Data Mining

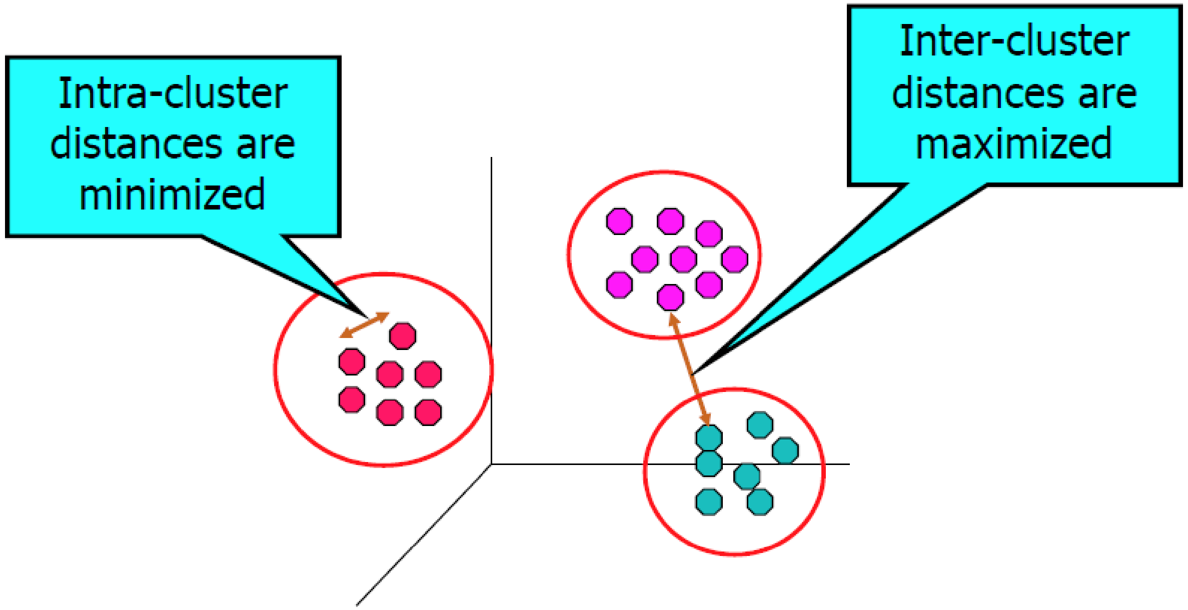
Cluster Analysis

2017. 5. 26.

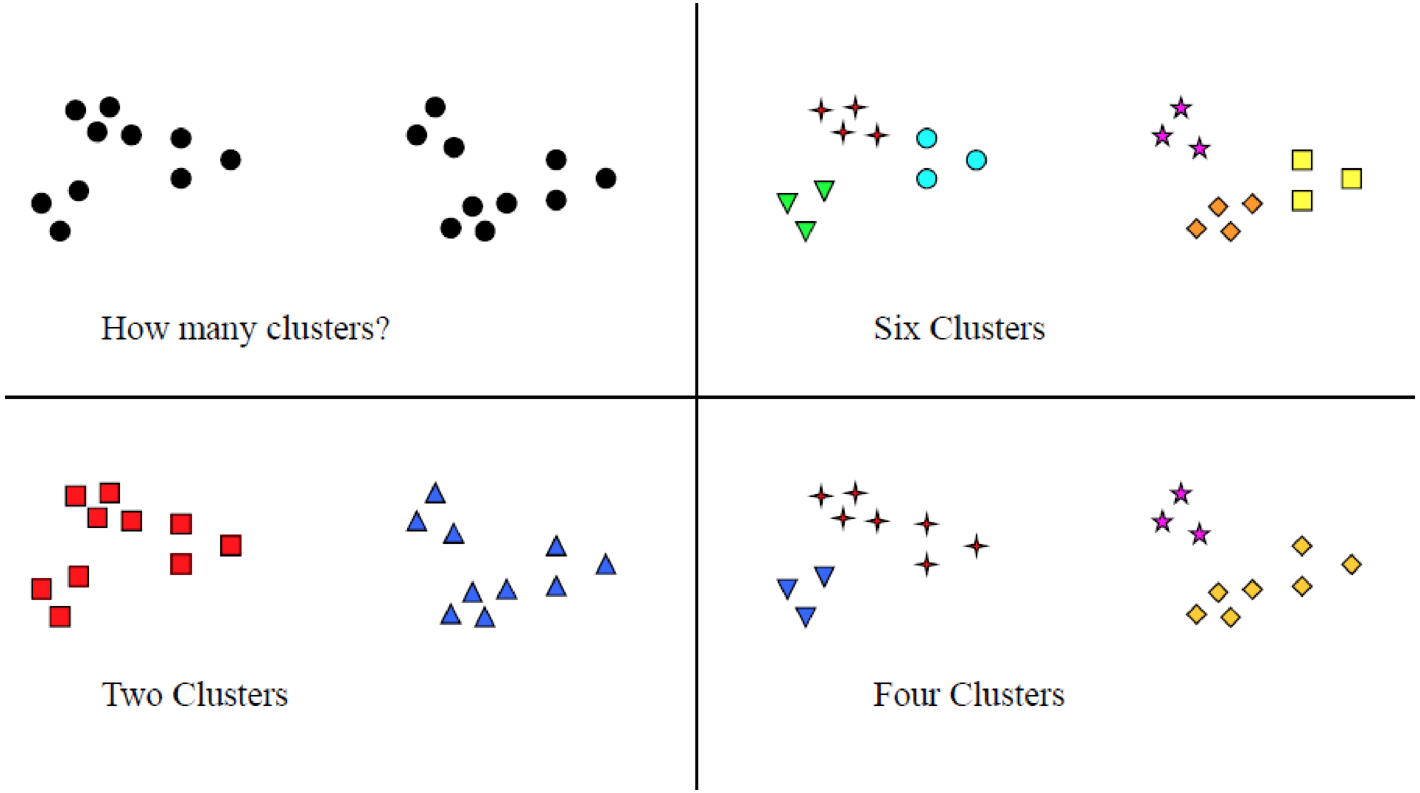
김영철

What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



Notion of a Cluster can be Ambiguous

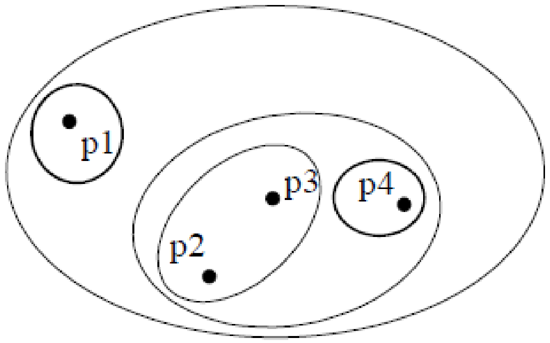


Types of Clustering

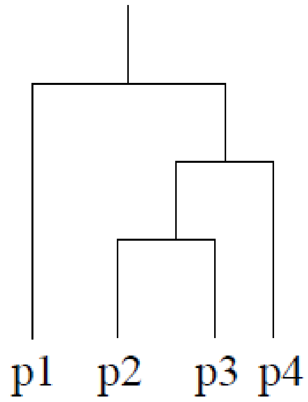
- Partitional Clustering
  - A division of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset

- Hierarchical Clustering
  - A set of nested clusters organized as a hierarchical tree

Hierarchical Clustering



Traditional Hierarchical Clustering



Traditional Dendrogram

Clustering Algorithms

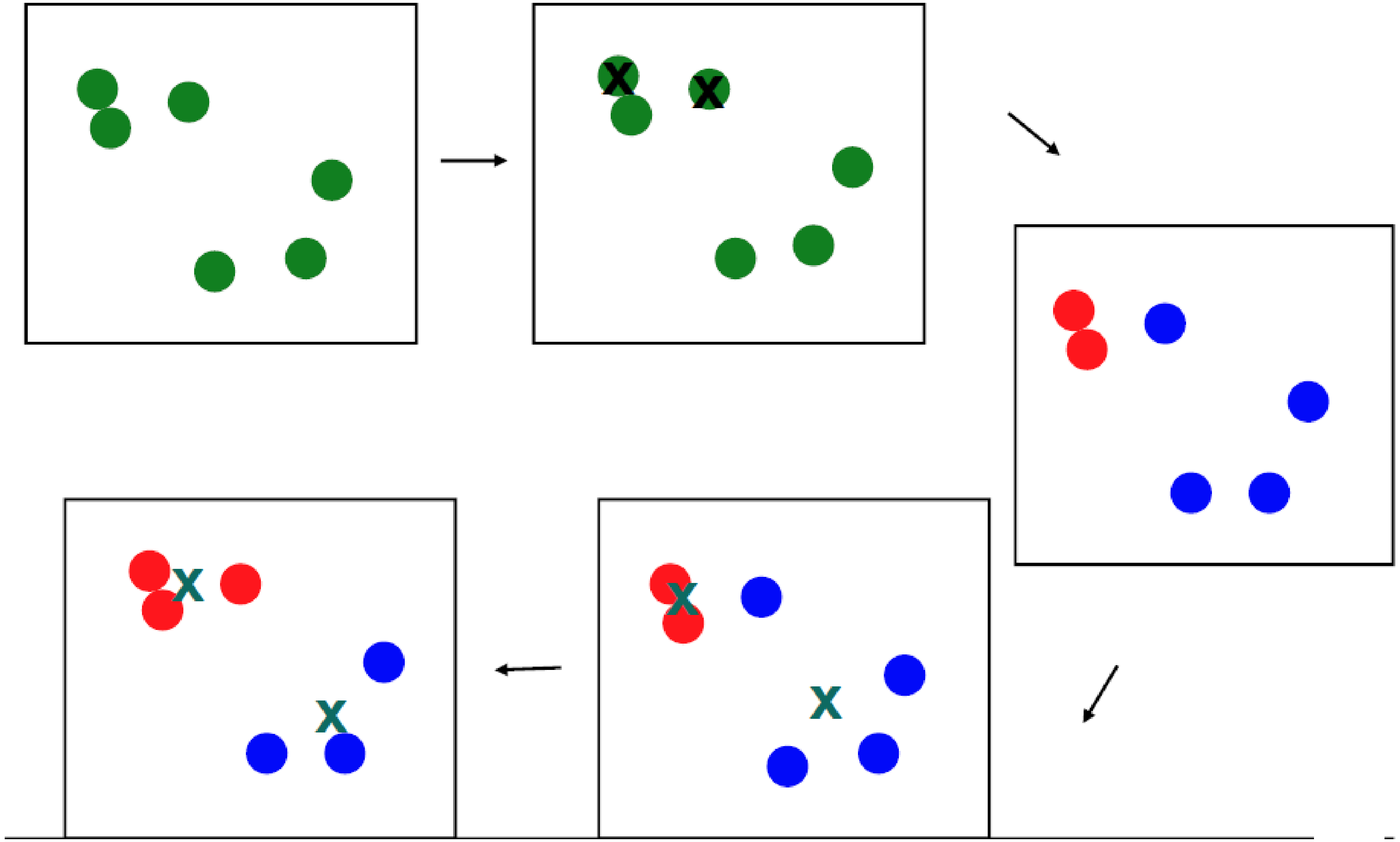
- K-means
- Hierarchical clustering
- ~~Density-based clustering~~

K-means Clustering

- K개의 centroid를 선택
  - 각 점 마다 가장 가까운 centroid에 모아서 K개의 클러스터 형성
  - centroid 다시 계산
  - centroid가 바뀌지 않을 때까지 반복

Select K points as the initial centroids.
repeat
    Form K clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change

K-means Clustering Example



Evaluating K-means Clusters

- Most common measure for the quality of a clustering is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

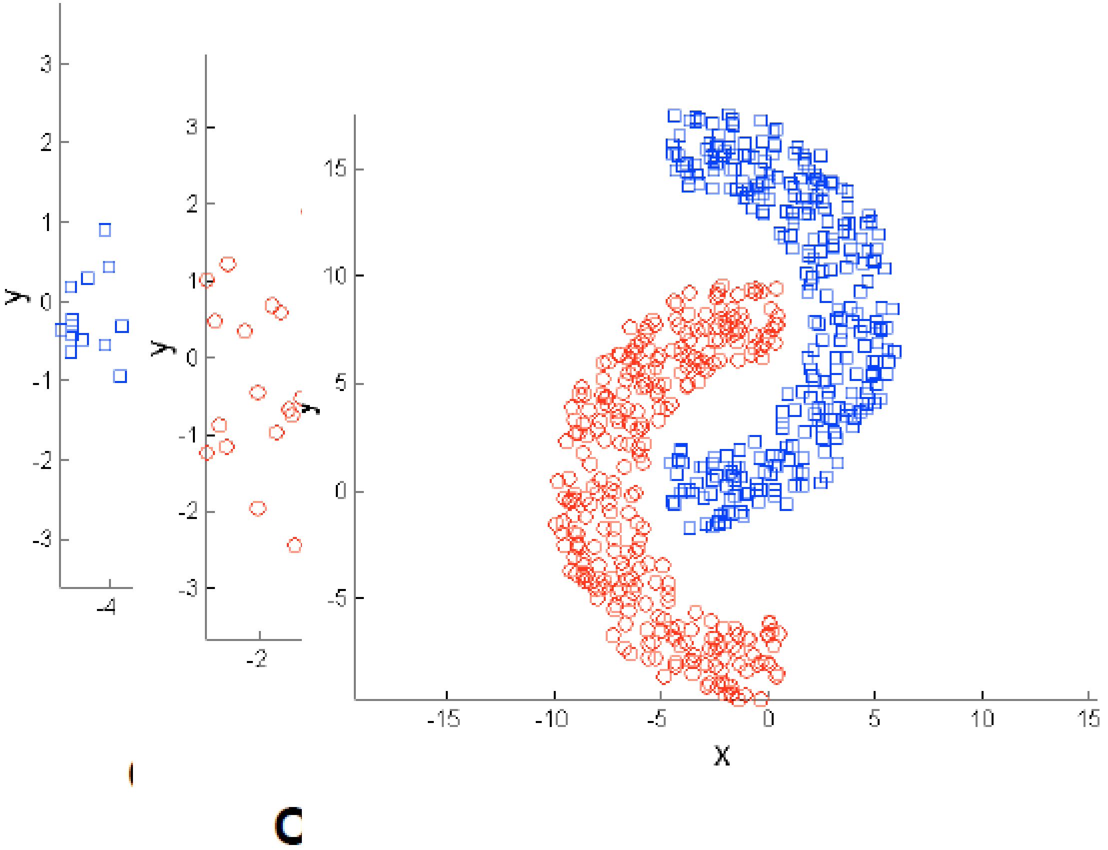
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

  - x is a data point in cluster C_i and m_i is the representative point for cluster C_i
    - can show that m_i corresponds to the center (mean) of the cluster
  - Given two clusters, we can choose the one with the smallest error

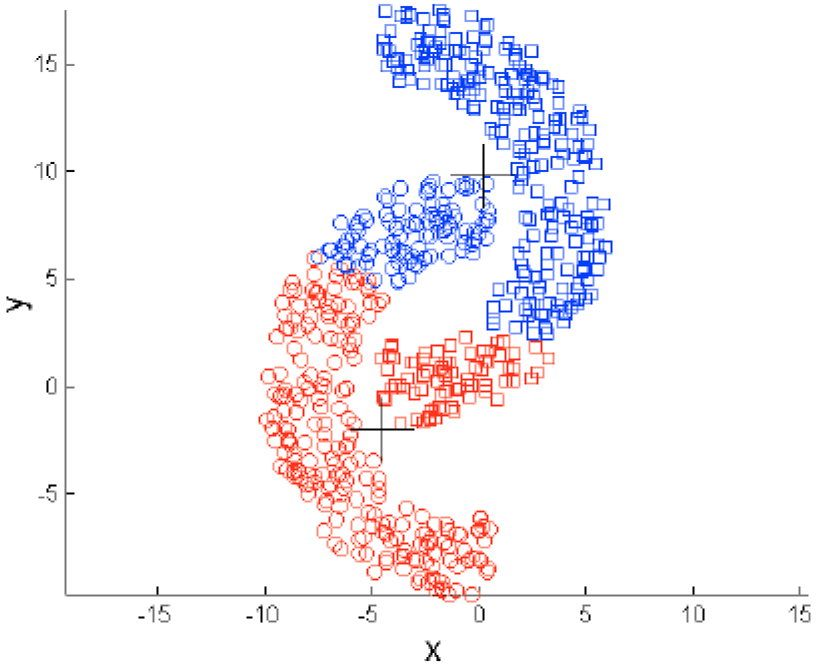
Updating Centers Incrementally

- In the basic K-means algorithm, centroids are updated after all points are assigned to a centroid.
- An alternative is to update the centroids after each assignment.
  - More expensive
  - Introduces an order dependency
  - Never get an empty cluster

Limitations of K-means



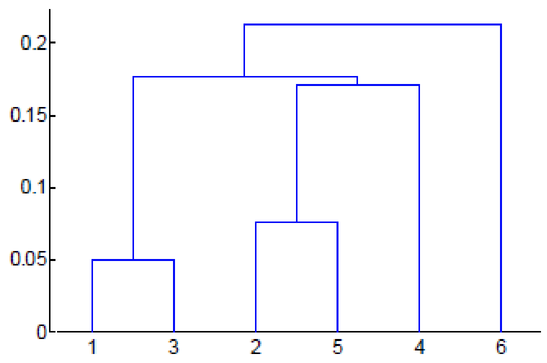
Original Points

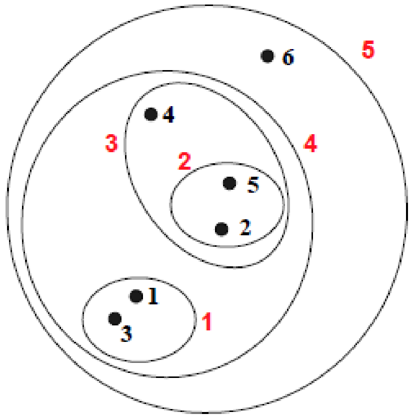


K-means (2 Clusters)

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





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 inly one cluster left
    - bottom-up

  - Divisive
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point
    - top-down

Agglomerative Clustering Algorithm

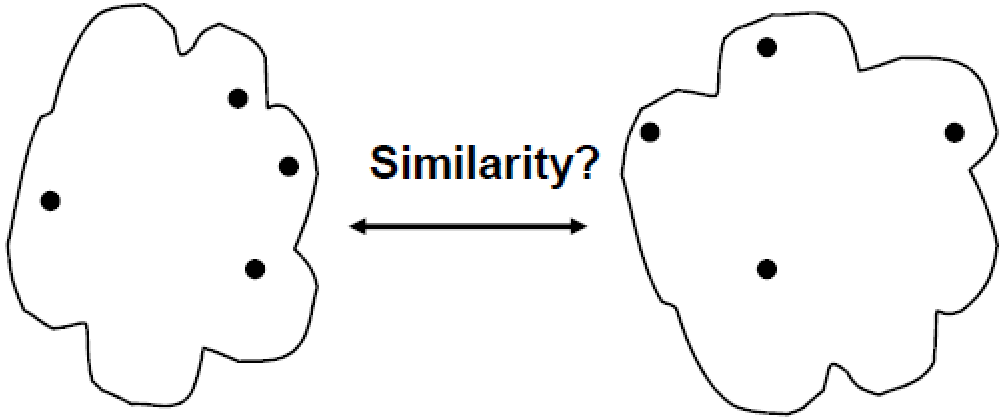
- Basic algorithm is straightforward

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

  - 가장 밀접한 두 클러스터를 병합하고, matrix 수정
  - 하나의 클러스터가 될 때까지 수행

두 클러스터가 밀접한 관계를 갖는다는 기준은?

How to Define Inter-Cluster Similarity

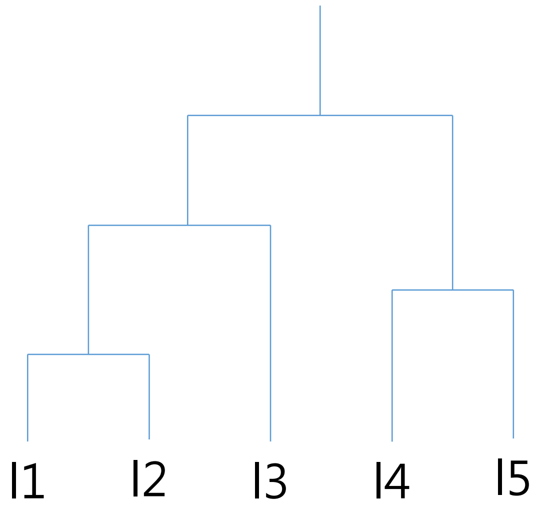


- MIN, MAX, Group Average … etc.

Cluster Similarity : MIN (Single Link)

- Similarity of two clusters is based on the two most similar points in the different 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



Cluster Similarity : MIN (Single Link)

- Similarity of two clusters is based on the two most similar points in the different clusters

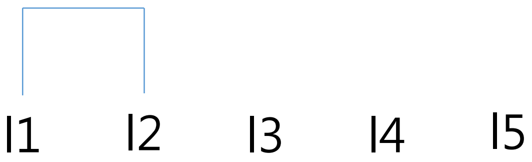
	I1	I2	I3	I4	I5
I1	1.00	$\boxed{0.90}$	~~0.10~~	0.65	~~0.20~~
I2	0.90	1.00	0.70	~~0.60~~	0.50
I3	~~0.10~~	0.70	1.00	0.40	0.30
I4	0.65	~~0.60~~	0.40	1.00	0.80
I5	~~0.20~~	0.50	0.30	0.80	1.00

I1 I2 I3 I4 I5

Cluster Similarity : MIN (Single Link)

- Similarity of two clusters is based on the two most similar points in the different clusters

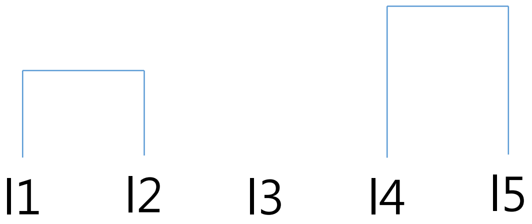
	I1 I2	I3	I4	I5
I1 I2	1.00	0.70	0.65	~~0.50~~
I3	0.70	1.00	0.40	~~0.30~~
I4	0.65	0.40	1.00	$\boxed{0.80}$
I5	~~0.50~~	~~0.30~~	0.80	1.00



Cluster Similarity : MIN (Single Link)

- Similarity of two clusters is based on the two most similar points in the different clusters

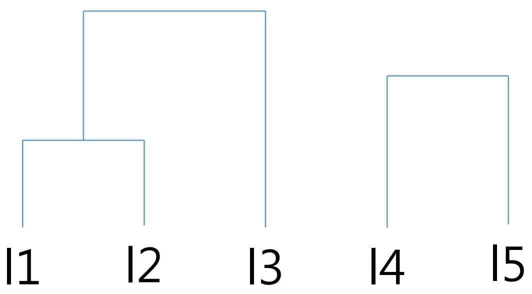
	I1, I2	I3	I4, I5
I1, I2	1.00	0.70	0.65
I3	0.70	1.00	~~0.40~~
I4, I5	0.65	~~0.40~~	1.00



Cluster Similarity : MIN (Single Link)

- Similarity of two clusters is based on the two most similar points in the different clusters

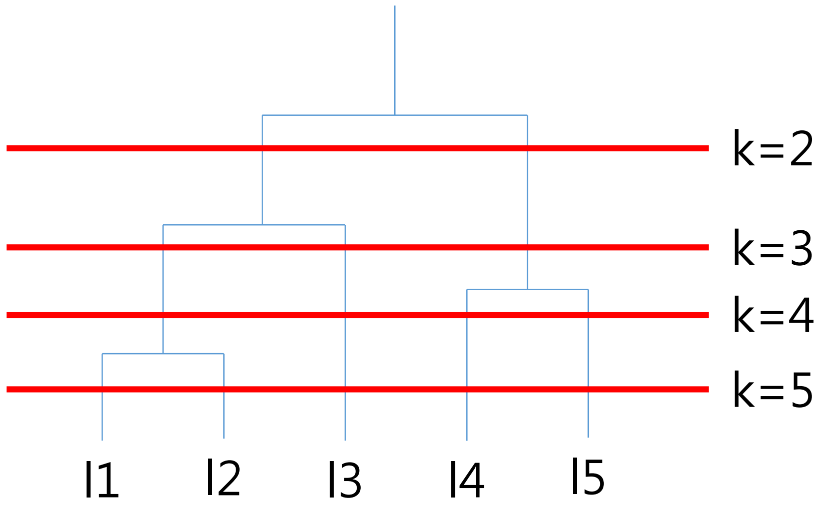
	I1 I2 I3	I4 I5
I1 I2 I3	1.00	0.65
I4 I5	0.65	1.00



Cluster Similarity : MIN (Single Linkage)

- Similarity of two clusters is based on the two most similar points in the different clusters

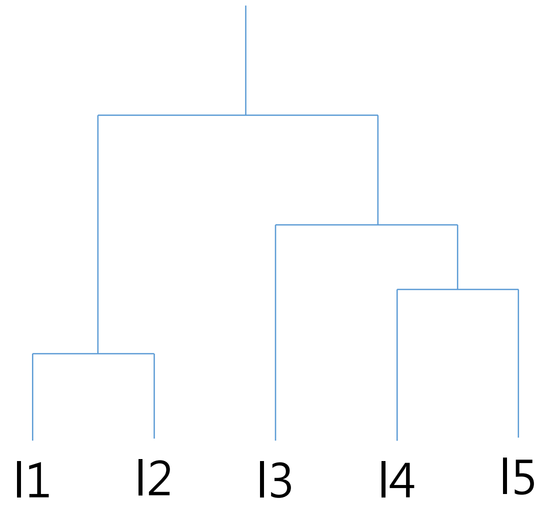
	I1	I2	I3	I4	I5
I1					
I2					
I3			1.00		
I4					
I5					



Cluster Similarity : MAX (Complete Linkage)

- Similarity of two clusters is based on the two least similar points in the different clusters

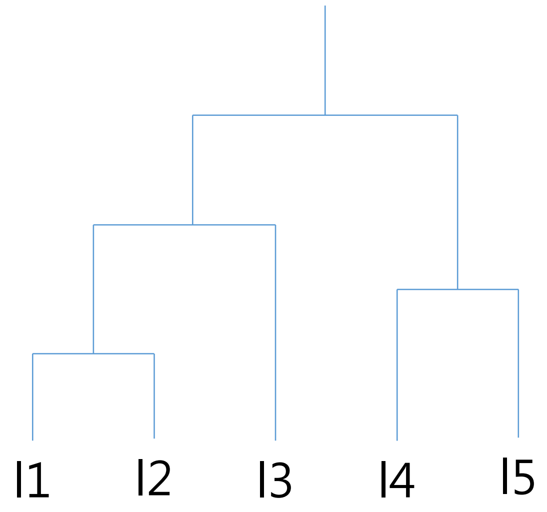
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



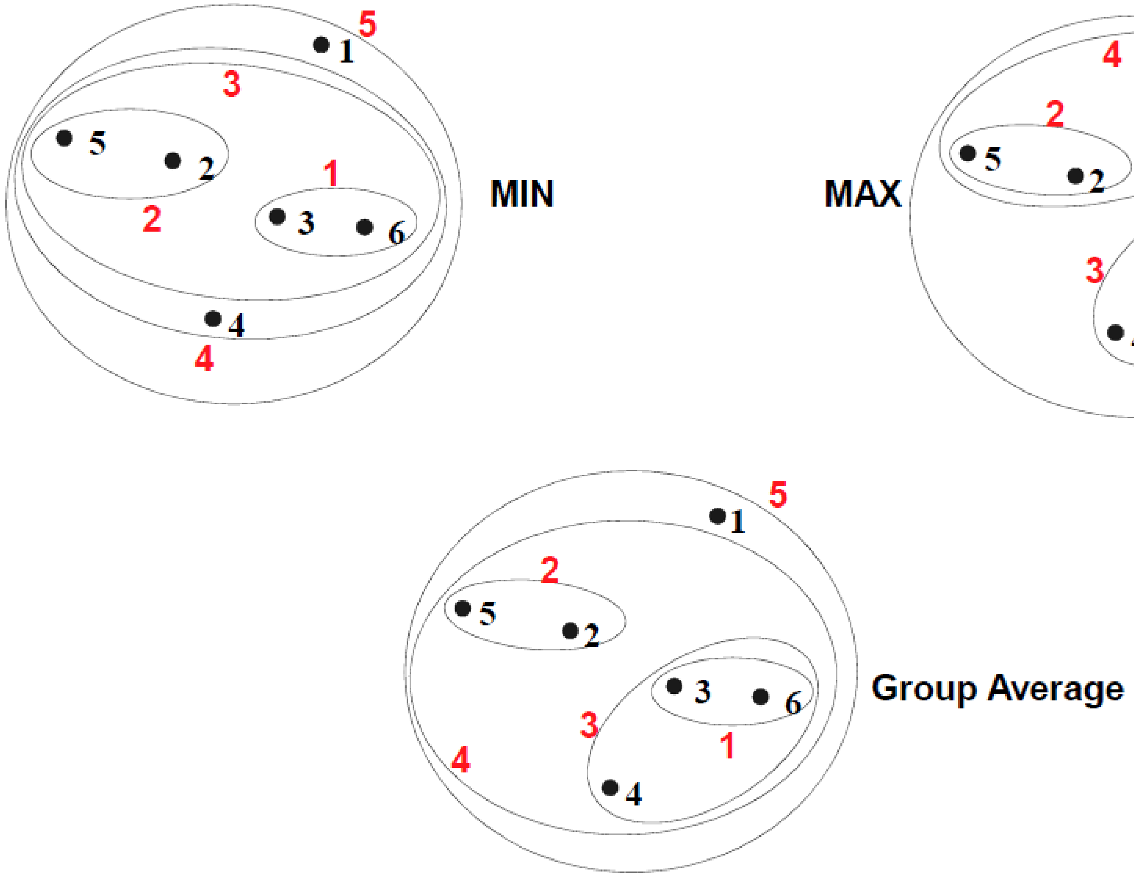
Cluster Similarity : Group Average

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

	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

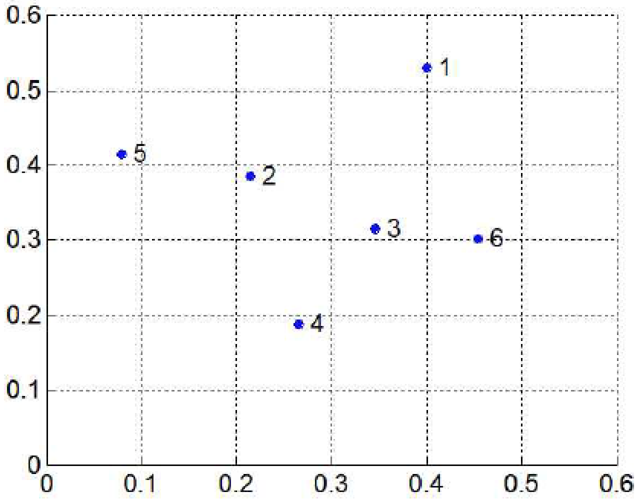


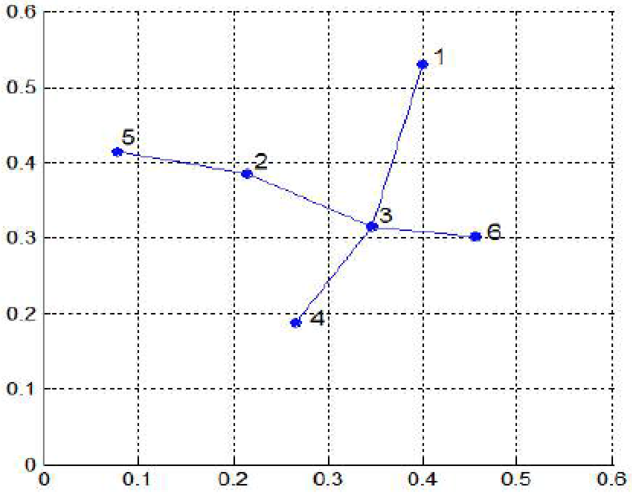
Hierarchical Clustering : Comparison



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
  - Add q to the tree and put an edge between p and q





MST : Divisive Hierarchical Clustering

- 거리가 긴 점 사이의 edge부터 끊어가면서 클러스터를 생성
- 클러스터들이 모두 singleton 클러스터가 될 때까지 반복

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