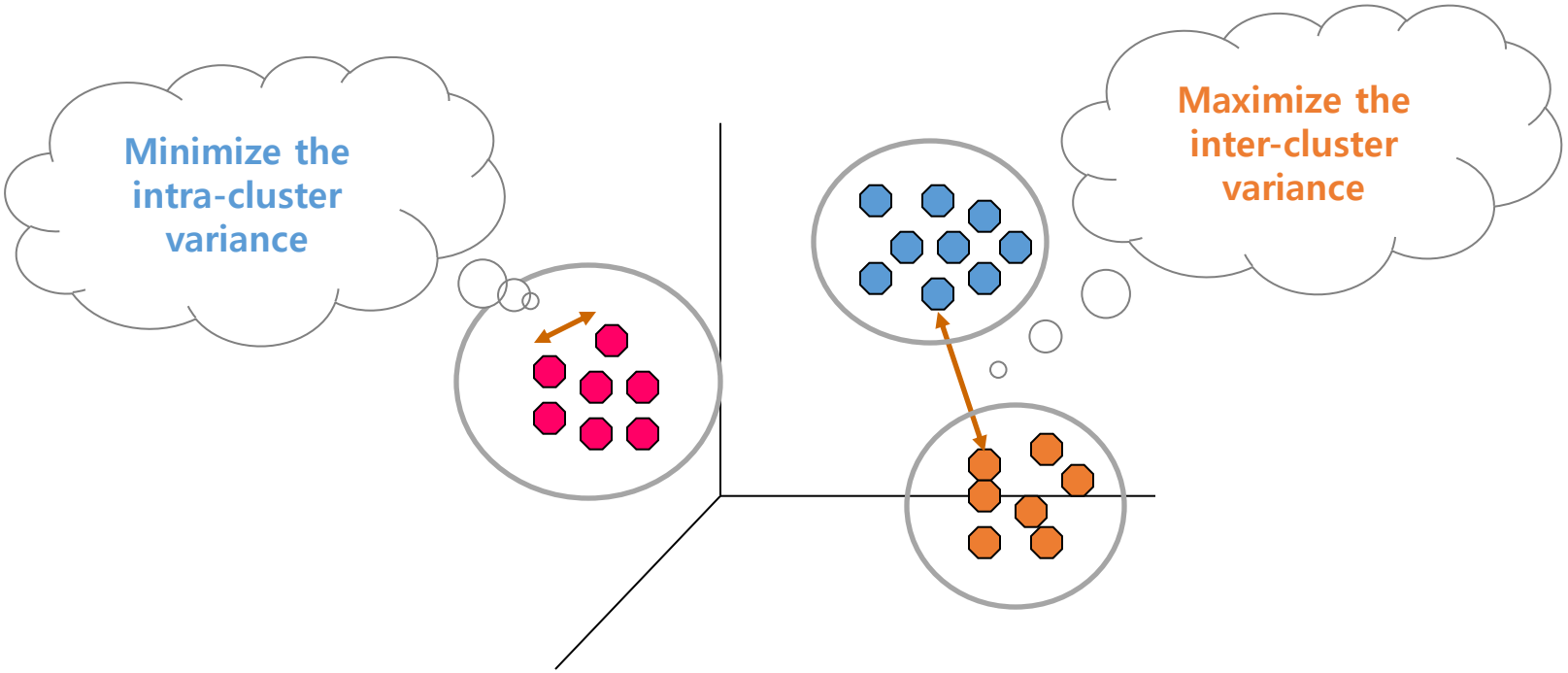
Clustering

Taehoon Ko (thoon.koh@gmail.com)

군집화

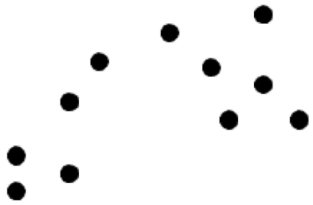
- **군집화는 데이터에서 비슷한 객체들을 하나의 그룹으로 묶는 것**
  - 각 객체들이 어떤 군집으로 할당되어야 하는가에 대한 정보(y)가 없기 때문에 unsupervised 알고리즘에 해당
  - 그러므로 군집화 방법들은 각 객체들의 유사도(거리) 정보를 기반으로 작동
  - Find **groups of objects** such that the **objects in a group will be similar (or related) to one another** and **different from (or unrelated to) the objects in other groups**

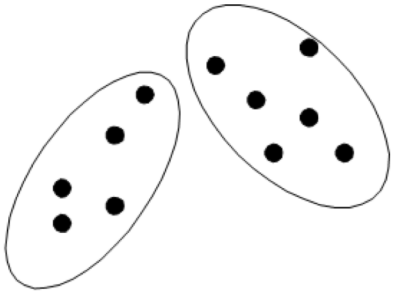


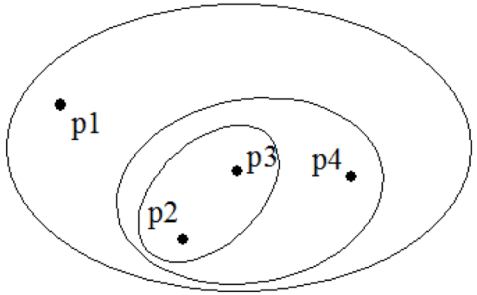
Clustering Overview: Types

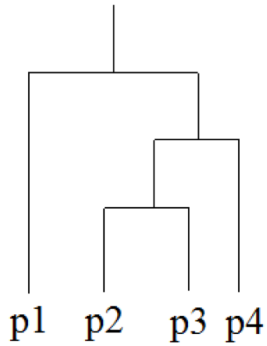
- **Partitional clustering**
  - Divide data into non-overlapping subsets such that each data object is in exactly one subset
  - ex. k-means clustering

- **Hierarchical clustering**
  - A set of nested clusters organized as a hierarchical tree
  - ex. Agglomerative hierarchical clus tering



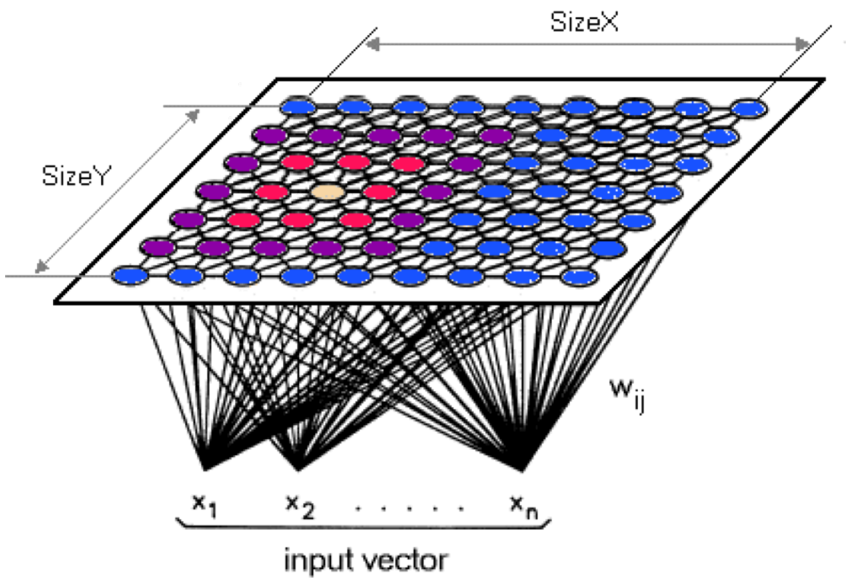




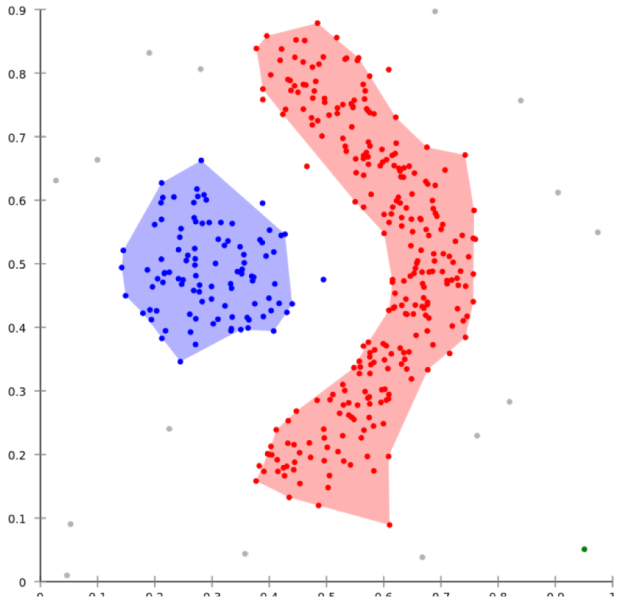


Clustering Overview: Types

- **Self-organizing map (SOM)**
  - Clustering data and ordering of the cluster centers in a two dimensional grid through neural network-based learning methods



- **Density-based clustering**
  - Define clusters as areas of higher density than the remainder of the data set
  - ex. Density-based spatial clustering of applic ations with noise (DBSCAN)



군집화

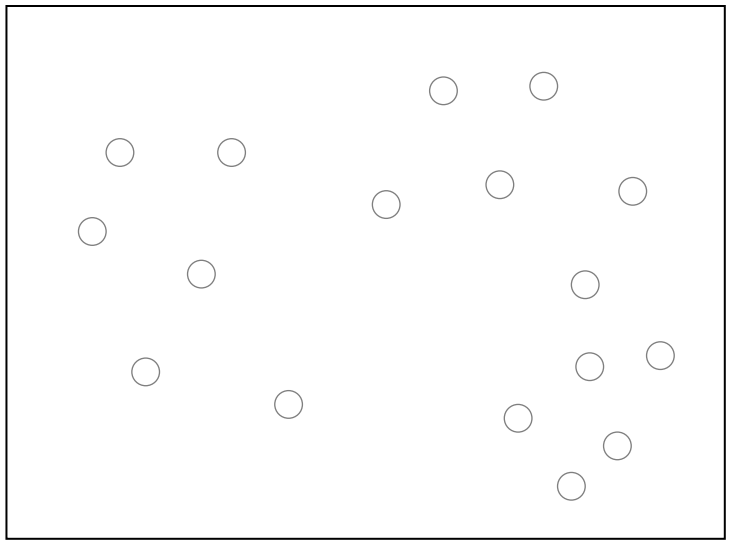
- **군집화는 (1) 객체/그룹 간의 유사도 정의와 (2) 그룹화의 방식에 따라 다양한 알고리즘이 제안됨**
  - (spherical) K-means
  - Hierarchical clustering
  - DBSCAN
  - Community detection
  - …
  - **유사도를 잘 정의**하기 위하여 적절한 representation이 필요하기도 함

K-means clustering

- **유사도**
  - n개의 데이터 X에 대하여 두 데이터 x_i, x_j간에 정의되는 임의의 거리 $d(x_i, x_j)$
    - 유클리디언, 코싸인 등 벡터에서 정의되는 모든 거리 척도

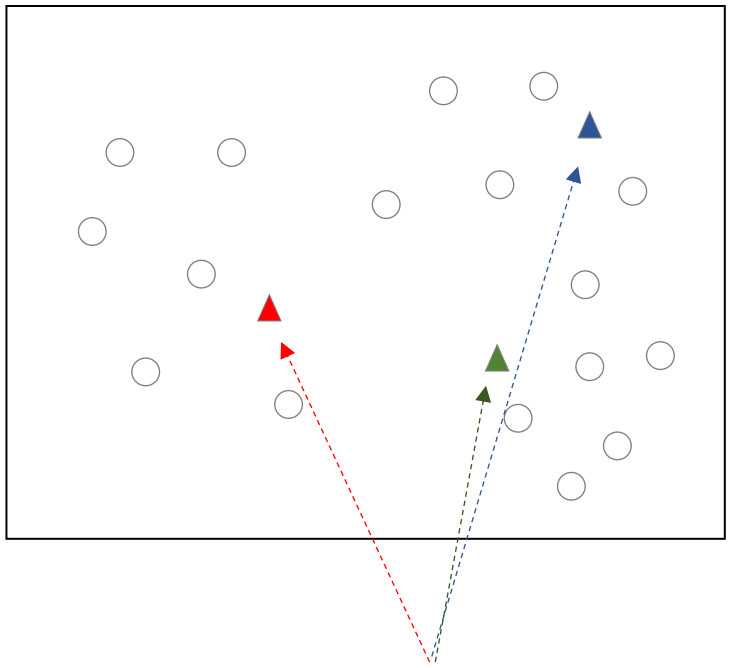
- **그룹화의 방식**
  - 그룹의 개수는 k개라고 가정
  - 각 그룹을 centroid vector (평균 벡터)로 표현한 뒤, 이를 업데이트 하는 방식

K-means clustering



0. Data

K-means clustering

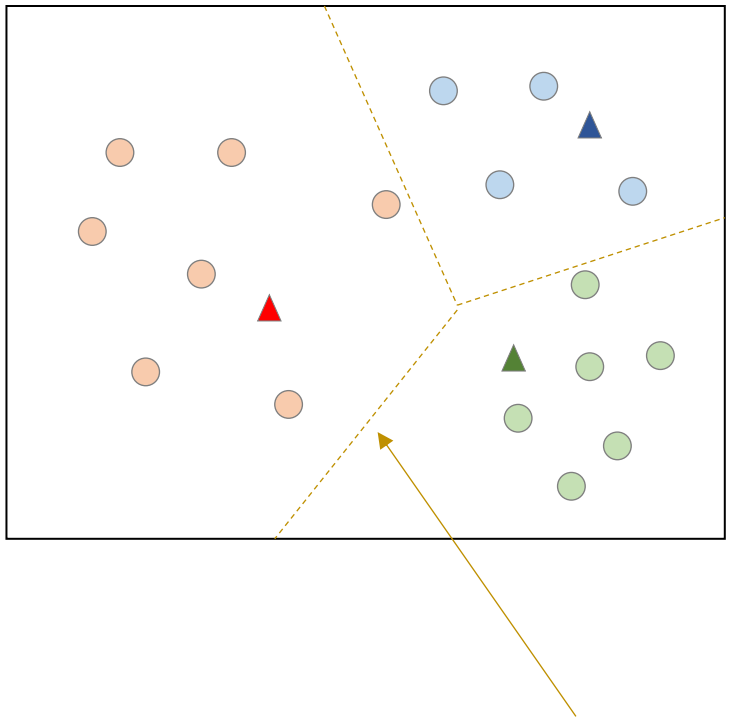


1. Initialize

k=3 이라 가정하면 (데이터 입력 공간에서)
3개의 점을 임의로 선택

데이터에는 존재하지 않는 가상의 centroids

K-means clustering



1. Initialize

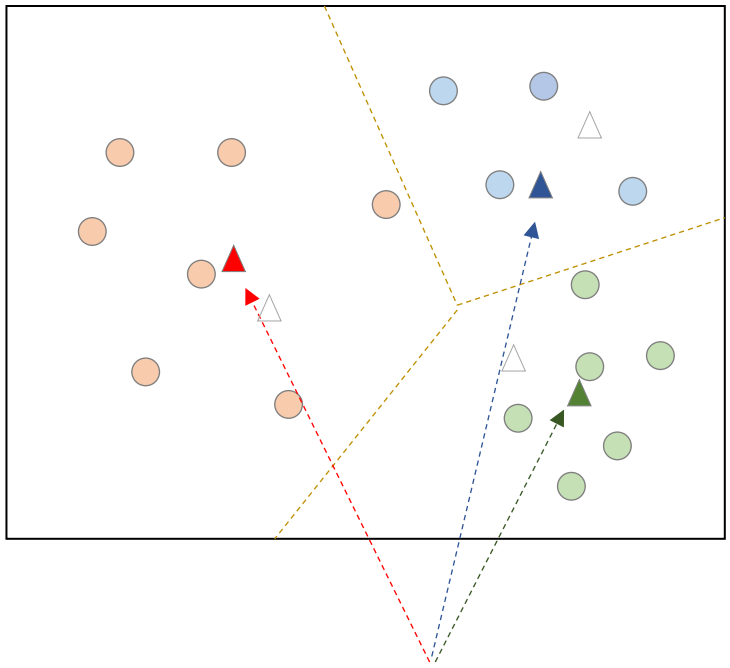
k=3 이라 가정하면 3개의 점을 임의로 선택

2. Assign (epoch=0)

모든 점을 *k*개의 centroid 중 가장 가까운 점의 색깔(label)로 할당

*k*개의 centroids에 의하여 분할된 공간의 경계면으로,
Voronoi partition, Voronoi diagram이라 부름

K-means clustering



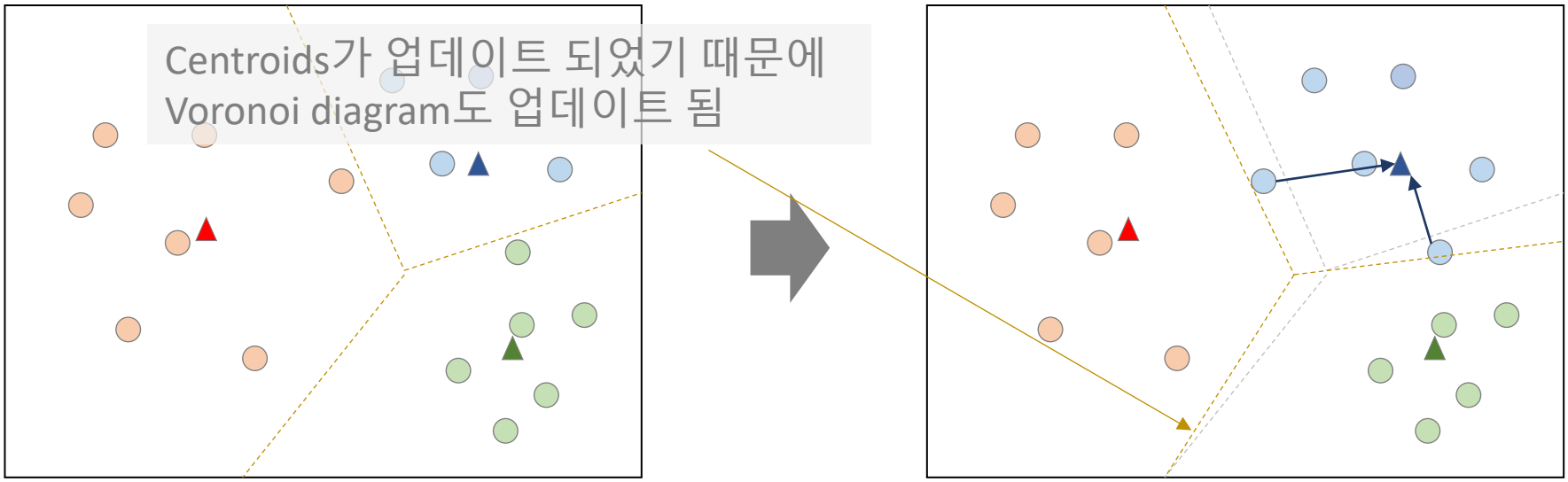
1. Initialize
k=3 이라 가정하면 3개의 점을 임의로 선택

2. Assign (epoch=0)
모든 점을 *k*개의 centroid 중 가장 가까운 점의 색깔(label)로 할당

3. Update centroid (epoch=0)
같은 색깔(label) 점들의 평균값을 가상의 centroids로 설정 ➔ centroid의 업데이트

데이터에는 존재하지 않는 가상의 centroids

K-means clustering



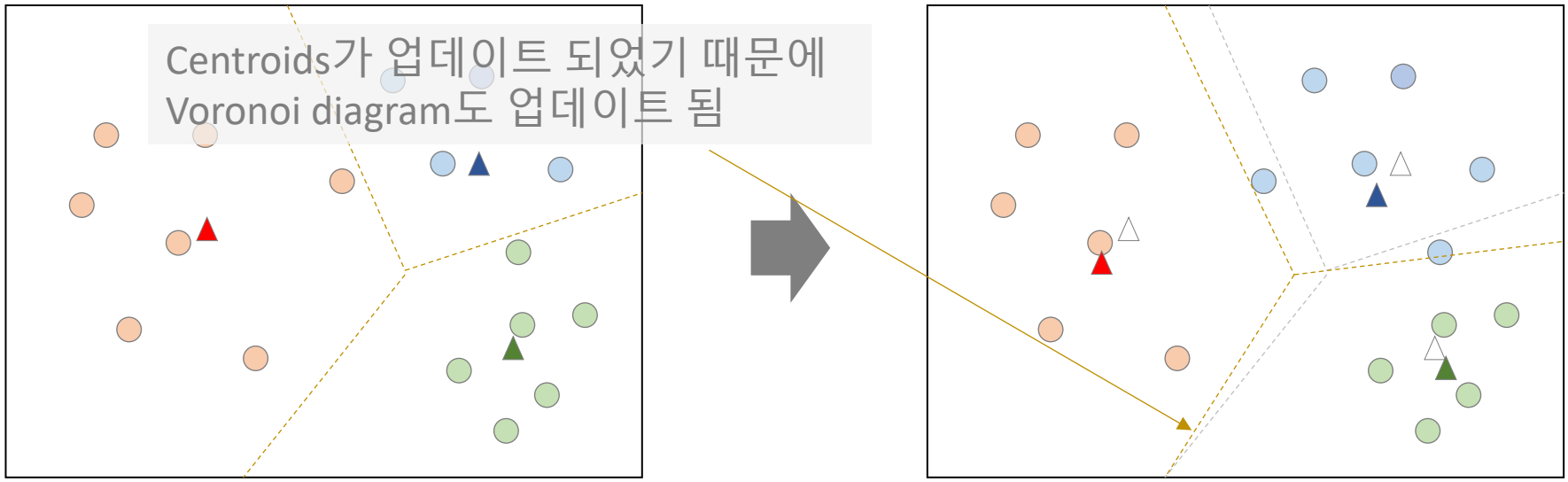
1. Initialize

k=3 이라 가정하면 3개의 점을 임의로 선택

2. Assign (epoch=1)

모든 점을 업데이트 된 centroids 중 가장 가까운 점으로 할당

K-means clustering



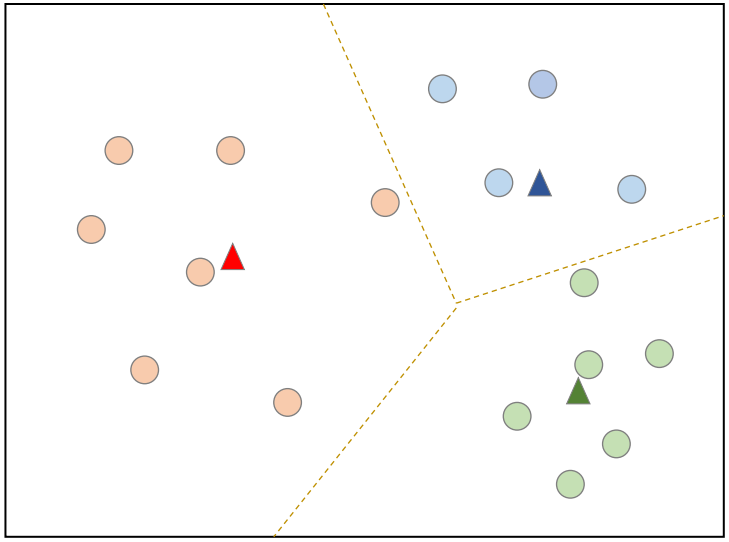
1. Initialize
k=3 이라 가정하면 3개의 점을 임의로 선택

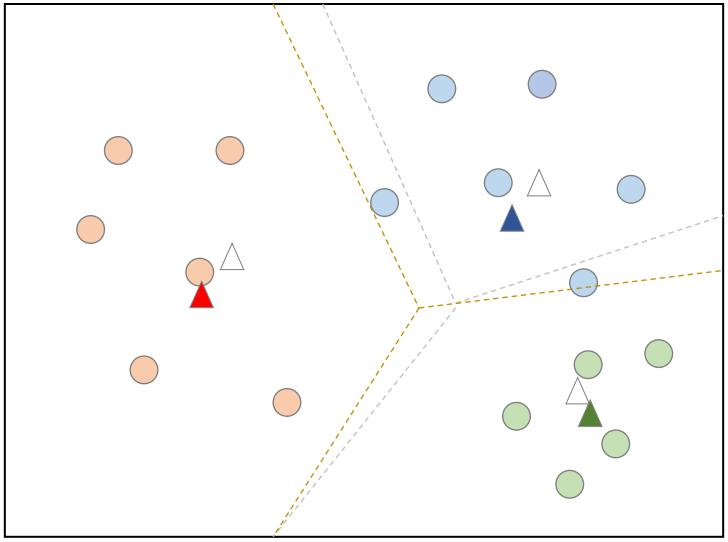
2. Assign (epoch=1)
모든 점을 업데이트 된 centroids 중 가장 가까운 점으로 할당

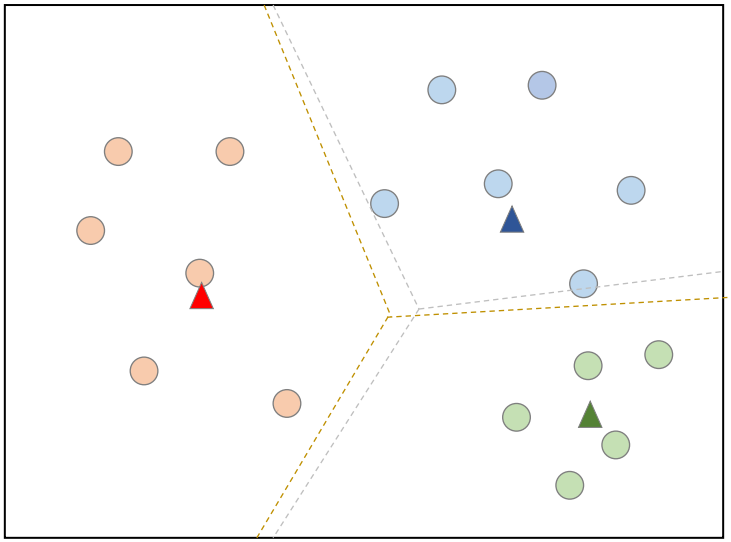
3. Update centroid (epoch=1)
색깔이 바뀐 점이 있기 때문에 Centroid를 다시 업데이트

알고리즘이 종료 될 때까지 2, 3을 반복

K-means clustering







1. Initialize

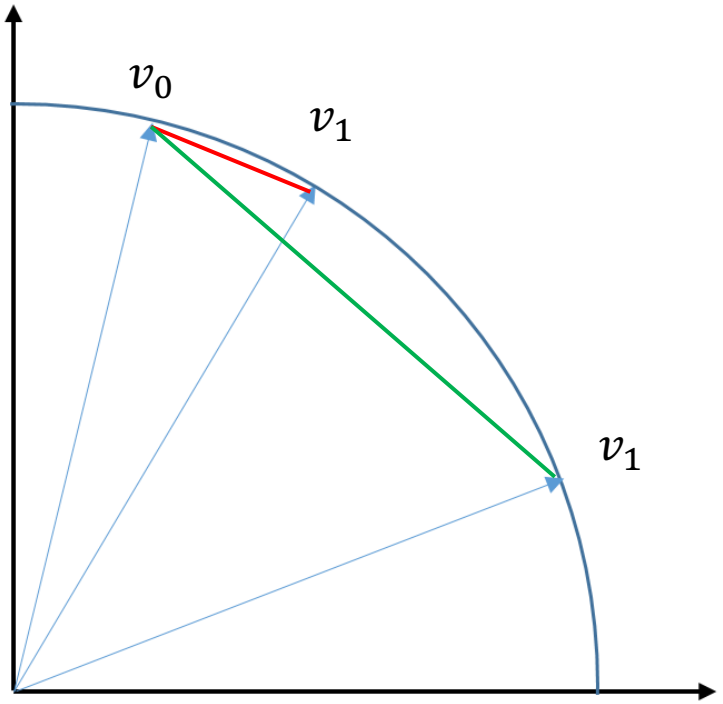
k=3 이라 가정하면 3개의 점을 임의로 선택

2. Assign (epoch=2)

모든 점을 가장 가까운 centroids로 할당하여도 색깔이 변하지 않으므로 알고리즘 종료

Spherical K-means clustering

- **Spherical *k*-means 알고리즘은 유클리디언이 아닌 코싸인을 거리 척도로 이용하는 *k*-means 알고리즘**
  - 길이가 같은 벡터에서 코싸인 거리가 짧을수록 유클리디언 거리에 근사함
    - $arc(v_i, v_j) = r * \theta(v_i, v_j)$
    - $arc(v_0, v_1) \cong |v_0 - v_1|^2$ when $v_0 \cong v_1$



(Spherical) K-means clustering

- ***k*-means 알고리즘은 다음의 단점이 있다고 알려짐**

1. Sensitive results from Initial points
   - 초기 centroids에 의하여 군집화 결과가 달라짐

2. Ball-shaped clusters
   - 군집의 모양은 centroid를 중심으로 한 구형으로 제한됨 (Voronoi diagram)

3. Sensitive to noise points
   - Centroids와 노이즈와의 거리가 멀 경우, 노이즈에 의해 잘못된 centroid가 학습됨

(Spherical) K-means clustering

1. Sensitive results from Initial points
   - 초기 centroids에 의하여 군집화 결과가 달라짐

   - Clustering ensembles이나 반복 수행으로 해결함.
   - sklearn.cluster.KMeans에서는 반복 수행으로 best results를 return

sklearn.cluster.KMeans¶

class sklearn.cluster. **KMeans** (*n_clusters=8, init='k-means++', n_init=10, max_iter=300, tol=0.0001, precompute_distances='auto', verbose=0, random_state=None, copy_x=True, n_jobs=1, algorithm='auto'*) [source]

n_init : int, default: 10

Number of time the k-means algorithm will be run with different centroid seeds. The final results will be the best output of n_init consecutive runs in terms of inertia.

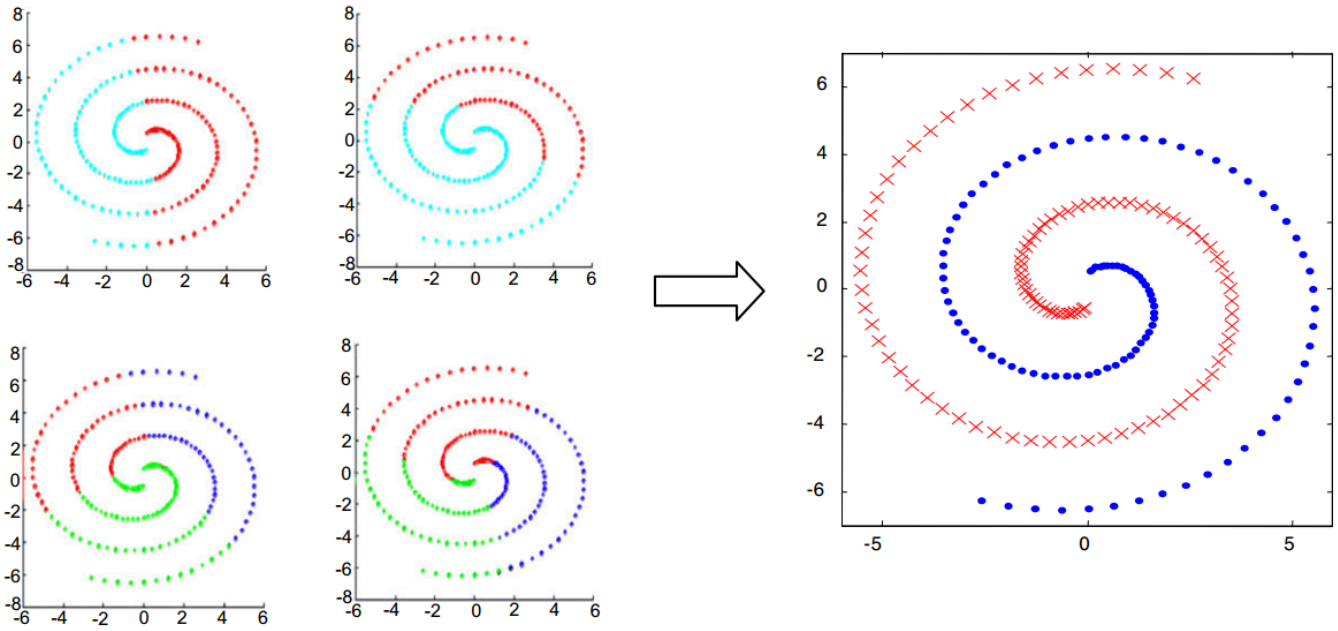
(Spherical) K-means clustering

2. Ball-shaped clusters
   - 군집의 모양은 centroid를 중심으로 한 구형으로 제한됨 (Voronoi diagram)
   - Clustering ensembles으로 구형이 아닌 모양의 군집을 찾을 수 있다고 하지만, 반드시 보장되는 방법은 아님
   - embedding을 통하여 *k*-means에 적합하게 representation을 바꾸거나, 다른 클러스터링 알고리즘을 사용

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern recognition letters, 31(8), 651-666. ISO 690

(Spherical) K-means clustering

- Clustering ensemble
  - 여러 번의 클러스터링 결과를 이용하여 데이터간의 co-occurrence 횟수를 similarity matrix로 이용하여 최종적인 군집화를 수행
  - 데이터의 representation을 co-occurrence vector로 바꾸는 의미이기도 함

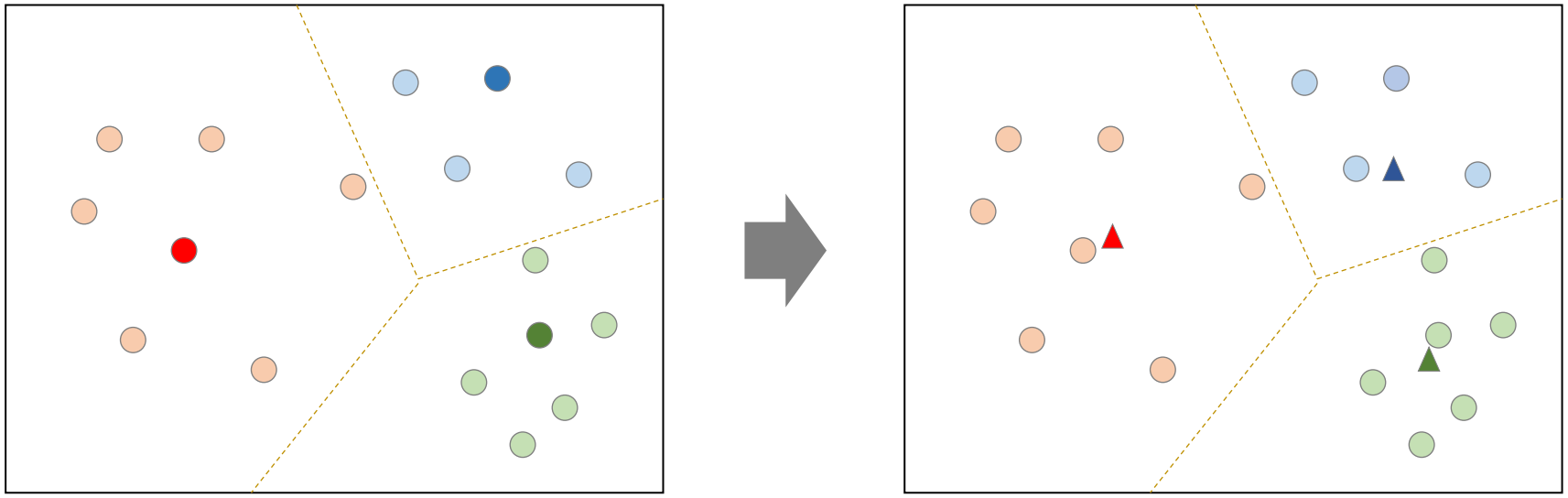


Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern recognition letters, 31(8), 651-666. ISO 690

(Spherical) K-means clustering

3. Sensitive to noise points

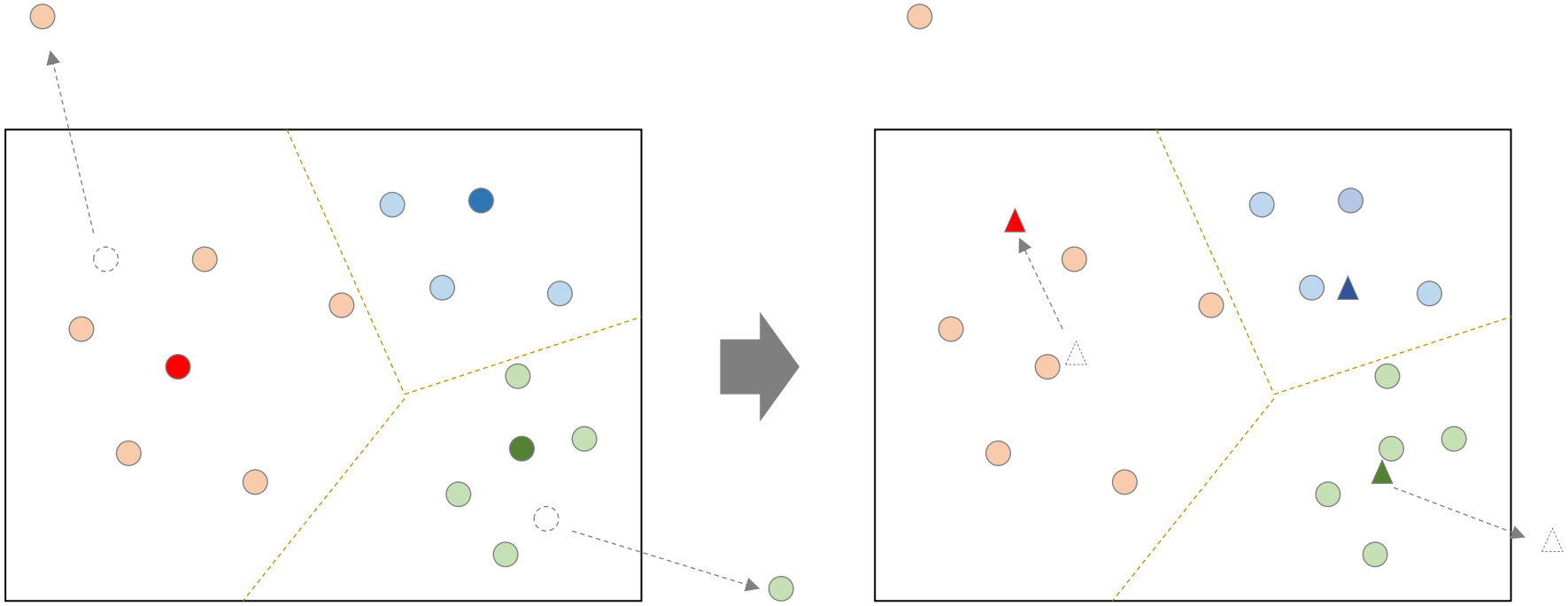
- 모든 점을 반드시 한 개 이상의 군집으로 assign해야 하기 때문에 노이즈 역시 가장 가까운 (하지만 의미적으로는 전혀 가깝지 않은) centroid에 할당이 됨



(Spherical) K-means clustering

3. Sensitive to noise points

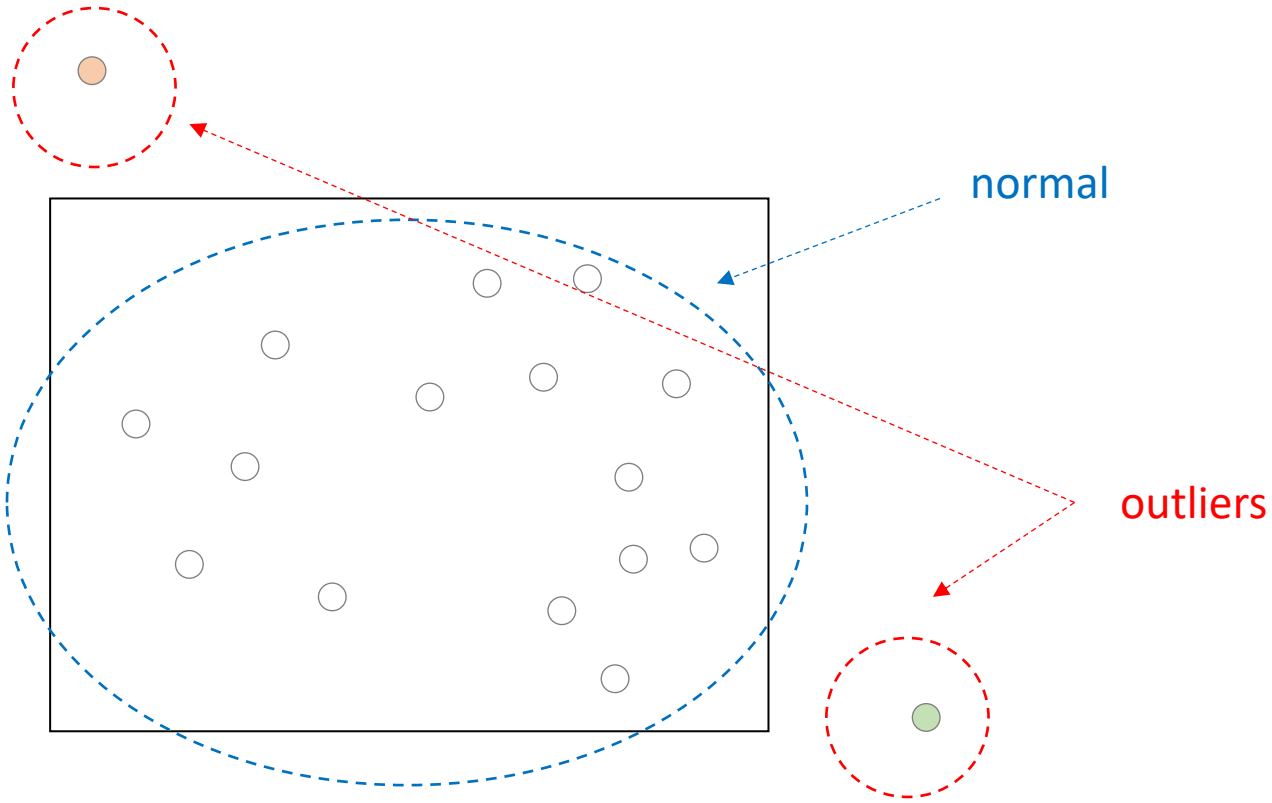
– 몇 개의 노이즈값에 의하여 centroids가 크게 흔들리고, 다음 단계에서 군집의 모양을 제대로 잡지 못함



(Spherical) K-means clustering

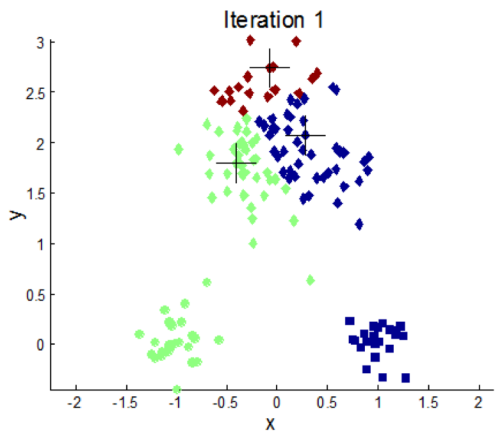
3. Sensitive to noise points

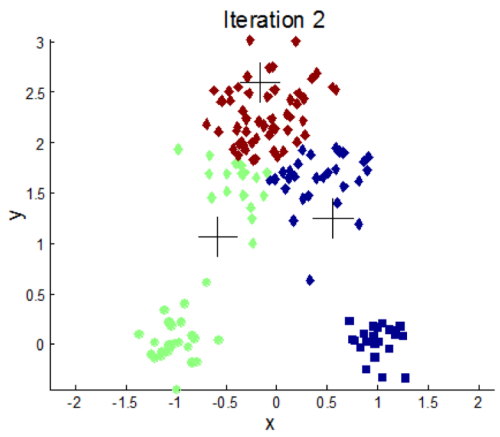
- LOF와 같은 outlier detection 알고리즘으로 데이터의 노이즈를 미리 제거

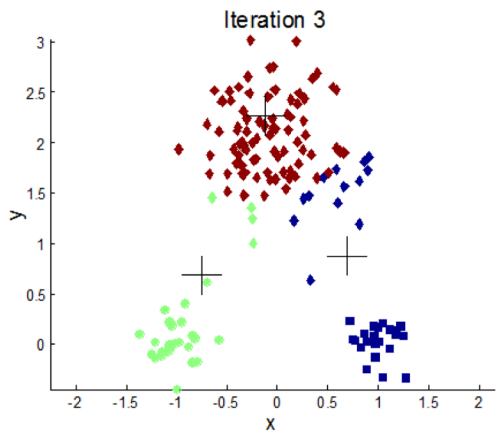


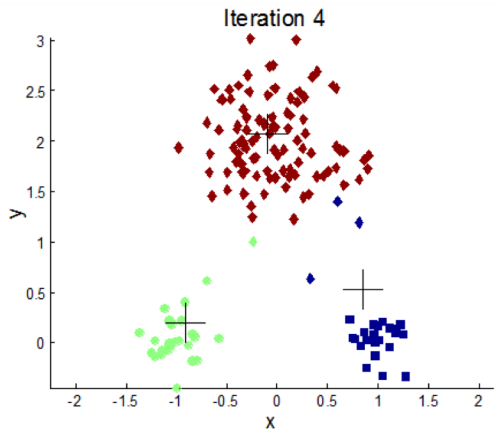
k-means Clustering

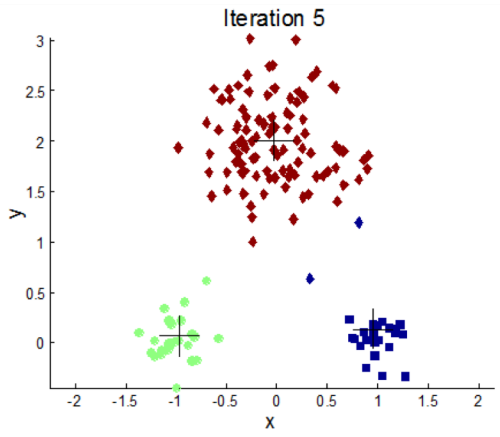
- **Effects of initial centroids**
  - Desirable centroid selection

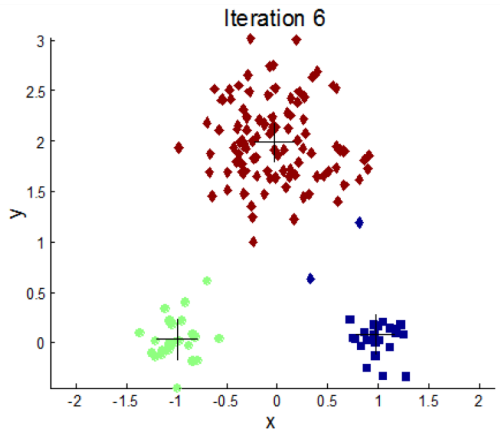






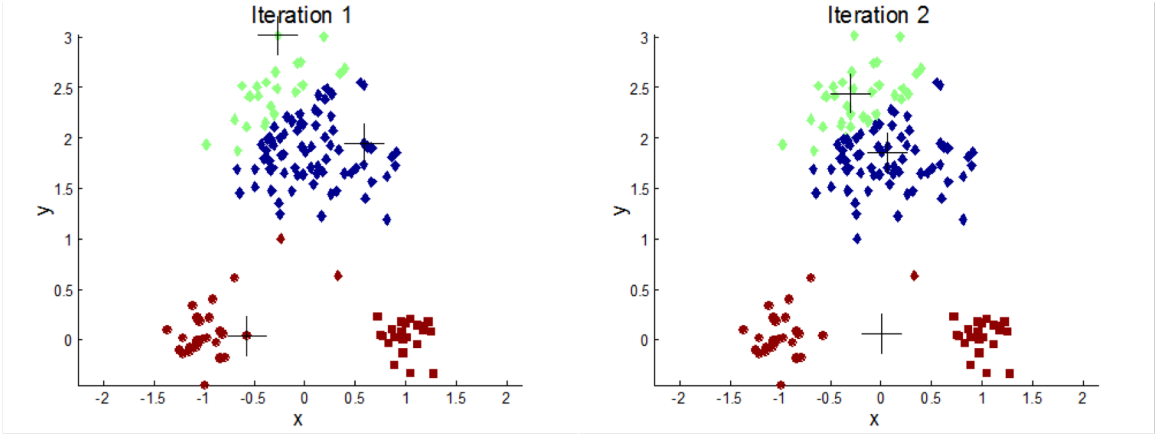


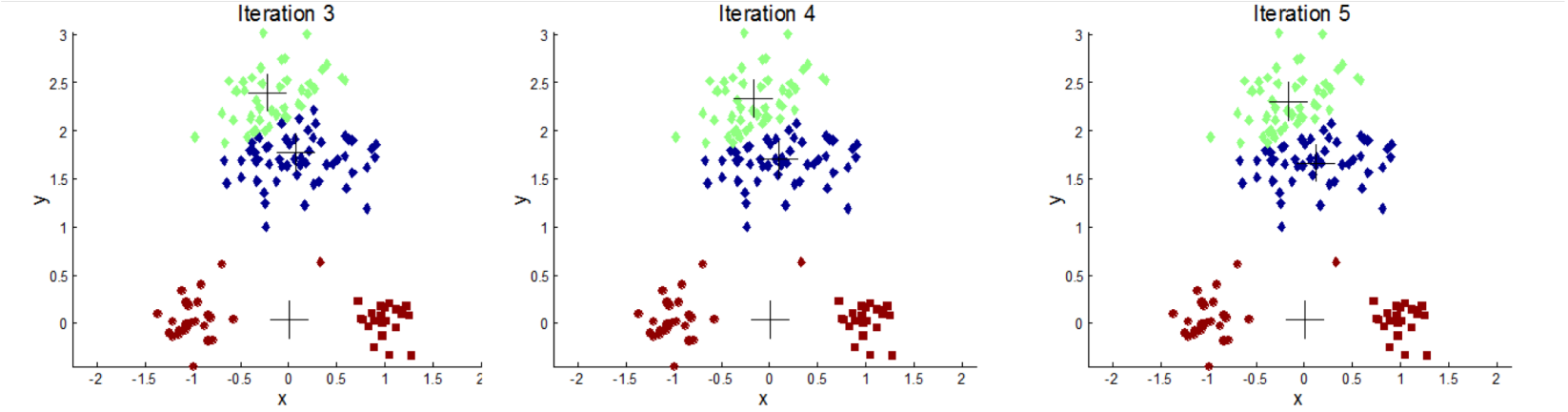




k-means Clustering

- **Effects of initial centroids**
  - Undesirable centroid selection





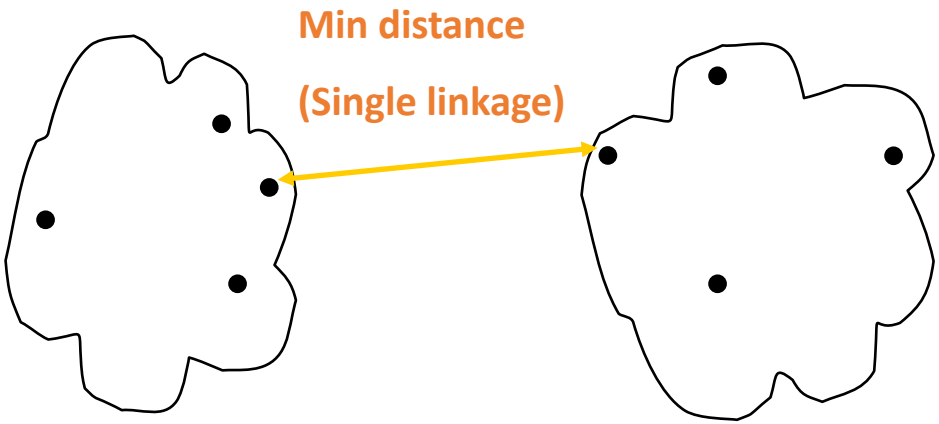
Hierarchical clustering

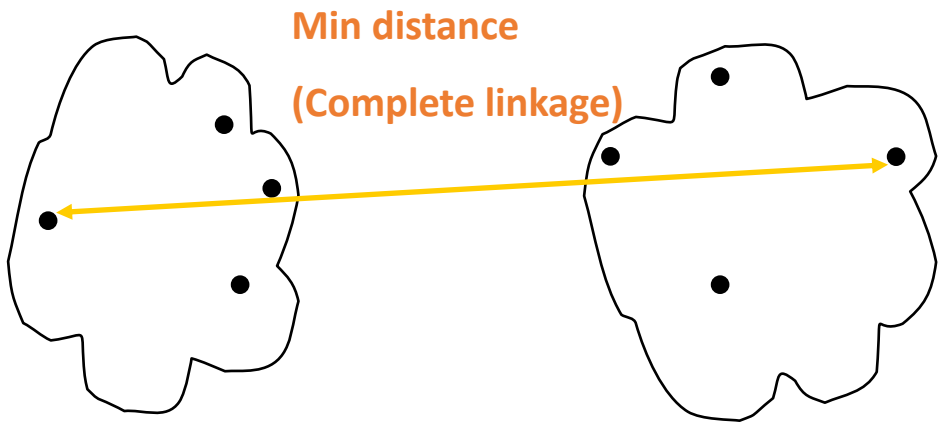
- **유사도**
  - n개의 데이터 X에 대하여 두 데이터 x_i, x_j간에 정의되는 임의의 거리 $d(x_i, x_j)$
    - 그룹 간의 거리는 $d(C_i, C_j)$를 기반으로 정의 (min, max, average등)
    - 하나의 그룹 C_i는 1개 이상의 데이터로 이뤄짐
      (1개의 데이터도 그룹으로 정의 됨)
    - 다양한 방식: single linkage, complete linkage, average linkage, centroid linkage, ward linkage

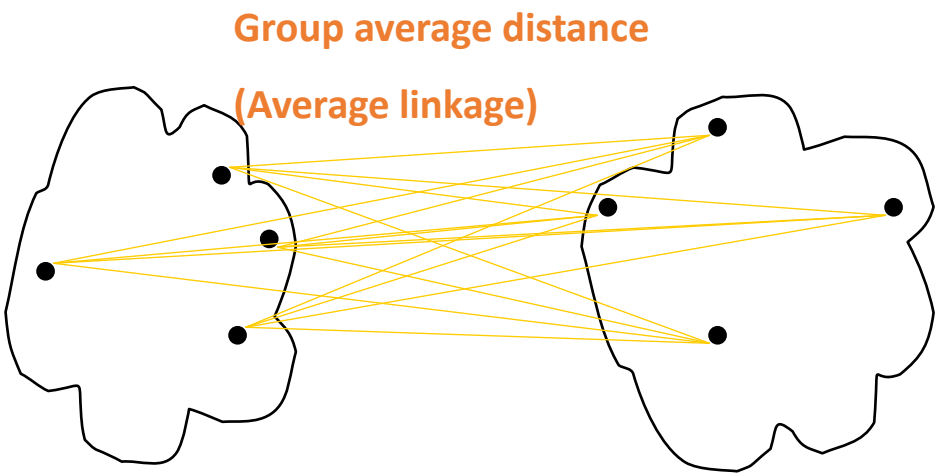
- **그룹화의 방식**
  - 그룹의 수는 정하지 않으며, 거리가 가장 가까운 점들을 하나의 집합으로 묶어감

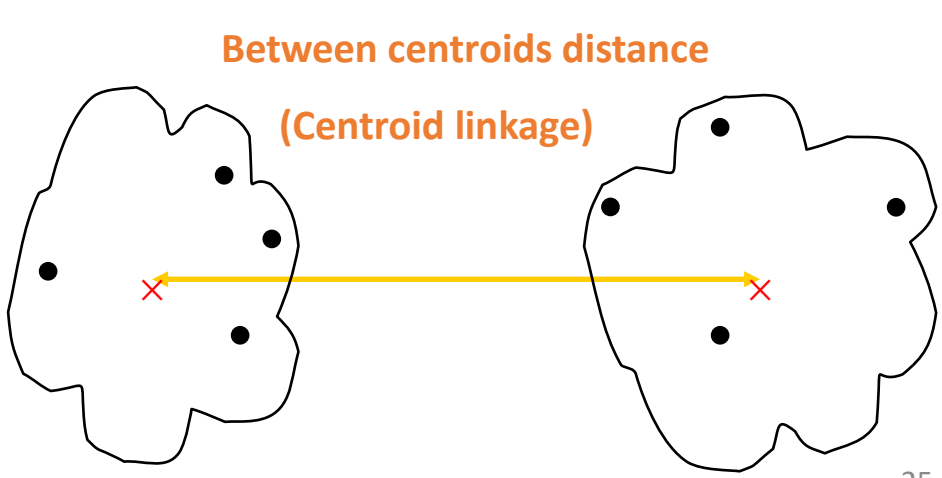
Hierarchical clustering

- 유사도 정의 방법



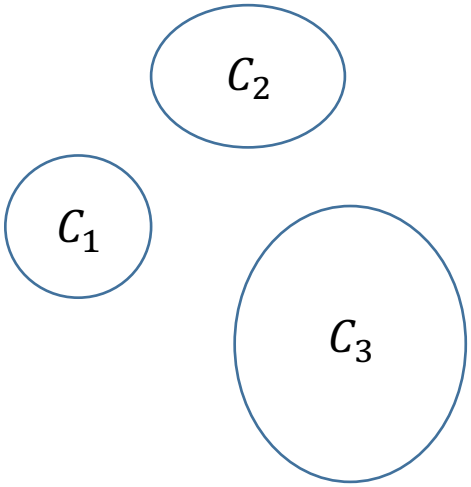






Hierarchical clustering

- **유사도 정의 방법: Ward's method / Ward linkage**
  - 군집을 합치면 군집의 분산 (중심과의 거리 제곱합) 이 커지게 될 것.
  - 분산의 증가량이 가장 작은 방향으로 군집화



$SS(C_1) = 100$ $SS(C_2) = 150$ $SS(C_3) = 250$

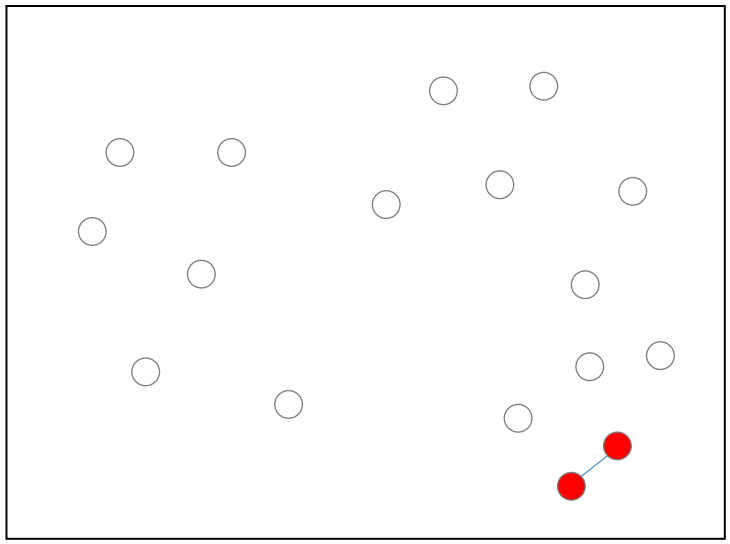
$SS(C_1 \cup C_2) = 300$ $SS(C_1 \cup C_2) - SS(C_1) - SS(C_2) = \mathbf{50}$

$SS(C_1 \cup C_3) = 500$ $SS(C_1 \cup C_3) - SS(C_1) - SS(C_3) = 150$

$SS(C_2 \cup C_3) = 600$ $SS(C_2 \cup C_3) - SS(C_2) - SS(C_3) = 200$

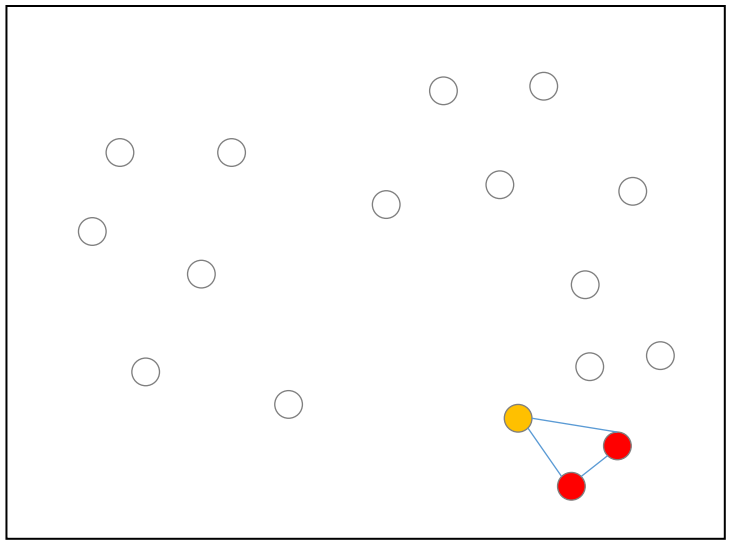
C_1과 C_2를 결합

Hierarchical clustering



Iter = 1
가장 가까운 두 점을 연결

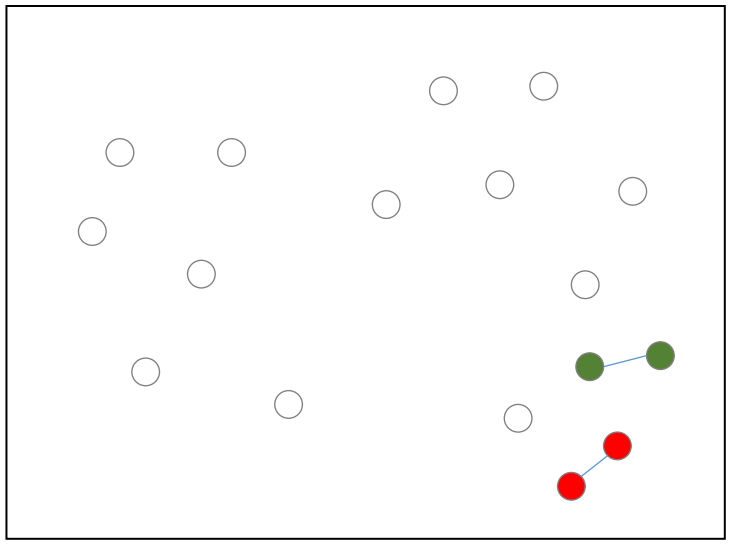
Hierarchical clustering



Iter = 1
가장 가까운 두 점을 연결

Iter = 2
$d(C_i, C_j)$를 $d(x_p, x_q)$의 평균으로 정의한다면
두 빨간색점들과의 거리 평균이 다른 점들보다
가까우므로 주황색 점이 연결
(completed linkage)

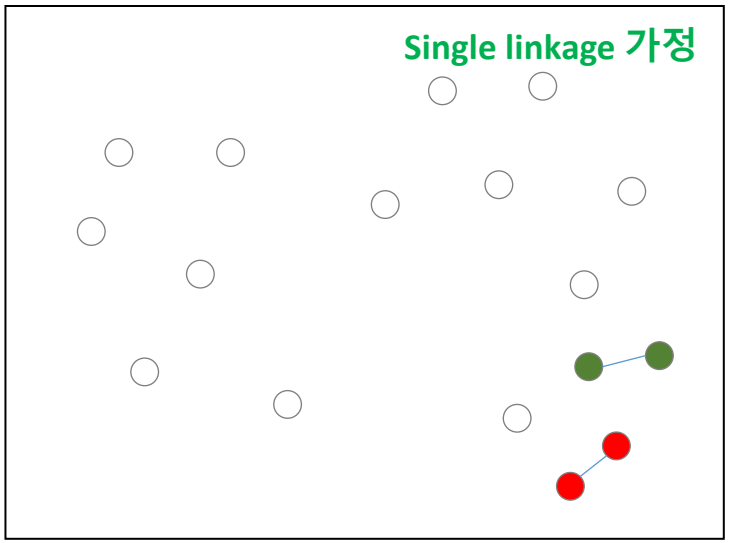
Hierarchical clustering

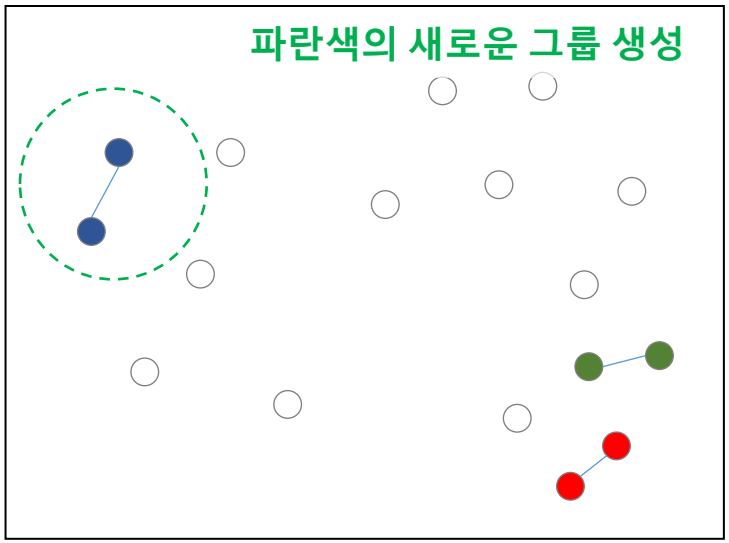


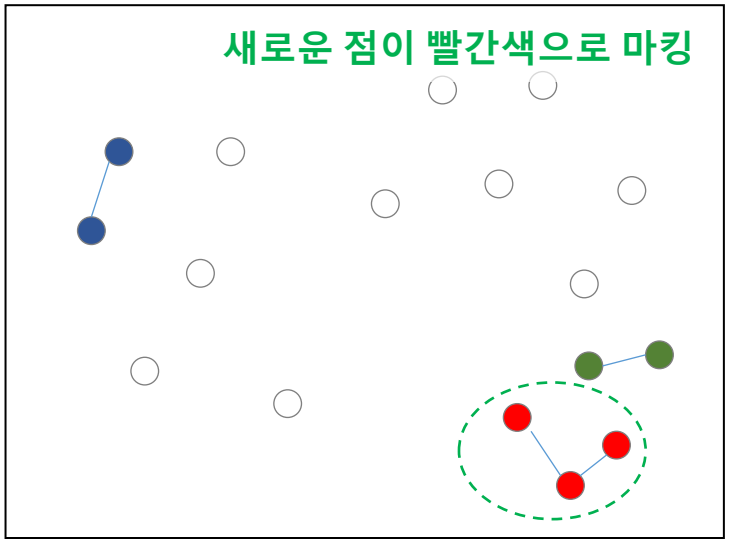
Iter = 1
가장 가까운 두 점을 연결

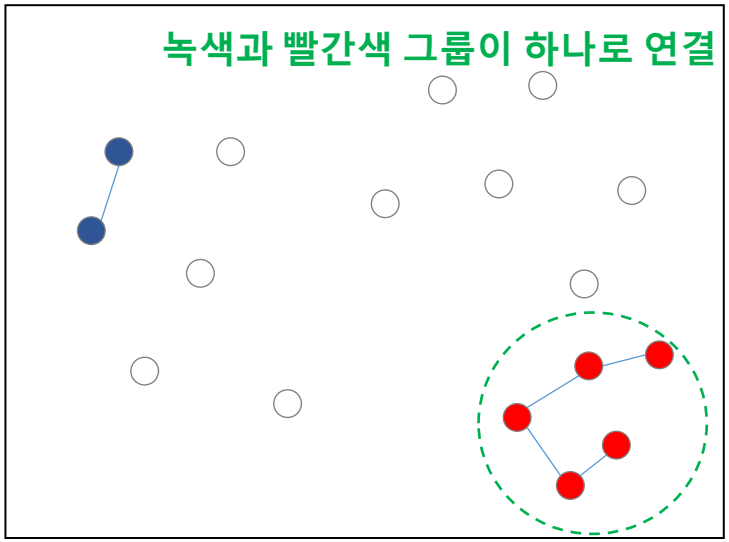
Iter = 2
$d(C_i, C_j)$를 $d(x_p, x_q)$의 min으로 정의한다면
녹색의 점이 하나로 연결
(single linkage)

Hierarchical clustering

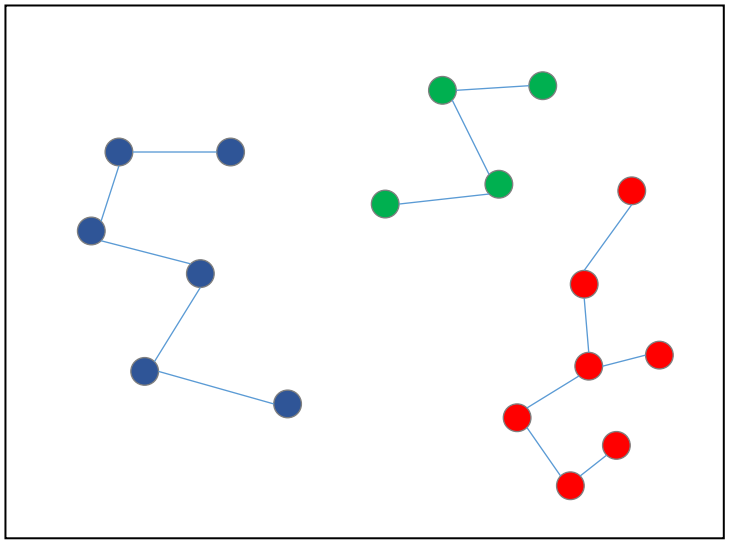


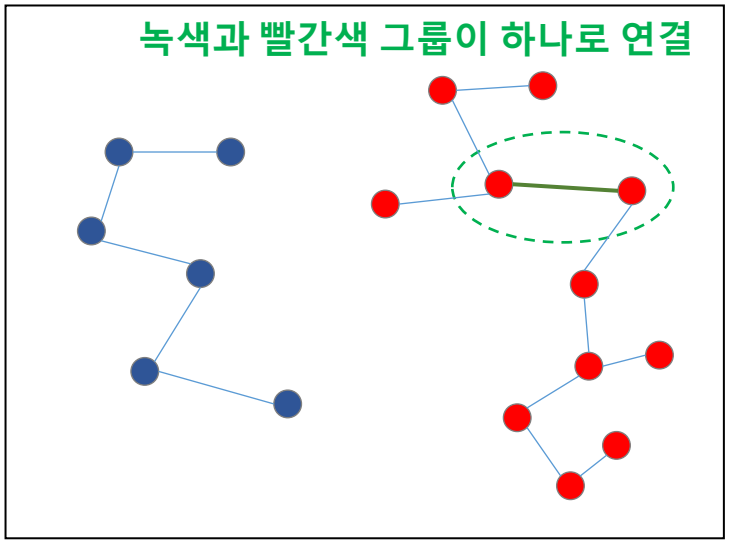


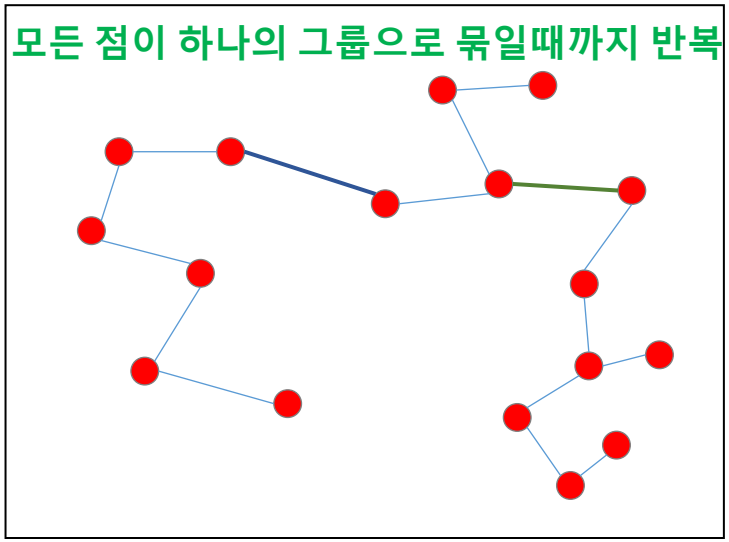


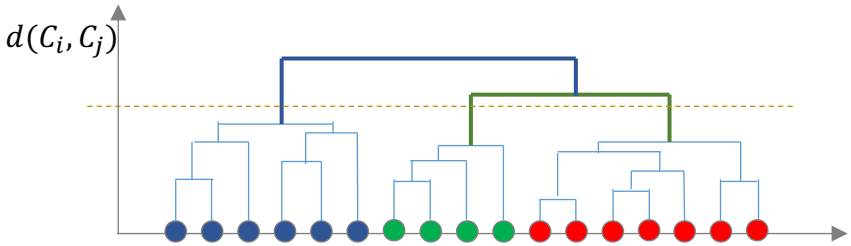


Hierarchical clustering





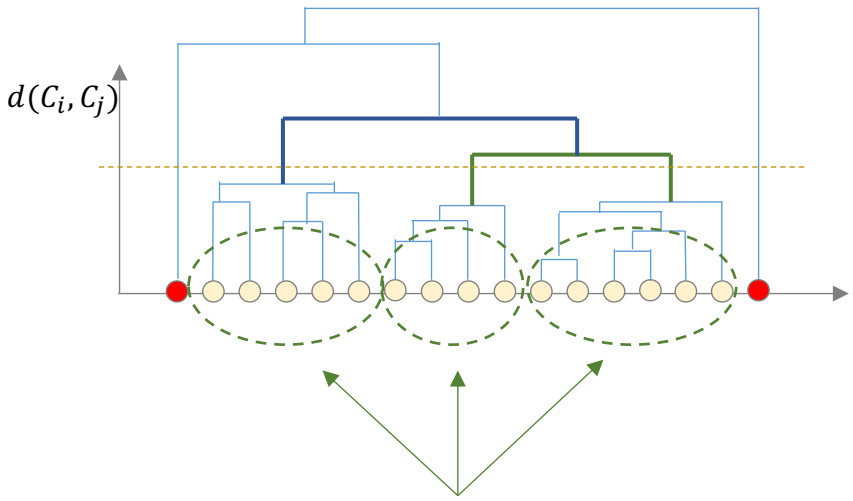




- Dendrogram은 링크가 생성되는 과정을 시각화한것
- 노란선의 distance로 cut한다는 것은 파란/녹색의 링크를 추가하지 않고 3개의 군집으로 묶겠다는 의미

Hierarchical clustering

- **Outliers를 알아서 걸러줄 수 있음**
  - Single linkage는 가장 가까운 점들을 하나씩 이어나가는 구조이기 때문에, 다른 점들이 큰 군집으로 묶여갈 때 까지 다른 점들과 잘 묶이지 않는 점이 outliers



다른 점들은 큰 3개의 그룹으로 묶이지만,
붉은색 점들은 마지막에 큰 군집으로 묶임

Hierarchical clustering

- **비싼 계산 비용**
  - 데이터의 개수가 N개라고 할 때, 모든 점들간의 거리를 계산해야 하기 때문에 $O(N^2)$ 계산 공간과 비용이 필요.
  - *k*-means는 i번의 반복을 할 때, O(i * k * N)의 비용이기 때문에 k < N일 때, *k*-means가 매우 빠르게 계산 됨

Hierarchical clustering

- **고차원 벡터에서 잘 작동하지 않음**

  - 고차원에서는 최인접이웃들의 거리 외에는 정보력이 없음
    (군집화 강의자료 마지막 부분 참조)

  - Completed linkage를 이용할 경우, 군집 안에 포함된 모든 점들간의 거리의 평균을 두 군집 간의 거리로 이용. 대부분 점들의 거리가 멀 경우, 군집간 거리가 잘 정의되지 않음

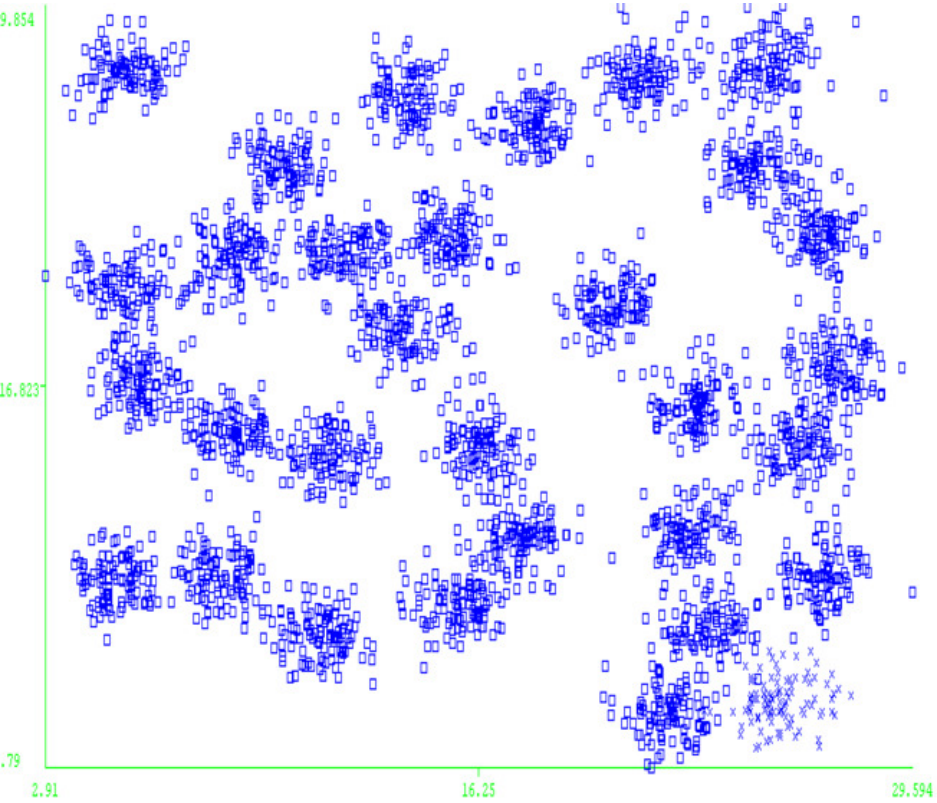
DBSCAN

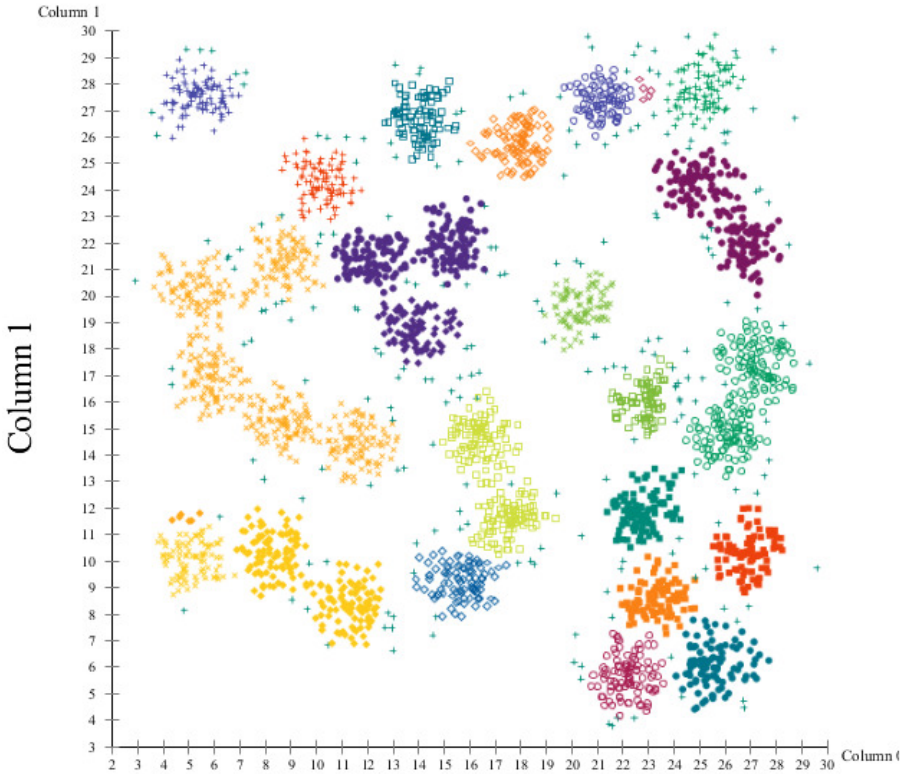
- **Density-Based Spatial Clustering of Applications with Noise**
  - 모든 점이 반드시 그룹에 속하지 않는다고 가정 (노이즈)

- **유사도**
  - n개의 데이터 X에 대하여 두 데이터 x_i, x_j간에 정의되는 임의의 거리 $d(x_i, x_j)$

- **그룹화의 방식**
  - Threshold 이상의 밀도를 지닌 점들을 모두 이어나가는 방식

DBSCAN





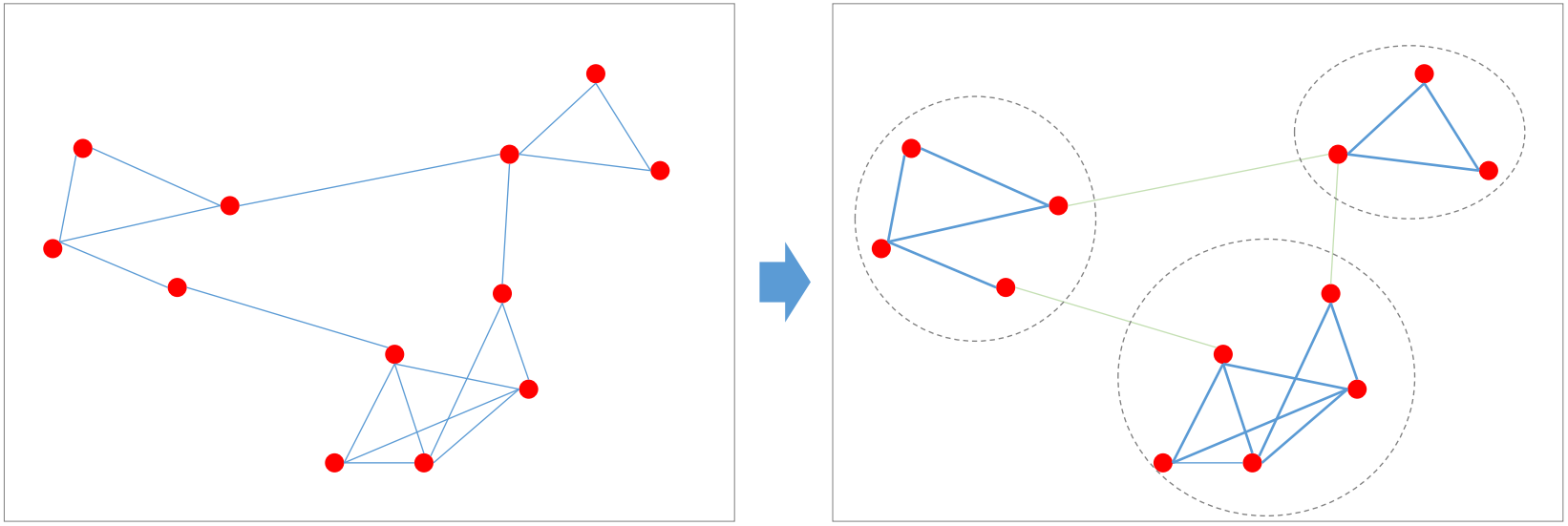
DBSCAN

- **Parameters에 따라 결과가 민감하게 작동**
  - 군집을 결정하는 밀도값 threshold에 의하여 데이터에서의 노이즈 비율이 예민하게 변함
- **높은 계산 비용**
  - DBSCAN은 모든 점들간의 거리를 한 번 이상 계산해야하기 때문에 $O(N^2)$의 계산 비용 필요

Community Detection

- ***k*-means, DBSCAN은 벡터로 표현된 데이터를 군집화하는 방법이지만, 커뮤니티 디텍션은 그래프로 표현된 데이터에서의 군집화 방법**

  - 소셜 네트워크 분석에서 사용자 군집화와 같은 문제에서 많이 이용됨
  - 대체로 계산이 오래걸리며, scikit-learn과 같은 머신러닝 라이브러리에 잘 들어있지 않음 (SNAP, at Stanford*)



http://snap.stanford.edu/snap/description.html

군집화

- ***k*-means는 centroids를 중심으로 구형의 군집을 만듦**
  - 유클리디언의 구형은 공간에 구를 만드는 형태
  - 코싸인의 구형은 각도를 파티셔닝하는 형태

- **Hierarchical clustering, DBSCAN은 임의의 모양의 군집추출을 위한 방법**
  - Sparse vector로 이뤄진 문서 공간은 복잡한 모양이 아님
  - 데이터가 복잡한 모양이 아니라는 가정을 할 수 있다면 단순한 알고리즘이 안정적

Clustering Overview: Issues

- How many clusters are optimal?

How many clusters?

Six Clusters

Two Clusters

Four Clusters

Clustering Overview: Issues

- **How to evaluate the clustering results?**
  - There is no rule of thumb.

External	• Rand Statistic • Jaccard Coefficient • Folks and Mallows index • (Normalized) Hurbert Γ statistic
Internal	• Cophenetic Correlation Coefficient • Sum of Squared error (SSE) • Cohesion and separation
Relative	• Dunn family of indices • Davies-Bouldin (DB) index • Semi-partial R-squared • SD validity index • Silhouette