Hierarchical Clustering

Ke Chen

Outline

- Introduction
- Cluster Distance Measures
- Agglomerative Algorithm
- Example and Demo
- Relevant Issues
- Summary

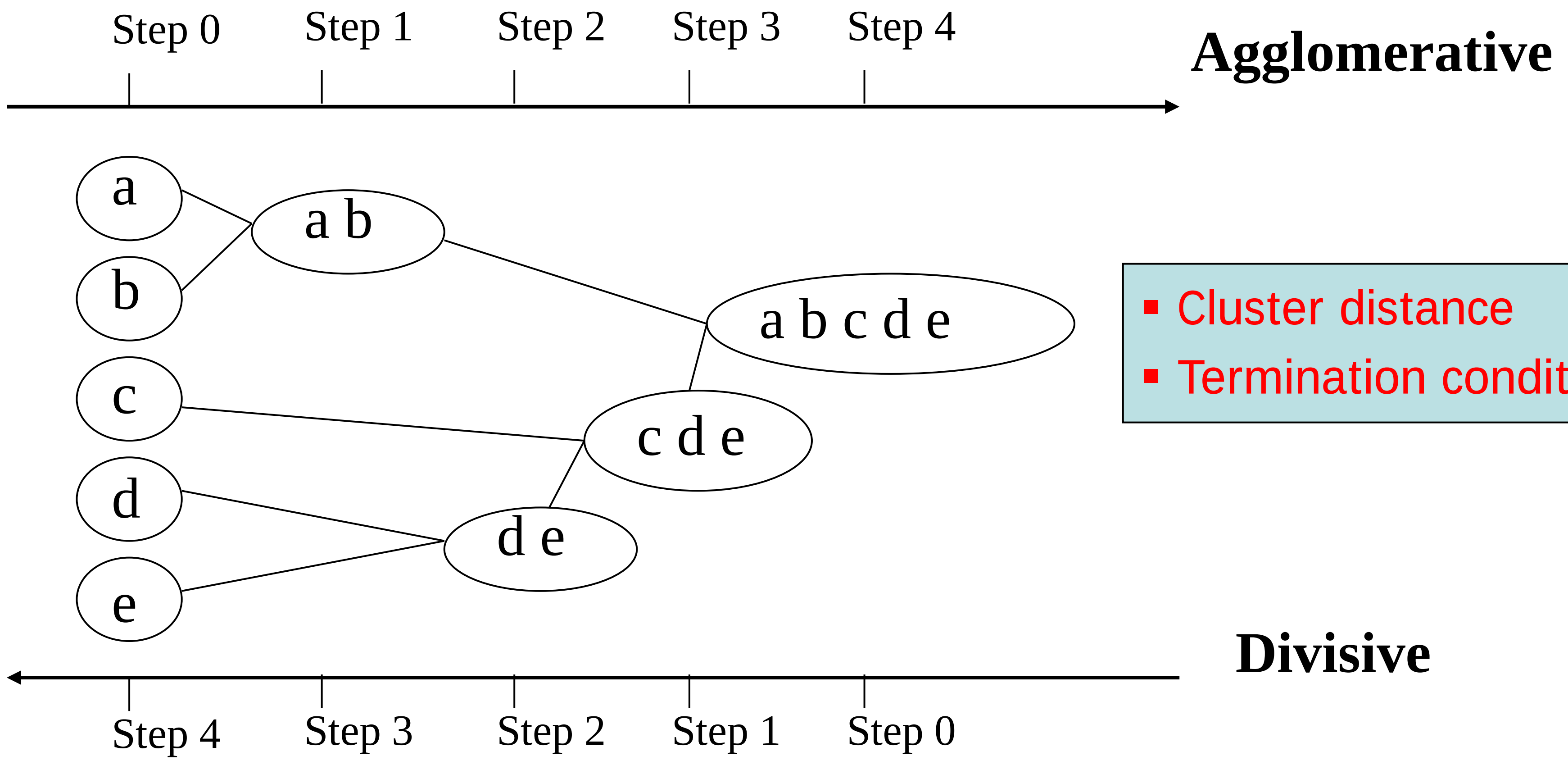
Introduction

- Hierarchical Clustering Approach
  - A typical clustering analysis approach via partitioning data set sequentially
  - Construct nested partitions layer by layer via grouping objects into a tree of clusters (without the need to know the number of clusters in advance)
  - Use (generalised) distance matrix as clustering criteria
- Agglomerative vs. Divisive
  - Two sequential clustering strategies for constructing a tree of clusters
  - Agglomerative: a bottom-up strategy
    - Initially each data object is in its own (atomic) cluster
    - Then merge these atomic clusters into larger and larger clusters
  - Divisive: a top-down strategy
    - Initially all objects are in one single cluster
    - Then the cluster is subdivided into smaller and smaller clusters

Introduction

- Illustrative Example

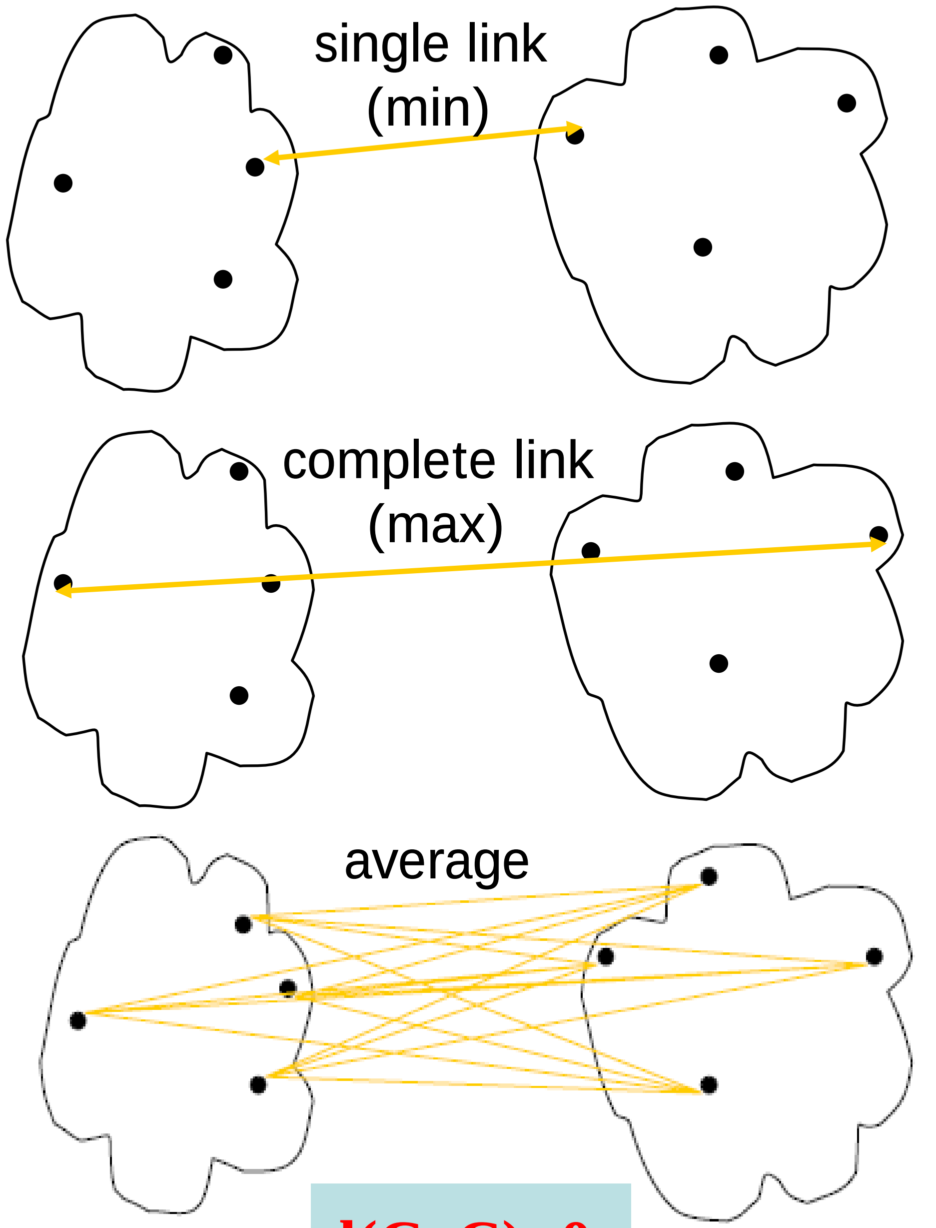
Agglomerative and divisive clustering on the data set {a, b, c, d ,e }



- Cluster distance
- Termination condition

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



d(C, C)=0

Cluster Distance Measures

Example: Given a data set of five objects characterised by a single continuous feature, assume that there are two clusters: 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.

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

2. Calculate three cluster distances between C_1 and C_2.

Single link

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

Complete link

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

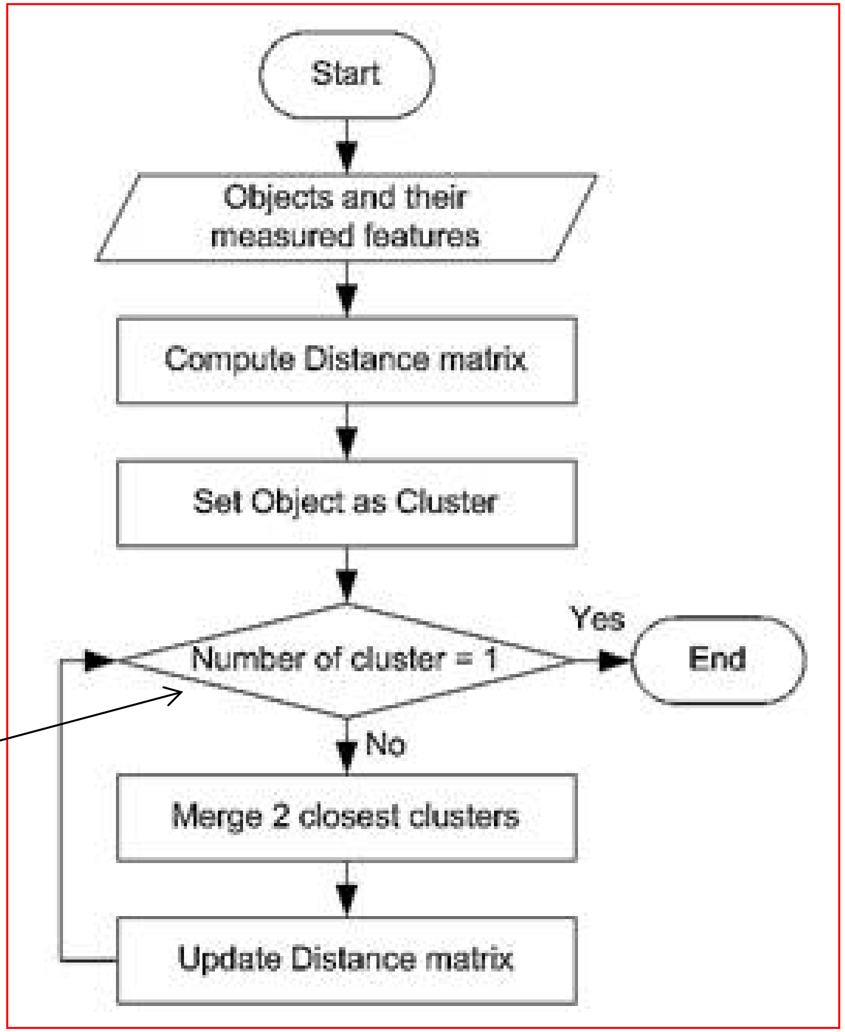
Average

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

Agglomerative Algorithm

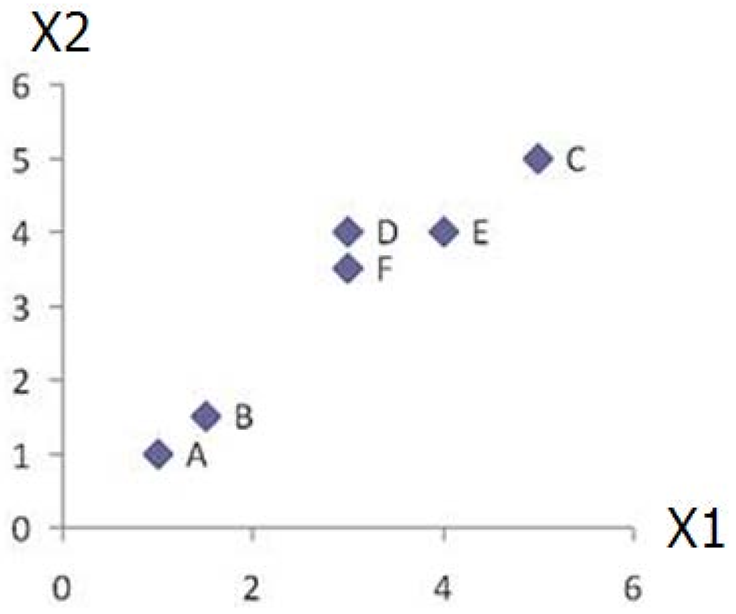
- The *Agglomerative* algorithm is carried out in three steps:

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

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

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

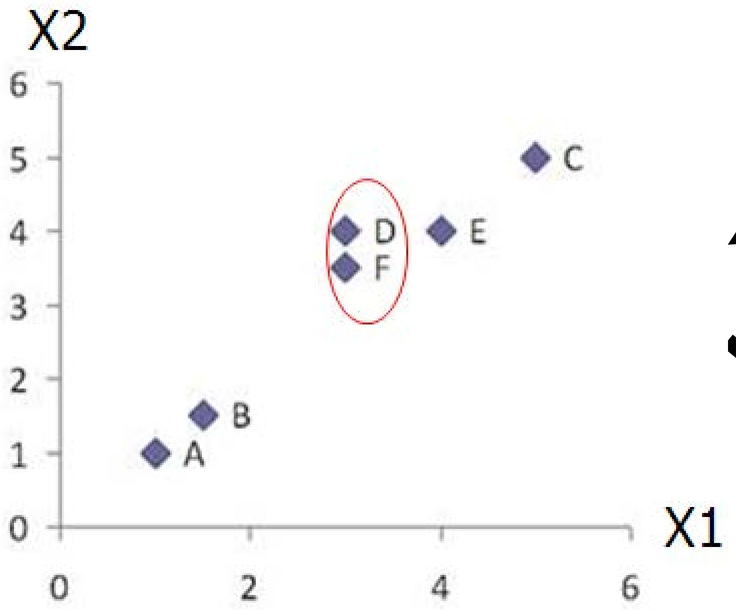
Euclidean distance

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

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

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

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

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

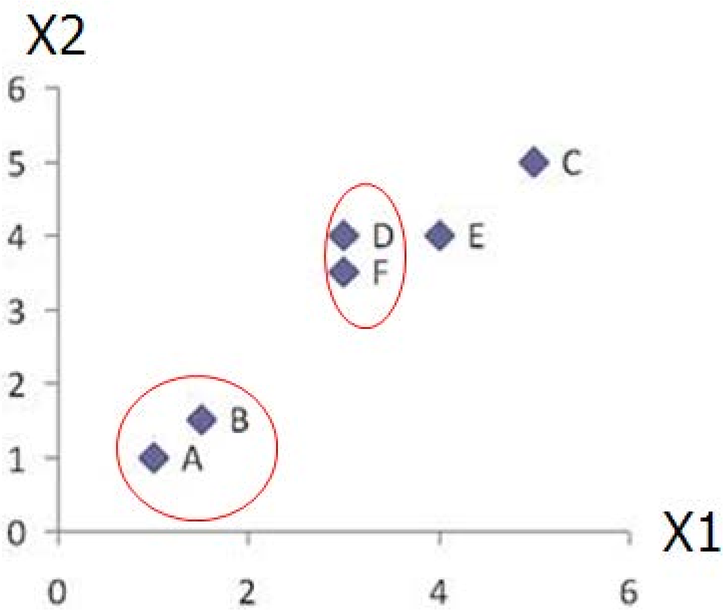
Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

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

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

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

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

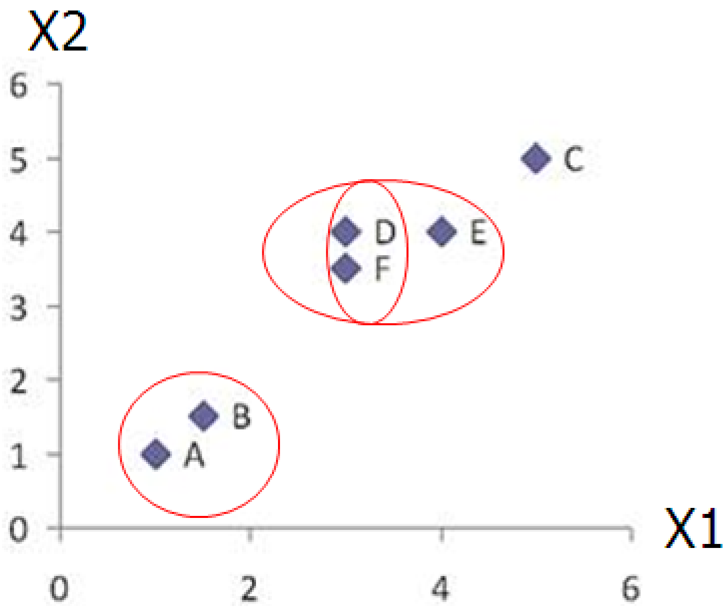
Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	**?**	0	2.24	1.41
(D, F)	**?**	2.24	0	1.00
E	**?**	1.41	1.00	0

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)

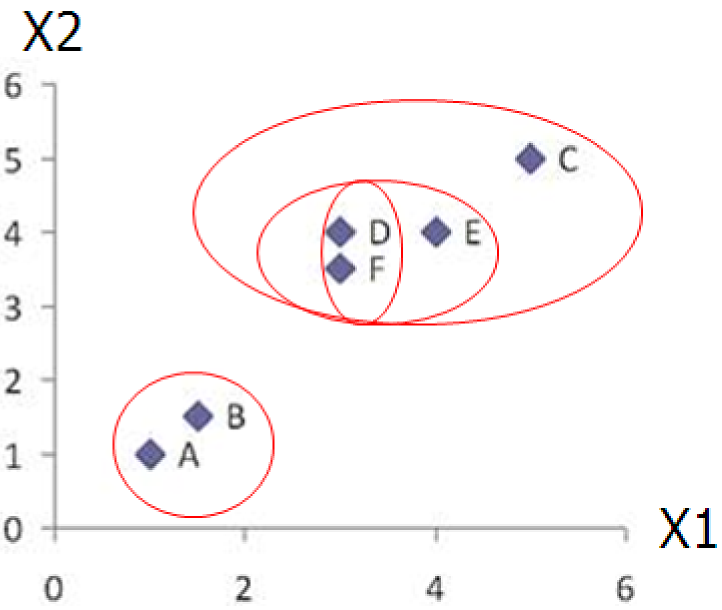
Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

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

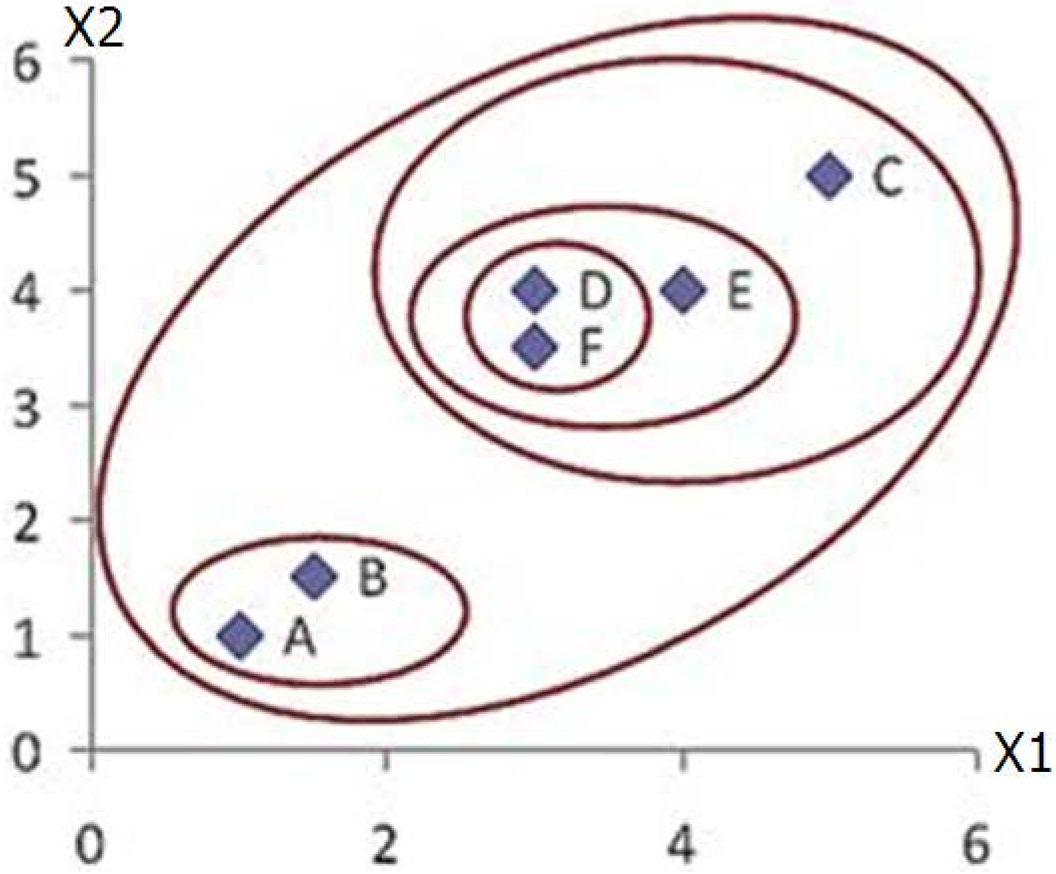
Min Distance (Single Linkage)

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

Example

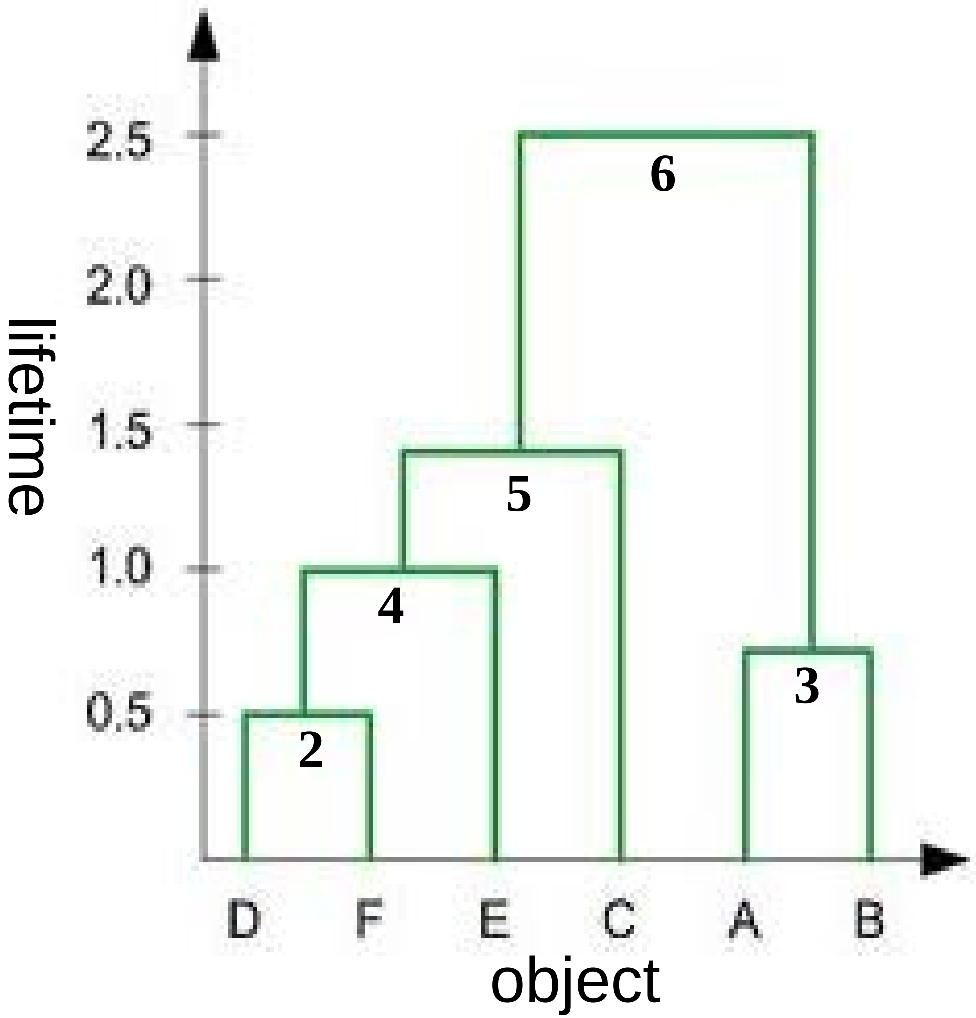
- Final result (meeting termination condition)

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



Example

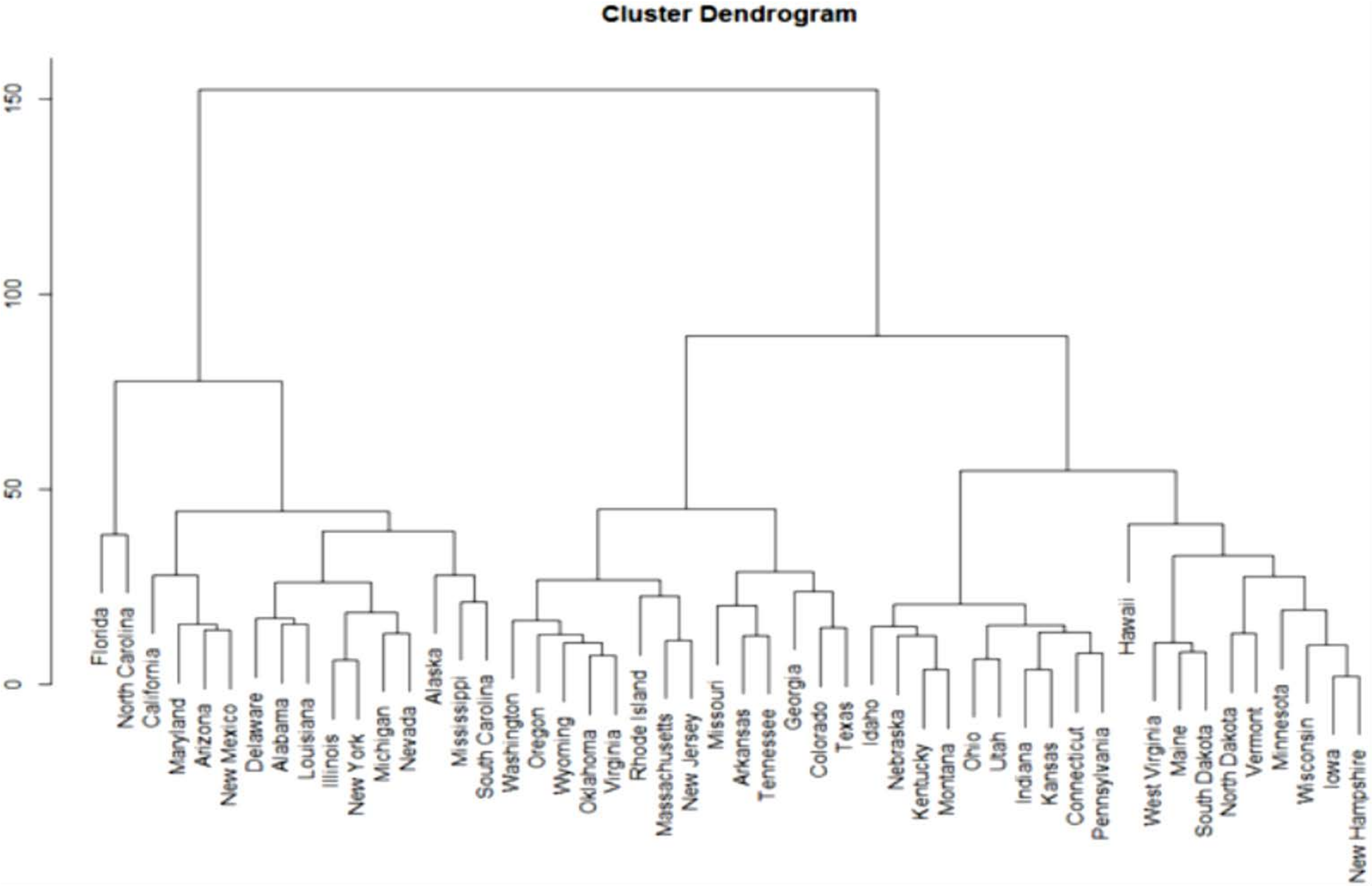
- Dendrogram tree representation



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



Exercise

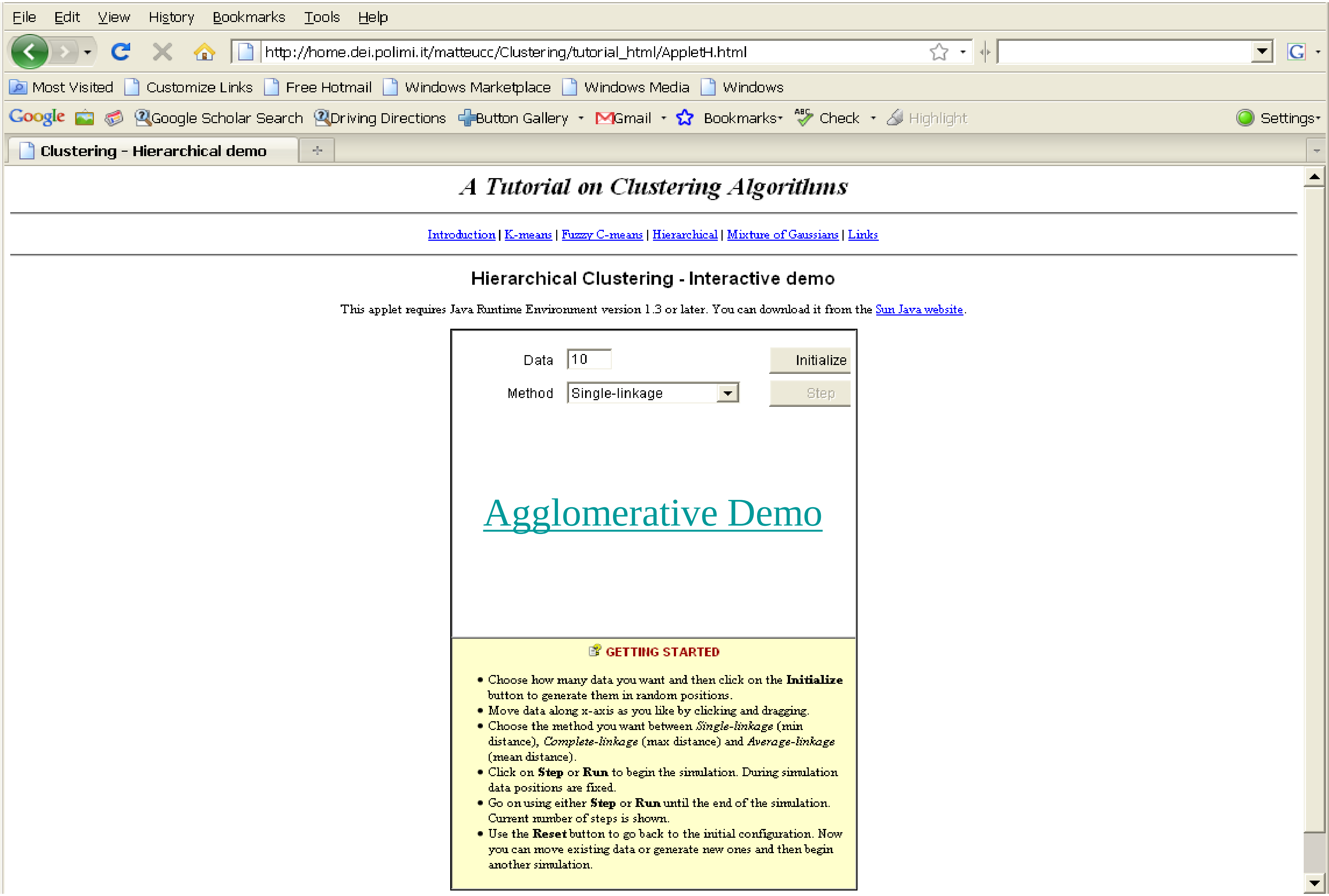
Given a data set of five objects characterised by a single continuous feature:

	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.

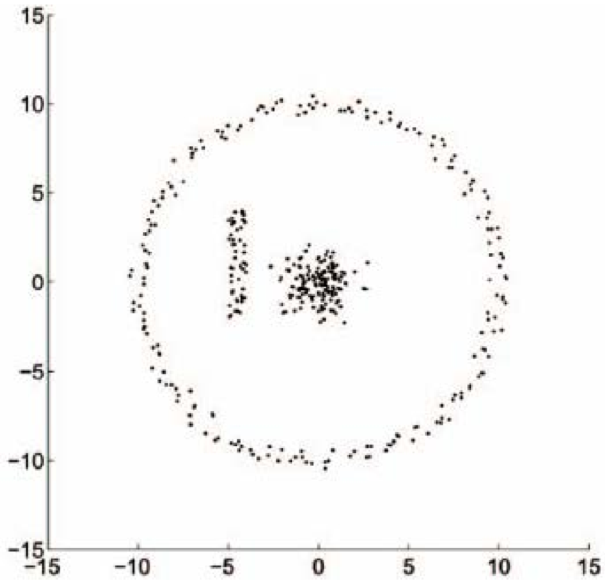
$$
\begin{array}{c|ccccc}
 & \mathbf{a} & \mathbf{b} & \mathbf{c} & \mathbf{d} & \mathbf{e} \\
\hline
\mathbf{a} & 0 & 1 & 3 & 4 & 5 \\
\mathbf{b} & 1 & 0 & 2 & 3 & 4 \\
\mathbf{c} & 3 & 2 & 0 & 1 & 2 \\
\mathbf{d} & 4 & 3 & 1 & 0 & 1 \\
\mathbf{e} & 5 & 4 & 2 & 1 & 0
\end{array}
$$

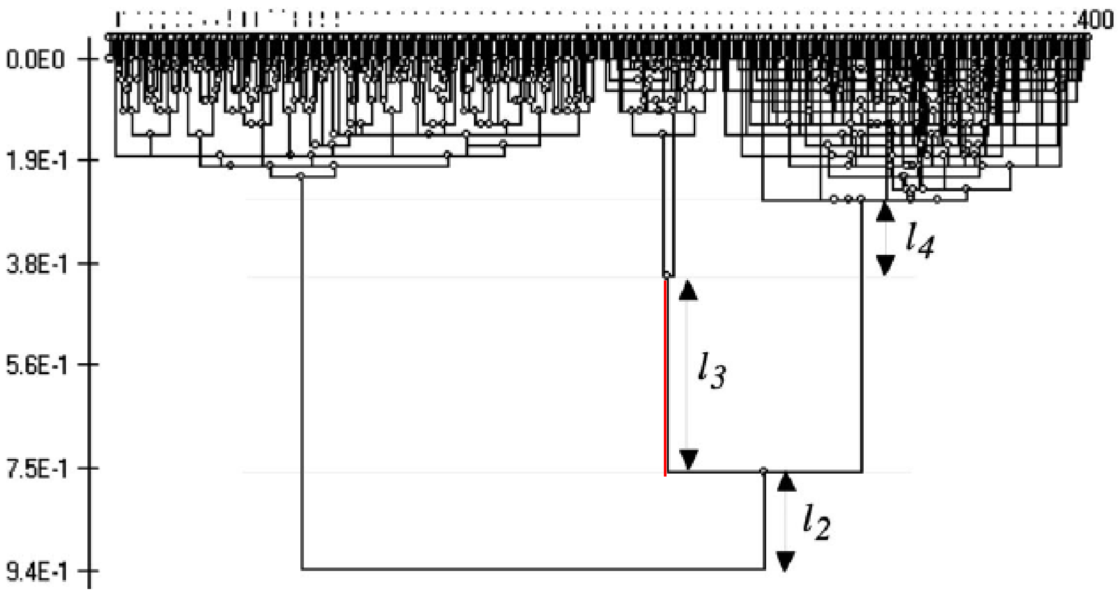
Demo


File Edit View History Bookmarks Tools Help
http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletH.html
Most Visited Customize Links Free Hotmail Windows Marketplace Windows Media Windows
Google Google Scholar Search Driving Directions Button Gallery Gmail Bookmarks Check Highlight Settings
Clustering - Hierarchical demo
A Tutorial on Clustering Algorithms
Introduction | K-means | Fuzzy C-means | Hierarchical | Mixture of Gaussians | Links
Hierarchical Clustering - Interactive demo
This applet requires Java Runtime Environment version 1.3 or later. You can download it from the Sun Java website.
Data 10
Initialize
Method Single-linkage
Step
Agglomerative Demo
GETTING STARTED
• Choose how many data you want and then click on the Initialize button to generate them in random positions.
• Move data along x-axis as you like by clicking and dragging.
• Choose the method you want between Single-linkage (min distance), Complete-linkage (max distance) and Average-linkage (mean distance).
• Click on Step or Run to begin the simulation. During simulation data positions are fixed.
• Go on using either Step or Run until the end of the simulation. Current number of steps is shown.
• Use the Reset button to go back to the initial configuration. Now you can move existing data or generate new ones and then begin another simulation.

Relevant Issues

- How to determine the number of clusters
  - If the number of clusters known, termination condition is given!
  - The *K*-cluster lifetime as the range of threshold value on the dendrogram tree that leads to the identification of *K* clusters
  - Heuristic rule: cut a dendrogram tree with maximum *K*-cluster life time





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

Online tutorial: the hierarchical clustering functions in Matlab

https://www.youtube.com/watch?v=aYzjenNNOcc