Topics

- Introduction
- Big Data
- Data preprocessing and measures
- Clustering → 비지도 학습의 대표
- Classification 분류

Clustering (군집)

Contents

- Introduction
- Clustering methods
- Partitional clustering (분할 군집화)
- Hierarchical clustering (계층 군집화)

Introduction

- What is cluster analysis?
- Applications of cluster analysis
- Methods for clustering
  - Distance-based algorithms
  - Model-based algorithms: (eg) finite mixture model
- Types of distance-based clustering
  - Partitional Clustering
  - Hierarchical Clustering

What is cluster analysis?

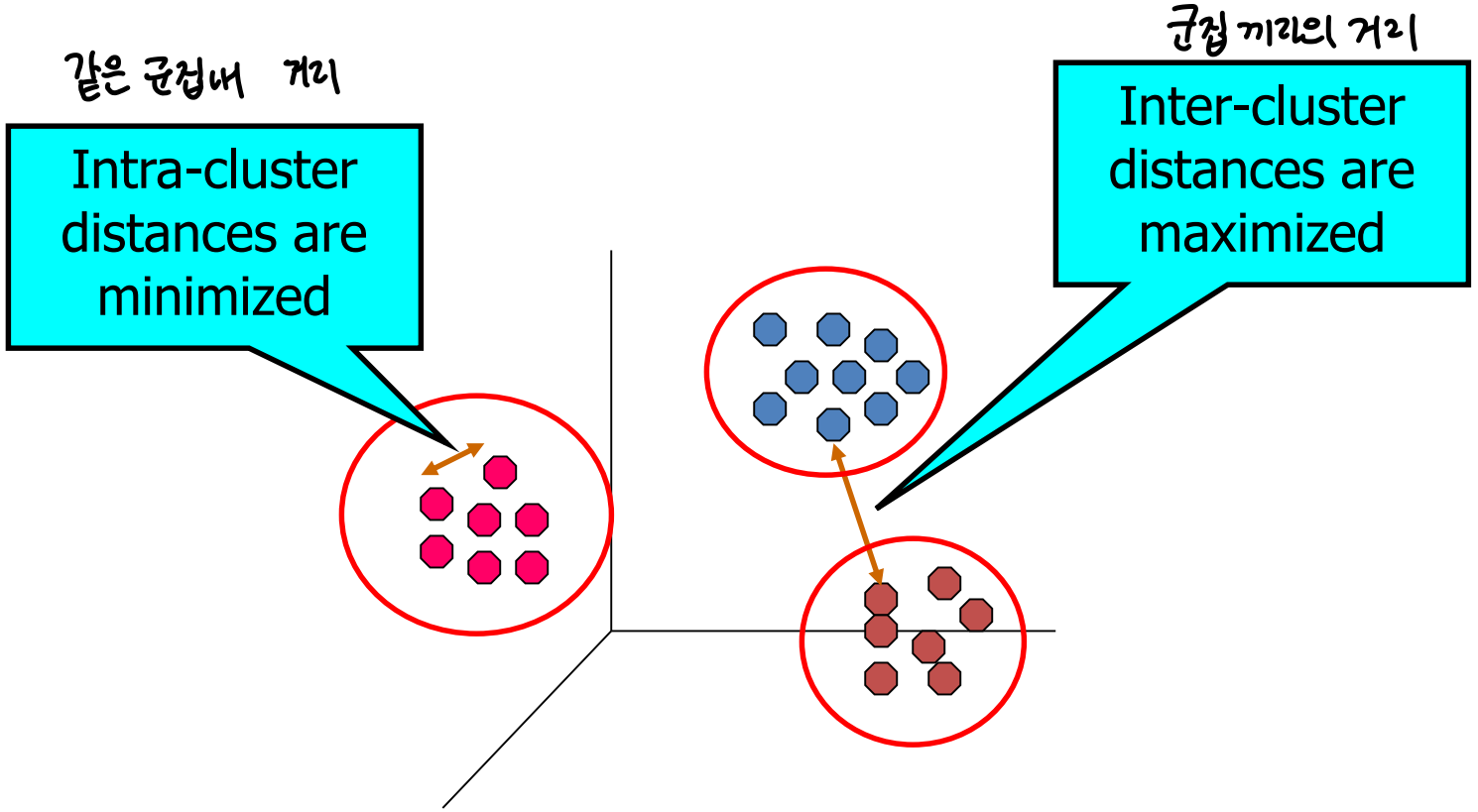
가장대표적 예시 : 학점내기

- 군집: 데이터들이 속한 그룹(혹은 cluster)을 찾는 것
- 각 cluster에 대한 사전 지식이 없는 상태에서 데이터를 분류 → unsupervised learning

정성적 기준 ex) 90점↑ 은 A 라는 기준이 없음!?

- 군집 기준: 같은 그룹에 속한 데이터들은 다른 그룹에 속한 데이터에 비하여 상대적으로 더 유사함
- 군집 방법
  - 통계적 모형
  - 데이터 간의 유사도/비유사도

What is cluster analysis?



Applications of cluster analysis

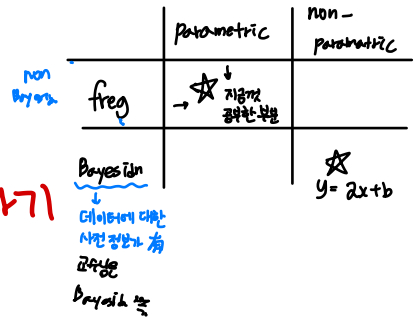
- Understanding
  - 상호 관련이 있는 웹 문서들을 그룹화 (혹은 cluster)
  - 유사한 기능을 가진 유전자 혹은 단백질들을 그룹화
  - 유사한 가격 변동을 가지는 주식들을 그룹화
  - 학점 산출

- Summarization
  - 대규모 데이터의 크기를 축소시킴
  - 데이터를 대표하는 값(cluster)

Difficulties of cluster analysis

- Hyper-parameters → 조금 어려움!!
  - Number of clusters
  - Definition of distance (or dissimilarity)

- Bayesian non-parametric method
  - Infinite mixture model
  - Infinite number of clusters
  - Let data choose the best number of clusters



Contents

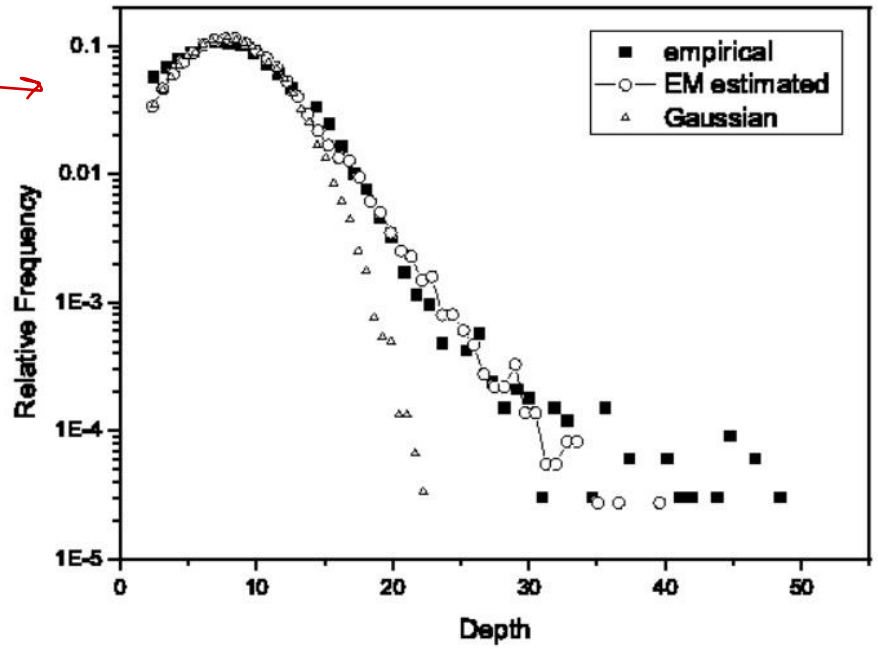
- Introduction
- Clustering methods
- Partitional clustering (분할 군집화)
- Hierarchical clustering (계층 군집화)

Methods for clustering

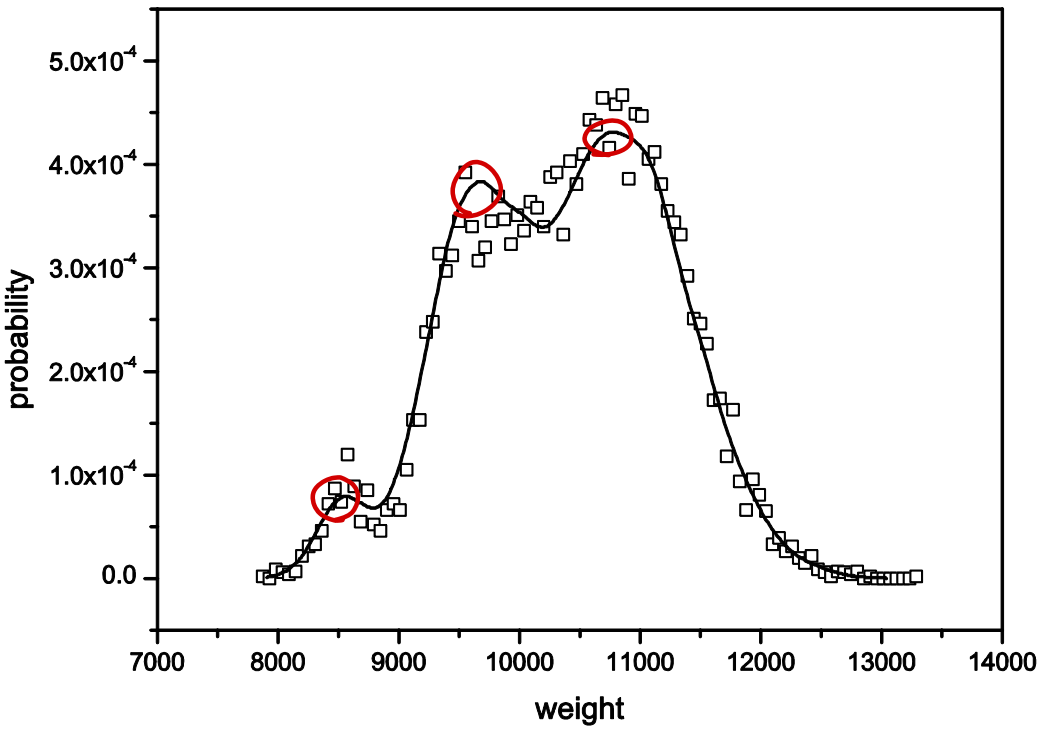
- Model-based algorithms
  - 통계적 모형 사용
  - Finite mixture model
  - Functional clustering analysis
- Distance-based algorithms
  - 데이터 간의 유사도/비유사도 사용
  - Partitional clustering
  - Hierarchical clustering
- Bayesian non-parametric method
  - Infinite mixture model

Model-based algorithms

- Finite mixture model
- (예) $p(X|\theta) = w_1 N(\mu_1, \sigma^2) + w_2 L(\mu_2, b)$
  - 두 확률 분포의 결합
  - 데이터를 사용하여 매개변수 $(w_1, \mu_1, \sigma^2, w_2, \mu_2, b)$ 추정 → EM 알고리즘



Finite mixture model



- 3개의 정규분포
- 분산, 가중치가 조금씩 다름.
- 앞서 나온 식으로 구한 파라메타로 만든 그래프임.

Mixture of three Gaussian distributions

* 이유를 생각해보는 습관을 길러보자

거리 기초 군집

Distance-based clustering

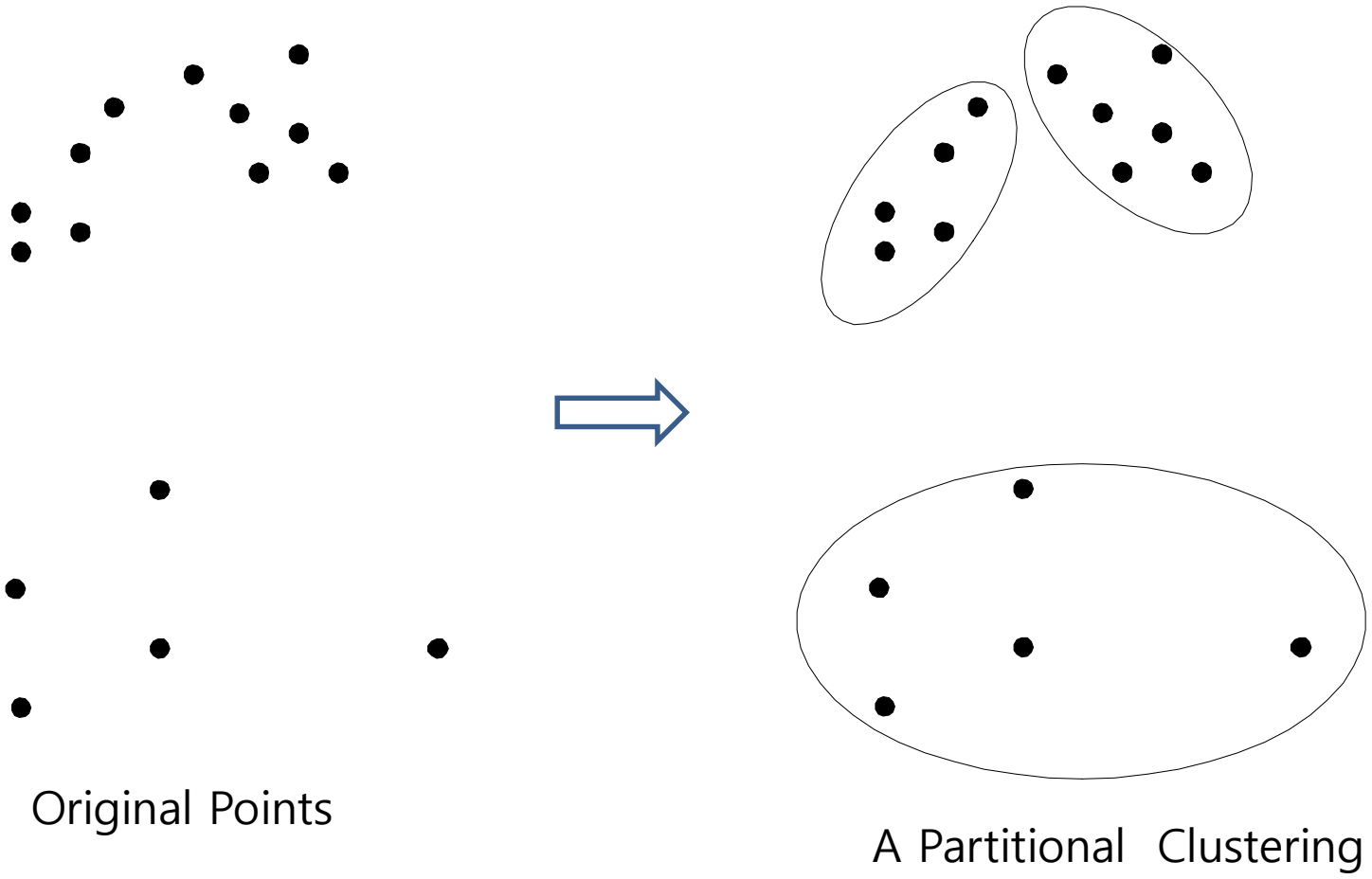
- 두 유형: partitional and hierarchical clustering
- Partitional clustering
  - 데이터가 1개 있으면 1개의 클러스터에만 속함
  - 시작 전 클러스터 개수 정해야 함. → 정한 개수만큼 클러스터가 나옴
  - 데이터가 중복되지 않도록 군집화 (평면적 클러스터링)
  - 동일한 데이터는 두 cluster 이상에 속하지 않음
  - 분류할 cluster의 개수를 미리 정함
  - (예) K-mean 알고리즘
- Hierarchical clustering
  - 클러스터 시작 시 개수 정하지 X
  - 그러나 끝나기 전 정해야 함
  - 클러스터 간에 계층이 존재
  - 클러스터 간의 거리도 필요. (데이터와 데이터 간의 거리 뿐만 아니라)
  - 클러스터를 hierarchical tree의 원소로 표현
  - 동일한 데이터가 두 개 이상의 cluster에 속할 수 있음
  - (예) Single-linkage clustering 알고리즘

위·아래가 있음

Contents

- Introduction
- Clustering methods
- Partitional clustering (분할 군집화)
- Hierarchical clustering (계층 군집화)

Partitional clustering



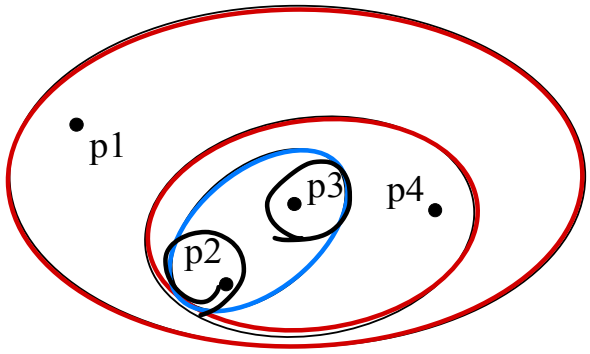
Original Points

A Partitional Clustering

모든 가능한 데이터의 개수가 다나옴 → 그 뒤 클러스터링

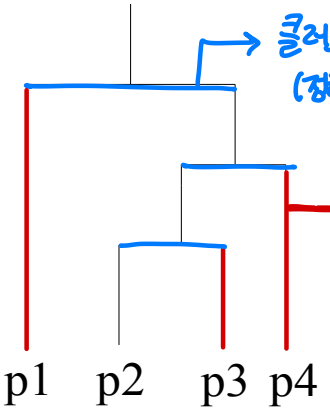
Hierarchical clustering

똑같은 데이터



Traditional Hierarchical Clustering

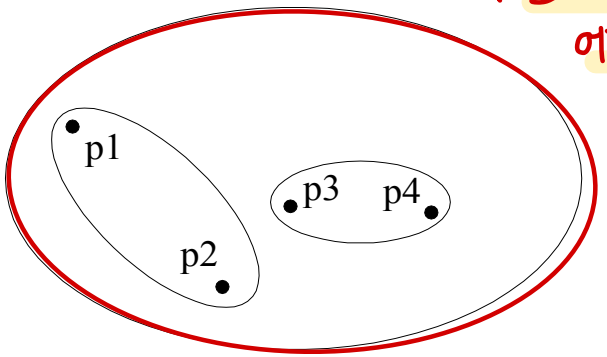
더 많이쌓임 →



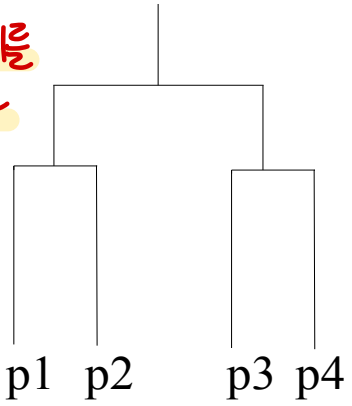
클러스터의 개수 (정할수 있음)

(데이터간의) 길이 = 거리

Traditional Dendrogram



★ 클러스터와 클러스터간의 거리를 어떻게 정의하냐에 따라 같은 데이터임에도 다른 결과가 나옴.



〈Hierarchical〉

〈Partitional〉

① 클러스터 개수 몇개?

② 거리가 얼마?

③ 클러스터 - 클러스터 간의 거리

통계적 방법 : p_1, p_2 거리 평균과 p_3, p_4 거리 평균 사이의 거리!

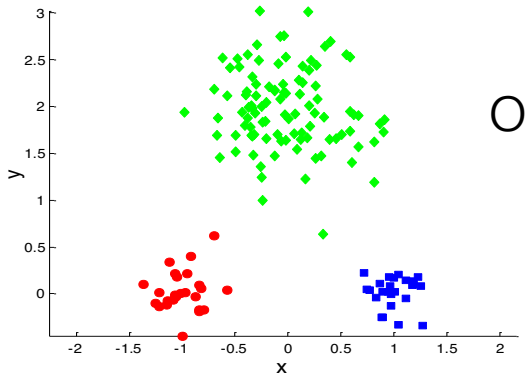
Clustering algorithms

- Partitional clustering
  - K-means
  - Bisecting k-means
- Hierarchical clustering
  - Single-linkage
  - Complete-linkage
  - Average-linkage
  - Ward's method
  - Method based on MST (minimum spanning tree)

K-means clustering

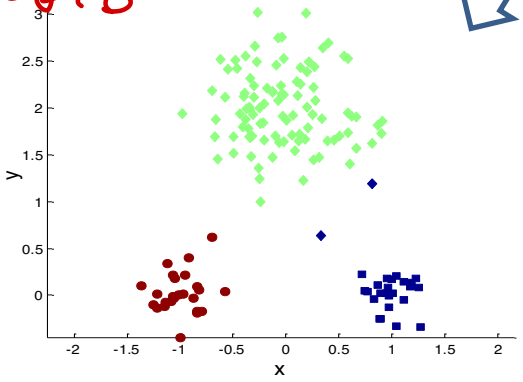
- Partitional clustering에 속한 방법
- n개 데이터를 k개의 클러스터로 분류
  - Cluster 개수 k는 미리 정함
  - Centroid: 각 cluster의 중심점
- 데이터를 가장 가까운 centroid가 속한 cluster에 할당
- 계산적으로 매우 어려움
  - NP-hard 문제
  - 휴리스틱스(heuristics)를 사용하여 지역 최적해를 구함

Two different K-means clustering



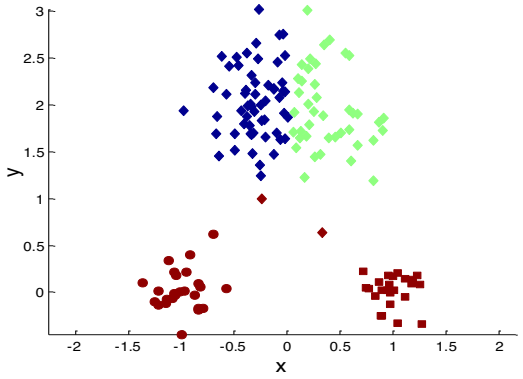
Original Points

데이터 개수가 얼마 안될땐 K-means로 정확한 optimal 군집구함



Optimal Clustering

데이터가 너무 많으면 근사함.



Sub-optimal Clustering

K-means clustering: centroid

(데이터가 n개 있음)

- Centroid 클러스터 안에 있는 데이터를 대표하는, 기준이 되는 Centroid를 사용
  - 클러스터에 속한 점들의 평균치 좌표
  - 초기 centroid는 임의로 선택
- Centroid와 데이터 사이의 근접도(closeness)
  - 유사도 척도 사용
  - 예: Euclidean distance, cosine similarity, correlation
- K-means 수렴
  - 위의 유사도 척도를 사용하는 경우에는 수렴
  - 대부분의 경우 빠른 수렴 → 클러스터 상태 유지

Evaluating K-means clusters

- Sum of squared error (SSE)
  - 오차제곱합, 가장 널리 사용되는 척도
  - 오차: 각 데이터에서 가장 가까운 cluster까지 거리
  - SSE는 오차를 제곱하여 모든 데이터들에 대하여 합

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

    x는 cluster C_i 에 속한 데이터

    m_i는 cluster C_i 를 대표하는 점으로 cluster의 중심점(평균)에 해당
- clusters 중에서 작은 오차를 가지는 cluster를 선택
  - SSE를 줄이는 가장 좋은 방법은 cluster의 개수를 크게 하는 것
  - 적은 K로 구한 좋은 clustering은 큰 K로 구한 나쁜 clustering보다 SSE가 적을 수 있음

표준 알고리즘

- K-means algorithm
  - Lloyd's algorithm
  - MacQueen algorithm
- m 차원의 n개 데이터 X를 k개 클러스터(C_1, C_2, … ,C_k)로 분류
  - 초기의 주어진 k개 평균 (m_1, m_2, … ,m_k)
  - 오차제곱합을 최소로 하는 클러스터 선택

$$argmin_C \sum_{i=1}^{k} \sum_{x \in C_i} ||x - m_i||^2$$

- 알고리즘: 아래 두 단계를 반복
  - 각 데이터를 클러스터에 할당
  - 평균값을 갱신하여 새로운 centroid 결정

표준 알고리즘

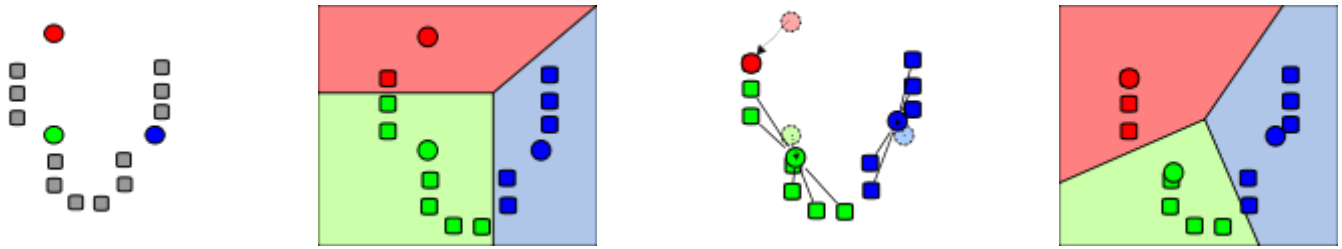
- 두 단계를 반복
  - 각 데이터를 클러스터에 할당

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\| \leq \|x_p - m_j^{(t)}\| \; \forall \; 1 \leq j \leq k\}$$

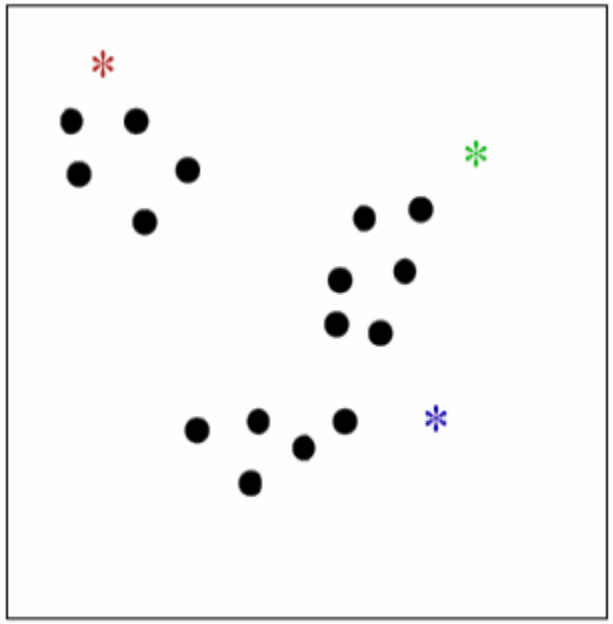
  - 평균값을 갱신하여 새로운 centroid 결정

$$\mathbf{m}_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{\mathbf{x}_j \in S_i^{(t)}} \mathbf{x}_j$$

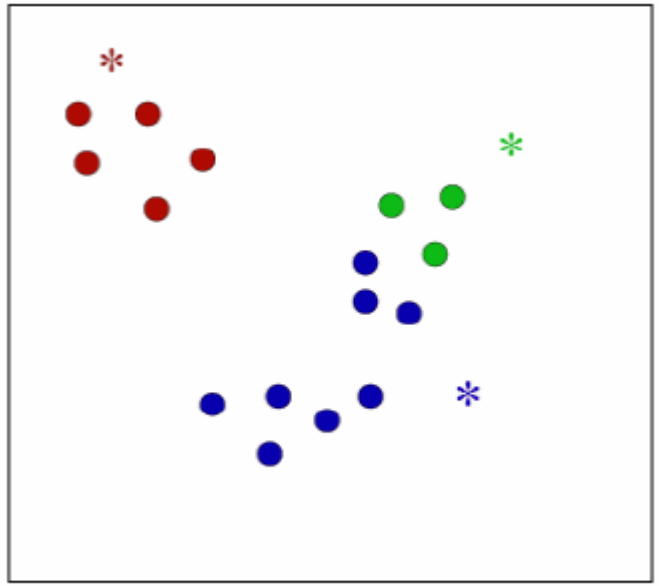
그림을 통한 이해



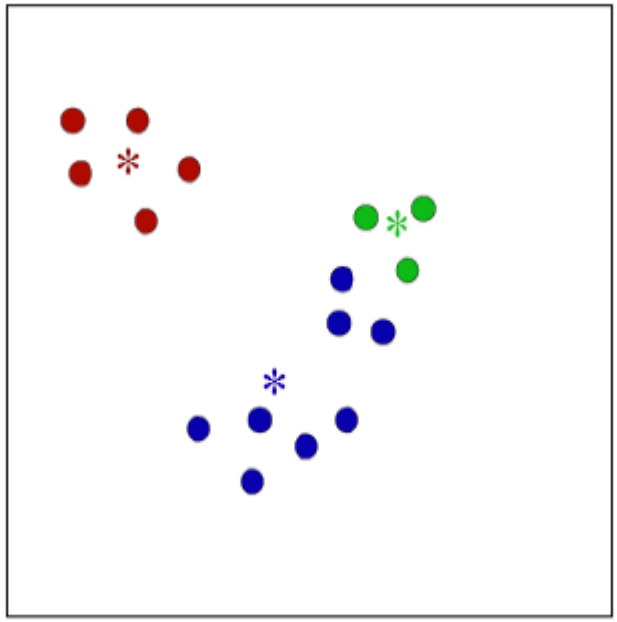
(1) 클러스터 개수(K=3)를 임의로 선정 (color)
(2) 모든 점을 k개 클러스터에 할당 → Voronoi diagram
(3) centroid 갱신: 각 클러스터에 속한 점들의 평균값
(4) 2-3단계를 수렴할 때까지 반복



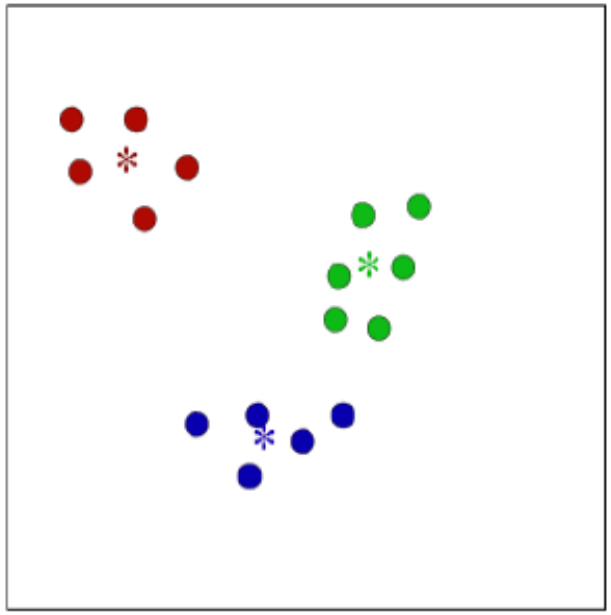
Initialize representatives (“means”)



Assign to nearest representative

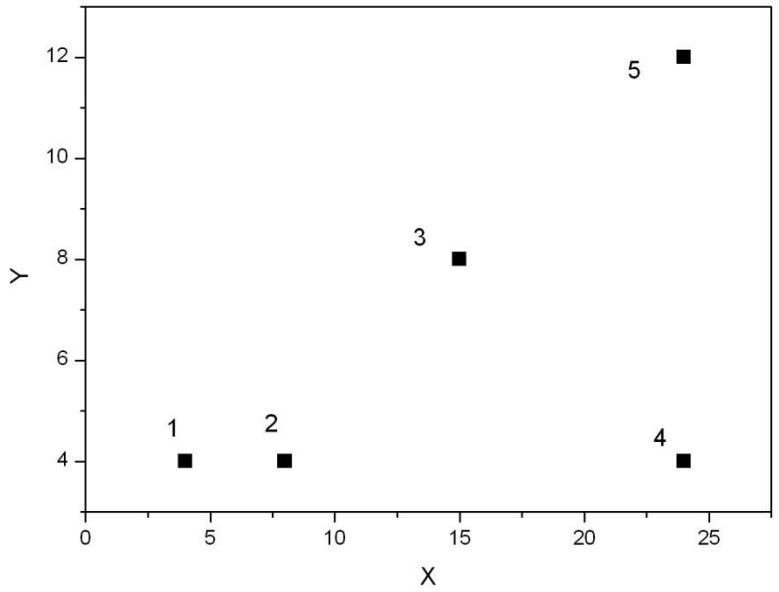


Re-estimate means



K-means algorithm 예제

- 2차원의 5개 데이터: (4, 4) (8, 4) (15, 8) (24, 4) (24, 12)
- 데이터간의 거리
  - Euclidean distance 사용
  - 다른 비유사도 척도 사용 가능 → 클러스터링 결과가 달라질 수 있음
- 클러스터 개수: 2개 (가정)
- 초기 시작점: 2개
  - (4, 4) (8, 4) 로 선정
  - 다른 점들을 선정해도 무방함

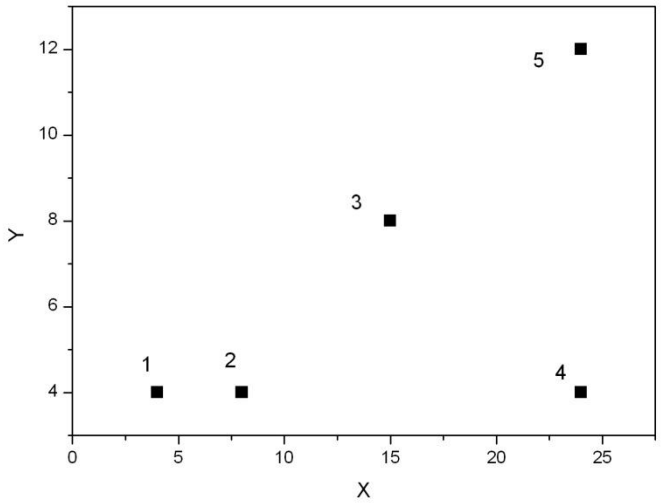


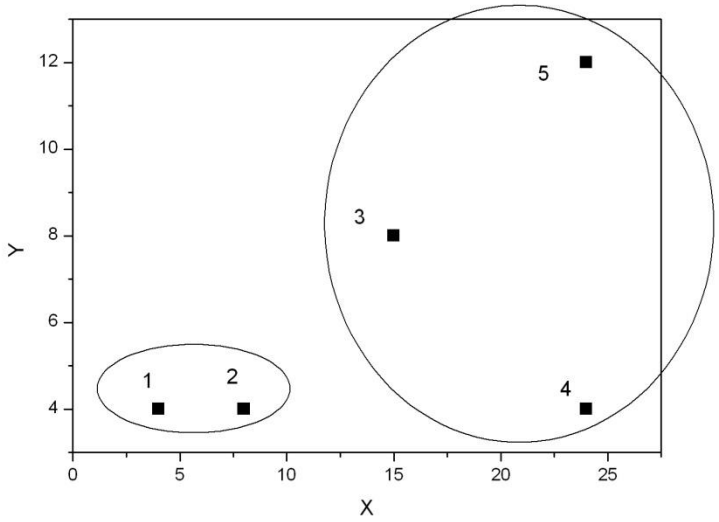
K-means algorithm 예제

- K-means algorithm 실행
  1. 임의로 정한 클러스터 수와 동일한 초기 시작점을 임의로 정하고 그 점을 각 클러스터의 centroid로 정한다. → centroid (4, 4) (8, 4)
  2. 각 object에 대하여 가장 가까이에 위치한 centroid를 찾아서 해당 클러스터에 그 object를 배정한다. →

     {(4, 4)} {(8, 4) (15, 8) (24, 4) (24, 12)}
  3. 변화된 클러스터에 대한 centroid를 update하여 2단계로 돌아간다. → update된 centroid : (4, 4) (17.75, 7)
  4. 변화된 클러스터 → {(4, 4) (8, 4)} {(15, 8) (24, 4) (24, 12)}
  5. update된 centroid : (6, 4) (21, 8)
  6. 변화된 클러스터 → {(4, 4) (8, 4)} {(15, 8) (24, 4) (24, 12)}
  7. 변화가 없음으로 알고리즘 종료

결과





Practice with R

- Data preparation

```
x <- iris[ , -5]
```

- K-means clustering

```
k <- kmeans(x, 3)  # '3' is number of clusters
k
names(k)
```

- Plotting results

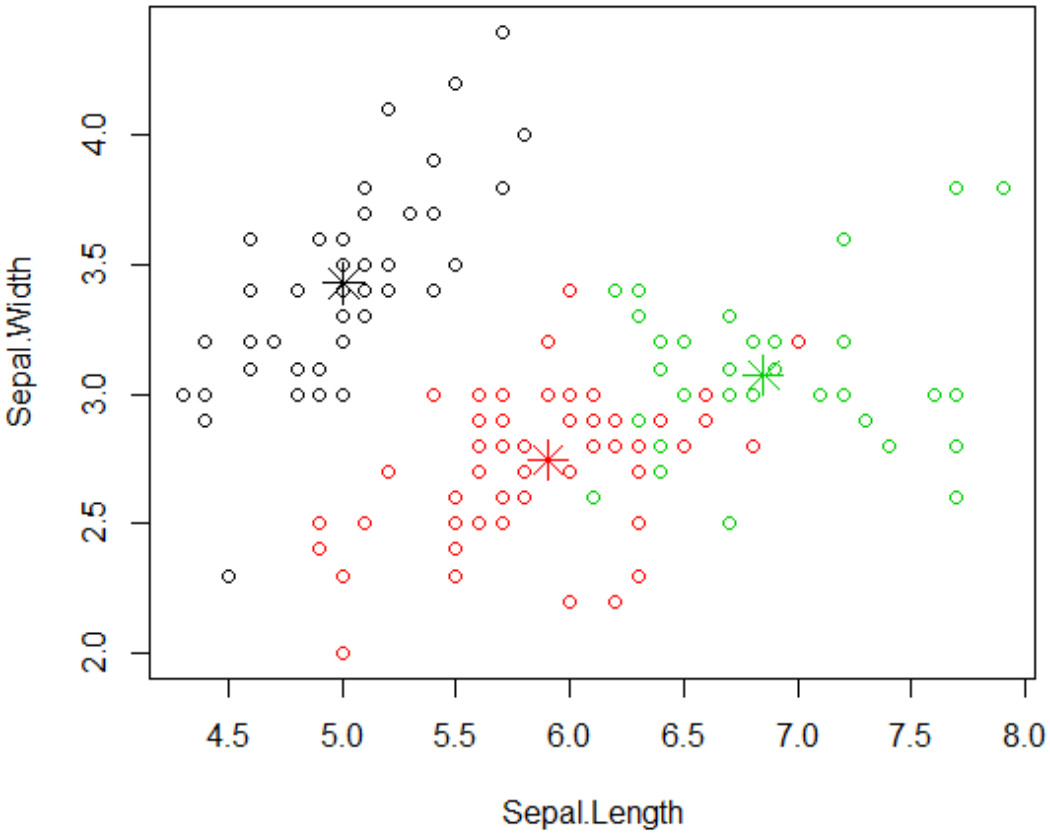
```
plot(x[1:2], col=k$cluster)
points(k$centers[ ,1:2], col=1:3, pch=8, cex=2)
```

(note) col: color; pch: plotting 'character';

cex: character (or symbol) expansion

- (note) the number of clusters? → package "NbClust"

Plot result



Bisecting K-means

- Bisecting K-means algorithm
  - K-means의 변형
  - Partitional 혹은 hierarchical clustering을 이룰 수도 있음
- Algorithm

Algorithm 3 Bisecting K-means Algorithm.

1: Initialize the list of clusters to contain the cluster containing all points.
2: **repeat**
3: Select a cluster from the list of clusters
4: **for** $i = 1$ to *number_of_iterations* **do**
5: Bisect the selected cluster using basic K-means
6: **end for**
7: Add the two clusters from the bisection with the lowest SSE to the list of clusters.
8: **until** Until the list of clusters contains K clusters

Bisecting K-means

1. 모든 데이터를 2개 군집(cluster)으로 분류한 후, 2개 군집을 리스트에 포함시킴
2. 군집 리스트에서 1개 군집을 임의로 선택하고, 선택된 군집을 리스트에서 제외시킴
3. 선택된 군집을 k-means를 사용하여 2개의 군집으로 분류
4. 분류된 2개의 군집을 리스트에 포함시킴
5. 2-4단계를 리스트에 속한 군집이 k개가 될 때가지 반복

끝