C12: Clustering

Ming-Syan Chen

November 27, 2019

Tentative Class Agenda

- Class 8 – (10/30) Introduction to data
- Class 9 – (11/6) Association (Project announced, HW#4)
- Class 10 – (11/13) Data Exploration, OLAP, Classification
- Class 11 – (11/20) More on Classification (HW#5)
- Class 12 – (11/27) Clustering; go over project abs.
- Class 13 –(12/4) R (HW#6)
- Class 14 – (12/11) Exam (in class, closed book)
- Class 15 – (12/18) Project presentation I
- Class 16 – (12/25) Project presentation II

What is Cluster Analysis?

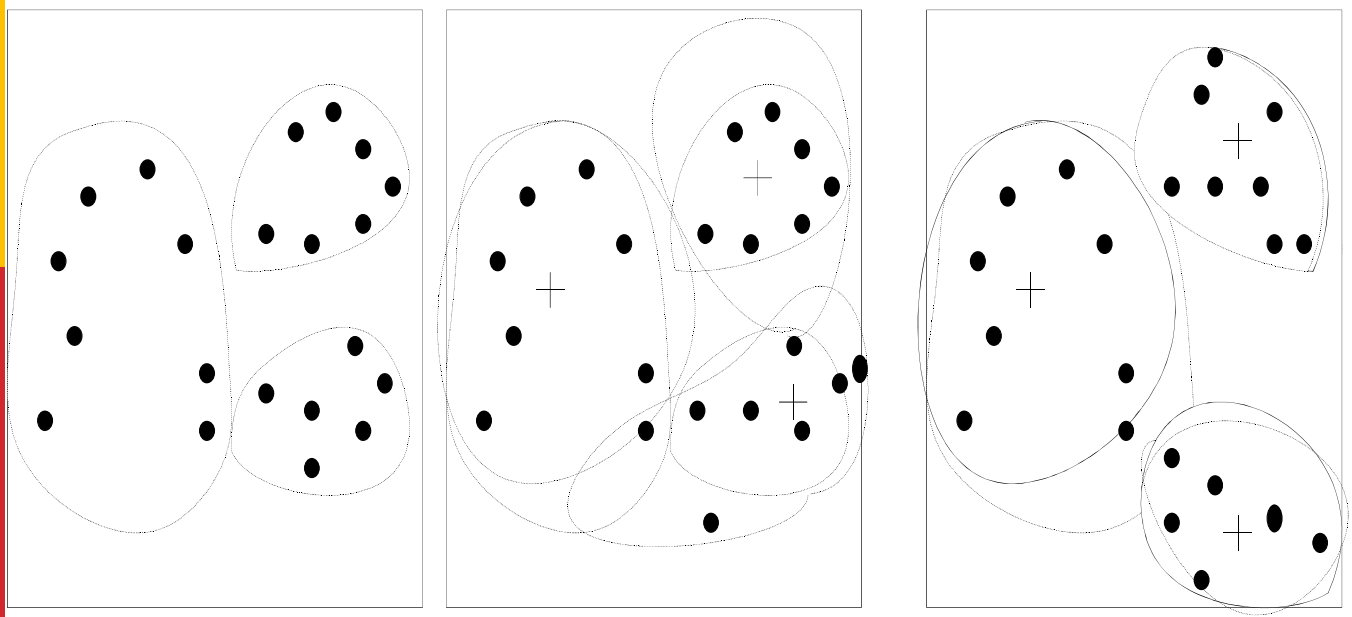
- Cluster: a collection of data objects
  - Similar to one another within the same cluster (intra-class similarity)
  - Dissimilar to the objects in other clusters (inter-class similarity)
- Cluster analysis
  - Grouping a set of data objects into clusters
- Clustering is unsupervised classification: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

What Is Good Clustering?

- A good clustering method will produce high quality clusters with (1) high intra-class similarity and (2) low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

Major Clustering Approaches

- Partitioning algorithms: Construct various partitions and then evaluate them by some criterion, K-Means, K-Medoids
- Hierarchy algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion
- Density-based: based on connectivity and density functions
- Grid-based: based on a multiple-level granularity structure
- Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other



(a) (b) (c)

k-means method

PAM

- Partition around Medoids
- To form k clusters, PAM finds a representative (medoid) in each cluster
- Oi is a medoid and Oj is not. Then, Oj belongs to the cluster represented by Oi if d(Oi, Oj) = min d(Oj,Oe) for all medoids Oe.

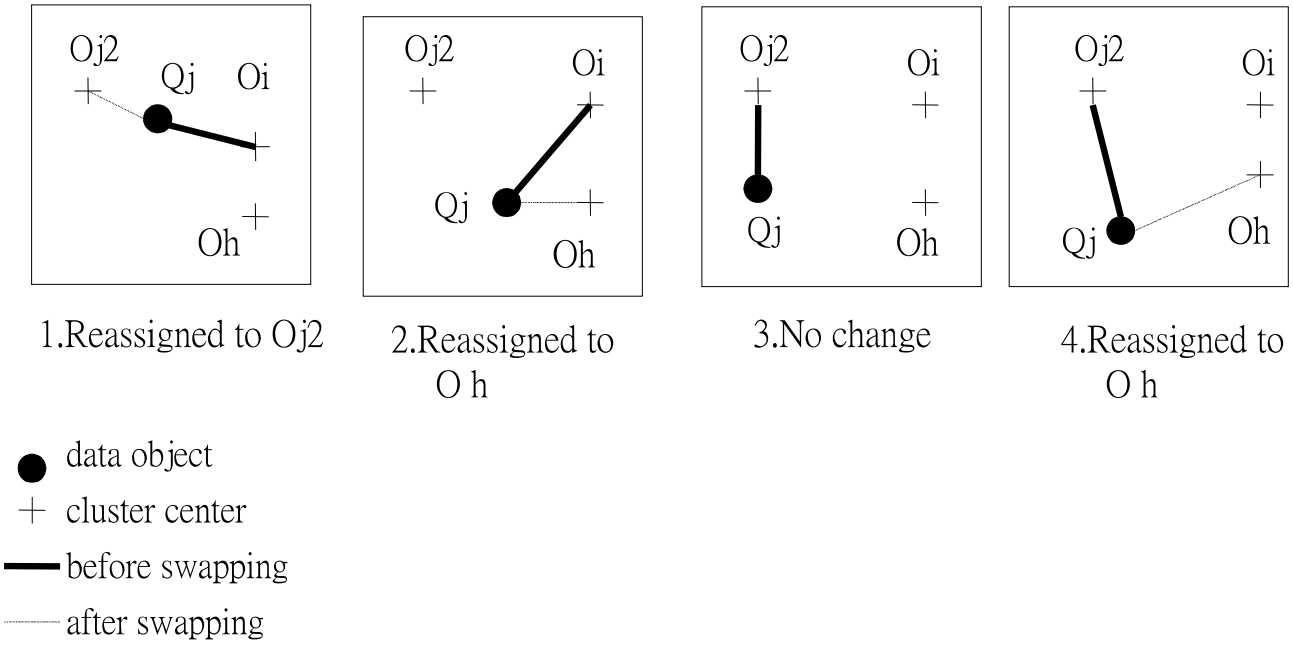
PAM

- Begin with an arbitrary selection of k objects
- Consider a swap between Oi and a non-selected object Oh.
- PAM computes cost Cjih for all non-selected objects Oj

First case

- Suppose O_j currently belongs to the cluster represented by O_i.
- Furthermore, let O_j be more similar to $O_{j,2}$ than O_h, i.e., $d(O_j, O_h) > d(O_j, O_{j,2})$, where $O_{j,2}$ is the second most similar medoid to O_j.

$$C_{jih} = d(O_j, O_{j,2}) - d(O_j, O_i) \qquad (1)$$



Second case

- O_j currently belongs to the cluster represented by O_i.
- But this time, O_j is less similar to $O_{j,2}$ than O_h, i.e., $d(O_j, O_h) < d(O_j, O_{j,2})$.

$$C_{jih} = d(O_j, O_h) - d(O_j, O_i) \qquad (2)$$

Third case

- Suppose that O_j currently belongs to a cluster (represented by Oj2) other than the one represented by O_i.
- But O_j is more similar to $O_{j,2}$ than O_h.

$$C_{jih} = 0. \qquad (3)$$

Fourth case

- O_j currently belongs to the cluster represented by $O_{j,2}$.
- But O_j is less similar to $O_{j,2}$ than O_h.

$$C_{jih} = d(O_j, O_h) - d(O_j, O_{j,2}) \qquad (4)$$

Summary

- Combining the four cases above, the total cost of replacing O_i with O_h is given by:

$$TC_{ih} = \sum_j C_{jih} \qquad (5)$$

Algorithm PAM

1. Select *k* representative objects arbitrarily.
2. Compute TC_{ih} for all pairs of objects O_i, O_h where O_i is currently selected, and O_h is not.
3. Select the pair O_i, O_h which corresponds to $\min_{O_i, O_h} TC_{ih}$. If the minimum TC_{ih} is negative, replace O_i with O_h, and go back to Step (2).
4. Otherwise, for each non-selected object, find the most similar representative object. Halt.

Two Questions

- Prove (or disprove by a counterexample) the following statements.
  - PAM always leads to optimal clsutering
  - K-means will not cause empty cluster(s) in its execution

CLARA

- The complexity of one iteration at PAM is O(k(n-k)(n-k))
- CLARA is designed to use a sampled set to determine medoids

Algorithm CLARA

1. For *i* = 1 to 5, repeat the following steps:

2. Draw a sample of 40 + 2k objects randomly from the entire data set, and call Algorithm PAM to find *k* medoids of the sample.

3. For each object O_j in the entire data set, determine which of the *k* medoids is the most similar to O_j.

Algorithm CLARA (cont'd)

4. Calculate the average dissimilarity of the clustering obtained in the previous step. If the value is less than the current minimum, use this value as the current minimum, and retain the *k* medoids found in Step (2) as the best set of medoids obtained so far.

5. Return to Step (1) to start the next iteration.

Complexity of CLARA

- The complexity of one iteration at CLARA is O(k(40+k)(40+k)+k(n-k))
- CLARA may miss good solutions

CLARANS

- CLARANS: a graph search abstraction
  - In Gnk, each node is represented by a set of k objects
  - two nodes are neighbors if their sets differ by one object
  - each node has k(n-k) neighbors

Algorithm CLARANS

1. Input parameters *numlocal* and *maxneighbor*. Initialize *i* to 1, and *mincost* to a large number.

2. Set *current* to an arbitrary node in $G_{n,k}$.

3. Set *j* to 1.

4. Consider a random neighbor *S* of *current*, and based on Equation (5), calculate the cost differential of the two nodes.

$$TCih = \sum Cjih \qquad (5)$$

Algorithm CLARANS (cont'd)

5. If *S* has a lower cost, set *current* to *S*, and go to Step (3).
6. Otherwise, increment *j* by 1. If $j \leq$ *maxneighbor*, go to Step (4).
7. Otherwise, when *j* > *maxneighbor*, compare the cost of current with *mincost*. If the former is less than *mincost*, set *mincost* to the cost of *current*, and set *bestnode* to current.
8. Increment *i* by 1. If *i* > *numlocal*, output *bestnode* and halt. Otherwise, go to Step (2).

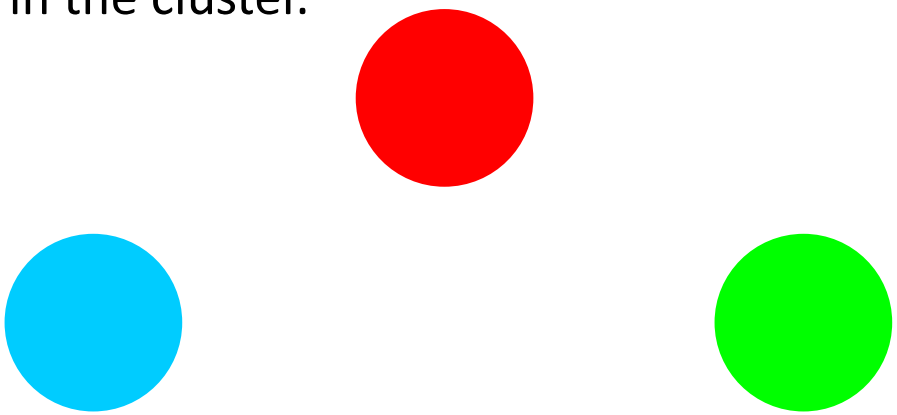
More on Clustering

Types of Clusters

- Well-separated clusters
- Center-based clusters
- Contiguous clusters
- Density-based clusters
- Property or Conceptual

Types of Clusters: Well-Separated

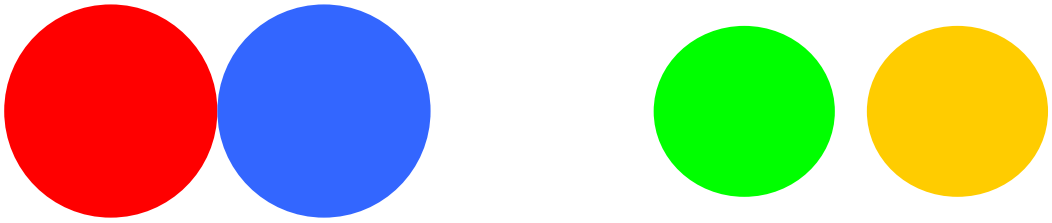
- Well-Separated Clusters:
  - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



3 well-separated clusters

Types of Clusters: Center-Based

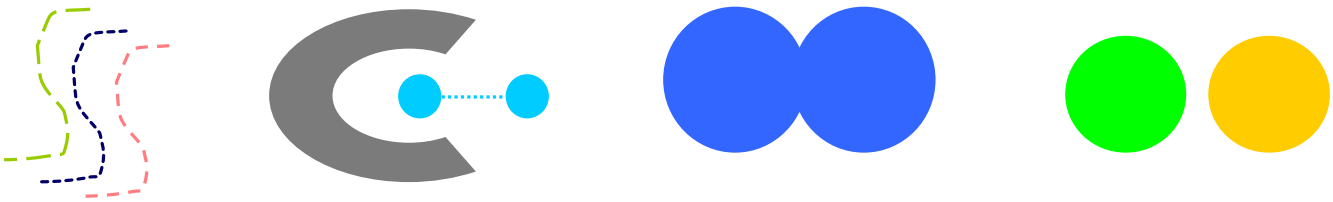
- Center-based
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the “center” of a cluster, than to the center of any other cluster
  - The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most “representative” point of a cluster



4 center-based clusters

Types of Clusters: Contiguity-Based

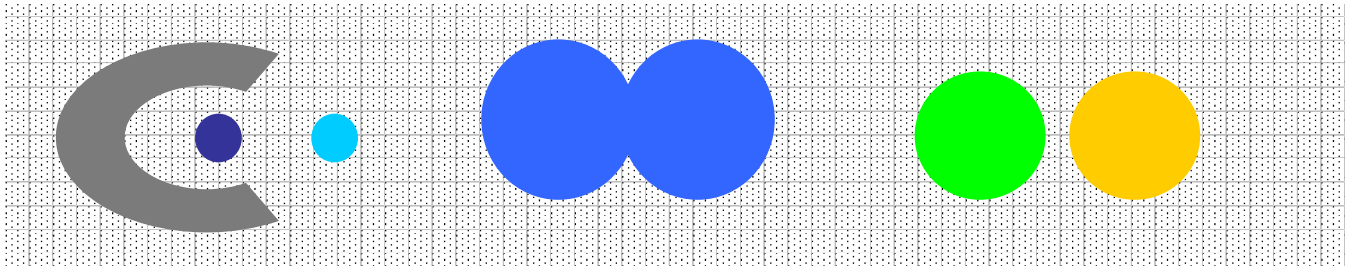
- Contiguous Cluster (Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



8 contiguous clusters

Types of Clusters: Density-Based

- Density-based
  - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  - Used when the clusters are irregular or intertwined, and when noise and outliers are present.

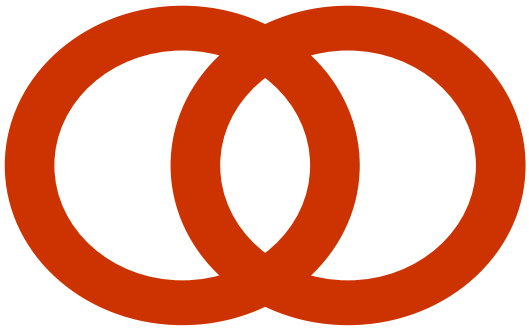


6 density-based clusters

Types of Clusters: Conceptual Clusters

- Shared Property or Conceptual Clusters
  - Finds clusters that share some common property or represent a particular concept.

.

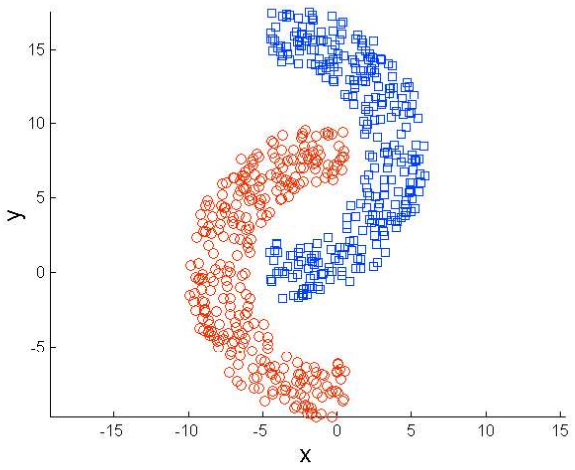


2 Overlapping Circles

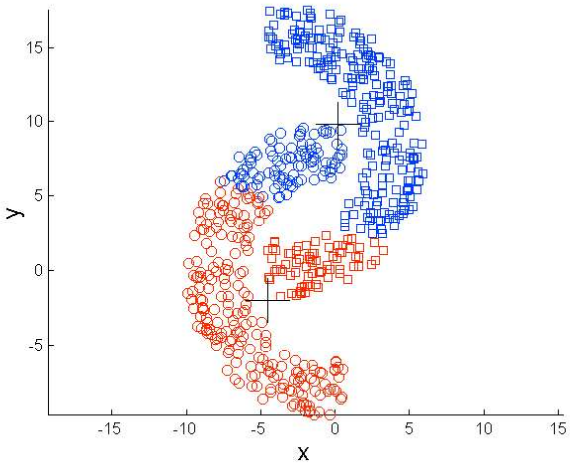
Clustering Algorithms

- K-means and its variants (i.e., PAM, Clara, Claran, etc.)
- Hierarchical clustering
- Density-based clustering

Limitations of K-means: Non-globular Shapes



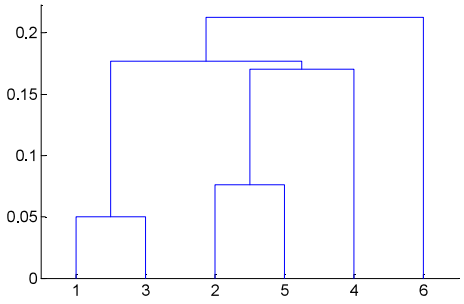
Original Points

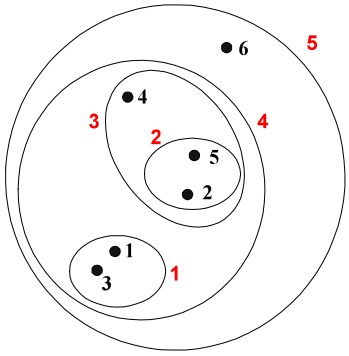


K-means (2 Clusters)

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 of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by 'cutting' the dendogram at the proper level
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Hierarchical Clustering

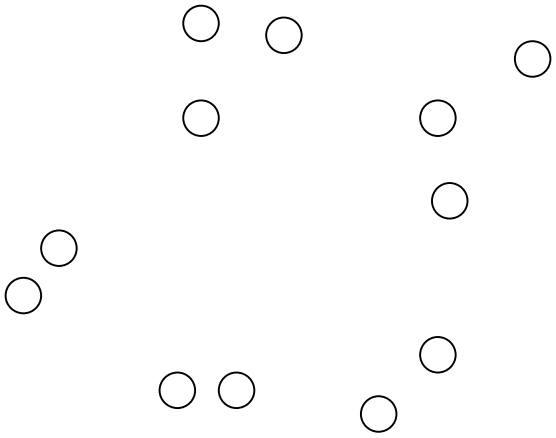
- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster 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**
  4. Merge the two closest clusters
  5. Update the proximity matrix
  6. **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

Starting Situation

- Start with clusters of individual points and a proximity matrix

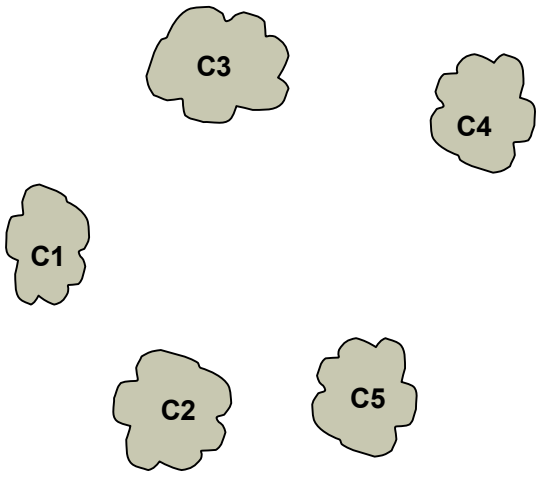


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

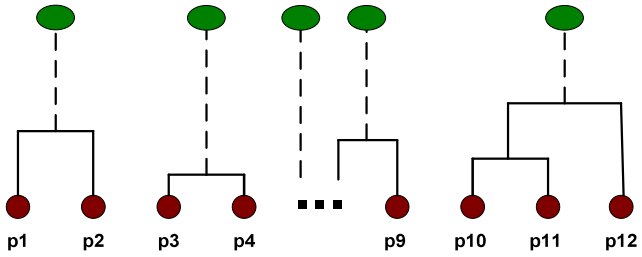
Intermediate Situation

- After some merging steps, we have some clusters



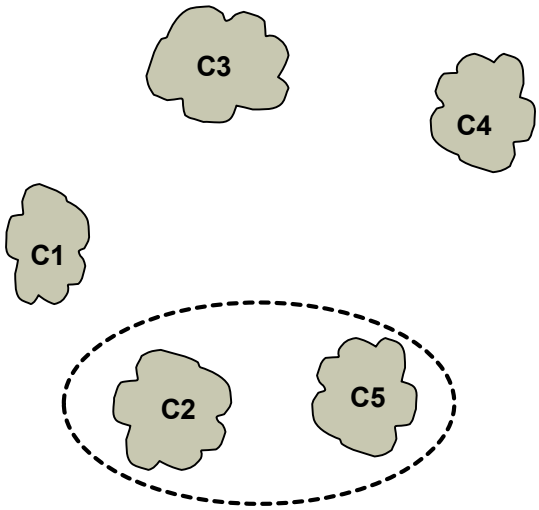
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



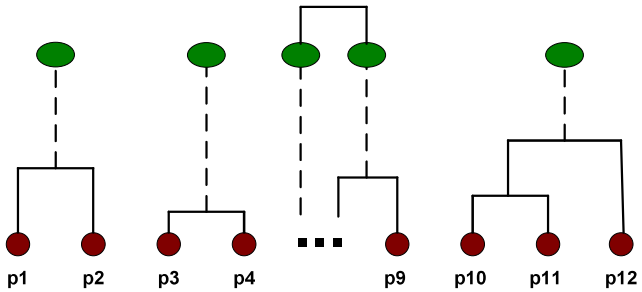
Intermediate Situation

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



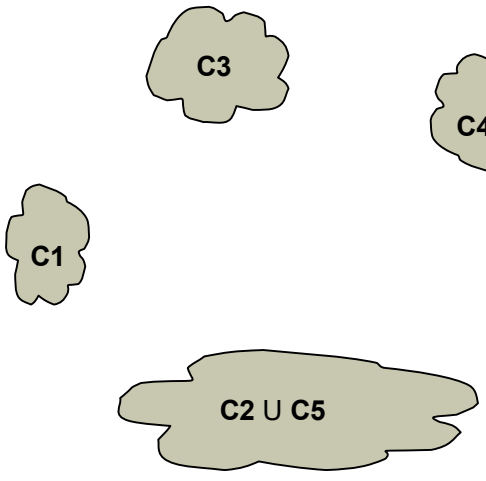
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



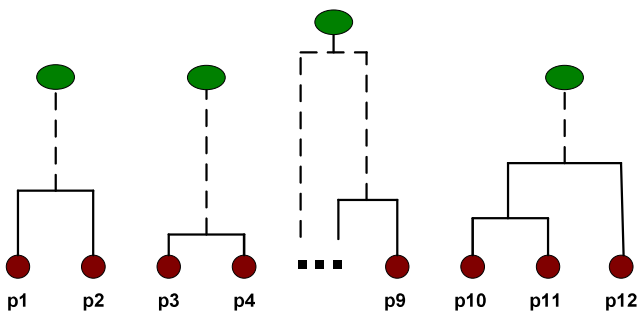
After Merging

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

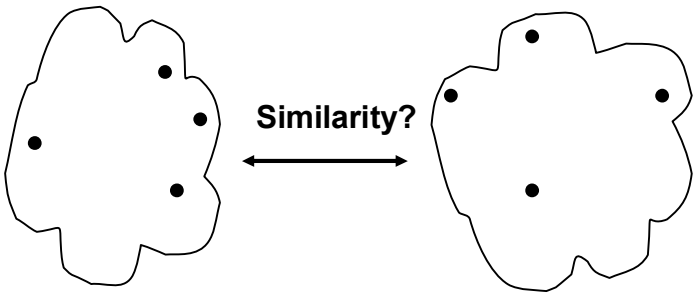


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity

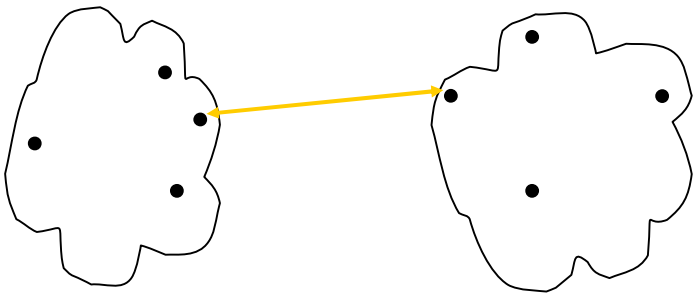


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Medoids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

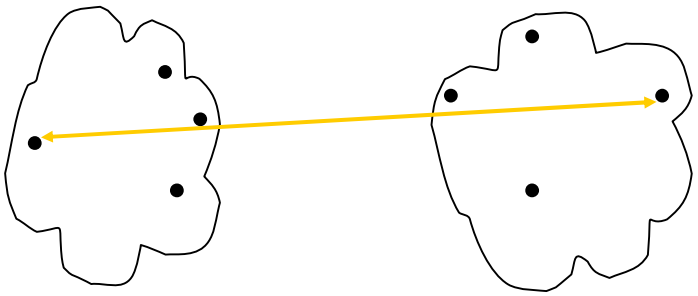


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Medoids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

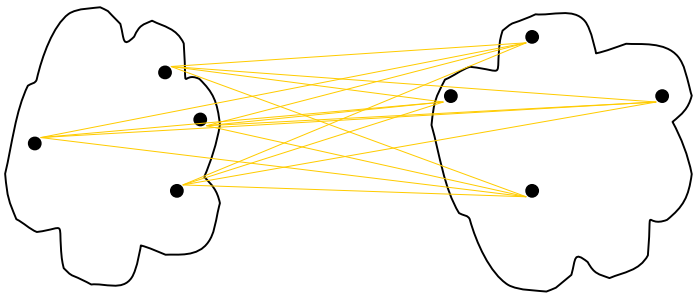


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Medoids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

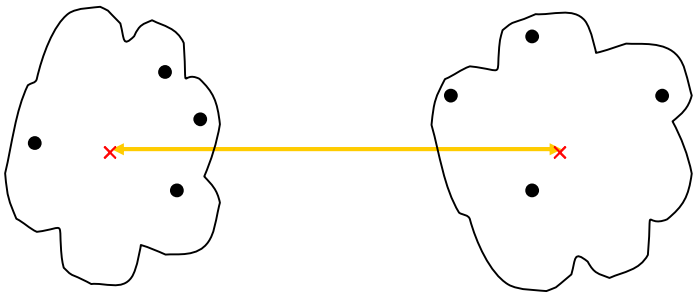


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Medoids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity



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

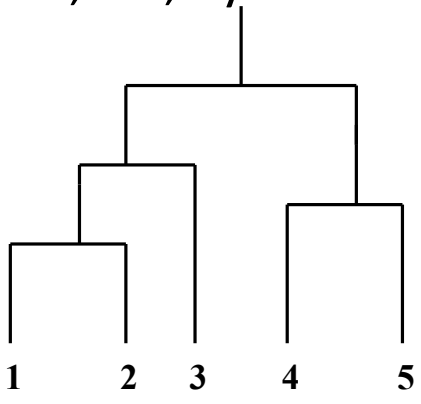
Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Medoids
- Other methods driven by an objective function
  - Ward's Method uses squared error

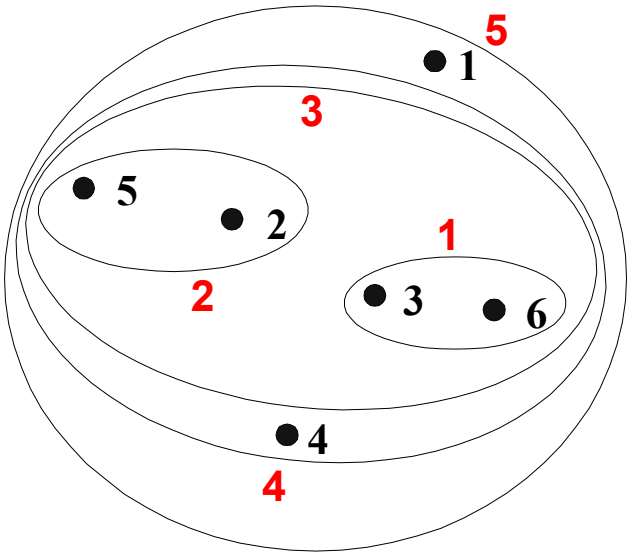
Cluster Similarity: MIN or Single Link

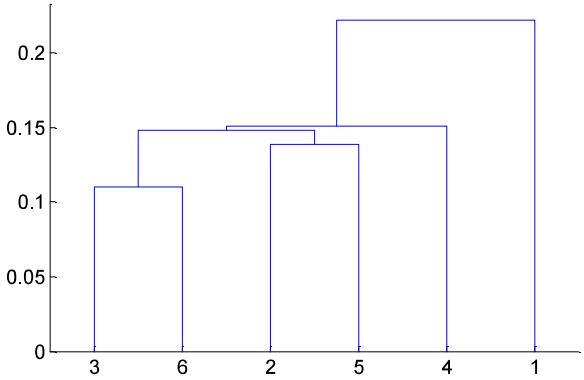
- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MIN



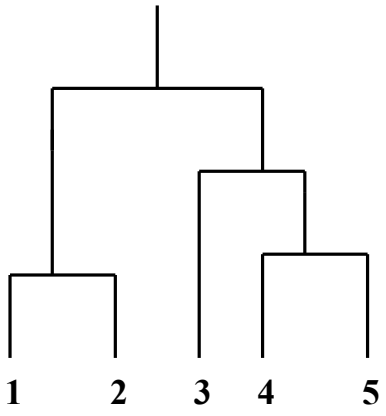


Nested Clusters **Dendrogram**

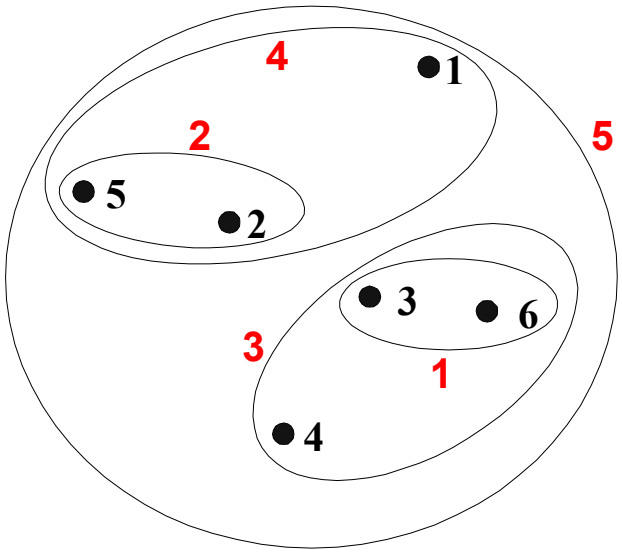
Cluster Similarity: MAX or Complete Linkage

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
  - Determined by all pairs of points in the two clusters

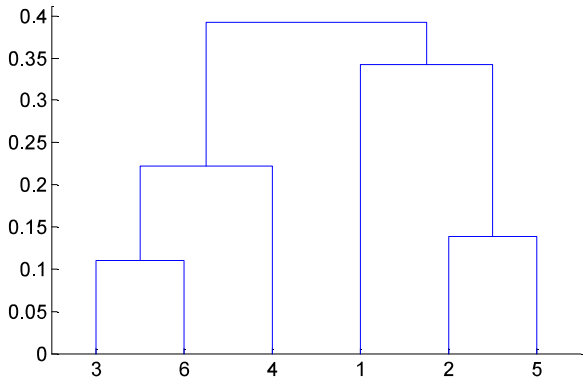
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MAX



Nested Clusters



Dendrogram

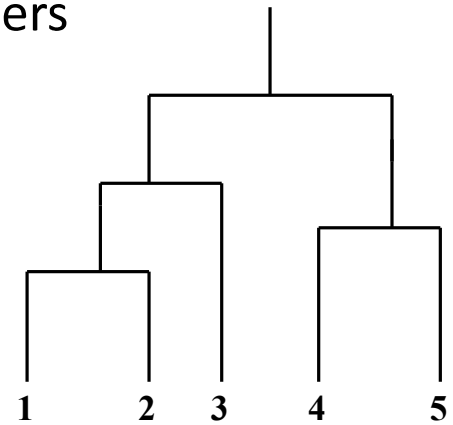
Cluster Similarity: Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

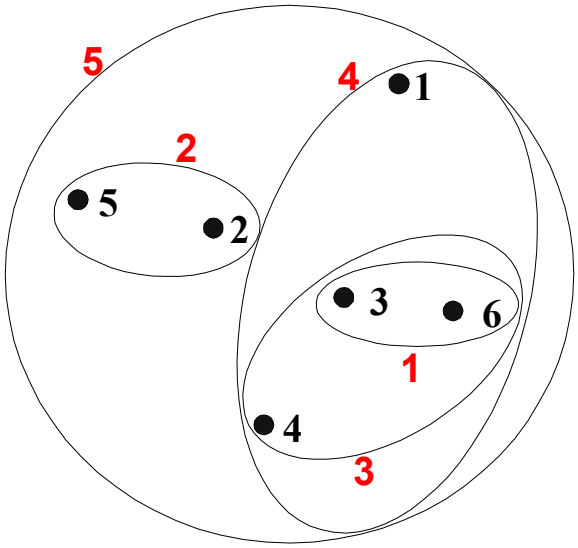
$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

- Need to use average connectivity for scalability since total proximity favors large clusters

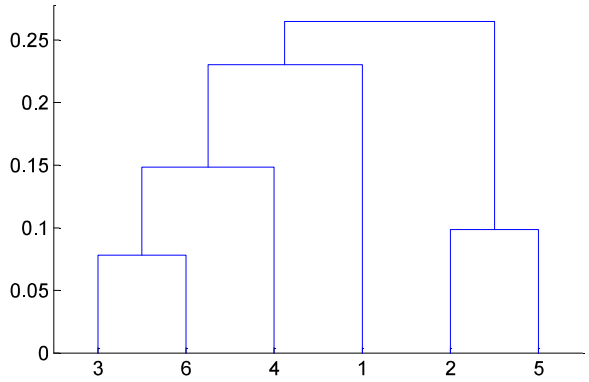
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: Group Average



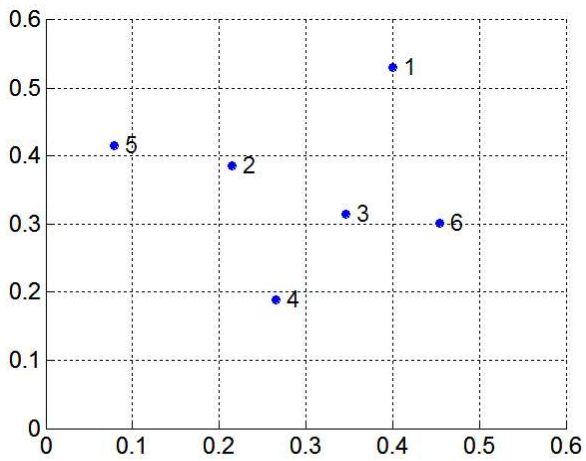
Nested Clusters

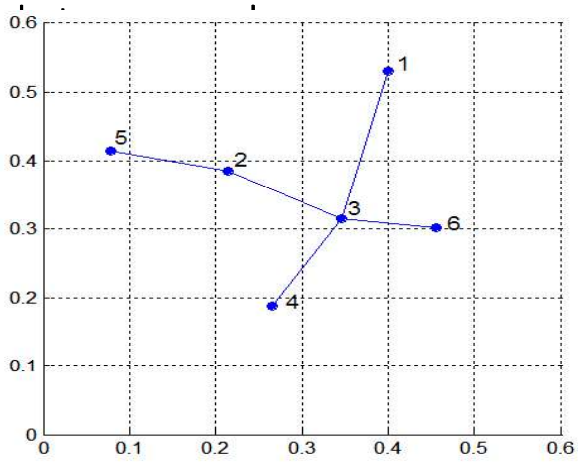


Dendrogram

MST: Divisive Hierarchical Clustering

- Build MST (Minimum Spanning Tree)
  - Start with a tree that consists of any point
  - In successive steps, look for the closest pair of points (p, q) such that one point (p) is in the current tree but the other (q) is not





MST: Divisive Hierarchical Clustering

- Use MST for constructing hierarchy of clusters

Algorithm 7.5 MST Divisive Hierarchical Clustering Algorithm

1: Compute a minimum spanning tree for the proximity graph.
2: **repeat**
3: Create a new cluster by breaking the link corresponding to the largest distance (smallest similarity).
4: **until** Only singleton clusters remain

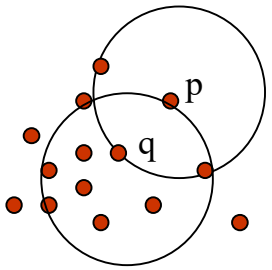
Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)

Density-Based Clustering: Basic Concepts

- Two parameters:
  - *Eps*: Maximum radius of the neighbourhood
  - *MinPts*: Minimum number of points in an Eps-neighbourhood of that point
- $N_{Eps}(p)$: $\{q \text{ belongs to } D \mid dist(p,q) <= Eps\}$
- Directly density-reachable: A point *p* is directly density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if
  - *p* belongs to $N_{Eps}(q)$
  - core point condition:

$$|N_{Eps}(q)| >= MinPts$$

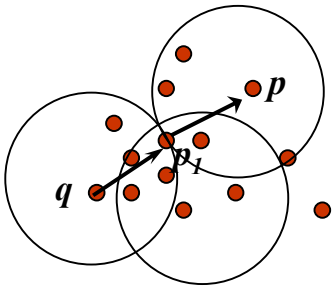


MinPts = 5

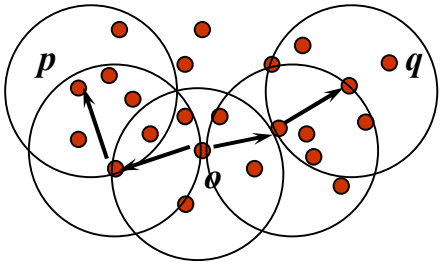
Eps = 1 cm

Density-Reachable and Density-Connected

- Density-reachable:
  - A point *p* is density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n, p_1 = q, p_n = p$ such that p_{i+1} is directly density-reachable from p_i



- Density-connected
  - A point *p* is density-connected to a point *q* w.r.t. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* w.r.t. *Eps* and *MinPts*



DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise