CLARA (Clustering Large Applications)

- CLARA (Kaufmann and Rousseeuw, 1990)
- It draws multiple samples of the dataset, applies PAM on each sample, and gives the best clustering as the output.
- Strength: deals with larger datasets than PAM
- Weakness:
  - Efficiency depends on the sample size
  - A good clustering based on samples will not necessarily represent a good clustering of the whole dataset if the sample is biased

What is the right number of clusters 1/2

- The number of clusters *k* is required as input by the partitioning algorithms. Choosing the right *k* is challenging.
- Silhouette coefficient (Kaufman & Rousseeuw 1990)
  - Let *a(o)* the distance of an object *o* to the representative of its cluster and *b(o)* the distance to the representatives of its "second best" cluster
  - Silhouette *s*(*o*) of an object *o:*

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}$$

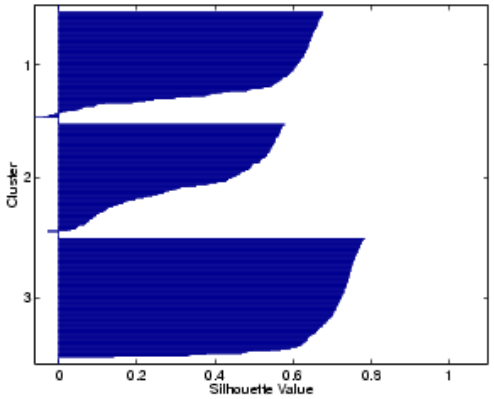
-1 ≤ s(o) ≤ +1

s(o) ~ -1 / 0 / +1 : bad / indifferent / good assignment

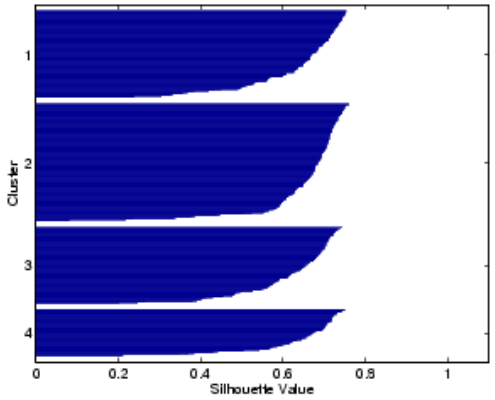
  - s(o)~1→ *a*(*o*) < < *b*(*o*). Small *a*(*o*) means it is well matched to its own cluster. Large b(o) means is badly matched to its neighbouring cluster.
  - s(o)~-1→ the neighbor cluster seems more appropriate
  - s(o)~0→ in the border between two natural clusters

What is the right number of clusters 2/2

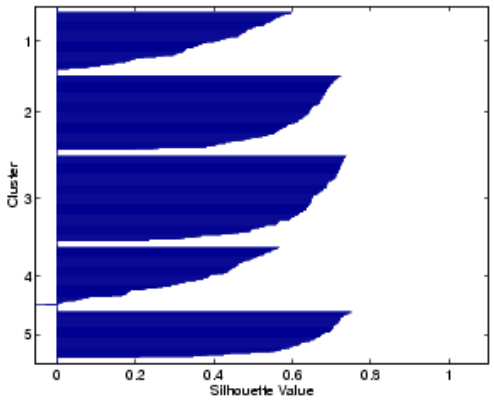
- The Silhouette coefficient of a cluster is the avg silhouette of all its objects
  - Is a measure of how tightly grouped all the data in the cluster are.
  - > 0,7: strong structure, > 0,5: usable structure ….
- The Silhouette coefficient of a clustering is the avg silhouette of all objects
  - is a measure of how appropriately the dataset has been clustered



K=3



K=4



K=5

Outline

- Unsupervised learning vs supervised learning
- A categorization of major clustering methods
- Partitioning-based clustering
- Homework/tutorial
- Things you should know from this lecture

Things you should know from this lecture

Things you should know …

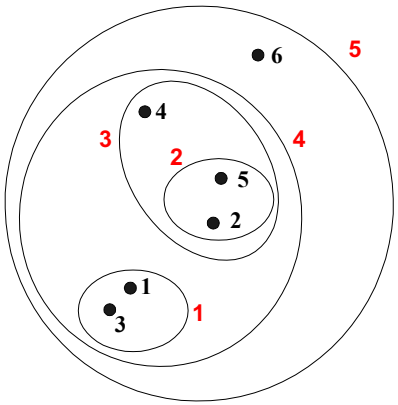
- Principal difference between unsupervised learning and supervised learning
- Categories of diverse clustering methods
- Principal partitioning-based clustering approaches
  - k-Means
  - k-Medoid

Outline

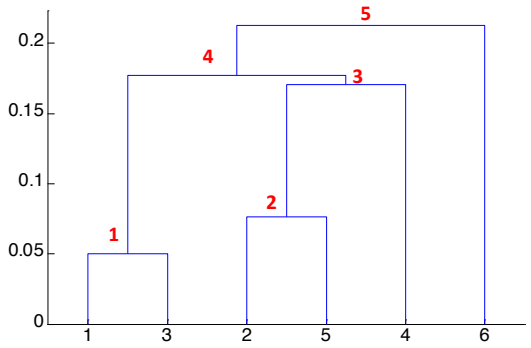
- Hierarchical-based clustering

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



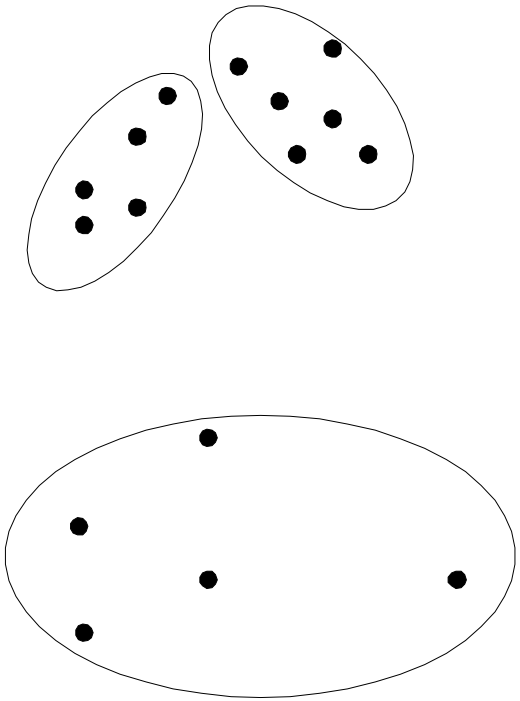
Dendrogram

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

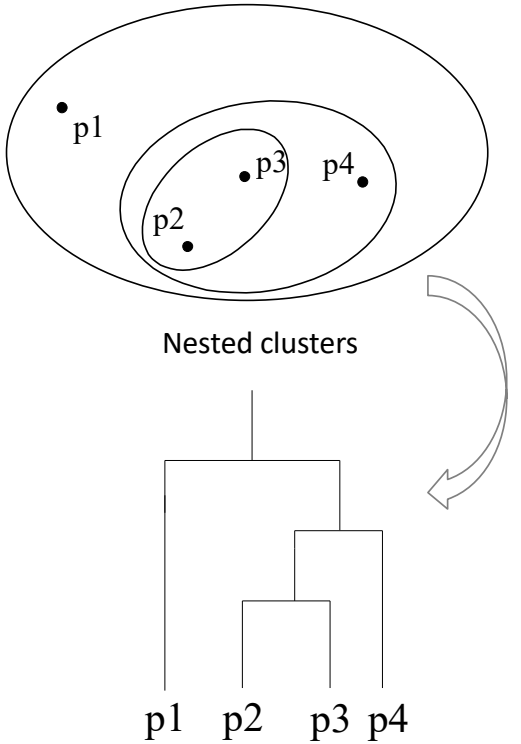
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Hierarchical vs Partitioning



Partitioning clustering

Partitioning algorithms typically have global objectives



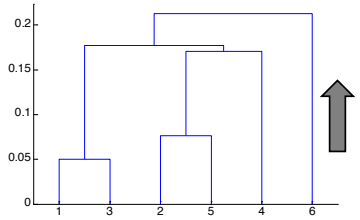
Nested clusters

Dendrogram

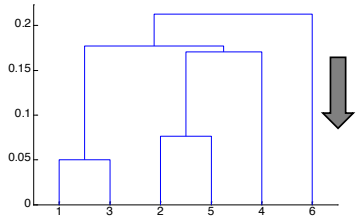
Hierarchical clustering algorithms typically have local objectives

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 or split one cluster at a time

Agglomerative clustering algorithm

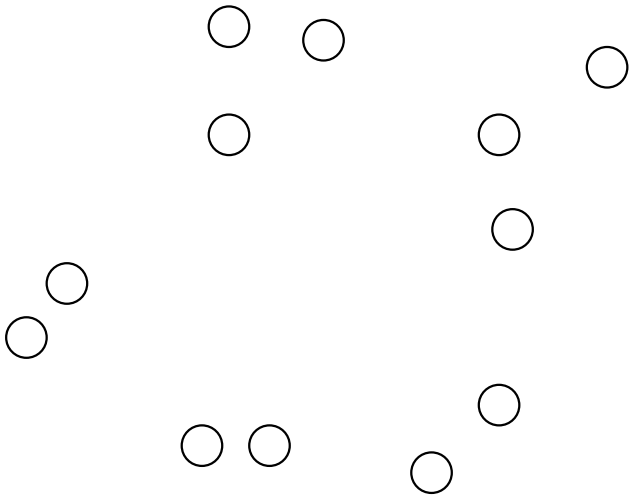
- More popular hierarchical clustering technique
- Basic algorithm is straightforward

1. Compute the proximity matrix
2. Let each data point be a cluster
3. **Repeat**
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

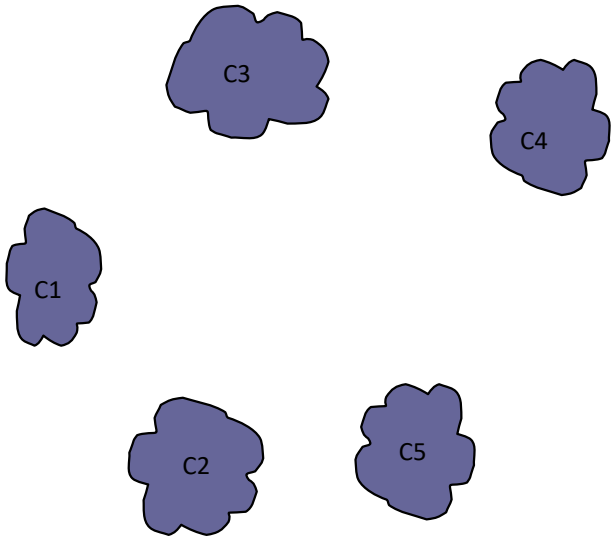


	p1	p2	p3	…	p12	
p1						
p2						
p3						
…						
p12						

Proximity matrix

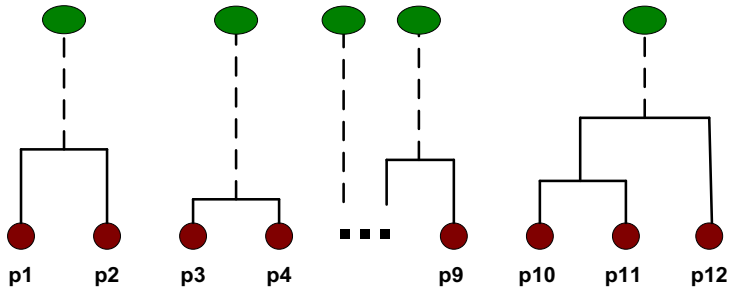
Intermediate situation I

- After some merging steps, we have some clusters



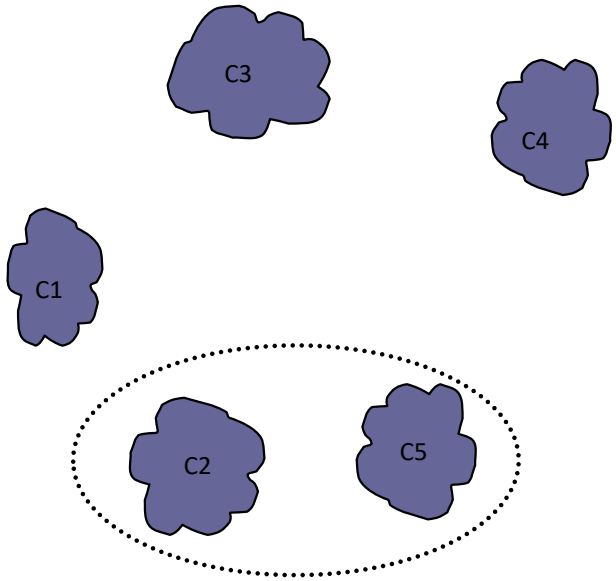
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

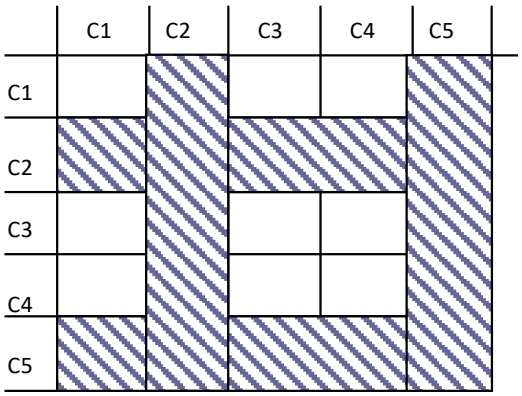
Proximity matrix



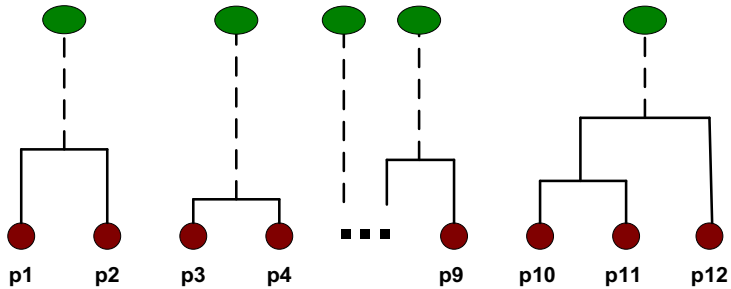
Intermediate situation II

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



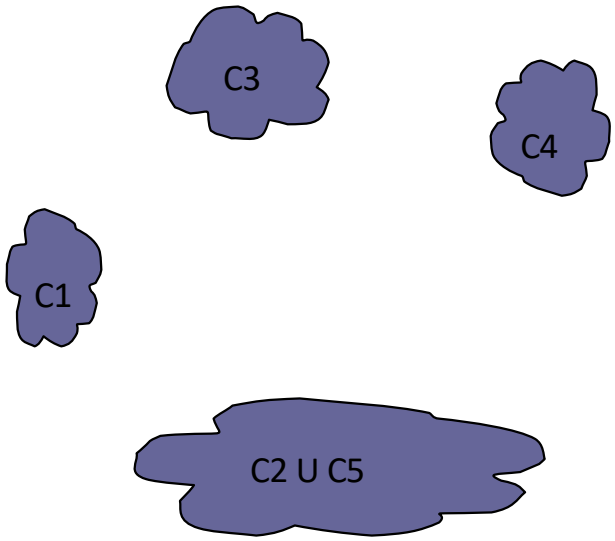


Proximity matrix



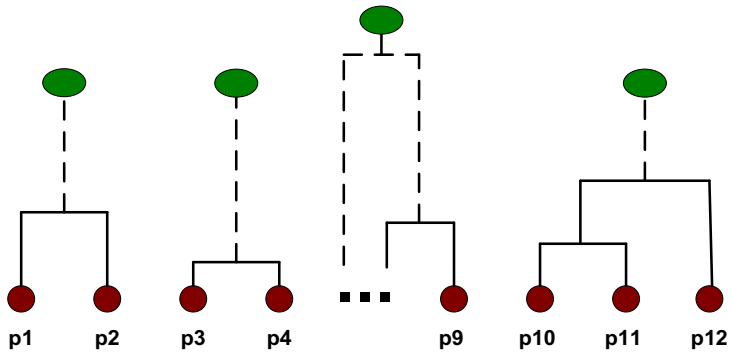
After merging

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

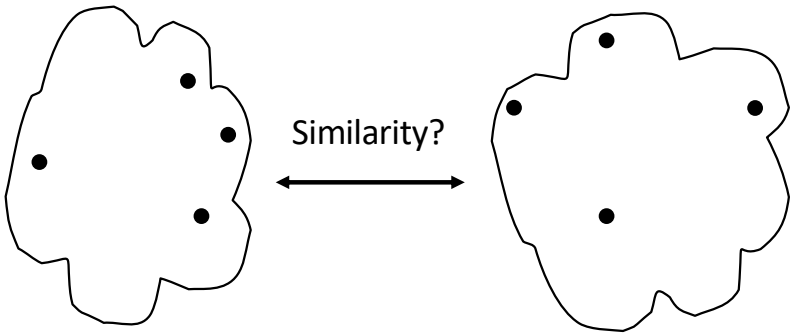


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

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