LECTURE 33

Dr.Vani V

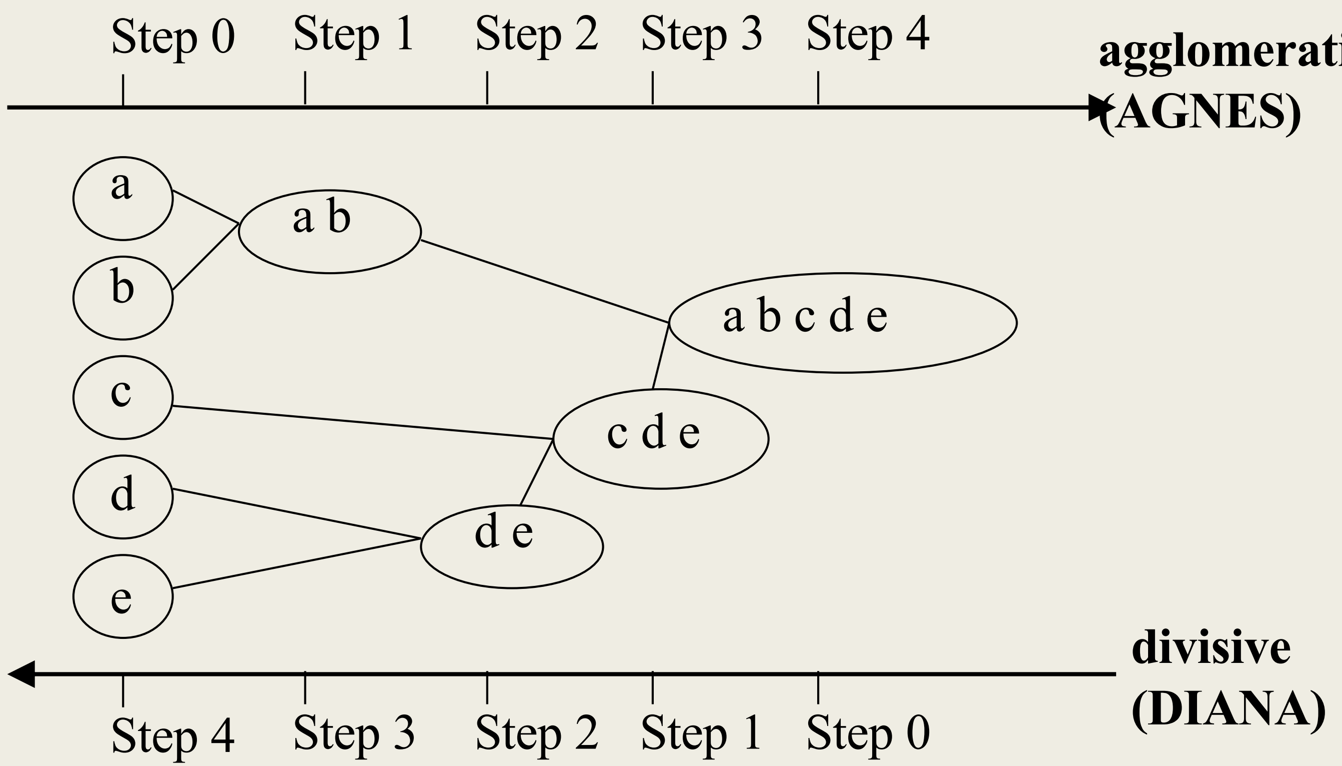
Source : Textbook 1 & Textbook 2

Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- ~~Grid-Based Methods~~
- Evaluation of Clustering
- Summary

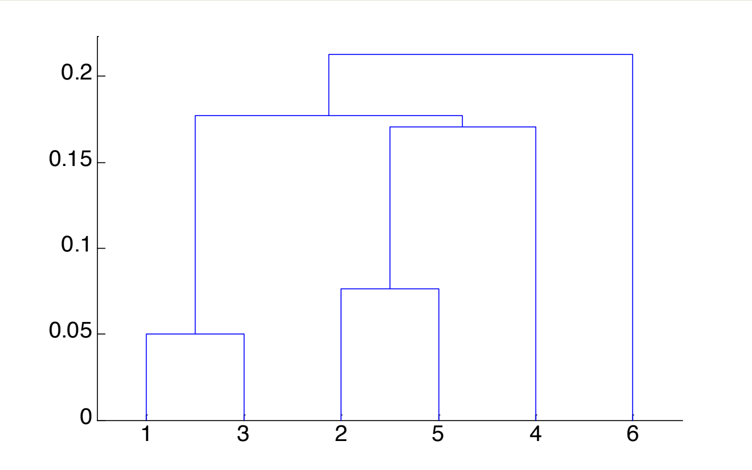
Hierarchical Clustering

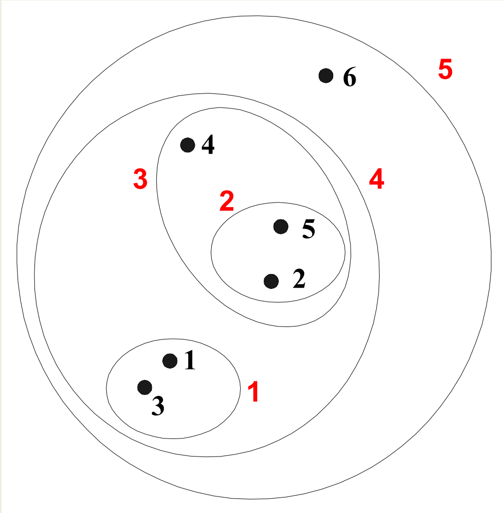
- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



Hierarchical Clustering

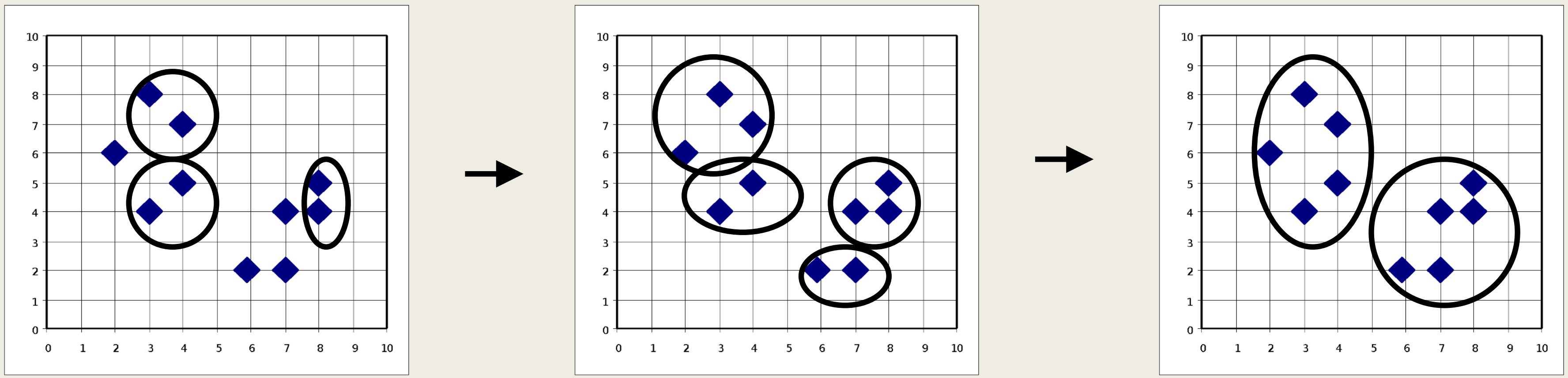
- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - *A tree like diagram that records the sequences of merges or splits*





AGNES (Agglomerative Nesting)

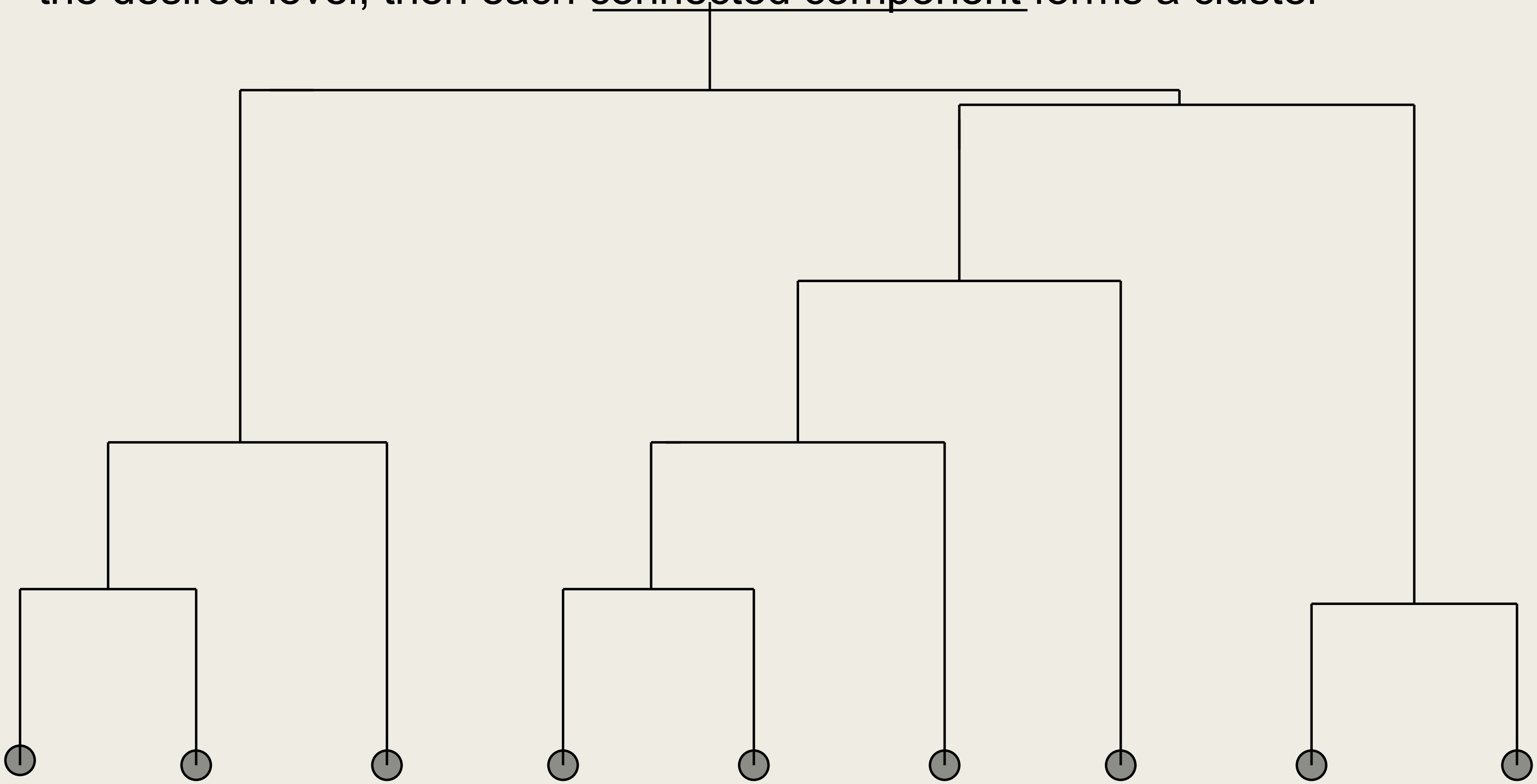
- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical packages, e.g., Splus
- Use the **single-link** method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



Dendrogram: Shows How Clusters are Merged

Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster



Agglomerative Clustering Algorithm

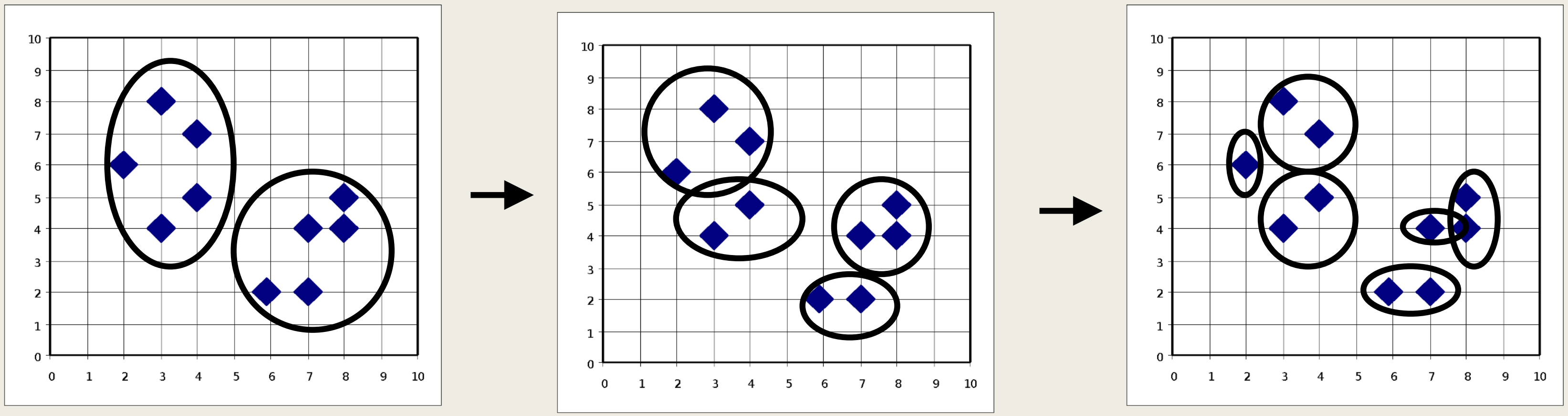
- More popular hierarchical clustering technique

- Basic algorithm is straightforward
  1. *Compute the proximity matrix*
  2. *Let each data point be a cluster*
  3. ***Repeat***
  4. *Merge the two closest clusters*
  5. *Update the proximity matrix*
  6. ***Until** only a single cluster remains*

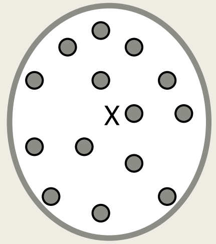
- Key operation is the computation of the proximity of two clusters
  - *Different approaches to defining the distance between clusters distinguish the different algorithms*

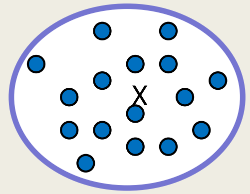
DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



Distance between Clusters



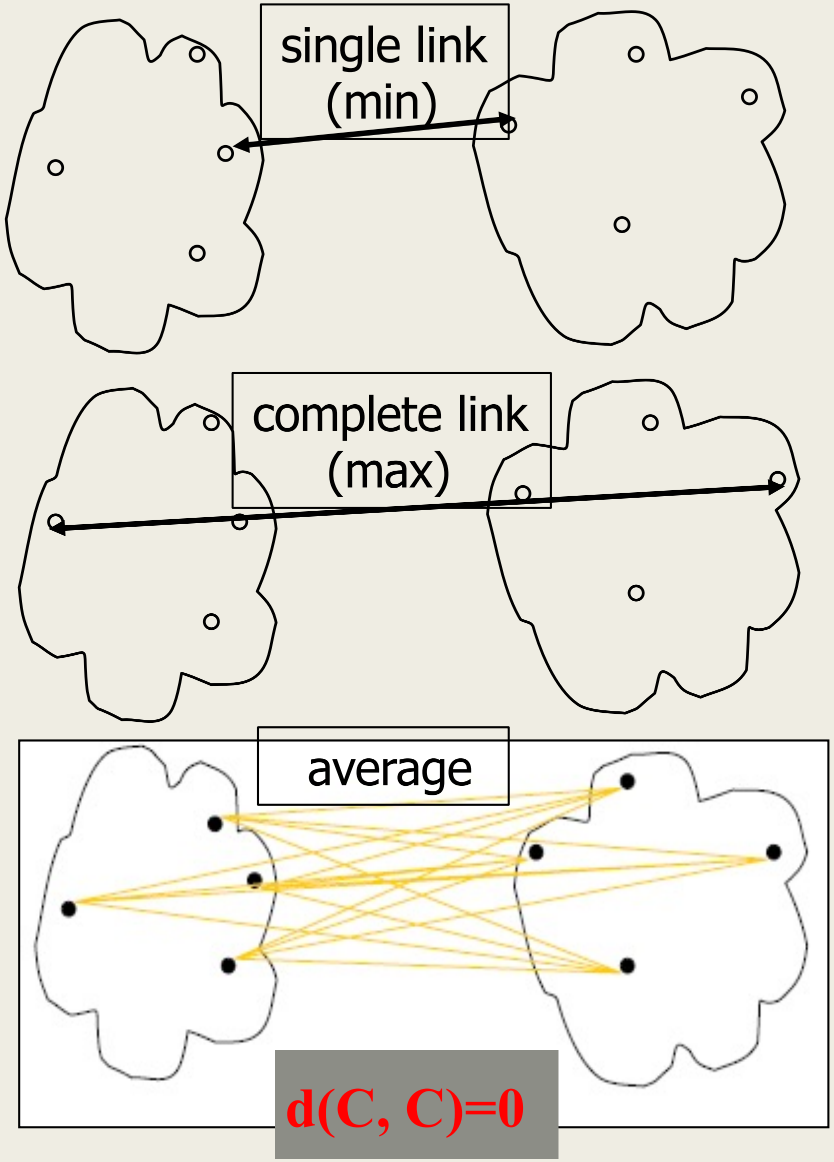


- Single link: smallest distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = min(t_{ip}, t_{jq})$
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = max(t_{ip}, t_{jq})$
- Average: avg distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = avg(t_{ip}, t_{jq})$
- Centroid: distance between the centroids of two clusters, i.e., $dist(K_i, K_j) = dist(C_i, C_j)$
- Medoid: distance between the medoids of two clusters, i.e., $dist(K_i, K_j) = dist(M_i, M_j)$
    - Medoid: a chosen, centrally located object in the cluster

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.,

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



Cluster Distance Measures

Example: *Given a data set of five objects characterised by a single continuous feature, assume that there are two clusters: C_1: {a, b} and C_2: {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 C_1 and C_2.

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

Single link

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

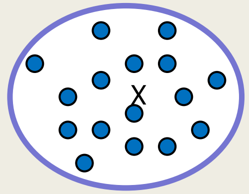
Complete link

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

Average

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

Centroid, Radius and Diameter of a Cluster (for numerical data sets)



- Centroid: the "middle" of a cluster

$$C_m = \frac{\sum_{i=1}^{N}(t_{ip})}{N}$$

- Radius: square root of average distance from any point of the cluster to its centroid

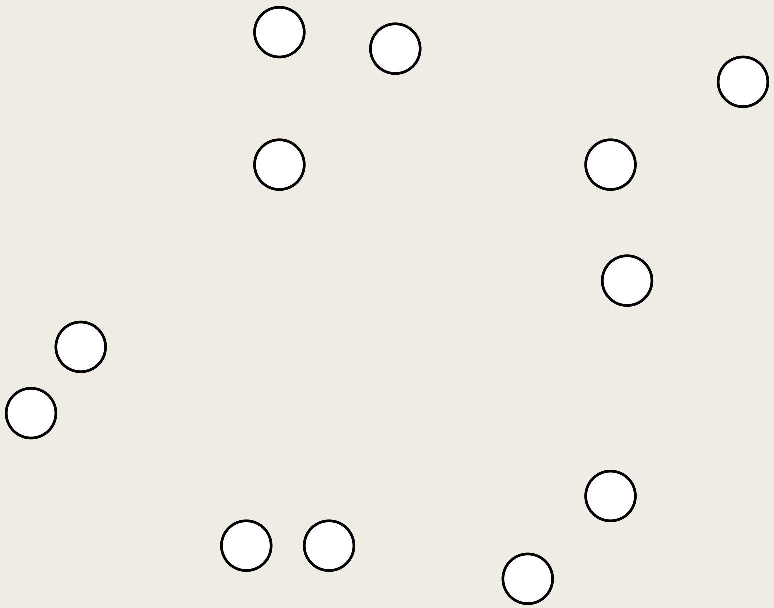
$$R_m = \sqrt{\frac{\sum_{i=1}^{N}(t_{ip} - c_m)^2}{N}}$$

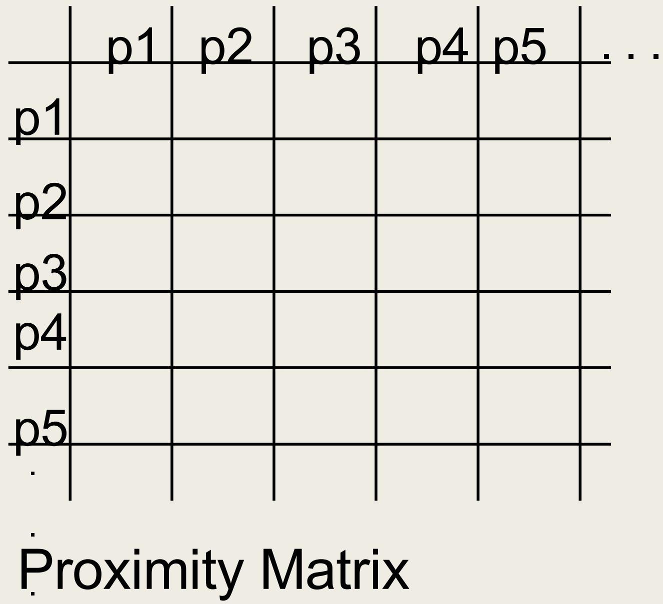
- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^{N}\sum_{i=1}^{N}(t_{ip} - t_{iq})^2}{N(N-1)}}$$

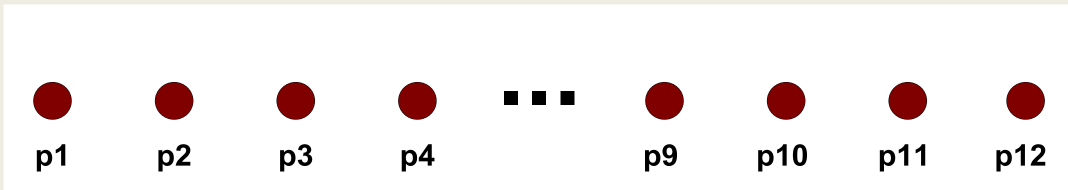
Starting Situation

- Start with clusters of individual points and a proximity matrix



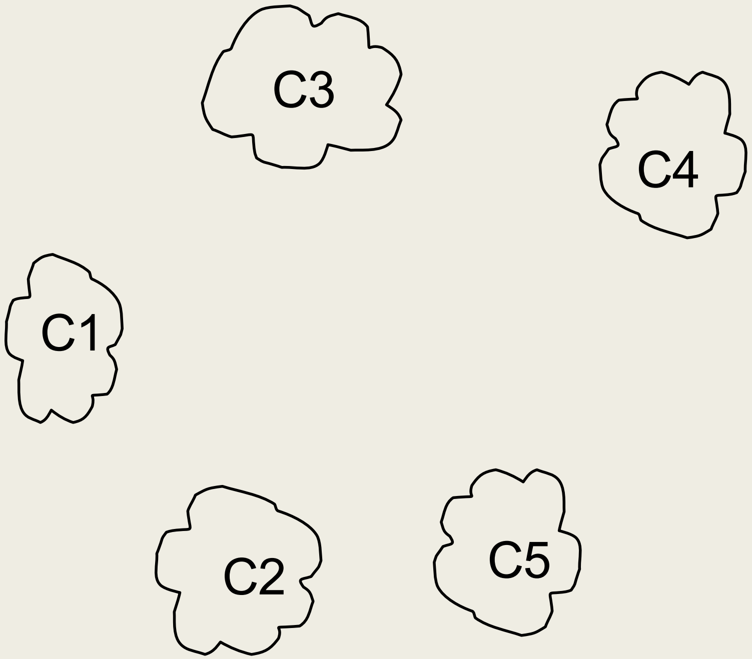


Proximity Matrix



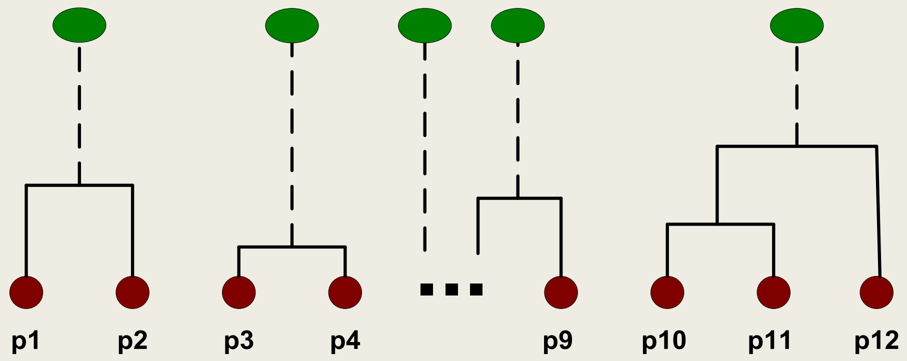
Intermediate Situation

- After some merging steps, we have some clusters



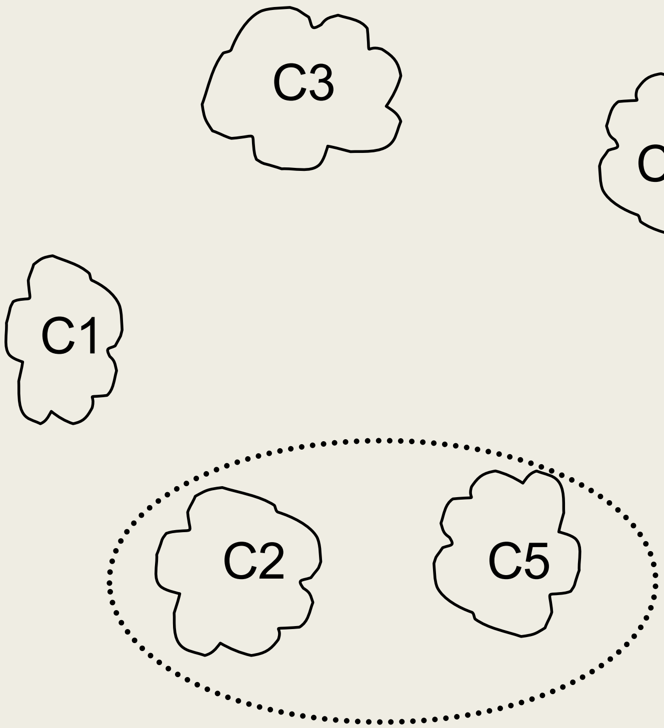
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

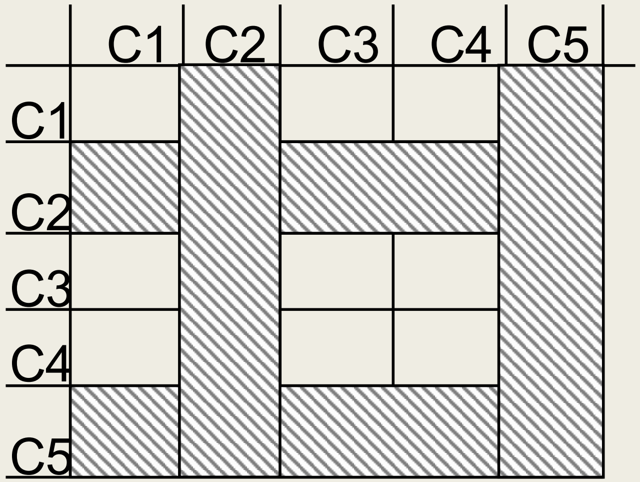
Proximity Matrix



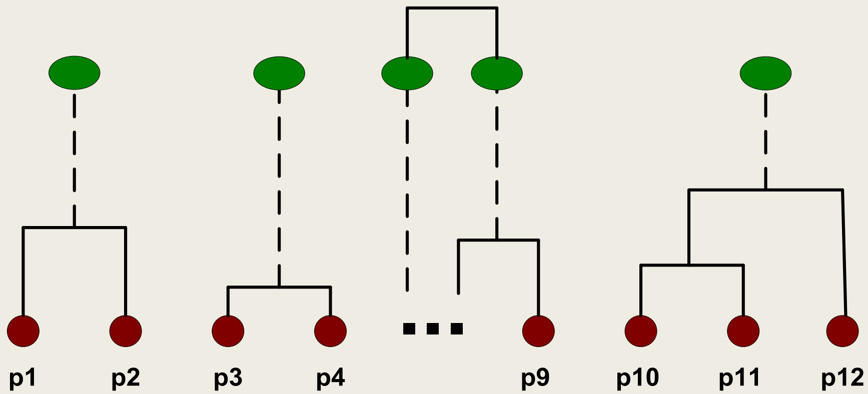
Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.





Proximity Matrix

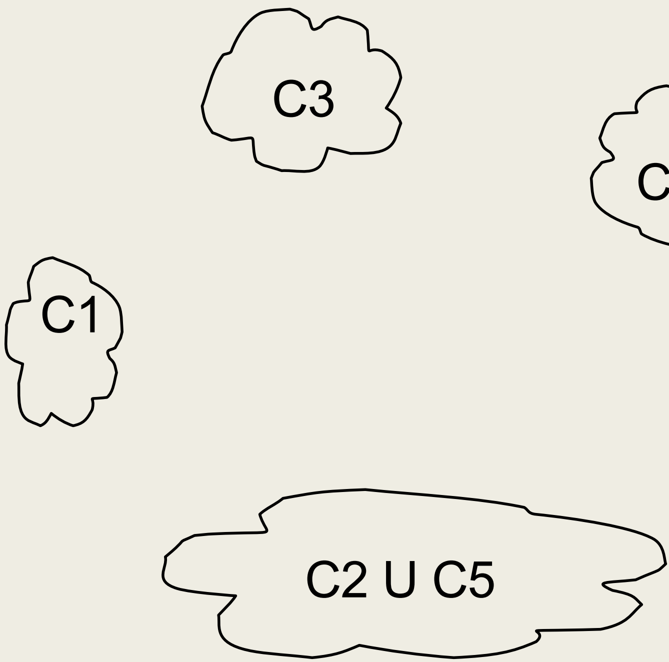


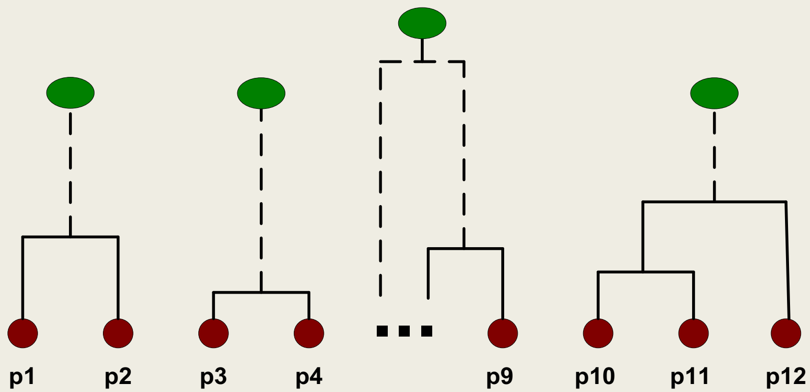
After Merging

- The question is “How do we update the proximity matrix?”

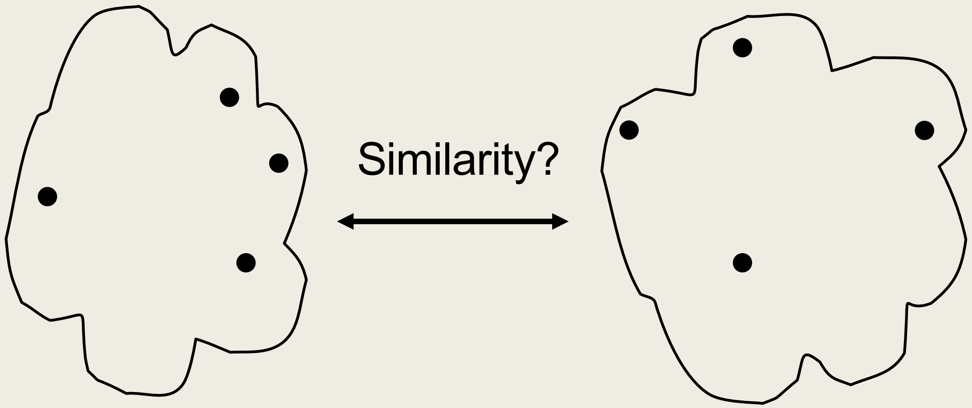
	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix





How to Define Inter-Cluster Similarity?

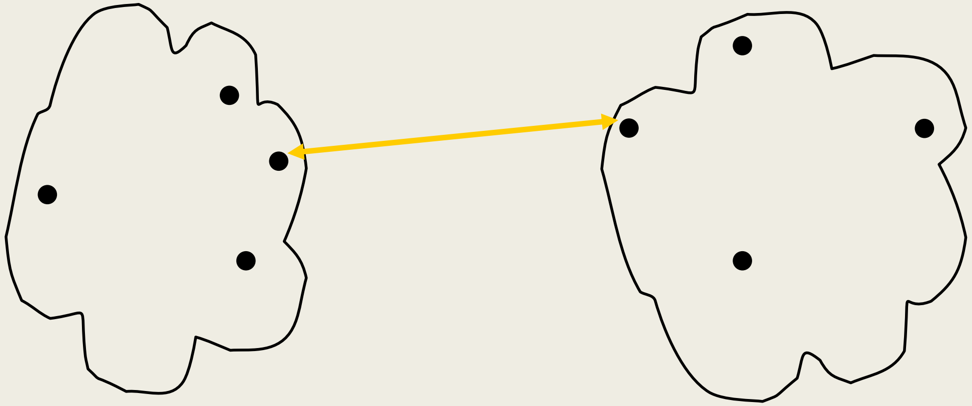


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  – Ward's Method uses squared error

How to Define Inter-Cluster Similarity

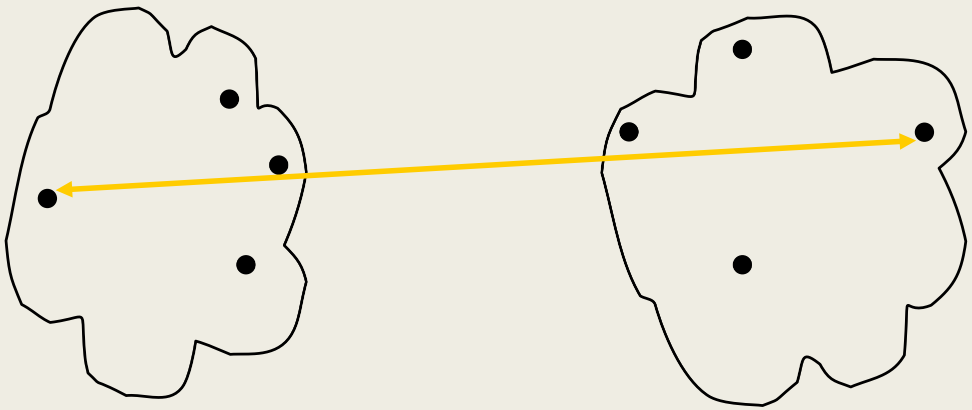


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  – Ward's Method uses squared error

How to Define Inter-Cluster Similarity

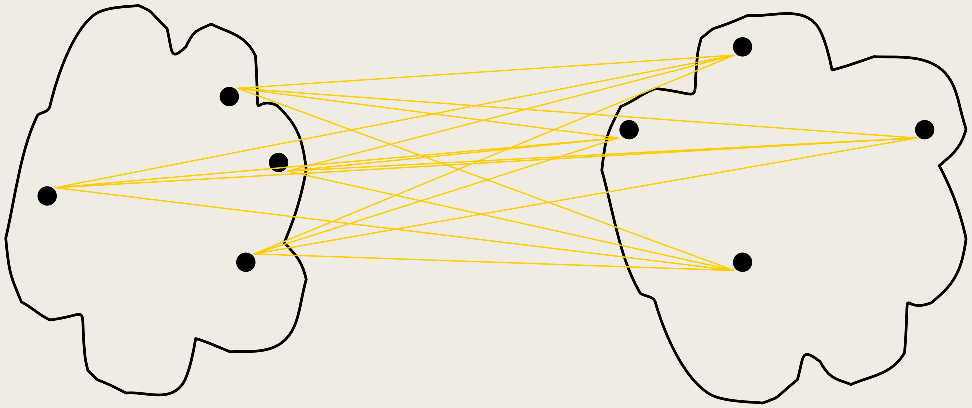


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  – Ward’s Method uses squared error

How to Define Inter-Cluster Similarity

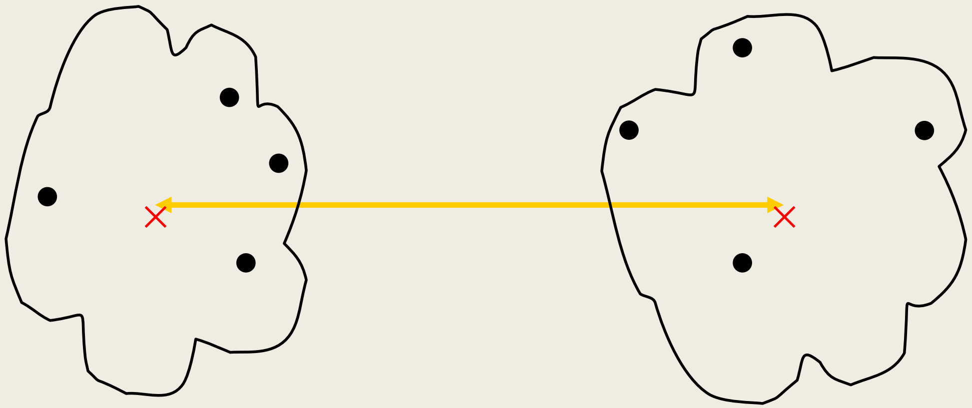


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  – Ward's Method uses squared error

How to Define Inter-Cluster Similarity



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

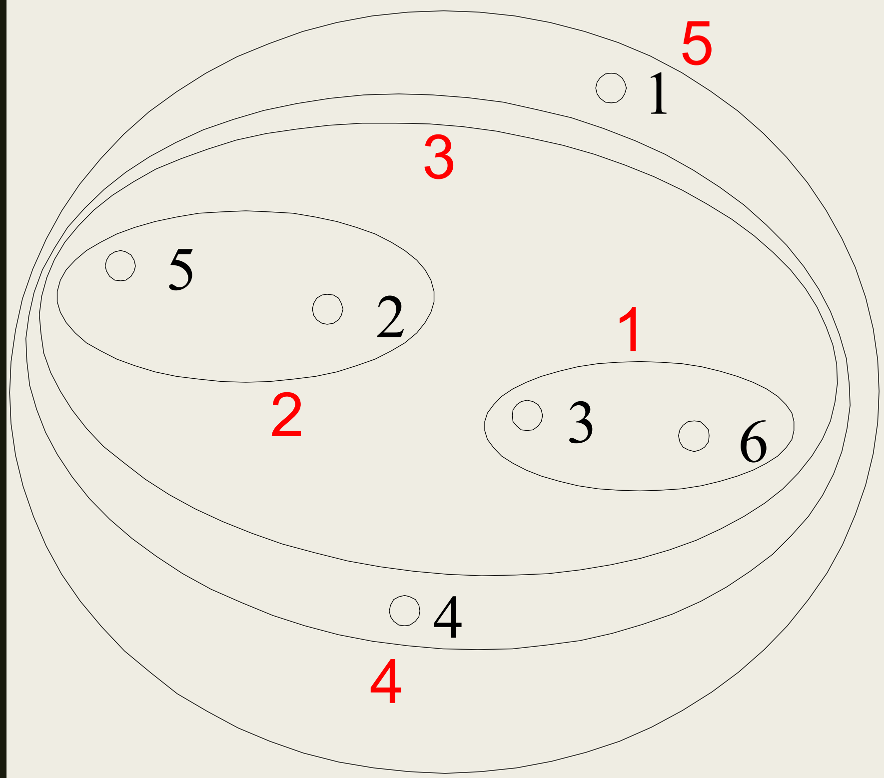
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

Single Link – Complete Link

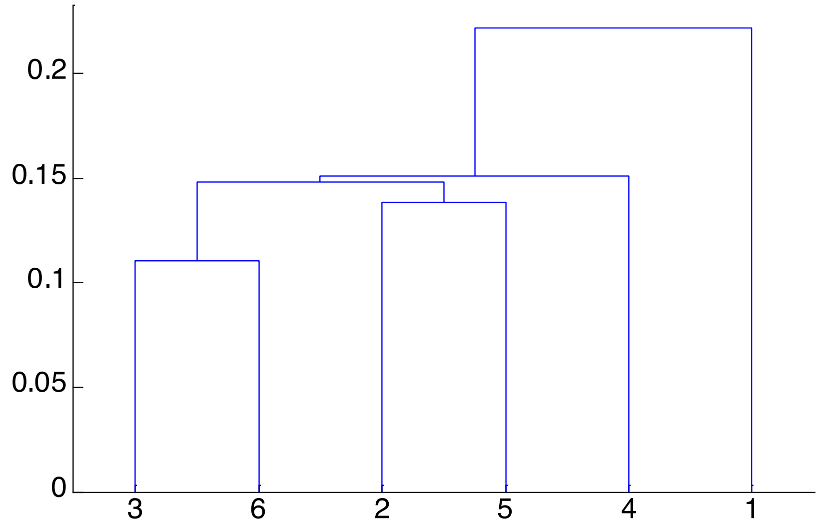
- Another way to view the processing of the hierarchical algorithm is that we create links between their elements in order of increasing distance
  - *The MIN – Single Link, will merge two clusters when a single pair of elements is linked*
  - *The MAX – Complete Linkage will merge two clusters when all pairs of elements have been linked.*

Hierarchical Clustering: MIN

	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0



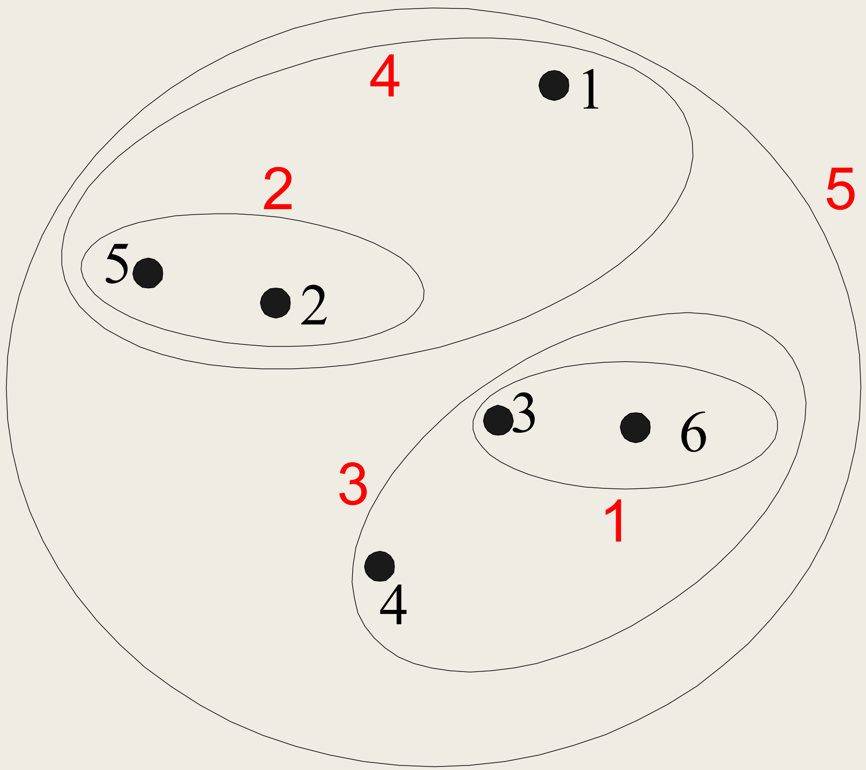
Nested Clusters



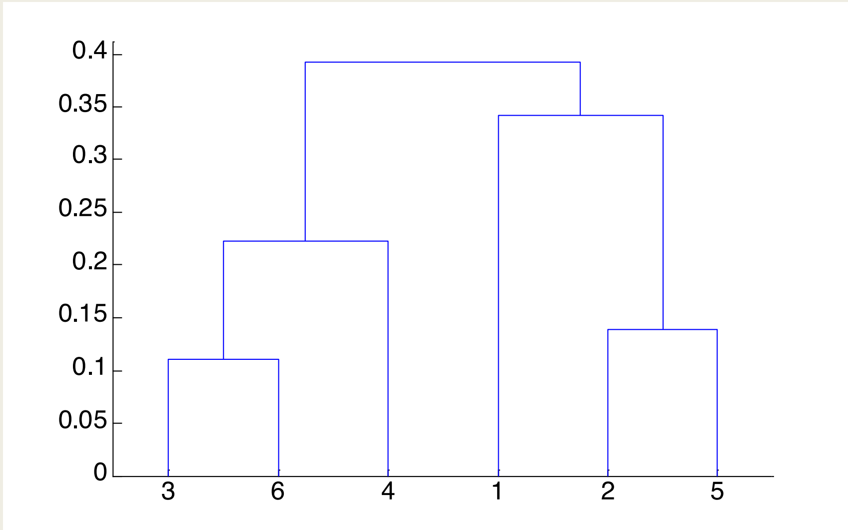
Dendrogram

Hierarchical Clustering: MAX

	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0



Nested Clusters



Dendrogram

Cluster Similarity: Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

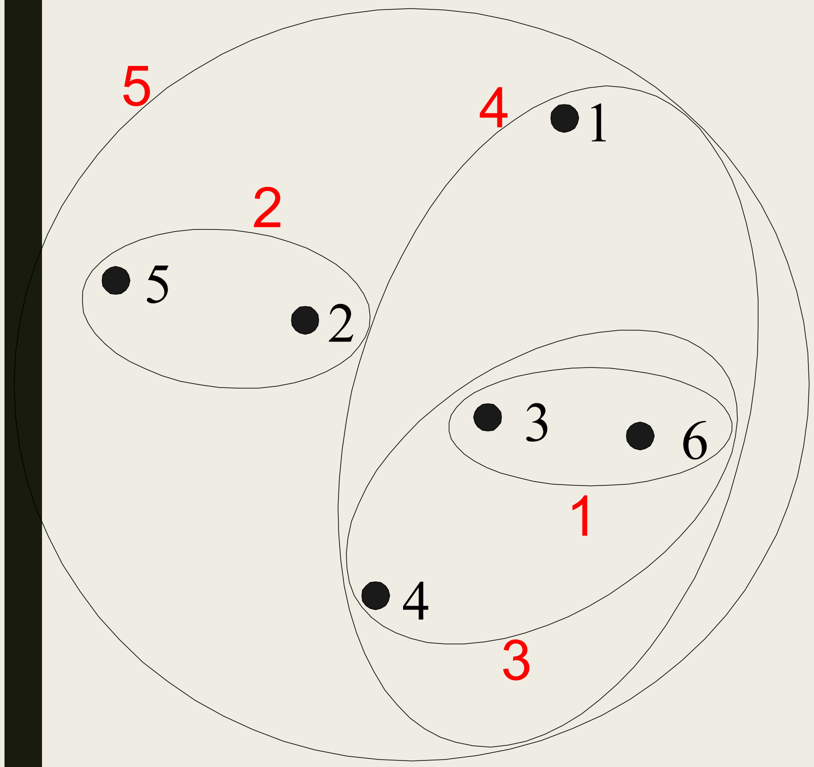
$$\mathbf{proximity}(\mathbf{Cluster}_i, \mathbf{Cluster}_j) = \frac{\displaystyle\sum_{\substack{p_i \in \mathbf{Cluster}_i \\ p_j \in \mathbf{Cluster}_j}} \mathbf{proximity}(p_i, p_j)}{|\mathbf{Cluster}_i| * |\mathbf{Cluster}_j|}$$

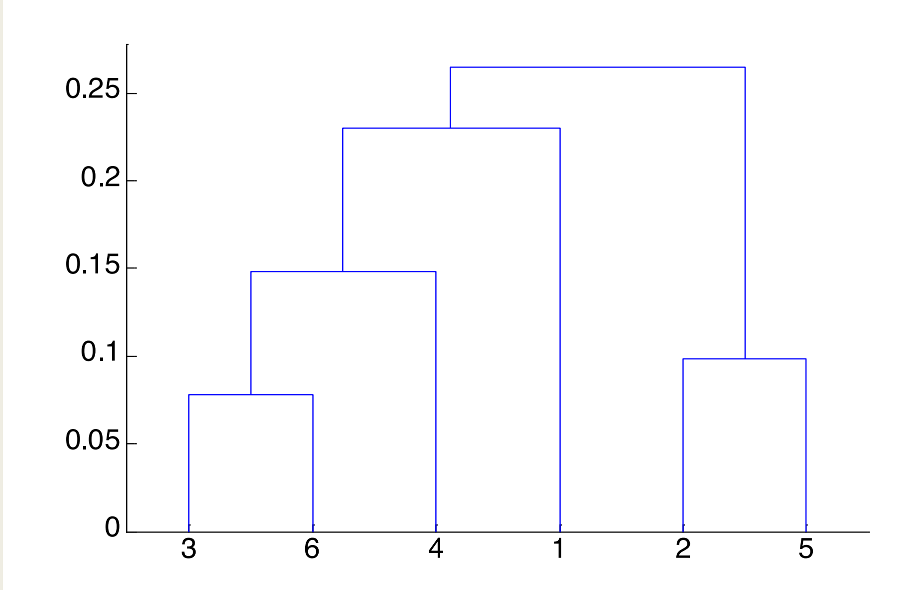
- Need to use average connectivity for scalability since total proximity favors large clusters

	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0

Hierarchical Clustering: Group Average

	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0





Nested Clusters

Dendrogram

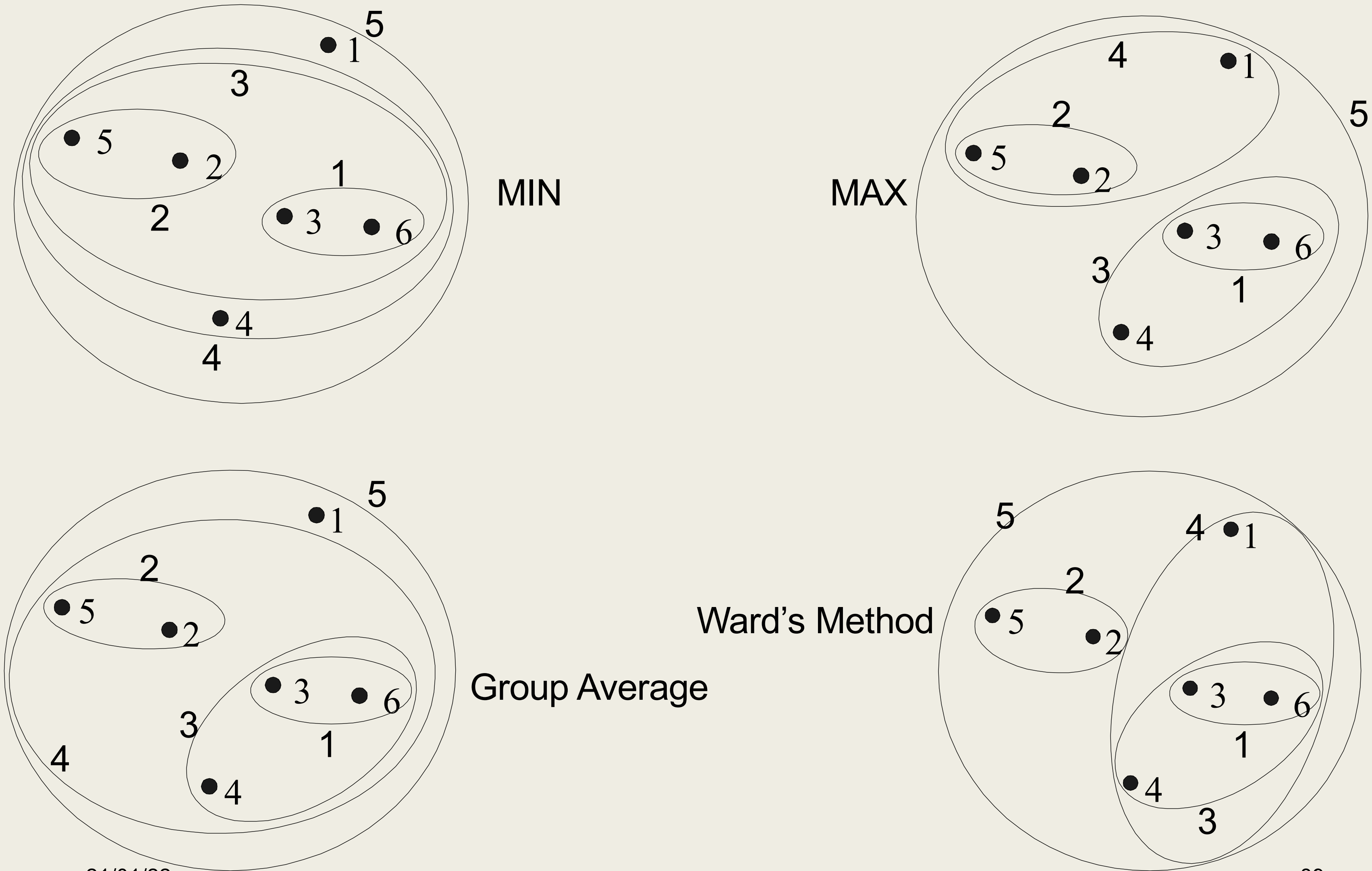
Hierarchical Clustering: Group Average

- Compromise between Single and Complete Link
- Strengths
  - *Less susceptible to noise and outliers*
- Limitations
  - *Biased towards globular clusters*

Cluster Similarity: Ward's Method

- Similarity of two clusters is based on the increase in squared error (SSE) when two clusters are merged
  - *Like group average if distance between points is distance squared*
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of K-means
  - *Can be used to initialize K-means*

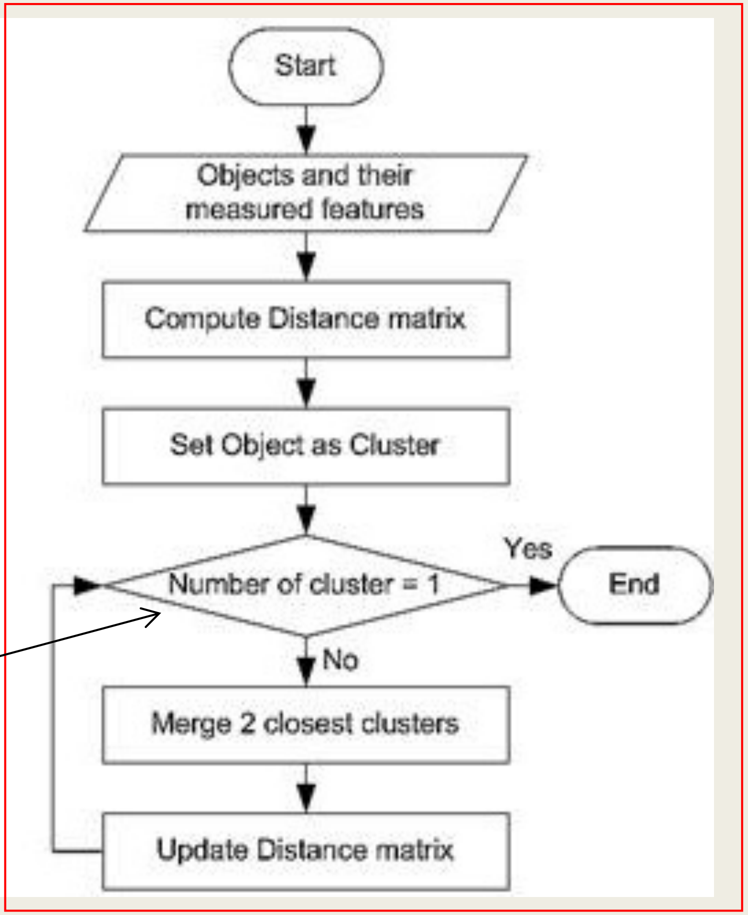
Hierarchical Clustering: Comparison



Agglomerative Algorithm

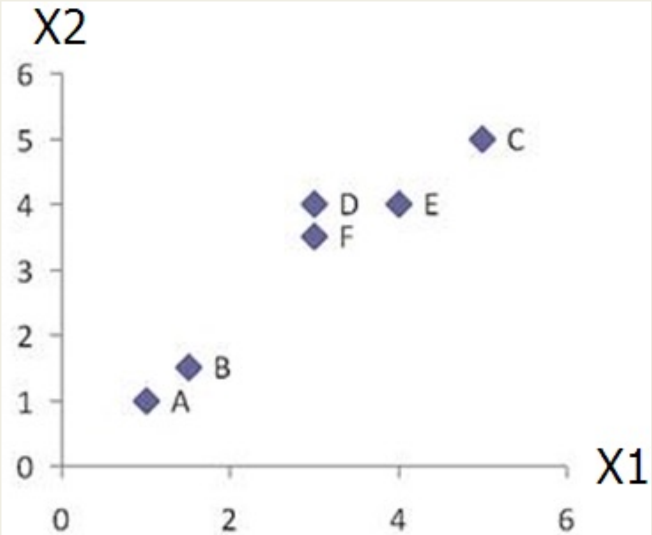
- The *Agglomerative* algorithm is carried out in three steps:

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"



Example

- Problem: clustering analysis with agglomerative algorithm



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

data matrix

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

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

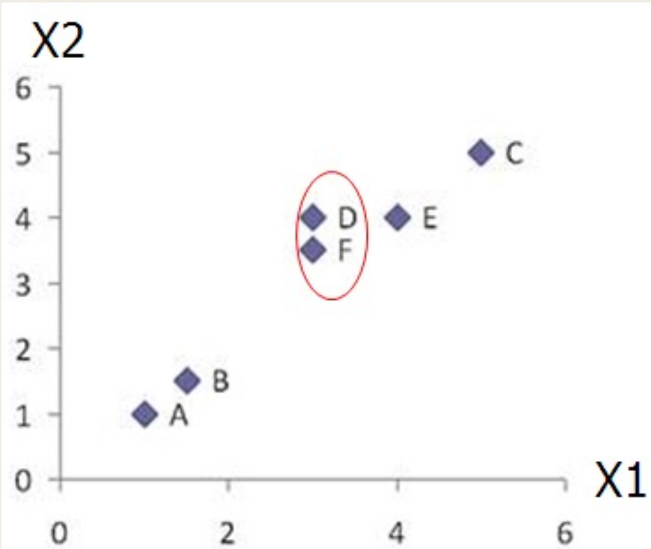
Euclidean distance

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

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

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

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

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

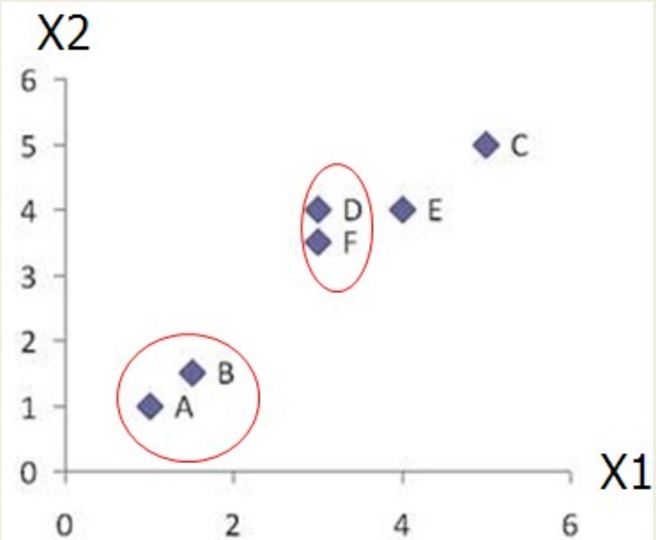
Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

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

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

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

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

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

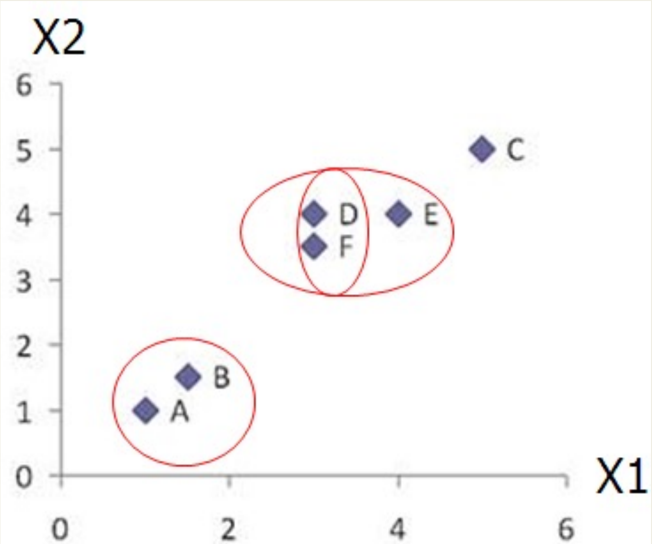
Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	**?**	0	2.24	1.41
(D, F)	**?**	2.24	0	1.00
E	**?**	1.41	1.00	0

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

Example

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



Min Distance (Single Linkage)

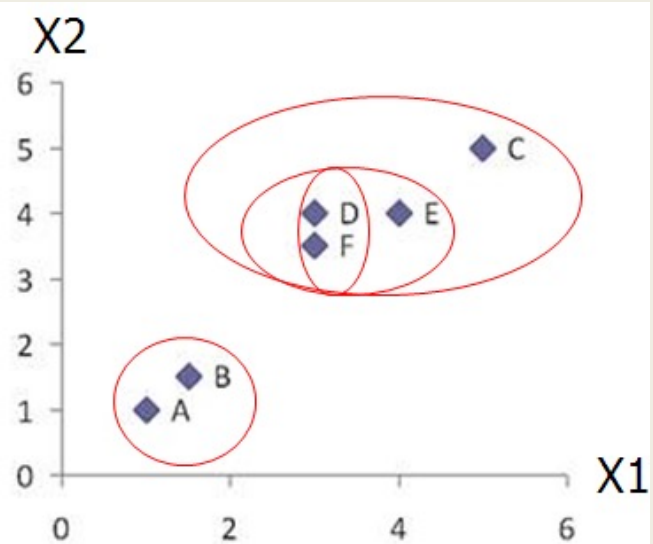
Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	4.95	0	2.24	1.41
(D, F)	2.50	2.24	0	1.00
E	3.54	1.41	1.00	0

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

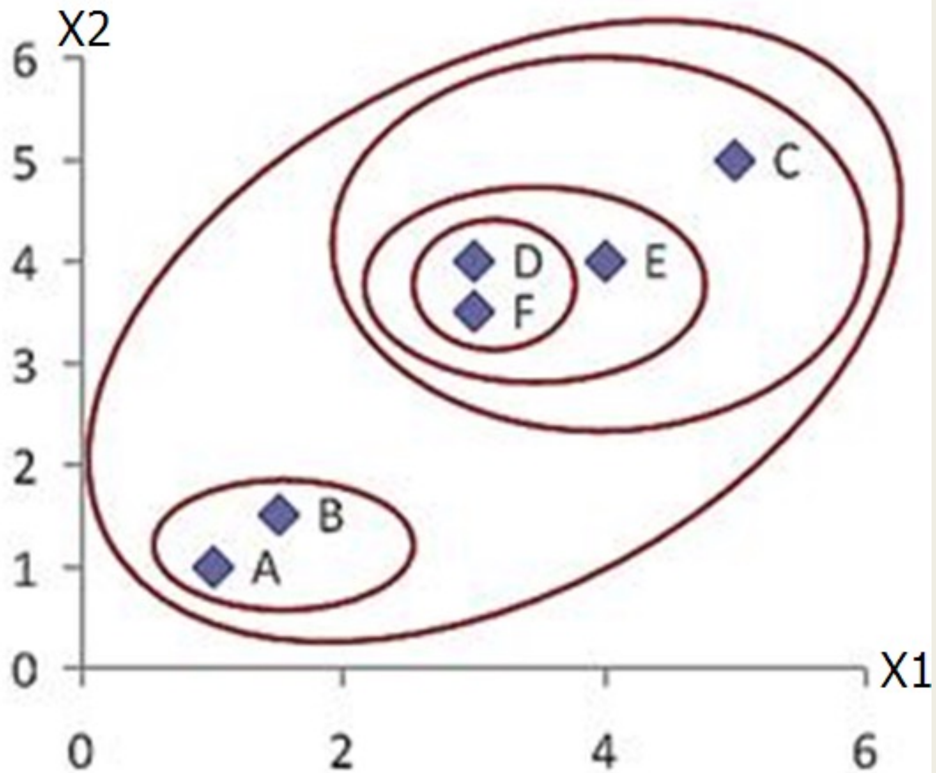
Min Distance (Single Linkage)

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

Example

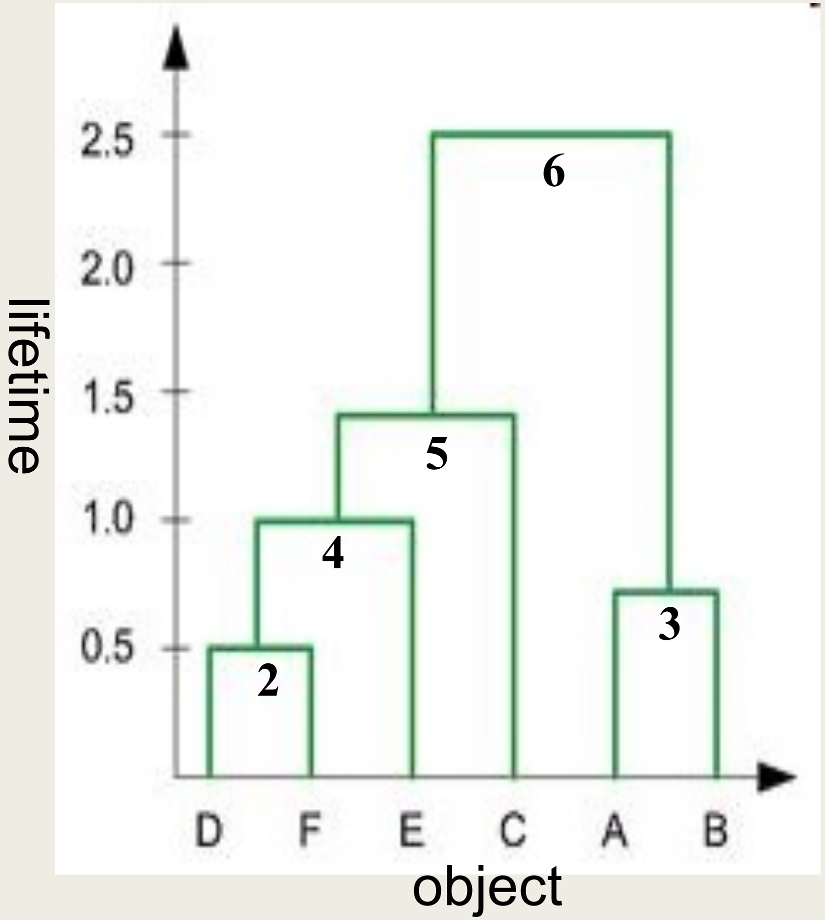
- Result (meeting termination condition)

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



Example

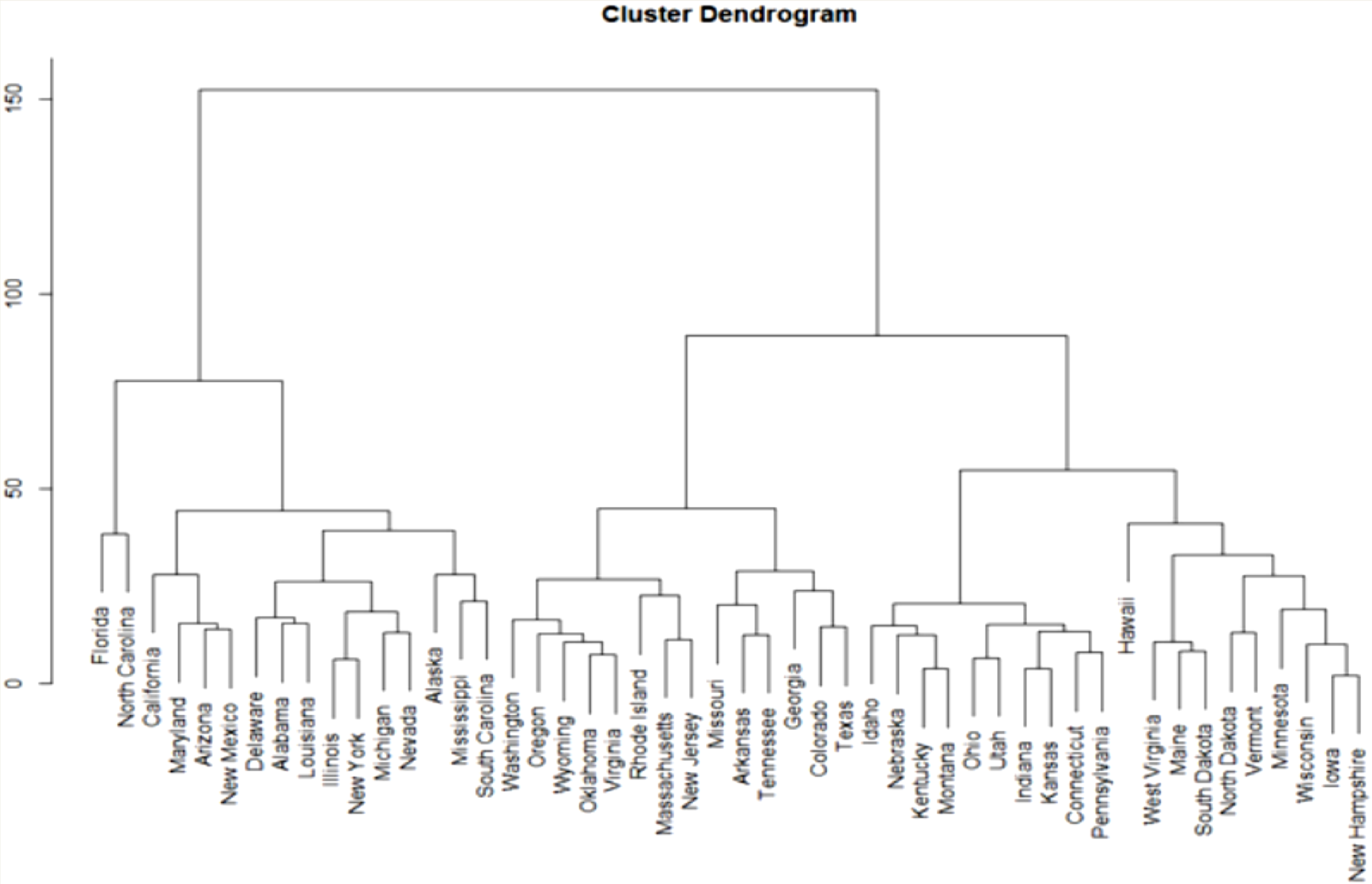
- Dendrogram tree representation



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

Example

- Dendrogram tree representation: "clustering" USA states



Strengths of Hierarchical Clustering

- Do not have to assume any number of clusters
  - *Any desired number of clusters can be obtained by 'cutting' the dendrogram at the proper level*

- They may correspond to meaningful taxonomies
  - *Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)*

Hierarchical Clustering: Problems and Limitations

- Computational complexity in time and space
- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - *Sensitivity to noise and outliers*
  - *Difficulty handling different sized clusters and convex shapes*
  - *Breaking large clusters*

Exercise

Given a data set of five objects characterized by a single continuous feature:

	a	b	C	d	e
Feature	1	2	4	5	6

Apply the agglomerative algorithm with single-link, complete-link and averaging cluster distance measures to produce three dendrogram trees, respectively.

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

Summary

- Hierarchical algorithm is a sequential clustering algorithm
  - *Use distance matrix to construct a tree of clusters (dendrogram)*
  - *Hierarchical representation without the need of knowing # of clusters (can set termination condition with known # of clusters)*
- 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*