

# AGNES

## Agglomerative

Dr. Mydhili K Nair  
&

Dr. Pushpalatha MN  
ISE Dept.

M S Ramaiah Institute of Technology

# Agglomerative Hierarchical clustering

- This algorithm works by grouping the data one by one on the basis of the nearest distance measure of all the pairwise distance between the data point.
- The distance between the data point is recalculated but which distance to consider when the groups has been formed? For this there are many available methods. Some of them are:
  1. single-nearest distance or single linkage
  2. complete-farthest distance or complete linkage
  3. average-average distance or average linkage
  4. centroid distance
  5. ward's method - sum of squared Euclidean distance is minimized
- This way we go on grouping the data until one cluster is formed
- Now on the basis of dendrogram graph we can calculate how many number of clusters should be actually present.

# How They Work!

- Given a set of  $N$  items to be clustered, and an  $N \times N$  distance (or similarity) matrix, the basic process of hierarchical clustering (defined by S.C. Johnson in 1967) is this:
  1. Start by assigning each item to a cluster, so that if you have  $N$  items, you now have  $N$  clusters, each containing just one item.
    - Let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain.
  2. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.
  3. Compute distances (similarities) between the new cluster and each of the old clusters.
  4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size  $N$ . (\*)
- Step 3 can be done in different ways, which is what distinguishes single-linkage from complete-linkage and average-linkage clustering.

# Types of Agglomerative method

1. Maximum or ***complete linkage***: The distance between two clusters is defined as the maximum value of all pairwise distances between the elements in cluster 1 and the elements in cluster 2. It tends to produce more compact clusters.
  2. Minimum or ***single linkage***: The distance between two clusters is defined as the minimum value of all pairwise distances between the elements in cluster 1 and the elements in cluster 2. It tends to produce long, “loose” clusters.
  3. Mean or ***average linkage***: The distance between two clusters is defined as the average distance between the elements in cluster 1 and the elements in cluster 2.
  4. ***Centroid linkage***: The distance between two clusters is defined as the distance between the centroid for cluster 1 (a mean vector of length  $p$  variables) and the centroid for cluster 2.
  5. ***Ward's minimum variance method***: It minimizes the total within-cluster variance. At each step the pair of clusters with minimum between-cluster distance are merged.
- This kind of hierarchical clustering is called *agglomerative* because it merges clusters iteratively
  - once you have got the complete hierarchical tree, if you want  $k$  clusters you just have to cut the  $k-1$  longest links.

# Algorithmic steps for Agglomerative Hierarchical clustering

Let  $X = \{x_1, x_2, x_3, \dots, x_n\}$  be the set of data points

- 1) Begin with the disjoint clustering having level  $L(0) = 0$  and sequence number  $m = 0$ .
- 2) Find the least distance pair of clusters in the current clustering, say pair  $(r), (s)$ , according to  $d[(r),(s)] = \min d[(i),(j)]$  where the minimum is over all pairs of clusters in the current clustering.
- 3) Increment the sequence number:  $m = m + 1$
- Merge clusters  $(r)$  and  $(s)$  into a single cluster to form the next clustering  $m$ . Set the level of this clustering to
  - $L(m) = d[(r),(s)]$
- 4) Update the distance matrix,  $D$ , by deleting the rows and columns corresponding to clusters  $(r)$  and  $(s)$  and adding a row and column corresponding to the newly formed cluster.

The distance between the new cluster, denoted  $(r,s)$  and old cluster  $(k)$  is defined in this way:

$$d[(k), (r,s)] = \min (d[(k),(r)], d[(k),(s)]).$$

- 5) If all the data points are in one cluster then stop, else repeat from step 2).

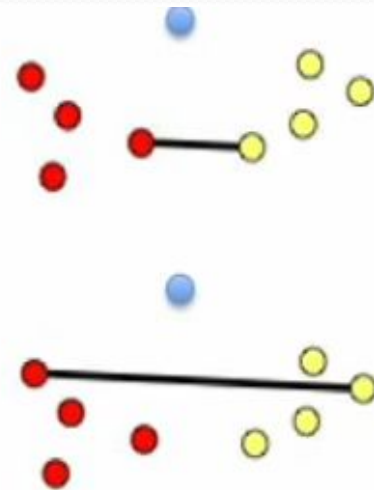
# Hierarchical Agglomerative Clustering Representation

- **S**ingle-nearest distance or single linkage.
- **C**omplete-farthest distance or complete linkage.
- **A**verage-average distance or average linkage.



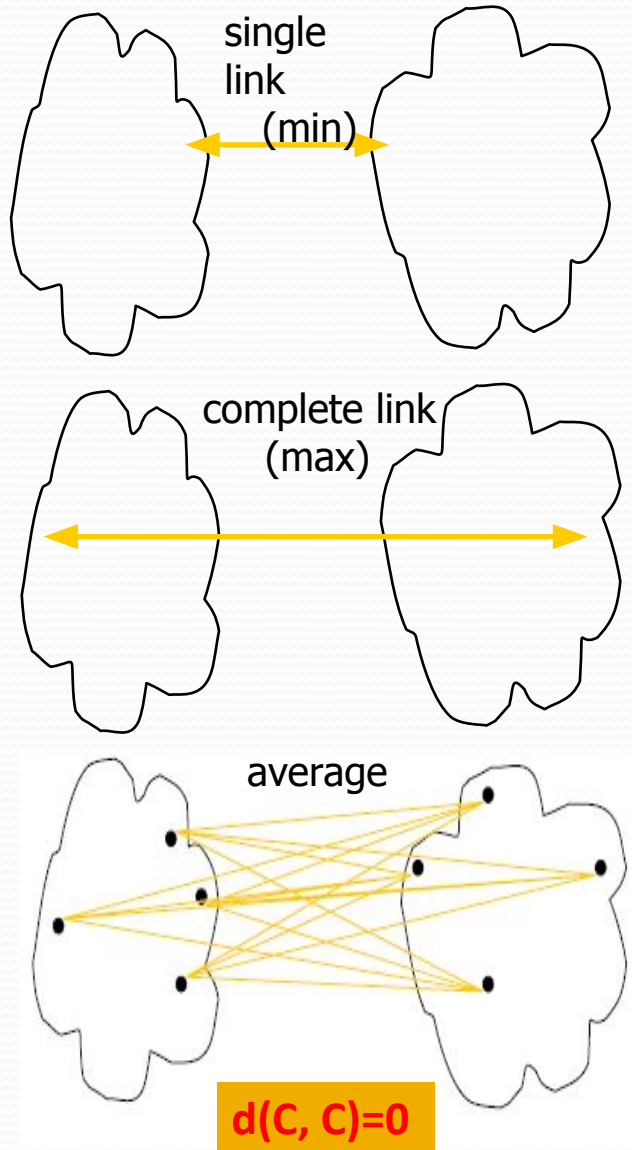
# Single link vs. complete link

- Single link:  $D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$ 
  - distance between closest elements in clusters
  - produces long chains  $a \rightarrow b \rightarrow c \rightarrow \dots \rightarrow z$
- Complete link:  $D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$ 
  - distance between farthest elements in clusters
  - forces “spherical” clusters with consistent “diameter”



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





# Hierarchical Agglomerative Clustering Representation

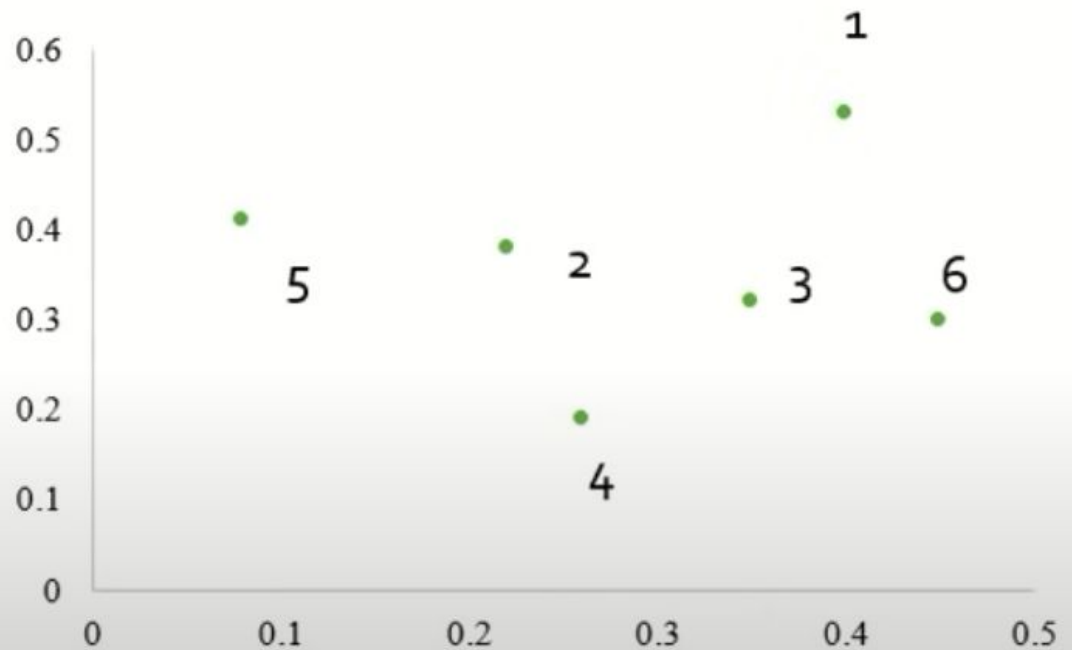
Single Linkage	This is the distance between the closest members of the two clusters.
Complete Linkage	This is the distance between the members that are farthest apart.
Average Linkage	This method involves looking at the distances between all pairs and averages all of these distances. This is also called Unweighted Pair Group Mean Averaging.

- Single-nearest distance or single linkage.

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

## ■ Single-nearest distance or single linkage.

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30



- Calculate Euclidean distance, create the distance matrix.

$$\text{Distance } [(x,y), (a,b)] = \sqrt{(x - a)^2 + (y - b)^2}$$

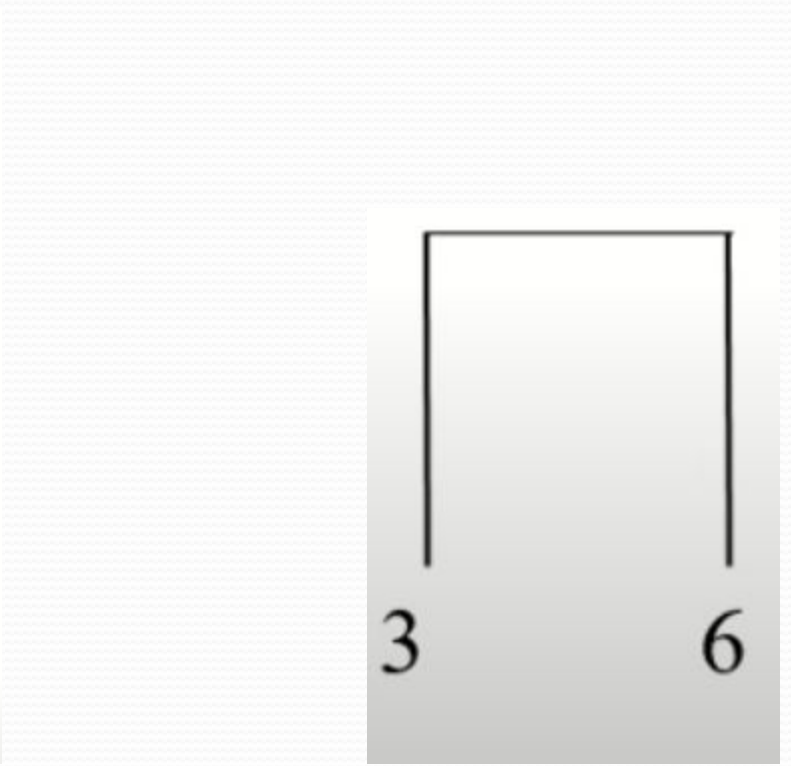
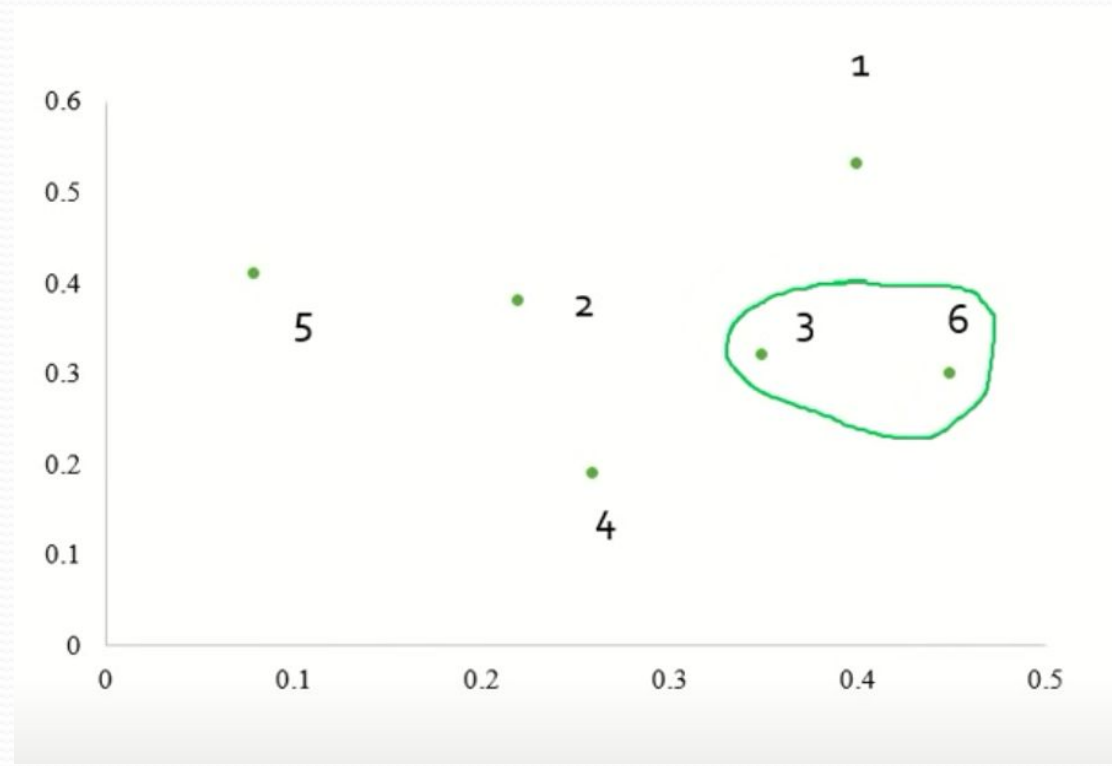
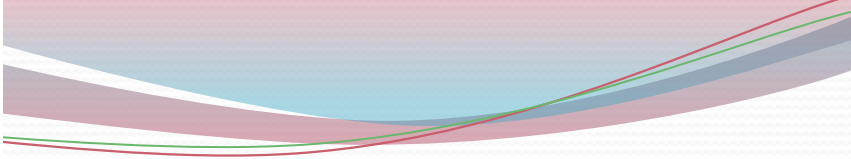
$$\begin{aligned}\text{Distance (P1,P2)} &= \sqrt{(0.40 - 0.22)^2 + (0.53 - 0.38)^2} \\ (0.40,0.53), (0.22,0.38) &= \sqrt{(0.18)^2 + (0.15)^2} \\ &= \sqrt{0.0324 + 0.0225} \\ &= \sqrt{0.0549} \\ &= 0.23\end{aligned}$$

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

- The distance matrix is

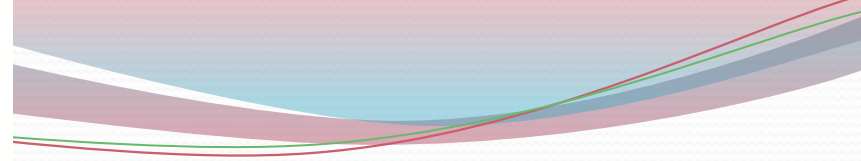
	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.24	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0



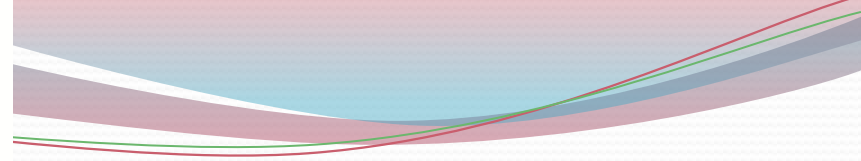


	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.24	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0



- To update the distance matrix  $\text{MIN}[\text{dist}(\text{P3}, \text{P6}), \text{P1}]$
- $\text{MIN}(\text{dist}(\text{P3}, \text{P1}), (\text{P6}, \text{P1}))$   
 $= \min[(0.22, 0.23)]$   
 $= 0.22$
- To update the distance matrix  $\text{MIN}[\text{dist}(\text{P3}, \text{P6}), \text{P2}]$
- $\text{MIN}(\text{dist}(\text{P3}, \text{P2}), (\text{P6}, \text{P2}))$   
 $= \min[(0.15, 0.25)]$   
 $= 0.15$

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.24	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

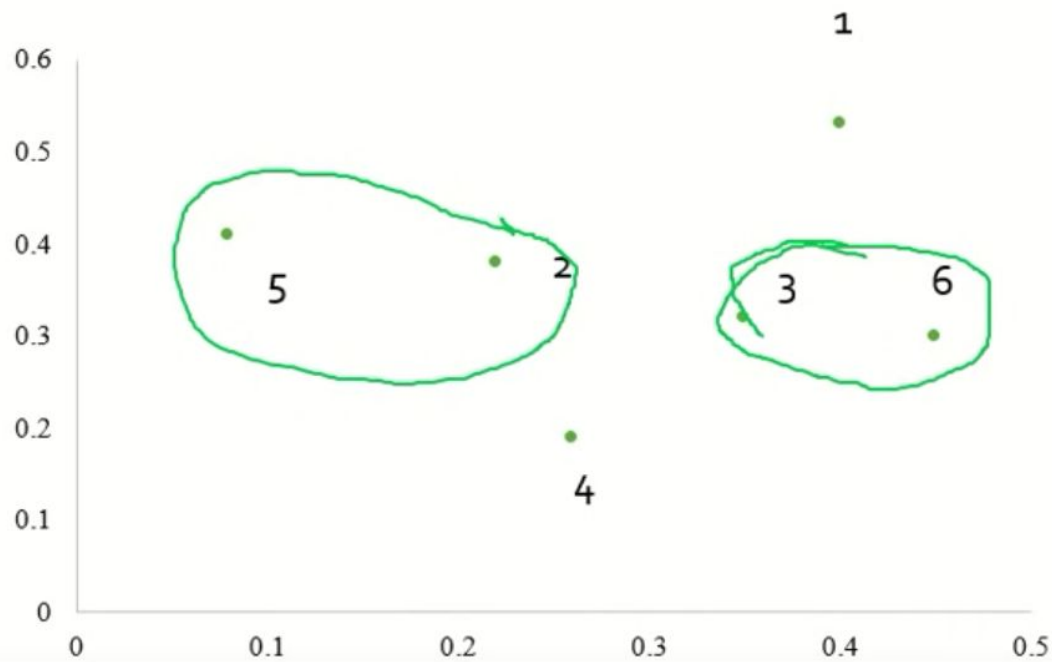


- To update the distance matrix  $\text{MIN}[\text{dist}(\text{P3}, \text{P6}), \text{P4}]$
- $\text{MIN}(\text{dist}(\text{P3}, \text{P4}), (\text{P6}, \text{P4}))$   
 $= \min[(0.15, 0.22)]$   
 $= 0.15$
- To update the distance matrix  $\text{MIN}[\text{dist}(\text{P3}, \text{P6}), \text{P5}]$
- $\text{MIN}(\text{dist}(\text{P3}, \text{P5}), (\text{P6}, \text{P5}))$   
 $= \min[(0.28, 0.39)]$   
 $= 0.28$

- The updated distance matrix for cluster P3, P6

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0



- The distance matrix fro cluster P2, P5

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

The distance matrix for cluster P2, P5

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

- To update the distance matrix  $\text{MIN}[\text{dist}(\text{P2}, \text{P5}), \text{P4}]$
- $\text{MIN}[\text{dist}(\text{P2}, \text{P4}), (\text{P5}, \text{P4})]$   
 $= \min[(0.20, 0.29)]$   
 $= 0.20$

- To update the distance matrix  $\text{MIN}[\text{dist}(\text{P2}, \text{P5}), \text{P1}]$
- $\text{MIN}[\text{dist}(\text{P2}, \text{P1}), (\text{P5}, \text{P1})]$   
 $= \min[(0.23, 0.34)]$   
 $= 0.23$
- To update the distance matrix  $\text{MIN}[\text{dist}(\text{P2}, \text{P5}), (\text{P3}, \text{P6})]$
- $\text{MIN}[\text{dist}(\text{P2}, (\text{P3}, \text{P6})), (\text{P5}, (\text{P3}, \text{P6}))]$   
 $= \min[(0.15, 0.28)]$   
 $= 0.15$

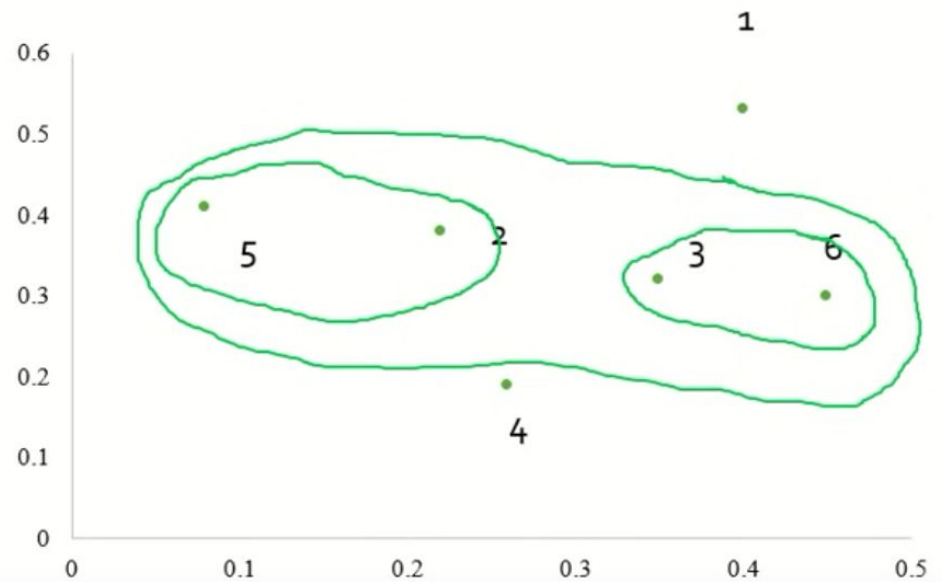
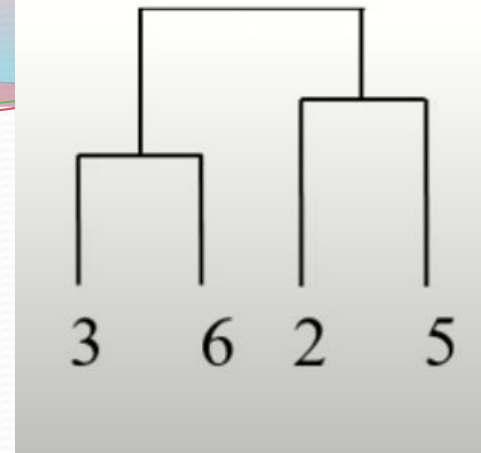
- The updated distance matrix for cluster P2,P5

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0



	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0



- To update the distance matrix  $\text{MIN}[\text{dist}((P2,P5),(P3,P6)),P4]$
- $\text{MIN}[\text{dist}((P2,P5),P4), ((P3,P6),P4)]$   
 $= \min[(0.20,0.15)]$   
 $= 0.15$

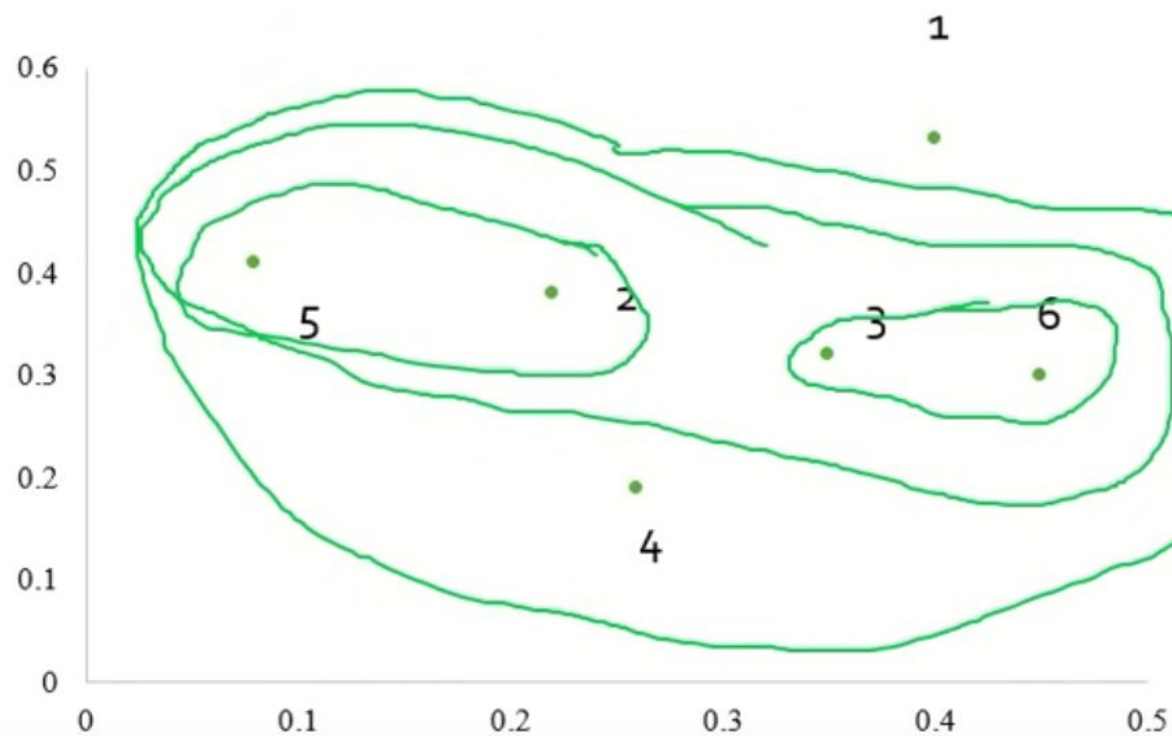
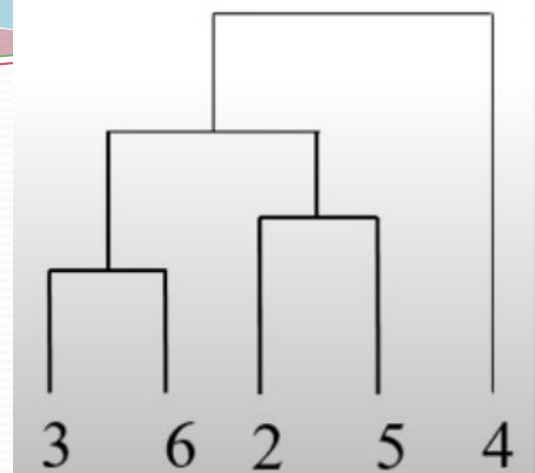
- To update the distance matrix  $\text{MIN}[\text{dist}((P2,P5),(P3,P6)),P1]$
- $\text{MIN}[\text{dist}((P2,P5),P1), ((P3,P6),P1)]$   
 $= \min[(0.23,0.22)]$   
 $= 0.22$

- The updated distance matrix for cluster P2,P5,P3,P6

	P1	P2,P5,P3,P6	P4
P1	0		
P2,P5,P3,P6	0.22	0	
P4	0.37	0.15	0

	P1	P2,P5,P3,P6	P4
P1	0		
P2,P5,P3,P6	0.22	0	
P4	0.37	0.15	0

	P1	P2,P5,P3,P6	P4
P1	0		
P2,P5,P3,P6	0.22	0	
P4	0.37	0.15	0

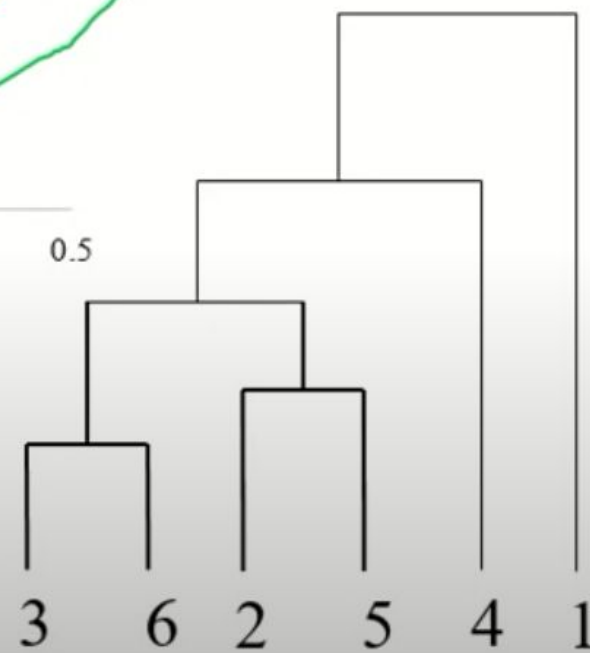
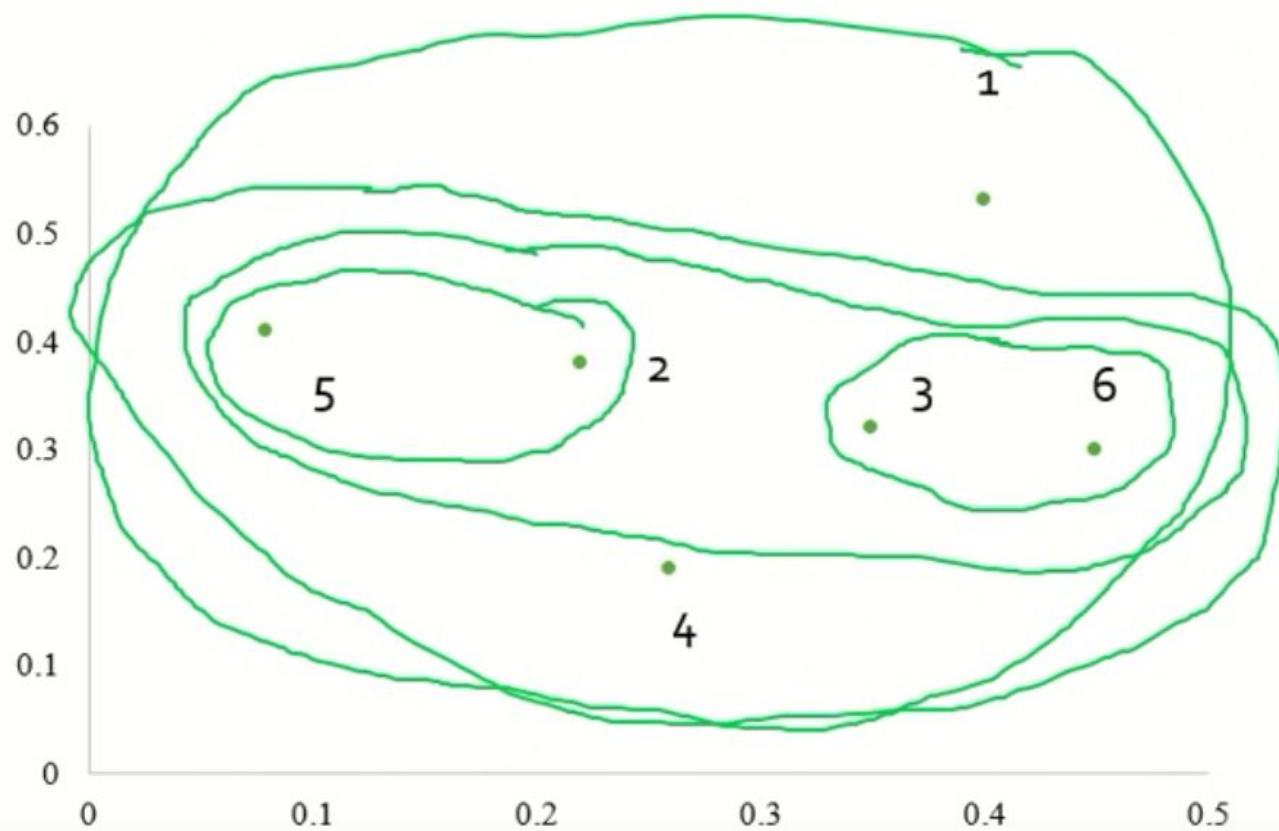


- To update the distance matrix  $\text{MIN}[\text{dist}(P2, P5, P3, P6), P4]$
- $\text{MIN}[\text{dist}((P2, P5, P3, P6), P1), (P4, P1)]$   
 $= \min[(0.22, 0.37)]$   
 $= 0.22$

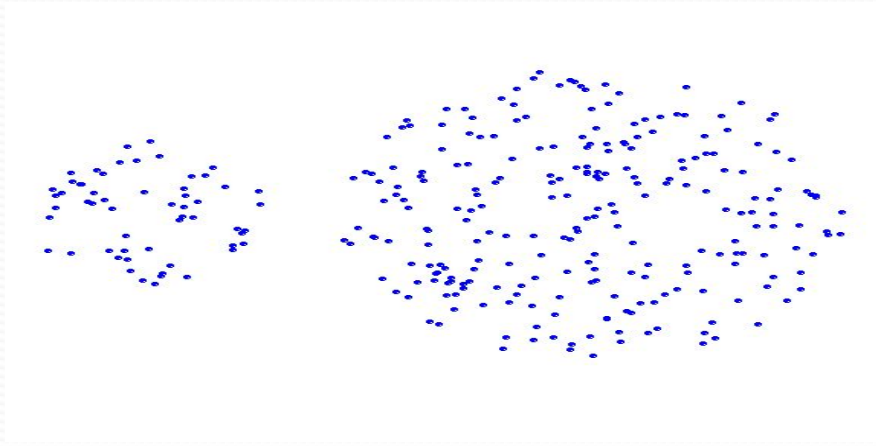
- The updated distance matrix for cluster P2,P5,P3,P6,P4

	P1	P2,P5,P3,P6,P4
P1	0	
P2,P5,P3,P6,P4	0.22	0

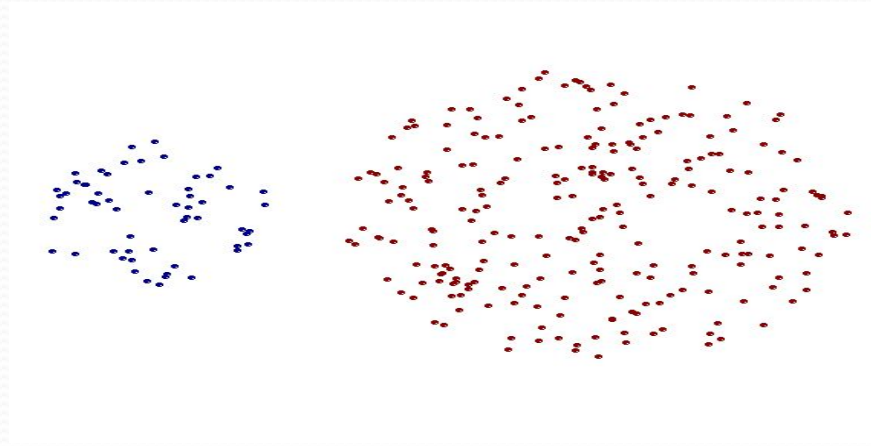




# Strength of MIN



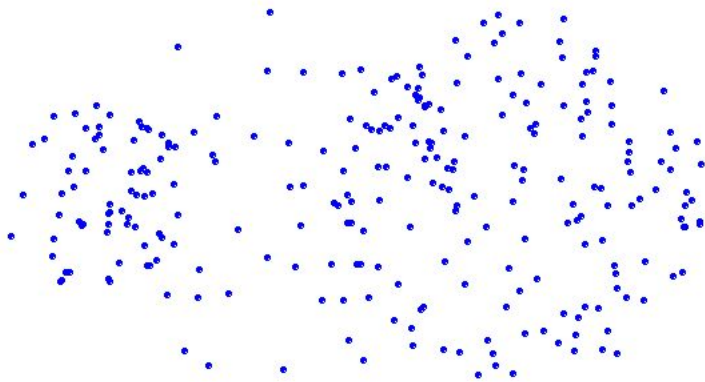
Original Points



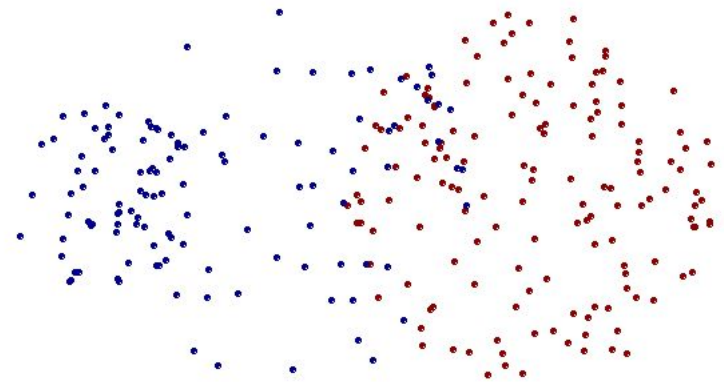
Two Clusters

- Can handle non-elliptical shapes

# Limitations of MIN



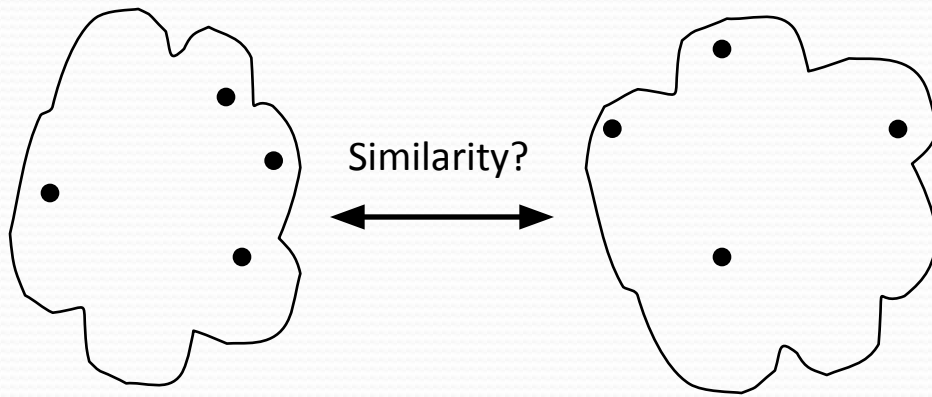
Original Points



Two Clusters

- Sensitive to noise and outliers

# How to Define Inter-Cluster Similarity

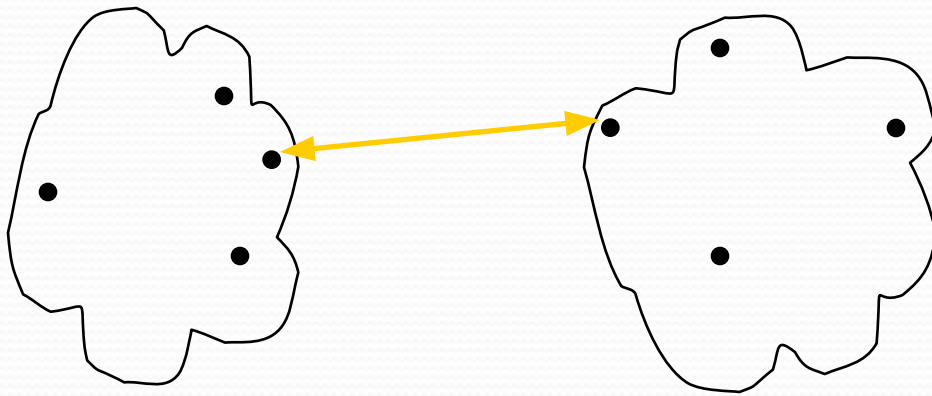


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

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

Proximity Matrix

# How to Define Inter-Cluster Similarity

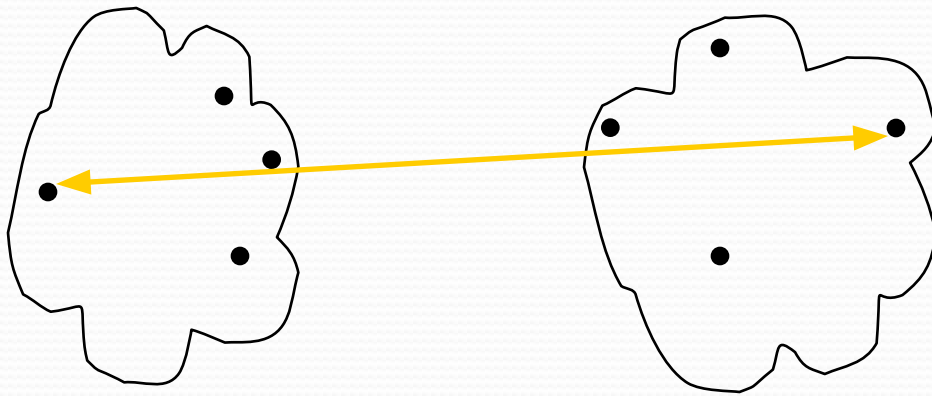


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

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

Proximity Matrix

# How to Define Inter-Cluster Similarity



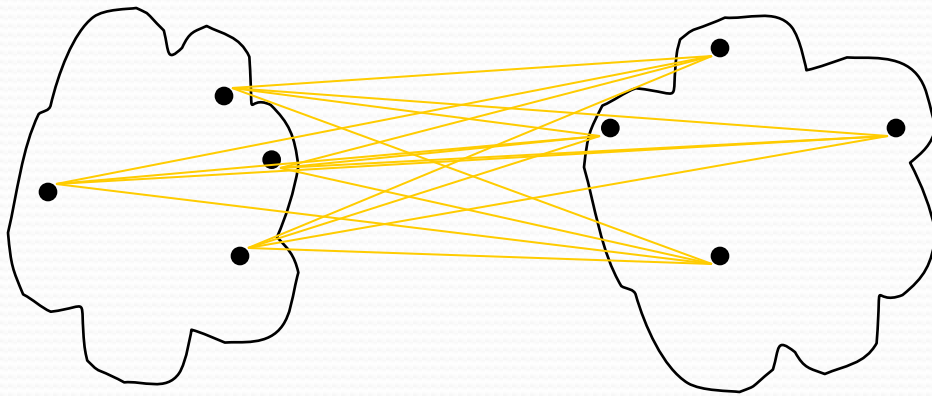
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

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

Proximity Matrix



# How to Define Inter-Cluster Similarity

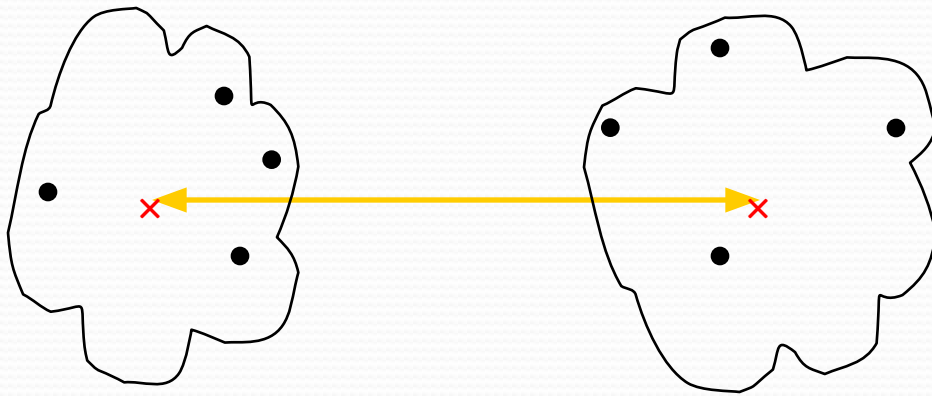


- MIN
- MAX
- **Group Average**
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

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

Proximity Matrix

# How to Define Inter-Cluster Similarity



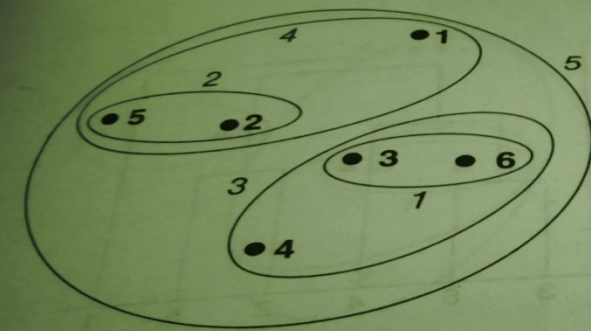
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

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

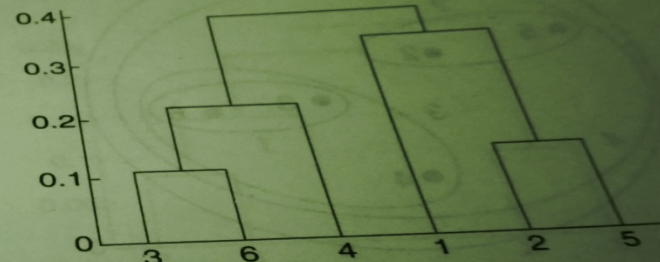
Proximity Matrix

# Complete-farthest distance or complete linkage.

## 8.3 Agglomerative Hierarchical Clustering 521



(a) Complete link clustering.



(b) Complete link dendrogram.

**Figure 8.17.** Complete link clustering of the six points shown in Figure 8.15.

$$\text{dist}(\{3,6\},\{4\})=\max(\text{dist}(3,4),\text{dist}(6,4))$$

$$=\max(0.15,0.22)$$

$$=0.22$$

$$\text{dist}(\{3,6\},\{2,5\})=\max(\text{dist}(3,2),\text{dist}(6,2),\text{dist}(3,5),\text{dist}(6,5))$$

$$=\max(0.15,0.25,0.28,0.39)$$

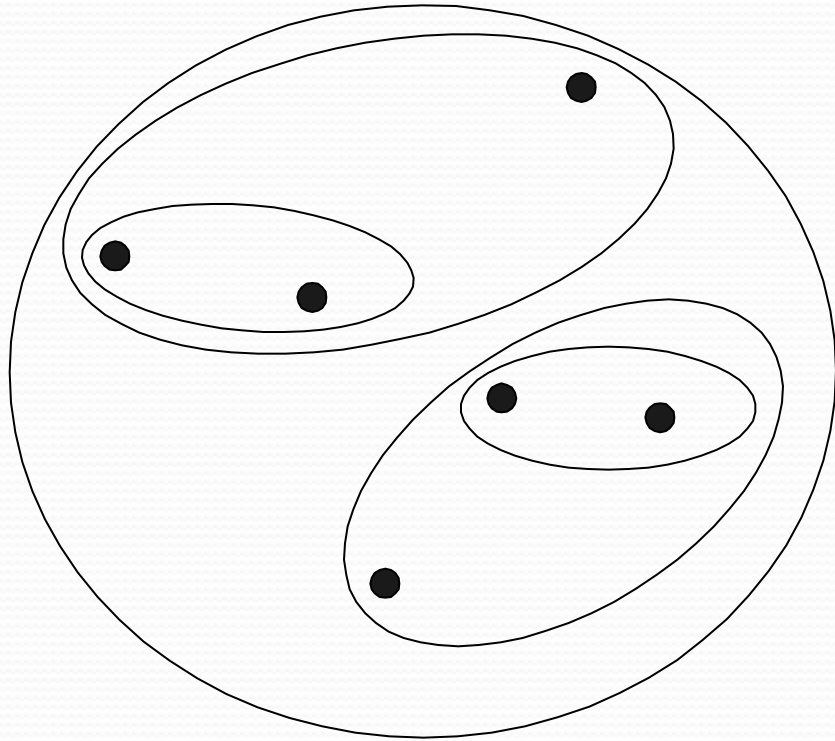
$$=0.39$$

$$\text{dist}(\{3,6\},\{1\})=\max(\text{dist}(3,1),\text{dist}(6,1))$$

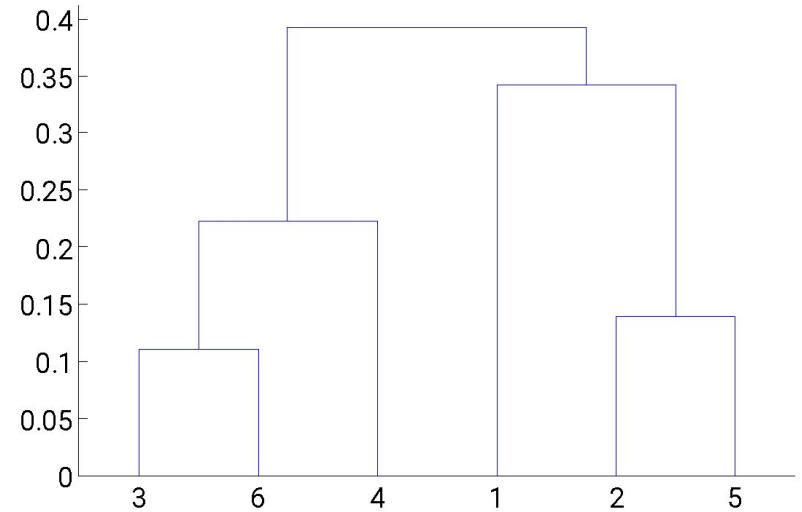
$$=\max(0.22,0.23)$$

$$=0.23$$

# Hierarchical Clustering: MAX

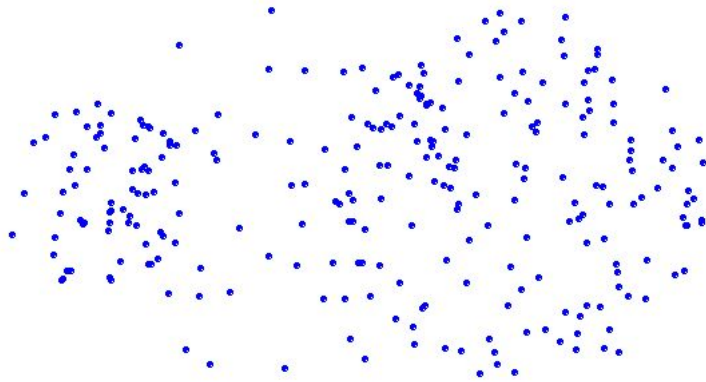


Nested Clusters

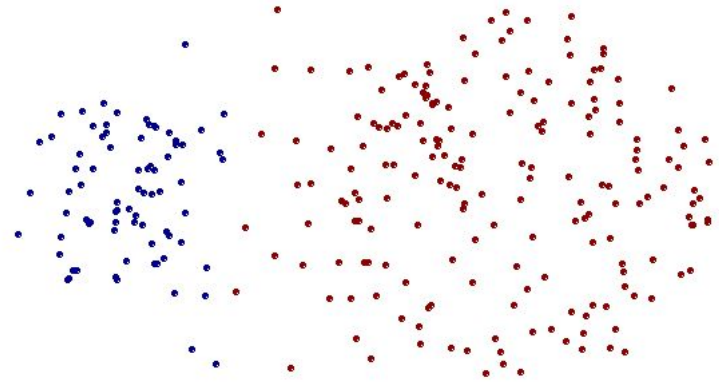


Dendrogram

# Strength of MAX



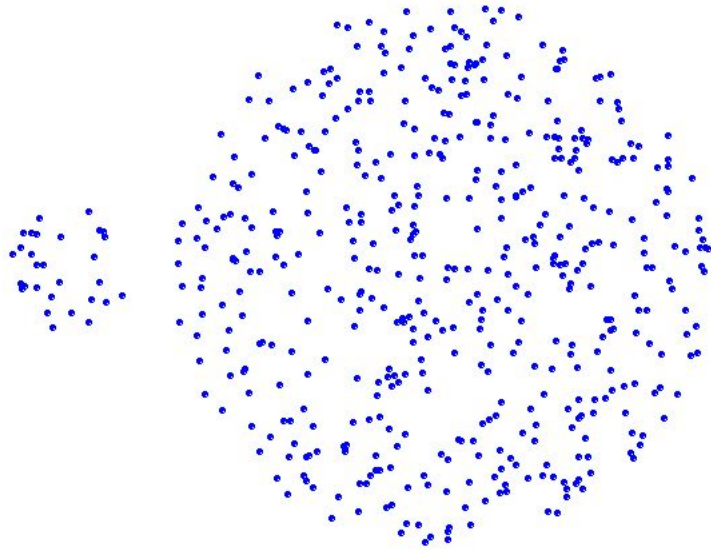
Original Points



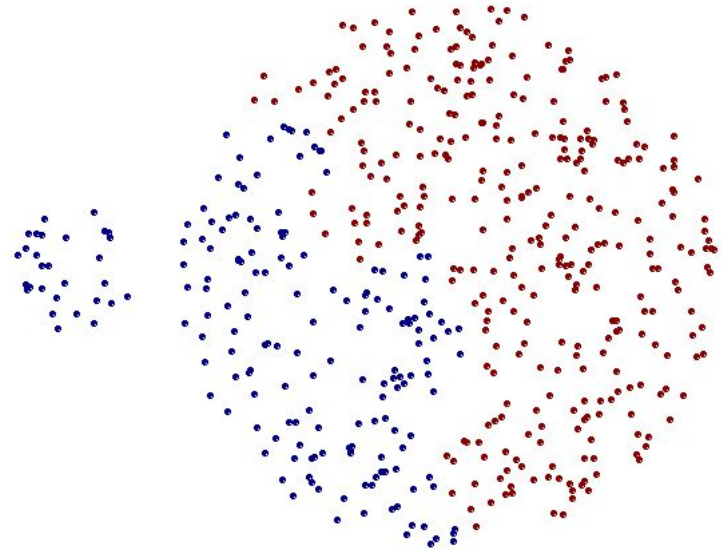
Two Clusters

- Less susceptible to noise and outliers

# Limitations of MAX



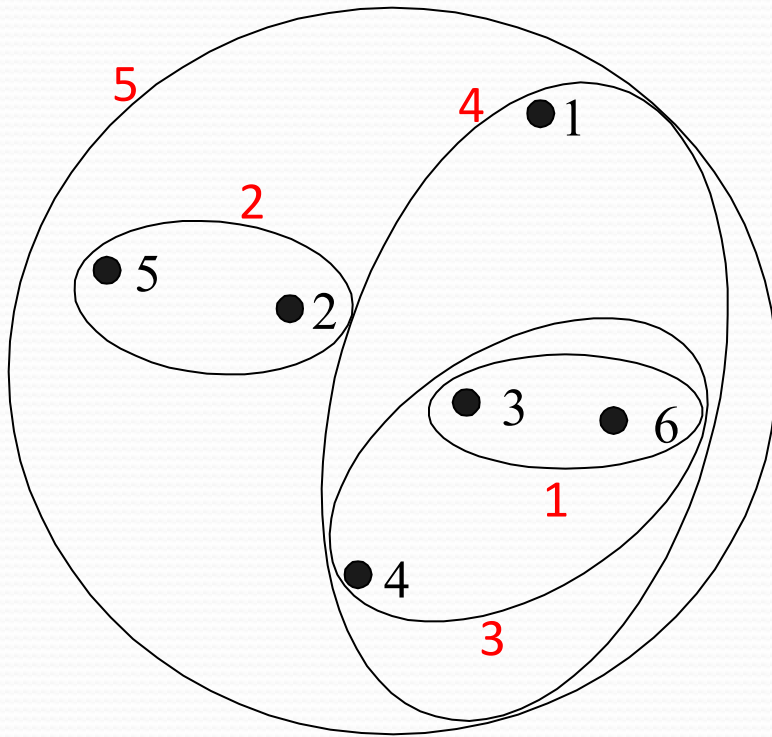
Original Points



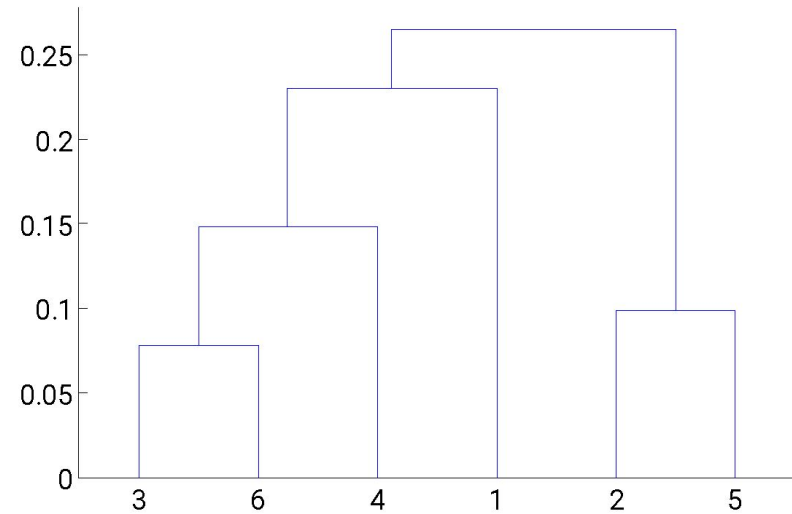
Two Clusters

- Tends to break large clusters
- Biased towards globular clusters

# Hierarchical Clustering: Group Average



Nested Clusters



Dendrogram

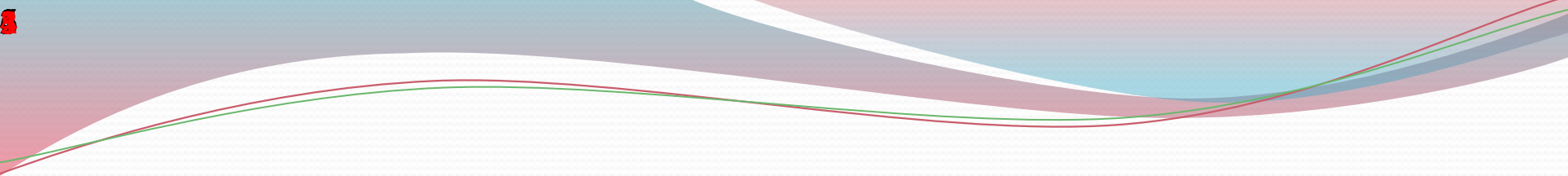
# Hierarchical Clustering: Group Average

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

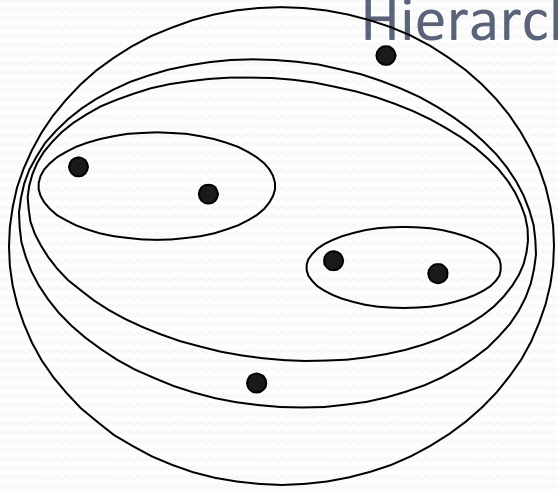


# Cluster Similarity: Ward's Method

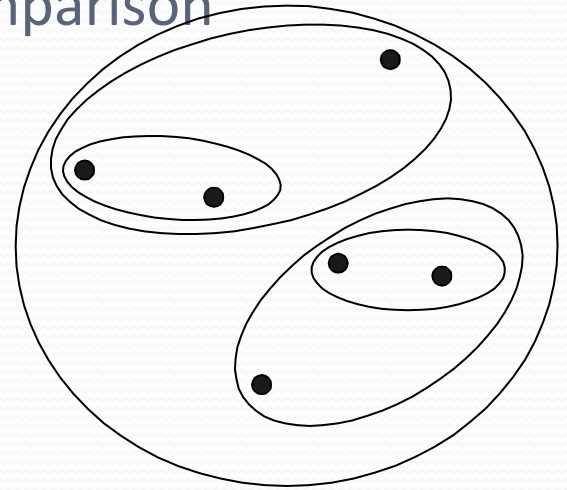
- Similarity of two clusters is based on the increase in squared error when two clusters are merged
  - Similar to group average if distance between points is distance squared
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of K-means
  - Can be used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



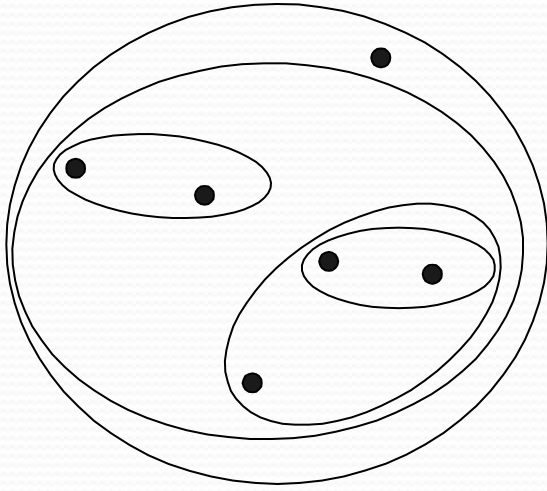
# Hierarchical Clustering: Comparison



MIN

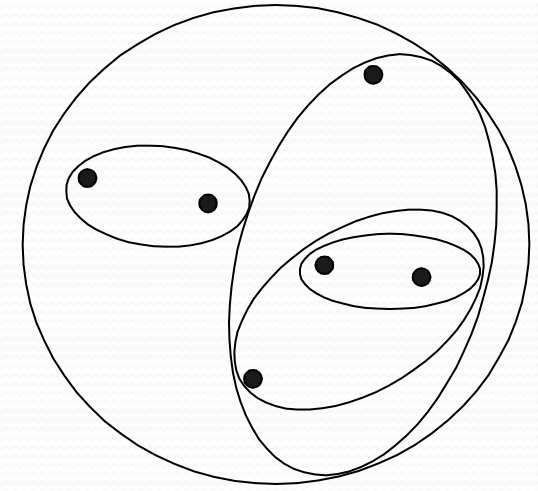


MAX



Group Average

Ward's Method



# Hierarchical Clustering: Time and Space requirements

- $O(N^2)$  space since it uses the proximity matrix.
  - $N$  is the number of points.
- $O(N^3)$  time in many cases
  - There are  $N$  steps and at each step the size,  $N^2$ , proximity matrix must be updated and searched
  - Complexity can be reduced to  $O(N^2 \log(N))$  time for some approaches

# 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

# Another Example

- A hierarchical clustering of distances in kilometers between some Italian cities. The method used is **single-linkage**.
- Input **distance matrix** ( $L = 0$  for all the clusters):

[http://home.deib.polimi.it/matteucc/Clustering/tutorial\\_html/hierarchical.html](http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html)

	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0

- The nearest pair of cities is MI and TO, at distance 138
- The level of the new cluster is  $L(\text{MI}/\text{TO}) = 138$  and the new sequence number is  $m = 1$ .
- Then we compute the distance from this new Compound object to all other objects

	BA	FI	MI/TO	NA	RM
BA	0	662	877	255	412
FI	662	0	295	468	268
MI/TO	877	295	0	754	564
NA	255	468	754	0	219
RM	412	268	564	219	0

- $\min d(i,j) = d(\text{NA}, \text{RM}) = 219 \Rightarrow$  merge NA and RM into a new cluster called NA/RM
- $L(\text{NA/RM}) = 219$
- $m = 2$

	<b>BA</b>	<b>FI</b>	<b>MI/TO</b>	<b>NA/RM</b>
<b>BA</b>	0	662	877	255
<b>FI</b>	662	0	295	268
<b>MI/TO</b>	877	295	0	564
<b>NA/RM</b>	255	268	564	0

- $\min d(i,j) = d(\text{BA}, \text{NA/RM}) = 255 \Rightarrow$  merge BA and NA/RM into a new cluster called BA/NA/RM
- $L(\text{BA/NA/RM}) = 255$
- $m = 3$

	<b>BA/NA/RM</b>	<b>FI</b>	<b>MI/TO</b>
<b>BA/NA/RM</b>	0	268	564
<b>FI</b>	268	0	295
<b>MI/TO</b>	564	295	0

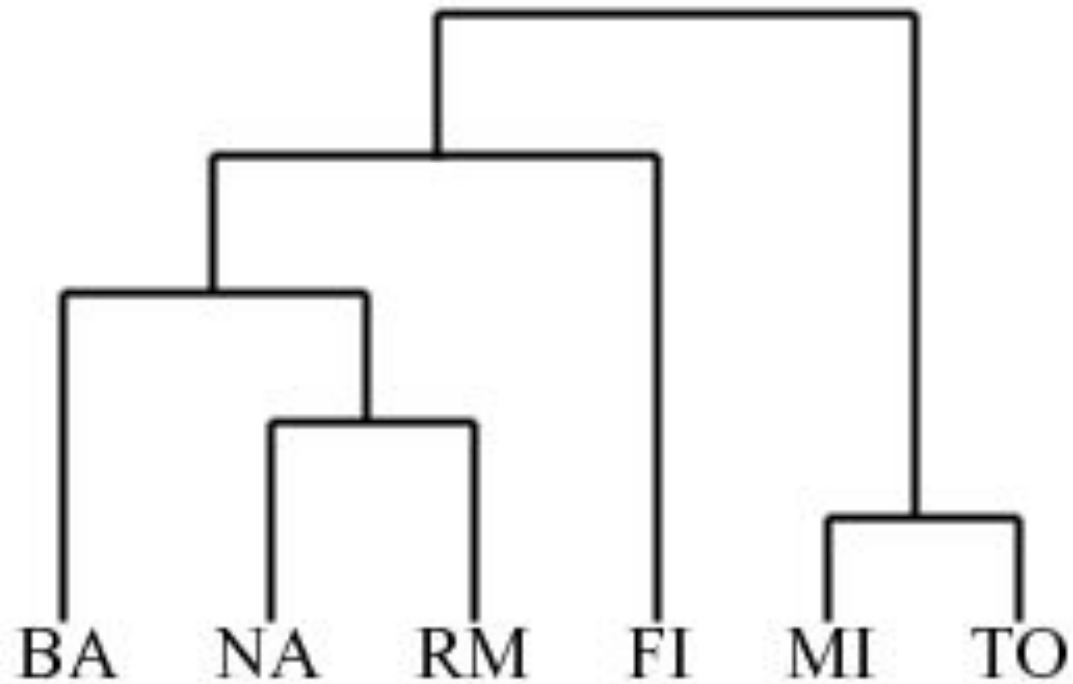


- $\min d(i,j) = d(\text{BA/NA/RM}, \text{FI}) = 268 \Rightarrow$  merge BA/NA/RM and FI into a new cluster called BA/FI/NA/RM
- $L(\text{BA/FI/NA/RM}) = 268$
- $m = 4$

	BA/FI/NA/RM	MI/TO
BA/FI/NA/RM	0	295
MI/TO	295	0

- Finally, we merge the last two clusters at level 295.

# Summary



# Advantages

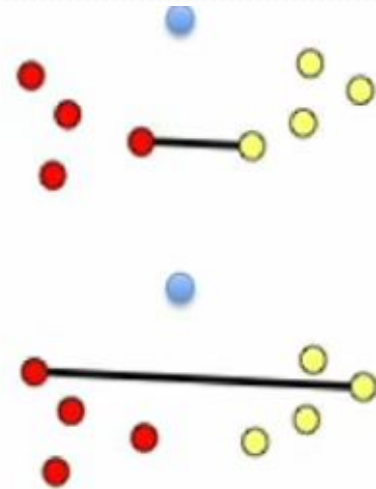
- No apriori information about the number of clusters required.
- Easy to implement and gives best result in some cases.
- [http://home.deib.polimi.it/matteucc/Clustering/tutorial\\_html/hierarchical.html](http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html)

# Disadvantages

- Algorithm can never undo what was done previously.
- Time complexity of at least  $O(n^2 \log n)$  is required, where 'n' is the number of data points.
- Based on the type of distance matrix chosen for merging different algorithms can suffer with one or more of the following:
  - Sensitivity to noise and outliers
  - Breaking large clusters
  - Difficulty handling different sized clusters and convex shapes
- No objective function is directly minimized
- Sometimes it is difficult to identify the correct number of clusters by the dendrogram.

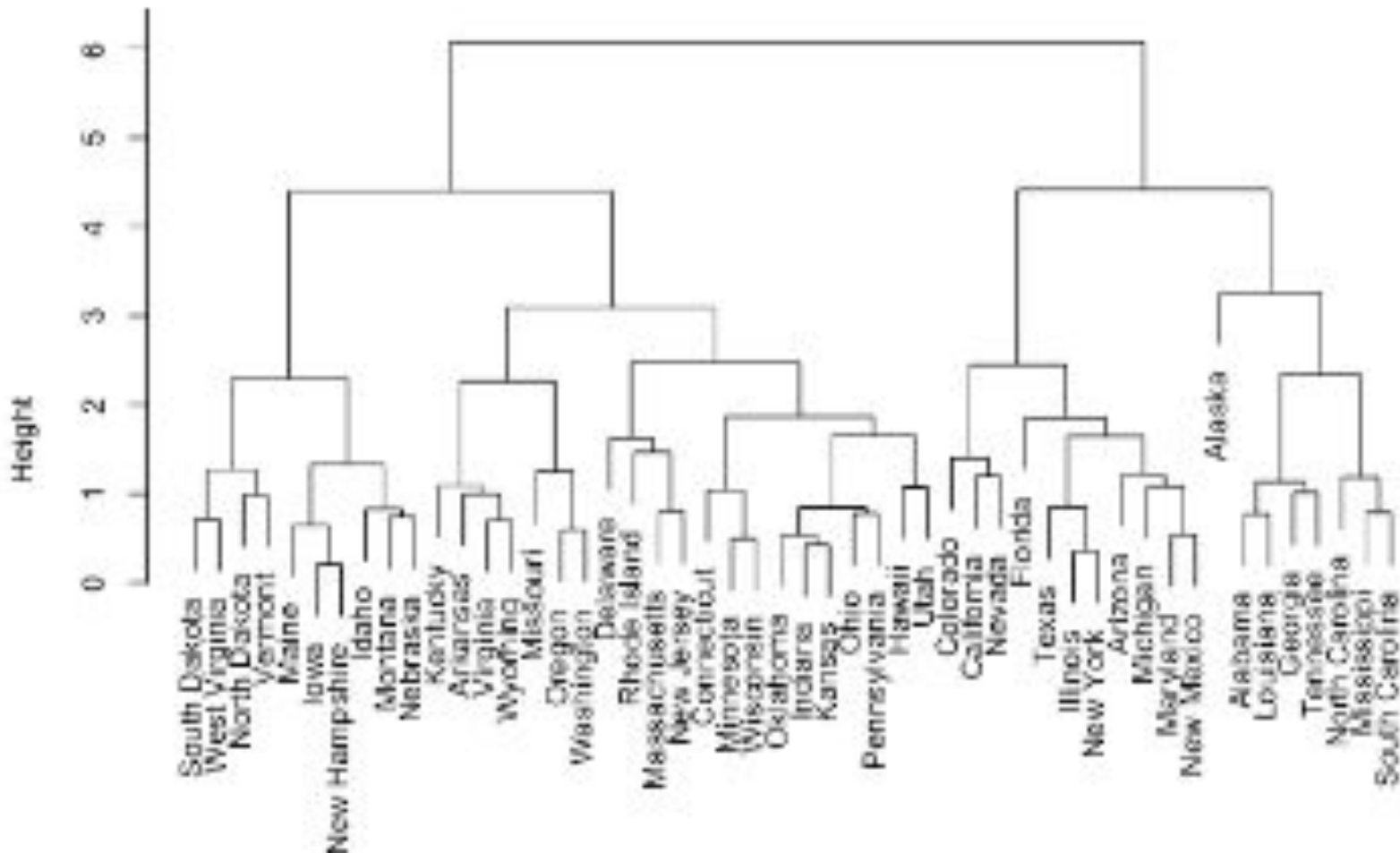
# Single link vs. complete link

- Single link:  $D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$ 
  - distance between closest elements in clusters
  - produces long chains  $a \rightarrow b \rightarrow c \rightarrow \dots \rightarrow z$
- Complete link:  $D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$ 
  - distance between farthest elements in clusters
  - forces “spherical” clusters with consistent “diameter”



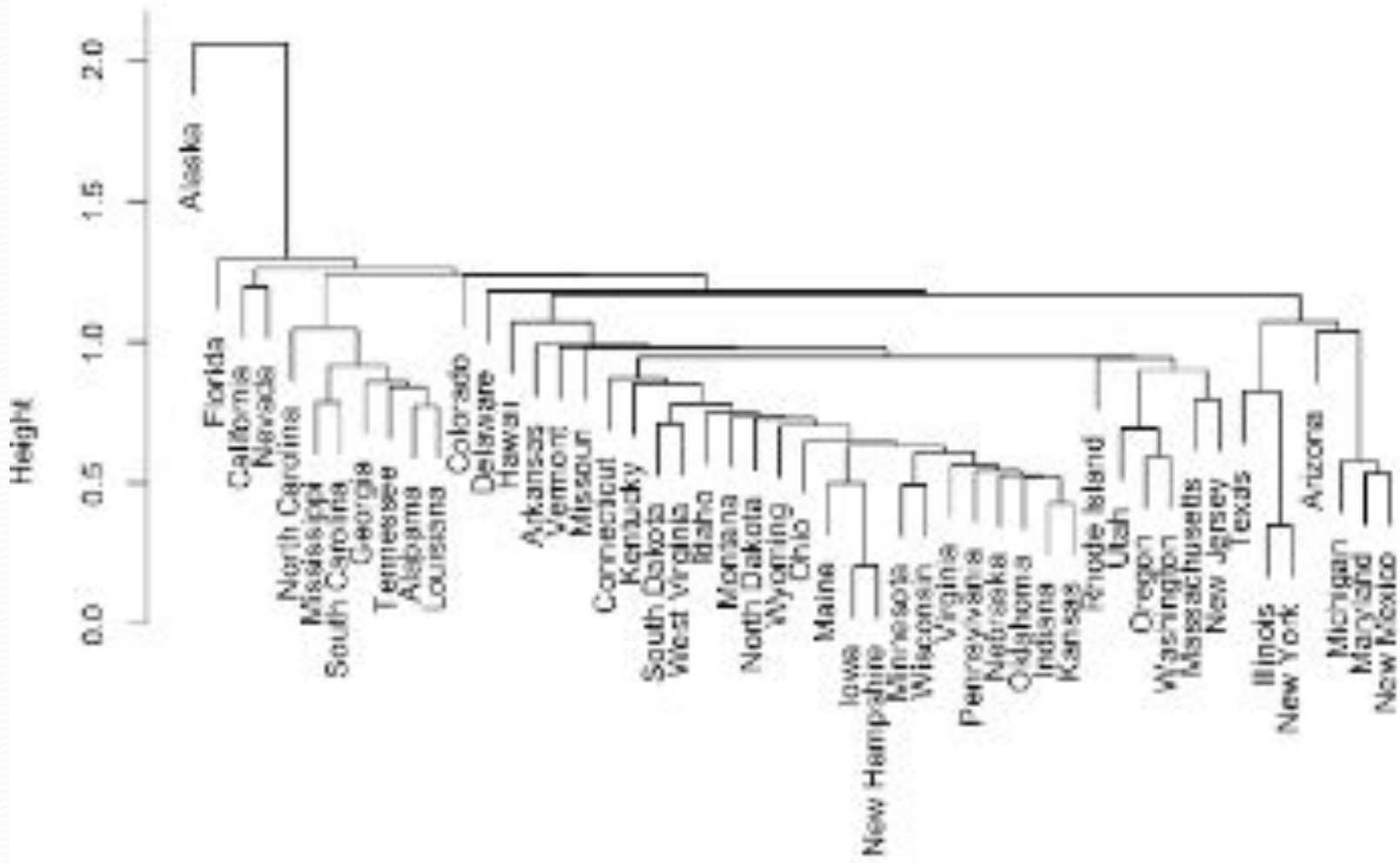
**Maximum or complete linkage clustering:** Computes all pairwise dissimilarities between the elements in cluster 1 and the elements in cluster 2, and considers the largest value of these dissimilarities as the distance between the two clusters. It tends to produce more compact clusters

**Complete Linkage**

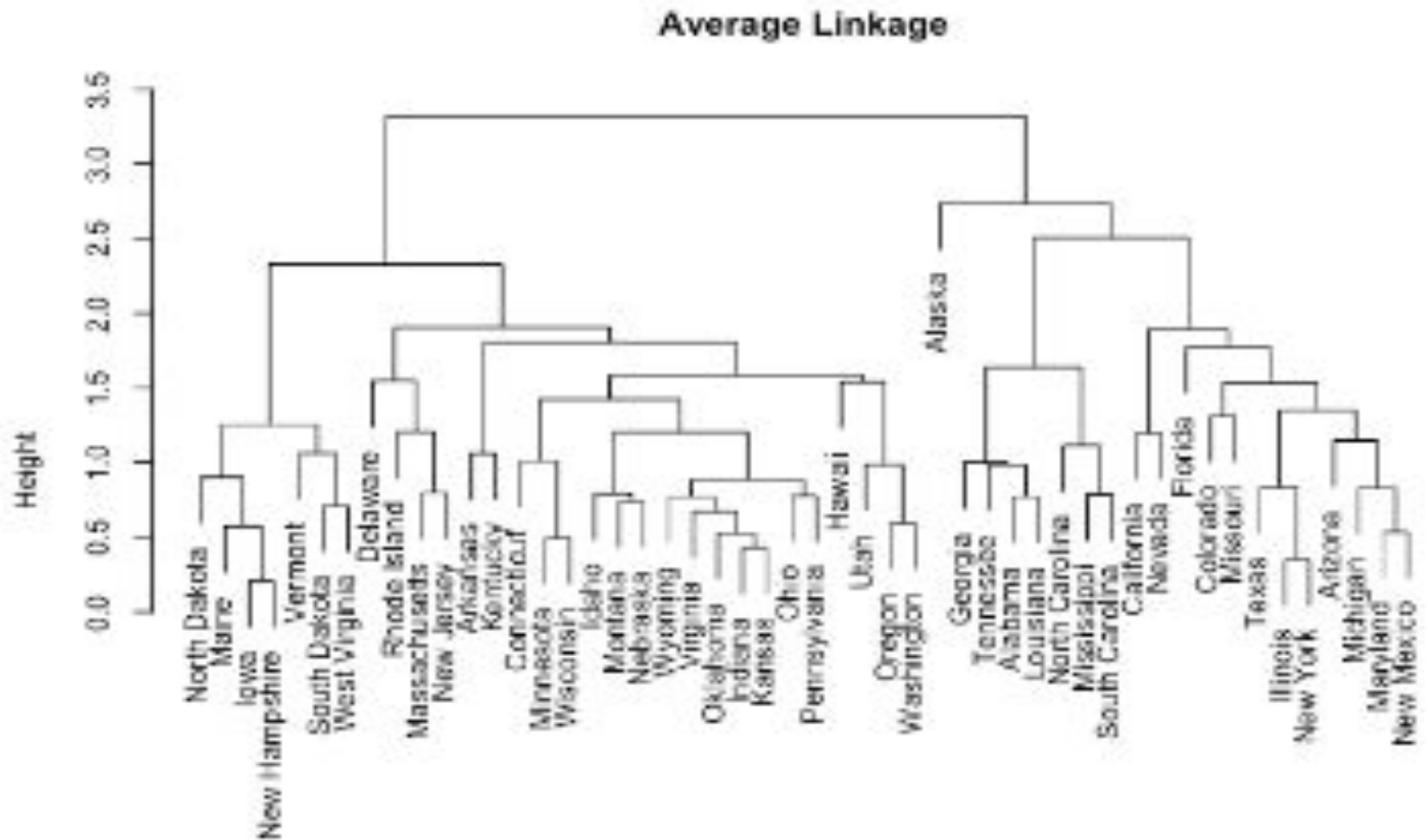


**Minimum or single linkage clustering:** Computes all pairwise dissimilarities between the elements in cluster 1 and the elements in cluster 2, and considers the smallest of these dissimilarities as a linkage criterion. It tends to produce long, “loose” clusters.

**Single Linkage**

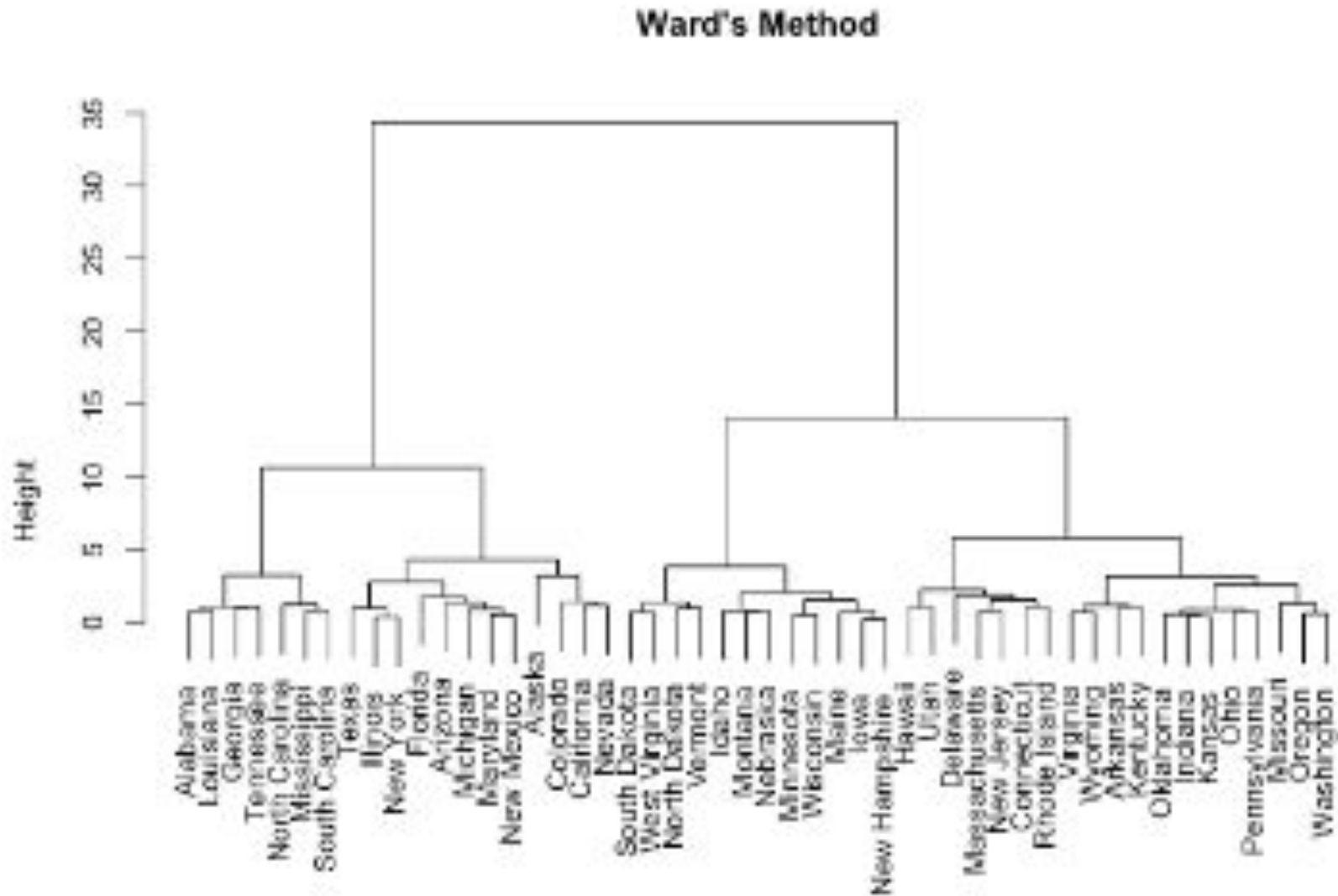


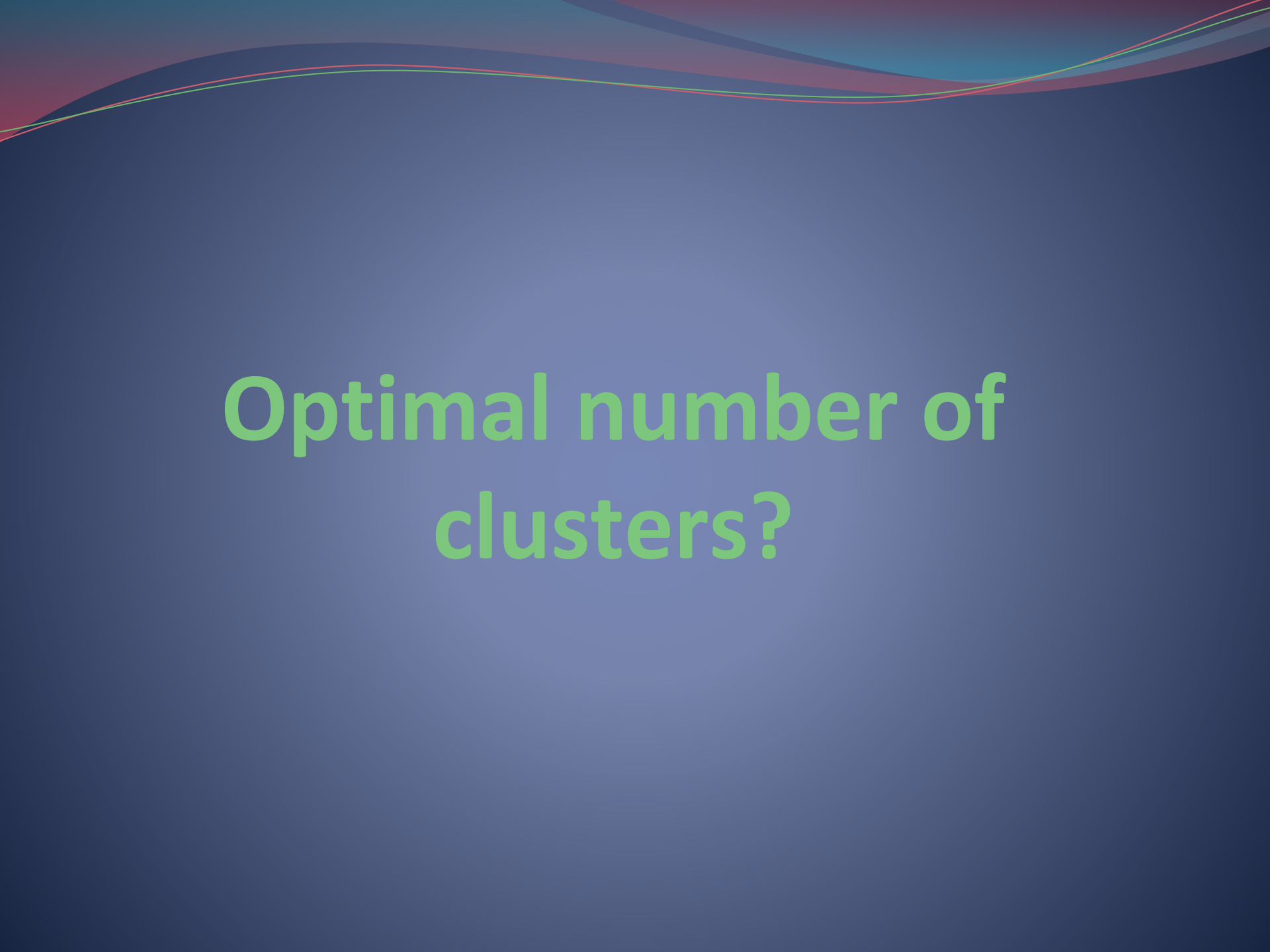
**Mean or average linkage clustering:** Computes all pairwise dissimilarities between the elements in cluster 1 and the elements in cluster 2, and considers the average of these dissimilarities as the distance between the two clusters. Can vary in the compactness of the clusters it creates





**Ward's minimum variance method:** Minimizes the total within-cluster variance. At each step the pair of clusters with the smallest between-cluster distance are merged. Tends to produce more compact clusters.





**Optimal number of  
clusters?**

# Comparison of three different methods to identify the optimal number of clusters

