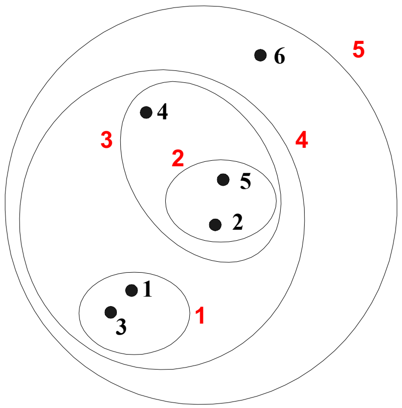
Outline

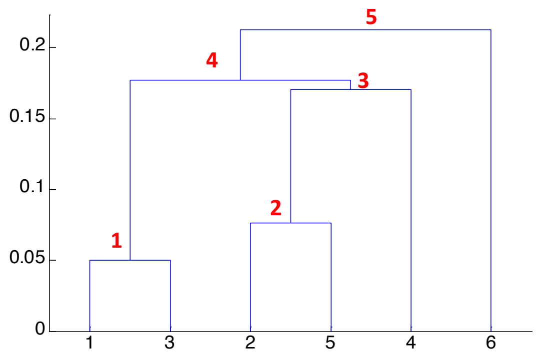
- Unsupervised learning vs supervised learning
- A categorization of major clustering methods
- Partitioning-based clustering
- Hierarchical-based clustering

Hierarchical methods idea

- 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
  - The height at which two clusters are merged in the dendrogram reflects their distance



Nested clusters



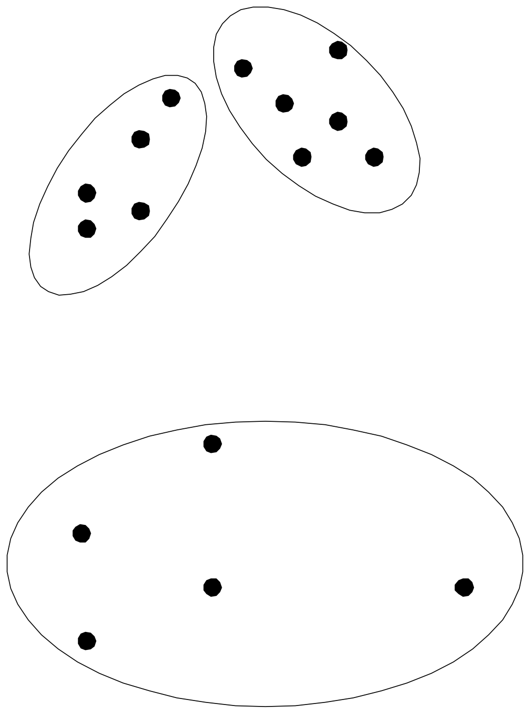
Dendrogram

Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level

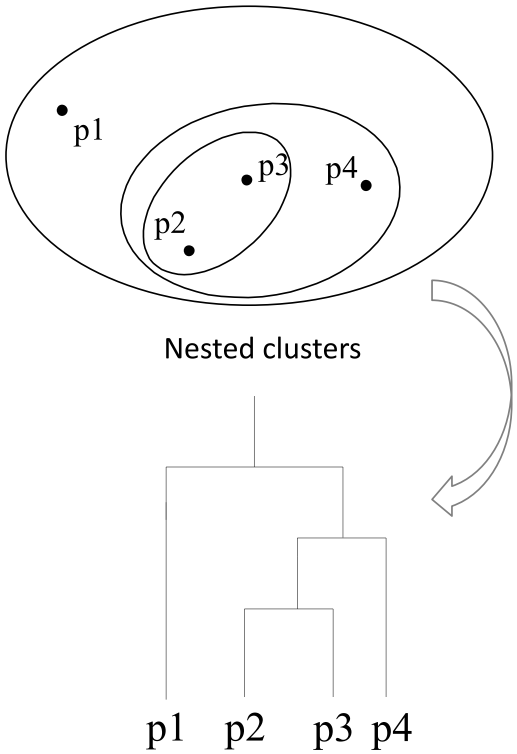
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Hierarchical vs Partitioning



Partitioning clustering

Partitioning algorithms typically have global objectives

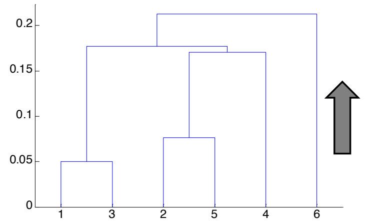


Dendrogram

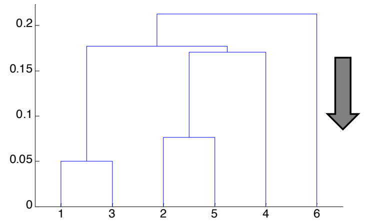
Hierarchical clustering algorithms typically have local objectives

Hierarchical clustering methods

- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or *k* clusters) left
    - e.g., AGNES



  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are *k* clusters)
    - e.g., DIANA



- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge <u>two in one</u> or split <u>one in two</u> cluster at a time

Agglomerative clustering algorithm

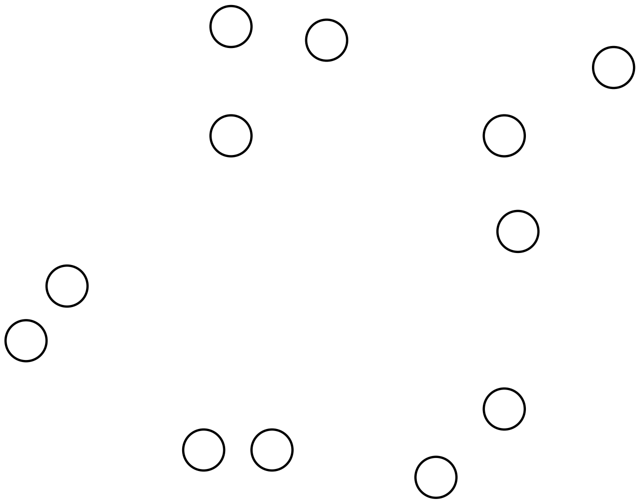
- More popular hierarchical clustering technique
- Basic algorithm is straightforward

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 points:
  - the computation of the proximity of two clusters
    - Different approaches to defining the distance between clusters distinguish the different algorithms (single link, complete link, …..)
  - the update of the proximity matrix due to cluster merges

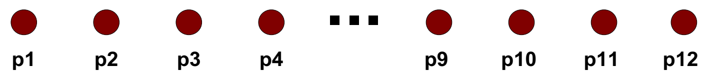
Starting situation

- Start with clusters of individual points and a proximity matrix



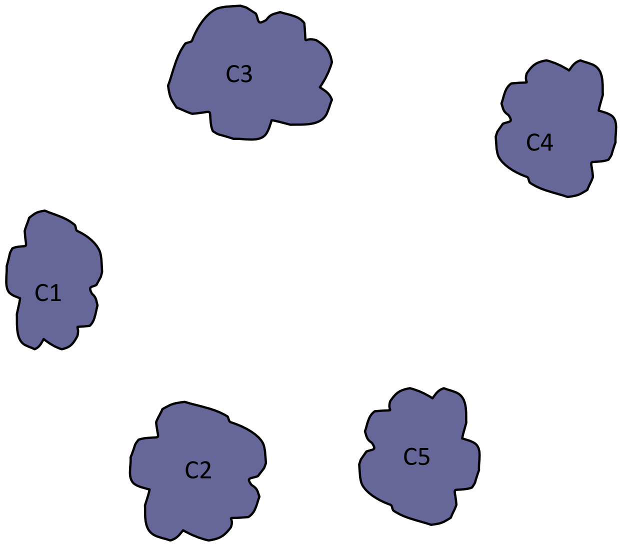
	p1	p2	p3	...	p12
p1					
p2					
p3					
...					
p12					

Proximity matrix



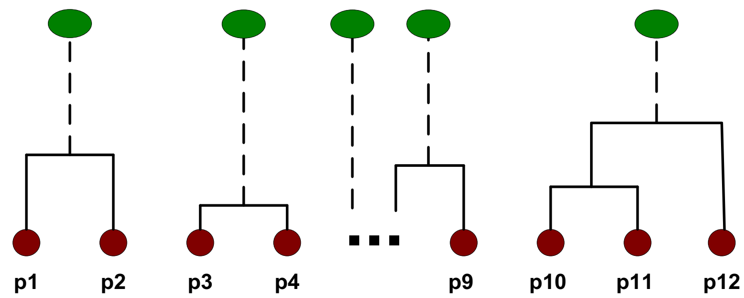
Intermediate situation I

- After some merging steps, we have some clusters



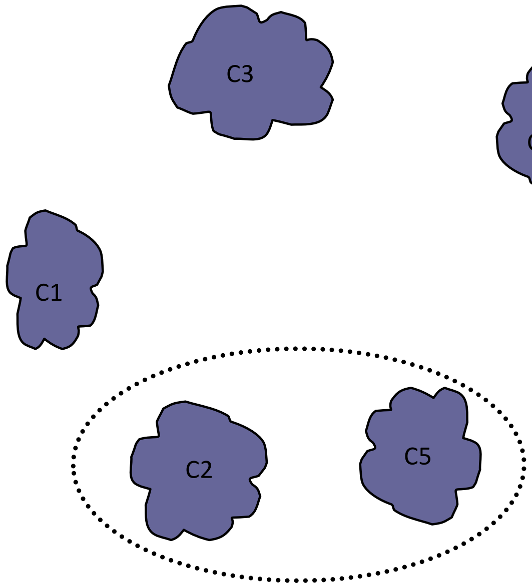
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

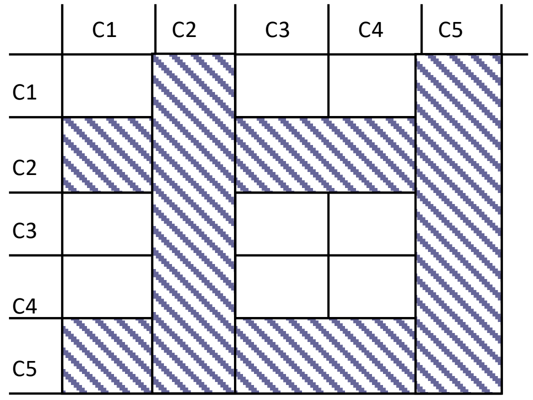
Proximity matrix



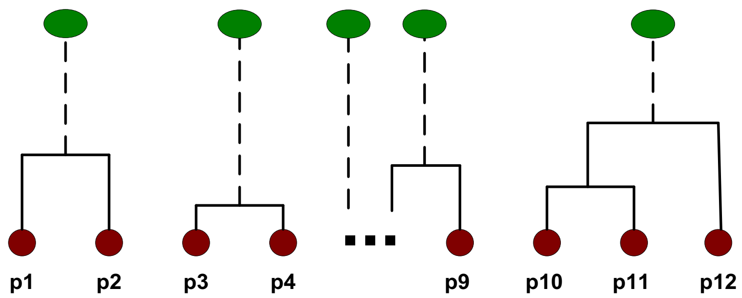
Intermediate situation II

- We want to merge the two closest clusters (C_2 and C_5) and update the proximity matrix.



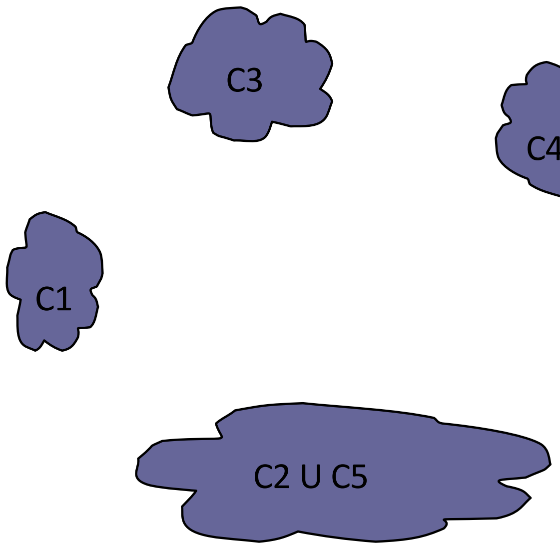


Proximity matrix



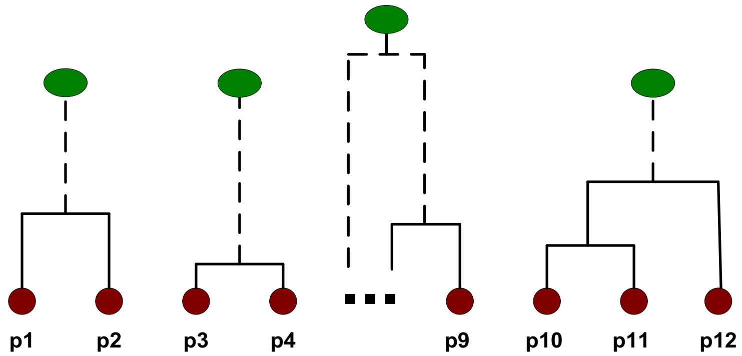
After merging

- The question is “How do we update the proximity matrix?” Or, in other words, what is the similarity between two clusters?

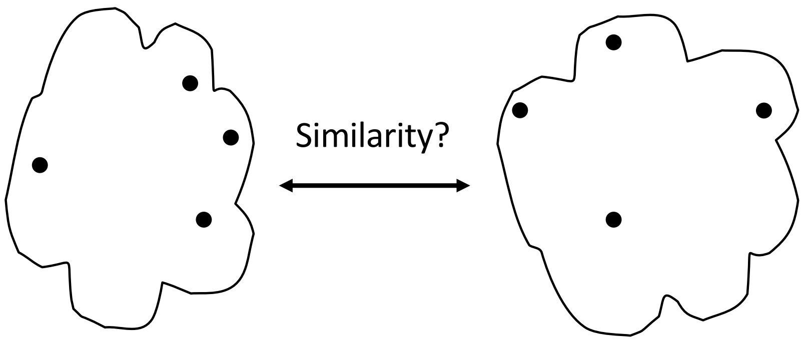


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity matrix



Measures of inter-cluster similarity I

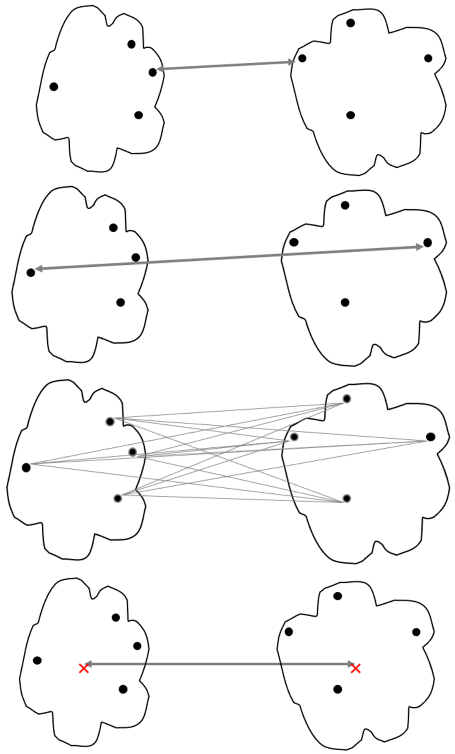


	p1	p2	p3	...	p12
p1					
p2					
p3					
...					
p12					

Proximity matrix

- A variety of different measures:
- Single link (or MIN)
- Complete link (or MAX)
- Group average
- Distance between centroids
- Distance between medoids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Typical alternatives to calculate the distance between clusters



- **Single link**: smallest distance between an element in one cluster and an element in the other, i.e., $dis_{sl}(C_i, C_j) = \min_{x,y} \{d(x,y) | x \in C_i, y \in C_j\}$
- **Complete link**: largest distance between an element in one cluster and an element in the other, i.e., $dis_{cl}(C_i, C_j) = \max_{x,y} \{d(x,y) | x \in C_i, y \in C_j\}$
- **Average**: avg distance between an element in one cluster and an element in the other, i.e.,

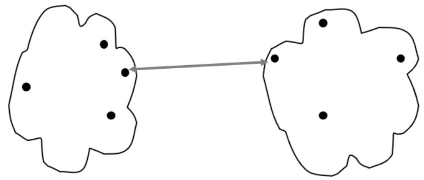
$$dis_{avg}(C_i, C_j) = \frac{\sum_{x \in C_i, y \in C_j} d(x,y)}{|C_i||C_j|}$$

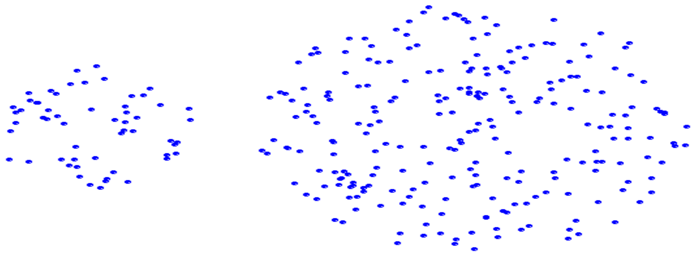
- **Centroid**: distance between the centroids of two clusters, i.e.,

$$dis_{centroids}(C_i, C_j) = d(c_i, c_j)$$

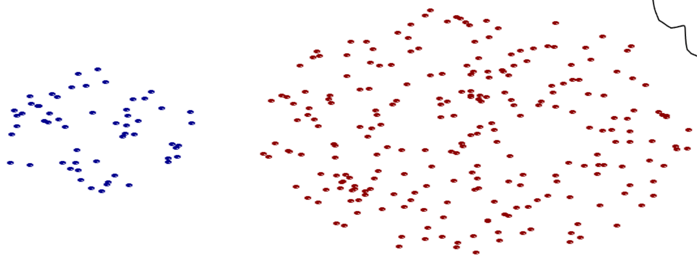
- **Medoid**: distance between the medoids of two clusters, i.e., dis(K_i, K_j) = dis(M_i, M_j)
  - Medoid: one chosen, centrally located object in the cluster

Single link distance (MIN): strengths



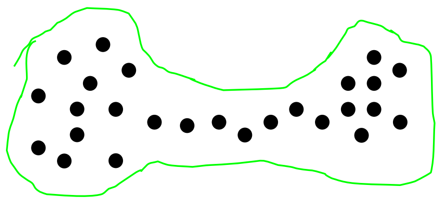


Original points

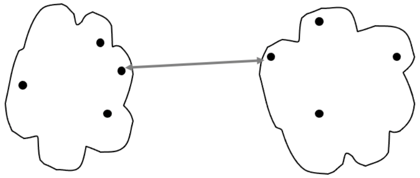


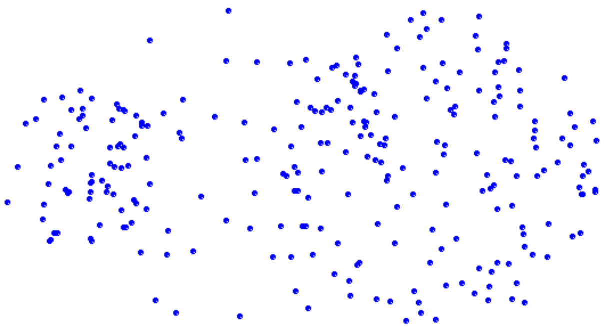
Two clusters

- Can handle non-elliptical shapes

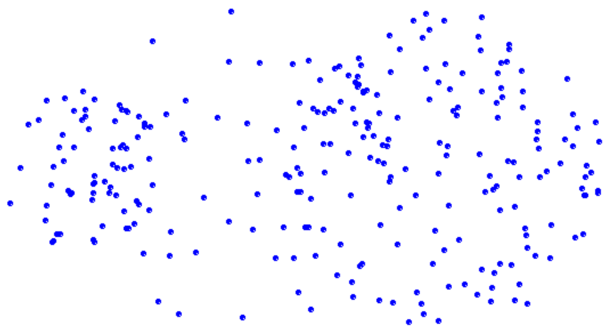


Single link distance (MIN): limitations



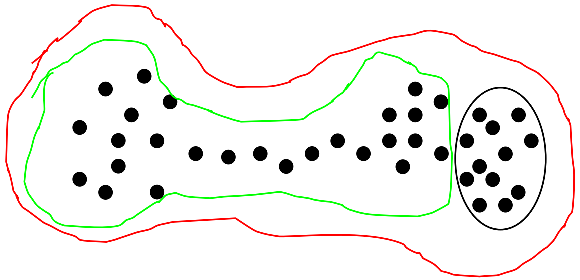


Original points

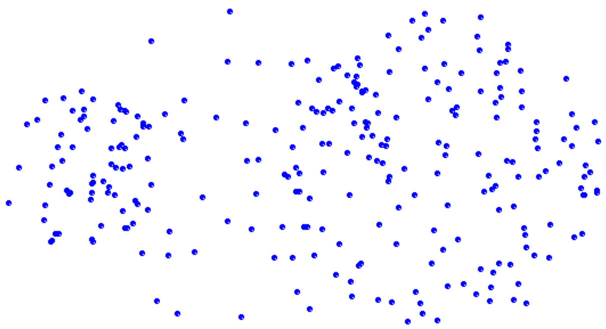


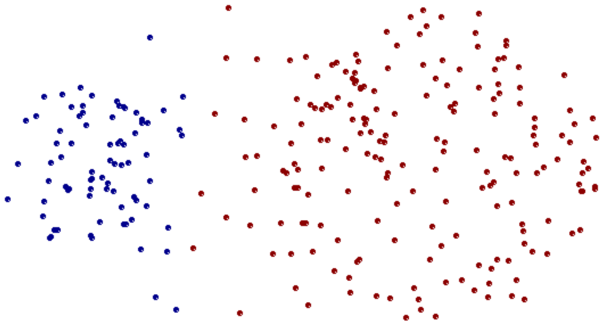
Two clusters easily merged into one cluster

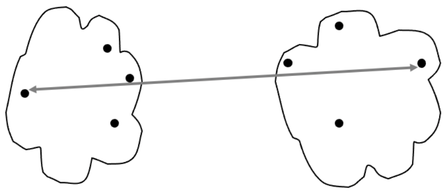
- Sensitive to noise and outliers
- Chain like clusters



Complete link distance (MAX): strengths





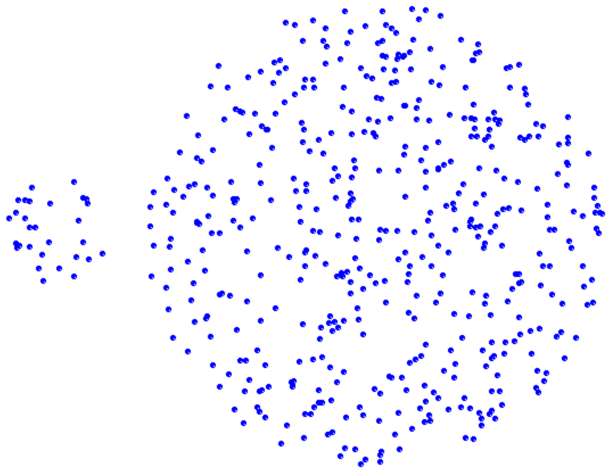


Original points

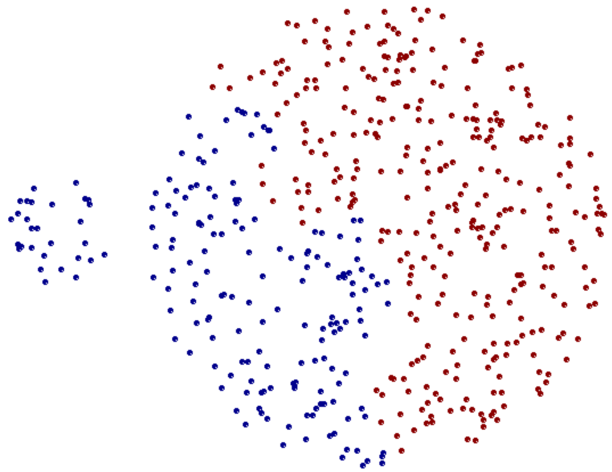
Two clusters

- Less susceptible to noise and outliers

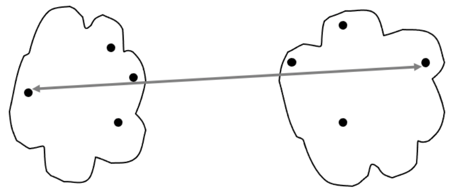
Complete link distance (MAX): limitations



Original points

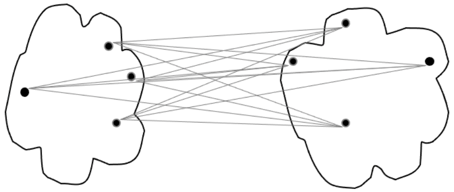


Two clusters



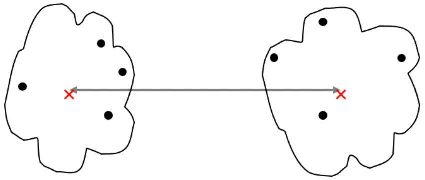
- Tends to break large clusters
- Biased towards spherical clusters

Group average: strengths and limitations



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

Centroid methods



- Difference to other measures (often considered bad): the possibility of inversions
  - Two clusters that are merged at step *k* might be more similar than the pair of clusters merged in step *k-1*
  - For the other methods, distance between clusters monotonically increases (or at worst does not increase)

Hierarchical clustering: overview

- No knowledge on the number of clusters
- Produces a hierarchy of clusters, not a flat clustering
  - A single clustering can be obtained from the dendrogram
- Merging decisions are final
  - Once a decision is made to combine two clusters, it cannot be undone
- Lack of a global objective function
  - Decisions are local, at each step
  - No objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Breaking large clusters
  - Difficulty handling different sized clusters and convex shapes
- Inefficiency, especially for large datasets