Clustering
Part 2

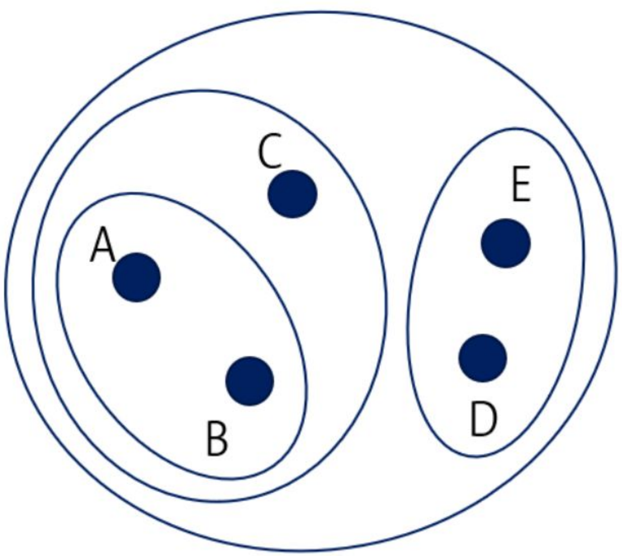
Mohammed Brahimi & Sami Belkacem

Outline

- **Overview of Clustering**
- **Major Clustering Approaches**
  - **K-means Clustering**
  - **Hierarchical Clustering**
  - **DBSCAN Clustering**
- **Cluster Evaluation**

Hierarchical Clustering

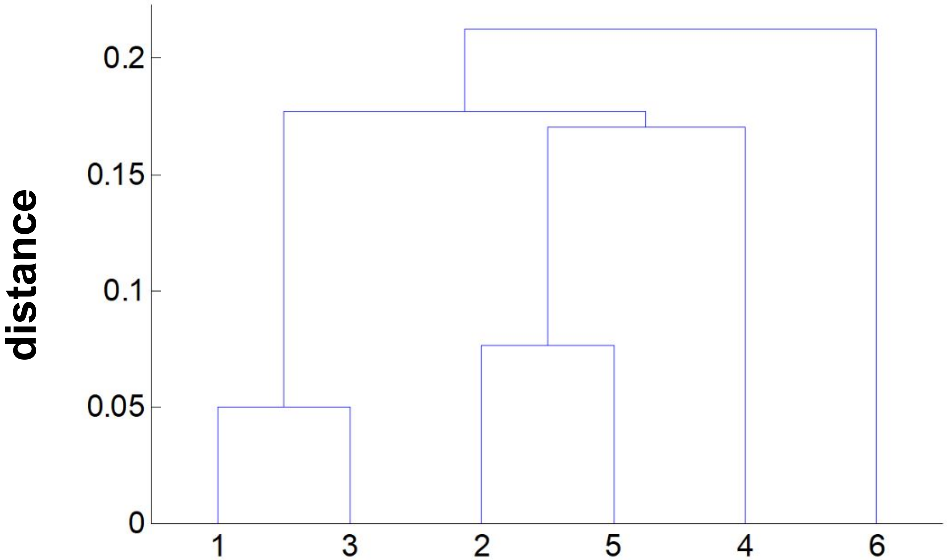
- Hierarchical Clustering produce a set of nested-clusters.
- It does not have to assume any particular number of clusters.
- It may correspond to meaningful taxonomies (e.g., biological taxonomy, animal kingdom, phylogeny reconstruction, …).



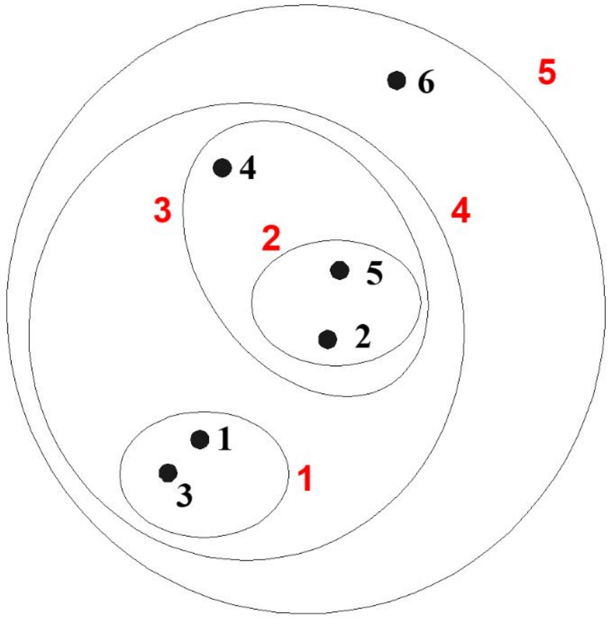
Nested clusters

Hierarchical Clustering

- The set of nested clusters can be organized as a hierarchical tree.
- The hierarchical tree of clusters is called a dendrogram, which records the sequences of merges or splits
- Different clustering of the data can be obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster



Dendrogram



5 nested clusters of 6 data points

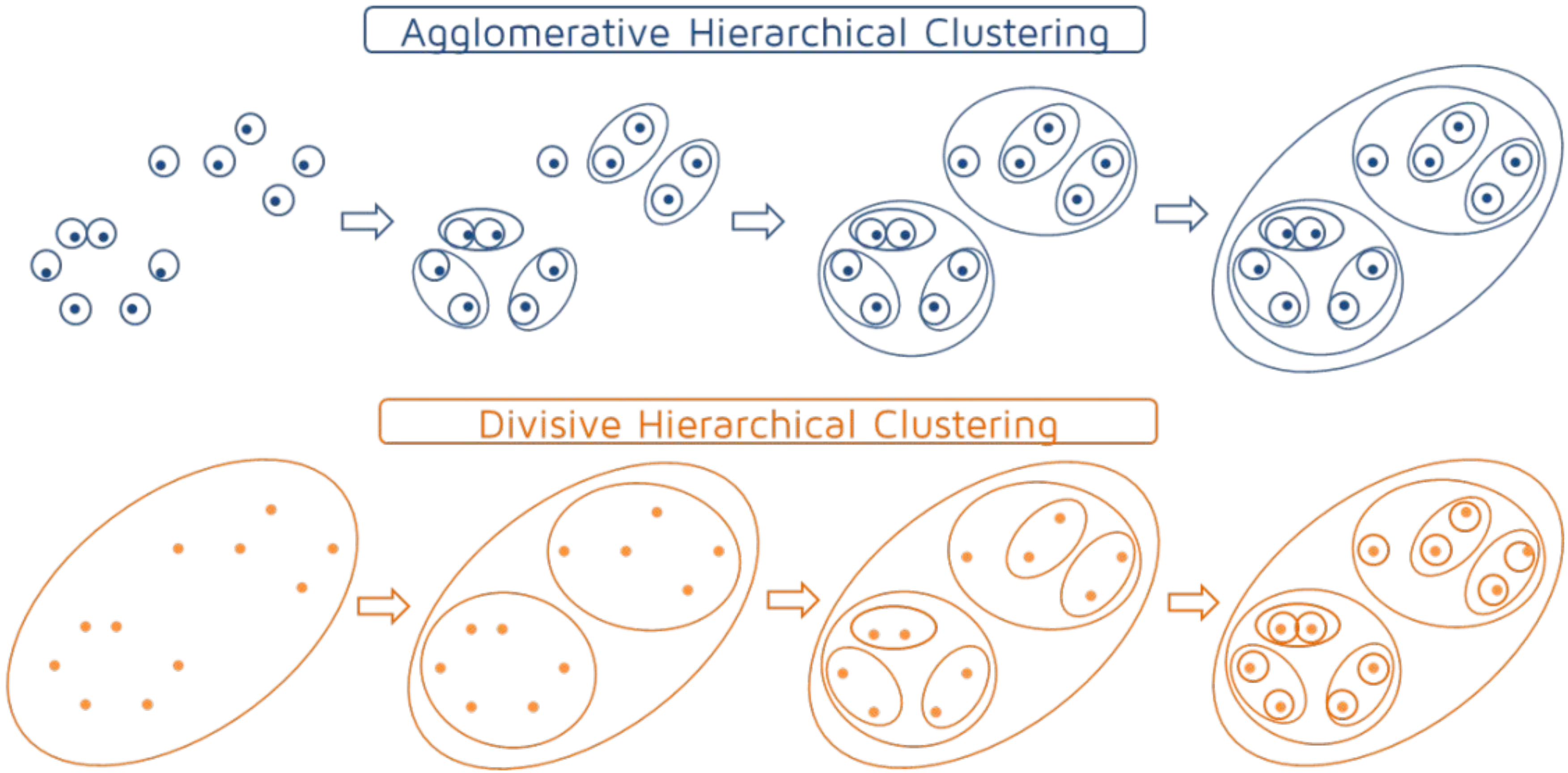
Types of Hierarchical Clustering

- **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
  - **Popular algorithm**: AGNES (Agglomerative Nesting)

- **Divisive:**
  - Start with one, all-inclusive cluster
  - At each step, split the least cohesive clusters until each cluster contains an individual point (or there are ***k*** clusters)
  - **Popular algorithm**: DIANA (Divisive Analysis)

- Hierarchical algorithms use a proximity matrix (similarity or distance)
  - Merge or split one cluster at a time

Agglomerative vs Divisive



Agglomerative Clustering Algorithm

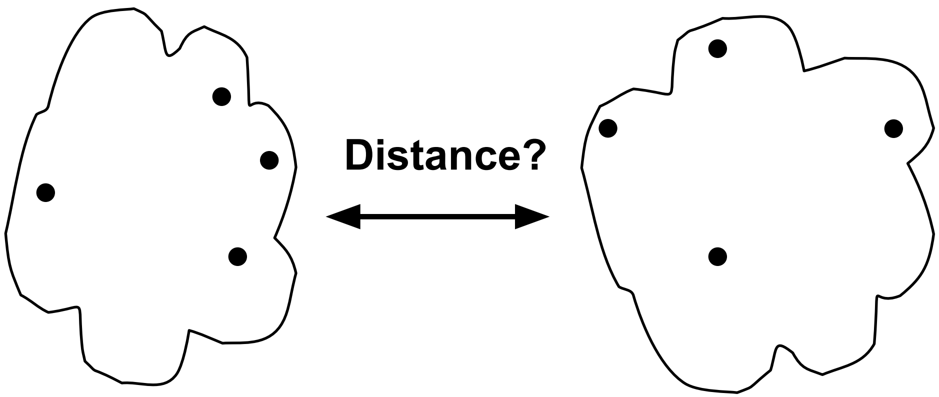
- **Key Idea:** Successively merge the closest clusters
- **Basic algorithm:**

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 (or ***k*** clusters left)

- Key operation is the computation of the proximity of two clusters:
  - Different approaches to defining the **distance between clusters** distinguish the different algorithms (Min, Max, etc.)

How to measure the distance between two clusters?

How to Define Inter-Cluster Distance

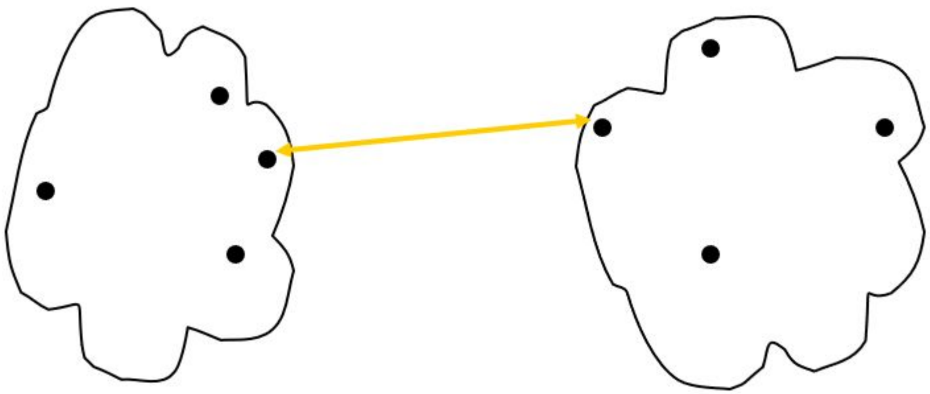


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

1. MIN
2. MAX
3. Group Average
4. Distance Between Centroids

How to Define Inter-Cluster Distance

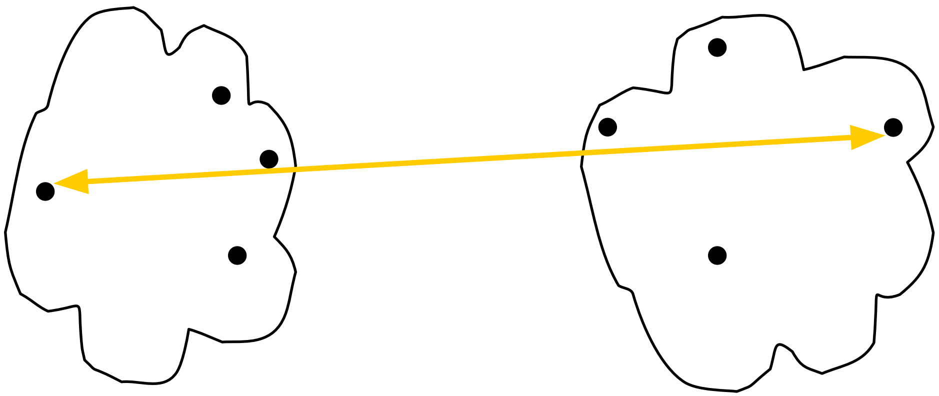


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

1. MIN
2. MAX
3. Group Average
4. Distance Between Centroids

How to Define Inter-Cluster Distance

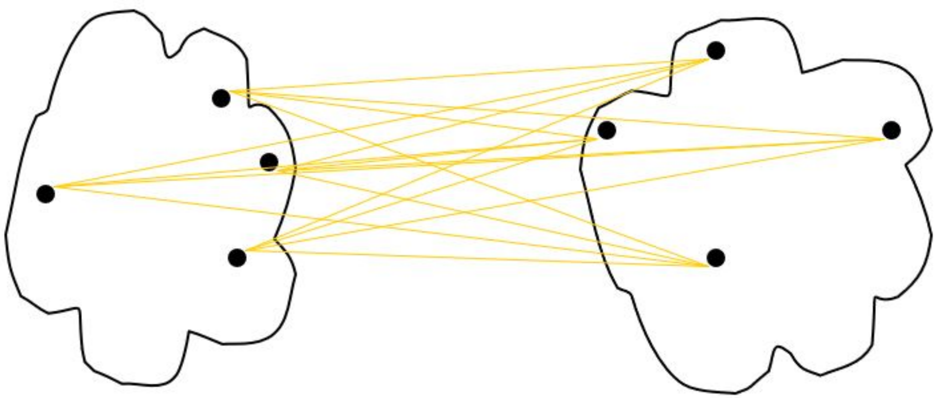


1. MIN
2. MAX
3. Group Average
4. Distance Between Centroids

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

Proximity Matrix

How to Define Inter-Cluster Distance



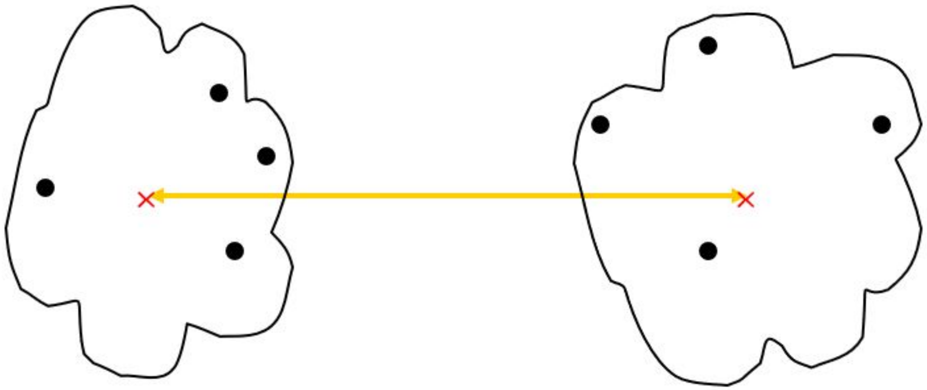
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

1. MIN
2. MAX
3. **Group Average**
4. Distance Between Centroids

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

How to Define Inter-Cluster Distance

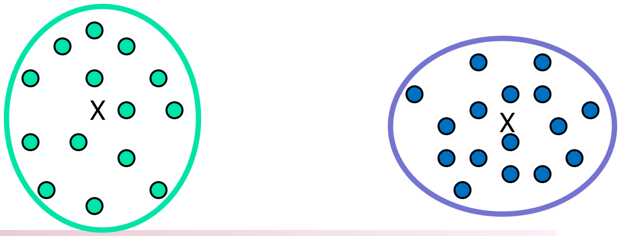


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

1. MIN
2. MAX
3. Group Average
4. Distance Between Centroids

Inter-Cluster Distance

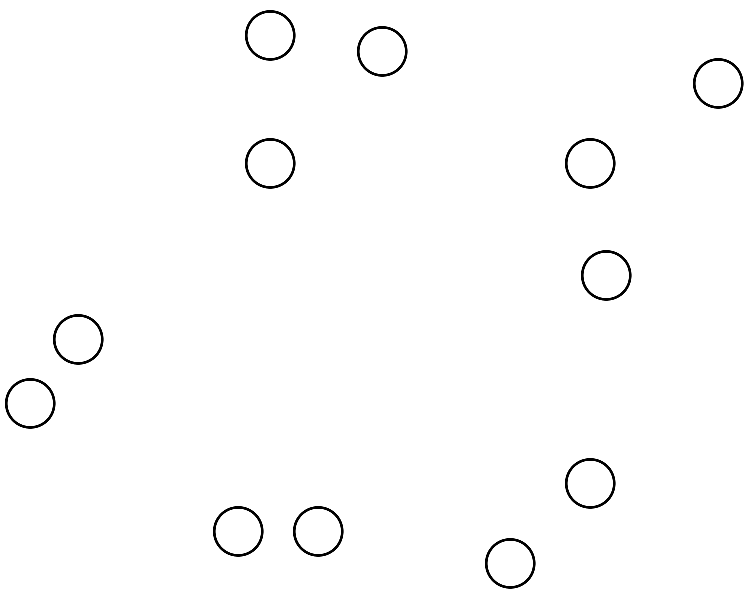


1. **Min (Single link):** smallest distance between an element in one cluster and an element in the other, $dist(K_i, K_j) = min(t_{ip}, t_{jq})$
2. **Max (Complete link):** largest distance between an element in one cluster and an element in the other, $dist(K_i, K_j) = max(t_{ip}, t_{jq})$
3. **Group Average:** avg distance between an element in one cluster and an element in the other, $dist(K_i, K_j) = avg(t_{ip}, t_{jq})$
4. **Centroid:** distance between the centroids of two clusters, $dist(K_i, K_j) = dist(C_i, C_j)$

Now that we've understood how to measure the distance between two clusters, let's go back to the steps of the Agglomerative Clustering algorithm.

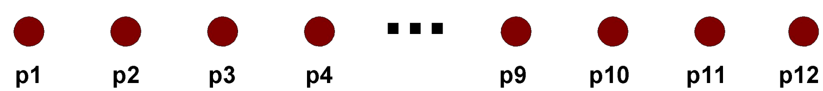
Agglomerative clustering: Steps 1 and 2

- Start with clusters of individual points and a proximity matrix



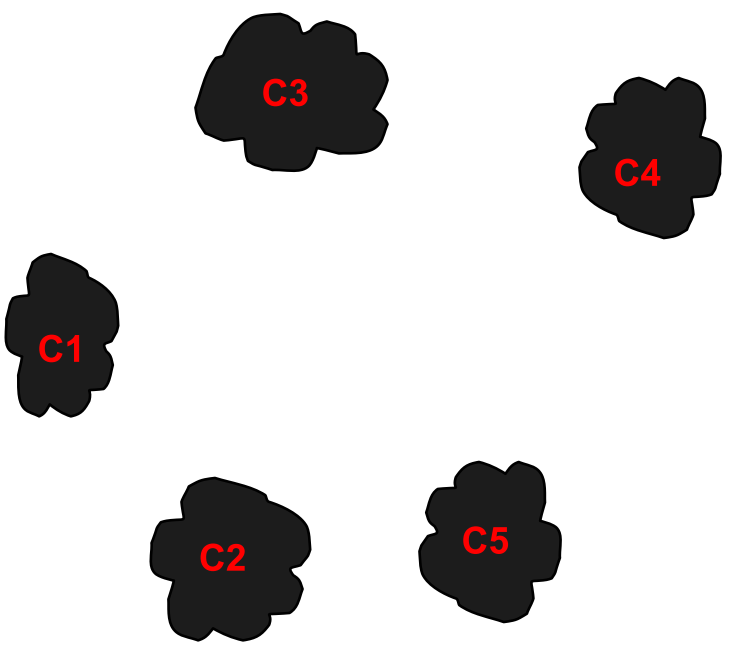
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix



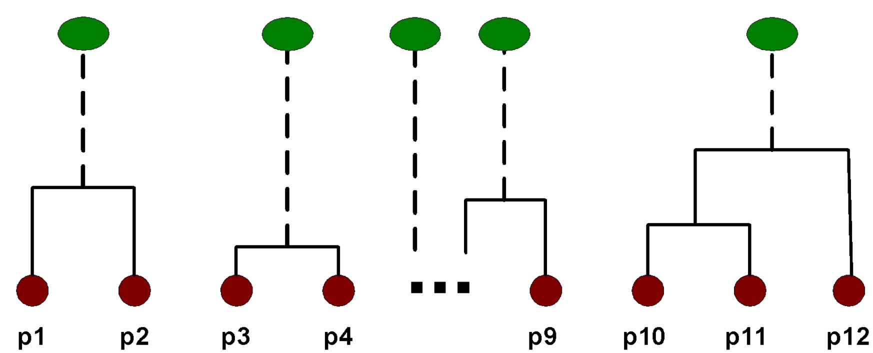
Intermediate Situation

- After some merging steps, we have some clusters



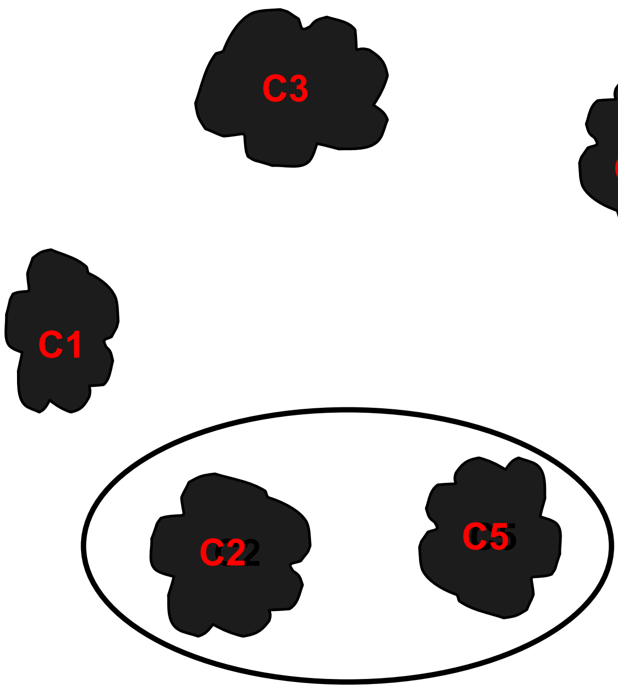
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

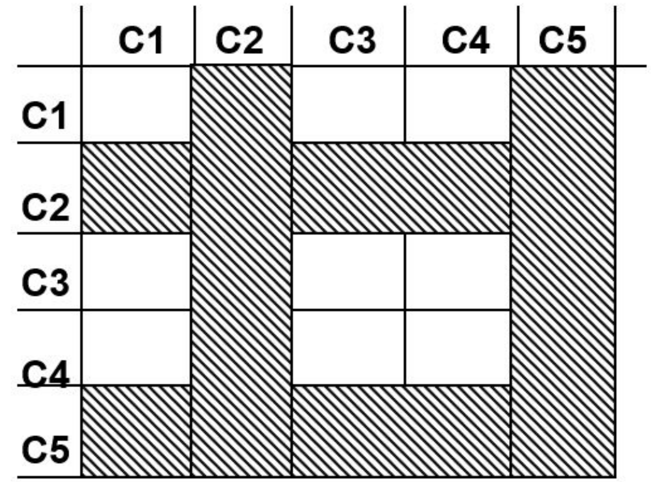
Proximity Matrix



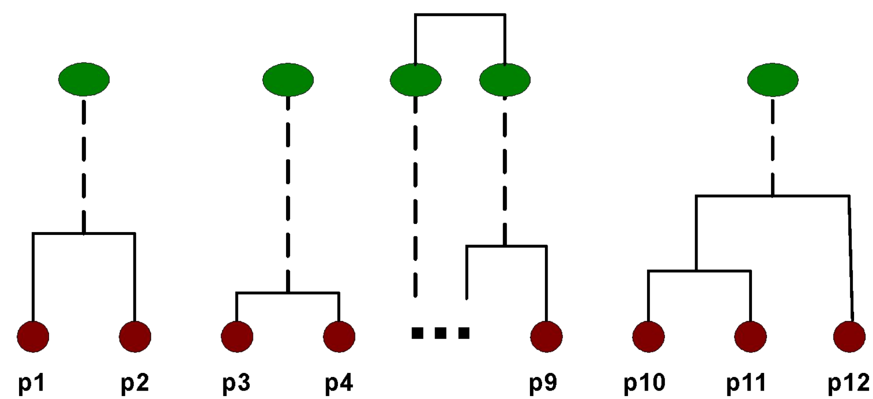
Step 4

- **Merge** the two closest clusters (C2 and C5) and **update** the matrix



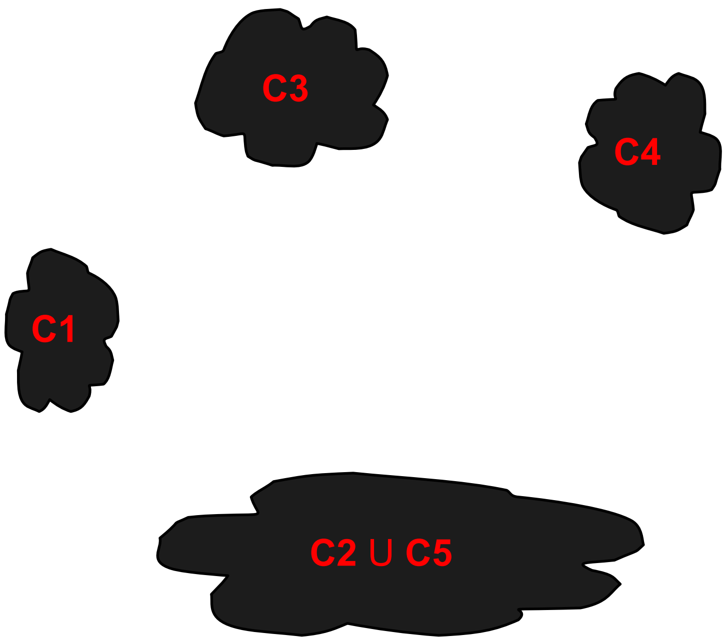


Proximity Matrix



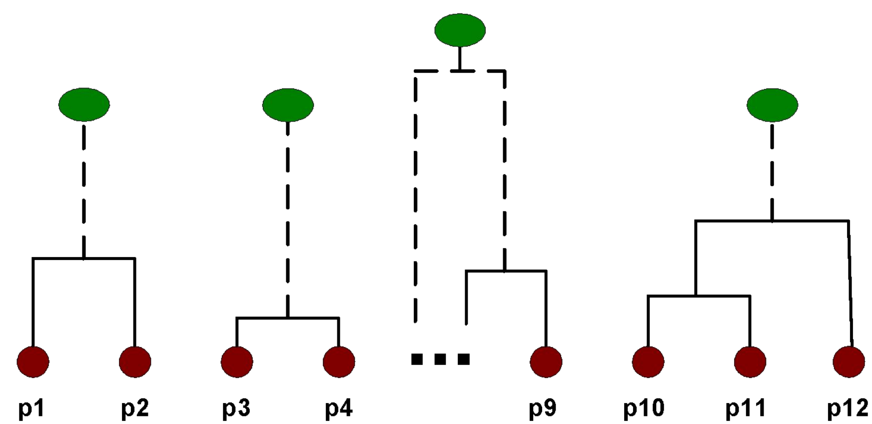
Step 5

- Now, the question is “**how do we update the proximity matrix**?”



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

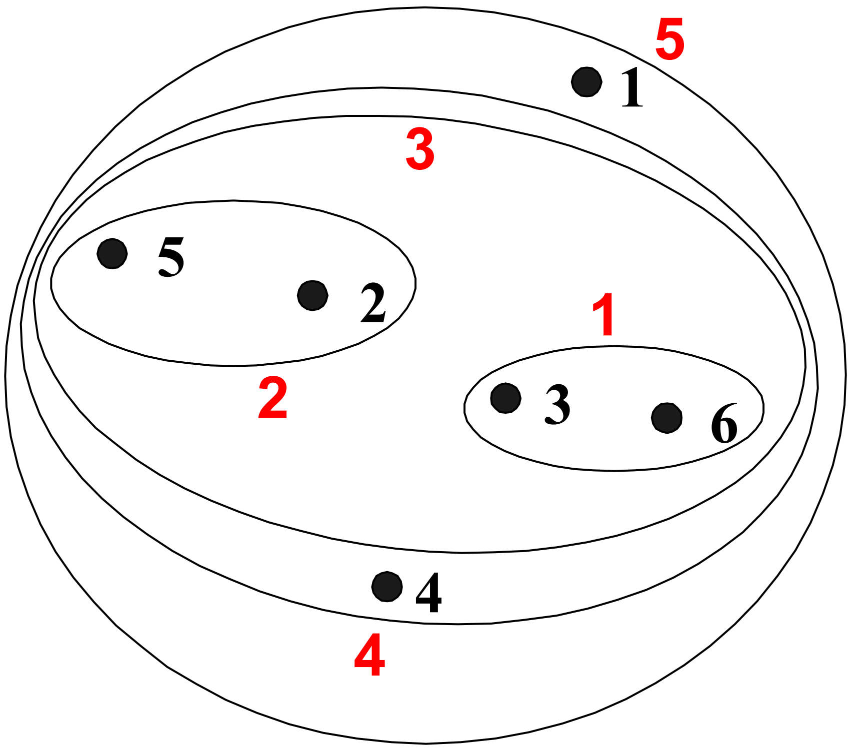
Proximity Matrix

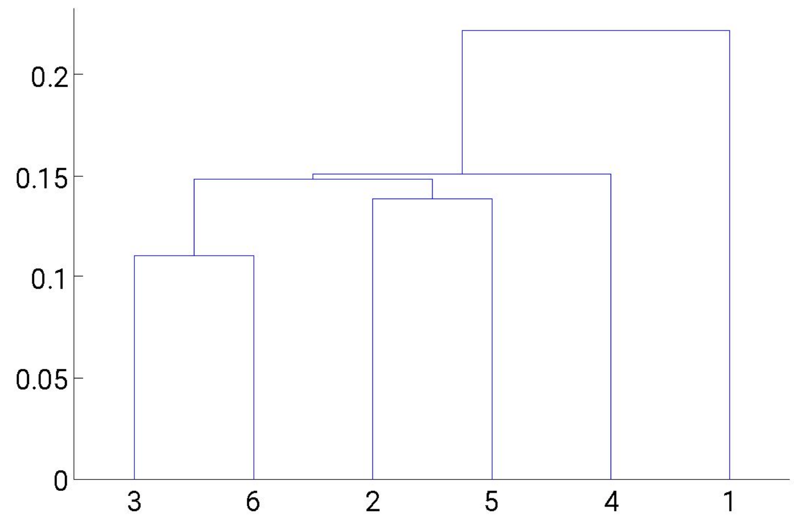


Answer: we update the proximity matrix using the different approaches to defining the distance between clusters (Min, Max, etc.)

Note: to compute the distance between an individual data point and a cluster, we consider that data point itself as a cluster

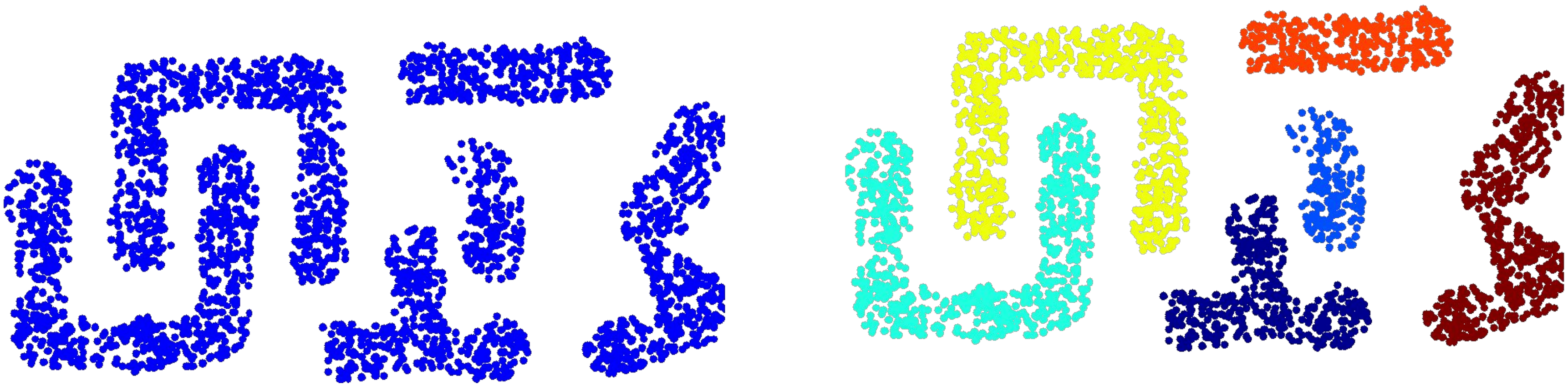
Hierarchical Clustering: MIN





Nested Clusters **Dendrogram**

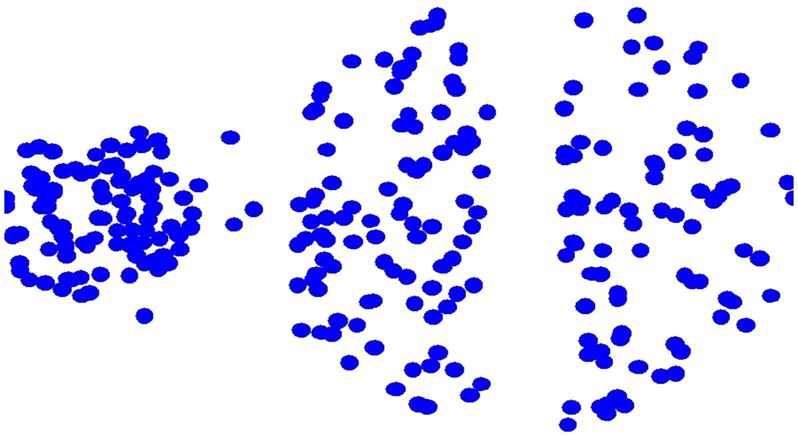
Strength of MIN



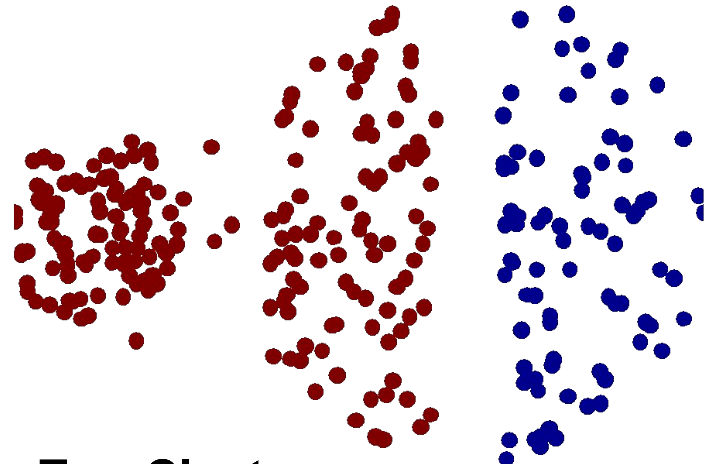
Original Points **Six Clusters**

- It detects clusters of any shape by focusing only on the nearest points between clusters, **ignoring overall shape**.
- Captures irregularly shaped clusters effectively without assuming specific geometrical forms like **elliptical shapes**.

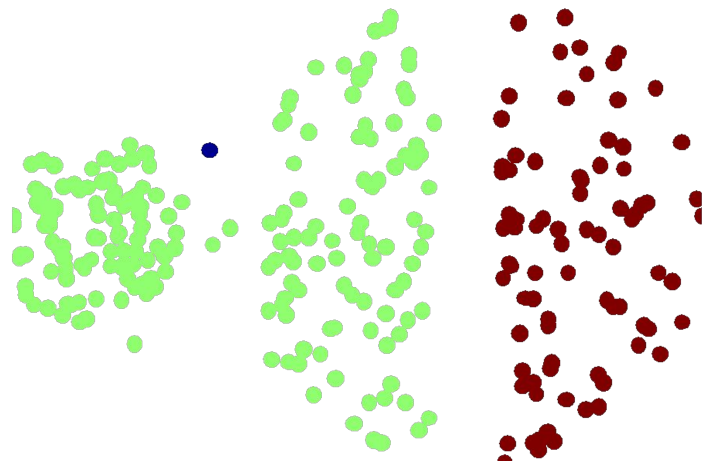
Limitations of MIN



Original Points



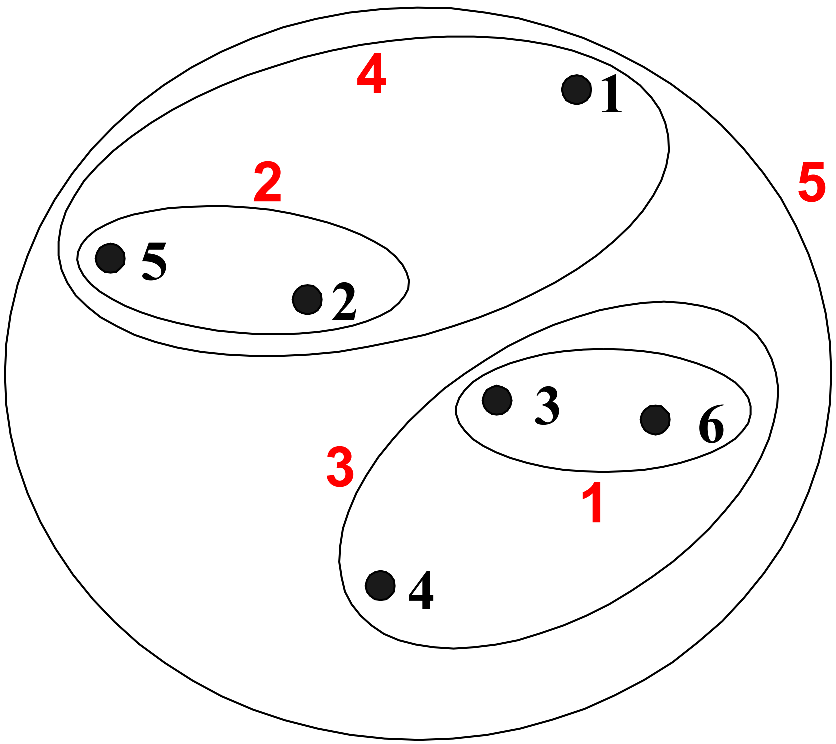
Two Clusters



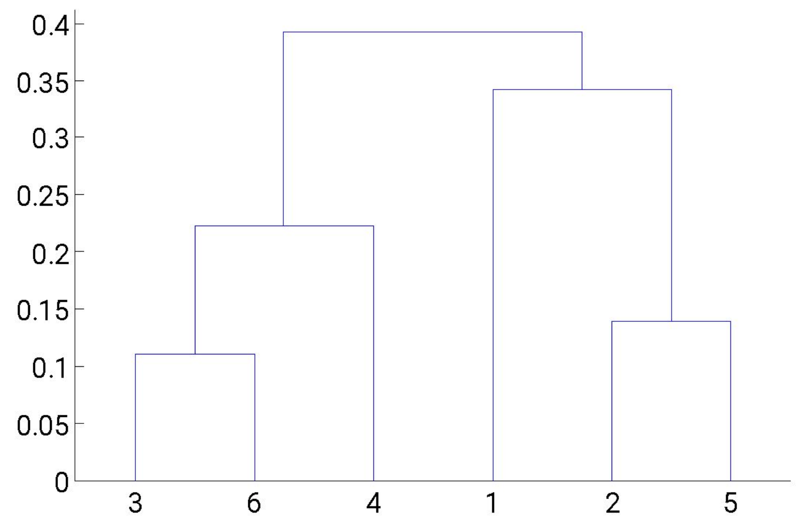
Three Clusters

- **Chaining effect:** Merges two clusters due to closely paired points, leading to a chain of combined clusters.
- **Noise sensitivity:** A single point can alter the cluster's shape.

Hierarchical Clustering: MAX

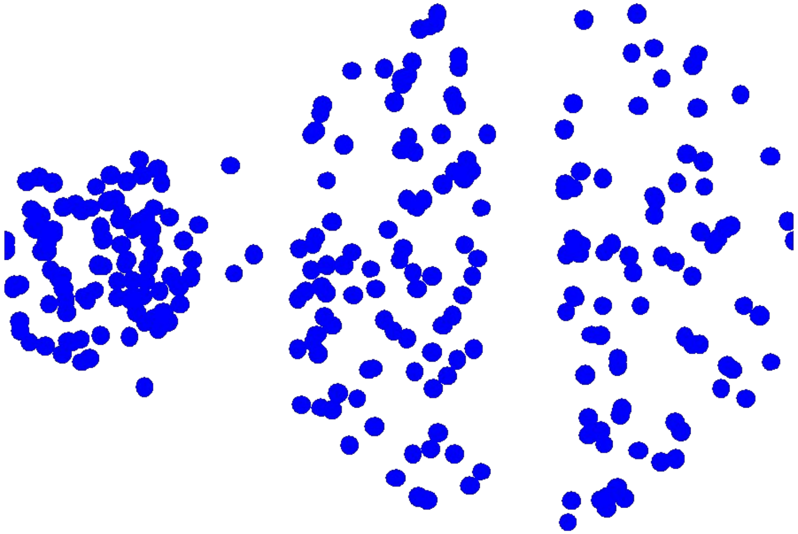


Nested Clusters

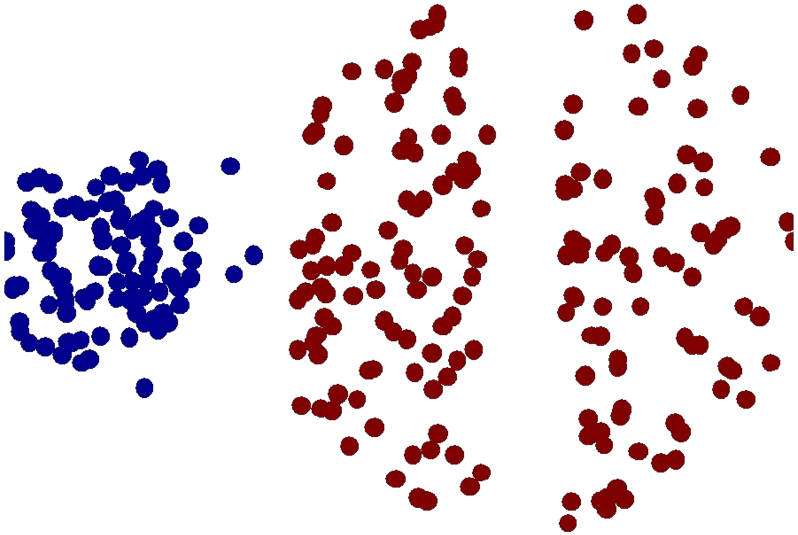


Dendrogram

Strength of MAX



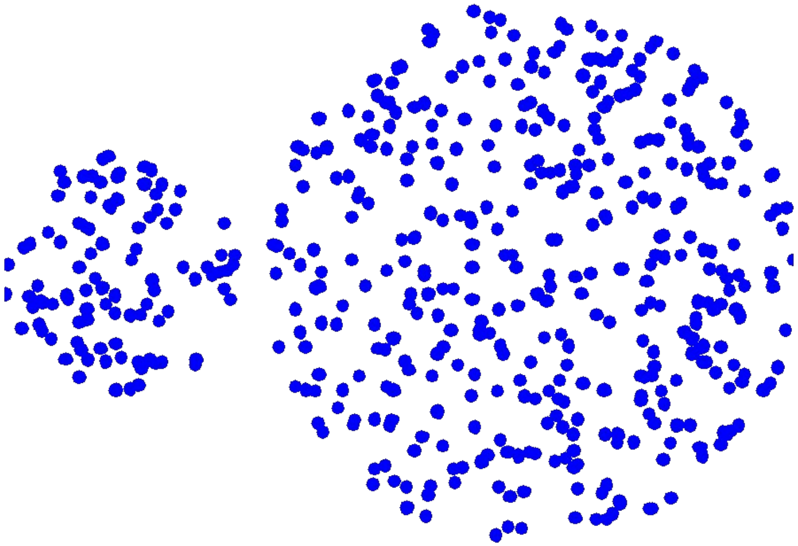
Original Points



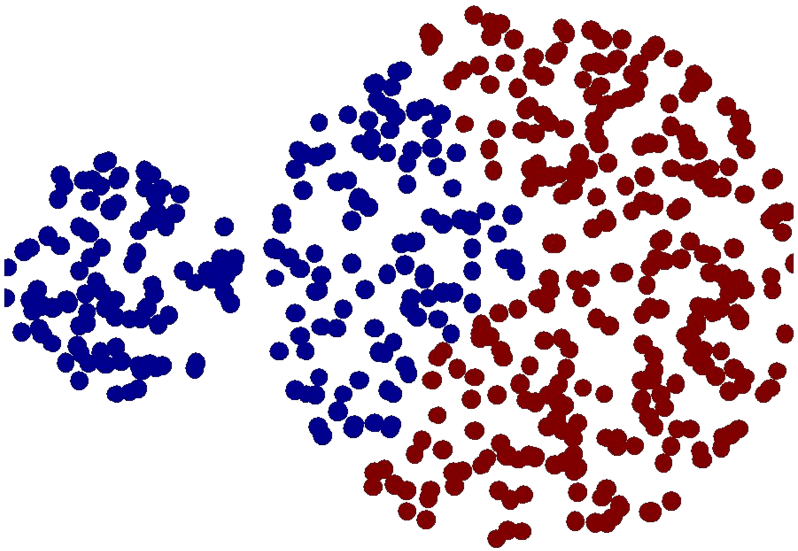
Two Clusters

- **Robustness to Noise:** Less affected by noise because it looks at the farthest points between clusters, forming compact groups less likely to be influenced by outliers.

Limitations of MAX



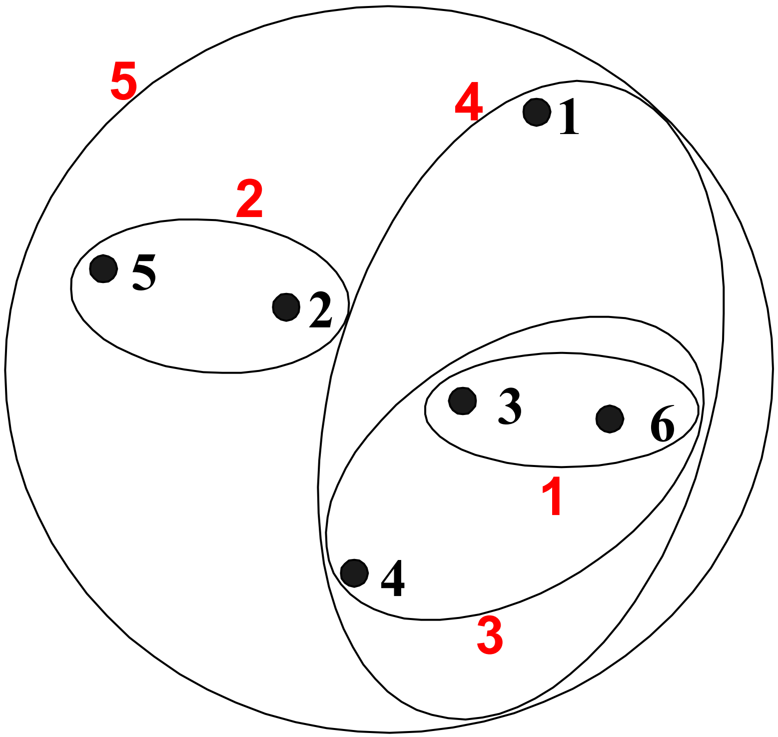
Original Points

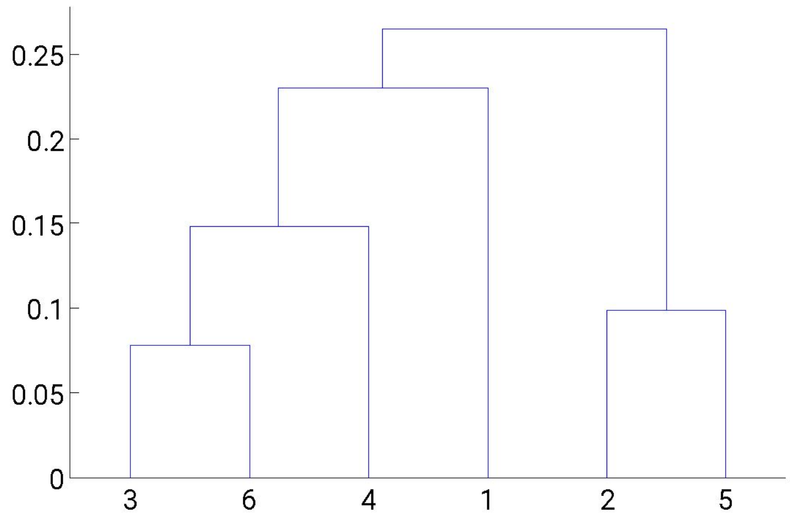


Two Clusters

- Tends to break large clusters into smaller, more distinct ones.
- Biased towards globular clusters

Hierarchical Clustering: Group Average





Nested Clusters

Dendrogram

Hierarchical Clustering: Group Average

- Compromise between Single and Complete Link

- **Strengths**
  - Averaging reduces the influence of noisy data points

- **Limitations**
  - Biased towards globular clusters because the average distance favors clusters with compact, closely located points

Hierarchical Clustering: Space and Time Complexity

- N is the number of data points or objects.
- **Space:** $O(N^2)$
  - $O(N^2)$ because the proximity matrix has N^2 entries for distances between N points.
- **Time:** $O(N^3)$
  - Find the min distance of the matrix $O(N^2)$ * N iterations $\Rightarrow O(N^3)$
  - Complexity can be reduced to $O(N^2 \log(N))$
    - Accelerate finding the minimum using a heap ….

Strength 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, …)

Weakness of Hierarchical Clustering

- Once a decision is made to combine two clusters, it cannot be undone
- Do not scale well: time complexity of $O(n^3)$, ***n*** is the number of objects
- No global objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise
  - Difficulty handling clusters of different sizes and non-globular shapes
  - Breaking large clusters

Improvements: Integration of hierarchical and distance-based clustering

- Example of Algorithms: BIRCH, CHAMELEON