Outline

- K-Means
- Hierarchical Clustering
- Model-based Clustering (GMM and Expectation Maximization)
- Evaluation of Clustering Algorithms

Challenge

How many clusters?

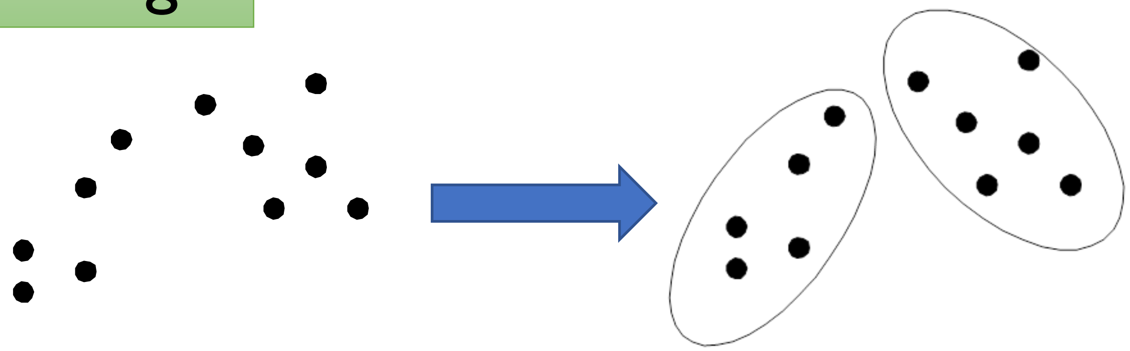
Six Clusters

Two Clusters

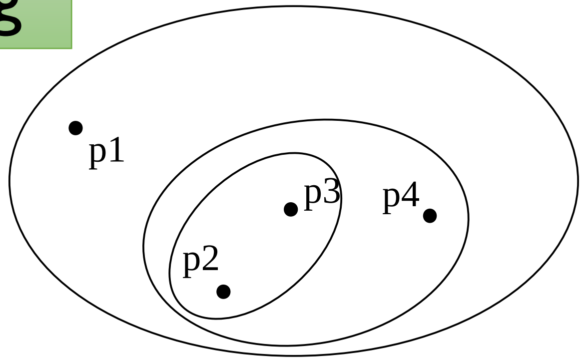
Four Clusters

Types of Clustering Methods

Partitional Clustering

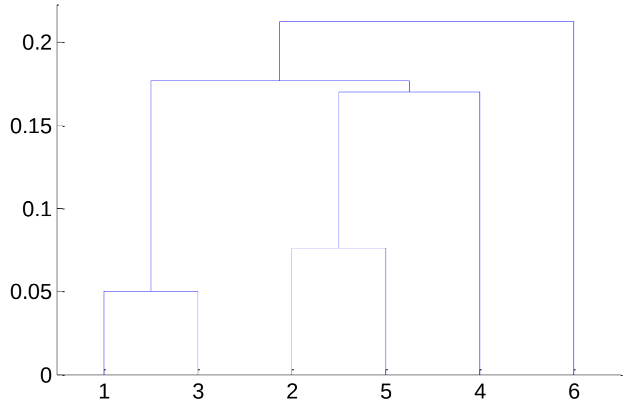


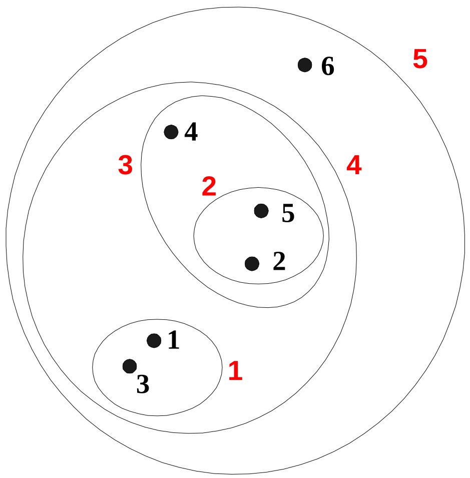
Hierarchical Clustering



Hierarchical Clustering

- 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





Strengths

- 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 Clustering

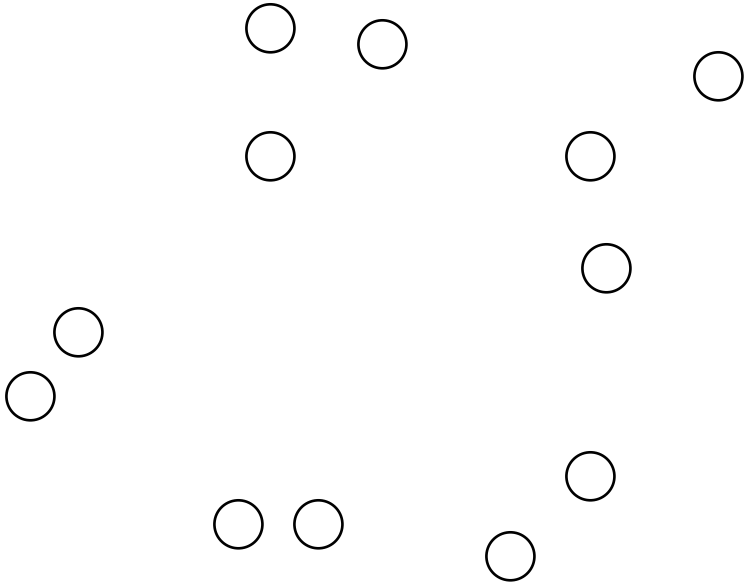
- Two main types of hierarchical clustering
  - **Agglomerative:**
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - **Divisive:**
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one 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
     1. Merge the two closest clusters
     2. Update the proximity matrix
  4. Until only a single cluster remains
- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

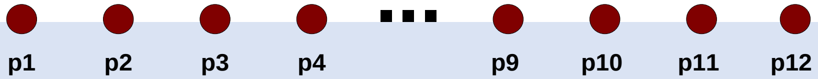
Methodology

- Start with clusters of individual points and a proximity matrix



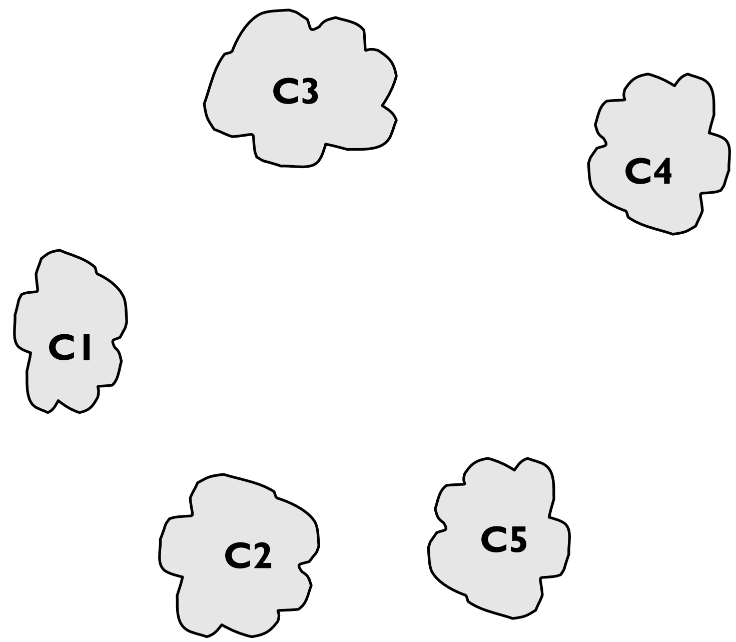
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix



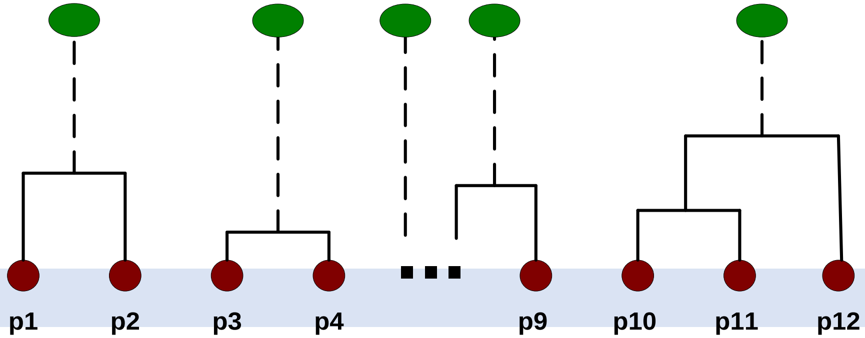
Methodology

- After some merging steps, we have some clusters



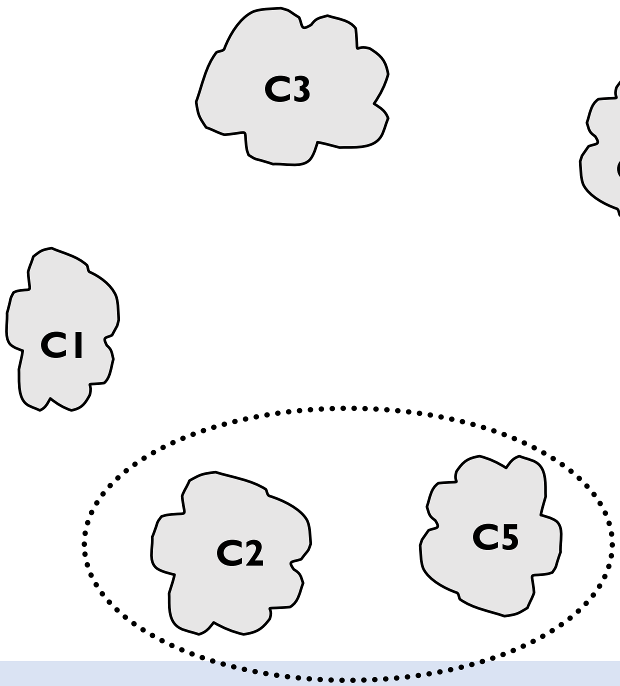
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

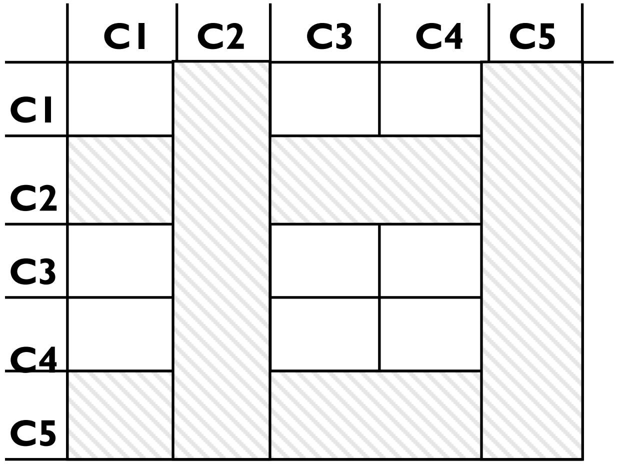
Proximity Matrix



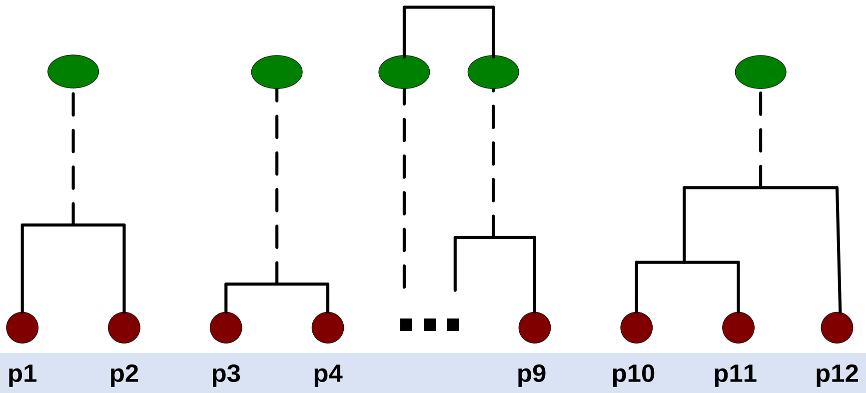
Methodology

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.



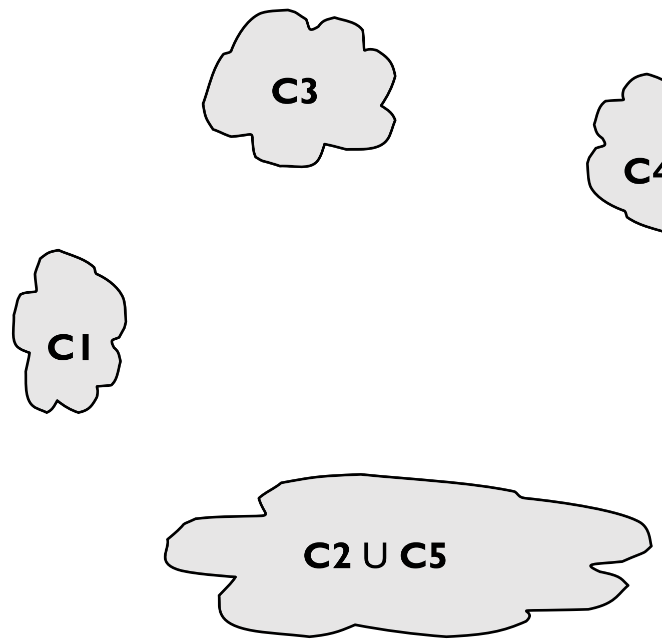


Proximity Matrix



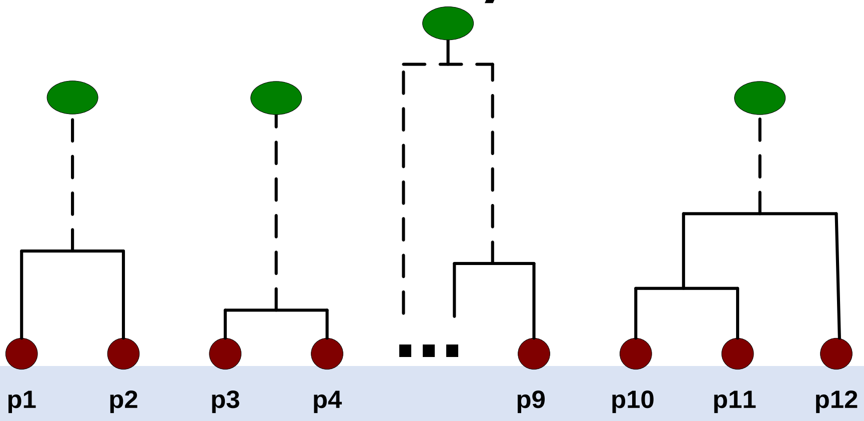
Methodology

- The question is "How do we update the proximity matrix?"

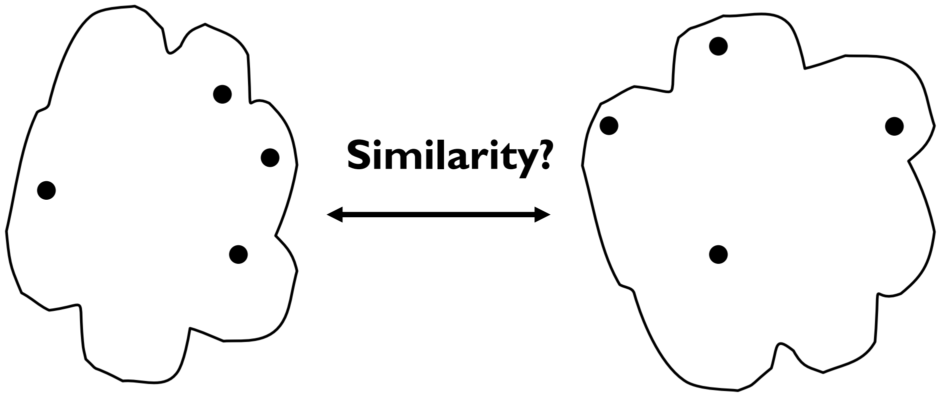


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



Defining Inter-cluster Similarity

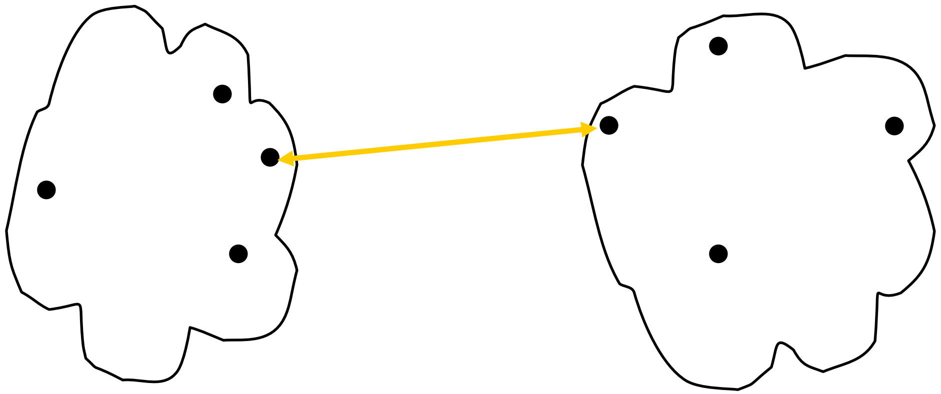


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

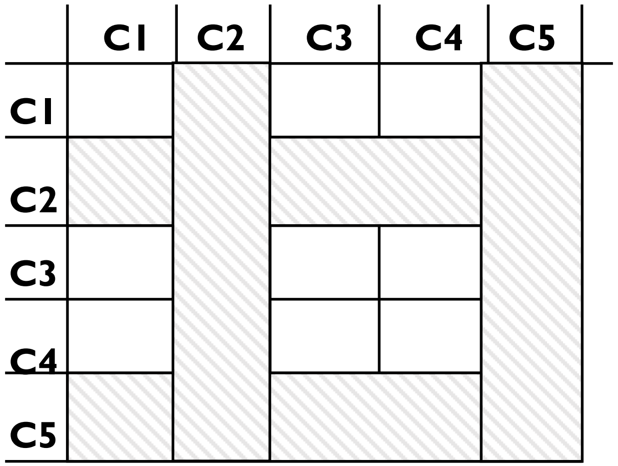
- MIN (Single-link)
- MAX (Complete-link)
- Group Average (Average-link)
- Distance Between Centroids

Defining Inter-cluster Similarity



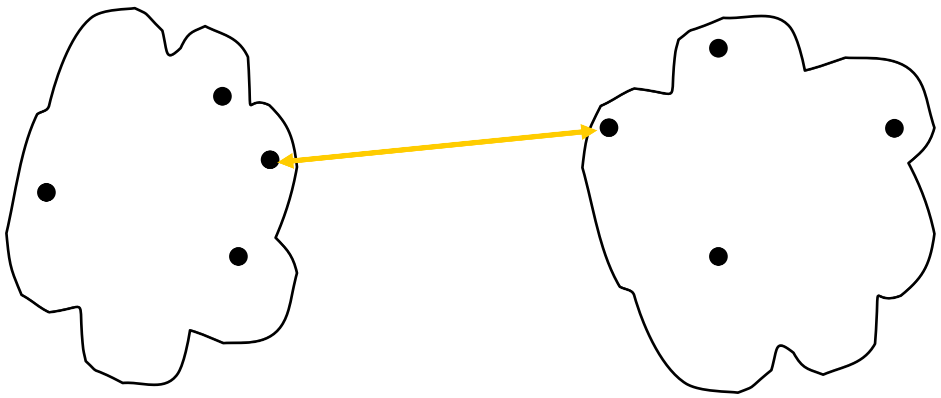
Sim(C1,C2) = Min Sim(Pi,Pj) such that Pi ∈ C1 & Pj ∈ C2

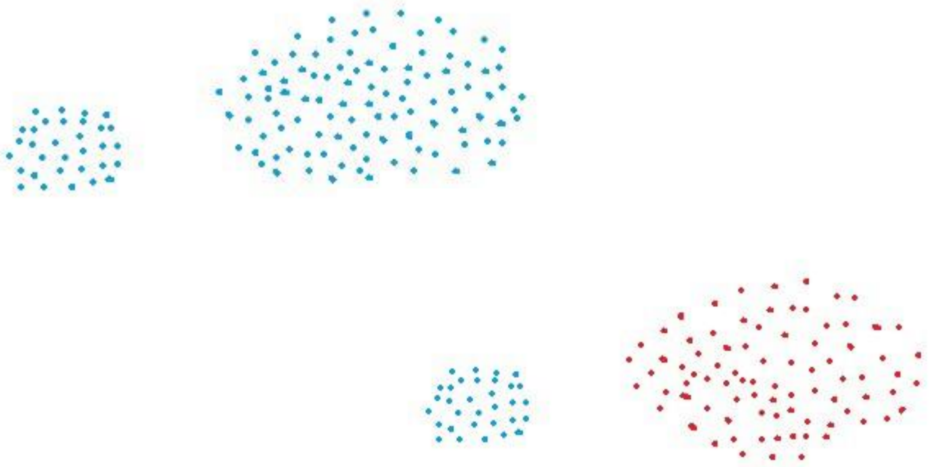
- **MIN (Single-link)**
- MAX (Complete-link)
- Group Average (Average-link)
- Distance Between Centroids



Proximity Matrix

Defining Inter-cluster Similarity

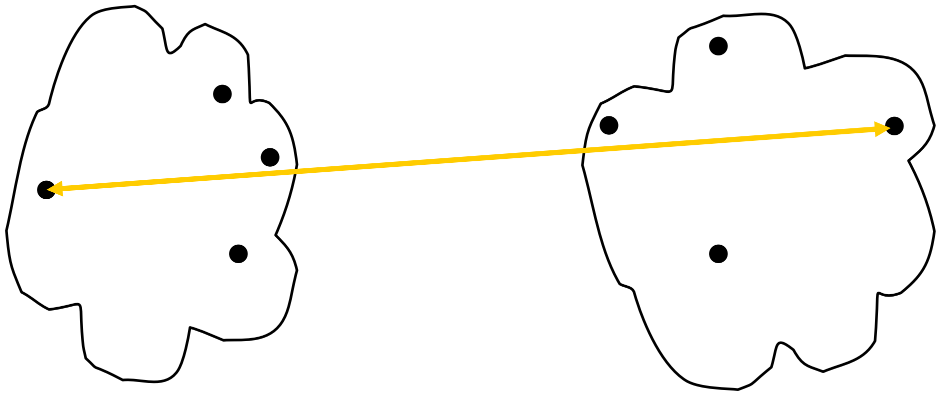




Sim(C1,C2) = Min Sim(Pi,Pj) such that Pi ∈ C1 & Pj ∈ C2

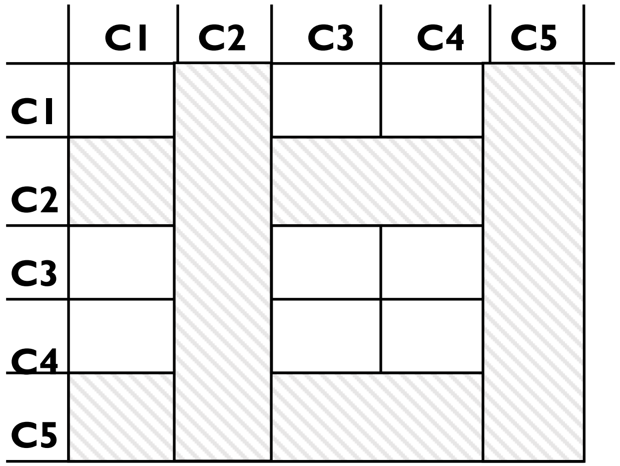
- **MIN (Single-link)**
- MAX (Complete-link)
- Group Average (Average-link)
- Distance Between Centroids

Defining Inter-cluster Similarity



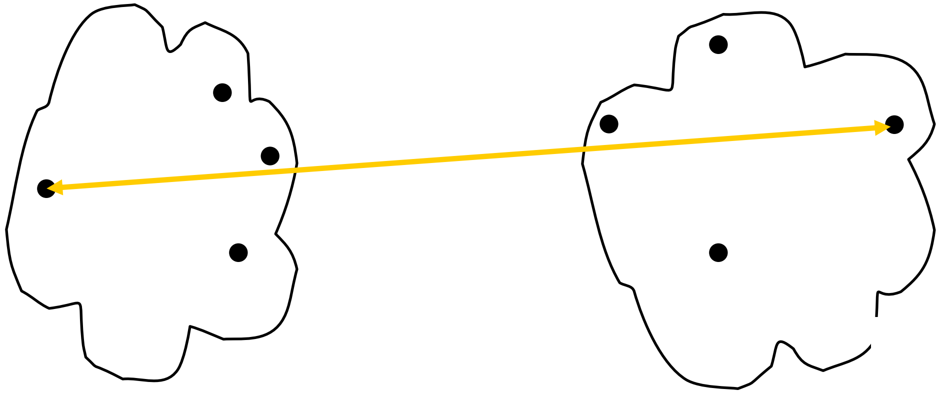
Sim(C1,C2) = Max Sim(Pi,Pj) such that Pi ∈ C1 & Pj ∈ C2

- MIN (Single-link)
- **MAX (Complete-link)**
- Group Average (Average-link)
- Distance Between Centroids

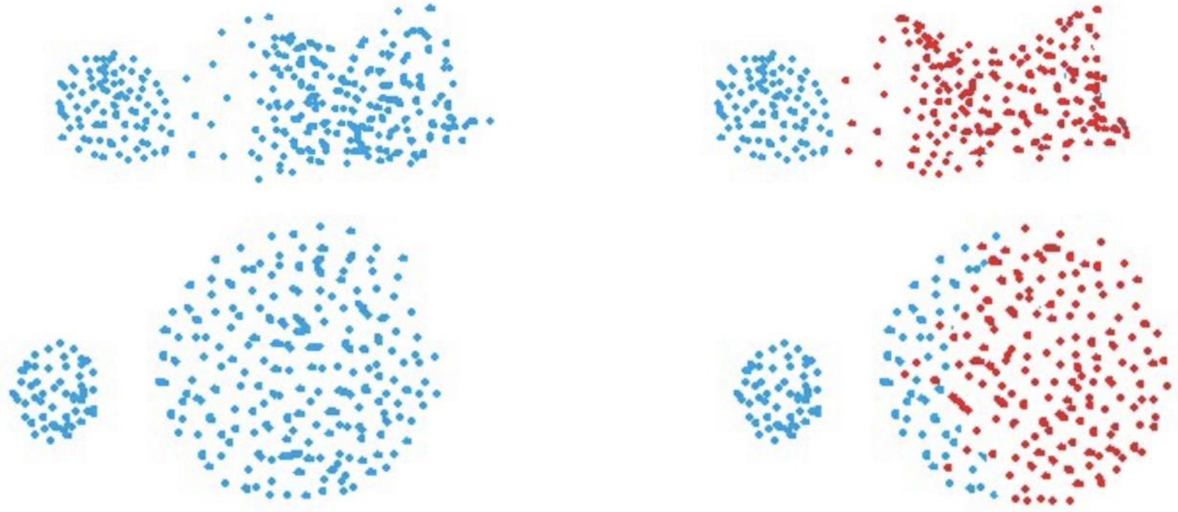


Proximity Matrix

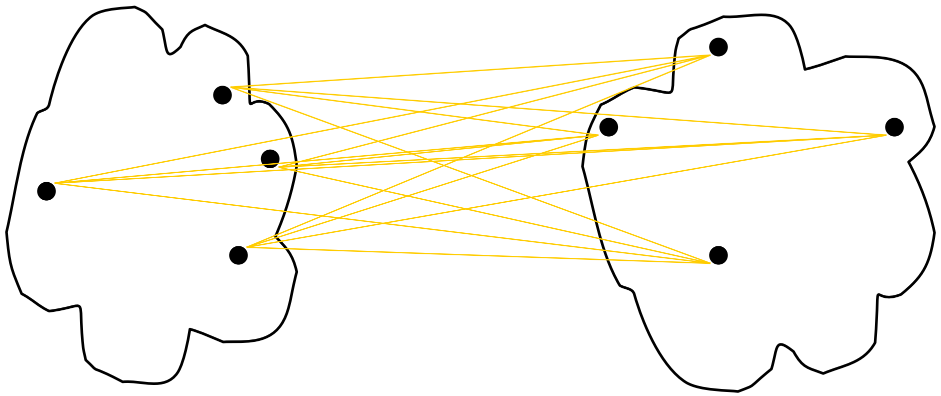
Defining Inter-cluster Similarity



- MIN (Single-link)
- **MAX (Complete-link)**
- Group Average (Average-link)
- Distance Between Centroids

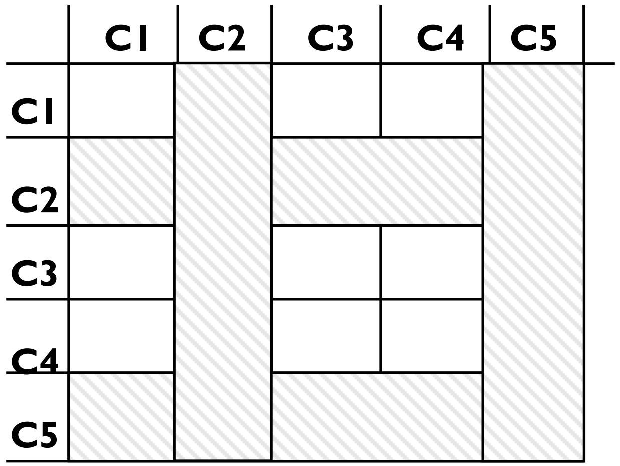


Defining Inter-cluster Similarity



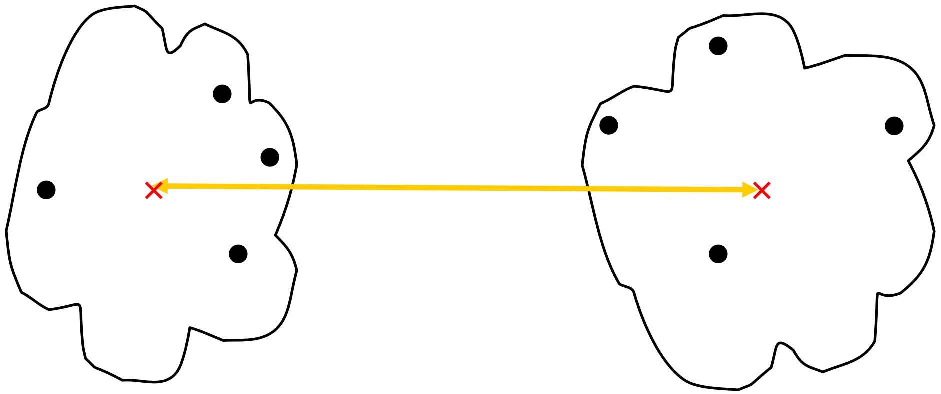
sim(C1,C2) = $\sum$ sim(Pi, Pj)/|C1|*|C2|
where, Pi $\in$ C1 & Pj $\in$ C2

- MIN (Single-link)
- MAX (Complete-link)
- **Group Average (Average-link)**
- Distance Between Centroids



Proximity Matrix

Defining Inter-cluster Similarity



- MIN (Single-link)
- MAX (Complete-link)
- Group Average (Average-link)
- **Distance Between Centroids**

- **Ward's Method:** This approach of calculating the similarity between two clusters is exactly the same as Group Average except that Ward's method calculates the sum of the square of the distances Pi and PJ.

Mathematically this can be written as,

$$sim(C1,C2) = \sum (dist(Pi, Pj))^2 / |C1| * |C2|$$

Hierarchical Clustering: Limitations

- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers (MIN)
  - Difficulty handling different sized clusters and non-convex shapes (Group average, MAX)
  - Breaking large clusters (MAX)
  - Space complexity : O(n^2), O(n^3)

Outline

- Evaluation of Clustering Algorithms

Cluster Validity

- External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
  - Entropy
- Internal Index: Used to measure the goodness of a clustering structure *without* respect to external information.
  - Sum of Squared Error (SSE)
- Relative Index: Used to compare two different clusterings or clusters.
  - Often an external or internal index is used for this function, e.g., SSE or entropy

Internal Measures

- Cluster Cohesion: Measures how closely related are objects in a cluster
  - Example: SSE
- Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters
- Example: Squared Error
  - Cohesion is measured by the within cluster sum of squares (SSE)

$$WSS = \sum_{i} \sum_{x \in C_i} (x - m_i)^2$$

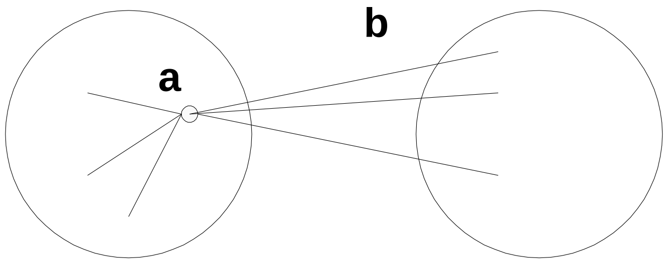
  - Separation is measured by the between cluster sum of squares

$$BSS = \sum_{i} |C_i| (m - m_i)^2$$

    - Where $|C_i|$ is the size of cluster i

Internal Measures: Silhouette Coefficient

- Combines ideas of both cohesion and separation, but for individual points as well as clusters
- For an individual point *i*
  - Calculate ***a*** = average distance of *i* to the points in its cluster
  - Calculate ***b*** = min (average distance of *i* to points in another cluster)
  - The silhouette coefficient for a point is then given by
    $s = 1 – a/b$ if $a < b$, (or $s = b/a - 1$ if $a \geq b$, not the usual case)
  - Typically between 0 and 1.
  - The closer to 1 the better.



External Indices: Entropy and Purity

Table K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

entropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij}, the 'probability' that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^{L} p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{i=1}^{K} \frac{m_i}{m} e_j$, where m_j is the size of cluster j, K is the number of clusters, and m is the total number of data points.

purity Using the terminology derived for entropy, the purity of cluster j, is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{i=1}^{K} \frac{m_i}{m} purity_j$.