CS4641B Machine Learning

Lecture 09: Hierarchical clustering

Rodrigo Borela ▸ rborelav@gatech.edu

These slides are based on slides from Aarti Singh, Eric Xing, Andrew Moore, Jiawei Han and Mahdi Roozbahani

Outline

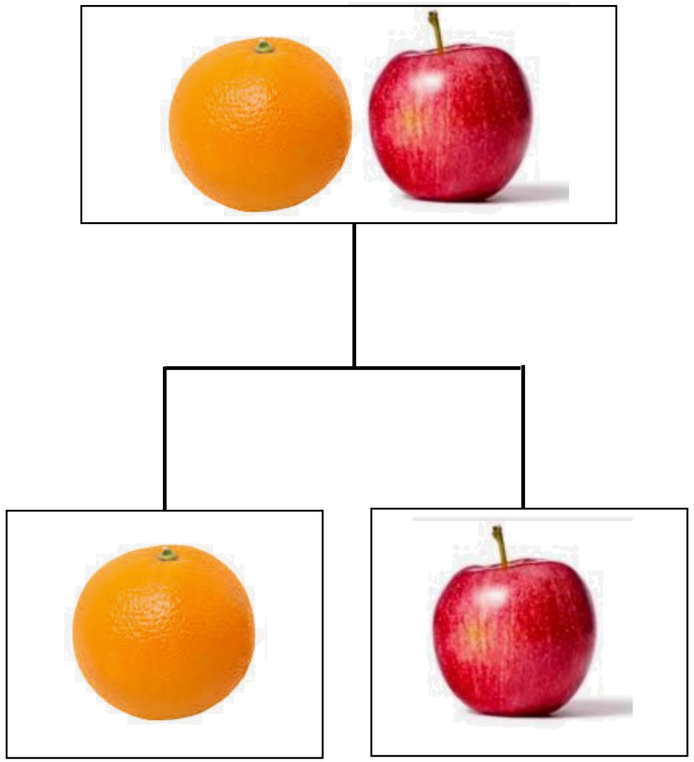
- Overview
- Bottom-Up vs Top-Down Clustering
- Measuring Distance between Clusters

Outline

- Overview
- Bottom-Up vs Top-Down Clustering
- Measuring Distance between Clusters

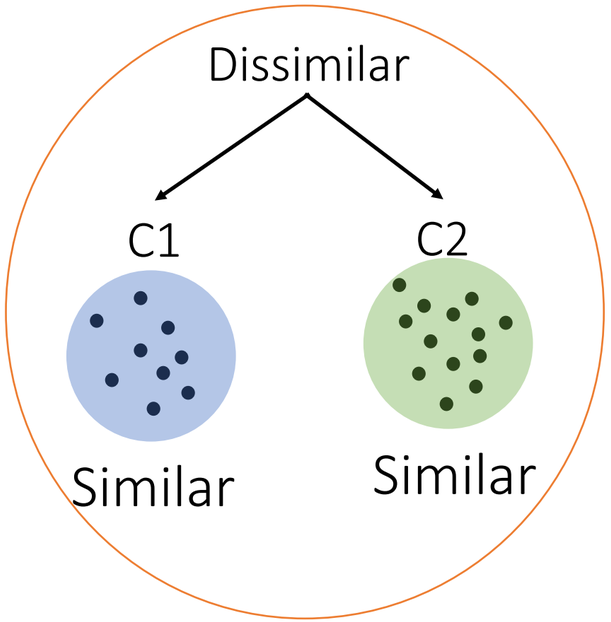
Hierarchical Clustering vs Partitional Clustering

Agglomerative Divisive



Tree structure (parent-child relationship)

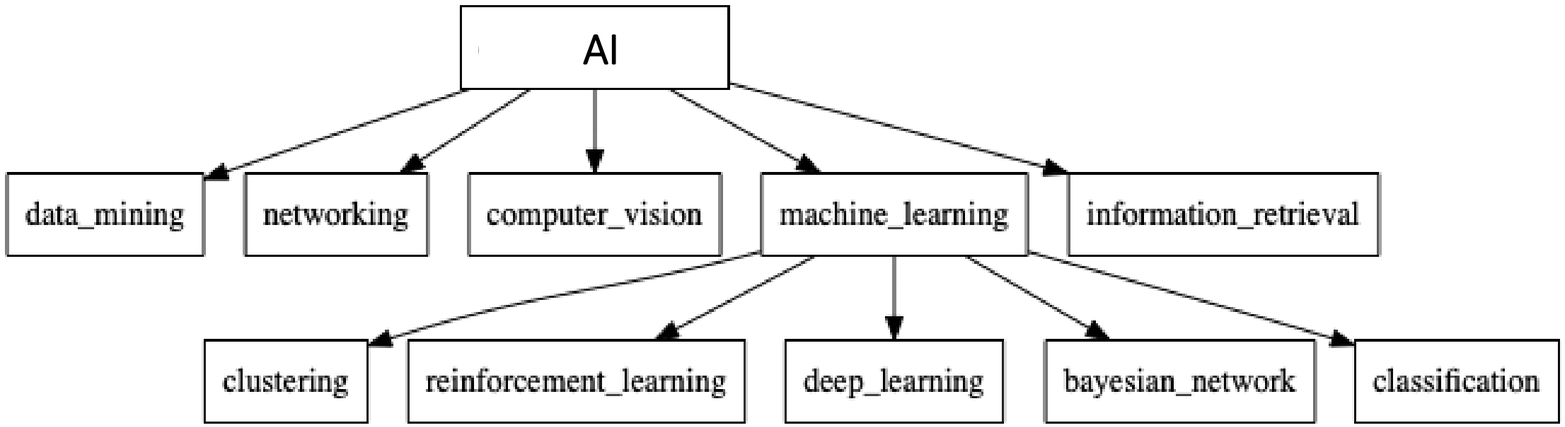
K-Means, DBSCAN



Hierarchical Clustering

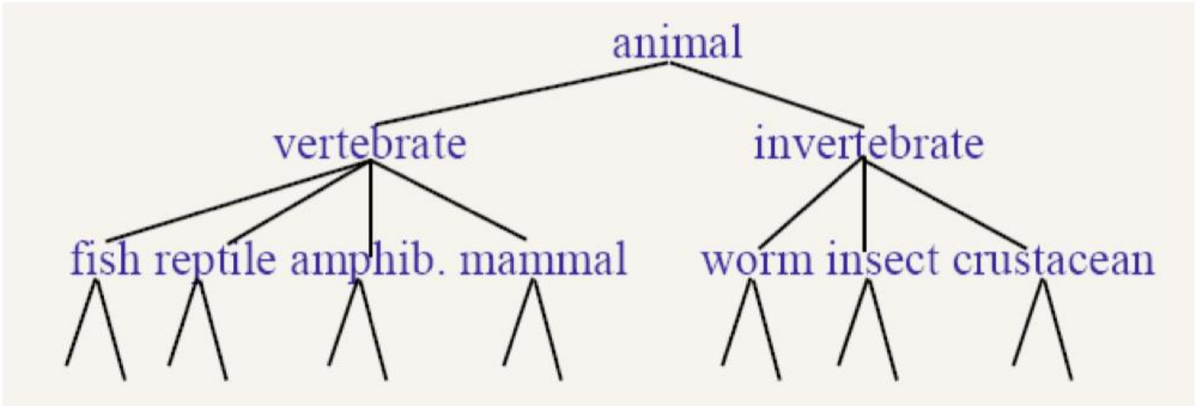
- How to organize a set of CS papers into a hierarchy?

?



Hierarchical Clustering

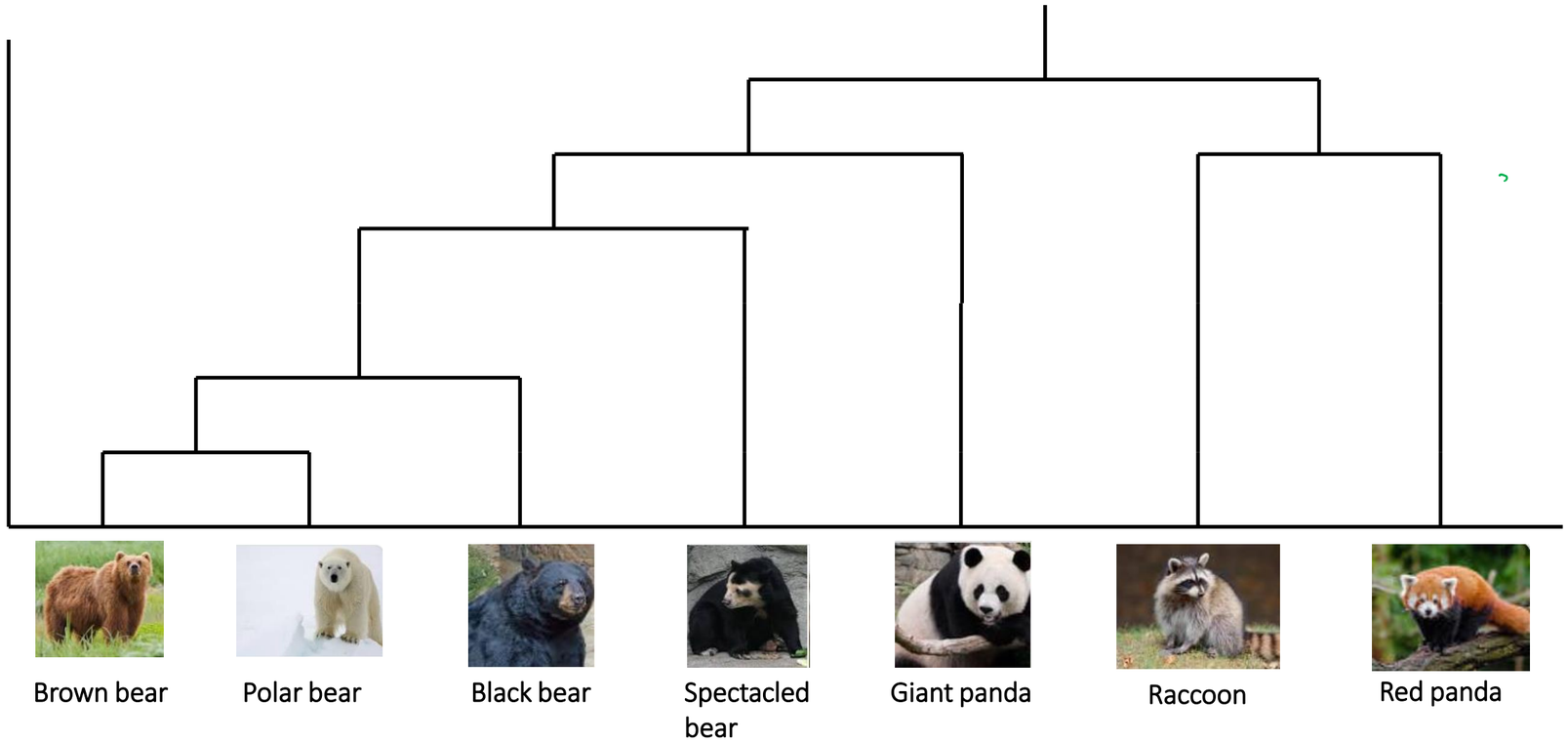
- Organize objects into a tree-based hierarchical taxonomy (dendrogram)



- Many applications in the real world
  - Web pages
  - News articles
  - Scientific papers

Example

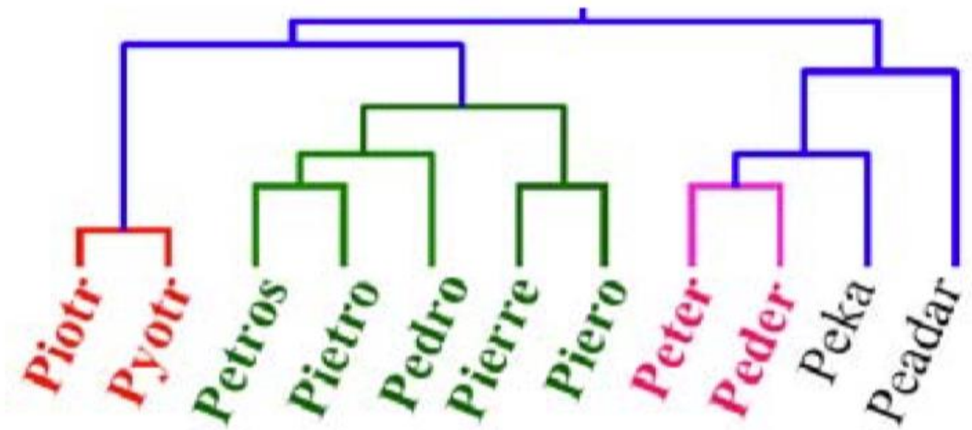
- DNA sequencing and hierarchical clustering to find the phylogenetic tree of animal evolution

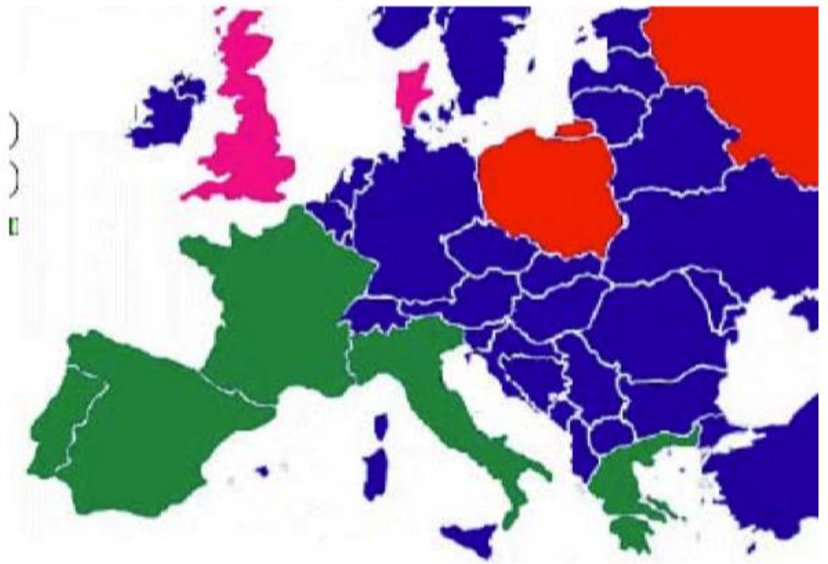


Using Hierarchical clustering, the researchers were able to place the giant pandas closer to bears

Hierarchical clustering

- Organizing data at multiple granularities
- Cutting the dendrogram at a desired level leads to a sub-cluster: each connected component forms a cluster





Outline

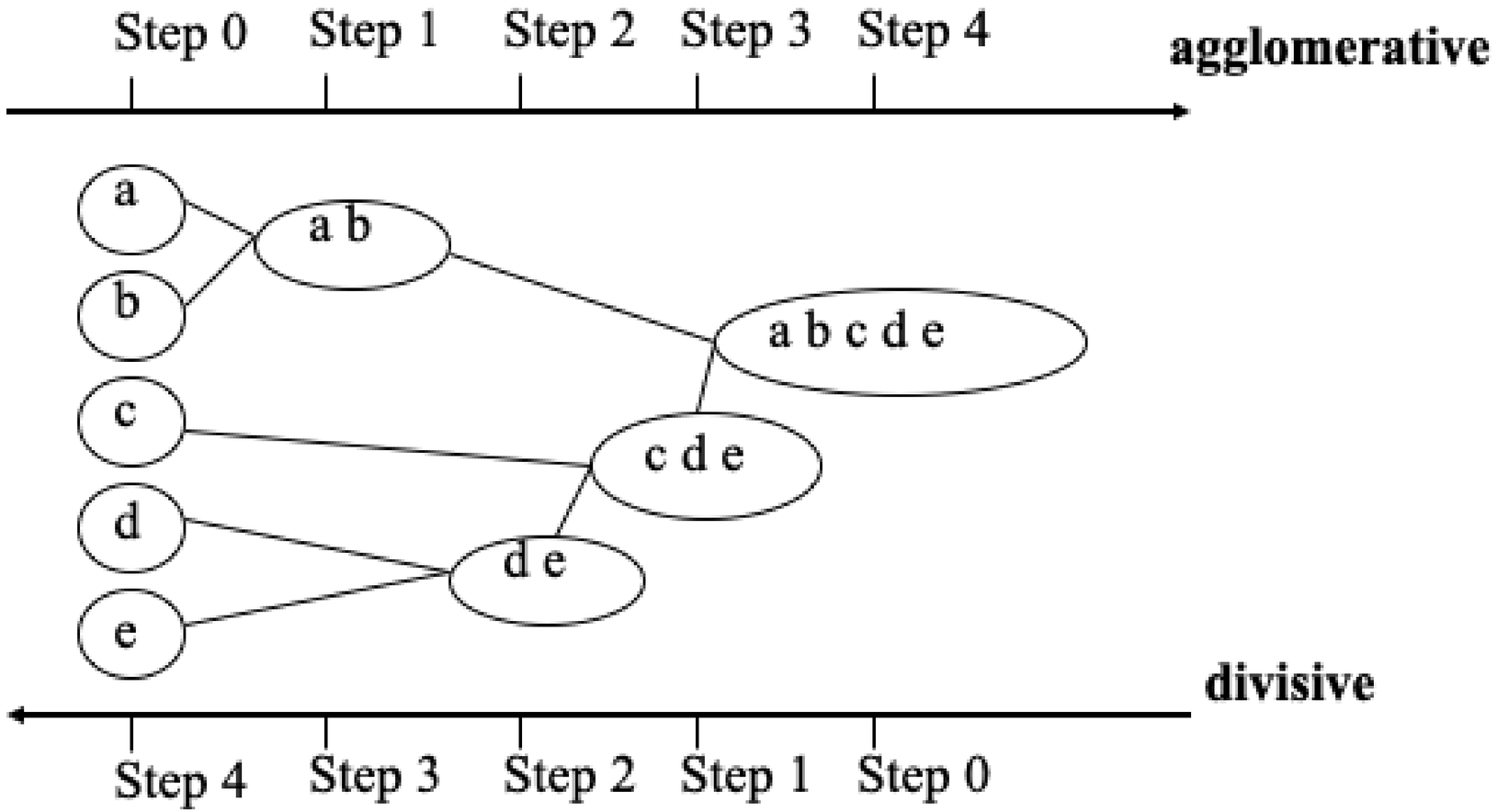
- Overview
- **Bottom-Up vs Top-Down Clustering**
- Measuring Distance between Clusters

Two Paradigms for hierarchical clustering

- **Bottom-up agglomerative clustering**
  - Start by considering each object as a separate cluster
  - Repeatedly join the closest pair of clusters
  - Stop when there is only one cluster left

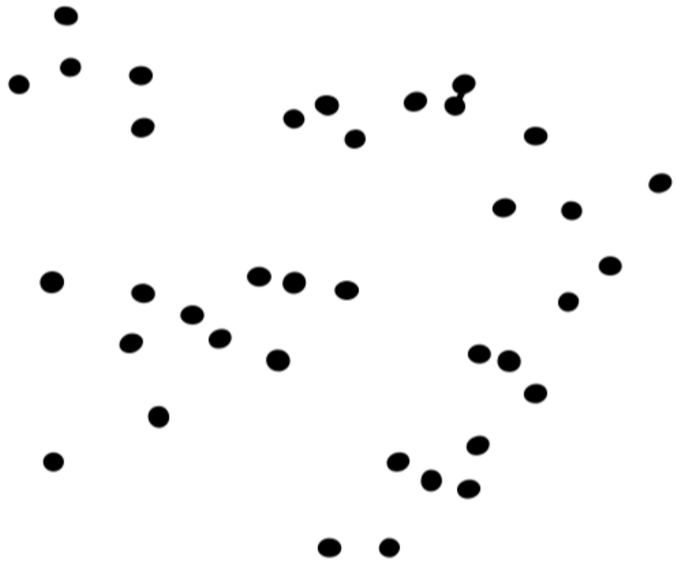
- **Top-down divisive clustering**
  - Start by considering all objects as one large cluster
  - Recursively divide each cluster into two sub-clusters
  - Stop when each cluster contains only one object

Bottom-up vs Top-down

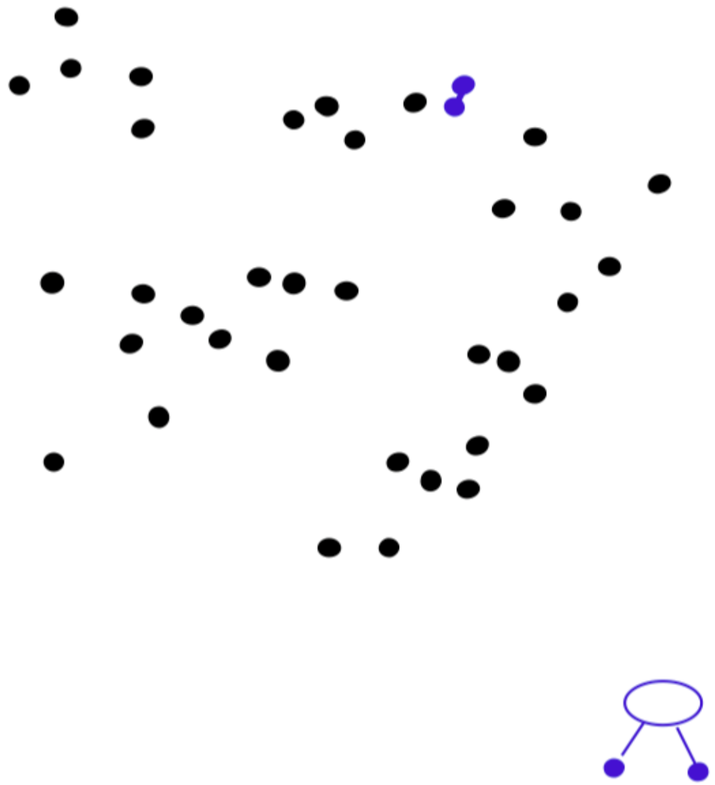


Bottom-Up agglomerative clustering

1. Say “every point is its own cluster”

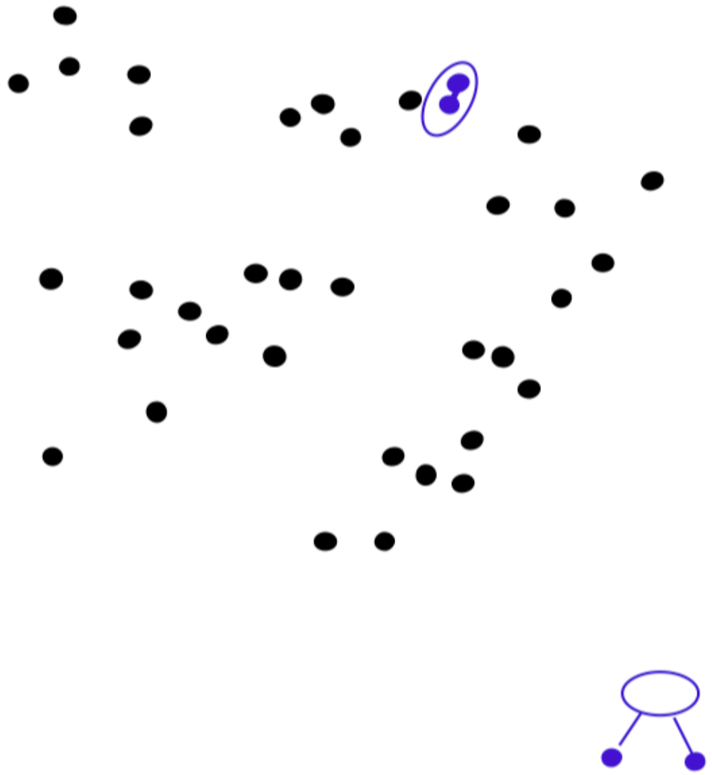


Bottom-Up agglomerative clustering



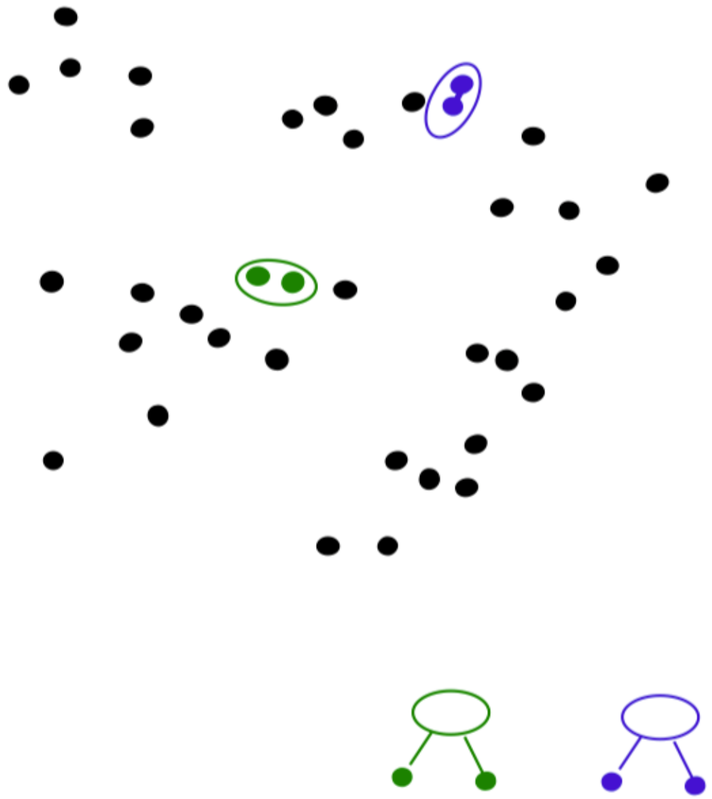
1. Say “every point is its own cluster”
2. Find “most similar” pair of clusters

Bottom-Up agglomerative clustering



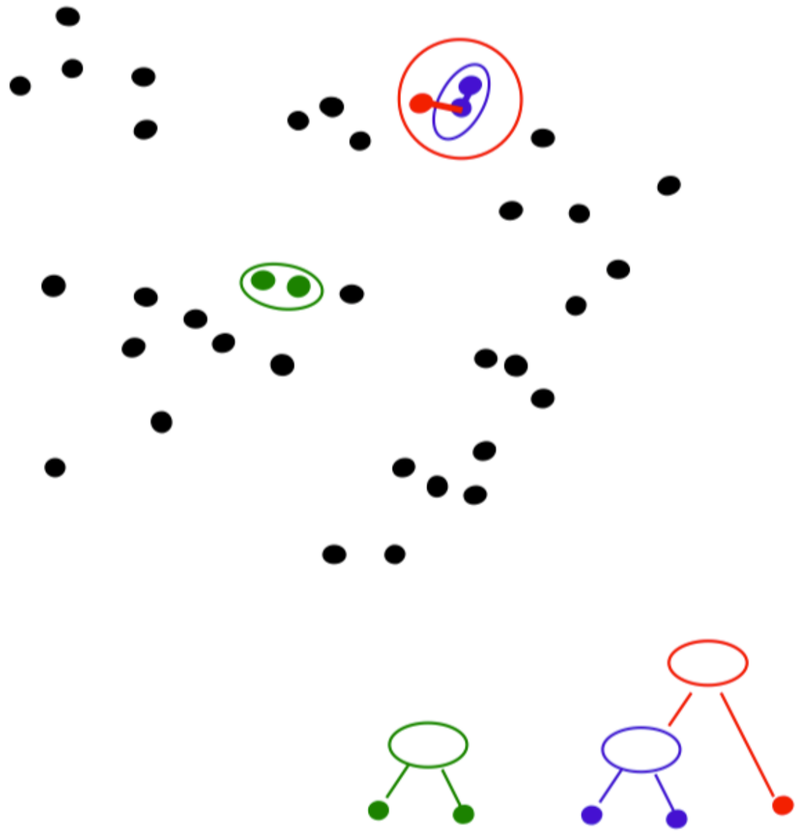
1. Say “every point is its own cluster”
2. Find “most similar” pair of clusters
3. Merge it into a parent cluster

Bottom-Up agglomerative clustering



1. Say “every point is its own cluster”
2. Find “most similar” pair of clusters
3. Merge it into a parent cluster
4. Repeat

Bottom-Up agglomerative clustering

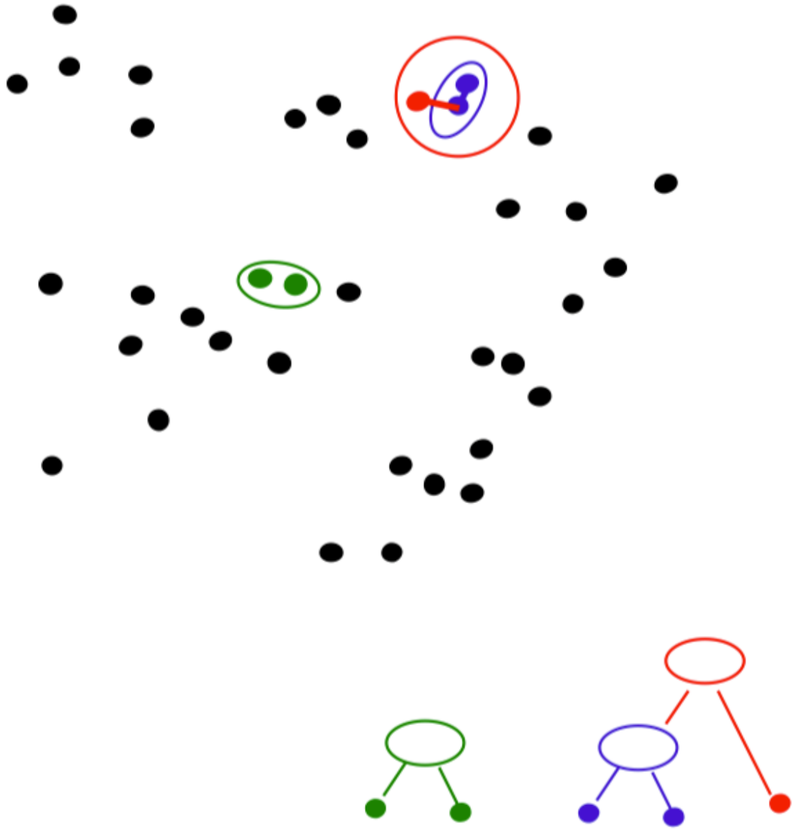


1. Say “every point is its own cluster”
2. Find “most similar” pair of clusters
3. Merge it into a parent cluster
4. Repeat

Outline

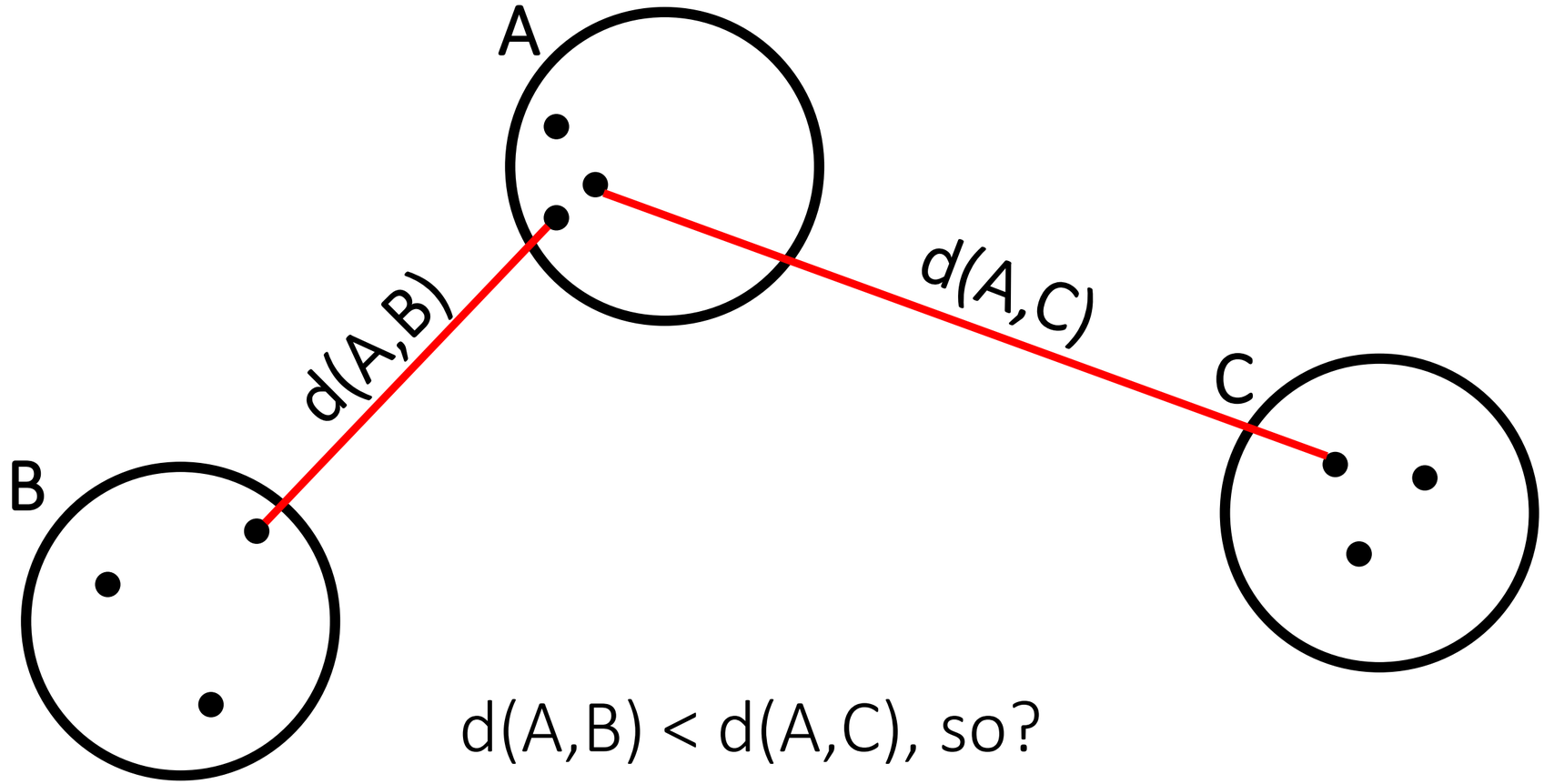
- Overview
- Bottom-Up vs Top-Down Clustering
- **Measuring Distance between Clusters**

Key question: similarity function



How to define “similarity” between two clusters?

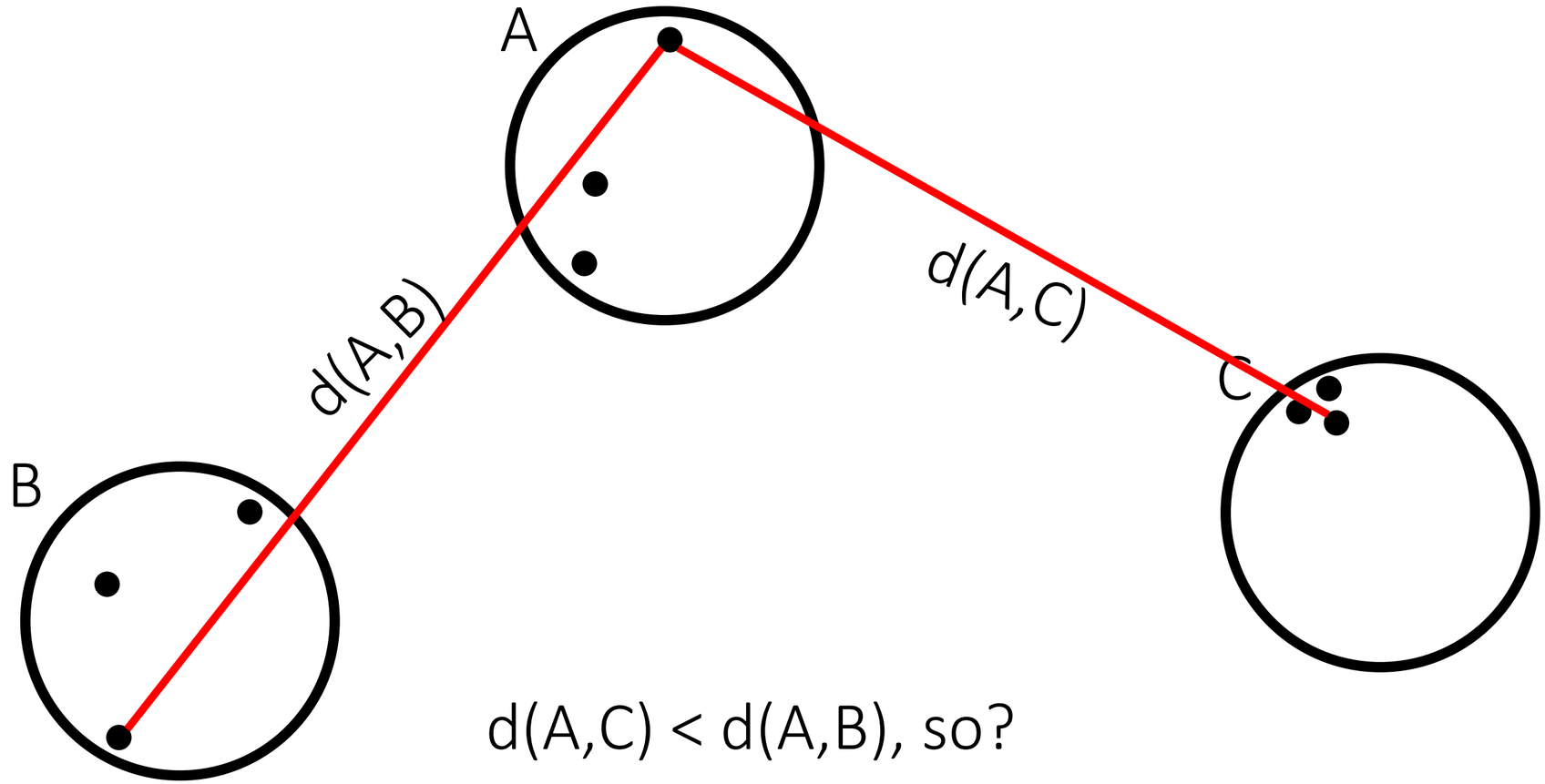
I am going to merge A with either B or C. Which one?



d(A,B) < d(A,C), so?

Single Link

I am going to merge A with either B or C. Which one?

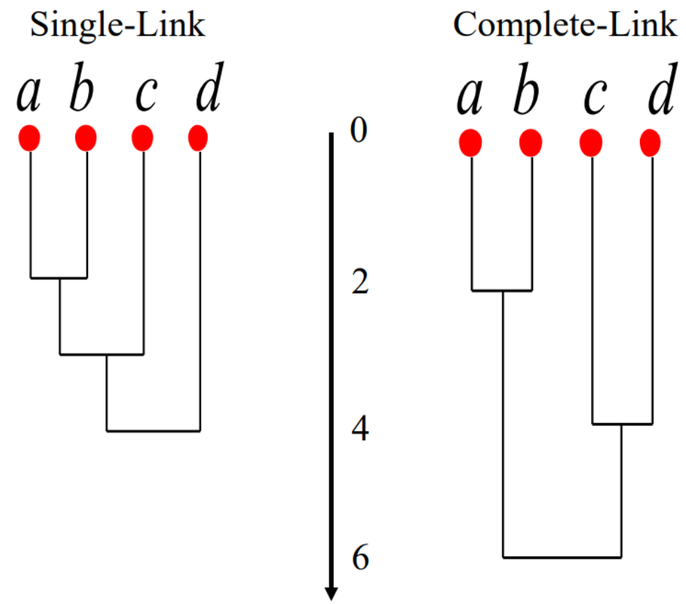


d(A,C) < d(A,B), so?

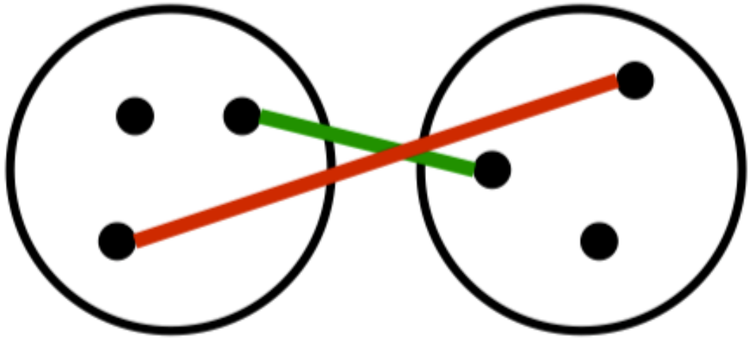
Complete Link

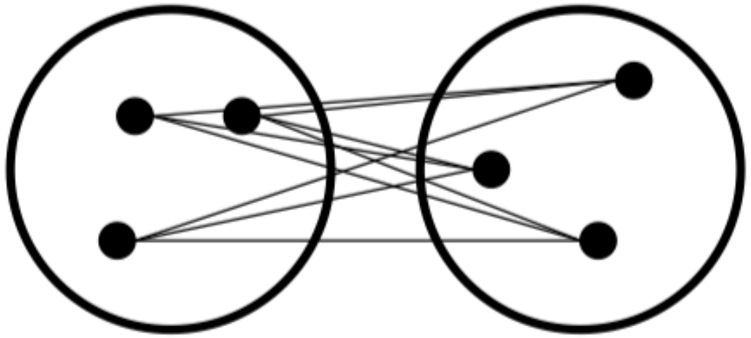
Key question: similarity function

- **Single link:** A chain of points can be extended for long distances without regard to the overall shape of the emerging cluster. This effect is called *chaining*. It is also sensitive to outliers. It is faster in general.
- **Complete link:** Clusters are split into two groups of roughly equal size when we cut the dendrogram at the last merge. In general, this is a more useful organization of the data than a clustering with chains. It avoids chaining and more robust to outliers. Generally slower.
- **Average link:** When you don't know which one may be better for you, start it with the average link method.



How to define distance between two clusters?

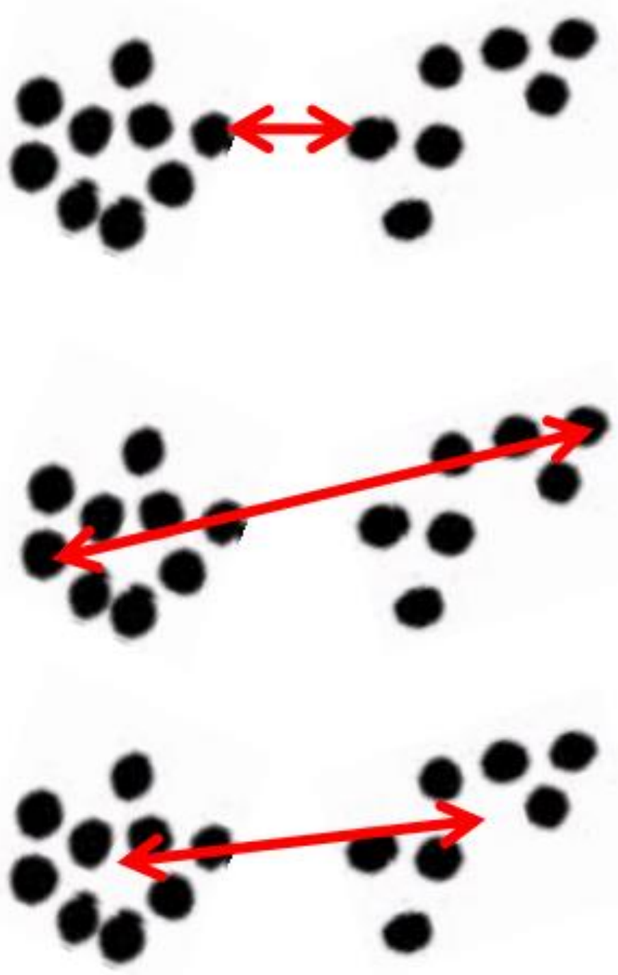




Bottom-up agglomerative clustering

Different algorithms differ in how the similarities are defined (and hence updated) between two clusters:

- **Single-link**
  - **Nearest neighbor:** similarity between their closest members
- **Complete-link**
  - **Furthest neighbor:** similarity between their furthest members
- **Centroid:**
  - Similarity between the centers of gravity
- **Average-link**
  - Average similarity of all cross-cluster pairs.

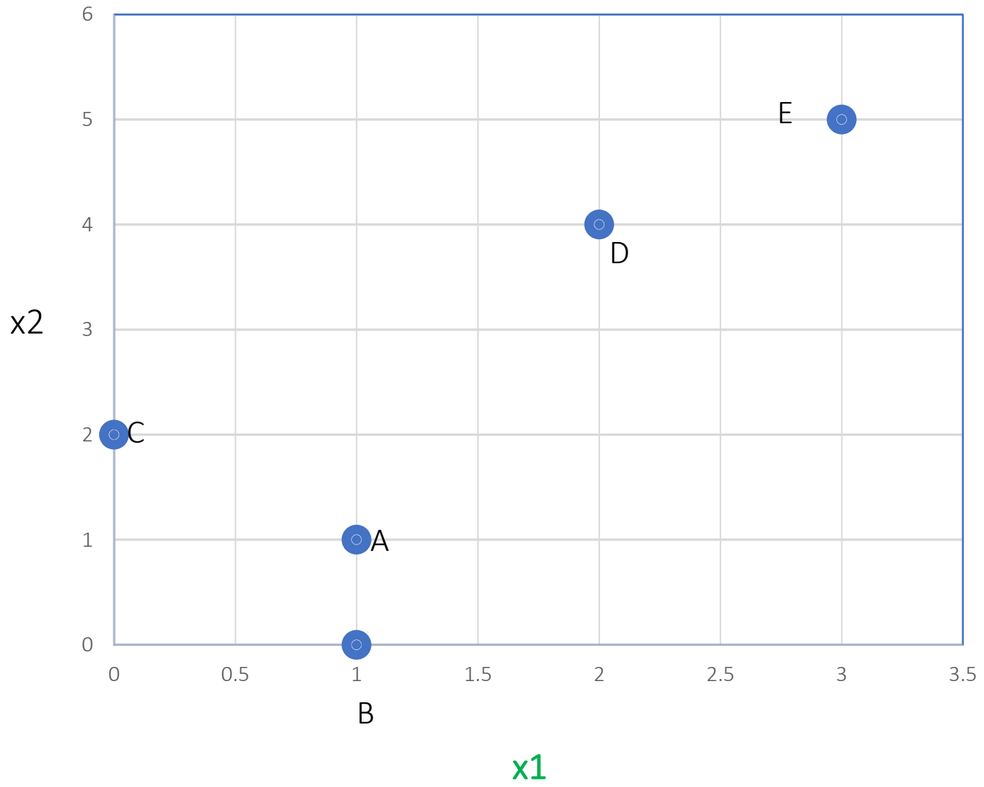


Different distance functions can lead to different results!

Example:

Distance based on centroid

i	x_1	x_2
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5



EUCLIDEAN DISTANCE

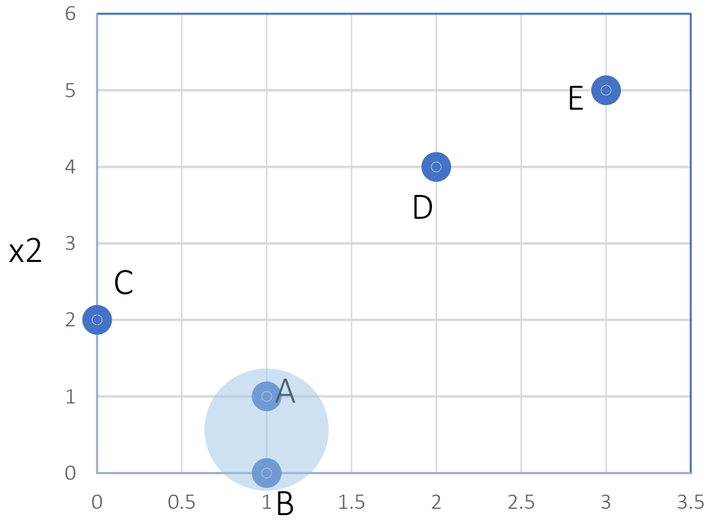
	A	B	C	D	E
A	0	1	1.4	3.2	4.5
B	1	0	2.2	4.1	5.4
C	1.4	2.2	0	2.8	4.2
D	3.2	4.1	2.8	0	1.4
E	4.5	5.4	4.2	1.4	0

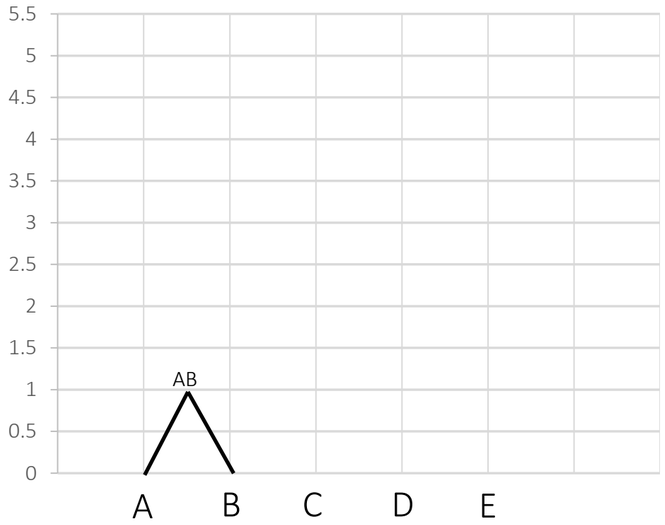
Distance based on centroid (bottom-up clustering)

	A	B	C	D	E
A	0	1	1.4	3.2	4.5
B	1	0	2.2	4.1	5.4
C	1.4	2.2	0	2.8	4.2
D	3.2	4.1	2.8	0	1.4
E	4.5	5.4	4.2	1.4	0

EUCLIDEAN DISTANCE

	(A,B)	C	D	E
(A,B)	0	1.8	3.6	4.9
C	1.8	0	2.8	4.2
D	3.6	2.8	0	1.4
E	4.9	4.2	1.4	0





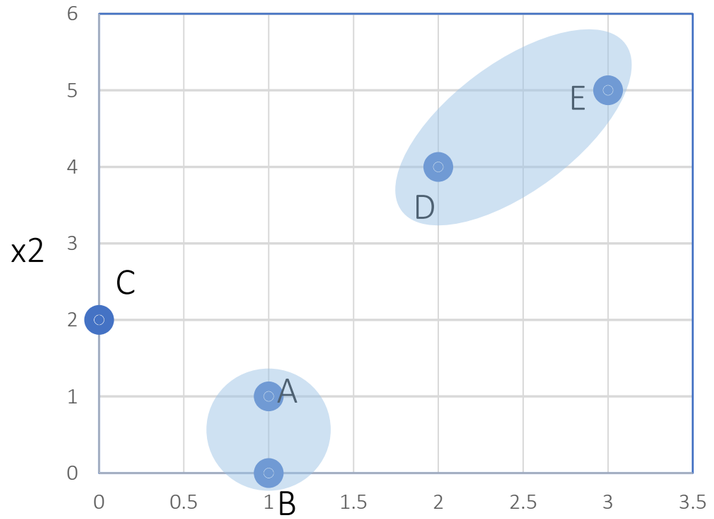
Dendrogram

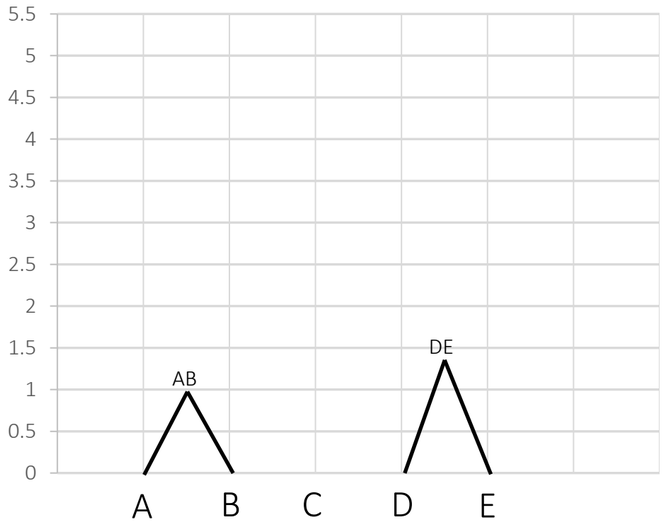
Distance based on centroid (bottom-up clustering)

	(A,B)	C	D	E
(A,B)	0	1.8	3.6	4.9
C	1.8	0	2.8	4.2
D	3.6	2.8	0	1.4
E	4.9	4.2	1.4	0

EUCLIDEAN DISTANCE

	(A,B)	C	(D,E)
(A,B)	0	1.8	4.25
C	1.8	0	3.5
(D,E)	4.25	3.5	0





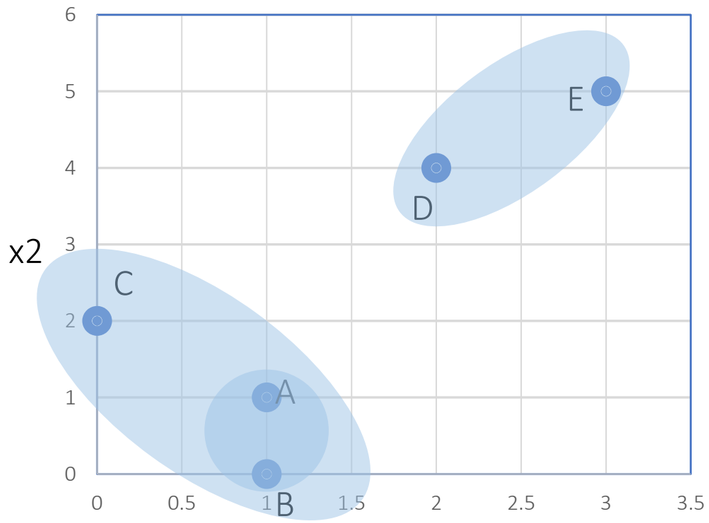
Dendrogram

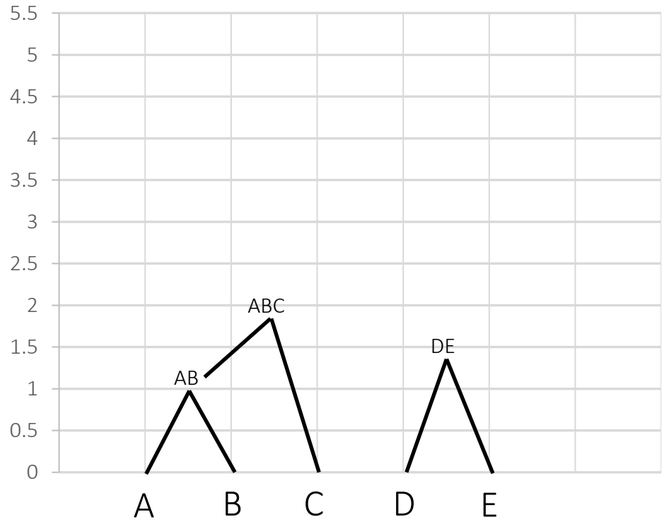
Distance based on centroid (bottom-up clustering)

	(A,B)	C	(D,E)
(A,B)	0	1.8	4.25
C	1.8	0	3.5
(D,E)	4.25	3.5	0

EUCLIDEAN DISTANCE

	((A,B),C)	(D,E)
((A,B),C)	0	3.875
(D,E)	3.875	0



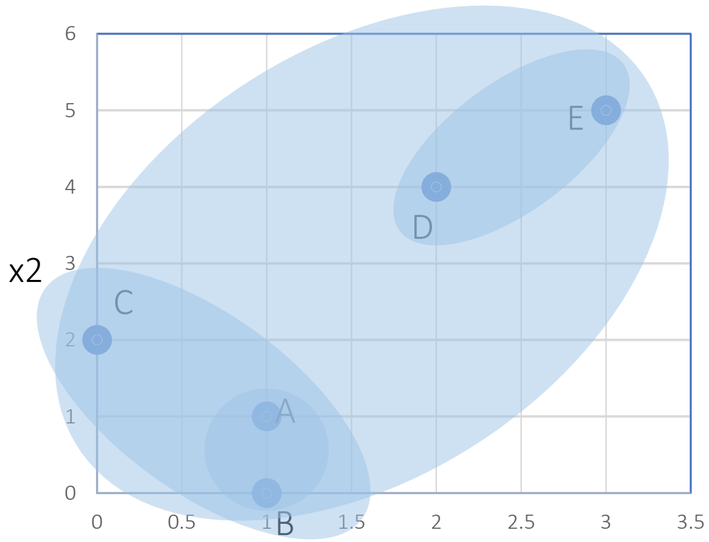


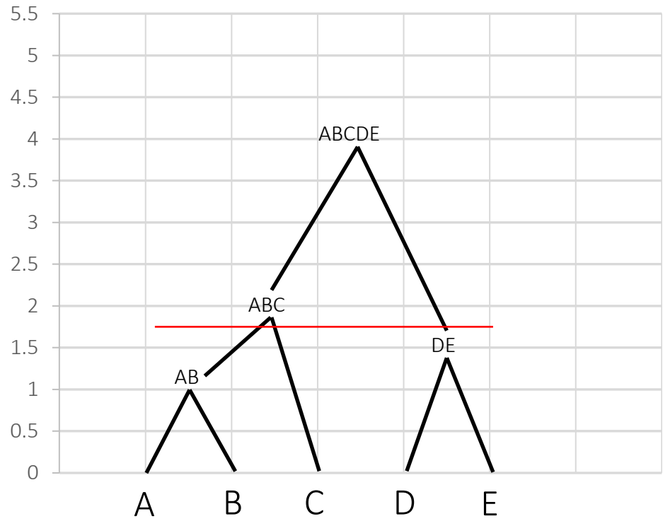
Dendrogram

Distance based on centroid (bottom-up clustering)

	((A,B),C)	(D,E)
((A,B),C)	0	3.875
(D,E)	3.875	0

	(((A,B),C),(D,E))
(((A,B),C),(D,E))	0



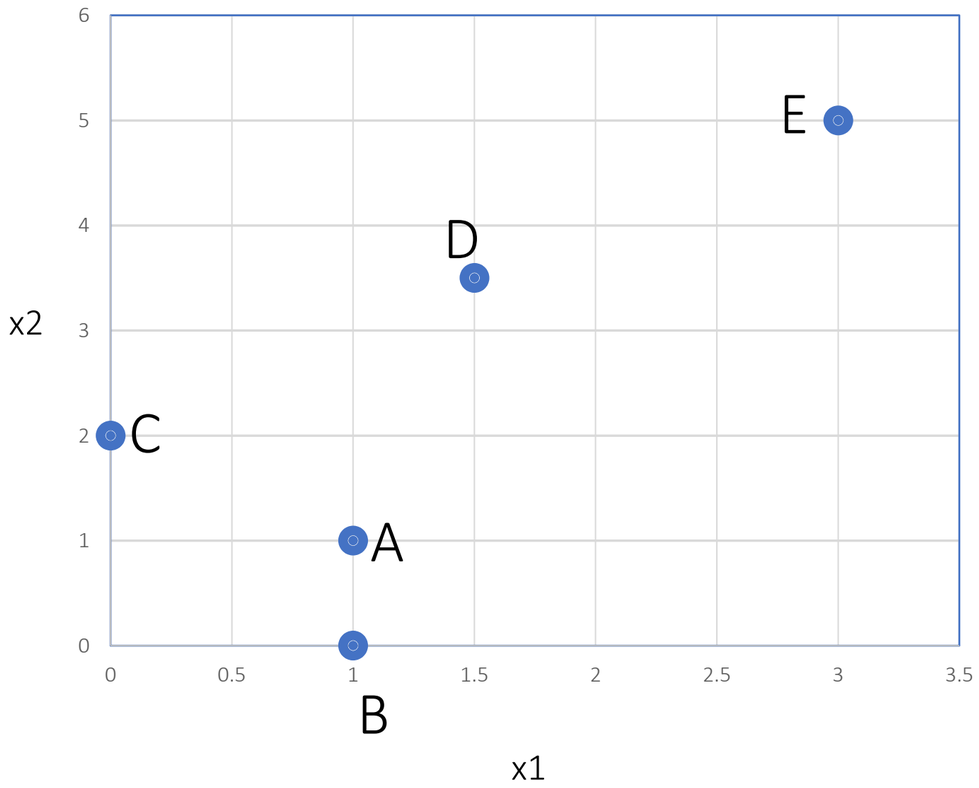


Dendrogram

Example:

Distance nearest points

i	x_1	x_2
A	1	1
B	1	0
C	0	2
D	1.5	3.5
E	3	5



EUCLIDEAN DISTANCE

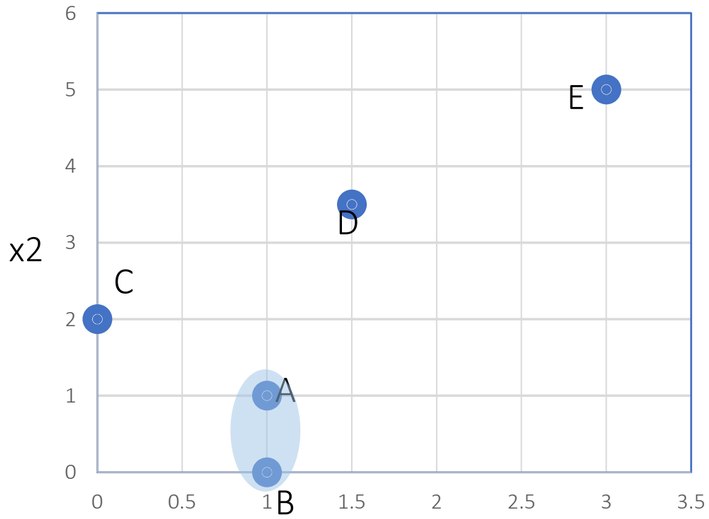
	A	B	C	D	E
A	0	1	1.4	2.55	4.5
B	1	0	2.2	3.53	5.4
C	1.4	2.2	0	2.12	4.2
D	2.55	3.53	2.12	0	2.12
E	4.5	5.4	4.2	2.12	0

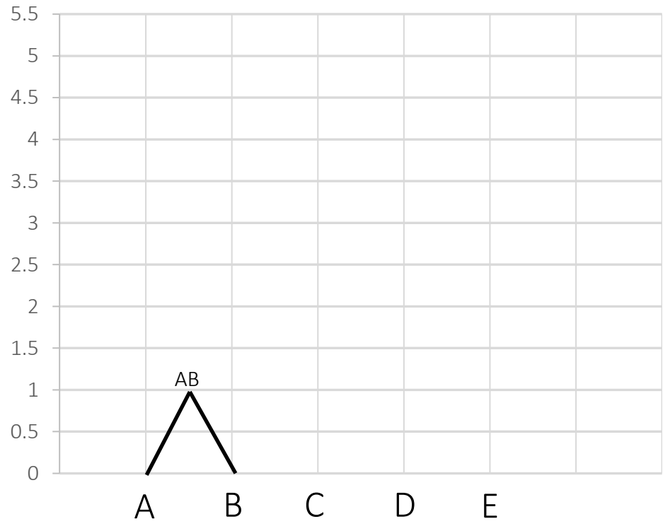
Distance based on single link (bottom-up clustering)

	A	B	C	D	E
A	0	1	1.4	2.55	4.5
B	1	0	2.2	3.53	5.4
C	1.4	2.2	0	2.12	4.2
D	2.55	3.53	2.12	0	2.12
E	4.5	5.4	4.2	2.12	0

EUCLIDEAN DISTANCE

	(A,B)	C	D	E
(A,B)	0	1.4	2.55	4.5
C	1.4	0	2.12	4.2
D	2.55	2.12	0	2.12
E	4.5	4.2	2.12	0





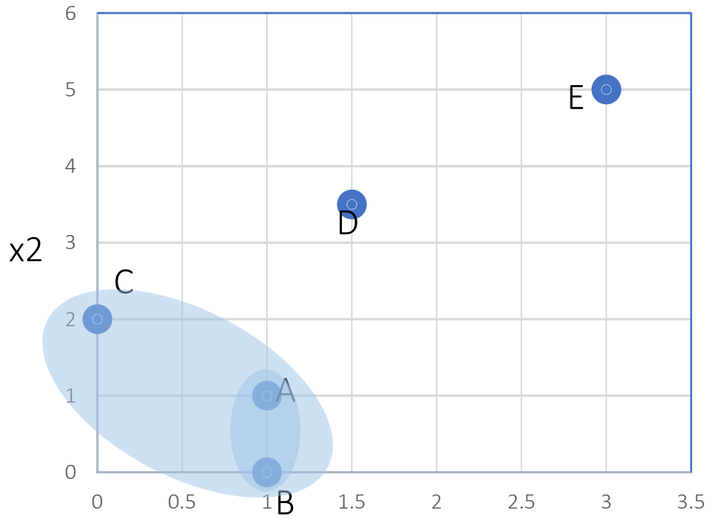
Dendrogram

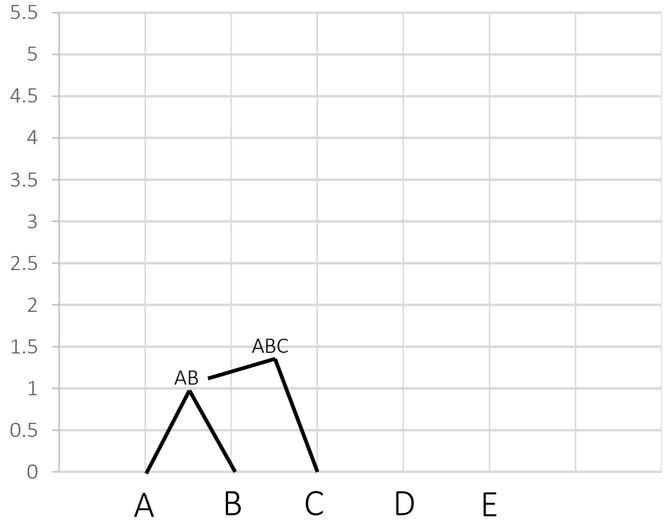
Distance based on single link (bottom-up clustering)

	(A,B)	C	D	E
(A,B)	0	1.4	2.55	4.5
C	1.4	0	2.12	4.2
D	2.55	2.12	0	2.12
E	4.5	4.2	2.12	0

EUCLIDEAN DISTANCE

	(A,B),C	D	E
(A,B),C	0	2.12	4.2
D	2.12	0	2.12
E	4.2	2.12	0





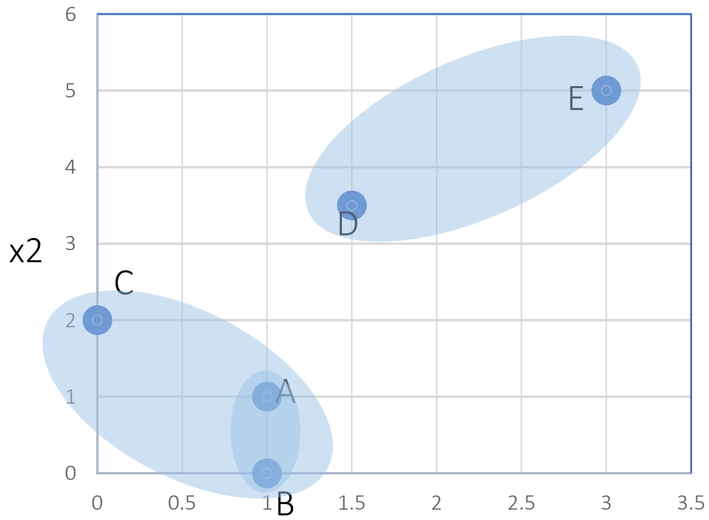
Dendrogram

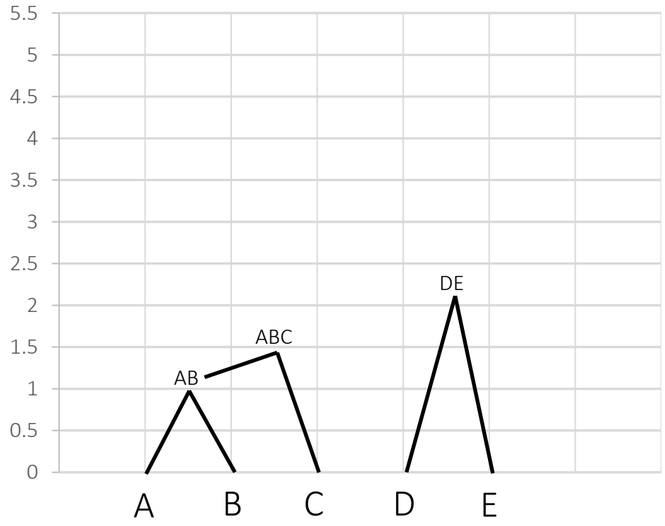
Distance based on single link (bottom-up clustering)

	(A,B),C	D	E
(A,B),C	0	2.12	4.2
D	2.12	0	2.12
E	4.2	2.12	0

EUCLIDEAN DISTANCE

	((A,B),C)	(D,E)
((A,B),C)	0	2.12
(D,E)	2.12	0





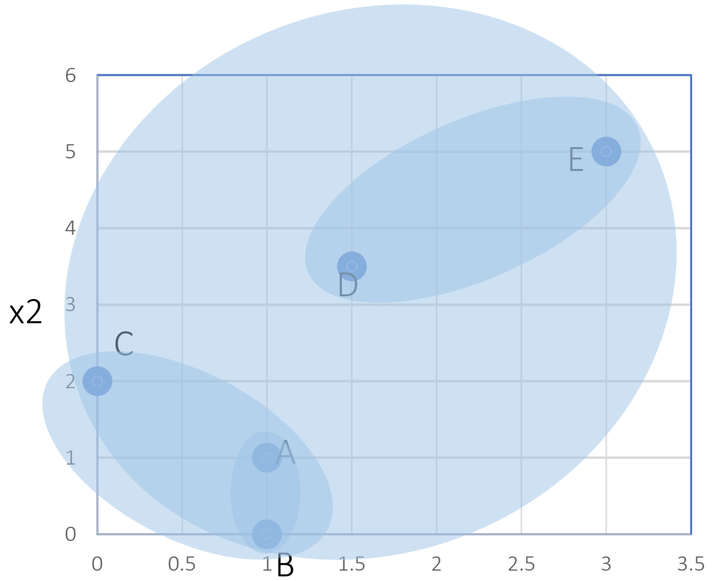
Dendrogram

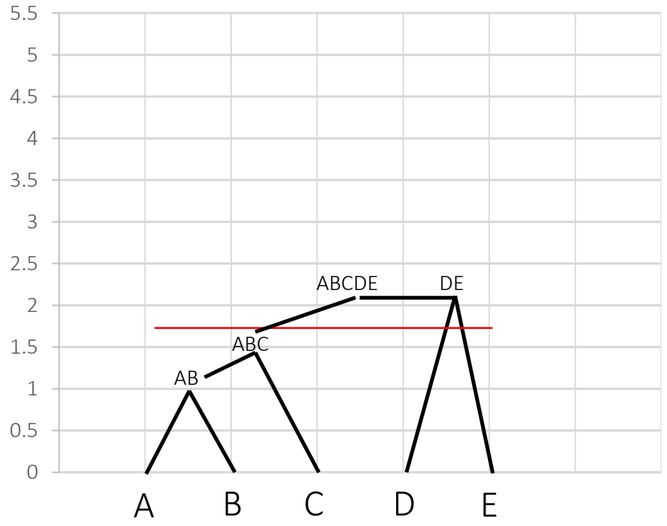
Distance based on single link (bottom-up clustering)

	((A,B),C)	(D,E)
((A,B),C)	0	2.12
(D,E)	2.12	0

EUCLIDEAN DISTANCE

	(((A,B),C),(D,E))
(((A,B),C),(D,E))	0



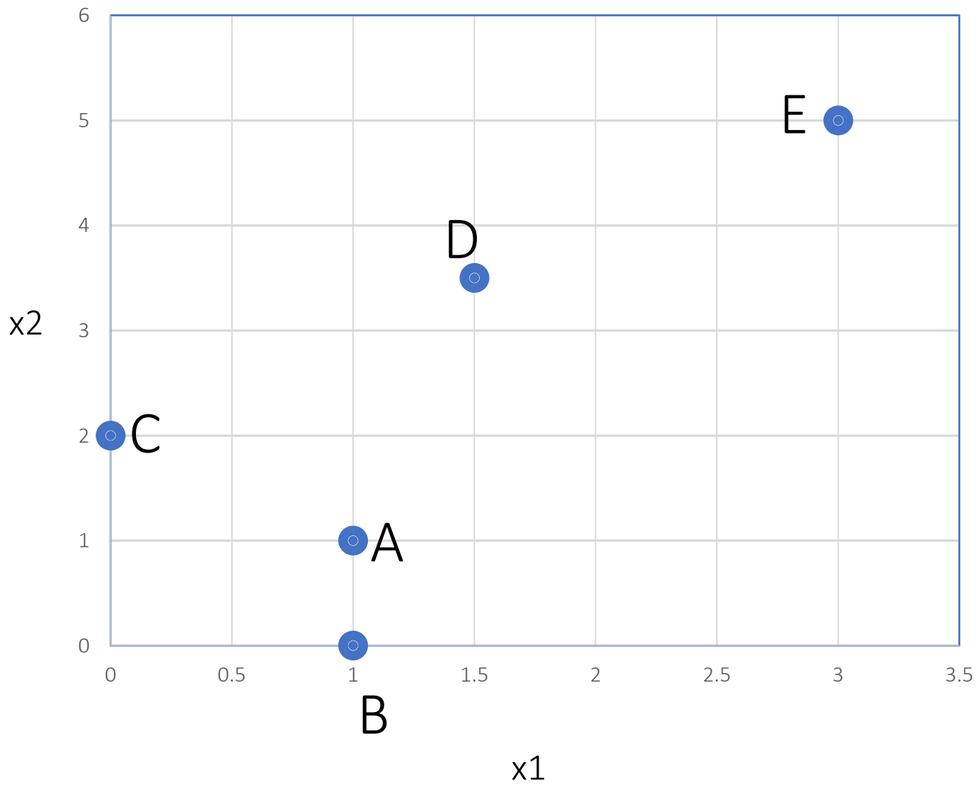


Dendrogram

Example:

Distance farthest points

i	x_1	x_2
A	1	1
B	1	0
C	0	2
D	1.5	3.5
E	3	5



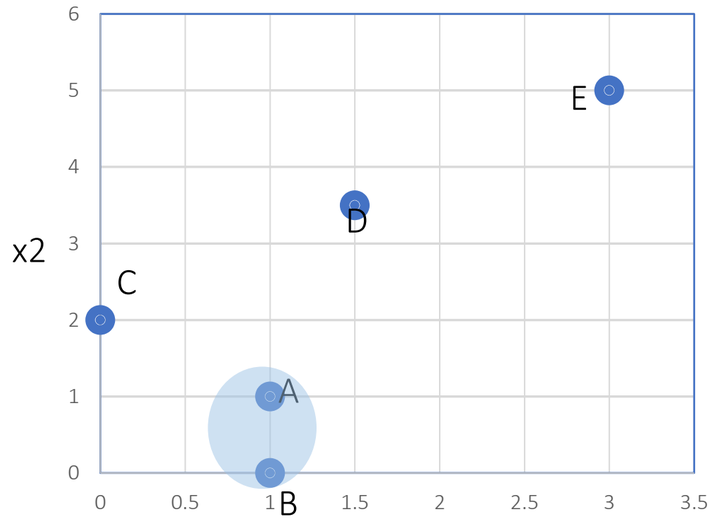
EUCLIDEAN DISTANCE

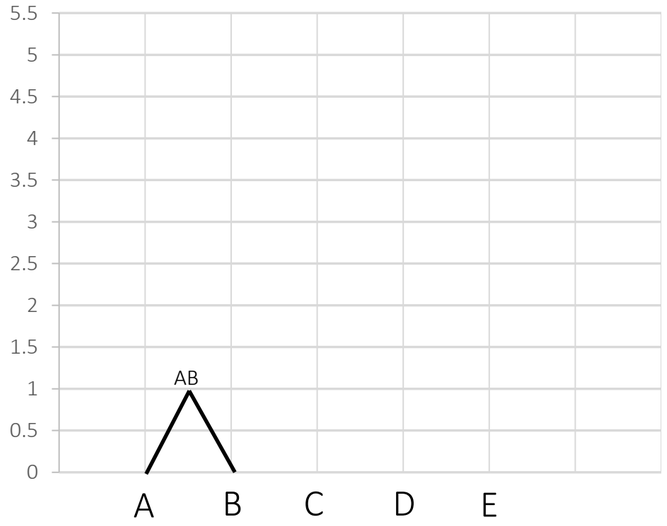
	A	B	C	D	E
A	0	1	1.4	2.55	4.5
B	1	0	2.2	3.53	5.4
C	1.4	2.2	0	2.12	4.2
D	2.55	3.53	2.12	0	2.12
E	4.5	5.4	4.2	2.12	0

Distance based on complete link (bottom-up clustering)

	A	B	C	D	E
A	0	1	1.4	2.55	4.5
B	1	0	2.2	3.53	5.4
C	1.4	2.2	0	2.12	4.2
D	2.55	3.53	2.12	0	2.12
E	4.5	5.4	4.2	2.12	0

	(A,B)	C	D	E
(A,B)	0	2.2	3.55	5.4
C	2.2	0	2.12	4.2
D	3.55	2.12	0	2.12
E	5.4	4.2	2.12	0





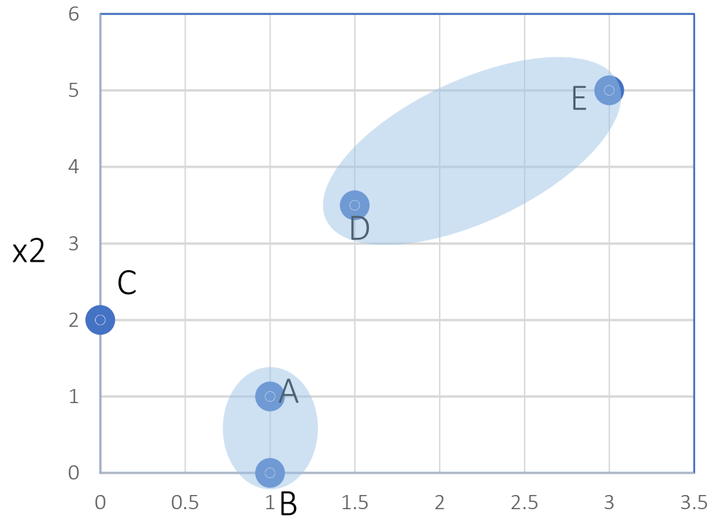
Dendrogram

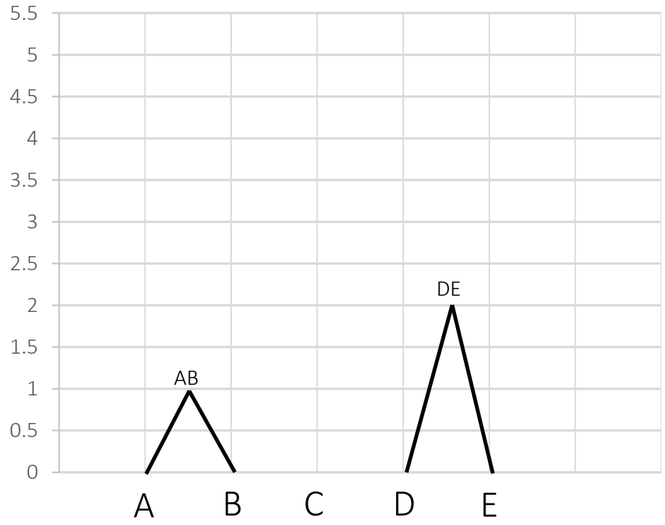
Distance based on complete link (bottom-up clustering)

	(A,B)	C	D	E
(A,B)	0	2.2	3.55	5.4
C	2.2	0	2.12	4.2
D	3.55	2.12	0	2.12
E	5.4	4.2	2.12	0

EUCLIDEAN DISTANCE

	(A,B)	C	(D,E)
(A,B)	0	2.2	5.4
C	2.2	0	4.2
(D,E)	5.4	4.2	0





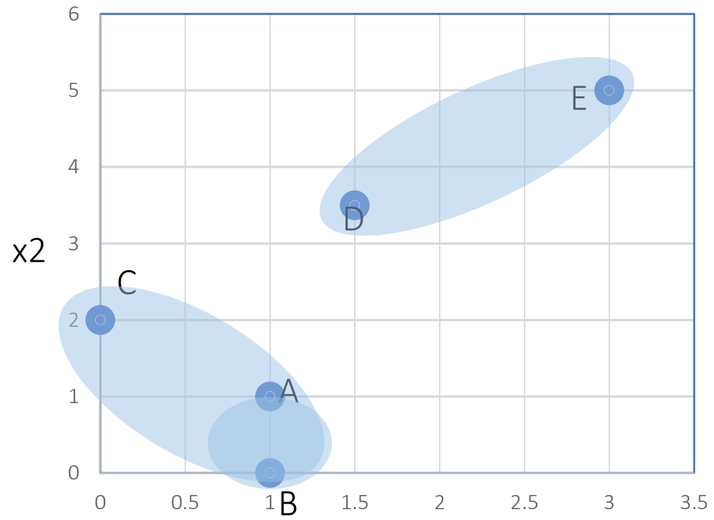
Dendrogram

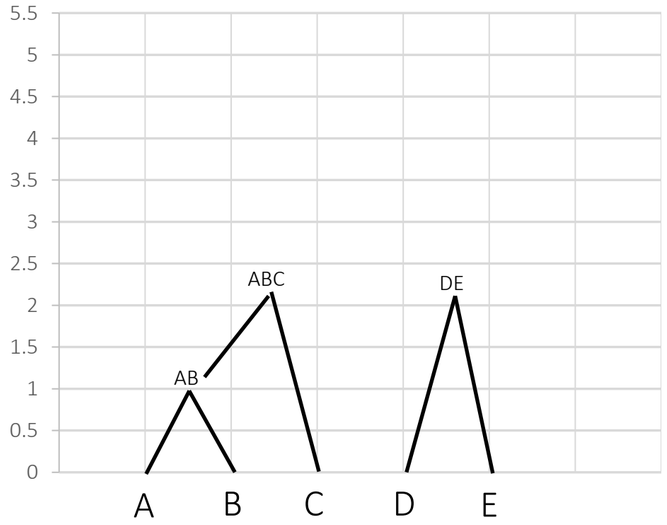
Distance based on complete link (bottom-up clustering)

	(A,B)	C	(D,E)
(A,B)	0	2.2	5.4
C	2.2	0	4.2
(D,E)	5.4	4.2	0

EUCLIDEAN DISTANCE

	((A,B),C)	(D,E)
((A,B),C)	0	5.4
(D,E)	5.4	0





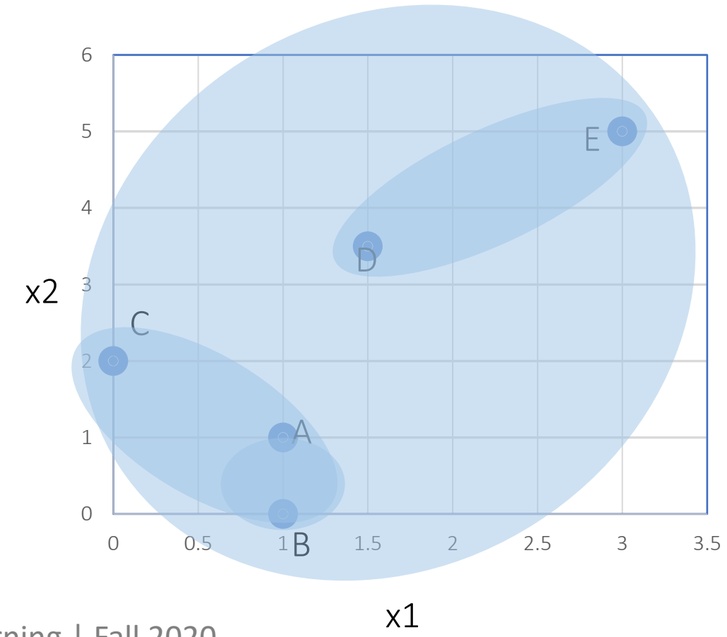
Dendrogram

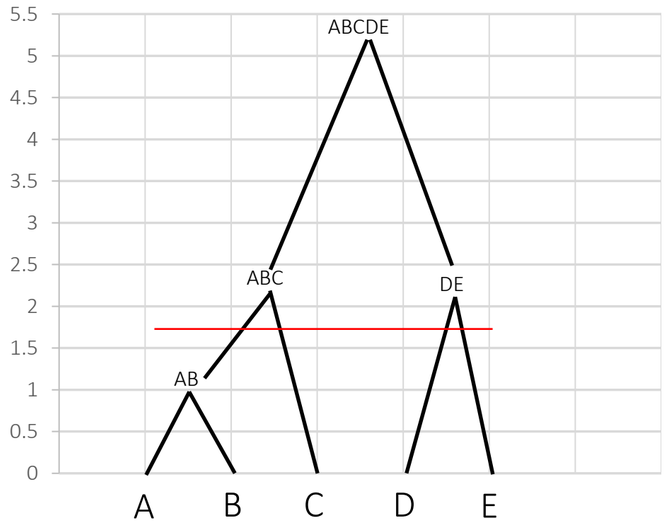
Distance based on complete link (bottom-up clustering)

	((A,B),C)	(D,E)
((A,B),C)	0	5.4
(D,E)	5.4	0

EUCLIDEAN DISTANCE

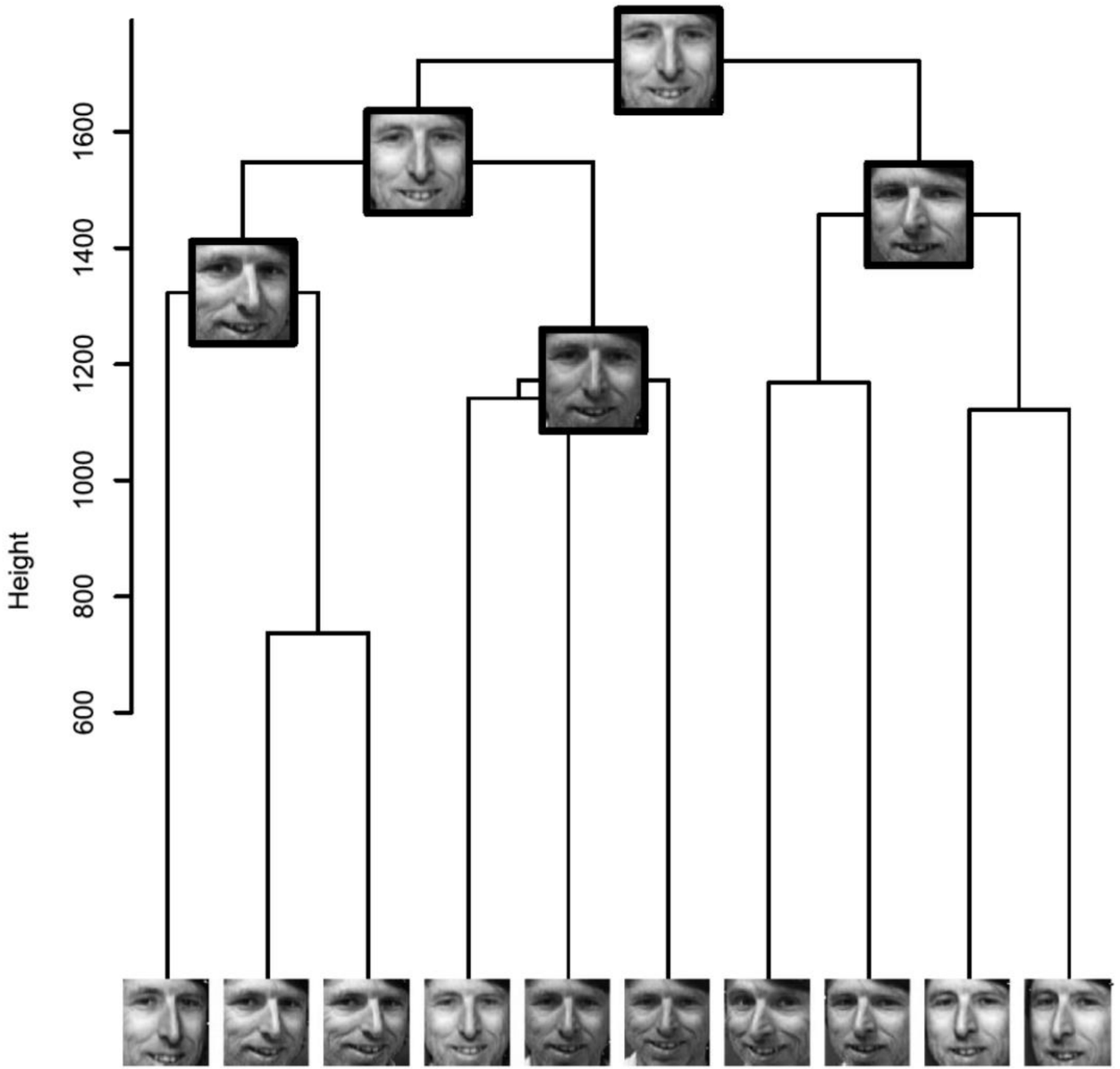
	(((A,B),C),(D,E))
(((A,B),C),(D,E))	0





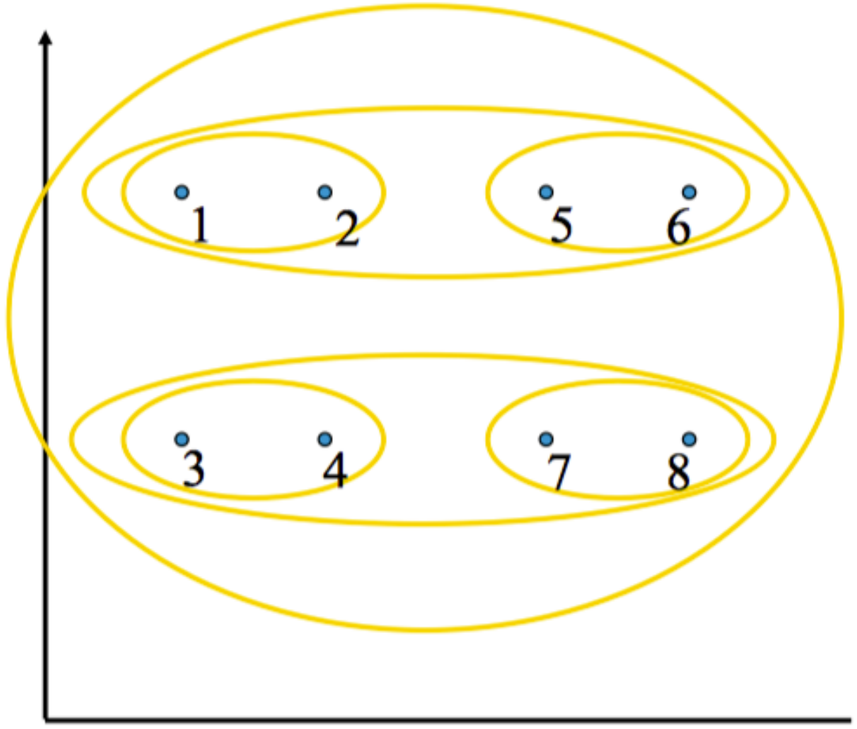
Dendrogram

Example

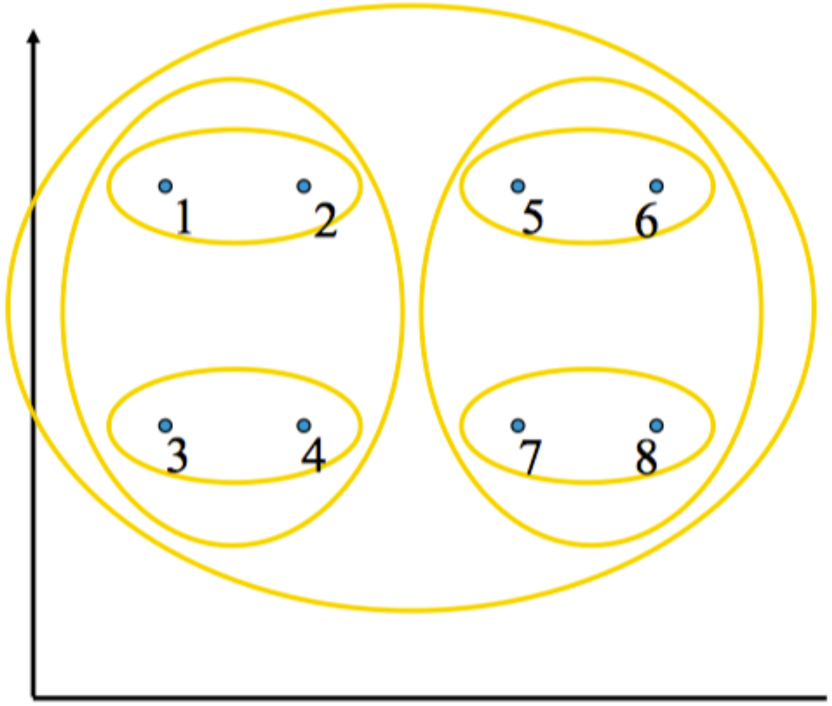


(From Bien et al. (2011))

Closest pair
(single-link clustering)



Farthest pair
(complete-link clustering)



Slide credit: David Sontag