Cluster Analysis – Hierarchical Algorithms

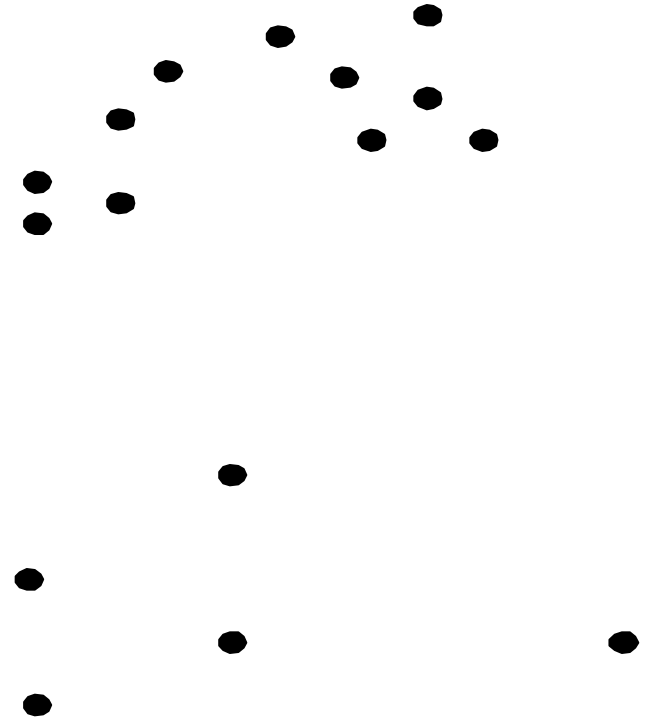
Proximity measures

- Proximity is a generic term that refers to either similarity or dissimilarity.
- Similarity
  - Numerical measure of how *alike* two data objects are.
  - Measure is *higher* when objects are *more alike*.
  - Often falls in the range [0, 1].
- Dissimilarity
  - Numerical measure of how *different* two data objects are.
  - Measure is *lower* when objects are *more alike*.
  - Minimum dissimilarity often 0, upper limit varies.
  - Distance sometimes used as a synonym, usually for specific classes of dissimilarities.

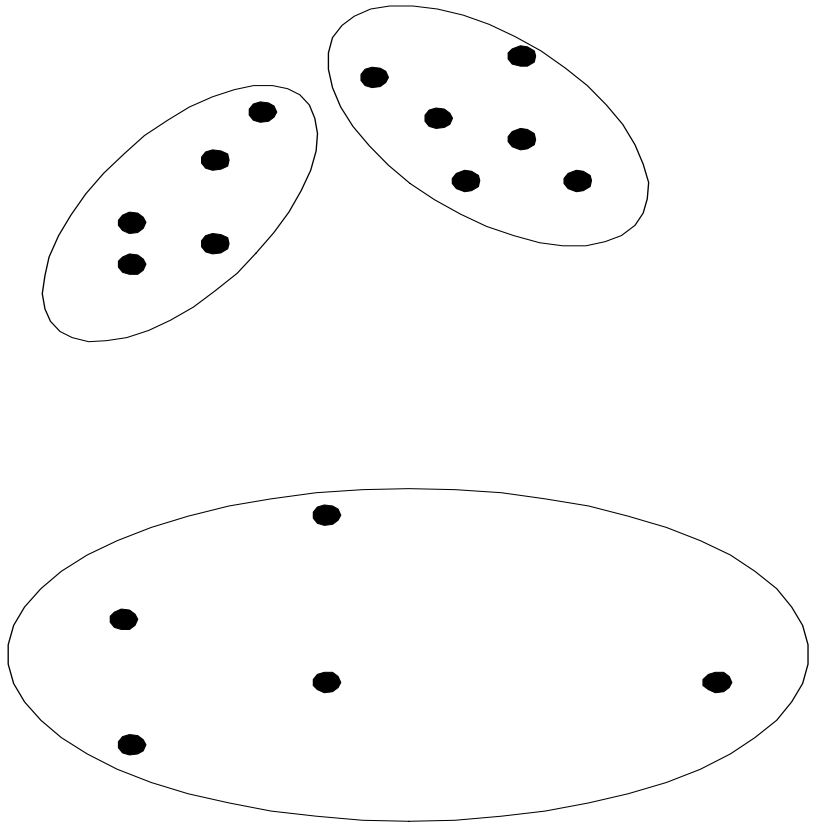
Hierarchical vs Partitional clustering

- Partitional: data points divided into finite number of *partitions* (non-overlapping subsets)
  - each data point is assigned to exactly one subset
- Hierarchical: data points placed into a set of nested clusters, organized into a *hierarchical tree*
  - tree expresses a continuum of similarities and clustering

Partitional clustering



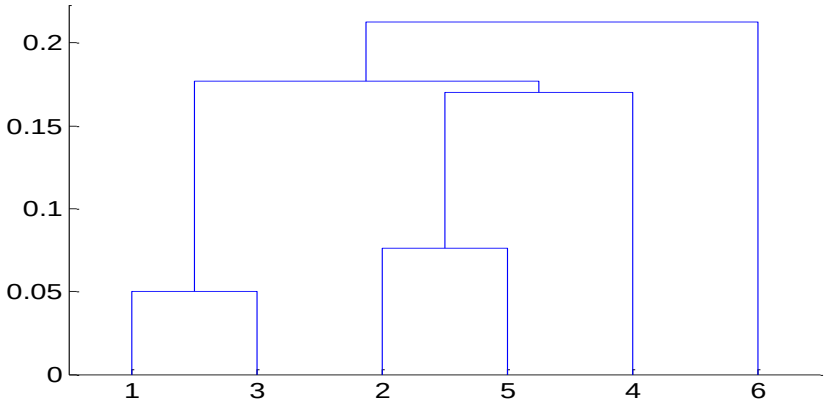
Original points

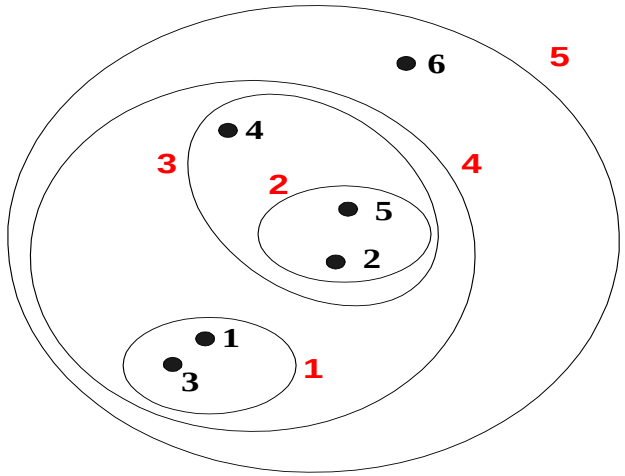


Partitional clustering

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 sequence of merges or splits





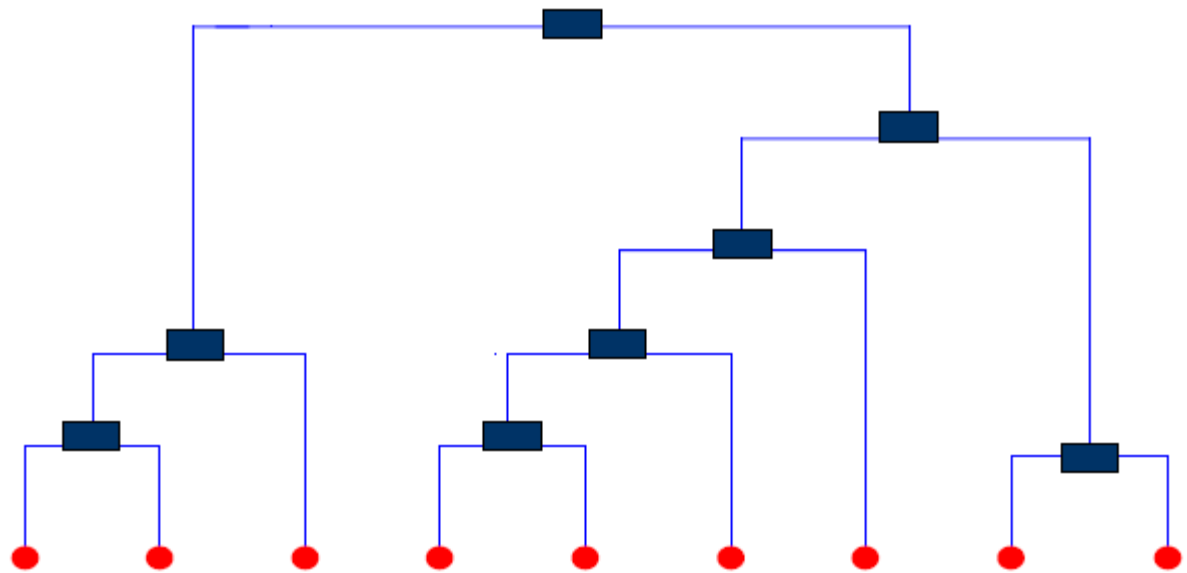
Strengths of hierarchical clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level

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

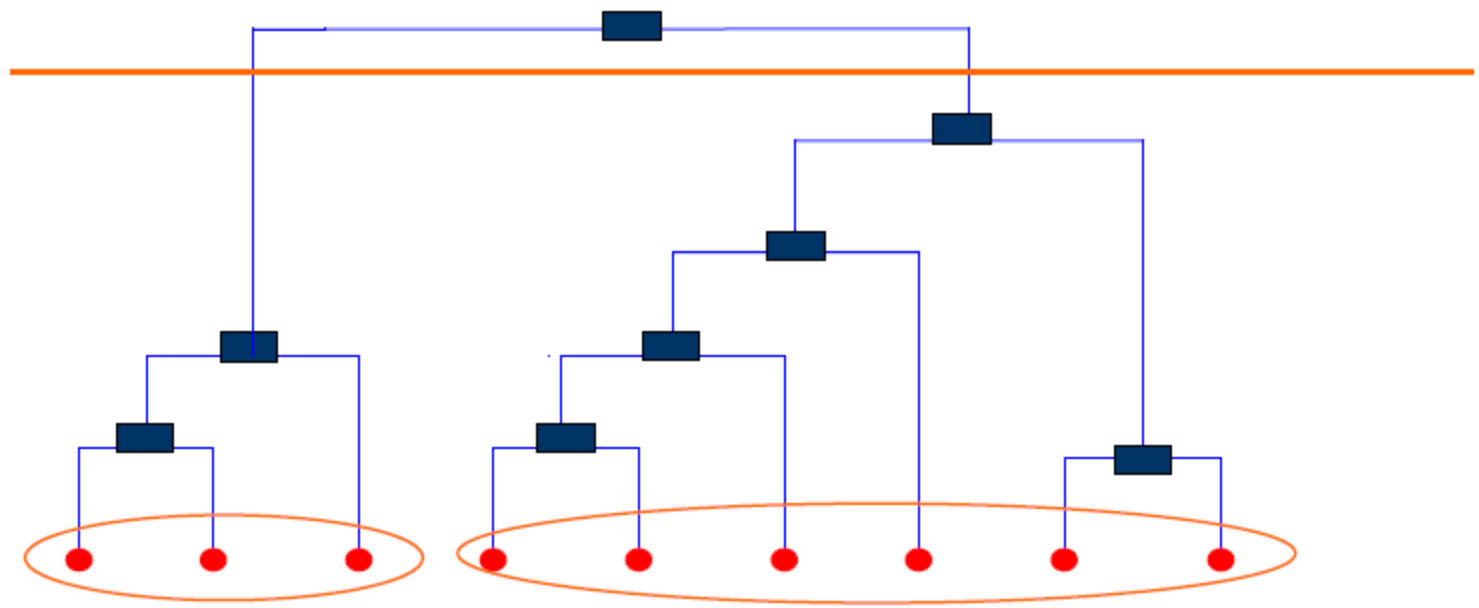
Dendrogram

- A tree that shows how clusters are merged/split hierarchically
- Each node on the tree is a cluster; each leaf node is a singleton cluster



Dendrogram

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



Hierarchical clustering

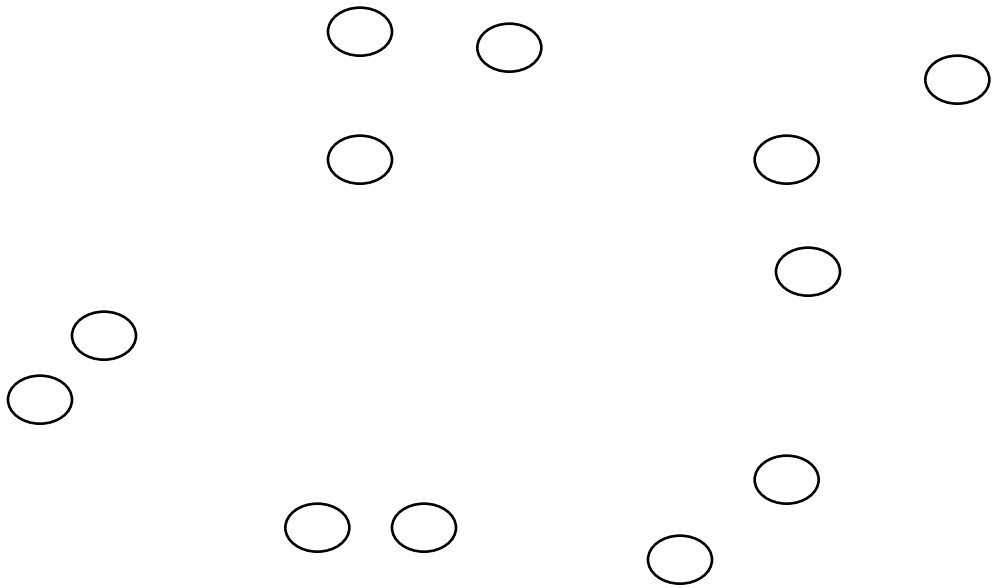
- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a proximity or distance matrix
  - Merge or split one cluster at a time

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 proximities between cluster pairs
  - Different approaches to defining the distance between clusters distinguish the different algorithms

Starting situation

- Start with clusters of individual points and a proximity matrix

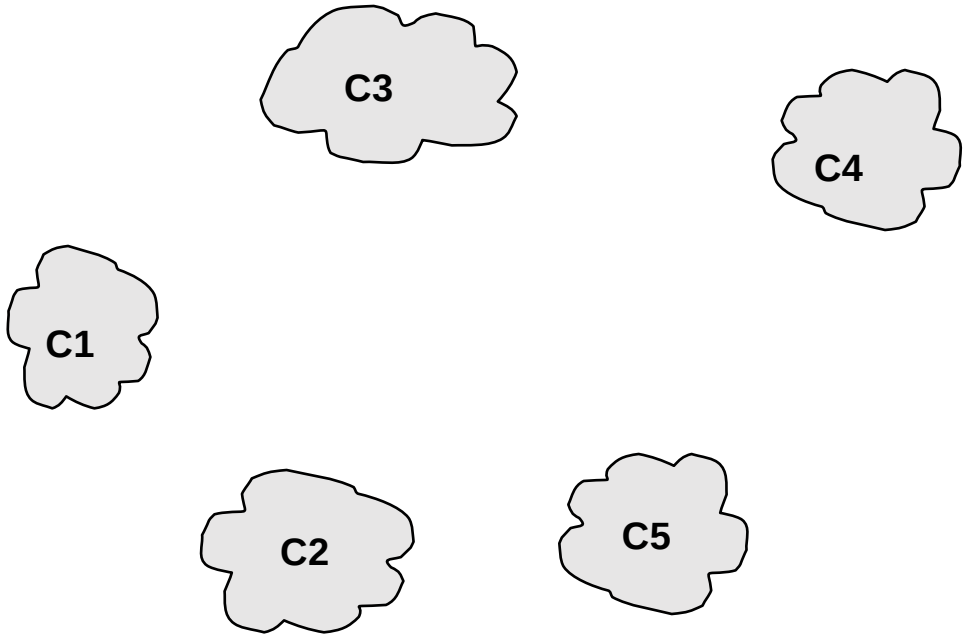


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

proximity matrix

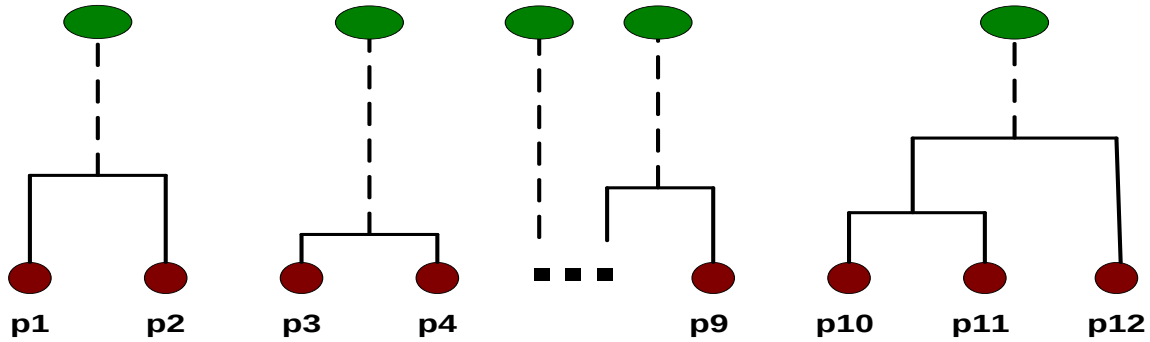
Intermediate situation

- After some merging steps, we have some clusters.



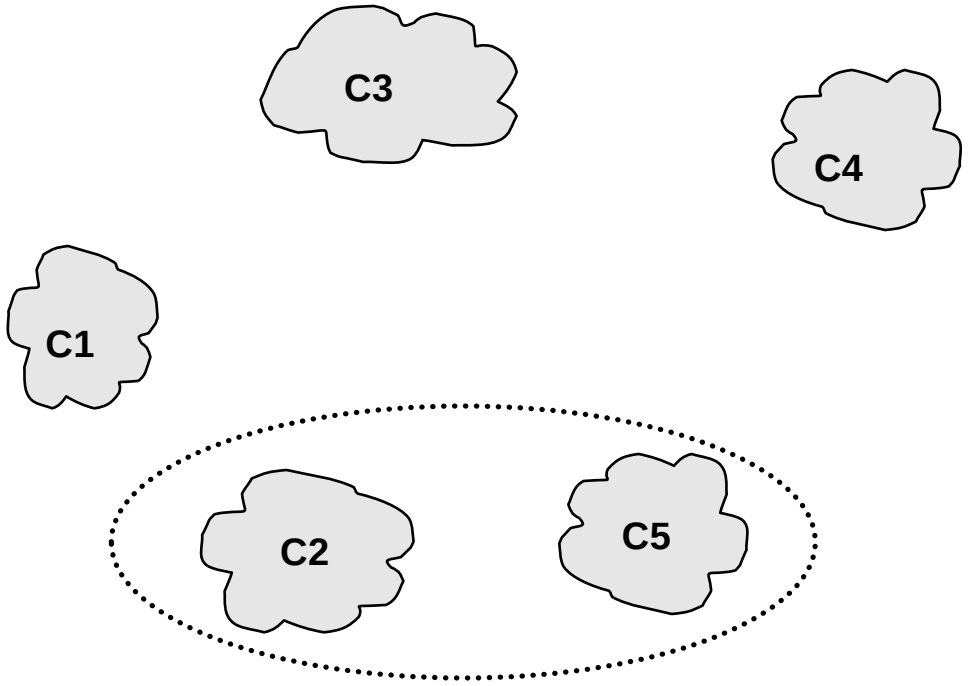
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

proximity matrix



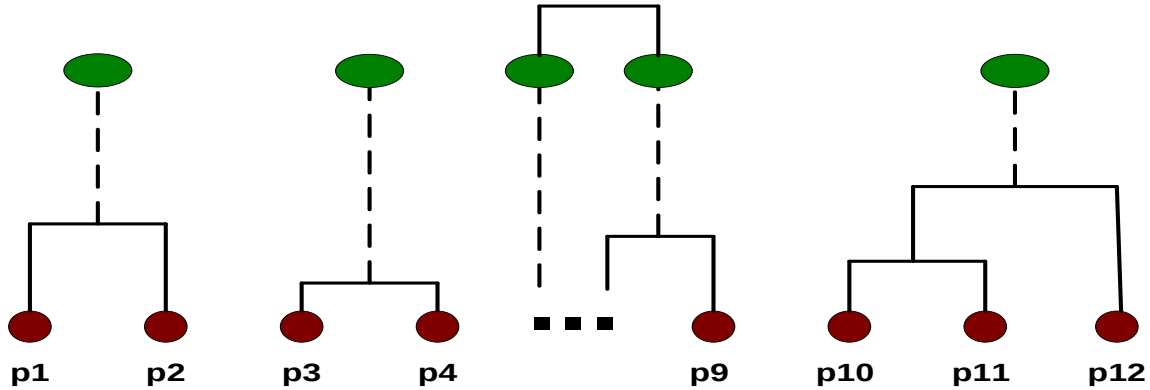
Intermediate situation

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



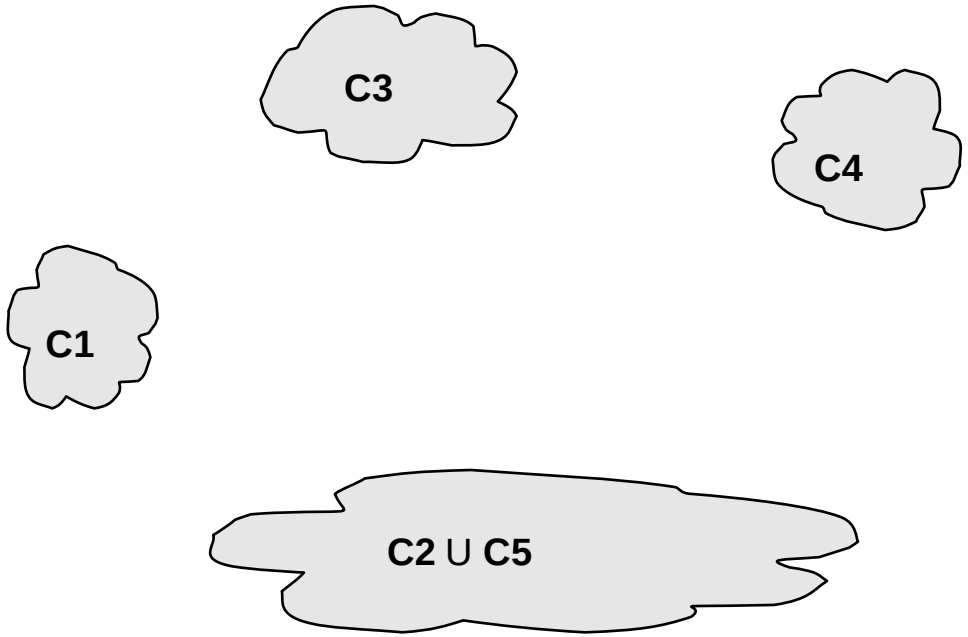
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

proximity matrix



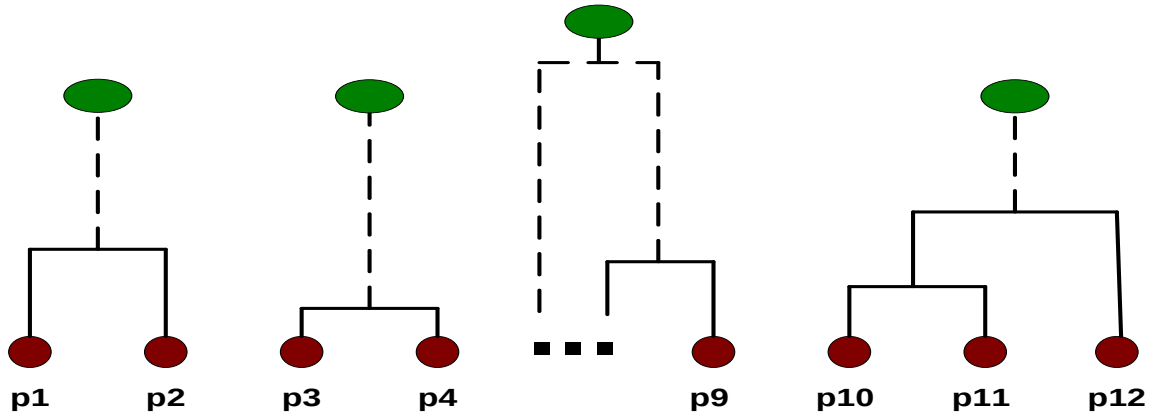
After merging

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

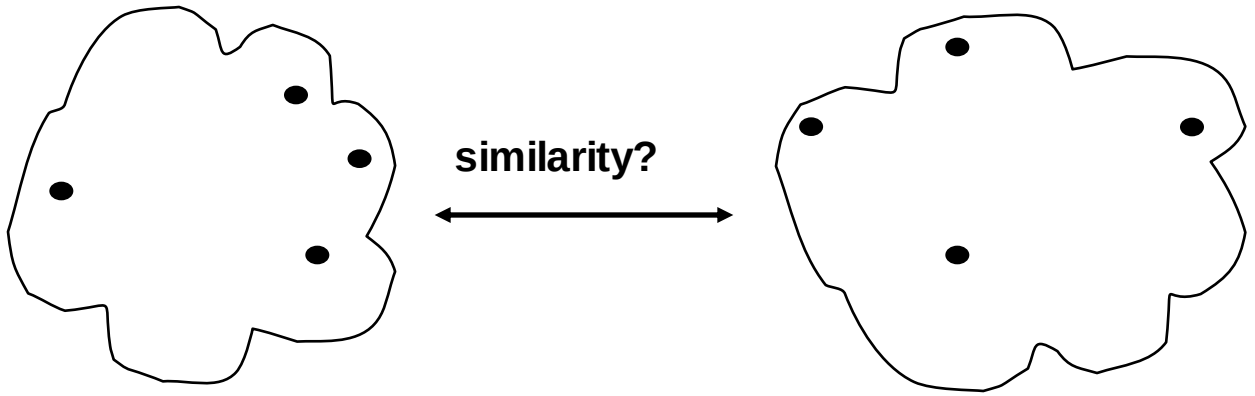


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	? ? ?		
C3		?		
C4		?		

proximity matrix



Defining inter-cluster similarity



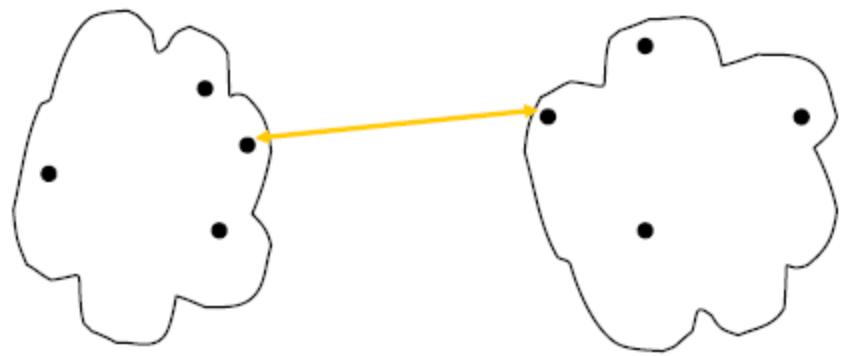
- MIN
- MAX
- Group average
- Distance between centroids
- Other methods driven by an objective function
  - Ward's method uses squared error

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

proximity matrix

MIN or Single Link

- **Inter-cluster distance**
  - The distance between two clusters is represented by the distance of the *closest pair of data objects* belonging to different clusters.
  - Determined by one pair of points, i.e., by one link in the proximity graph



$$d_{\min}(C_i, C_j) = \min_{p \in C_i, q \in C_j} d(p, q)$$

Numerical Example

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

Min Distance (Single Linkage)

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

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

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

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

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

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

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

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

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

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

$$d_{((D,F)\backslash E)\to(A,B)} = \min\left(d_{DA}, d_{DB}, d_{FA}, d_{FB}, d_{EA}, d_{EB}\right) = \min\left(3.61, 2.92, 3.20, 2.50, 4.24, 3.54\right) = 2.50$$

$$d_{((D,F)\backslash E)\to C} = \min\left(d_{DC}, d_{FC}, d_{EC}\right) = \min\left(2.24, 2.50, 1.41\right) = 1.41$$

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

$$d_{(((D,F)\backslash E)\backslash C)\to(A,B)} = \min\left(d_{DA}, d_{DB}, d_{FA}, d_{FB}, d_{EA}, d_{EB}, d_{CA}, d_{CB}\right)$$

$$d_{(((D,F)\backslash E)\backslash C)\to(A,B)} = \min\left(3.61, 2.92, 3.20, 2.50, 4.24, 3.54, 5.66, 4.95\right) = 2.50$$

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

In the beginning we have 6 clusters: A, B, C, D, E and F

We merge cluster D and F into cluster (D, F) at distance 0.50

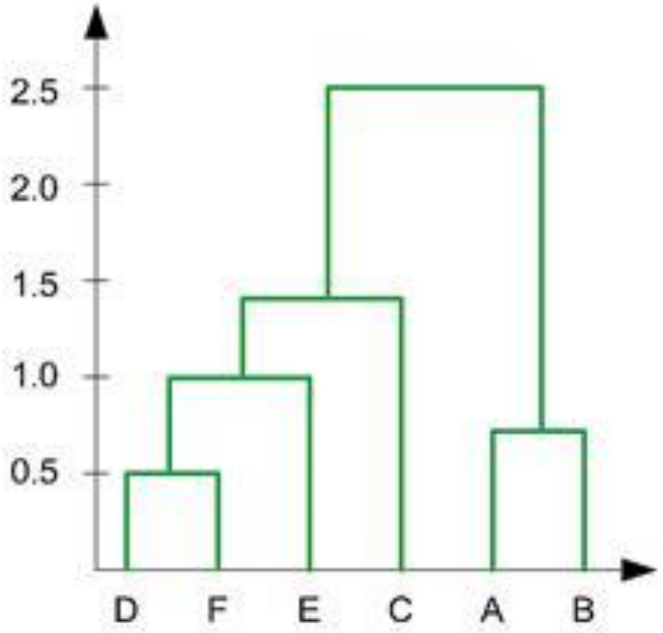
We merge cluster A and cluster B into (A, B) at distance 0.71

We merge cluster E and (D, F) into ((D, F), E) at distance 1.00

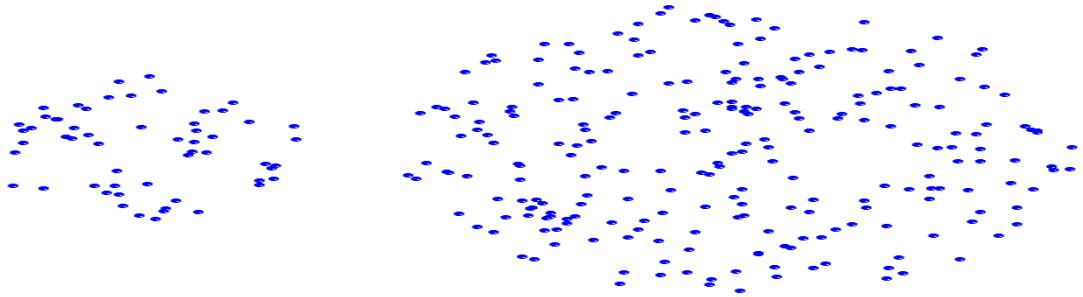
We merge cluster ((D, F), E) and C into (((D, F), E), C) at distance 1.41

We merge cluster (((D, F), E), C) and (A, B) into ((((D, F), E), C), (A, B)) at distance 2.50

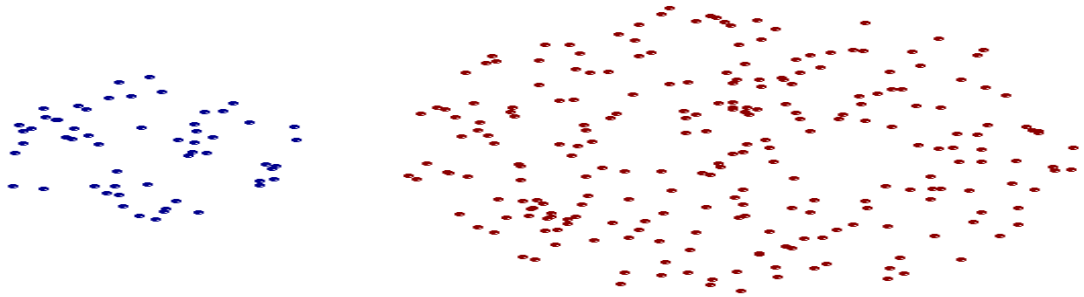
The last cluster contain all the objects, thus conclude the computation



Strength of MIN



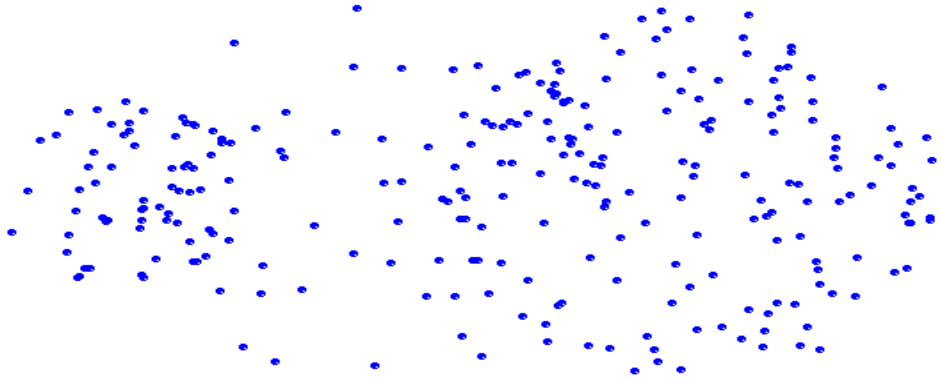
original points



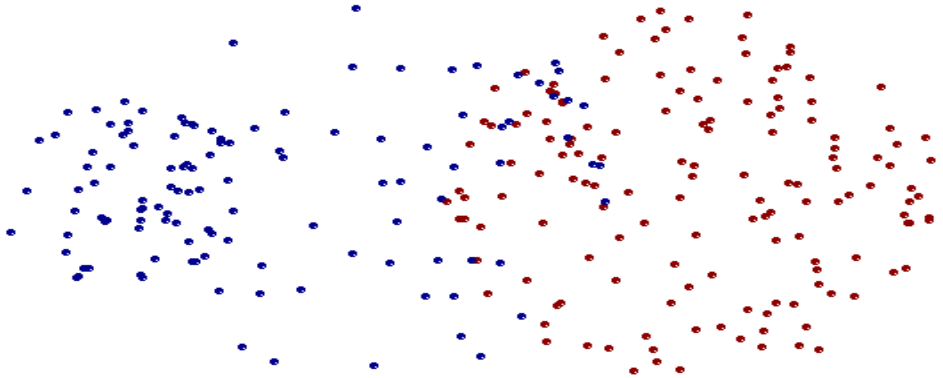
two clusters

- **Can handle non-elliptical shapes**

Limitations of MIN



original points

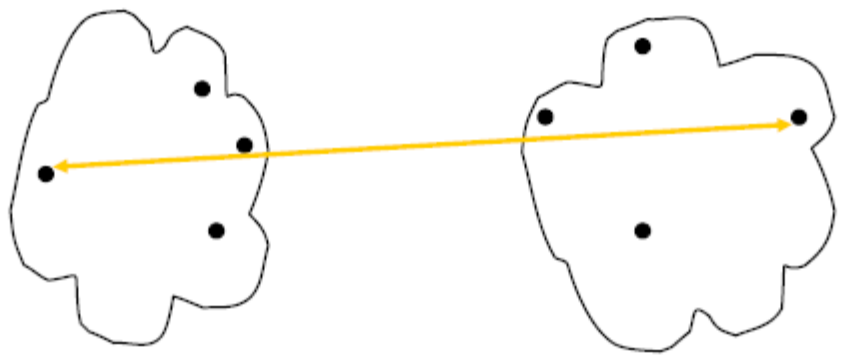


two clusters

- **Sensitive to noise and outliers**

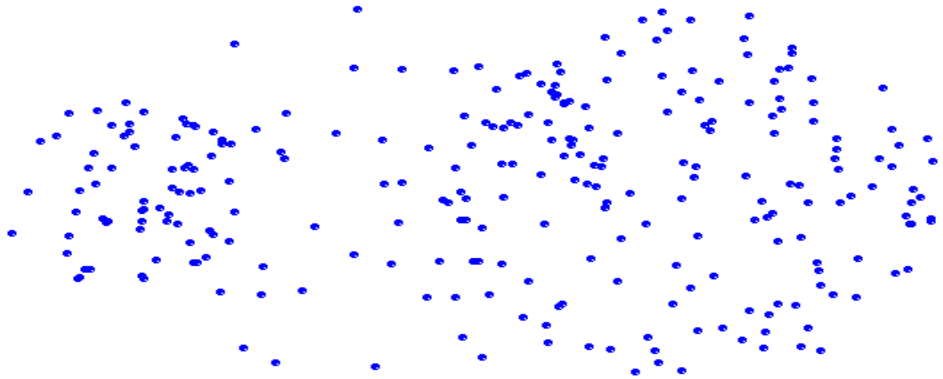
MAX or Complete Link

- **Inter-cluster distance**
  - The distance between two clusters is represented by the distance of the *farthest pair of data objects* belonging to different clusters

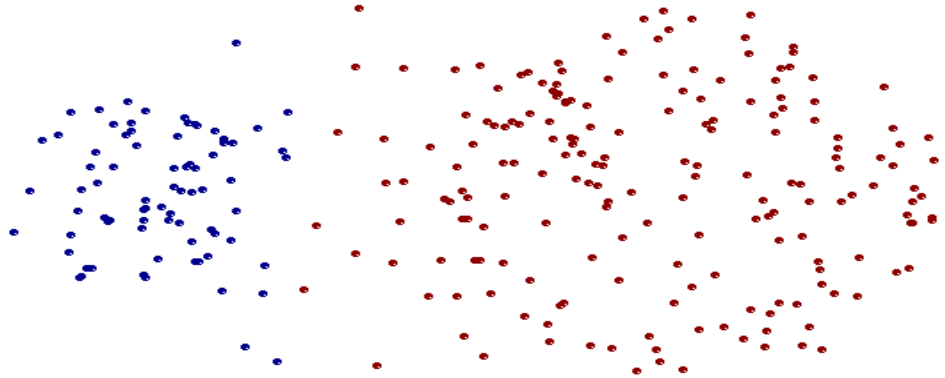


$$d_{\min}(C_i, C_j) = \max_{p \in C_i, q \in C_j} d(p, q)$$

Strength of MAX



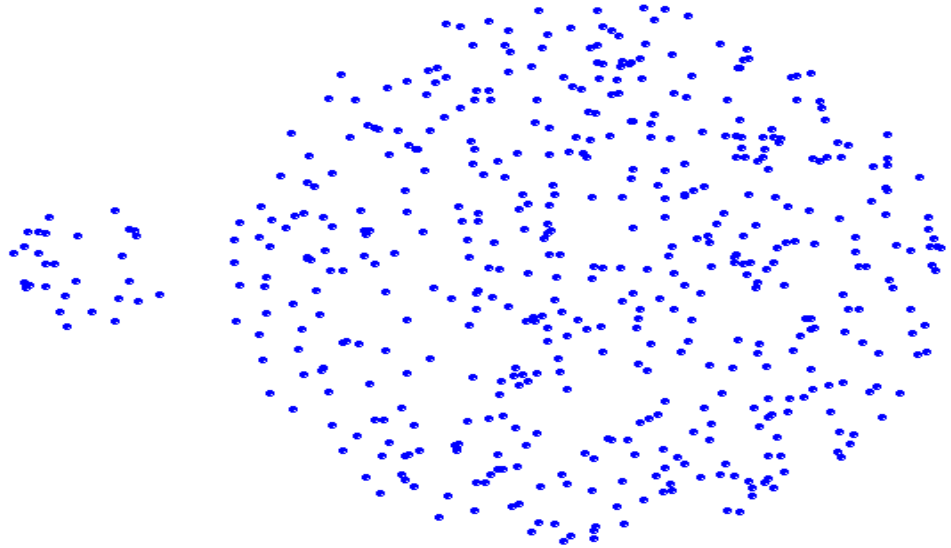
original points



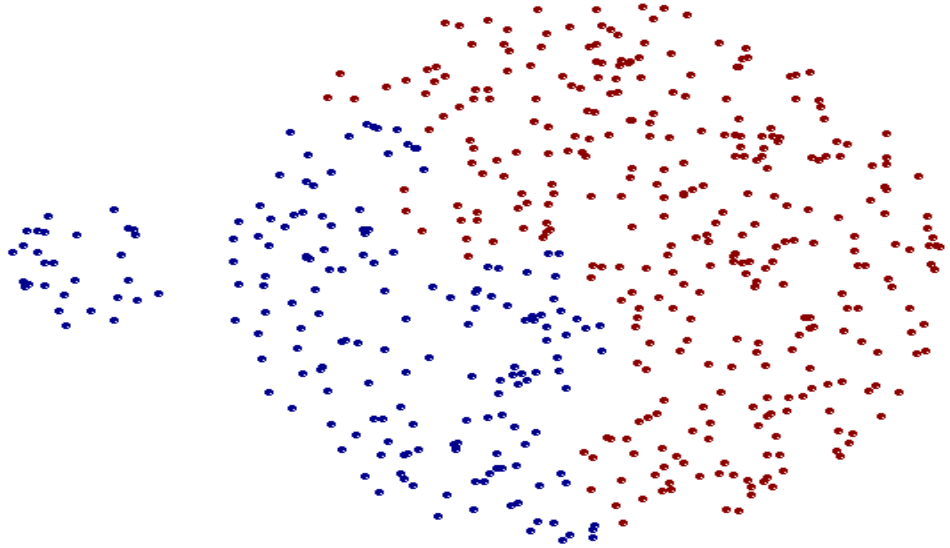
two clusters

- **Less susceptible to noise and outliers**

Limitations of MAX



original points

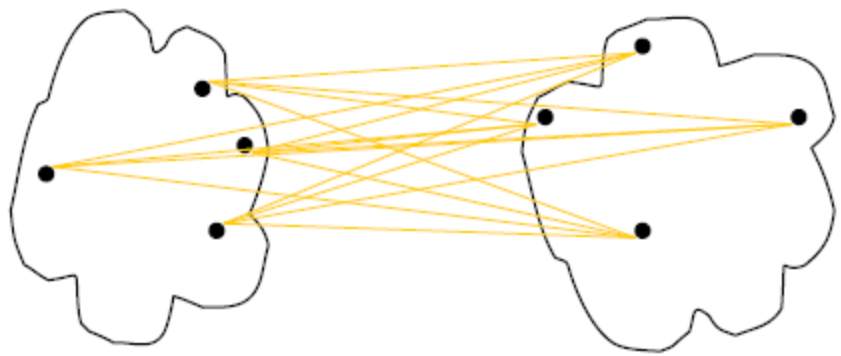


two clusters

- **Tends to break large clusters**
- **Biased towards globular clusters**

Group Average or Average Link

- **Inter-cluster distance**
  - The distance between two clusters is represented by the *average* distance of *all pairs of data objects* belonging to different clusters
  - Determined by all pairs of points in the two clusters



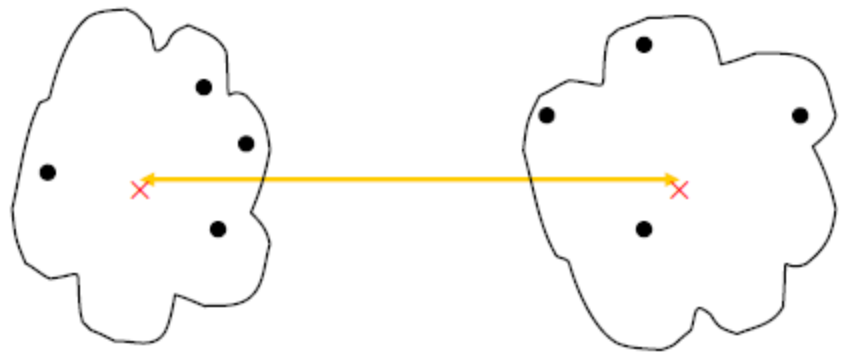
$$d_{\min}(C_i, C_j) = \underset{p \in C_i, q \in C_j}{avg} d(p,q)$$

Hierarchical clustering: group average

- Compromise between single and complete link
- Strengths:
  - Less susceptible to noise and outliers
- Limitations:
  - Biased towards globular clusters

Centroid Distance

- **Inter-cluster distance**
  - The distance between two clusters is represented by the distance between *the centers of the clusters*
  - Determined by cluster centroids

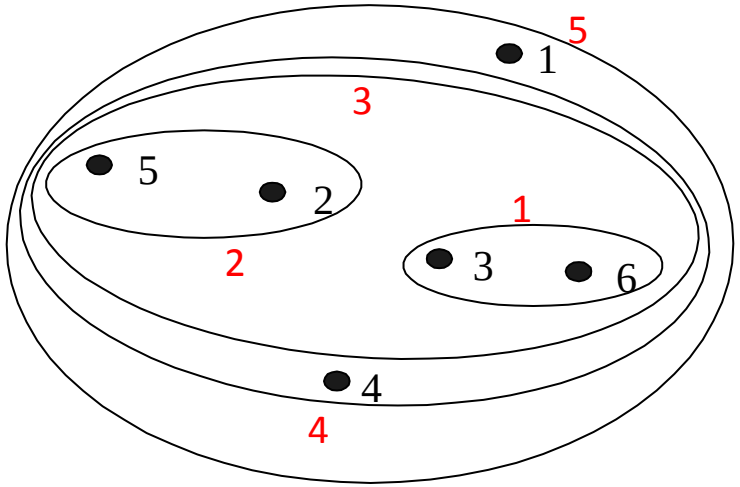


$$d_{mean}(C_i, C_j) = d(m_i, m_j)$$

Cluster similarity: Ward's method

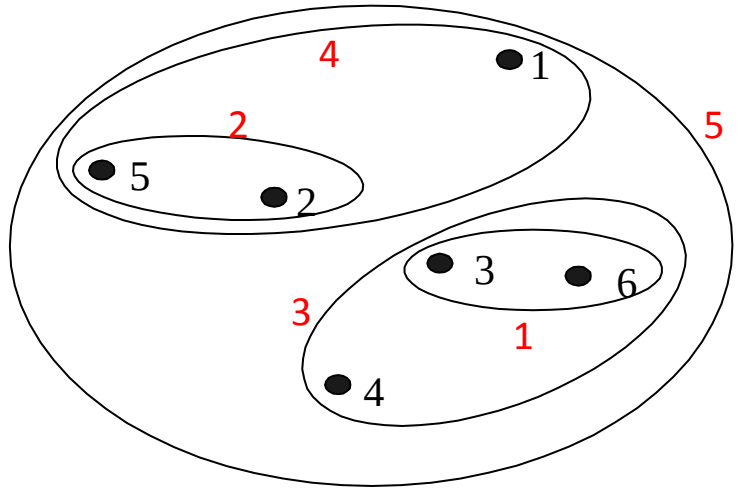
- Similarity of two clusters is based on the increase in squared error when two clusters are merged
  - Similar to group average if distance between points is distance squared
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of k-means
  - Can be used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

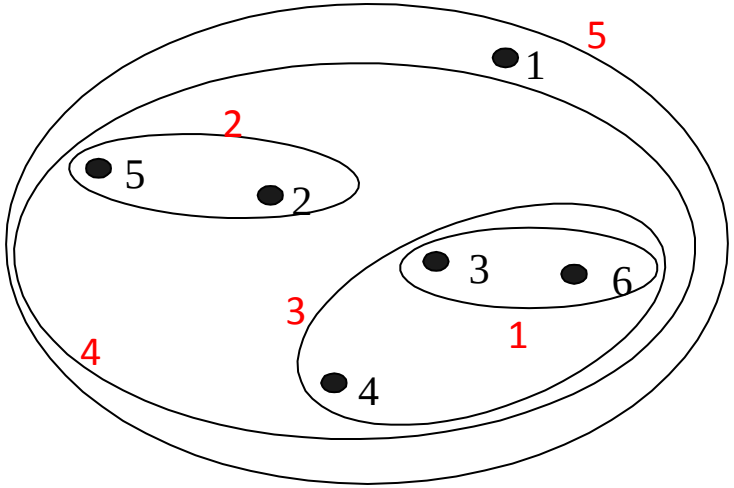
Hierarchical clustering comparison



MIN

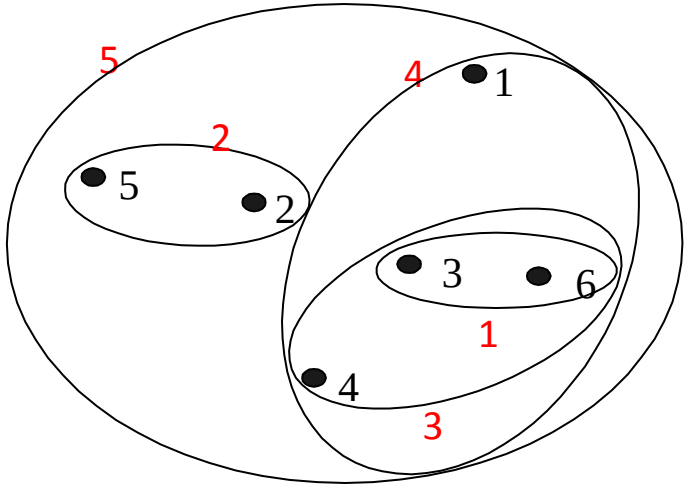
MAX





group average

Ward's method

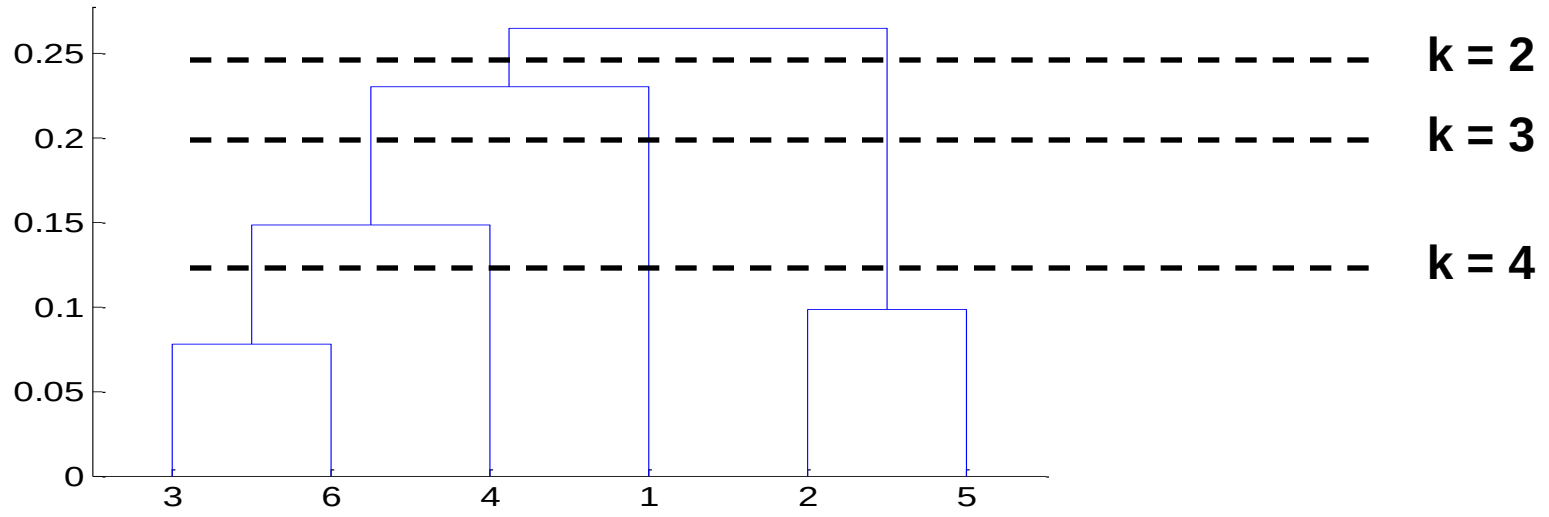


Hierarchical clustering

- Problems and limitations
  - 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

From hierarchical to partitional clustering

- Cut tree at some height to get desired number of partitions k



Hierarchical clustering

- Time and space complexity
  - n = number of datapoints or objects
  - Space requirement ~ O(n^2) since it uses the proximity matrix.
  - Time complexity ~ O(n^3) many cases.
    - There are n steps and at each step the proximity matrix (size n^2) must be searched and updated.
    - Can be reduced to O(n^2 log(n)) time for some approaches.