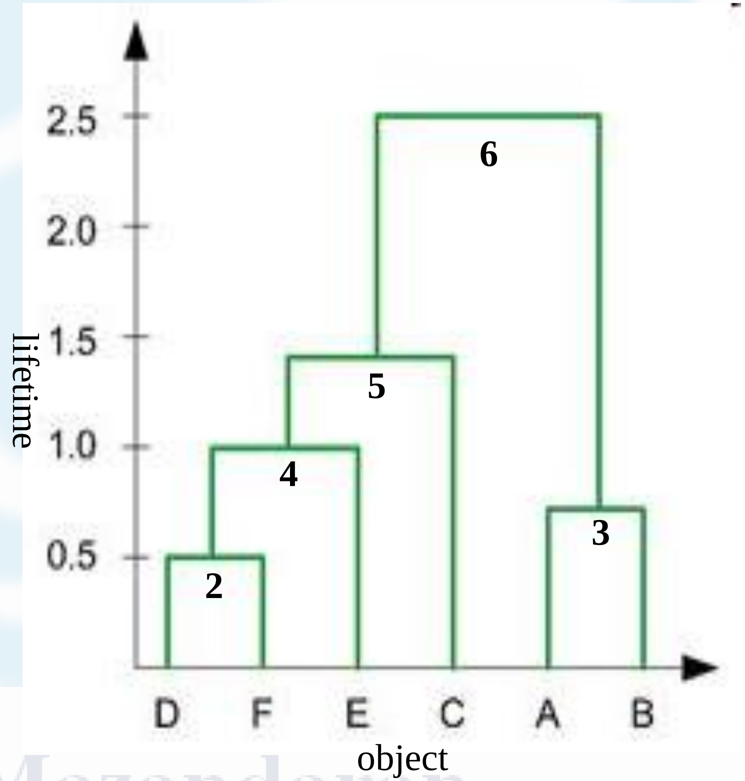
  The distance between that a cluster is created and that it disappears (merges with other clusters during clustering).

  e.g. lifetime of A, B, C, D, E and F are 0.71, 0.71, 1.41, 0.50, 1.00 and 0.50, respectively, the life time of (A, B) is 2.50 – 0.71 = 1.79, ……

- **K-cluster Lifetime**

  The distance from that K clusters emerge to that K clusters vanish (due to the reduction to K-1 clusters). e.g.

  5-cluster lifetime is 0.71 - 0.50 = 0.21
  4-cluster lifetime is 1.00 - 0.71 = 0.29
  3-cluster lifetime is 1.41 – 1.00 = 0.41
  2-cluster lifetime is 2.50 – 1.41 = 1.09



Key Concepts in Hierarchical Clustering

- How to determine the number of clusters
  - If the number of clusters known, termination condition is given!
  - The *K*-cluster lifetime as the range of threshold value on the dendrogram tree that leads to the identification of *K* clusters
  - Heuristic rule: cut a dendrogram tree with maximum life time to find a "proper" *K*
- Major weakness of agglomerative clustering methods
  - Can never undo what was done previously
  - Sensitive to cluster distance measures and noise/outliers
  - Less efficient: $O(n^2 \log n)$, where n is the number of total objects
- There are several ***variants*** to overcome its weaknesses
  - BIRCH: scalable to a large data set
  - ROCK: clustering categorical data
  - CHAMELEON: hierarchical clustering using dynamic modelling

References

[1] Hierarchical and Ensemble Clustering, COMP24111, The university of Manchester.