Hierarchical Clustering

Hierarchical Clustering

Two main types of hierarchical clustering:

Agglomerative:

1. Start with every point in its own cluster
2. At each step, merge the two closest clusters
3. Stop when every point is in the same cluster

Divisive:

1. Start with every point in the same cluster
2. At each step, split until every point is in its own cluster

Hierarchical Clustering

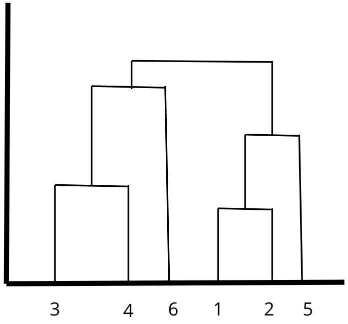
Our main focus will be on **agglomerative** methods

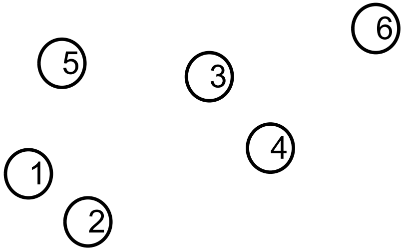
Agglomerative Clustering Algorithm

1. Let each point in the dataset be in its own cluster
2. Compute the distance between all pairs of clusters
3. Merge the two closest clusters
4. Repeat 3 & 4 until all points are in the same cluster

Hierarchical Clustering

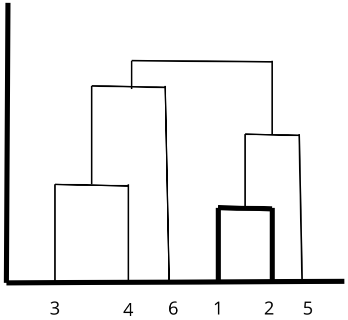
At every step, we record which clusters were merged in order to produce a dendrogram:

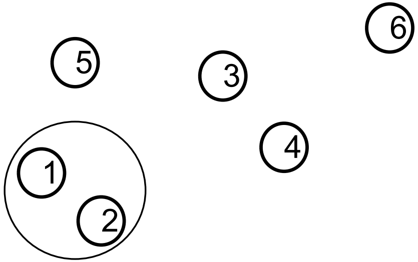




Hierarchical Clustering

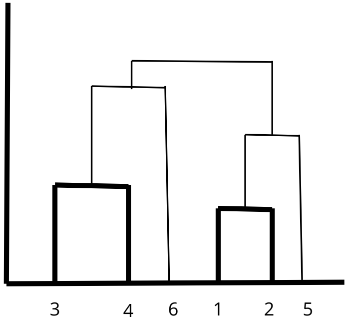
At every step, we record which clusters were merged in order to produce a dendrogram:

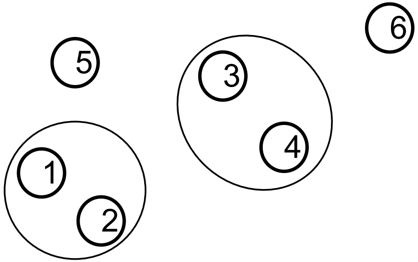




Hierarchical Clustering

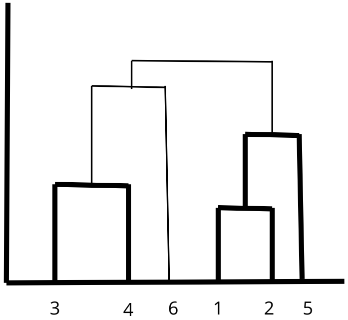
At every step, we record which clusters were merged in order to produce a dendrogram:

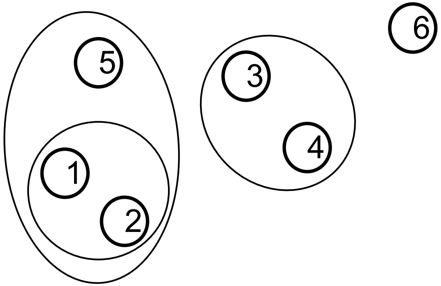




Hierarchical Clustering

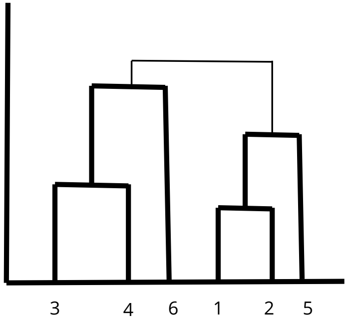
At every step, we record which clusters were merged in order to produce a dendrogram:

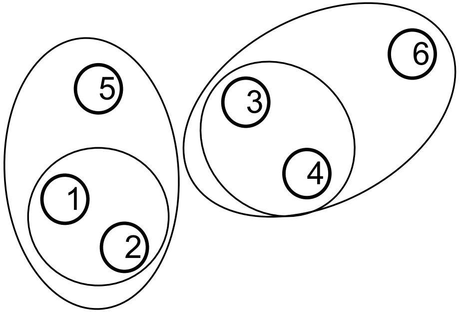




Hierarchical Clustering

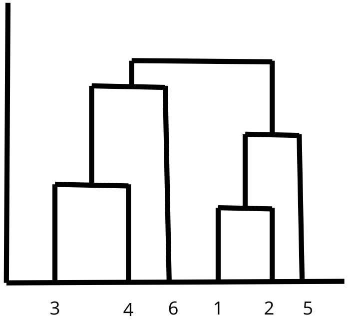
At every step, we record which clusters were merged in order to produce a dendrogram:

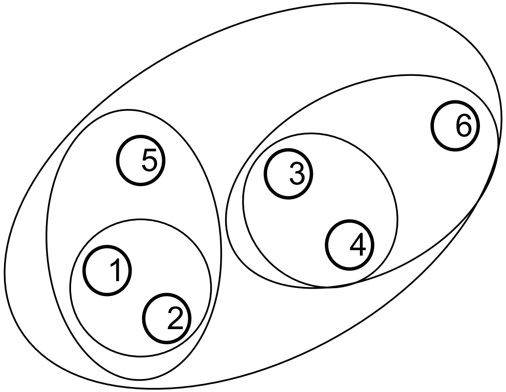




Hierarchical Clustering

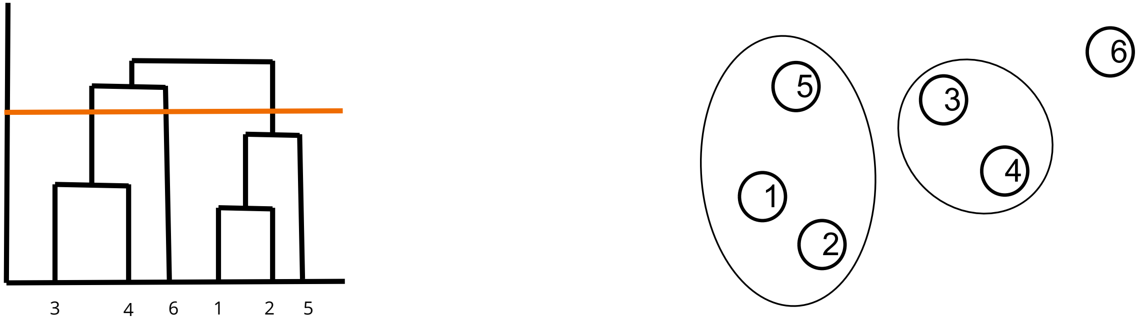
At every step, we record which clusters were merged in order to produce a dendrogram:





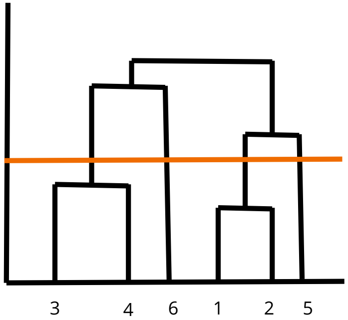
Hierarchical Clustering

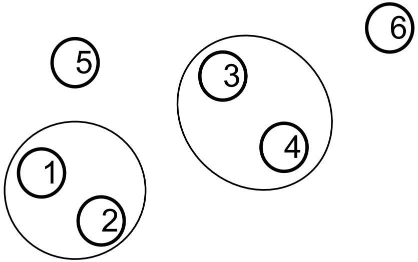
We can “cut” the dendrogram at any threshold to produce any number of clusters



Hierarchical Clustering

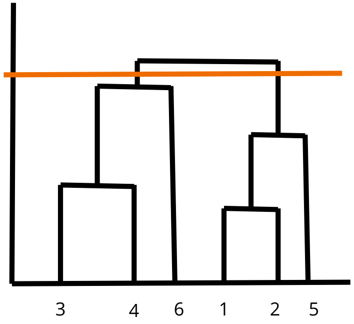
We can “cut” the dendrogram at any threshold to produce any number of clusters

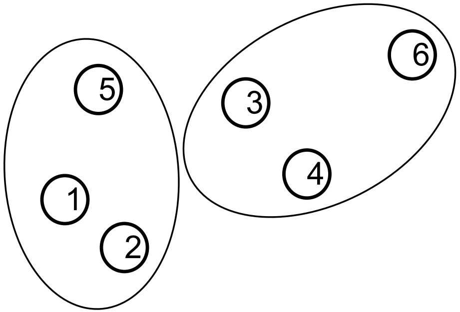




Hierarchical Clustering

We can “cut” the dendrogram at any threshold to produce any number of clusters





Hierarchical Clustering

Can we implement this? Are we missing anything?

How do we compute the distance between clusters?

Distance between clusters can be thought of as distance between two sets of points. What ideas come to mind?

Hierarchical Clustering - Distance Functions

Let's first define:

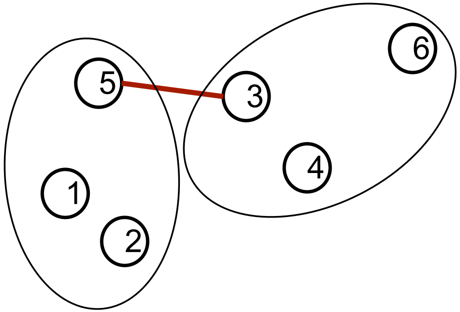
Distance between points: $\mathbf{d(p_1, p_2)}$

Distance between clusters: $\mathbf{D(C_1, C_2)}$

Single-Link Distance

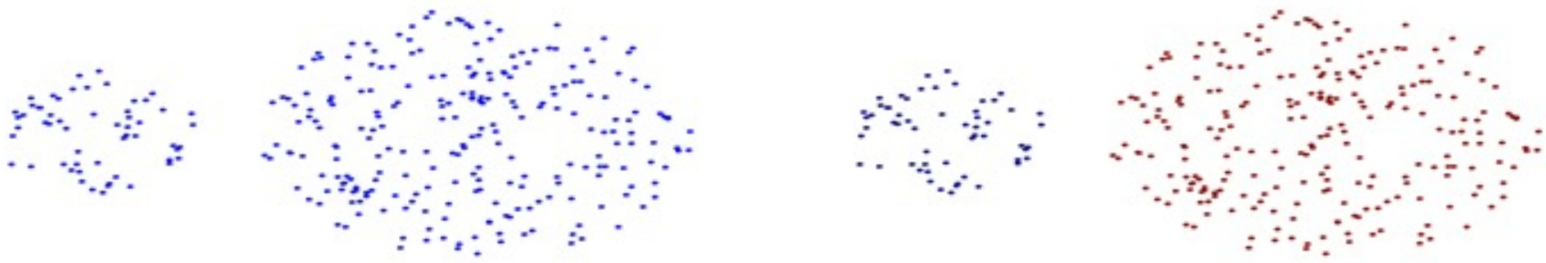
Is the **minimum** of all pairwise distances between a point from one cluster and a point from the other cluster.

$$D_{SL}(C_1, C_2) = \min \{d(p_1, p_2) \mid p_1 \in C_1, p_2 \in C_2\}$$



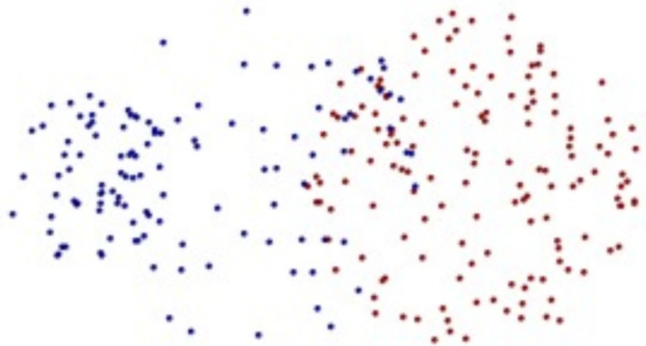
Depends on choice of **d**

Single-Link Distance



Can handle clusters of different sizes

Single-Link Distance

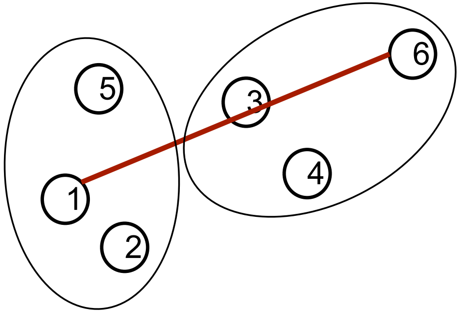


But... Sensitive to noise points
Tends to create elongated clusters
拉長的

Complete-Link Distance

Is the **maximum** of all pairwise distances between a point from one cluster and a point from the other cluster.

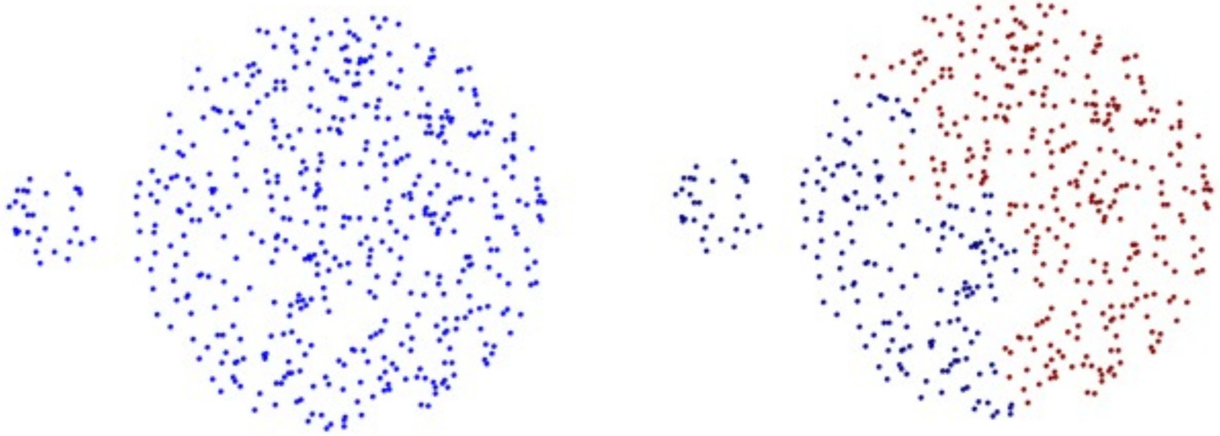
$$D_{CL}(C_1, C_2) = \max\{d(p_1, p_2) \mid p_1 \in C_1, p_2 \in C_2\}$$



Complete-Link Distance

Less susceptible to noise
Creates more balanced (equal diameter) clusters

Complete-Link Distance

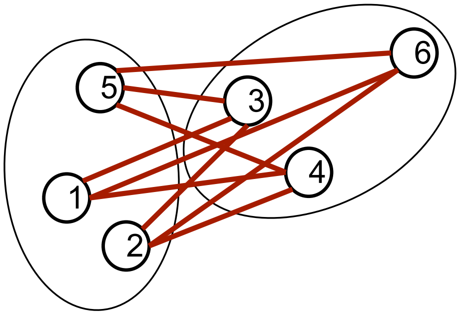


But... Tends to split up large clusters.
All clusters tend to have the same diameter

Average-Link Distance

Is the **average** of all pairwise distances between a point from one cluster and a point from the other cluster.

$$D_{AL}(C_1, C_2) = \frac{1}{|C_1| \cdot |C_2|} \sum_{p_1 \in C_1, p_2 \in C_2} d(p_1, p_2)$$



Average-Link Distance

Less susceptible to noise and outliers.

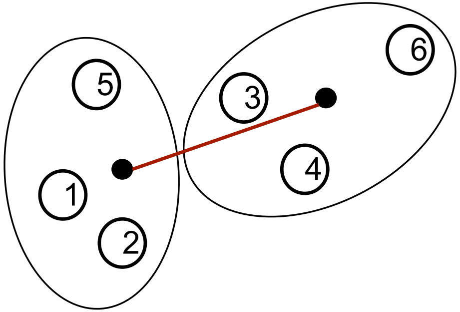
But… Tends to be biased toward globular clusters
球形的

幾何中心

Centroid Distance

The distance between the centroids of clusters.

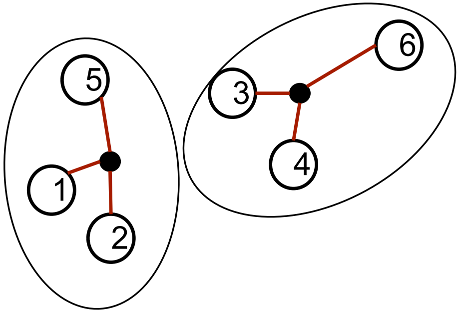
$$D_C(C_1, C_2) = d(\mu_1, \mu_2)$$



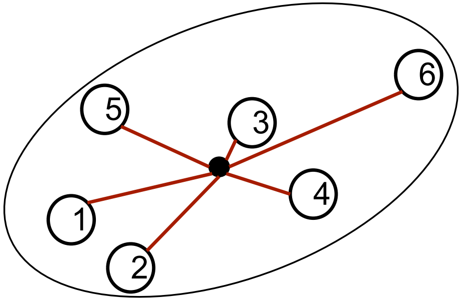
Ward's Distance

Is the difference between the spread / (variance) of points in the merged cluster and the unmerged clusters.

$$D_{WD}(C_1, C_2) = \sum_{p \in C_{12}} d(p, \mu_{12}) - \sum_{p_1 \in C_1} d(p_1, \mu_1) - \sum_{p_2 \in C_2} d(p_2, \mu_2)$$



VS



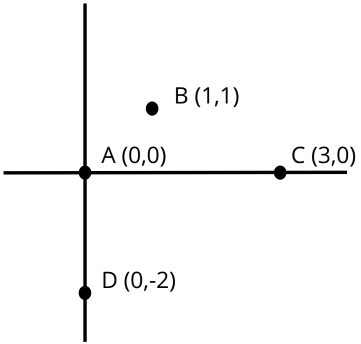
聚合的

Agglomerative Clustering Algorithm

1. Let each point in the dataset be in its own cluster
2. Compute the distance between all pairs of clusters
3. Merge the two closest clusters
4. Repeat 3 & 4 until all points are in the same cluster

Example

d = Euclidean
D = Single-Link

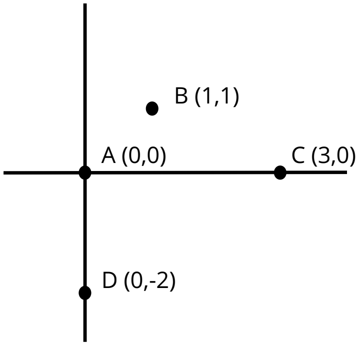


Distance Matrix

	A	B	C	D
A				
B				
C				
D				

Example

d = Euclidean
D = Single-Link

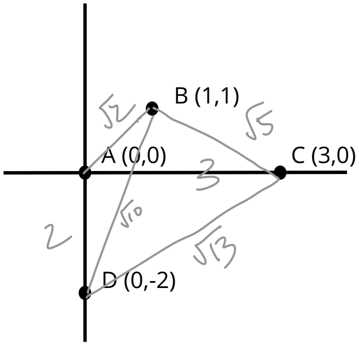


Distance Matrix

	A	B	C	D
A	0			
B		0		
C			0	
D				0

Example

d = Euclidean
D = Single-Link

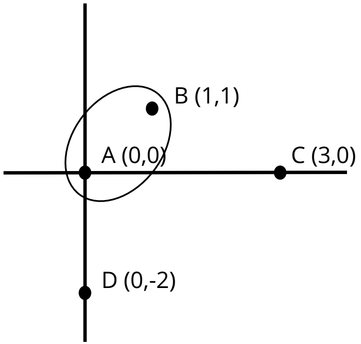


Distance Matrix

	A	B	C	D
A	0	√2	3	2
B	√2	0	√5	√10
C	5	√5	0	√13
D	2	√10	√13	0

Example

d = Euclidean
D = Single-Link

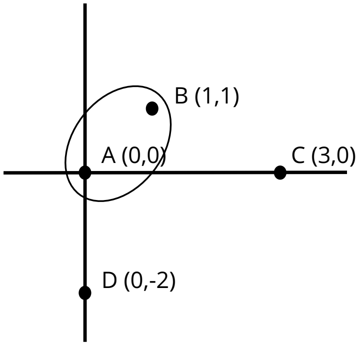


Distance Matrix

	A	B	C	D
A	0	√2	3	2
B	√2	0	√5	√10
C	5	√5	0	√13
D	2	√10	√13	0

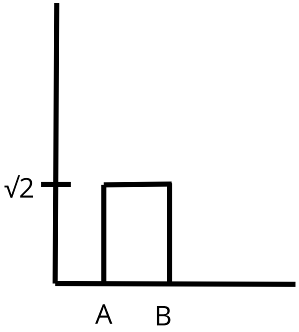
Example

d = Euclidean
D = Single-Link



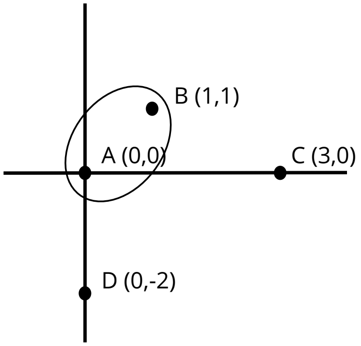
系統樹圖

Dendrogram



Example

d = Euclidean
D = Single-Link



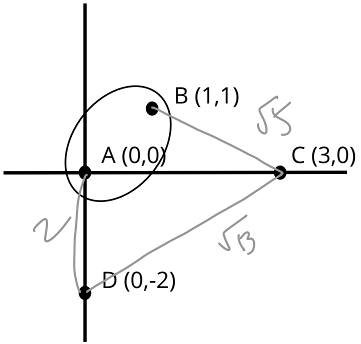
Distance Matrix

	A & B	C	D
A & B	0		
C		0	√13
D		√13	0

Example

d = Euclidean
D = Single-Link

minimum

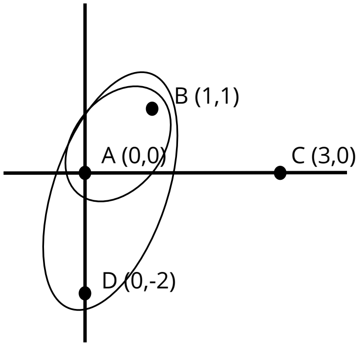


Distance Matrix

	A & B	C	D
A & B	0	√5	2
C	√5	0	√13
D	2	√13	0

Example

d = Euclidean
D = Single-Link

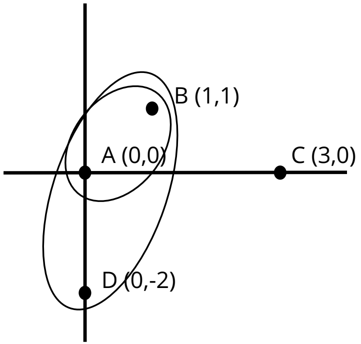


Distance Matrix

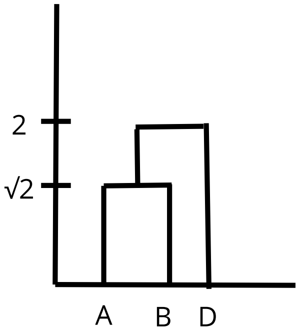
	A & B	C	D
A & B	0	√5	2
C	√5	0	√13
D	2	√13	0

Example

d = Euclidean
D = Single-Link

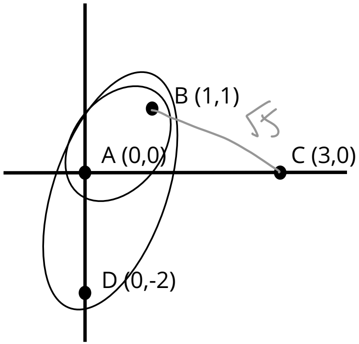


Dendrogram



Example

d = Euclidean
D = Single-Link

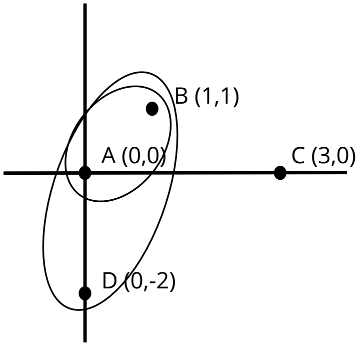


Distance Matrix

	A & B & D	C
A & B & D	0	
C		0

Example

d = Euclidean
D = Single-Link

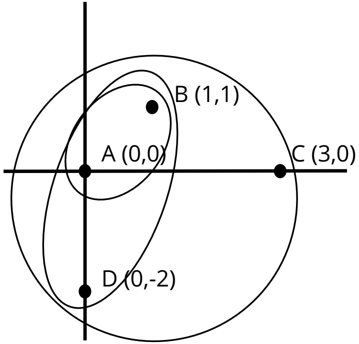


Distance Matrix

	A & B & D	C
A & B & D	0	√5
C	√5	0

Example

d = Euclidean
D = Single-Link

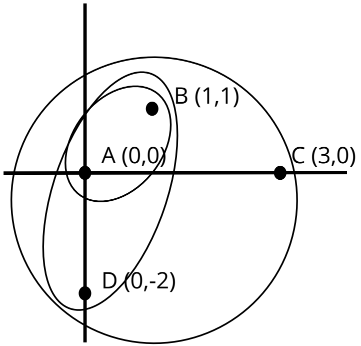


Distance Matrix

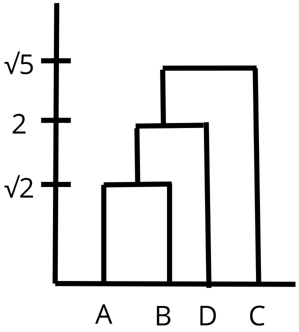
	A & B & D	C
A & B & D	0	√5
C	√5	0

Example

d = Euclidean
D = Single-Link



Dendrogram



Hierarchical Clustering

Finding the threshold with which to cut the dendrogram requires exploration and tuning. But in general hierarchical clustering is used to expose a hierarchy in the data (ex: finding/defining species via DNA similarity).

To capture the difference between clusterings you can use a cost function, or methods that we will discuss later when we look at clustering aggregation.