

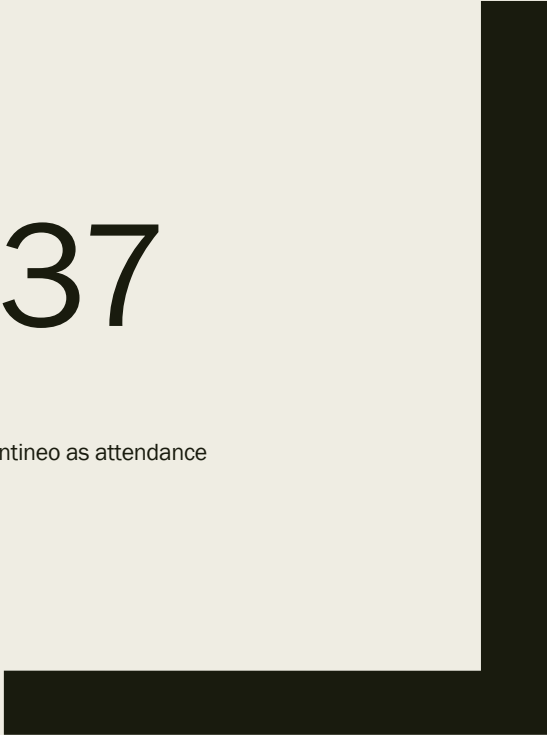


LECTURE 33-37

LA Presentations

Lecture 33: 19-Jan-2022 (Wed): 2:30 – 3:30

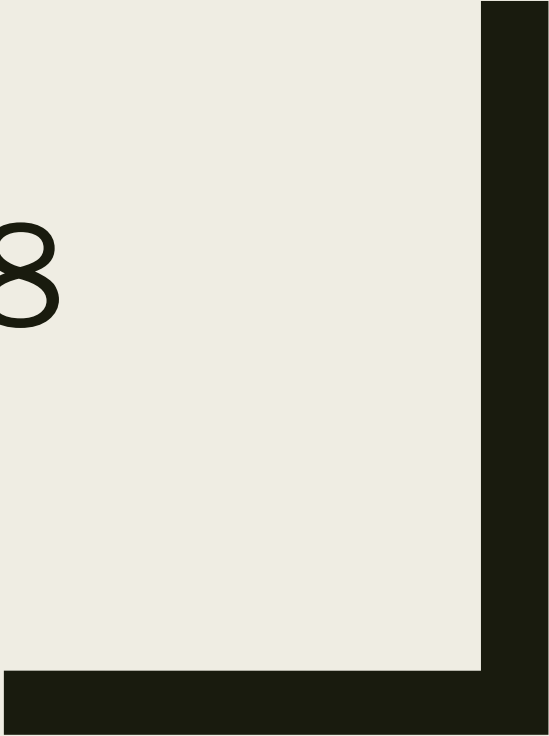
Lecture 34-37 : 20-Jan-2022(Thurs :8:50 – 12:30 pm) - Unable to enter attendance in contineo as attendance was already taken for 9:45-10:45 hr and 11:55 – 12:30 hrs





LECTURE 38

Expectation Maximization Algorithm ...
Hierarchical Clustering
Supervised Learning after Clustering
Choosing K
Dr.Vani V



EM in Gaussian Mixtures

- $z_i^t = 1$ if \mathbf{x}^t belongs to G_i , 0 otherwise (labels r^t of supervised learning);
assume $p(\mathbf{x} | G_i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

- E-step:

$$E[z_i^t | \mathcal{X}, \Phi'] = \frac{p(\mathbf{x}^t | G_i, \Phi') p(G_i)}{\sum_j p(\mathbf{x}^t | G_j, \Phi') p(G_j)}$$

- M-step:

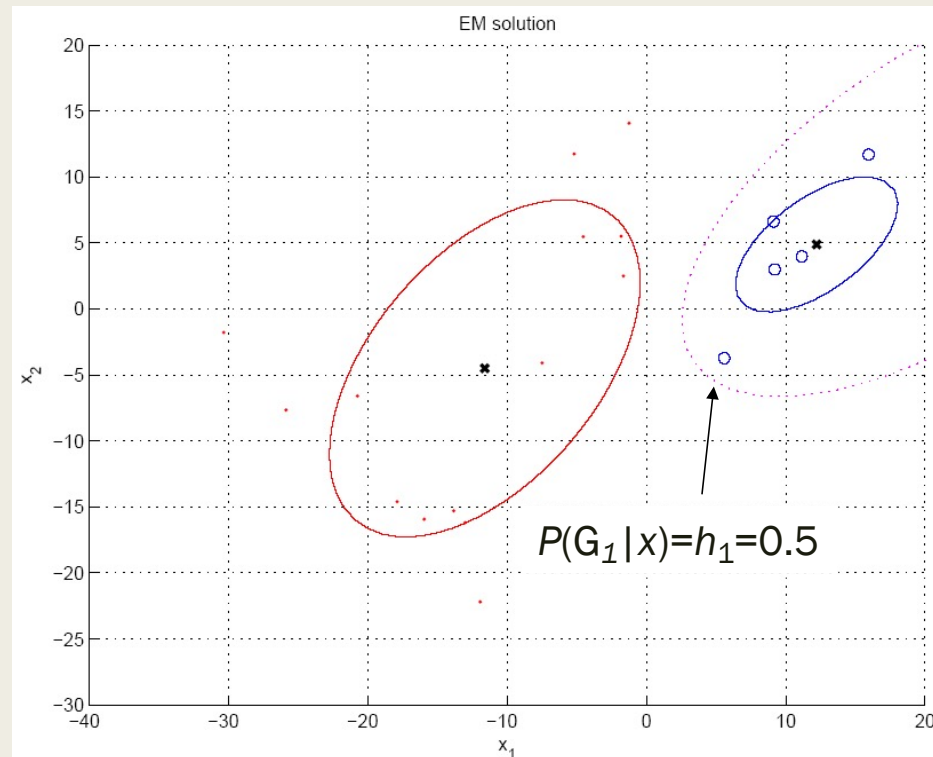
$$= p(G_i | \mathbf{x}^t, \Phi') \equiv h_i^t$$

$$p(G_i) = \frac{\sum_t h_i^t}{N} \quad \mathbf{m}_i^{l+1} = \frac{\sum_t h_i^t \mathbf{x}^t}{\sum_t h_i^t}$$

*Use estimated labels
in place of unknown
labels*

$$\mathbf{S}_i^{l+1} = \frac{\sum_t h_i^t (\mathbf{x}^t - \mathbf{m}_i^{l+1})(\mathbf{x}^t - \mathbf{m}_i^{l+1})^T}{\sum_t h_i^t}$$

Data points and the fitted Gaussians by EM, initialized by one k-means iteration. Unlike in k-means, EM allows estimating the covariance matrices. The data points labeled by greater h_i , the contours of the estimated Gaussian densities, and the separating curve of $h_i = 0.5$ (dashed line) are shown.



Mixtures of Latent Variable Models

- Regularize clusters
 1. Assume shared/diagonal covariance matrices
 2. Use PCA/FA to decrease dimensionality: Mixtures of PCA/FA

$$p(\mathbf{x}_t | G_i) = \mathcal{N}(\mathbf{m}_i, \mathbf{V}_i \mathbf{V}_i^T + \boldsymbol{\Psi}_i)$$

Can use EM to learn \mathbf{V}_i (Ghahramani and Hinton, 1997; Tipping and Bishop, 1999)

After Clustering

- Dimensionality reduction methods find correlations between features and group features
- Clustering methods find similarities between instances and group instances
- Allows knowledge extraction through
 - number of clusters,*
 - prior probabilities,*
 - cluster parameters, i.e., center, range of features.*

Example: CRM, customer segmentation

Clustering as Preprocessing

- Estimated group labels h_j (soft) or b_j (hard) may be seen as the dimensions of a new k dimensional space, where we can then learn our discriminant or regressor.
- **Local** representation (only one b_j is 1, all others are 0; only few h_j are nonzero) vs **Distributed** representation (After PCA; all z_j are nonzero)

Mixture of Mixtures

- In classification, the input comes from a mixture of classes (supervised).
- If each class is also a mixture, e.g., of Gaussians, (unsupervised), we have a mixture of mixtures:

$$p(\mathbf{x} | C_i) = \sum_{j=1}^{k_i} p(\mathbf{x} | G_{ij}) P(G_{ij})$$

$$p(\mathbf{x}) = \sum_{i=1}^K p(\mathbf{x} | C_i) P(C_i)$$

Hierarchical Clustering

- Cluster based on similarities/distances
- Distance measure between instances \mathbf{x}^r and \mathbf{x}^s

Minkowski (L_p) (Euclidean for $p = 2$)

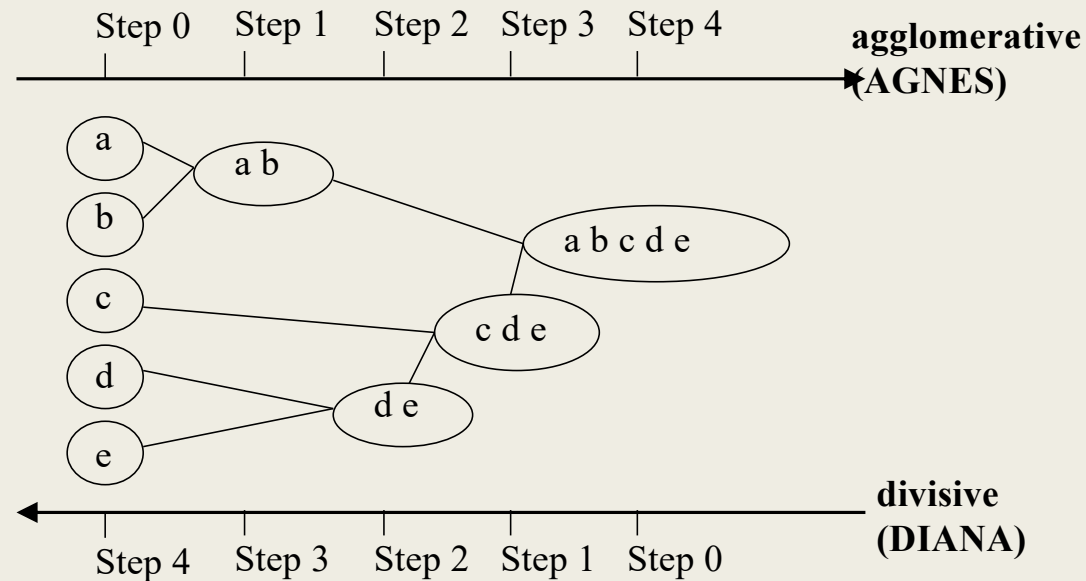
$$d_m(\mathbf{x}^r, \mathbf{x}^s) = \left[\sum_{j=1}^d (x_j^r - x_j^s)^p \right]^{1/p}$$

City-block distance

$$d_{cb}(\mathbf{x}^r, \mathbf{x}^s) = \sum_{j=1}^d |x_j^r - x_j^s|$$

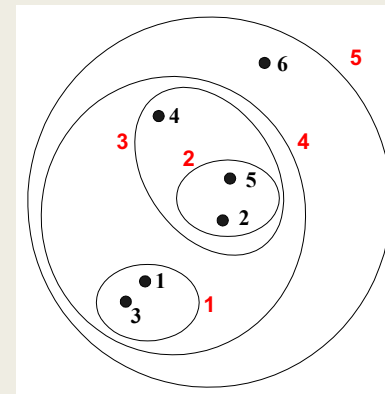
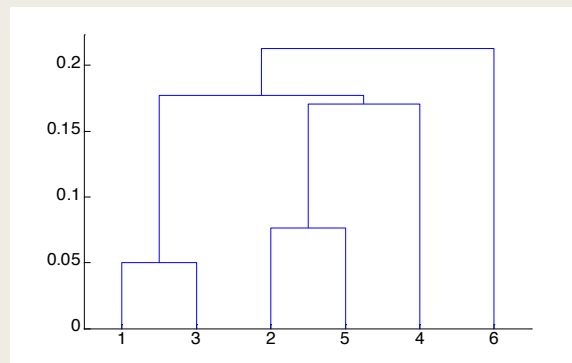
Hierarchical Clustering

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



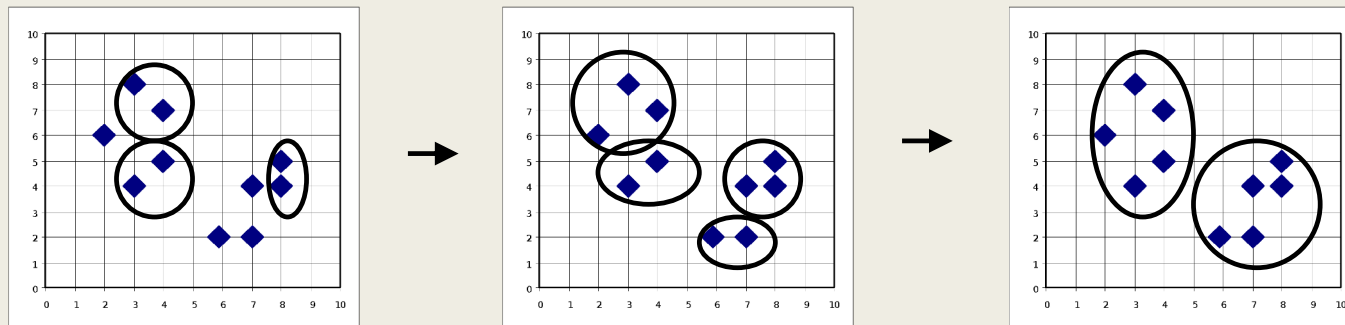
Hierarchical Clustering

- 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)

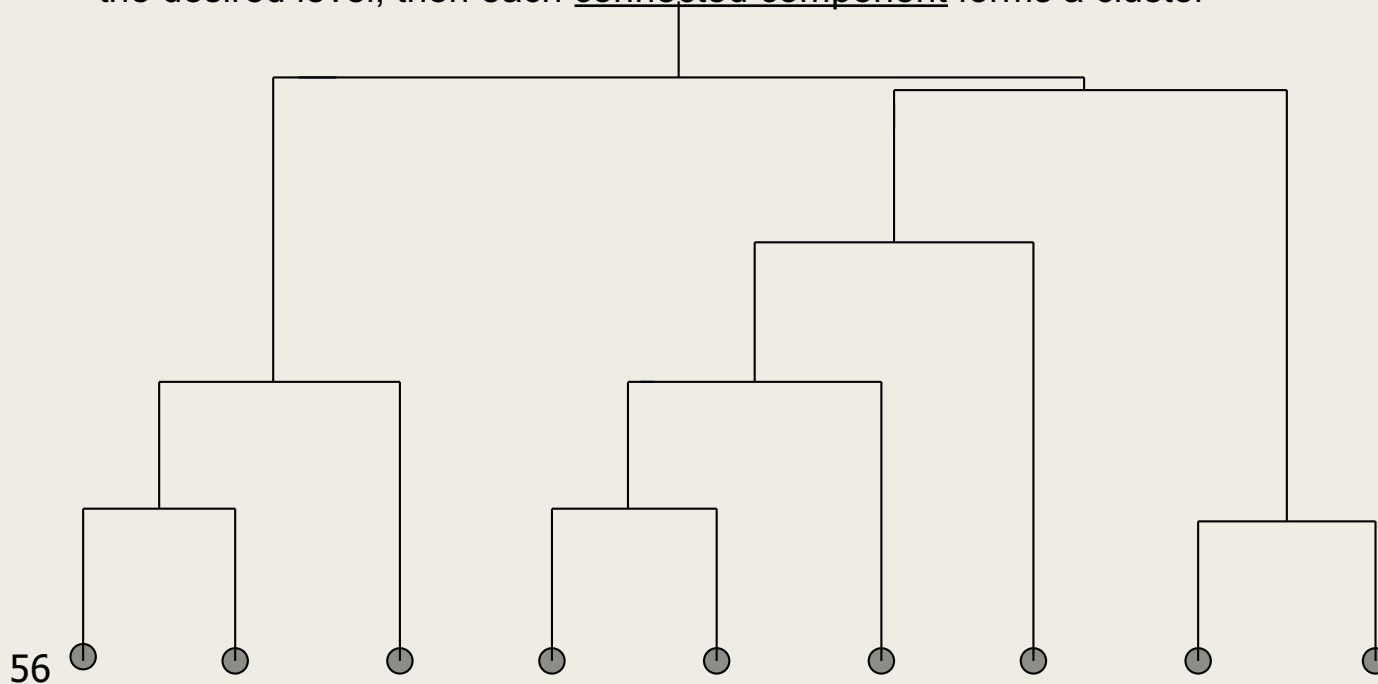
- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical packages, e.g., Splus
- Use the **single-link** method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



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

- Start with N groups each with one instance and merge two closest groups at each iteration
- Distance between two groups G_i and G_j :

- *Single-link:*

$$d(G_i, G_j) = \min_{\mathbf{x}^r \in G_i, \mathbf{x}^s \in G_j} d(\mathbf{x}^r, \mathbf{x}^s)$$

- *Complete-link:*

$$d(G_i, G_j) = \max_{\mathbf{x}^r \in G_i, \mathbf{x}^s \in G_j} d(\mathbf{x}^r, \mathbf{x}^s)$$

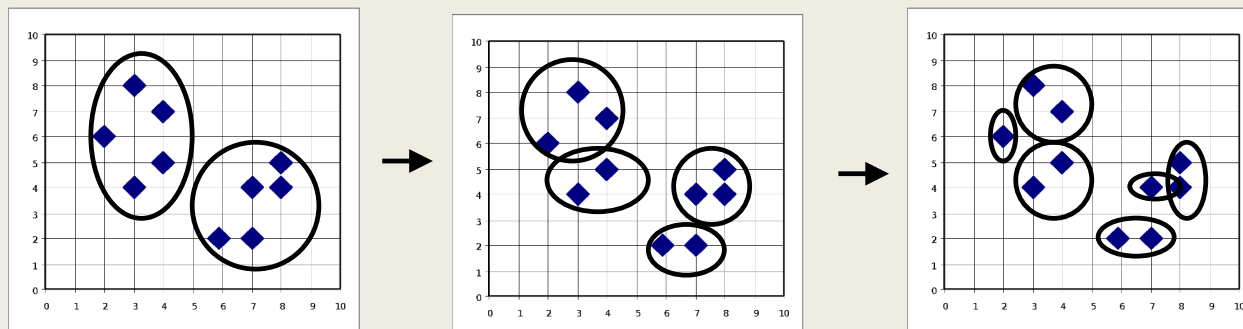
- *Average-link, centroid*

Agglomerative Clustering Algorithm

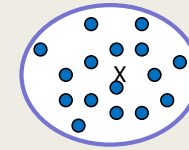
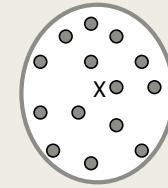
- More popular hierarchical clustering technique
- Basic algorithm is straightforward
 1. Compute the *proximity matrix*
 2. Let each data point be a cluster
 3. **Repeat**
 4. *Merge* the two closest clusters
 5. *Update* the proximity matrix
 6. **Until** only a single cluster remains
- Key operation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



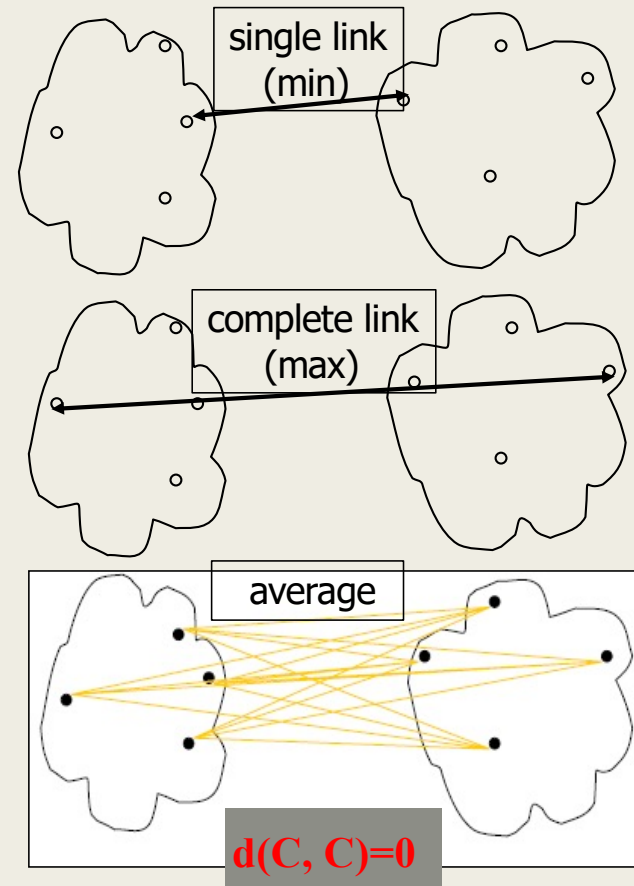
- Single link: smallest distance between an element in one cluster and an element in the other, i.e., $\text{dist}(K_i, K_j) = \min(t_{ip}, t_{jq})$
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., $\text{dist}(K_i, K_j) = \max(t_{ip}, t_{jq})$
-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

	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

Single link

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

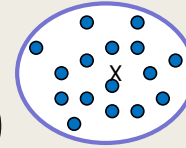
Complete link

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

Average

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

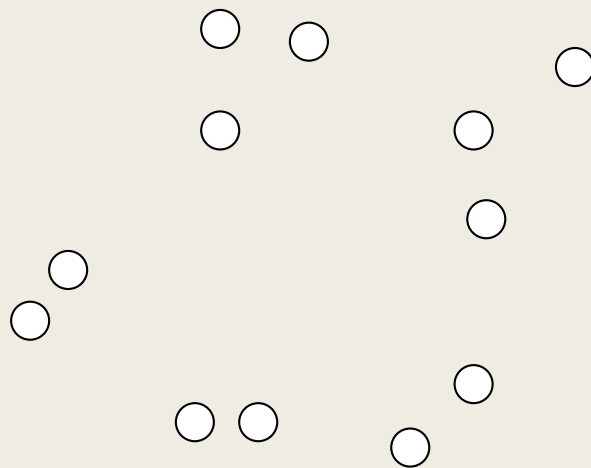
Centroid, Radius and Diameter of a Cluster (for numerical data sets)



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

Starting Situation

- Start with clusters of individual points and a proximity matrix



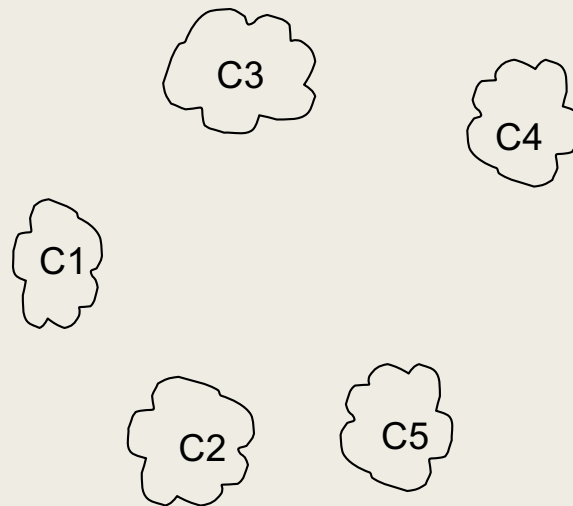
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
...						

Proximity Matrix



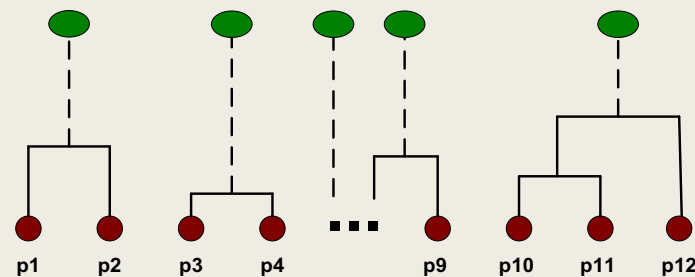
Intermediate Situation

- After some merging steps, we have some clusters



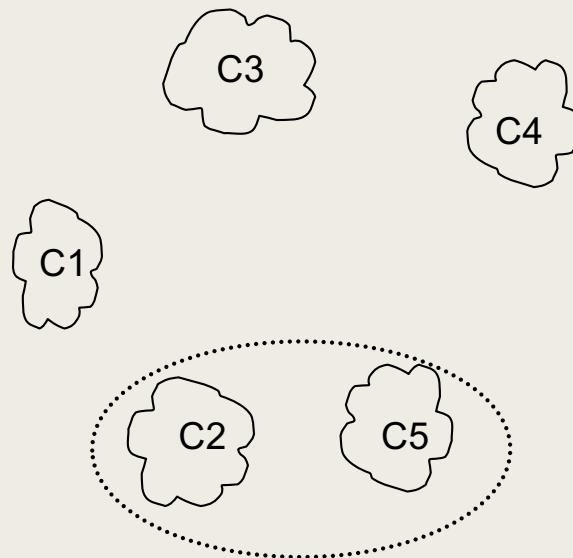
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



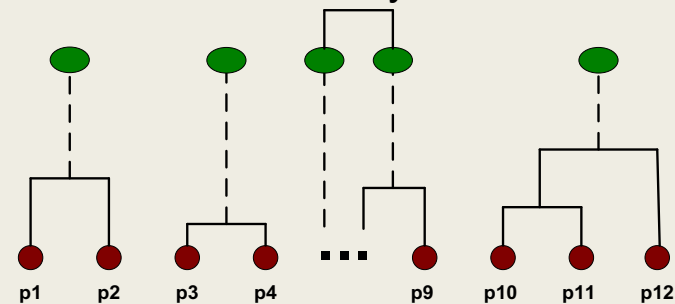
Intermediate Situation

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



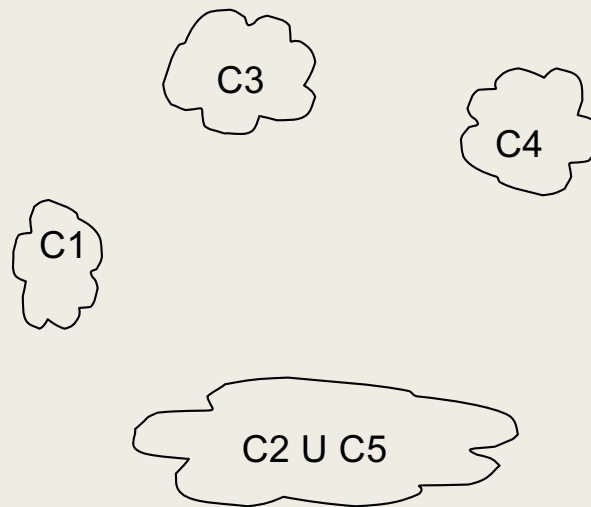
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



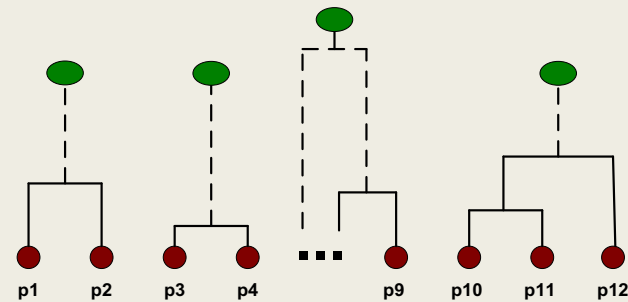
After Merging

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

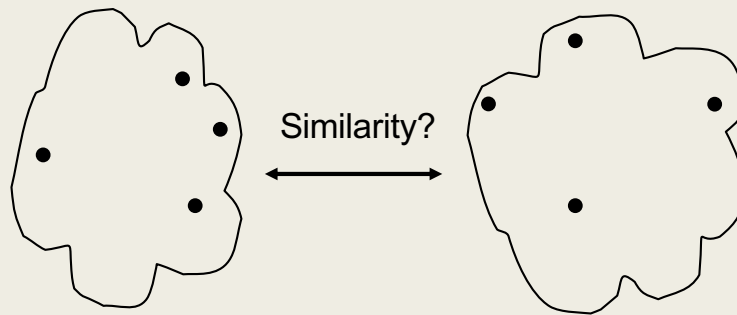


		C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity?

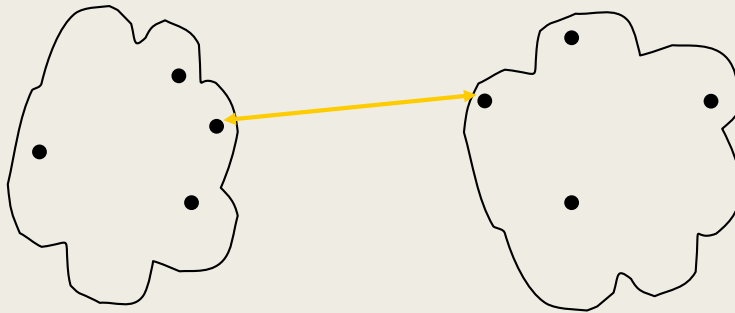


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Similarity

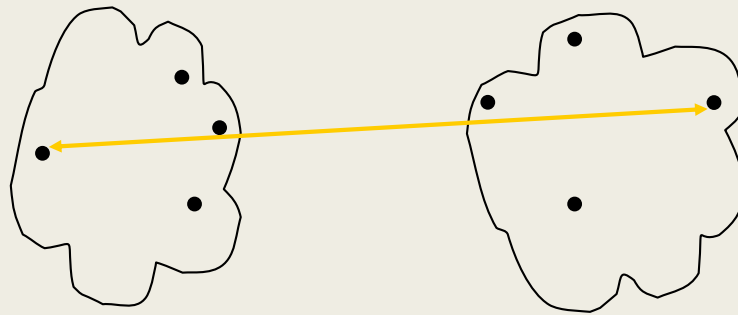


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Similarity

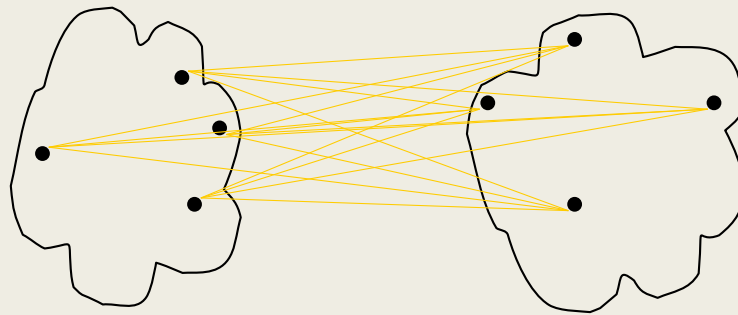


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Similarity

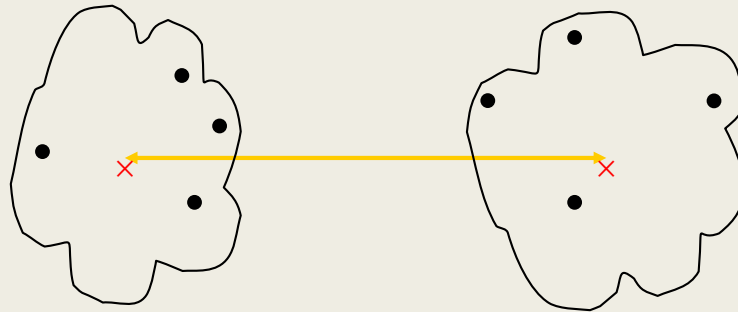


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Proximity Matrix

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

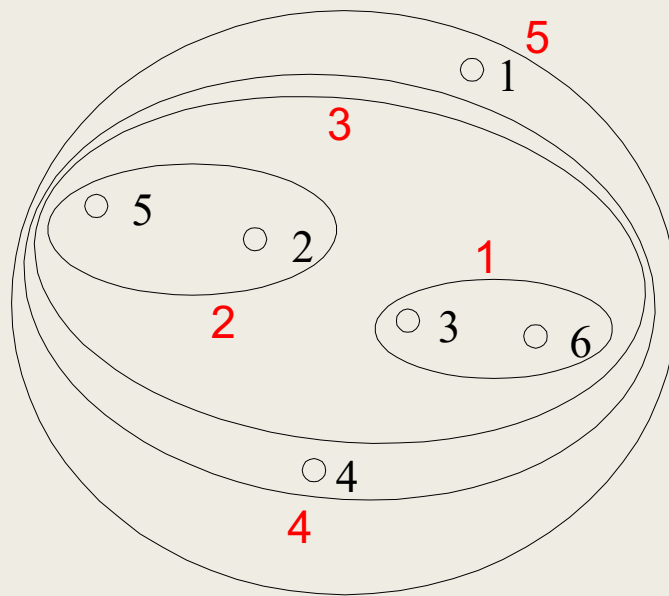
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

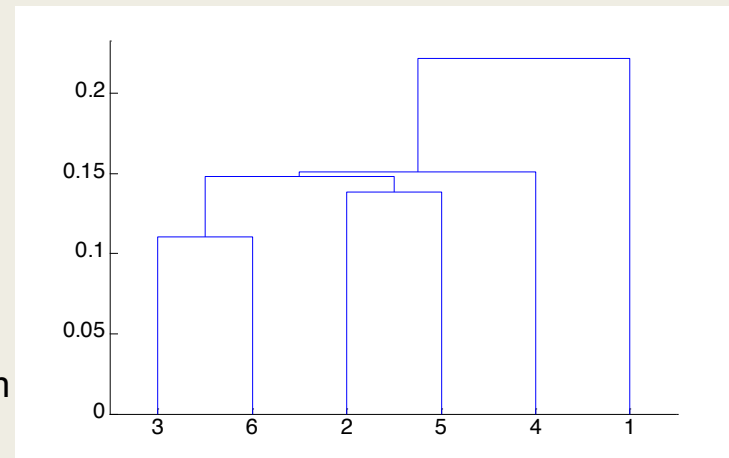
Single Link – Complete Link

- 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



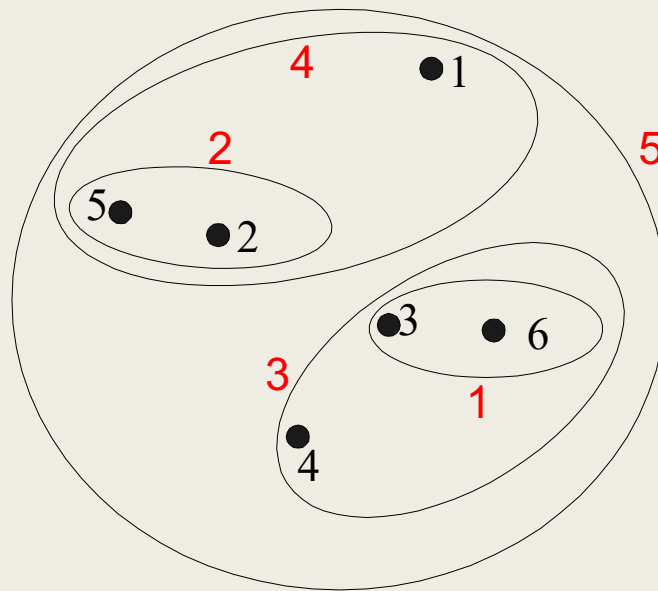
	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0



Nested Clusters

Dendrogram

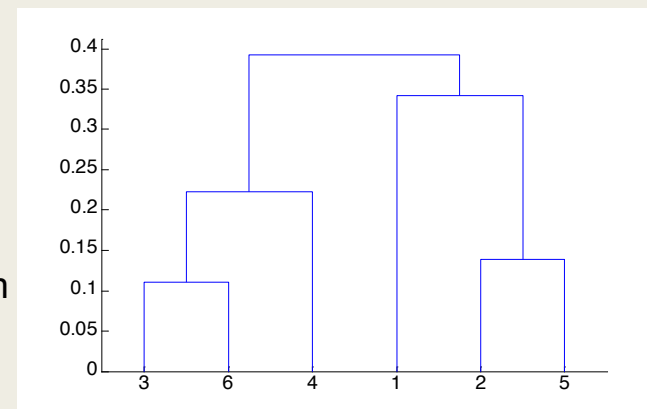
Hierarchical Clustering: MAX



Nested Clusters

Dendrogram

	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



Cluster Similarity: Group Average

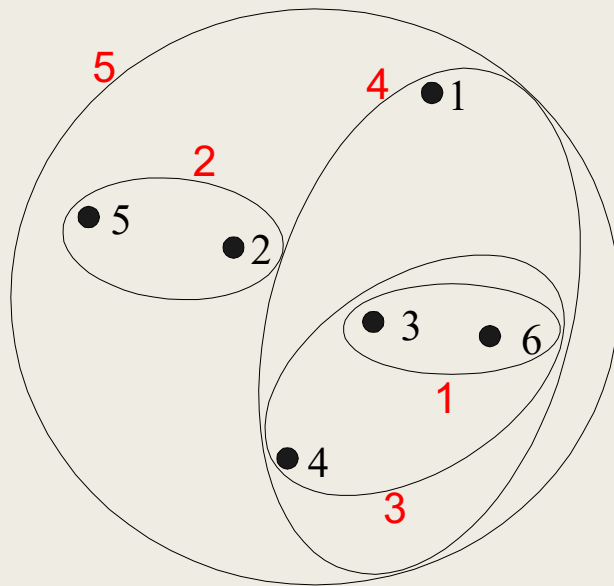
- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

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

- 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

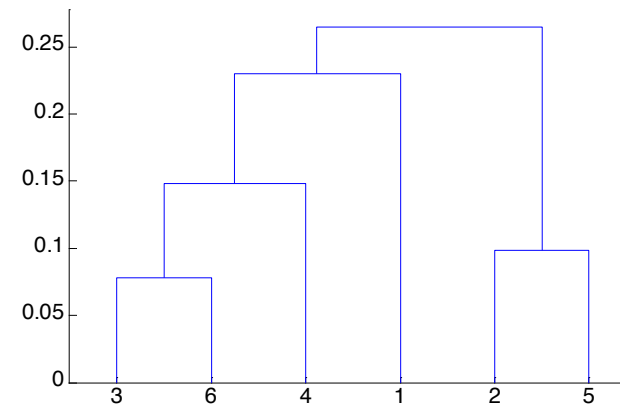
Hierarchical Clustering: Group Average



Nested Clusters

Dendrogram

	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



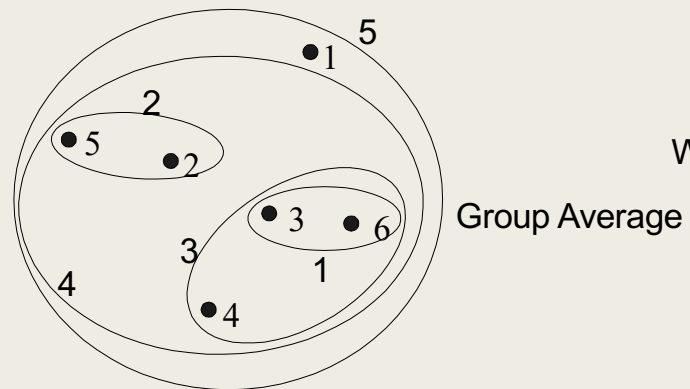
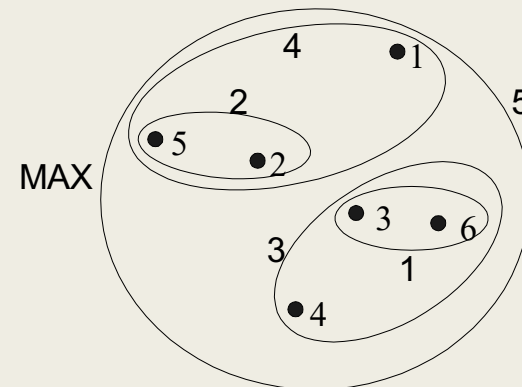
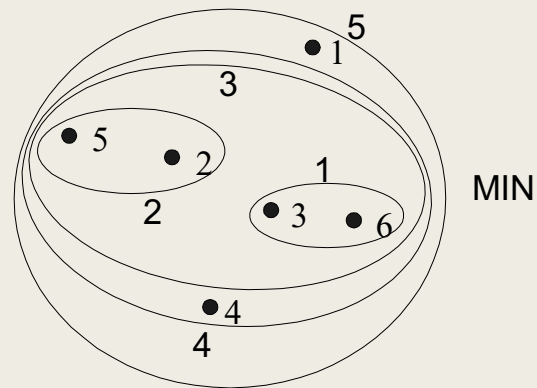
Hierarchical Clustering: Group Average

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

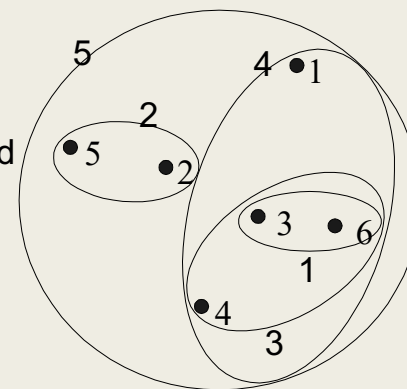
Cluster Similarity: Ward's Method

- 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



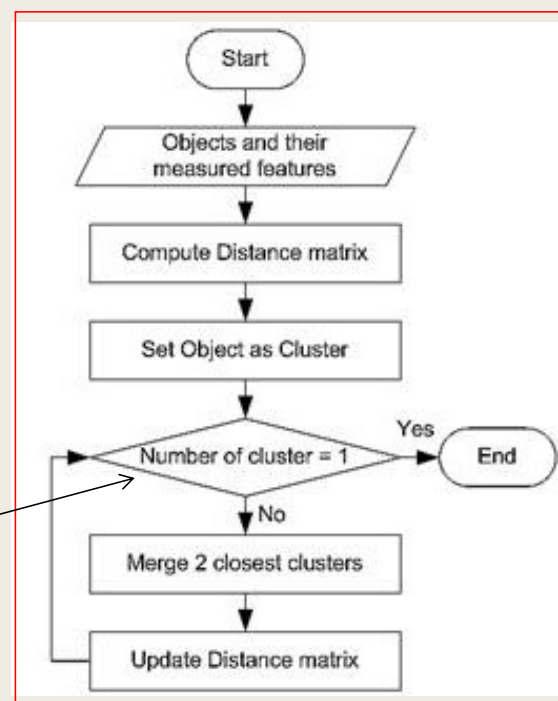
Ward's Method



Agglomerative Algorithm

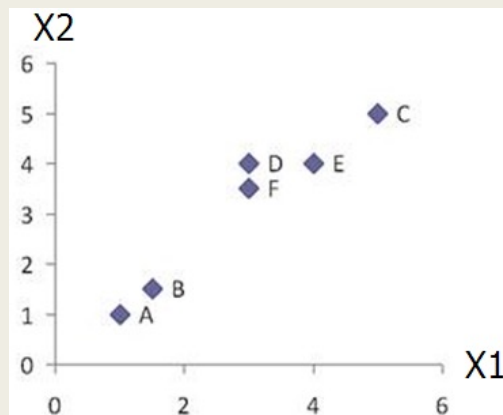
- The *Agglomerative* algorithm is carried out in three steps:

- 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

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

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

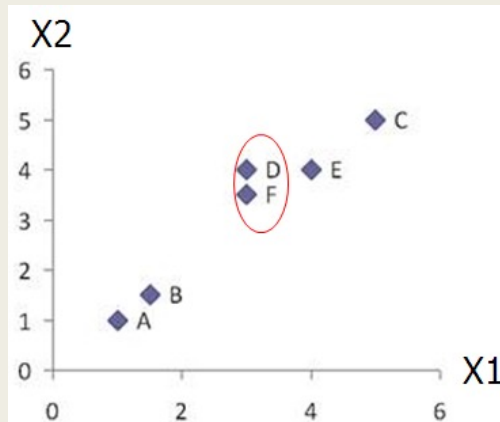
Euclidean distance

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

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

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

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

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

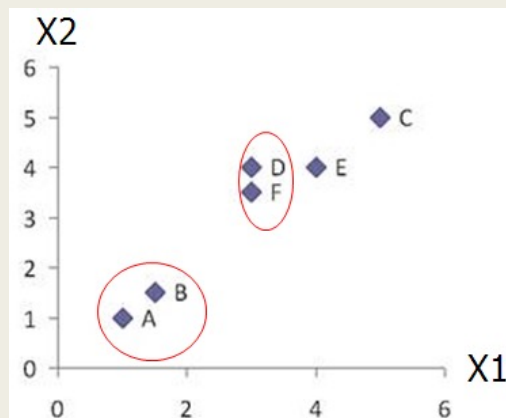
Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

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

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

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

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

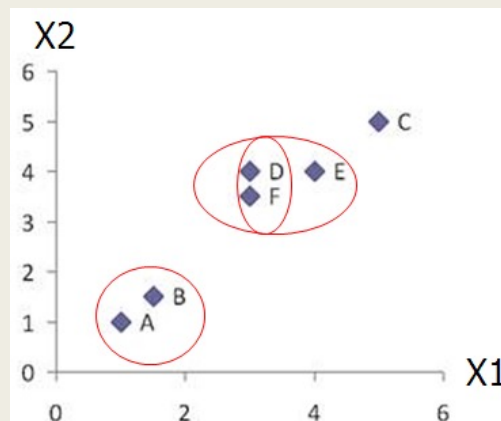
Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	?	0	2.24	1.41
(D, F)	?	2.24	0	1.00
E	?	1.41	1.00	0

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)

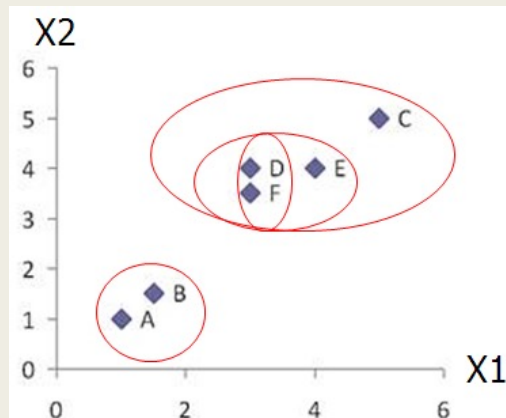
Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	4.95	0	2.24	1.41
(D, F)	2.50	2.24	0	1.00
E	3.54	1.41	1.00	0

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)

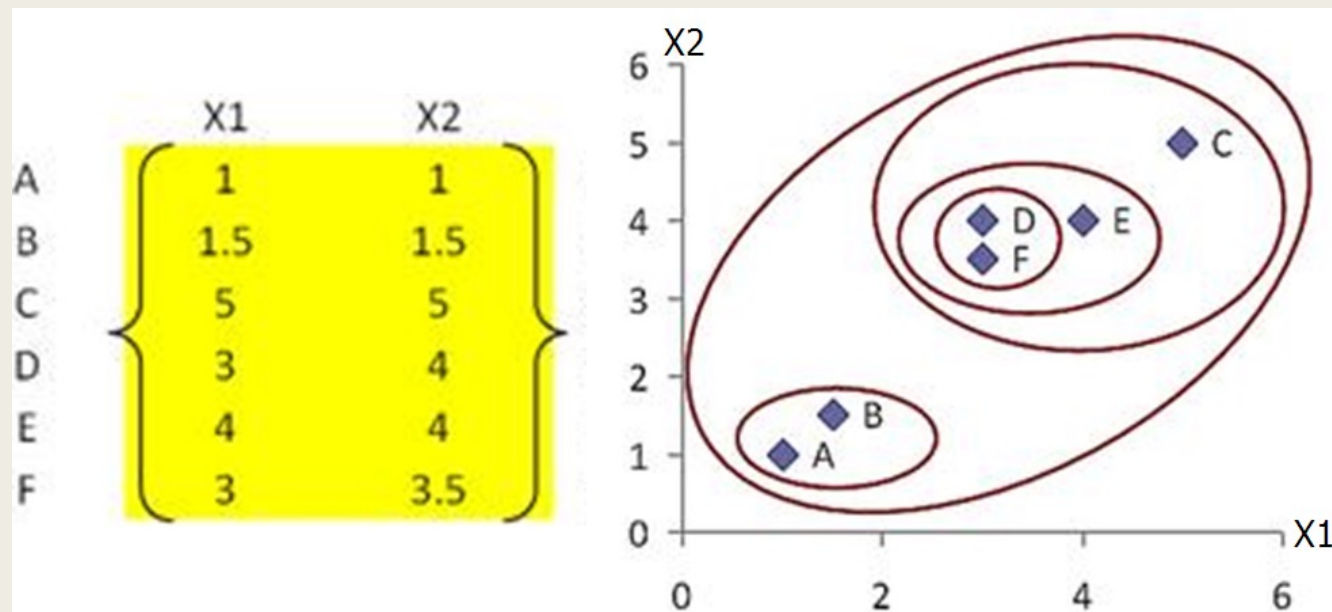
Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	2.50	1.41	0.00

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

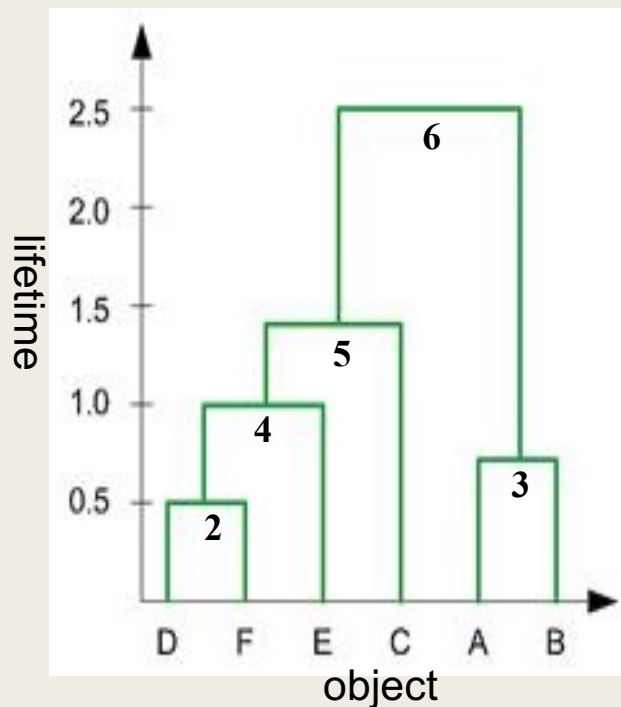
Example

- Result (meeting termination condition)



Example

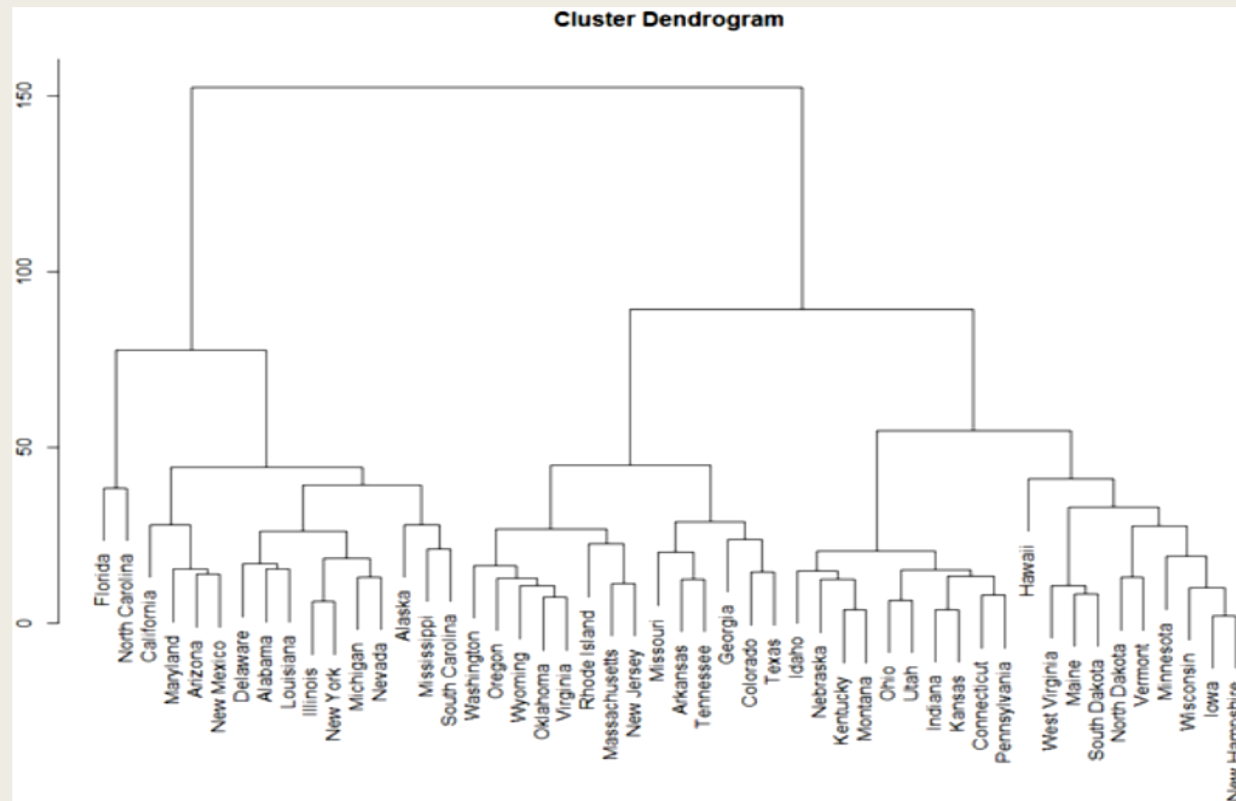
■ Dendrogram tree representation



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*

Choosing k

- Defined by the application, e.g., image quantization
- Plot data (after PCA) and check for clusters
- Incremental (leader-cluster) algorithm: Add one at a time until “elbow” (reconstruction error/log likelihood/intergroup distances)
- Manually check for meaning