Chap5 Clustering

khai thác dữ liệu và ứng dụng (Trường Đại học Công nghiệp Thành phố Hồ Chí Minh)

Data Mining and Applications

Clustering

What is Clustering in Data Mining?

- Clustering is a process of partitioning a set of data (or objects) in a set of meaningful sub-classes, called clusters
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis (or *clustering*, *data segmentation, …*)
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes (i.e., *learning by observations* vs. learning by examples: supervised)
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Clustering for Data Understanding and Applications

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults
- Climate: understanding earth climate, find patterns of atmospheric and ocean
- Economic Science: market research

Clustering as a Preprocessing Tool (Utility)

- Data reduction
  - Summarization: Preprocessing for regression, PCA, classification, and association analysis
  - Compression: Image processing: vector quantization
- Summarization:
  - Preprocessing for regression, PCA, classification, and association analysis
- Prediction based on groups
  - Cluster & find characteristics/patterns for each group
- Finding K-nearest Neighbors
  - Localizing search to one or a small number of clusters
- Outlier detection
  - Outliers are often viewed as those “far away” from any cluster

Basic Steps to Develop a Clustering Task

- Feature selection / Preprocessing
  - Select info concerning the task of interest
  - Minimal information redundancy
  - May need to do normalization/standardization
- Distance/Similarity measure
  - Similarity of two feature vectors
- Clustering criterion
  - Expressed via a cost function or some rules
- Clustering algorithms
  - Choice of algorithms
- Validation of the results
- Interpretation of the results with applications

Distance or Similarity Measures

- Common Distance Measures:

$$X = \langle x_1, x_2, \cdots, x_n \rangle \quad Y = \langle y_1, y_2, \cdots, y_n \rangle$$

  - Manhattan distance:

$$dist(X,Y) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|$$

  - Euclidean distance:

$$dist(X,Y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}$$

  - Cosine similarity:

$$dist(X,Y) = 1 - sim(X,Y) \qquad sim(X,Y) = \frac{\sum_i (x_i \times y_i)}{\sqrt{\sum_i x_i^2 \times \sum_i y_i^2}}$$

More Similarity Measures

- In vector-space model many similarity measures can be used in clustering

Simple Matching

$$sim(X,Y) = \sum_i x_i \times y_i$$

Dice's Coefficient

$$sim(X,Y) = \frac{2 \cdot \sum_i x_i \times y_i}{\sum_i x_i^2 + \sum_i y_i^2}$$

Cosine Coefficient

$$sim(X,Y) = \frac{\sum_i (x_i \times y_i)}{\sqrt{\sum_i x_i^2 \times \sum_i y_i^2}}$$

Jaccard's Coefficient

$$sim(X,Y) = \frac{\sum_i x_i \times y_i}{\sum_i x_i^2 + \sum_i y_i^2 - \sum_i x_i \times y_i}$$

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters
  - high intra-class similarity: cohesive within clusters
  - low inter-class similarity: distinctive between clusters
- The quality of a clustering method depends on
  - the similarity measure used
  - its implementation, and
  - Its ability to discover some or all of the hidden patterns

Major Clustering Approaches

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue

Major Clustering Approaches (cont.)

- Grid-based approach:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE
- Model-based:
  - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  - Typical methods: EM, SOM, COBWEB
- Frequent pattern-based:
  - Based on the analysis of frequent patterns
  - Typical methods: p-Cluster

Major Clustering Approaches (cont.)

- User-guided or constraint-based:
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering
- Link-based clustering:
  - Objects are often linked together in various ways
  - Massive links can be used to cluster objects: SimRank, LinkClus

Partitioning Algorithms

- Partitioning method: Partitioning a database ***D*** of ***n*** objects into a set of ***k*** clusters, such that the sum of squared distances is minimized (where c_i is the centroid or medoid of cluster C_i)

$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (p - c_i)^2$$

- Given *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67, Lloyd'57/'82): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- The objective is to minimize the sum of distances of the points to their respective centroid

The *K-Means* Clustering Method

- Given the number of desired clusters *k*, the *k-means* algorithm is implemented in four steps:
  1. Partition objects into *k* nonempty subsets
  2. Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., *mean point*, of the cluster)
  3. Assign each object to the cluster with the nearest seed point
  4. Go back to Step 2, stop when the assignment does not change

K-means Clustering

- Most common definition is with euclidean distance, minimizing the Sum of Squares Error (SSE) function
  - Sometimes K-means is defined like that

- **Problem:** Given a set X of n points in a d-dimensional space and an integer K group the points into K clusters $C = \{C_1, C_2, \ldots, C_k\}$ such that

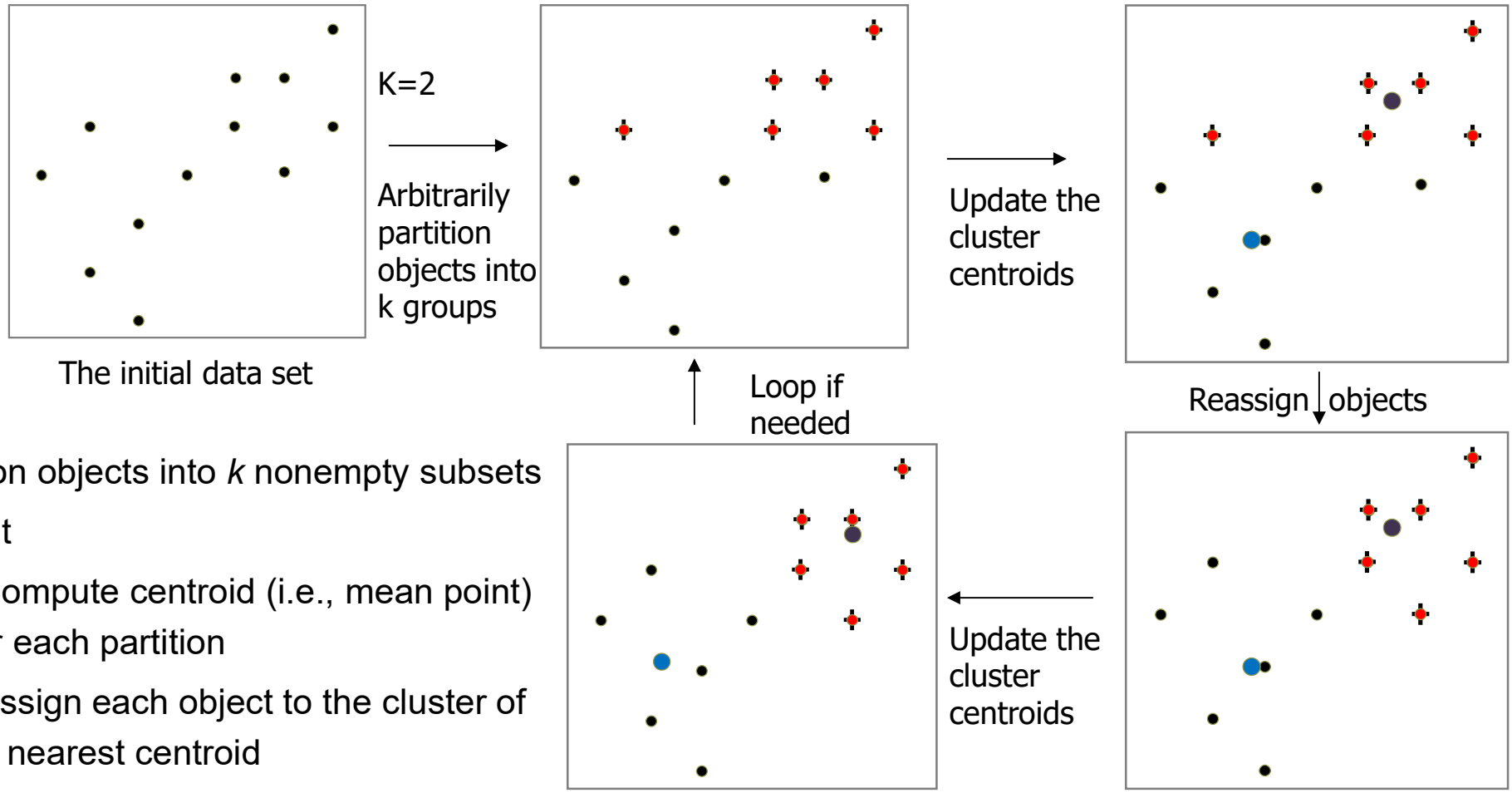
$$\text{SSE} = Cost(C) = \sum_{i=1}^{k} \sum_{x \in C_i} (x - c_i)^2$$

Sum of Squares Error (SSE)

is minimized, where c_i is the mean of the points in cluster C_i

- **Ví dụ:**
  - Let 2 clusters with it's centroid $m_1=3$, $m_2=4$
  - $K_1=\{2,3\}$, $K_2=\{4,10,12,20,30,11,25\}$
  - $SSE = 1^2+0+0+6^2+8^2+16^2+26^2+7^2+21^2 = 1523$

An Example of *K-Means* Clustering



- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

Comments on the *K-Means* Method

- Strength: *Efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t << n$.
  - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- Comment: Often terminates at a *local optimal*.
- Weakness
  - Applicable only to objects in a continuous n-dimensional space
    - Using the k-modes method for categorical data
    - In comparison, k-medoids can be applied to a wide range of data
  - Need to specify *k,* the *number* of clusters, in advance (there are ways to automatically determine the best k (see Hastie et al., 2009)
  - Sensitive to noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

Bài tập

Cho tập dữ liệu 1 chiều gồm { 2, 4, 10, 12, 3, 20, 30, 11, 25} và k = 2.

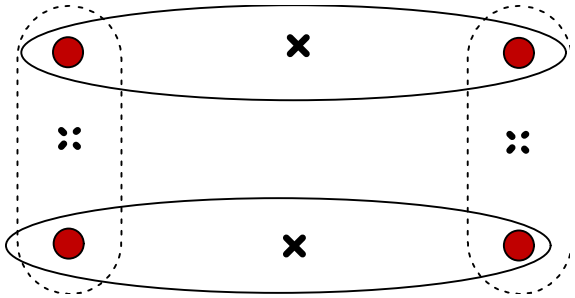
Với trung tâm các nhóm là m_1, m_2, sử dụng thuật toán k-mean với độ đo Euclidean để xác định các cụm. Tính độ đo E cho từng nhóm ở vòng lặp đầu tiên và cuối cùng

Bài tập 2

Cho tập dữ liệu 1 chiều gồm { 2, 3, 4, 10, 12, 20, 25, 30} và k = 2. Với trung tâm các nhóm là m_1 =5, m_2=10, sử dụng thuật toán k-mean để xác định các cụm. Tính độ đo E cho từng nhóm ở vòng lặp đầu tiên và cuối cùng

Variations of the *K-Means* Method

- Most of the variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes*
  - Replacing means of clusters with <u>modes</u>
  - Using new dissimilarity measures to deal with categorical objects
  - Using a <u>frequency</u>-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

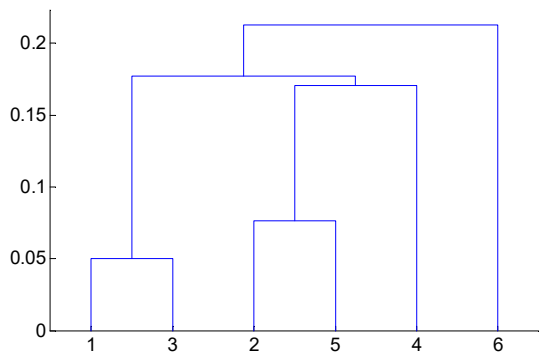


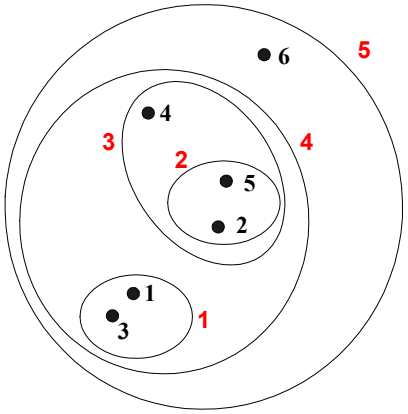
The K-Medoid Clustering Method

- *K-Medoids* Clustering: Find *representative* objects (medoids) in clusters
  - *PAM* (Partitioning Around Medoids, Kaufmann & Rousseeuw 1987)
    - Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
    - *PAM* works effectively for small data sets, but does not scale well for large data sets (due to the computational complexity)
- Efficiency improvement on PAM
  - *CLARA* (Kaufmann & Rousseeuw, 1990): PAM on samples
  - *CLARANS* (Ng & Han, 1994): Randomized re-sampling

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 dendrogram at the proper level

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

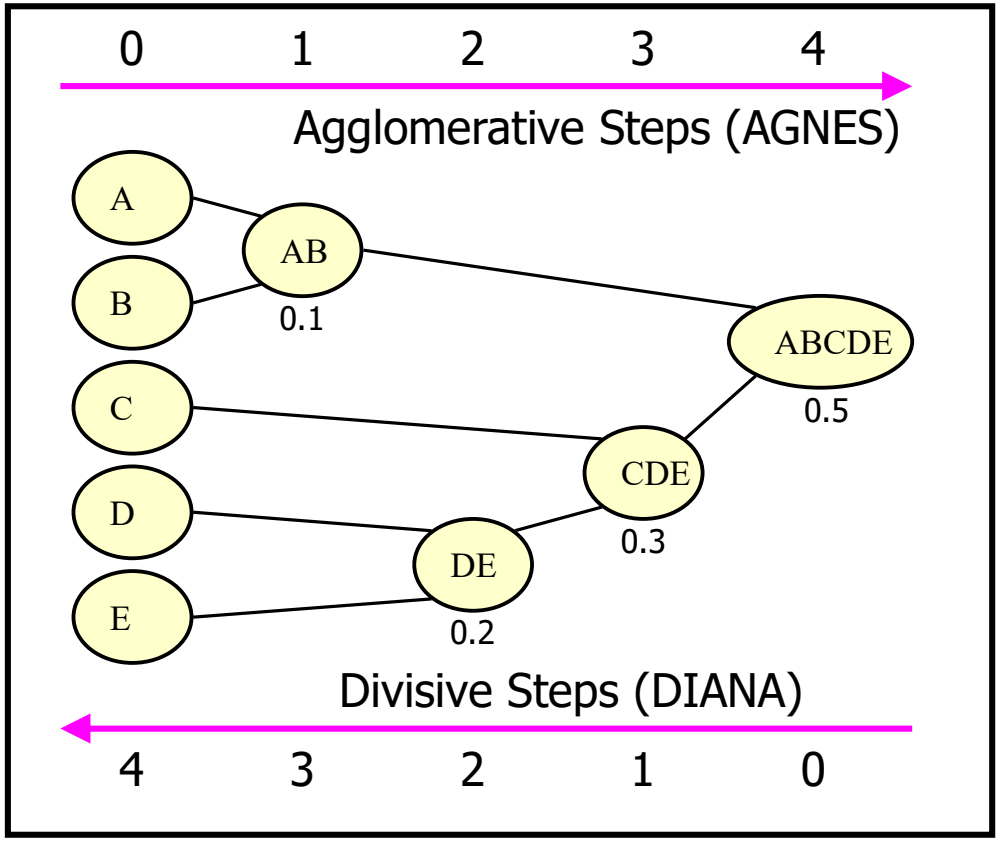
Hierarchical Clustering Algorithms

- 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

Hierarchical Clustering - Example

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition
- Tree structure describing merge / split history.

d(*,*)	A	B	C	D	E
A	0	0.1	0.8	0.7	1.0
B	0.1	0	0.5	0.6	0.9
C	0.8	0.5	0	0.3	0.4
D	0.7	0.6	0.3	0	0.2
E	1.0	0.9	0.4	0.2	0



Agglomerative Clustering Algorithm

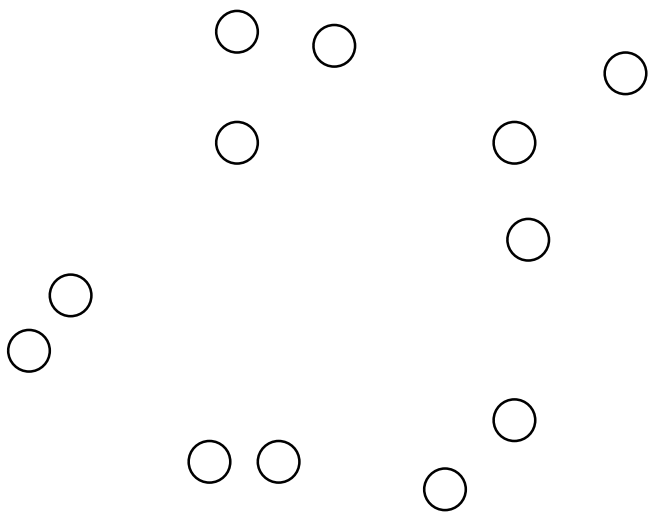
- Agglomerative based on the bottom-up approach
- Key Idea: Successively merge closest clusters
- Basic algorithm: AGNES (AGlomerative NEsting), Kaufman & Rousseeuw, 1990

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

Steps 1 and 2

- Start with clusters of individual points and a proximity matrix

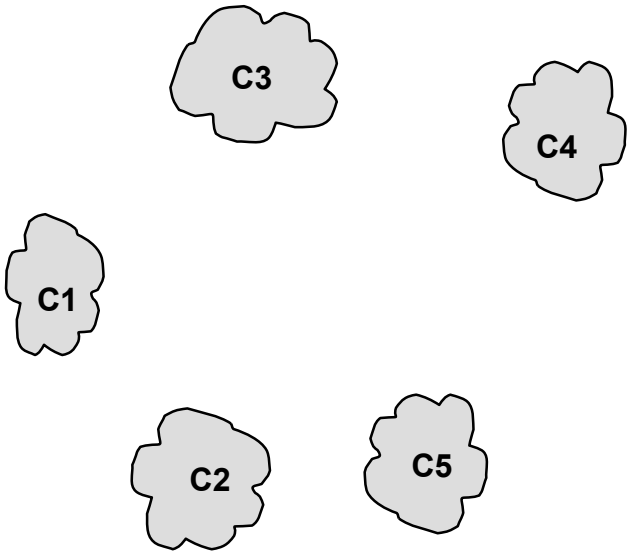


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

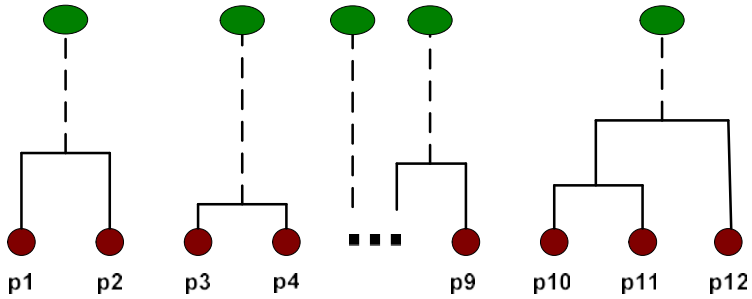
Intermediate Situation

- After some merging steps, we have some clusters



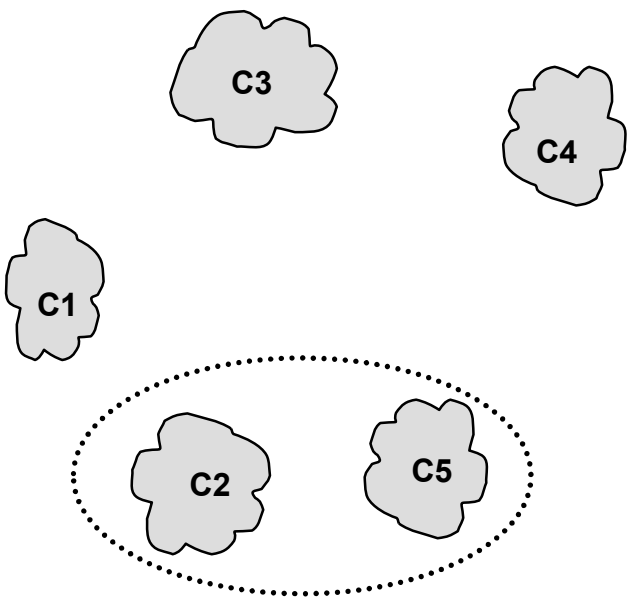
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



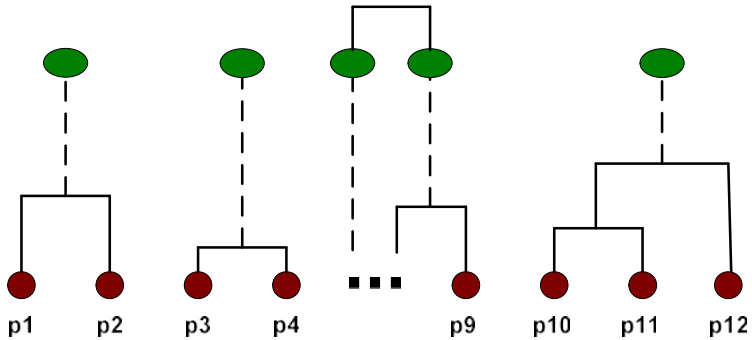
Step 4

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



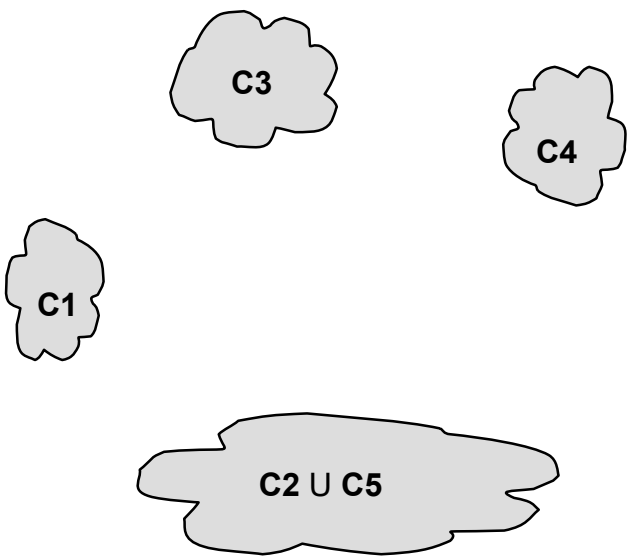
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



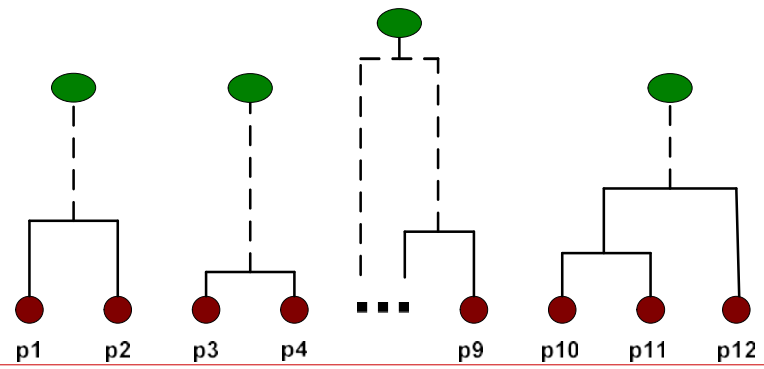
Step 5

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

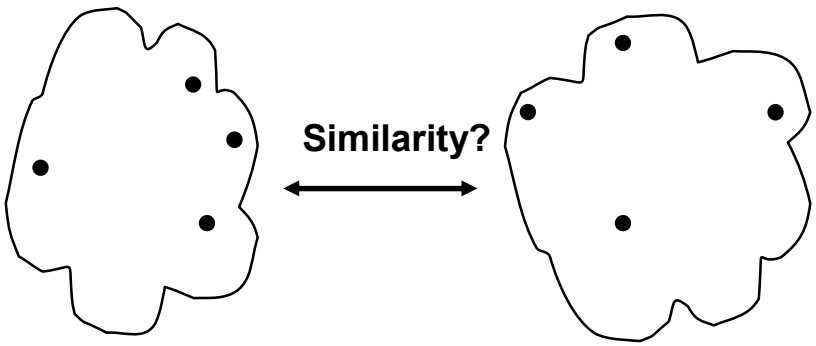


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Distance

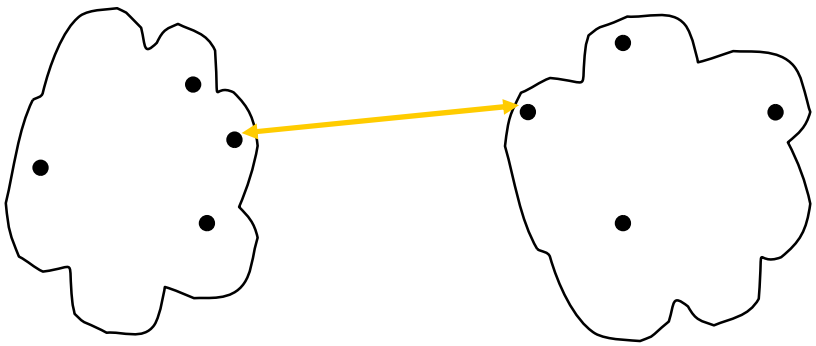


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

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

How to Define Inter-Cluster Similarity

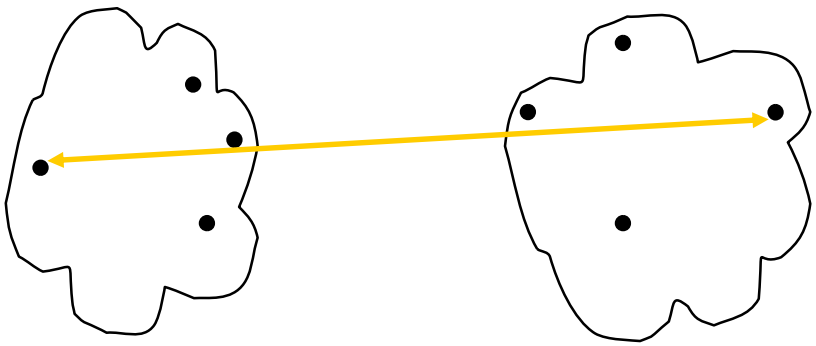


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN (Single Link)
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

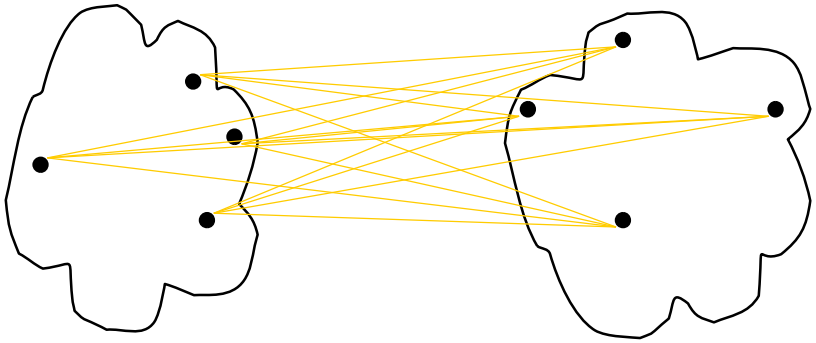


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX (Complete Link)
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

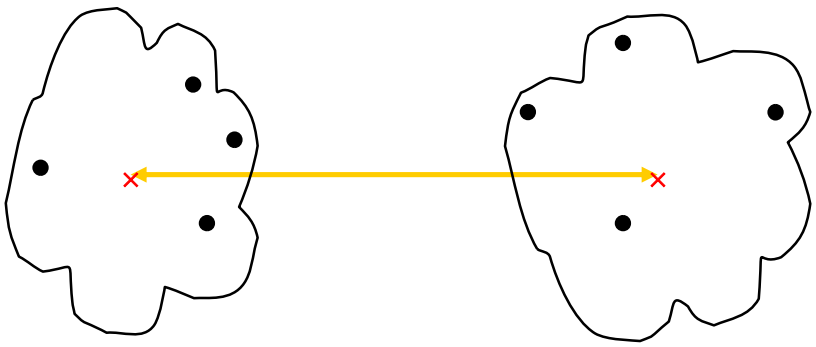


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

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

Centroid, Radius and Diameter of a Cluster

(for numerical data sets)

- Centroid: the "middle" of a cluster

$$C_m = \frac{\sum_{i=1}^{N}(t_{ip})}{N}$$

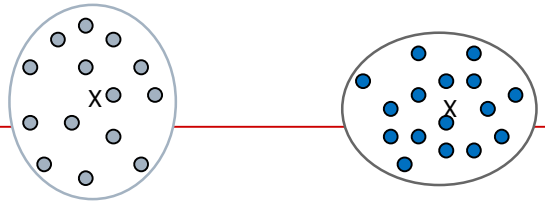
- Radius: square root of average distance from any point of the cluster to its centroid

$$R_m = \sqrt{\frac{\sum_{i=1}^{N}(t_{ip} - c_m)^2}{N}}$$

- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^{N}\sum_{i=1}^{N}(t_{ip} - t_{iq})^2}{N(N-1)}}$$

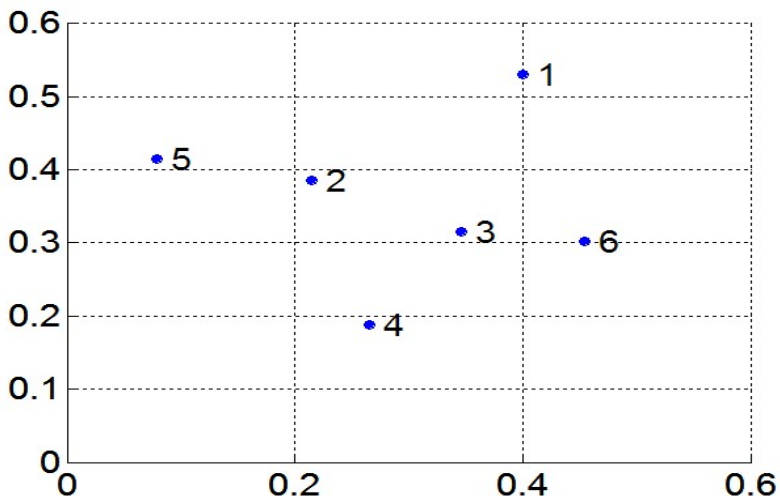
Distance between Clusters



- Single link: smallest distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = min(t_{ip}, t_{jq})$
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = max(t_{ip}, t_{jq})$
- Average: avg distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = avg(t_{ip}, t_{jq})$
- Centroid: distance between the centroids of two clusters, i.e., $dist(K_i, K_j) = dist(C_i, C_j)$
- Medoid: distance between the medoids of two clusters, i.e., $dist(K_i, K_j) = dist(M_i, M_j)$
  - Medoid: a chosen, centrally located object in the cluster

MIN or Single Link

- Proximity of two clusters is based on the two closest points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph
- Example:

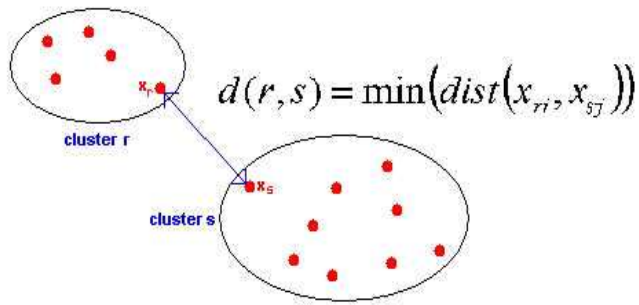


Distance Matrix:

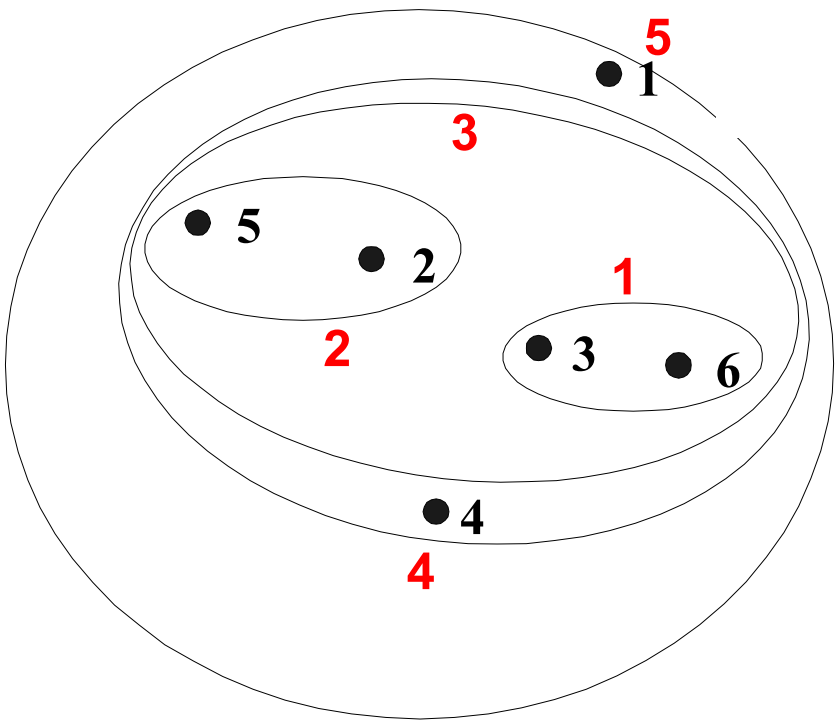
	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Hierarchical Clustering: MIN

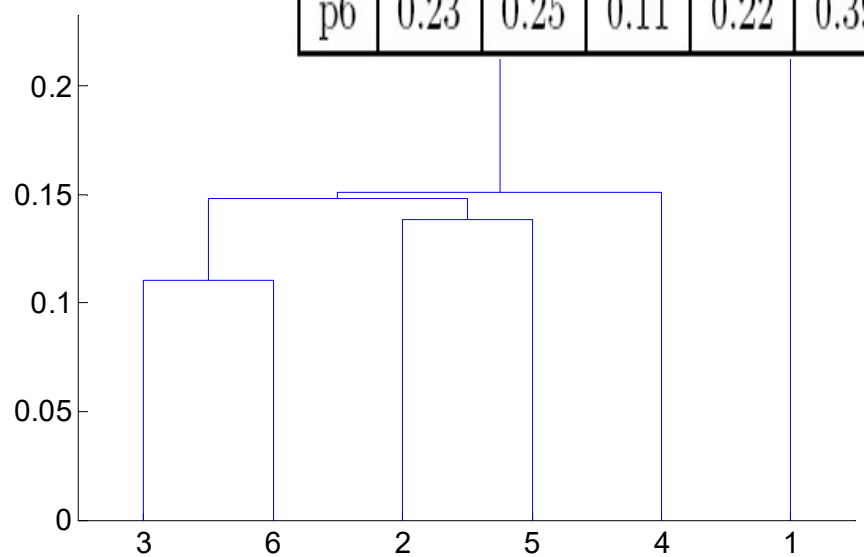
	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00



$$d(r,s) = \min\left(dist\left(x_{ri}, x_{sj}\right)\right)$$

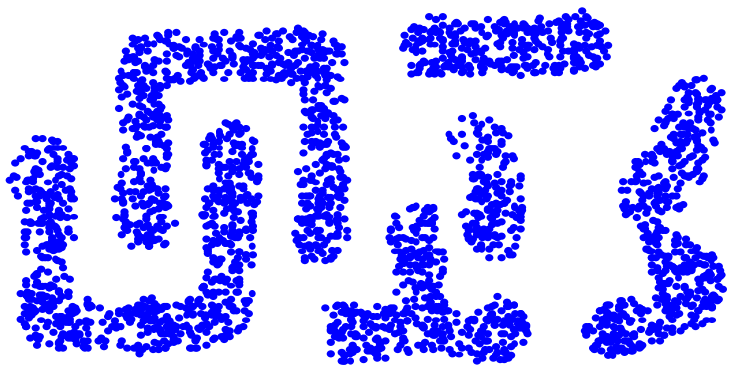


Nested Clusters

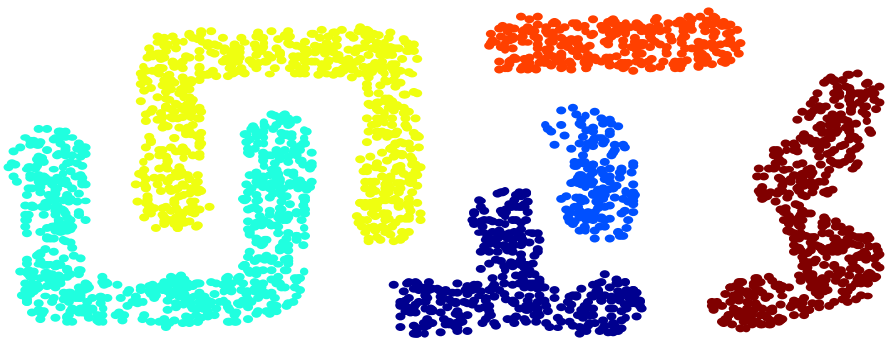


Dendrogram

Strength of MIN



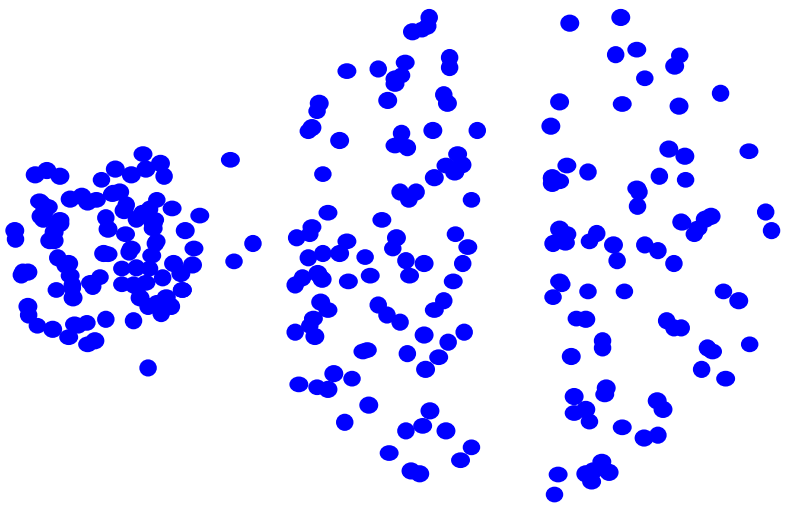
Original Points



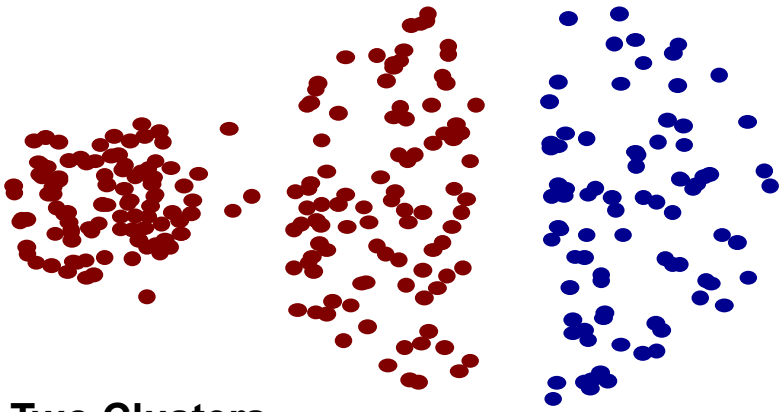
Six Clusters

- Can handle non-elliptical shapes

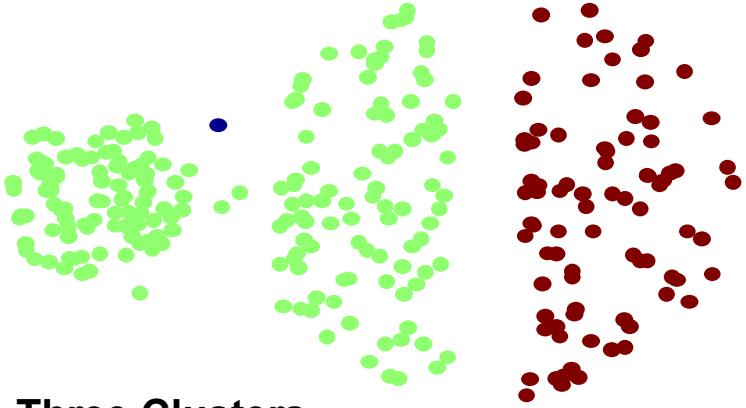
Limitations of MIN



Original Points



Two Clusters

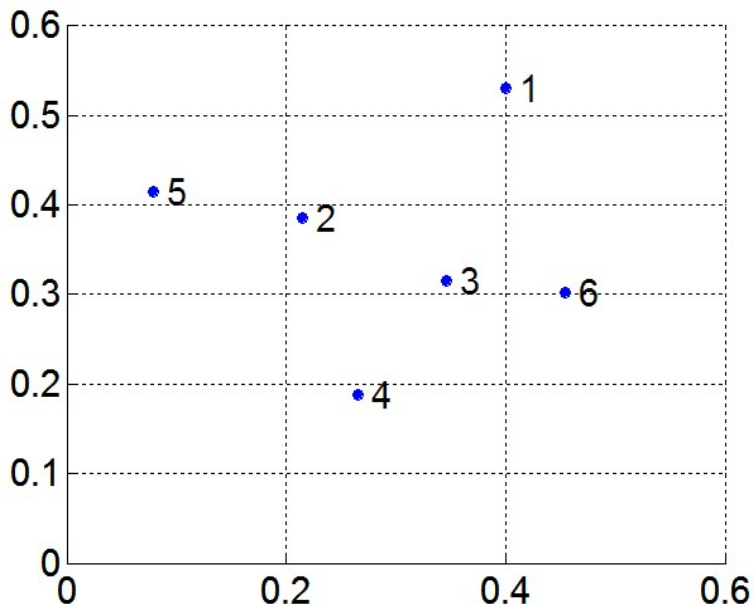


Three Clusters

- Sensitive to noise and outliers

MAX or Complete Linkage

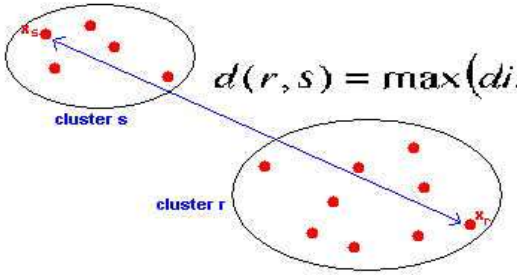
- Proximity of two clusters is based on the two most distant points in the different clusters
  - Determined by all pairs of points in the two clusters



Distance Matrix:

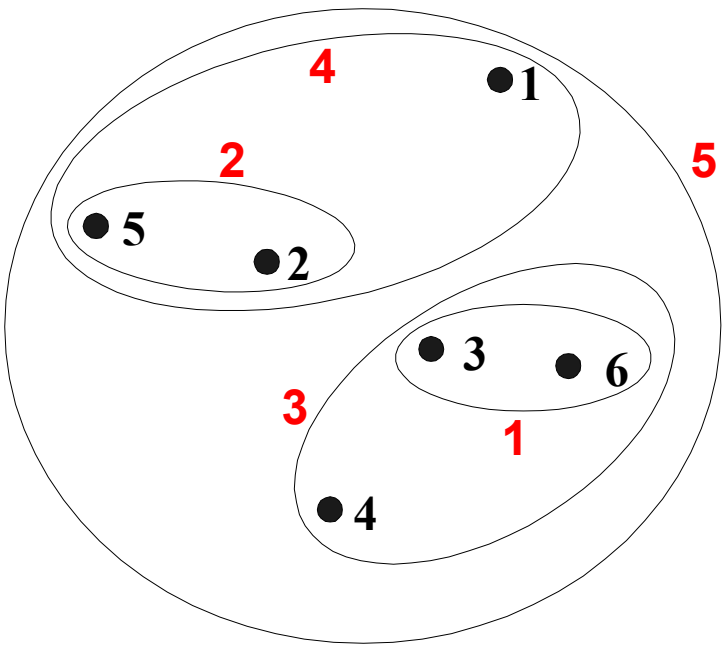
	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Hierarchical Clustering: MAX

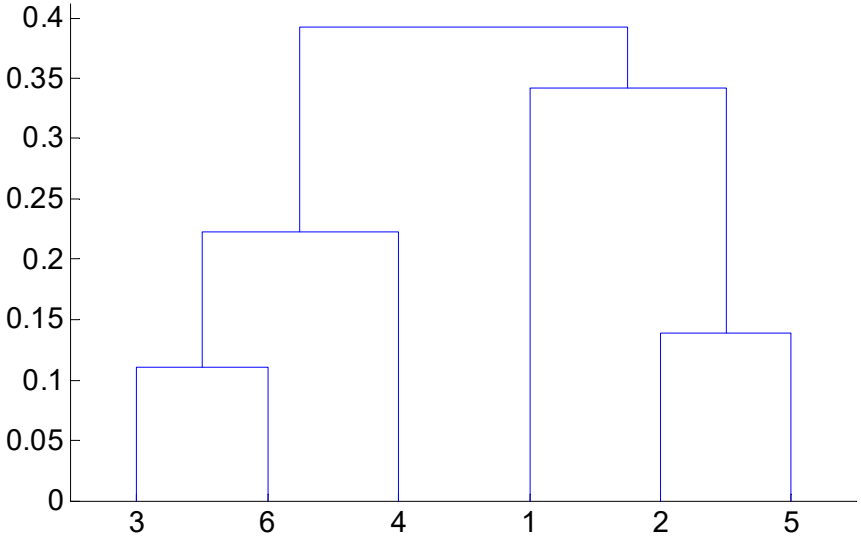


$$d(r,s) = \max\left(dist\left(x_{ri}, x_{sj}\right)\right)$$

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

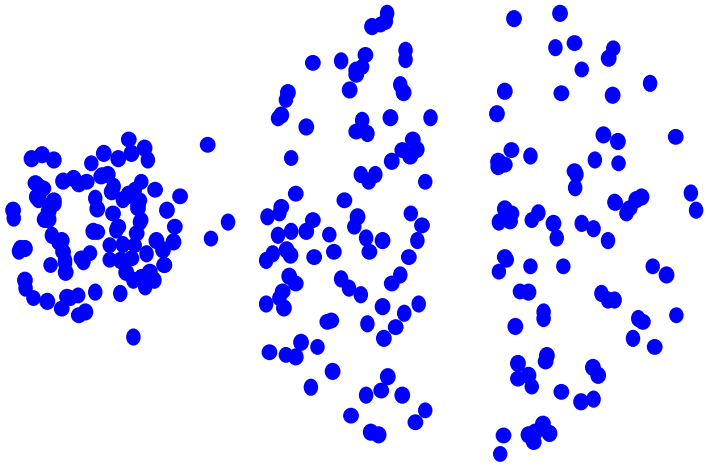


Nested Clusters

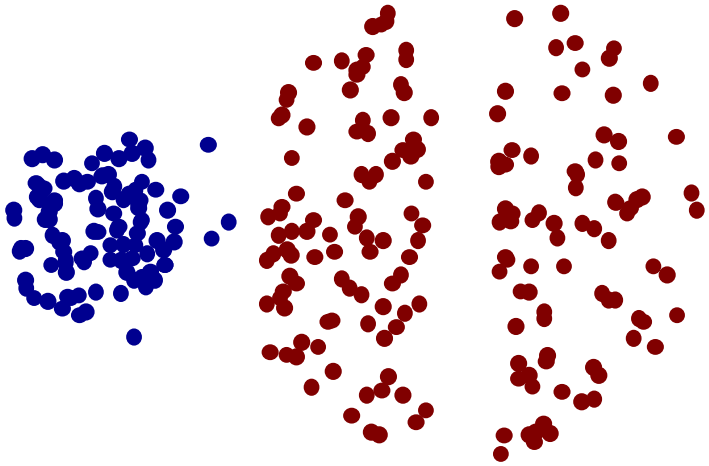


Dendrogram

Strength of MAX



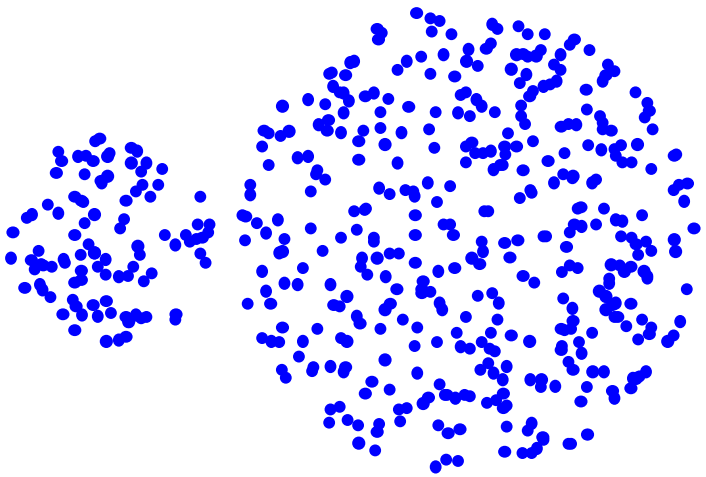
Original Points



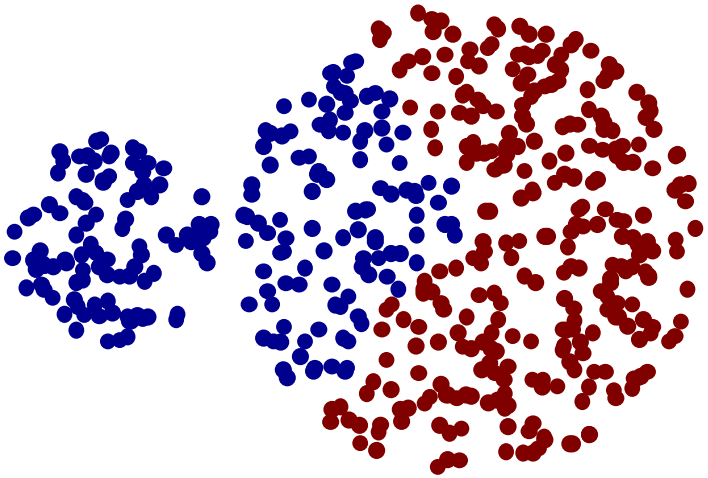
Two Clusters

- Less susceptible to noise

Limitations of MAX



Original Points



Two Clusters

- Tends to break large clusters
- Biased towards globular clusters

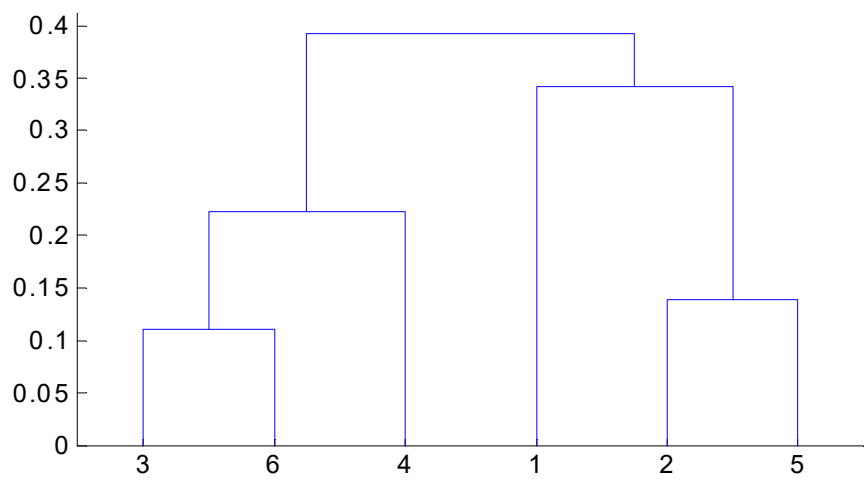
Hierarchical Clustering: Group Average

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

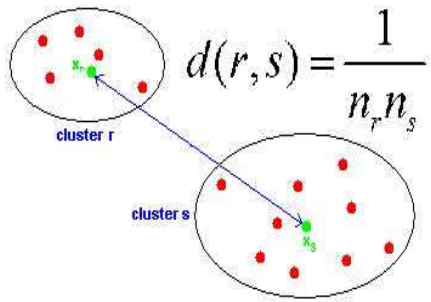
$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

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

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

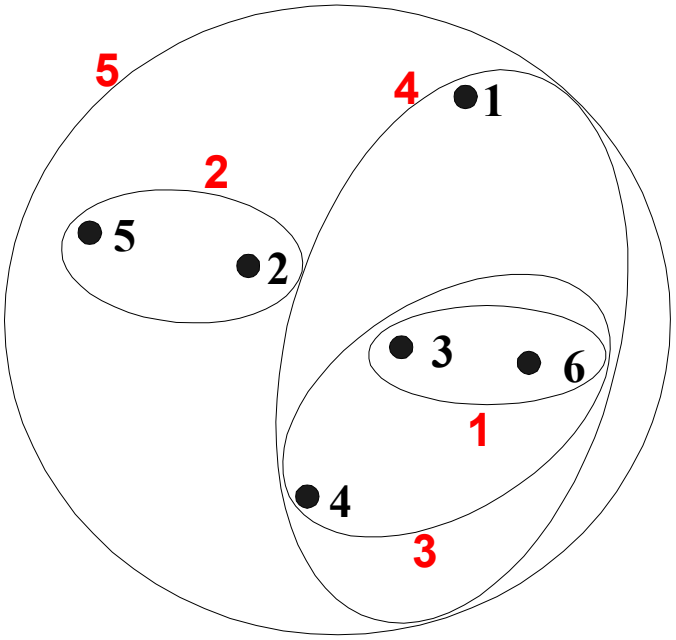


Hierarchical Clustering: Group Average

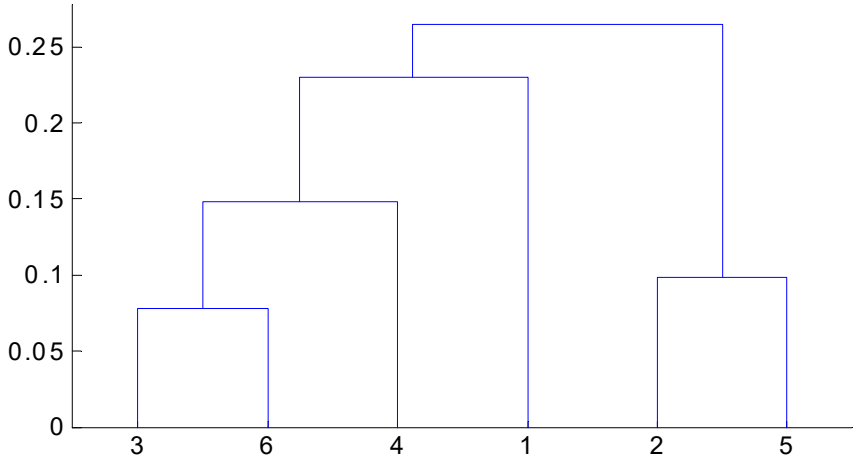


$$d(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} dist\left(x_{ri}, x_{sj}\right)$$

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00



Nested Clusters

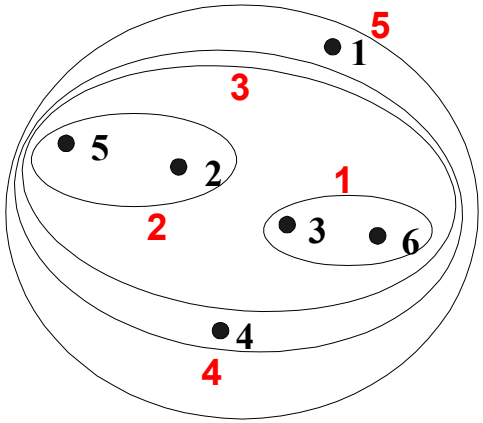


Dendrogram

Hierarchical Clustering: Group Average

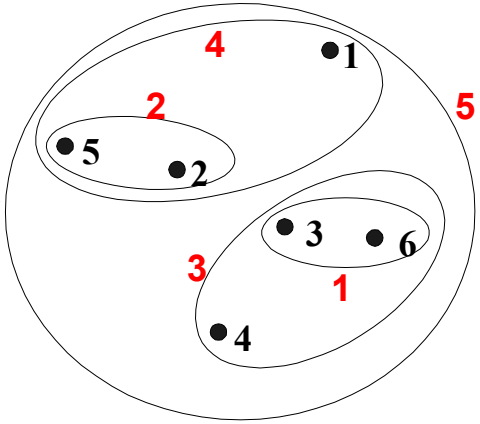
- Compromise between Single and Complete Link
- Strengths
  - Less susceptible to noise
- Limitations
  - Biased towards globular clusters

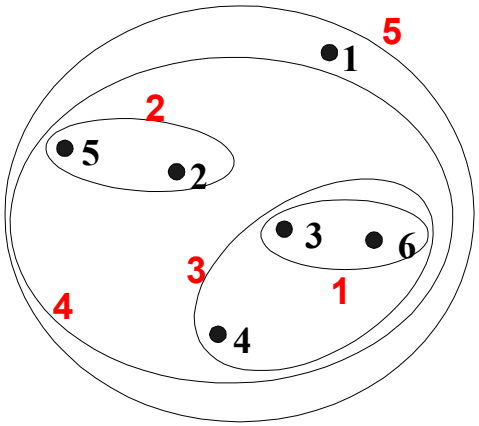
Hierarchical Clustering: Comparison



MIN

MAX





Group Average

How to Define Inter-Cluster Similarity

Example: Given dataset D include:

	X_1	X_2
r1	1	1
r2	2	1
r3	3	3
r4	3	2
r5	4	2

Let two clusters C1 = {r1, r2}, C2 = {r3, r4, r5}. Determine distance d(C1,C2)

How to Define Inter-Cluster Similarity

Distance Matrix :

	r_3	r_4	r_5
r_1	4	3	4
r_2	3	2	3

single-link:

$$d(C_1, C_2) = d(r_2, r_4) = 2$$

complete-link:

$$d(C_1, C_2) = d(r_1, r_3) = d(r_1, r_5) = 4$$

average distance:

$$d(C_1, C_2) = 19/6 = 3.17$$

centroid-link:

$$m_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = \left(\frac{3}{2}, 1\right) \qquad m_2 = \left(\frac{3+3+4}{3}, \frac{3+2+2}{3}\right) = \left(\frac{10}{3}, \frac{7}{3}\right)$$

$$d(C_1, C_2) = d(m_1, m_2) = \left|\frac{3}{2} - \frac{10}{3}\right| + \left|1 - \frac{7}{3}\right| = \frac{19}{6}$$

Divisive approach

- Divisive (top-down) approach:
  - Basic method: DIANA (DIvisive ANAlysis), Kaufman & Rousseeuw, 1990.
  - Inverse order of AGNES
  - Initially, all objects in a single cluster.
  - At each step, a cluster is split into two.
  - Choice of cluster according to a distance criterion between the two clusters generated by the split.
  - Eventually each cluster contains a single object

Exercise

Given a dataset consisting of 6 points in 2D. Use the AGNES algorithm with Single link/Complete link to cluster points in the following dataset:

Point	x	y
P1	0.40	0.53
P2	0.22	0.38
P3	0.353	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

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)

Evaluation of Clustering

Determine the Number of Clusters

- Empirical method
  - # of clusters $\approx\sqrt{n/2}$ for a dataset of n points
- Elbow method
  - Use the turning point in the curve of sum of within cluster variance w.r.t the # of clusters
- Cross validation method
  - Divide a given data set into *m* parts
  - Use *m* – 1 parts to obtain a clustering model
  - Use the remaining part to test the quality of the clustering
    - E.g., For each point in the test set, find the closest centroid, and use the sum of squared distance between all points in the test set and the closest centroids to measure how well the model fits the test set
  - For any k > 0, repeat it *m* times, compare the overall quality measure w.r.t. different *k's*, and find # of clusters that fits the data the best

Bài tập

Bài 1: Thế nào là gom nhóm? Trình bày chi tiết phương pháp phân hoạch, phân cấp. Cho ví dụ cụ thể từng phương pháp. So sánh ưu, khuyết điểm của 2 phương pháp.

Bài 2a: Cho 8 đối tượng sau (biểu diễn thông qua tọa độ (x,y)) : A1(2,10), A2(2,5), A3(8,4), B1(5,8), B2(7,5), B3(6,4), C1(1,2), C2(4,9).
Giả sử gán A1, B1, C1 là các trung tâm của các nhóm tương ứng. Sử dụng thuật toán k-means (với k=3) để phân cụm các đối tượng trên:
–Tính độ đo SSE cho các nhóm sau vòng lặp thi hành đầu tiên.
–Tính độ đo SSE cho các nhóm sau vòng lặp thi hành cuối cùng.

Bài tập

Bài 2b: Cho 8 đối tượng sau (biểu diễn thông qua tọa độ (x,y)) : $A_1(2,10)$, $A_2(2,5)$, $A_3(8,4)$, $B_1(5,8)$, $B_2(7,5)$, $B_3(6,4)$, $C_1(1,2)$, $C_2(4,9)$.

Tự chọn 3 trung tâm nhóm bất kỳ không trùng với 8 đối tượng đã cho. Sử dụng k-mean (k=3) để xác định các nhóm cho các đối tượng trên.

–Tính độ đo SSE cho các nhóm sau vòng lặp thi hành đầu tiên.
–Tính độ đo SSE cho các nhóm sau vòng lặp thi hành cuối cùng.

Bài tập

Bài 3: Ta có 8 đối tượng sau (biểu diễn thông qua tọa độ (x,y)) : $A_1(2,10)$, $A_2(2,5)$, $A_3(8,4)$, $B_1(5,8)$, $B_2(7,5)$, $B_3(6,4)$, $C_1(1,2)$, $C_2(4,9)$.
Sử dụng thuật toán phân cấp lần lượt với Single link và Complete link để xác định 3 nhóm từ DL trên. Vẽ sơ đồ hình cây tương ứng

Bài tập

Bài 4: Sử dụng k-mean để gom cụm với k =3 cho tập dữ liệu bên dưới. Tính độ đo SSE và so sánh kết quả

Customer	Age	Income (K)	No. cards
Thảo	35	37	3
Hưng	25	51	3
Gia	29	44	1
Thành	45	100	3
Thủy	20	30	4
Đức	33	57	2
Minh	65	200	1
Nhung	54	142	2
Nhật	58	175	1
Tùng	25	40	5

Bài tập

Bài 5: Cho tập DL gồm 5 điểm trong không gian 2 chiều với ma trận khoảng cách đã cho.

a. Sử dụng thuật toán AGNES lần lượt với Single Link và Complete link để gom nhóm. Vẽ sơ đồ hình cây.

b. Xác định 3 nhóm thu được từ sơ đồ hình cây theo cả 2 cách

	P1	**P2**	**P3**	**P4**	**P5**
P1	0.00	0.10	0.41	0.55	0.35
P2	0.10	0.00	0.64	0.47	0.98
P3	0.41	0.64	0.00	0.44	0.85
P4	0.55	0.47	0.44	0.00	0.76
P5	0.35	0.98	0.85	0.76	0.00