Hierarchical Clustering

Hierarchical Clustering

- Hierarchical clustering is another unsupervised machine learning algorithm.
- The hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the dendrogram.
- There is no requirement to predetermine the number of clusters.

Introduction

Hierarchical Clustering Approach Types

- Agglomerative: Agglomerative is a bottom-up approach, in which the algorithm starts by taking all data points as single clusters and merging them until one cluster is left.
- Divisive: Divisive algorithm is the reverse of the agglomerative algorithm as it is a top-down approach.

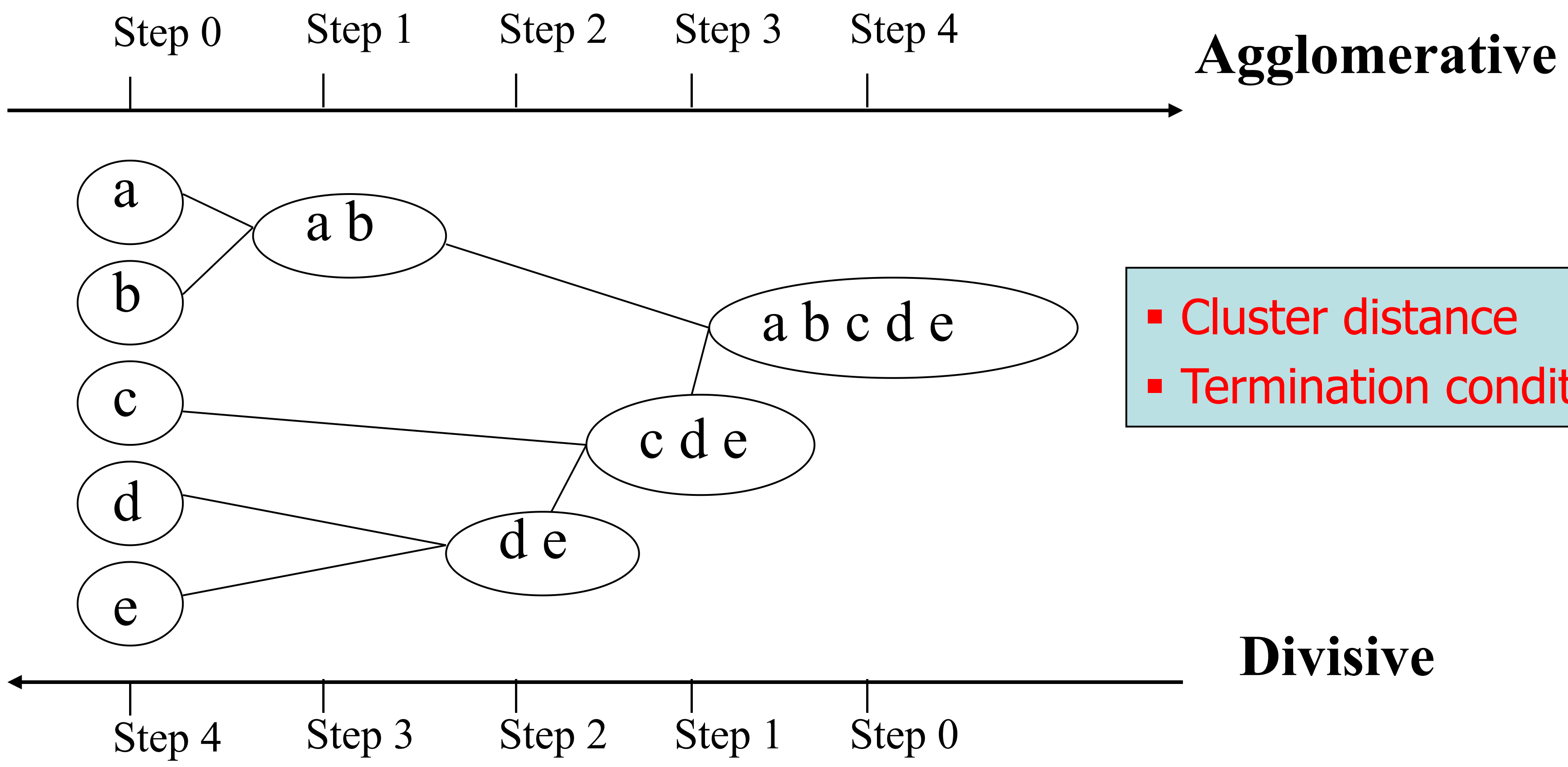
Introduction

- Agglomerative vs. Divisive
  - Agglomerative: a bottom-up strategy
    - Initially each data object is in its own (atomic) cluster
    - Then merge these atomic clusters into larger and larger clusters
  - Divisive: a top-down strategy
    - Initially all objects are in one single cluster
    - Then the cluster is subdivided into smaller and smaller clusters
- Clustering Ensemble
  - Using multiple clustering results for robustness and overcoming weaknesses of single clustering algorithms.

Introduction: Illustration

- Illustrative Example: Agglomerative (AGNES) vs. Divisive (DIANA)

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

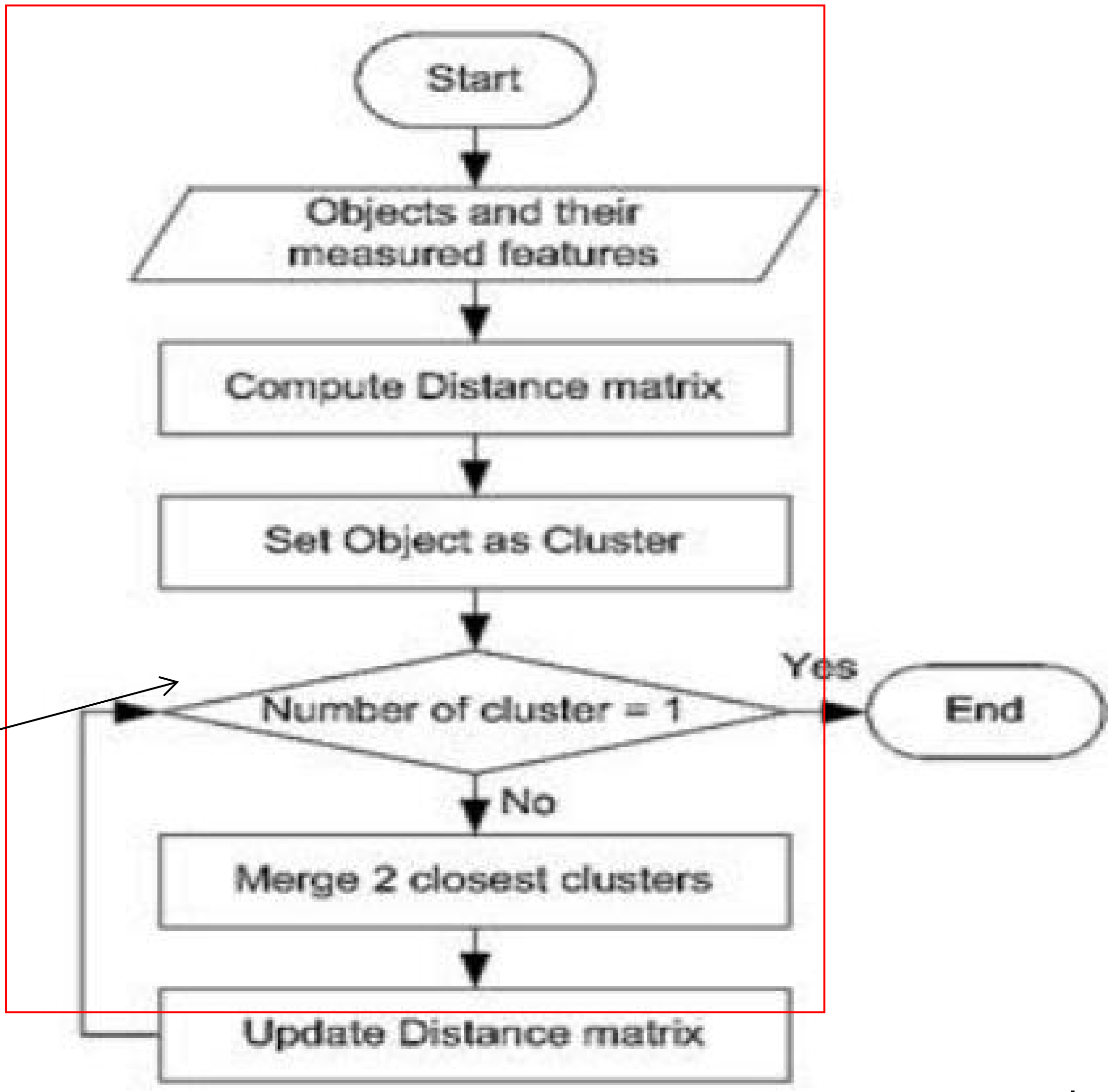


- Cluster distance
- Termination condition

Agglomerative Algorithm

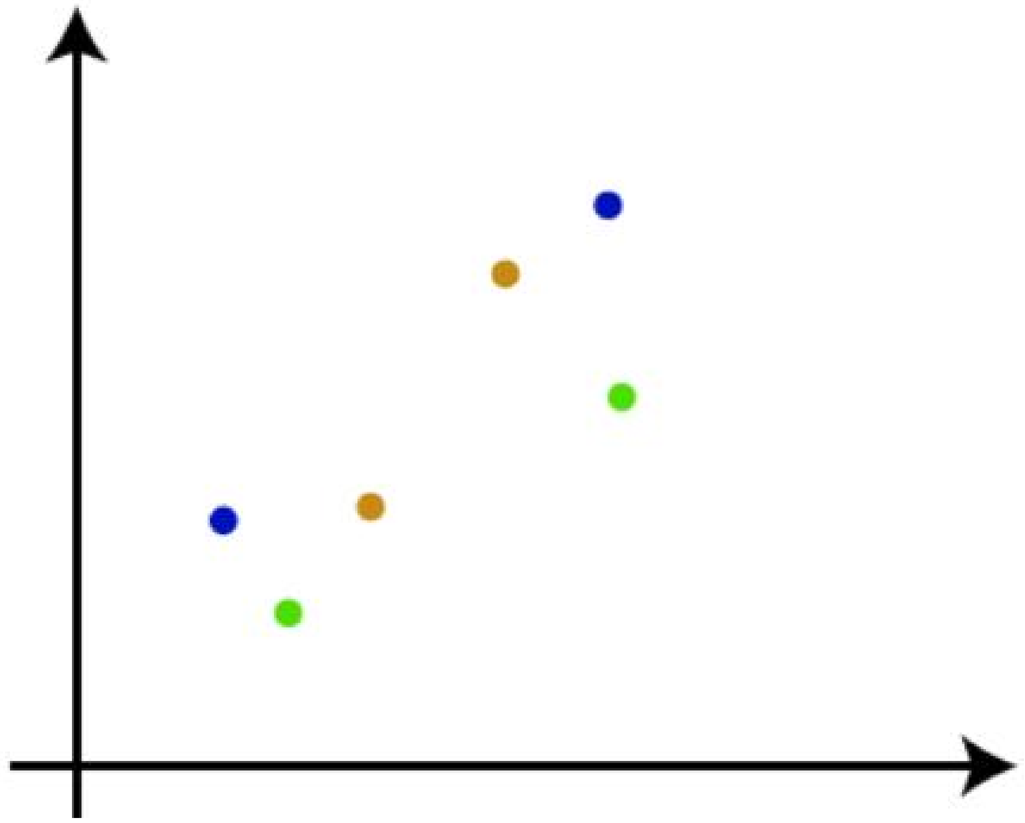
- The *Agglomerative* algorithm is carried out in three steps:

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"



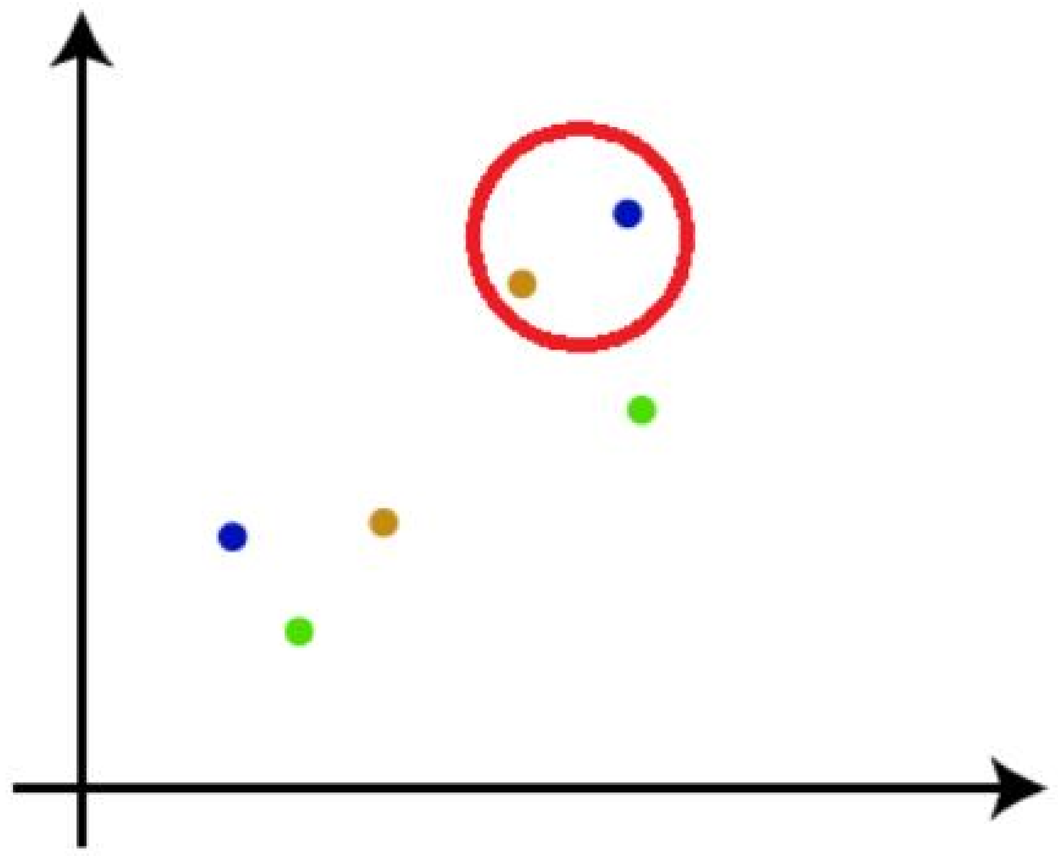
Agglomerative Algorithm

- Step-1: Create each data point as a single cluster. Let's say there are N data points, so the number of clusters will also be N.



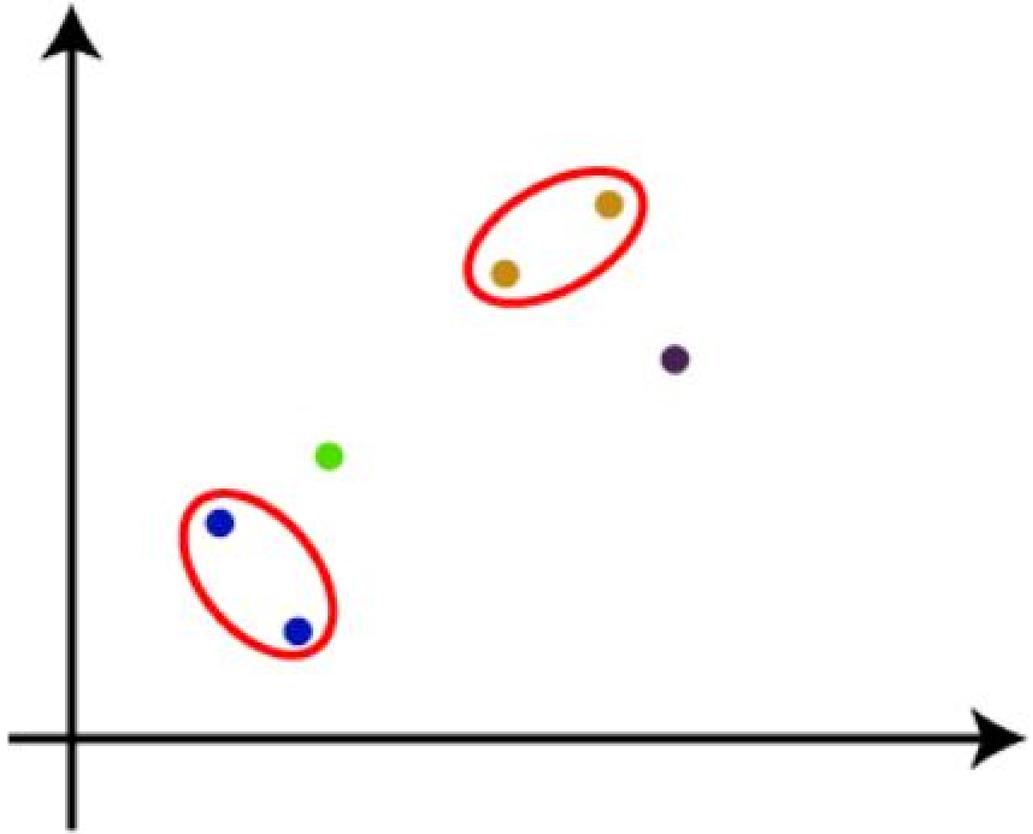
Agglomerative Algorithm

- Step-2: Take two closest data points or clusters and merge them to form one cluster. So, there will now be N-1 clusters.



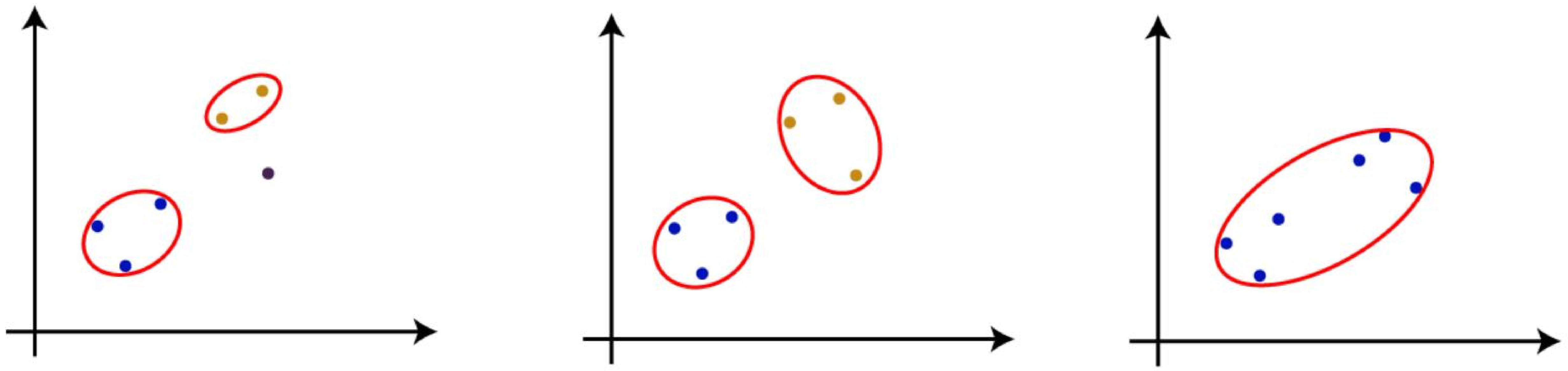
Agglomerative Algorithm

- **Step-3**: Again, take the two closest clusters and merge them together to form one cluster. There will be N-2 clusters.



Agglomerative Algorithm

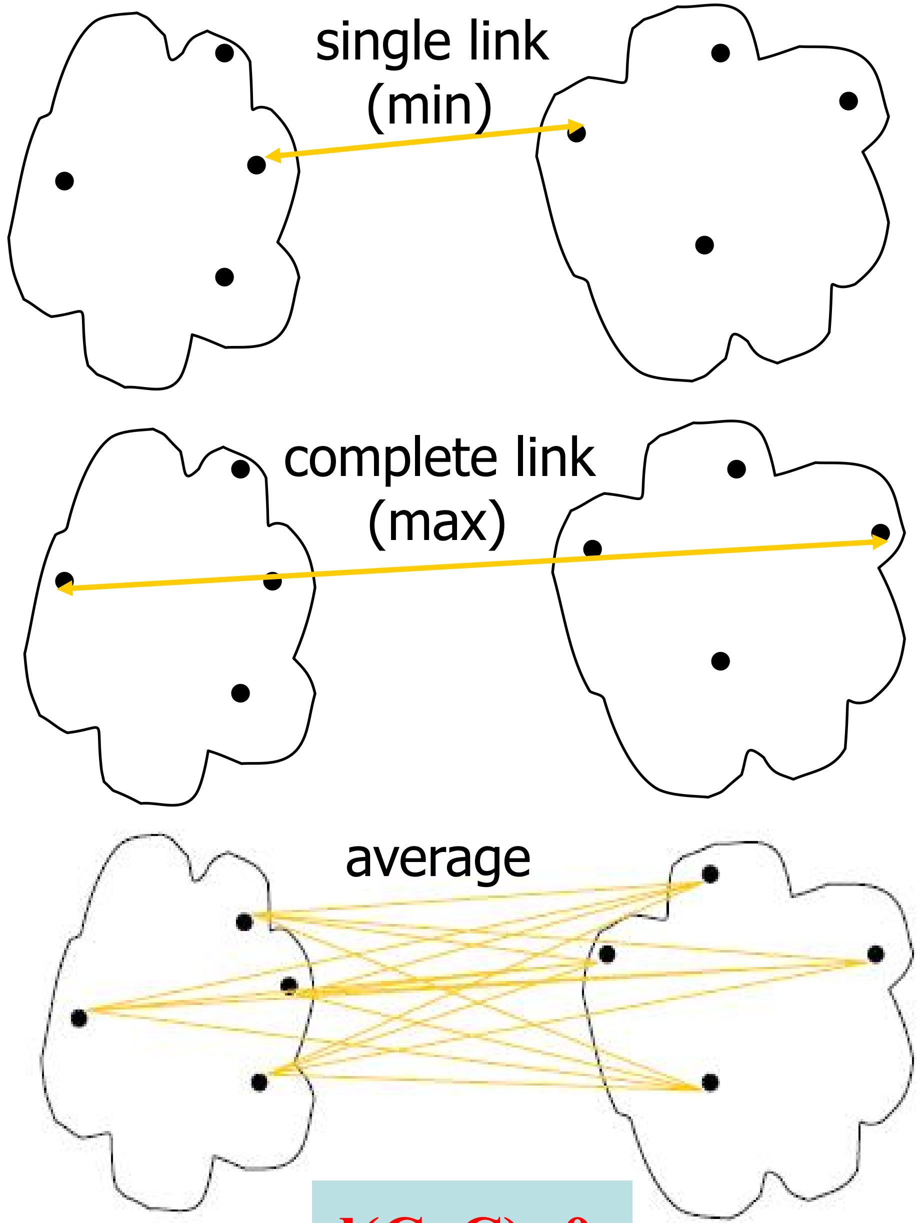
Step-4: Repeat Step 3 until only one cluster left. So, we will get the following clusters.



•**Step-5:** Once all the clusters are combined into one big cluster, develop the dendrogram to divide the clusters as per the problem.

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., $d(C_i, C_j) = \text{avg}\{d(x_{ip}, x_{jq})\}$



$d(C, C)=0$

Cluster Distance Measures

Example: Given a data set of five objects characterised 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 .

	a	b	c	d	e
a	0	1	3	4	5
b	1	0	2	3	4
c	3	2	0	1	2
d	4	3	1	0	1
e	5	4	2	1	0

2. Calculate three cluster distances between C1 and C2.

Single link

$$\begin{aligned}\mathrm{dist}(\mathrm{C}_1,\mathrm{C}_2) &= \min\{d(\mathrm{a},\mathrm{c}), d(\mathrm{a},\mathrm{d}), d(\mathrm{a},\mathrm{e}), d(\mathrm{b},\mathrm{c}), d(\mathrm{b},\mathrm{d}), d(\mathrm{b},\mathrm{e})\} \\ &= \min\{3, 4, 5, 2, 3, 4\} = 2\end{aligned}$$

Complete link

$$\begin{aligned}\mathrm{dist}(\mathrm{C}_1,\mathrm{C}_2) &= \max\{d(\mathrm{a},\mathrm{c}), d(\mathrm{a},\mathrm{d}), d(\mathrm{a},\mathrm{e}), d(\mathrm{b},\mathrm{c}), d(\mathrm{b},\mathrm{d}), d(\mathrm{b},\mathrm{e})\} \\ &= \max\{3, 4, 5, 2, 3, 4\} = 5\end{aligned}$$

Average

$$\begin{aligned}\mathrm{dist}(\mathrm{C}_1,\mathrm{C}_2) &= \frac{d(\mathrm{a},\mathrm{c}) + d(\mathrm{a},\mathrm{d}) + d(\mathrm{a},\mathrm{e}) + d(\mathrm{b},\mathrm{c}) + d(\mathrm{b},\mathrm{d}) + d(\mathrm{b},\mathrm{e})}{6} \\ &= \frac{3+4+5+2+3+4}{6} = \frac{21}{6} = 3.5\end{aligned}$$

The dendrogram in Hierarchical clustering

- The dendrogram is a tree-like structure that is mainly used to store each step as a memory that the HC algorithm performs.
- In the dendrogram plot, the Y-axis shows the Euclidean distances between the data points, and the x-axis shows all the data points of the given dataset.
- Result of hierarchical clustering can be represented **as a binary tree**:

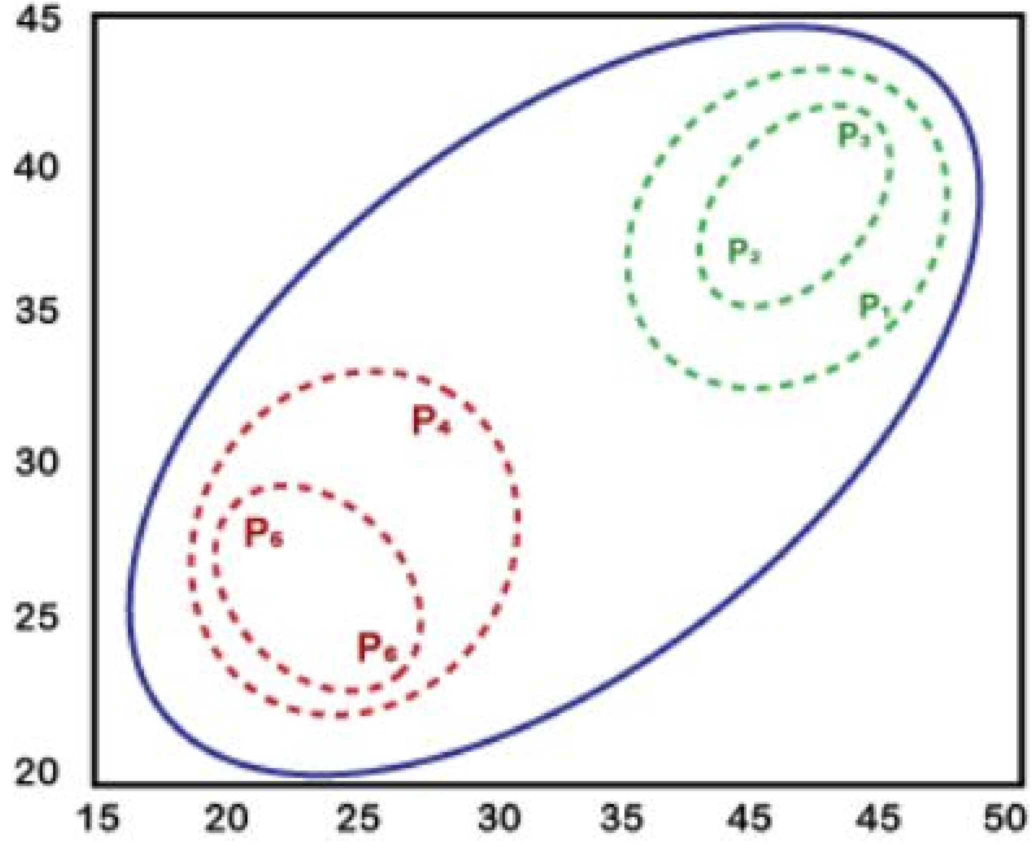
  The root of the tree represents the entire collection

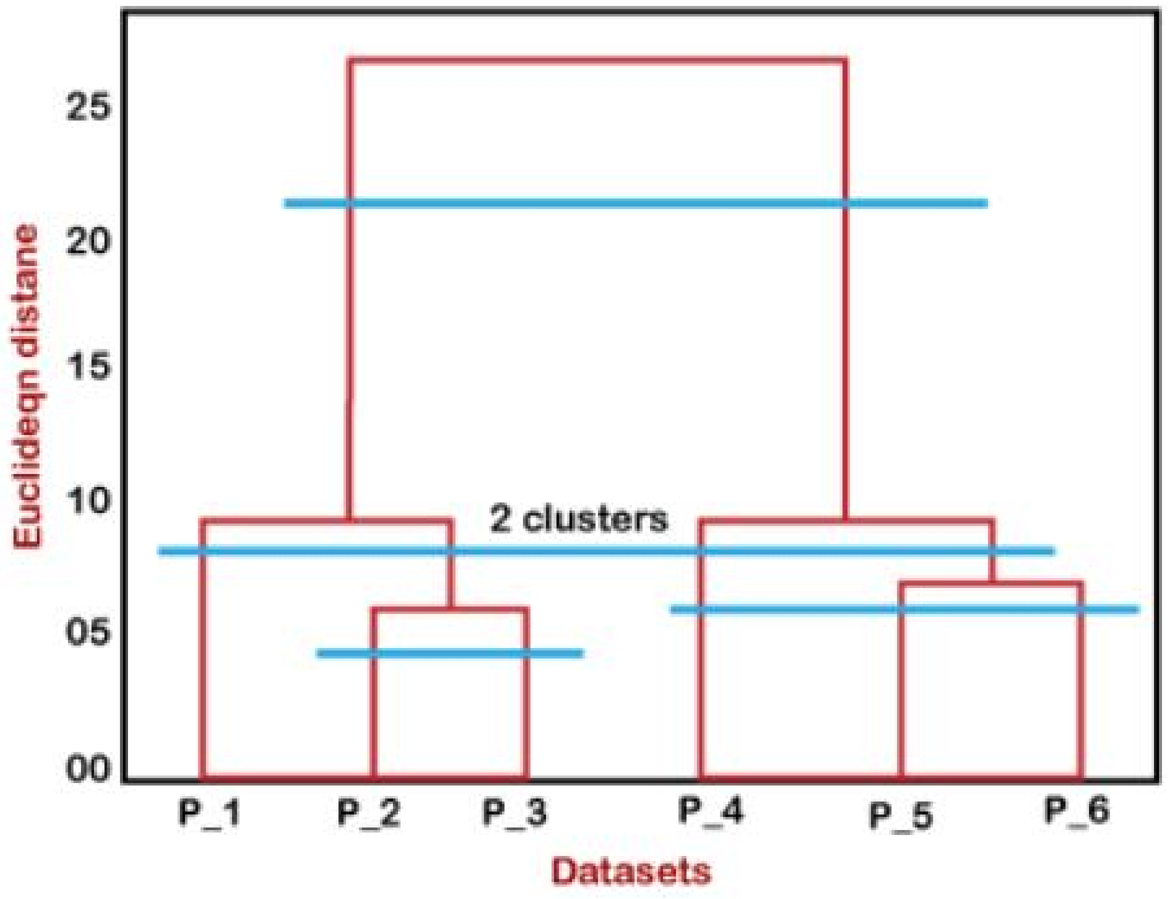
  Terminal nodes represent observations

  Each interior node represents a cluster

  Each subtree represents a partition

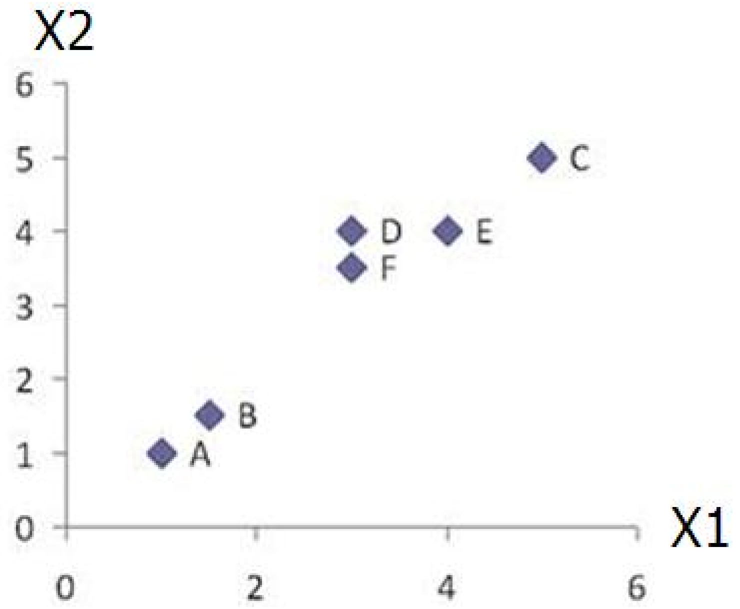
The dendrogram in Hierarchical clustering





Example

- Problem: clustering analysis with agglomerative algorithm



$$\begin{array}{c|cc} & X1 & X2 \\ \hline A & 1 & 1 \\ B & 1.5 & 1.5 \\ C & 5 & 5 \\ D & 3 & 4 \\ E & 4 & 4 \\ F & 3 & 3.5 \end{array}$$

data matrix

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

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

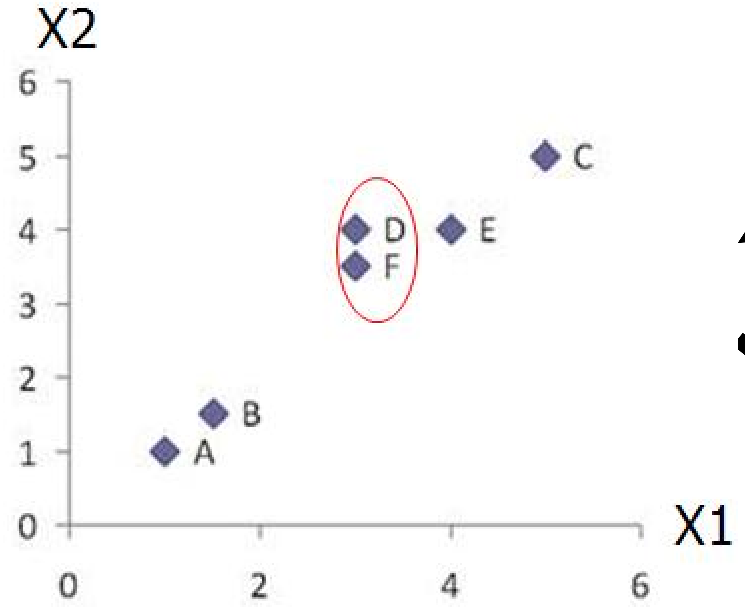
Euclidean distance

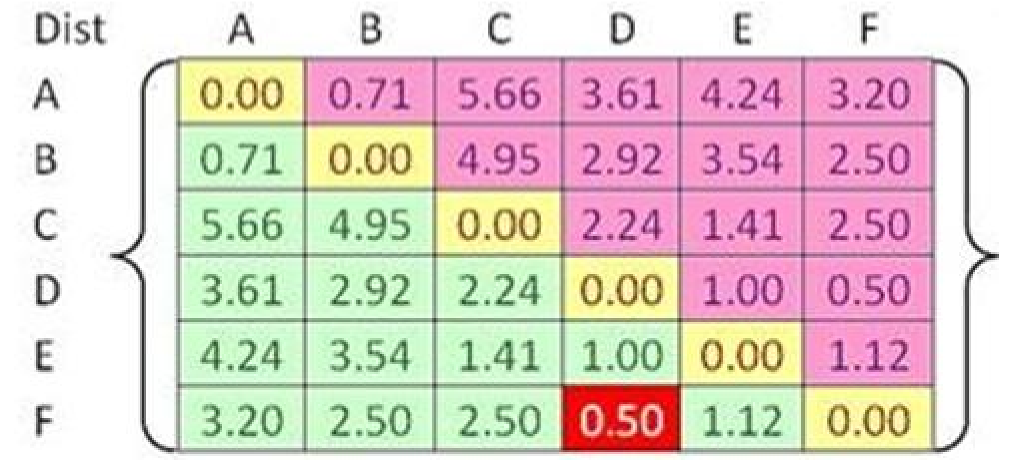
Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

distance matrix

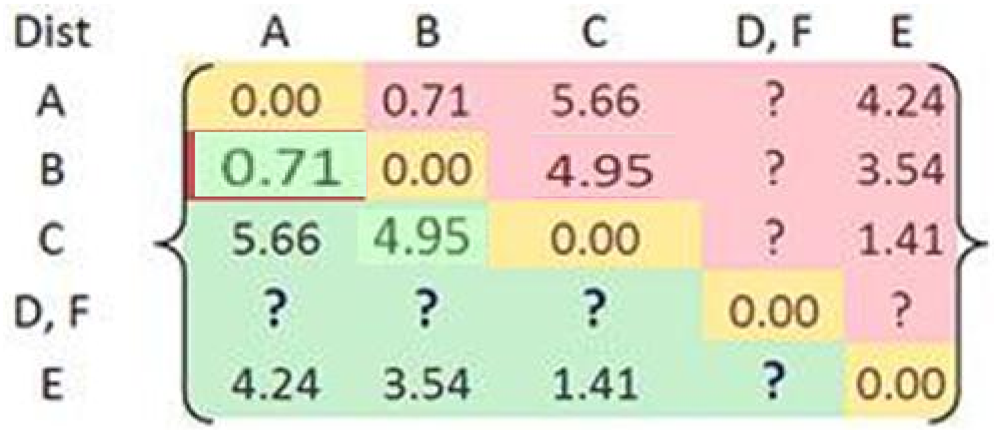
Example

- Merge two closest clusters (iteration 1)





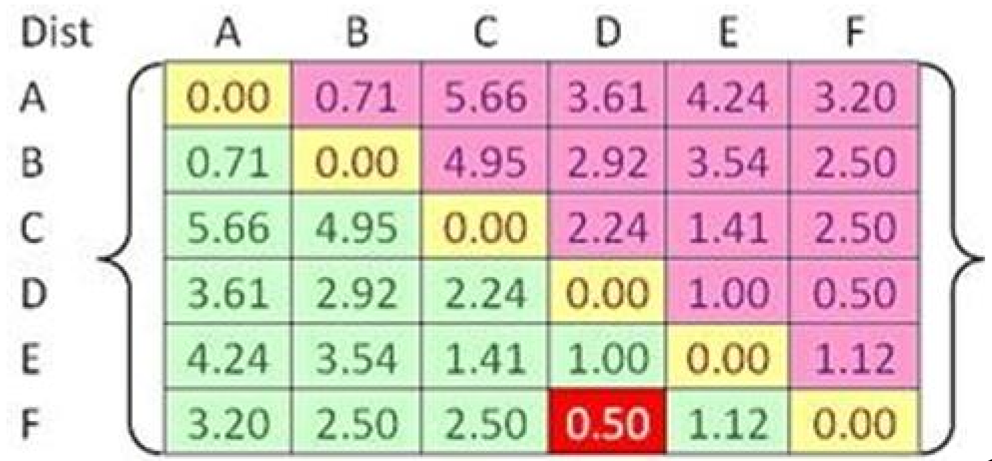
Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00



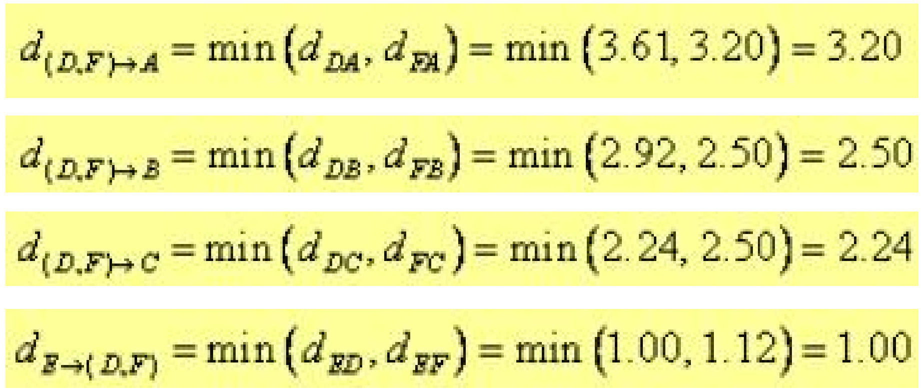
Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

Example

- Update distance matrix (iteration 1)



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

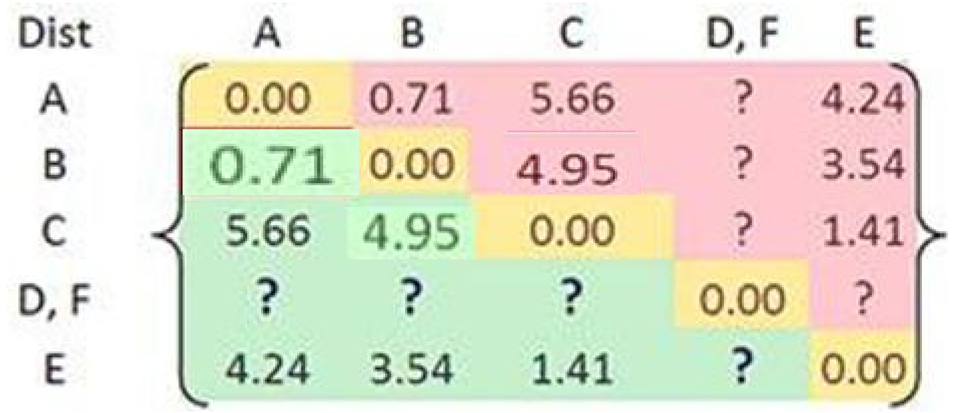


$$d_{(D,F)\mapsto A} = \min(d_{DA}, d_{FA}) = \min(3.61, 3.20) = 3.20$$

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

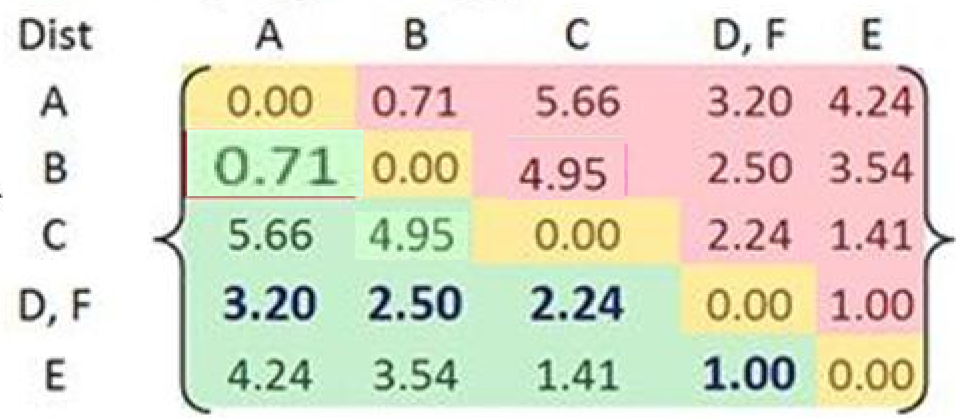
$$d_{(D,F)\mapsto C} = \min(d_{DC}, d_{FC}) = \min(2.24, 2.50) = 2.24$$

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



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

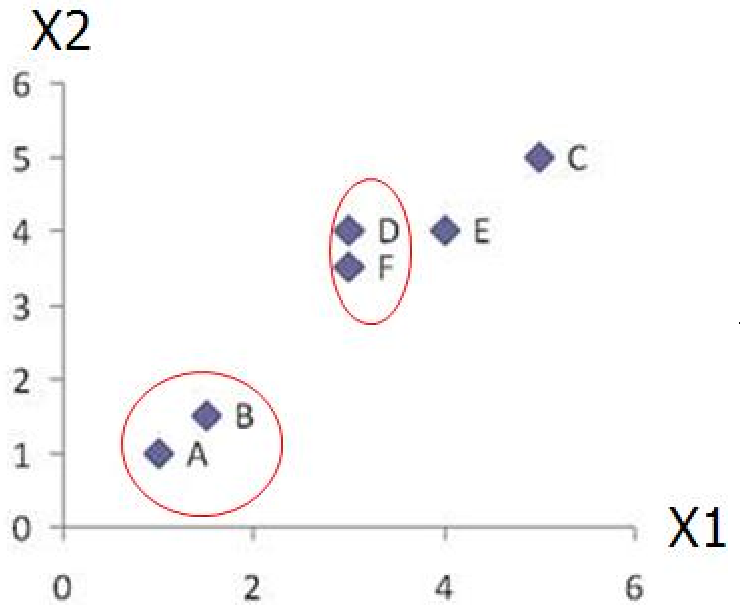
Min Distance (Single Linkage)

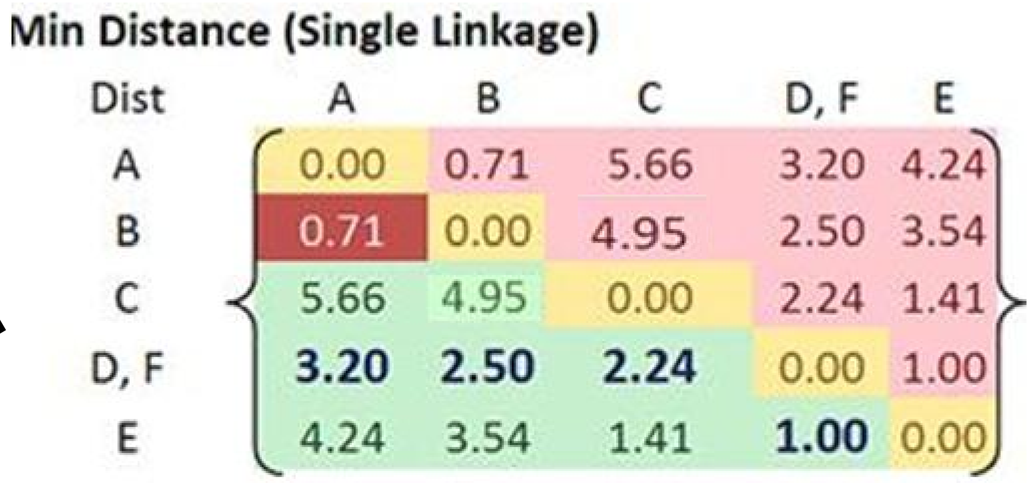


Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

Example

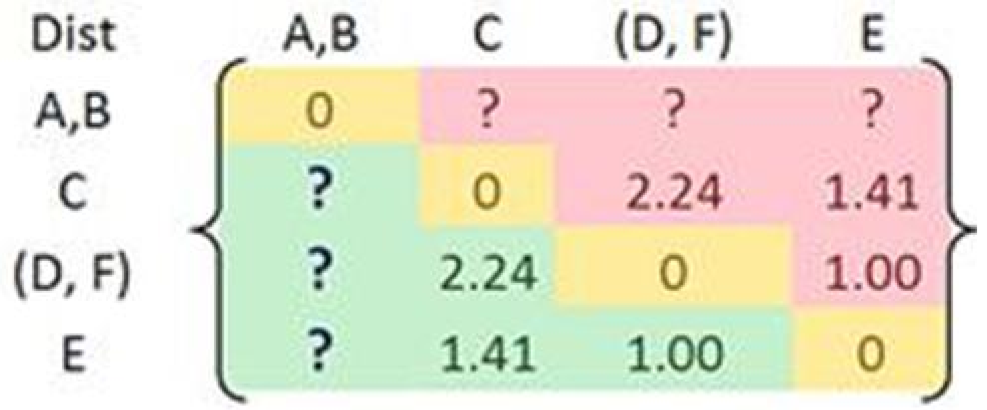
- Merge two closest clusters (iteration 2)





Min Distance (Single Linkage)

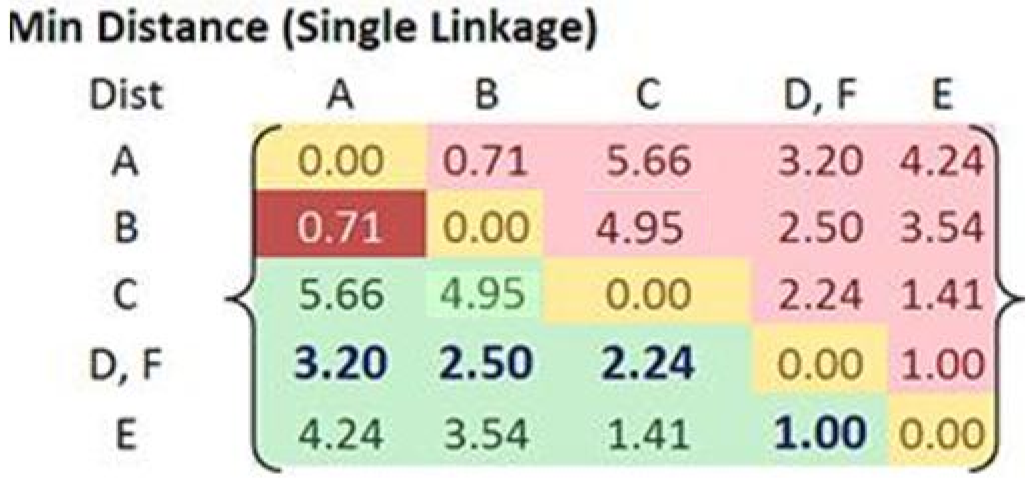
Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00



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

Example

- Update distance matrix (iteration 2)



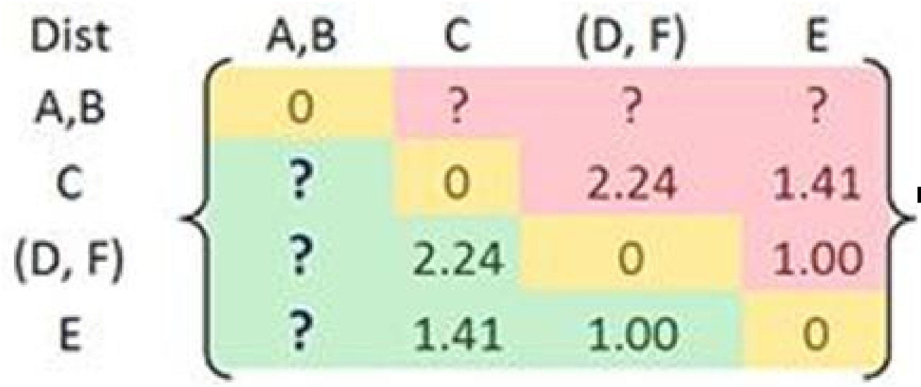
Min Distance (Single Linkage)

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

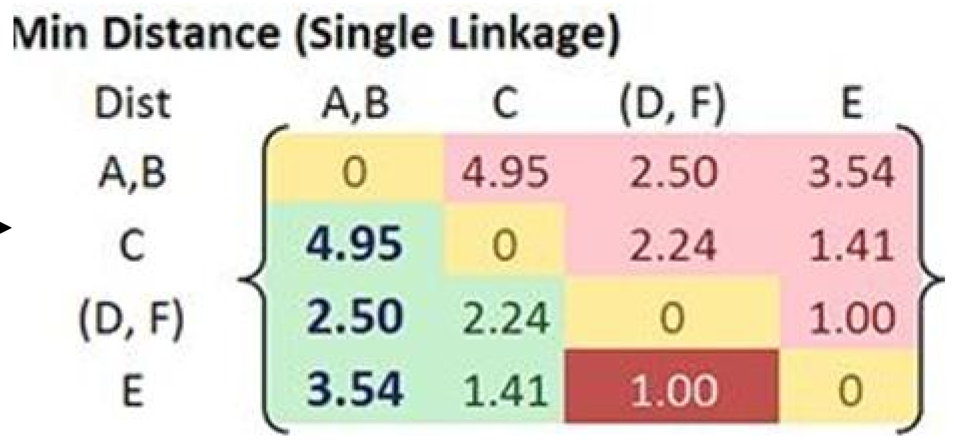
$$d_{C\to(A,B)} = \min\left(d_{CA}, d_{CB}\right) = \min\left(5.66, 4.95\right) = 4.95$$

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

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



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

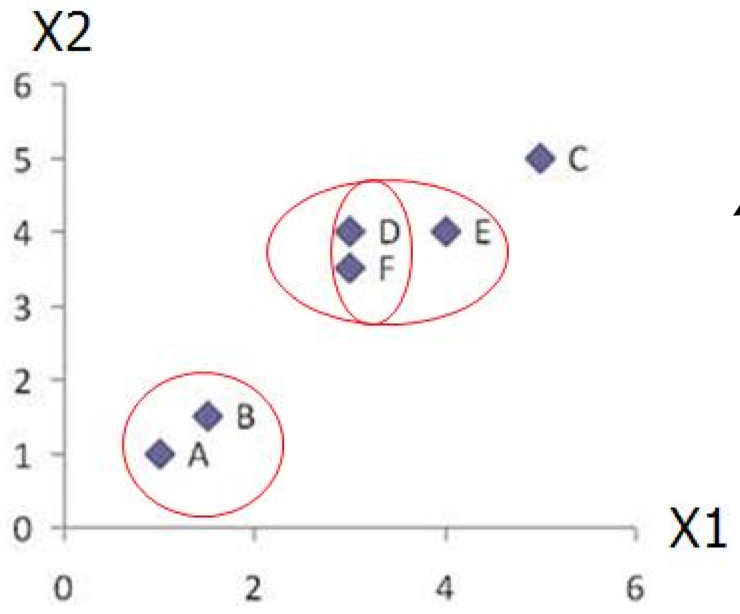


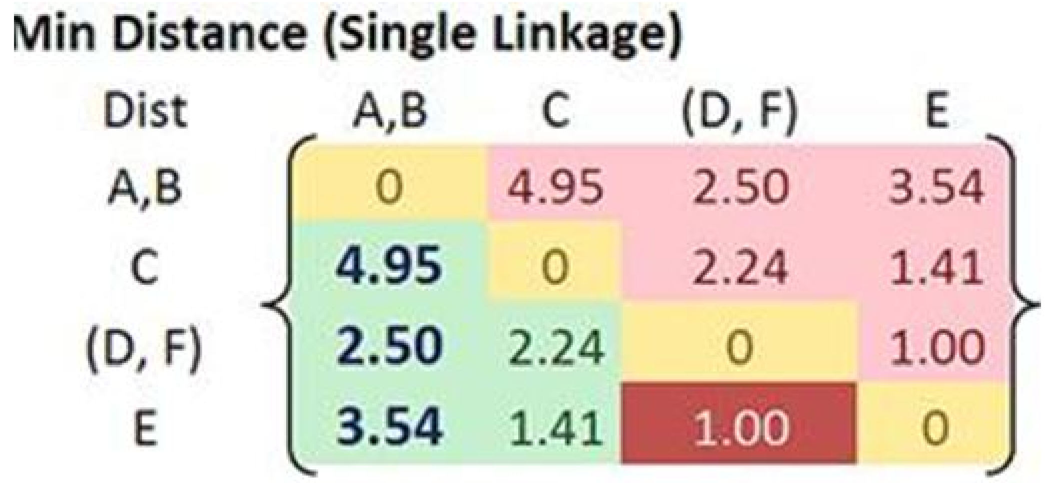
Min Distance (Single Linkage)

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

Example

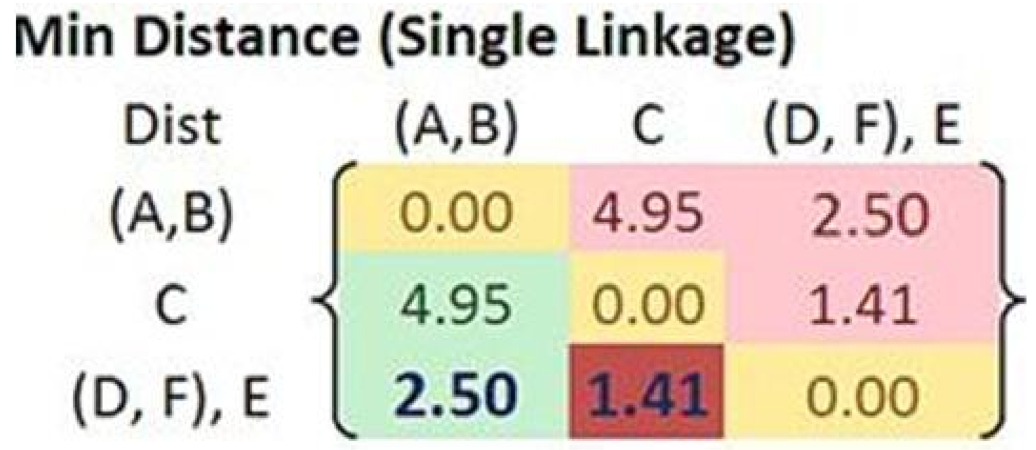
- Merge two closest clusters/update distance matrix (iteration 3)





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

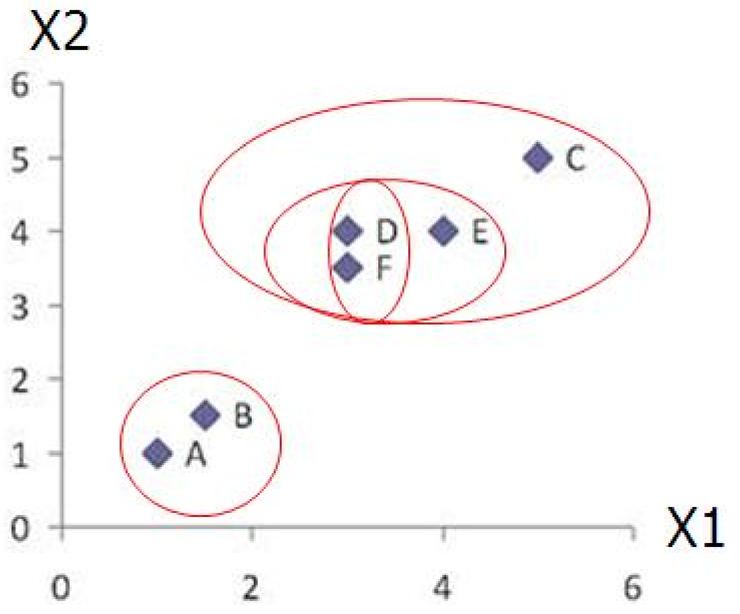


Min Distance (Single Linkage)

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

Example

- Merge two closest clusters/update distance matrix (iteration 4)



Min Distance (Single Linkage)

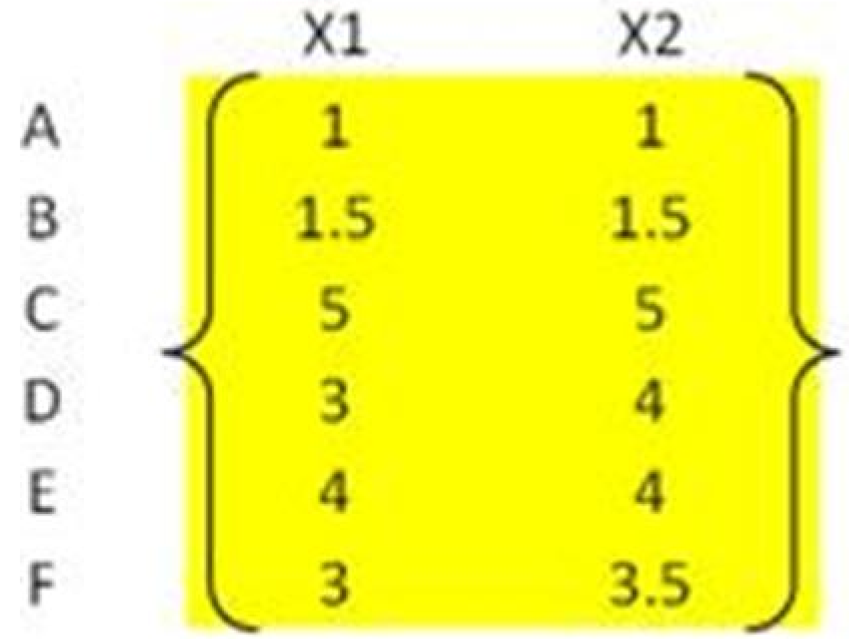
Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

Min Distance (Single Linkage)

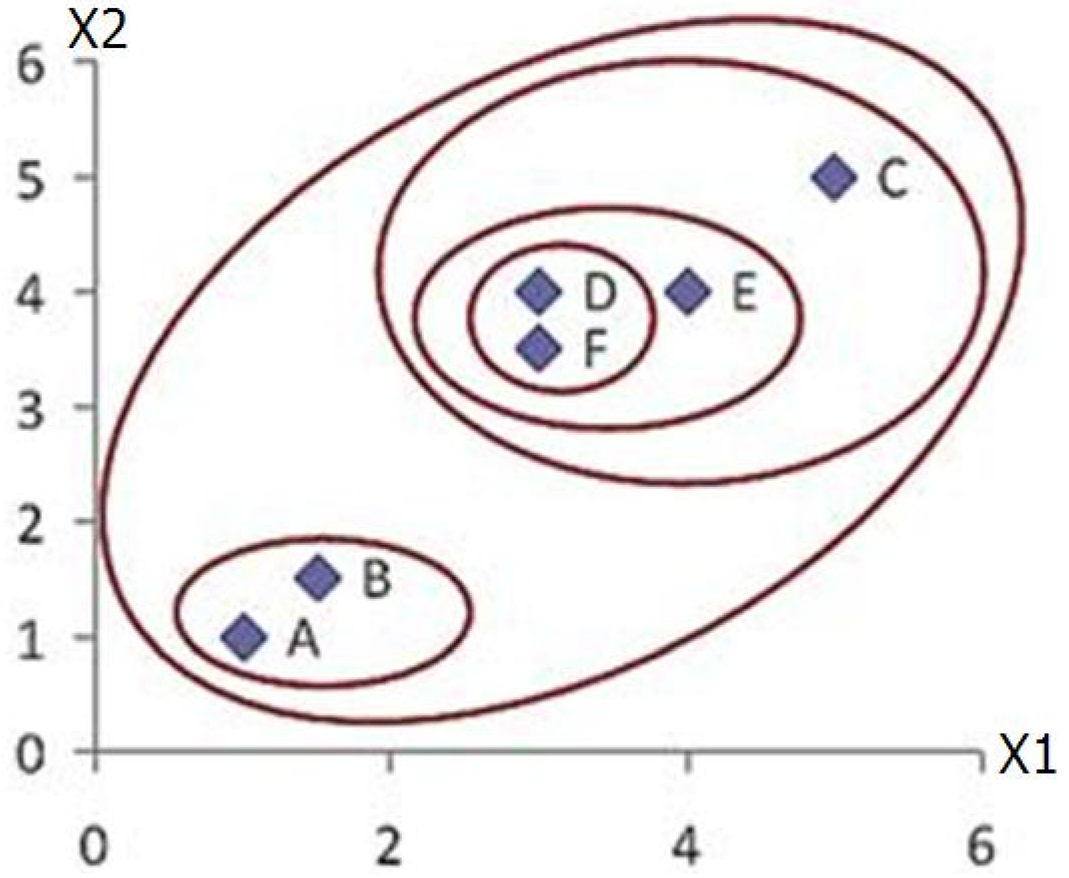
Dist	(A,B)	((D, F), E),C
(A,B)	0.00	2.50
((D, F), E),C	**2.50**	0.00

Example

- Final result (meeting termination condition)

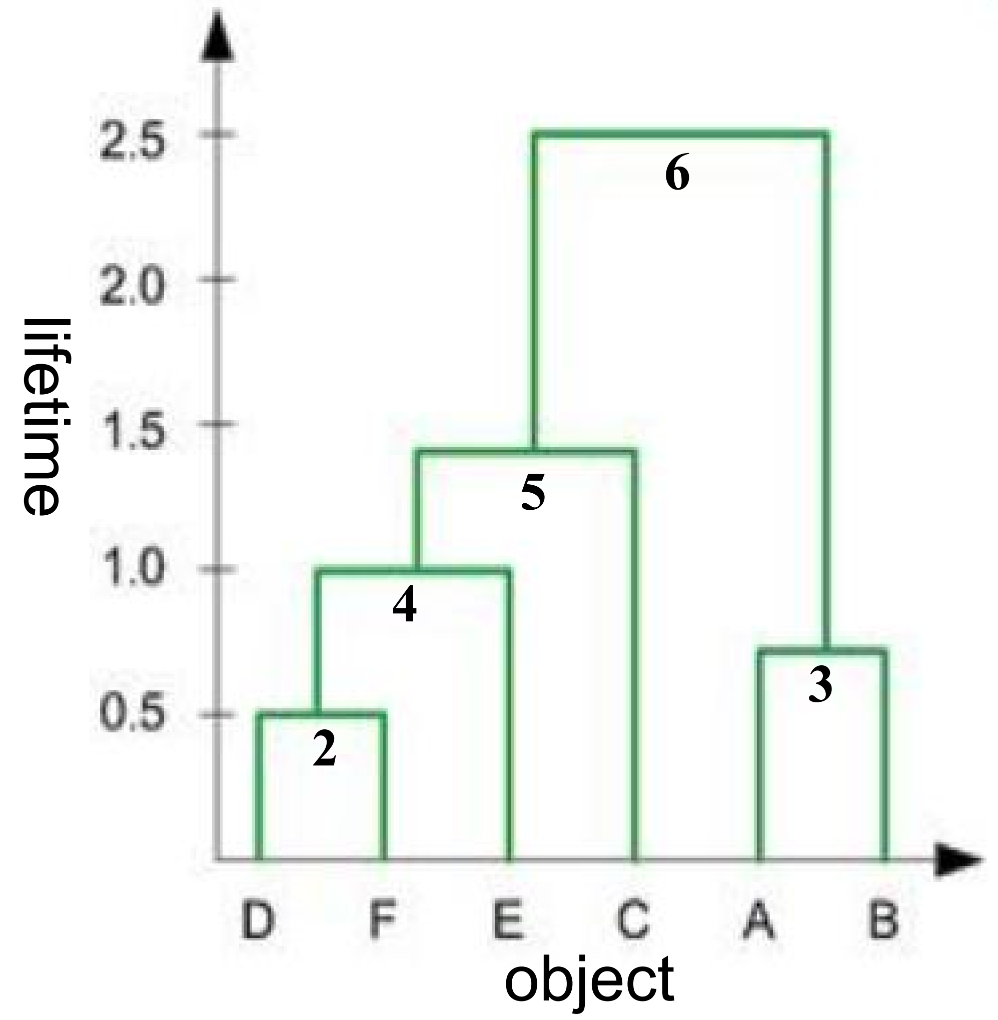


	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5



Key Concepts in Hierarchal Clustering

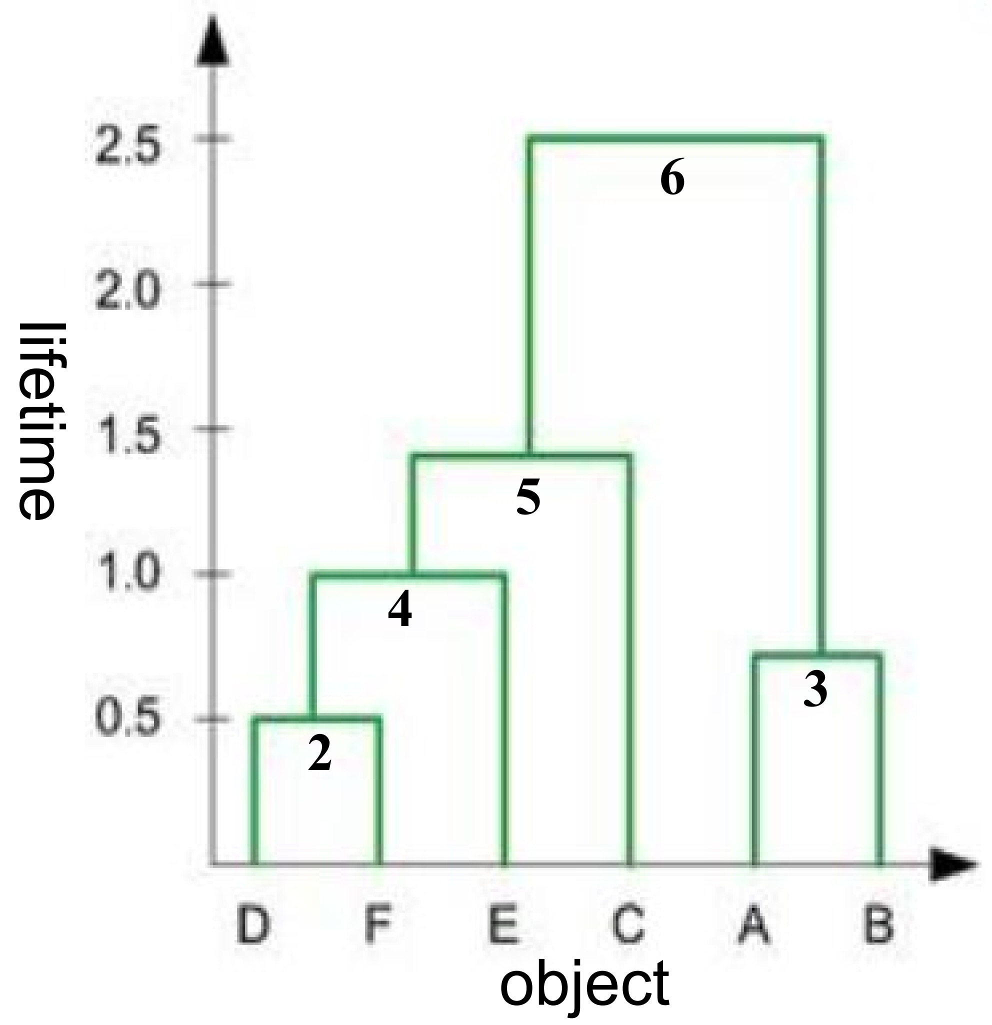
- Dendrogram tree representation



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 Hierarchal 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

Relevant Issues

- 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