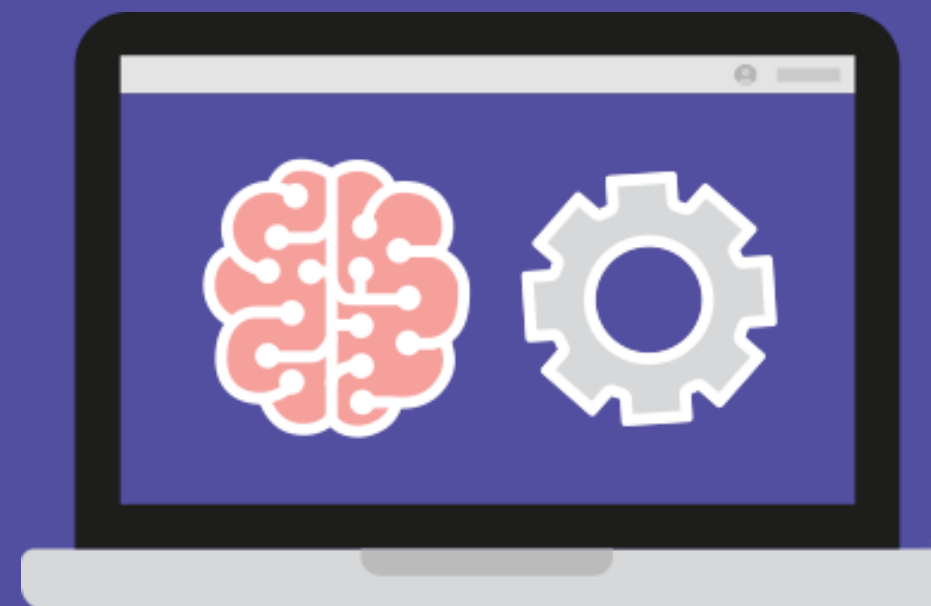


/* elice */

양재 AI School 인공지능 캠프

Lecture 10

Clustering K-means, KNN, GMM, HC



김도경 선생님

수업 목표

1 ○

클러스터링(Clustering)

1. 클러스터링(Clustering) 개요
2. 클러스터링의 타당성 평가

2 ○

여러 클러스터링 알고리즘

1. K-means
2. KNN
3. 가우시안 혼합모델(GMM)과 EM 알고리즘
4. 계층적(Hierarchical) 클러스터링(HC)
5. 알고리즘 비교

1. Clustering(클러스터링)

1-1. 클러스터링(Clustering) 개요

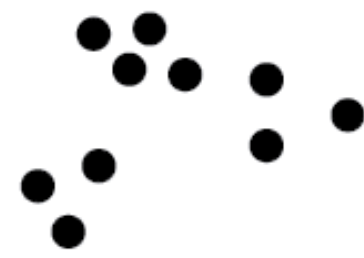
- **클러스터링 vs 분류**
- 클러스터링: 정답이 없는 비지도학습(unsupervised)
각 개체의 그룹 정보 없이 비슷한 개체끼리 묶어보는 것
- 분류: 정답이 있는 지도학습(supervised)
- **클러스터링의 목표**
 - (1) 군집 간 분산(inter-cluster variance) 최대화
 - (2) 군집 내 분산(inner-cluster variance) 최소화

1-2. 클러스터링의 타당성 평가

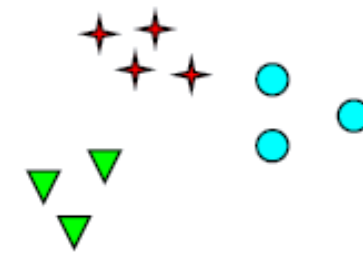
- **군집 타당성 평가**

비지도학습의 경우 정답이 없기 때문에 일반적인 머신러닝 알고리즘처럼 단순정확도(Accuracy) 등 지표로 평가할 수 없음

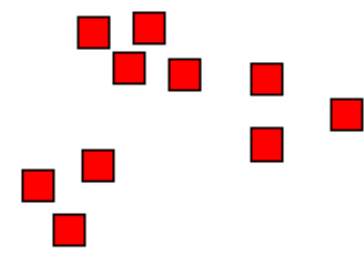
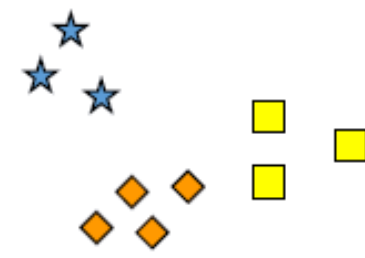
- 그림과 같이 최적의 군집 개수를 정답 없이 알아내기 쉽지 않음



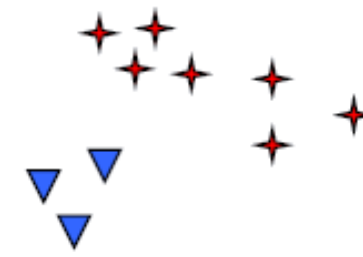
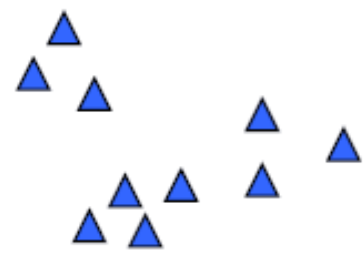
How many clusters?



Six Clusters



Two Clusters



Four Clusters

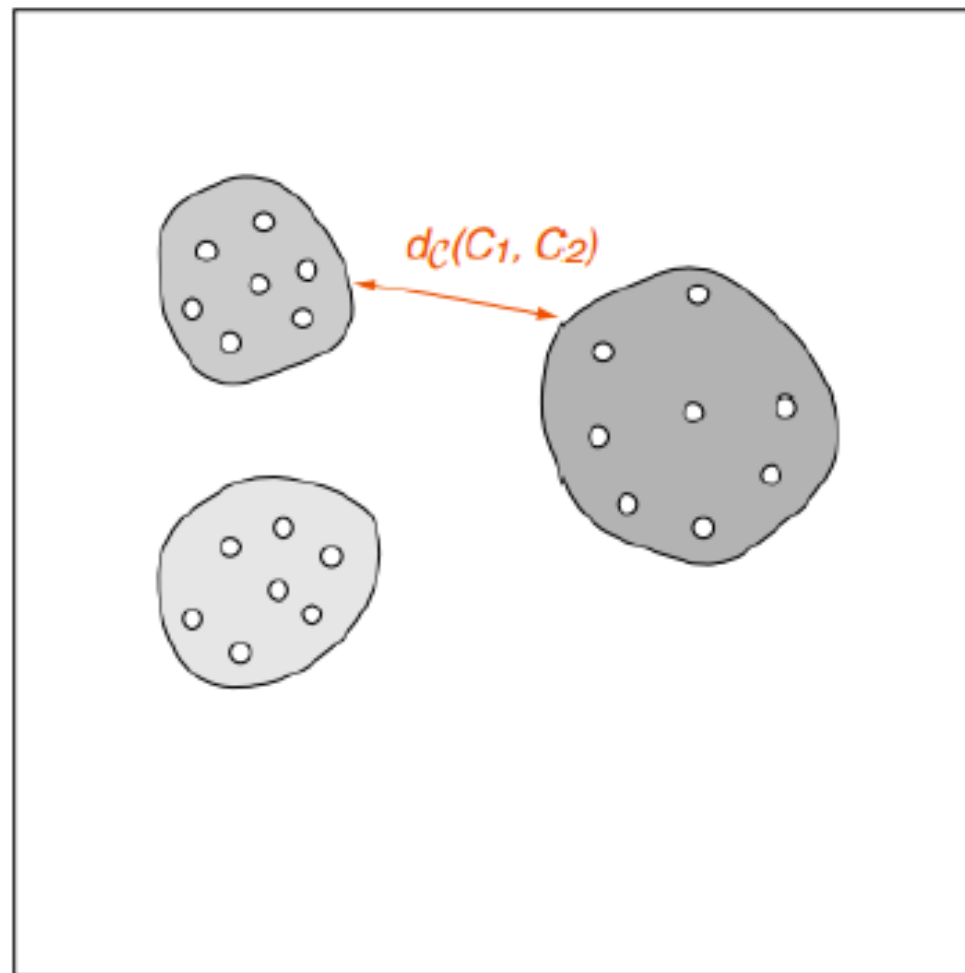


1-2. 클러스터링의 타당성 평가

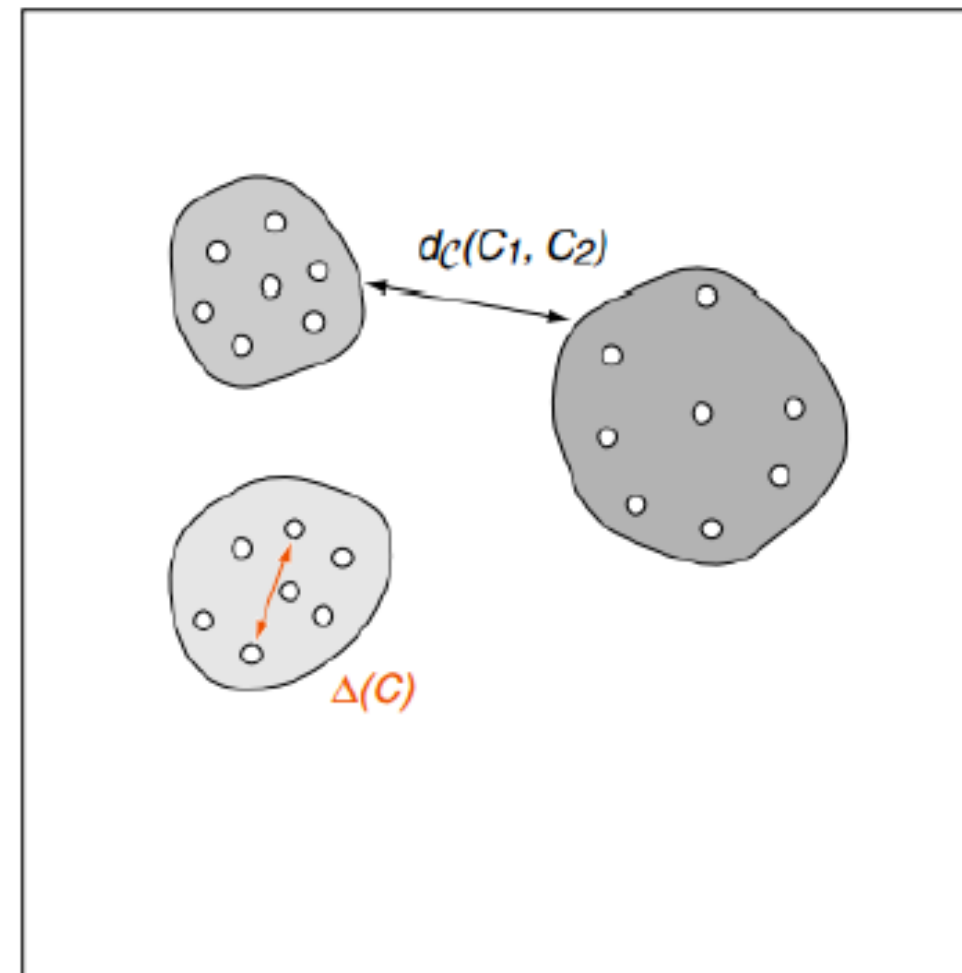
- 군집 타당성 지표

(1) 군집 간 거리 (2) 군집의 지름 (3) 군집의 분산

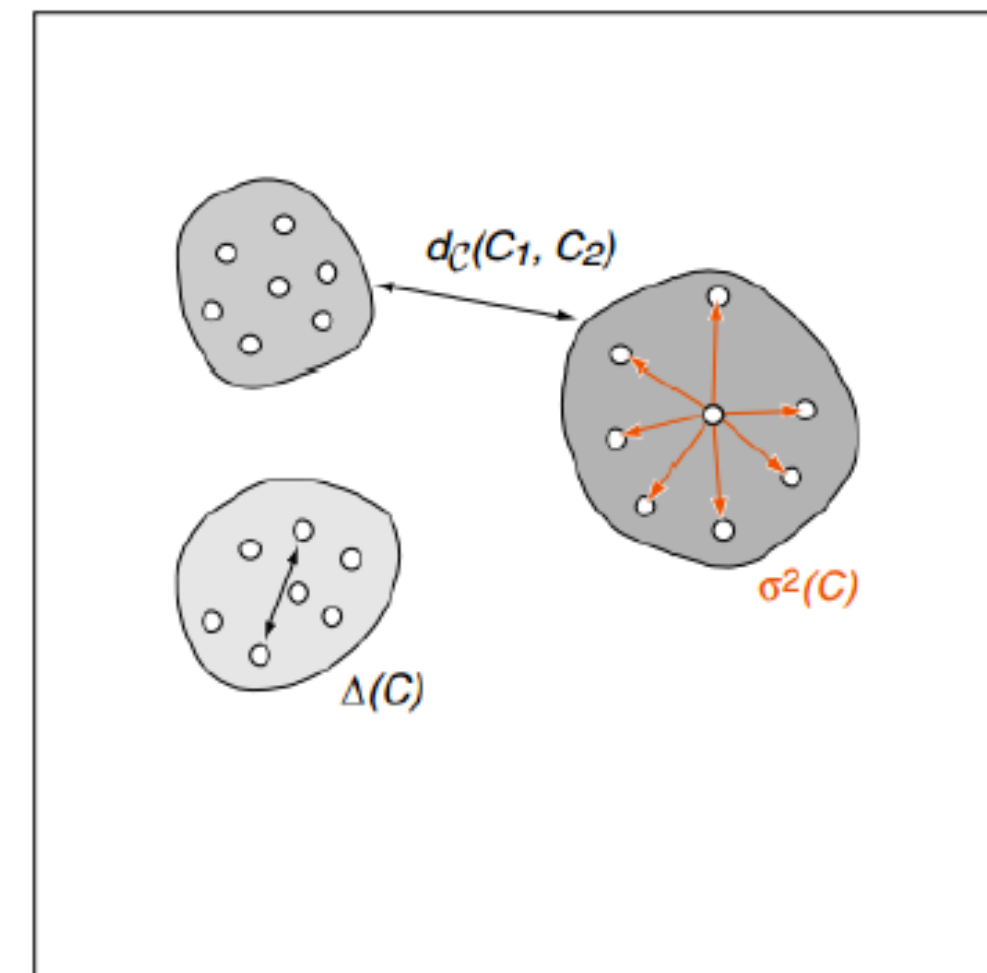
Distance between two clusters



Diameter of a cluster



Scatter within a cluster (SSE)

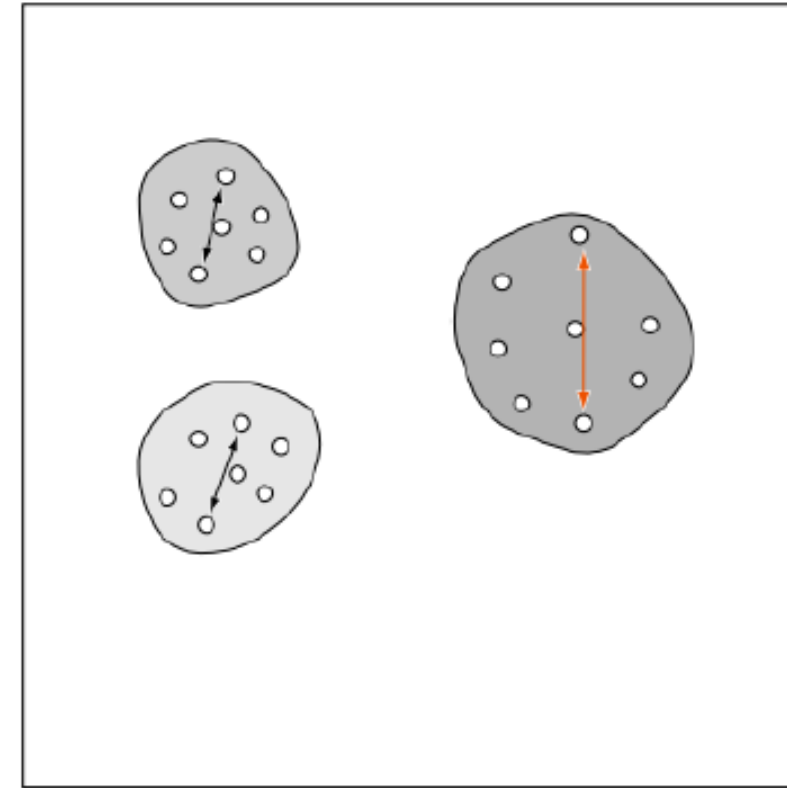
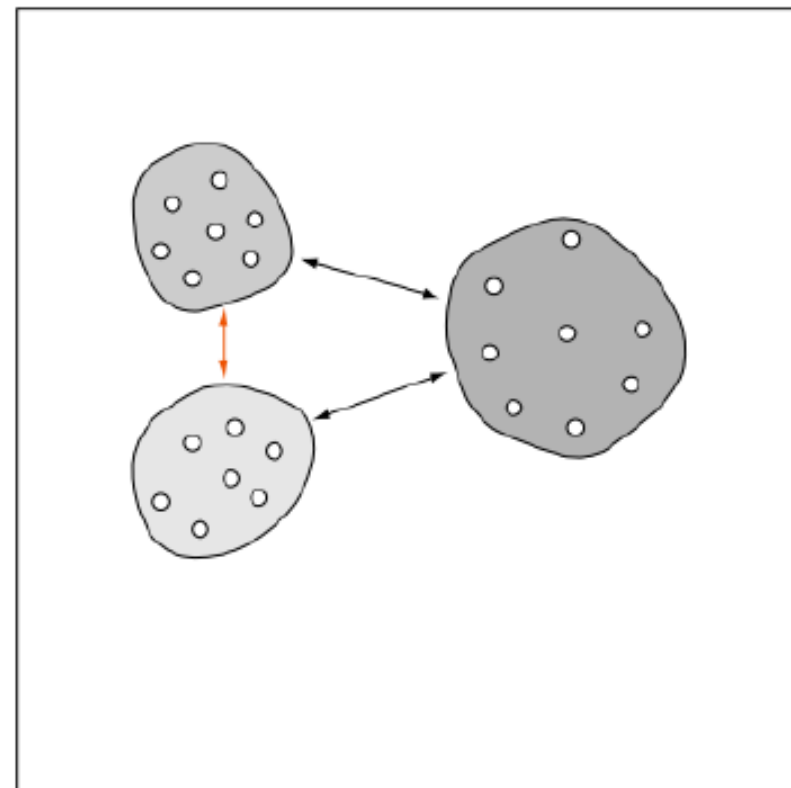


1-2. 클러스터링의 타당성 평가

- Dunn Index

군집 간 거리의 최소값(하단 좌측)을 분자, 군집 내 요소 간 거리의 최대값(하단 우측)을 분모로 하는 지표

$$I(C) = \frac{\min_{i \neq j} \{d_c(C_i, C_j)\}}{\max_{1 \leq l \leq k} \{\Delta(C_l)\}}$$



2. 여러 클러스터링 알고리즘

2-1. K-means

- K-means는 **EM 알고리즘**을 기반으로 함
 1. EM알고리즘은 **Expectation** 스텝과 **Maximization** 스텝으로 나뉨
 2. 이를 수렴할 때까지 반복
 3. 동시에 해를 찾기 어려운 문제를 풀 때 많이 사용되는 방법론
- E 스텝: 각 군집 중심의 위치를 구함
- M 스텝: 각 점이 어떤 클러스터에 속해야 하는지 멤버십 갱신

2-1. K-means

- **입력**

1. K (클러스터의 수)
2. 트레이닝 셋 $x^{(1)}, x^{(2)}, \dots, x^{(m)} \in \mathbb{R}^n$

- **알고리즘**

K개의 무게중심 $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$ 을 랜덤하게 초기화

Repeat

 i는 1부터 m까지

$c^{(i)} := x^{(i)}$ 에서 가장 가까운 클러스터 인덱스

 k는 1부터 K까지

$\mu_k :=$ 클러스터 k에 속해있는 점들의 평균

2-1. K-means

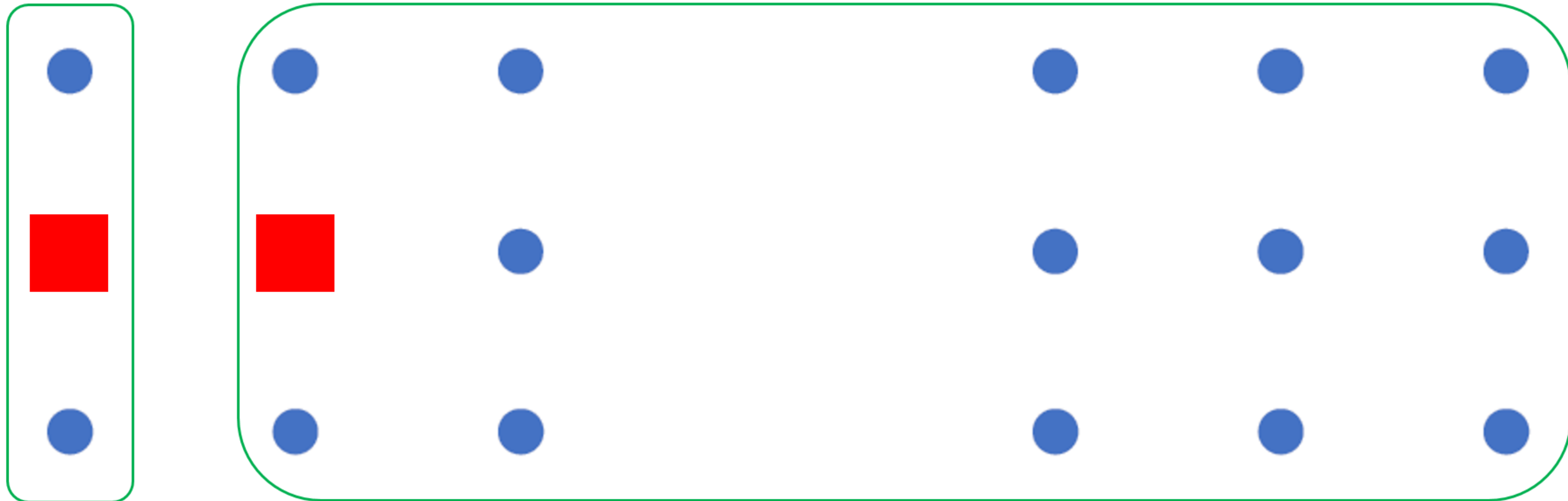
- 클러스터의 수 K 를 2로 정함
- 군집의 무게중심(빨간색 점)을 랜덤 초기화
(이 경우에는 점들 중에서)



2-1. K-means

- E스텝

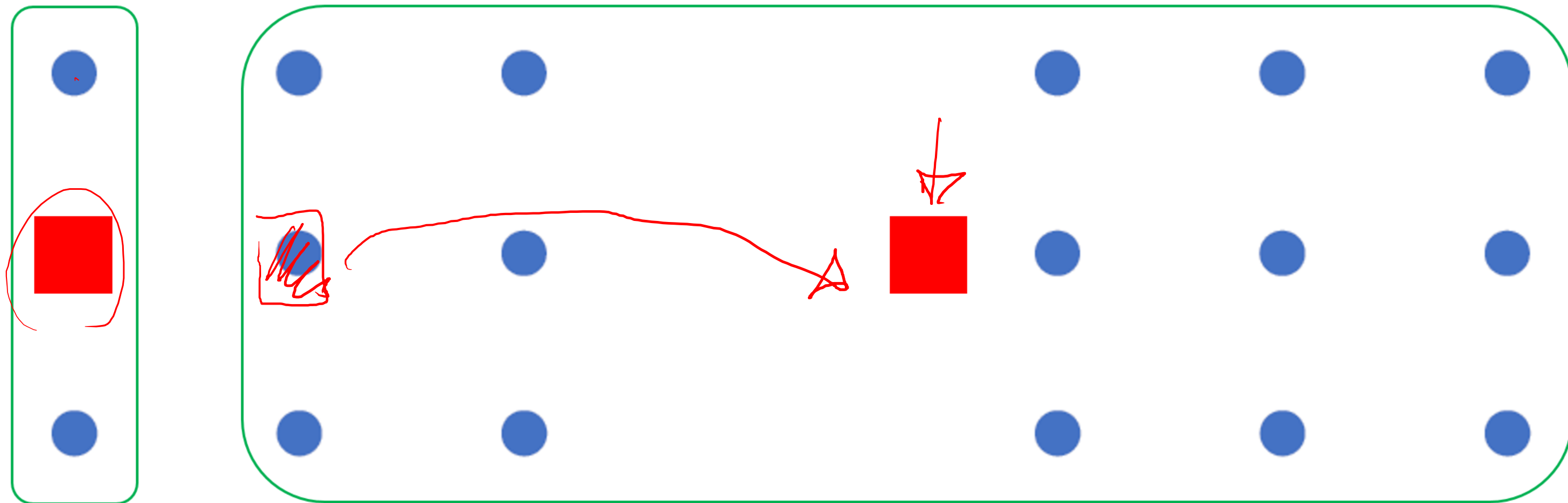
모든 점들(파란색 점)을 가장 가까운 무게중심을 기준으로 클러스터링



2-1. K-means

- M스텝

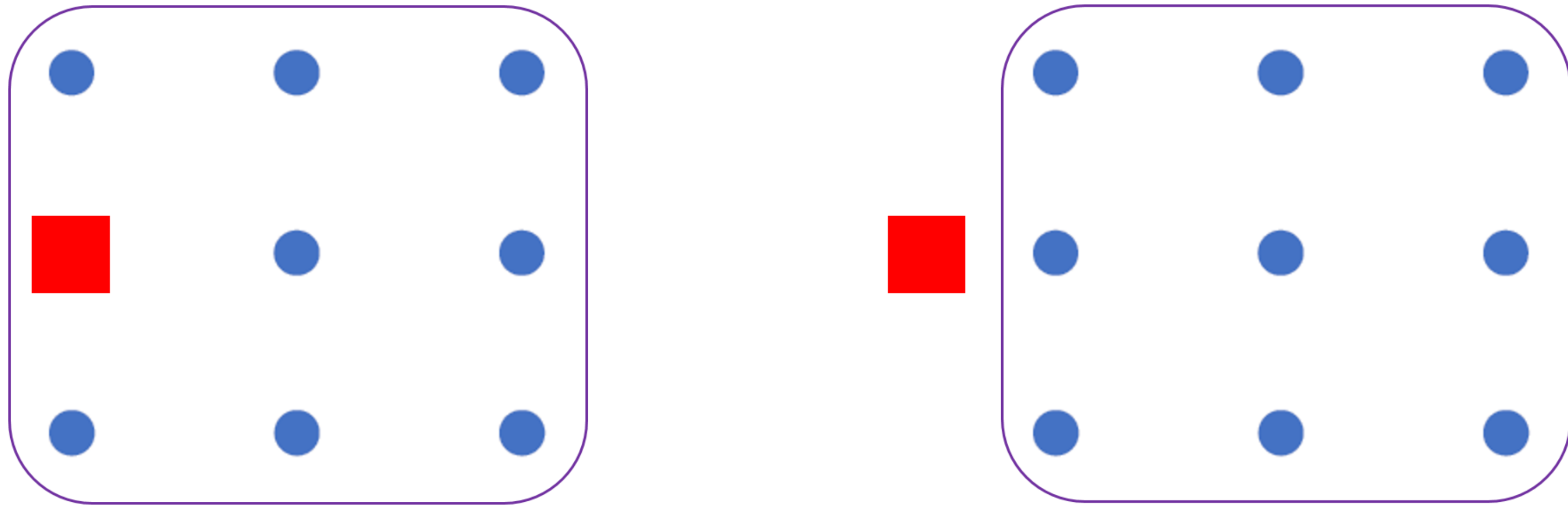
클러스터의 무게중심을 업데이트



2-1. K-means

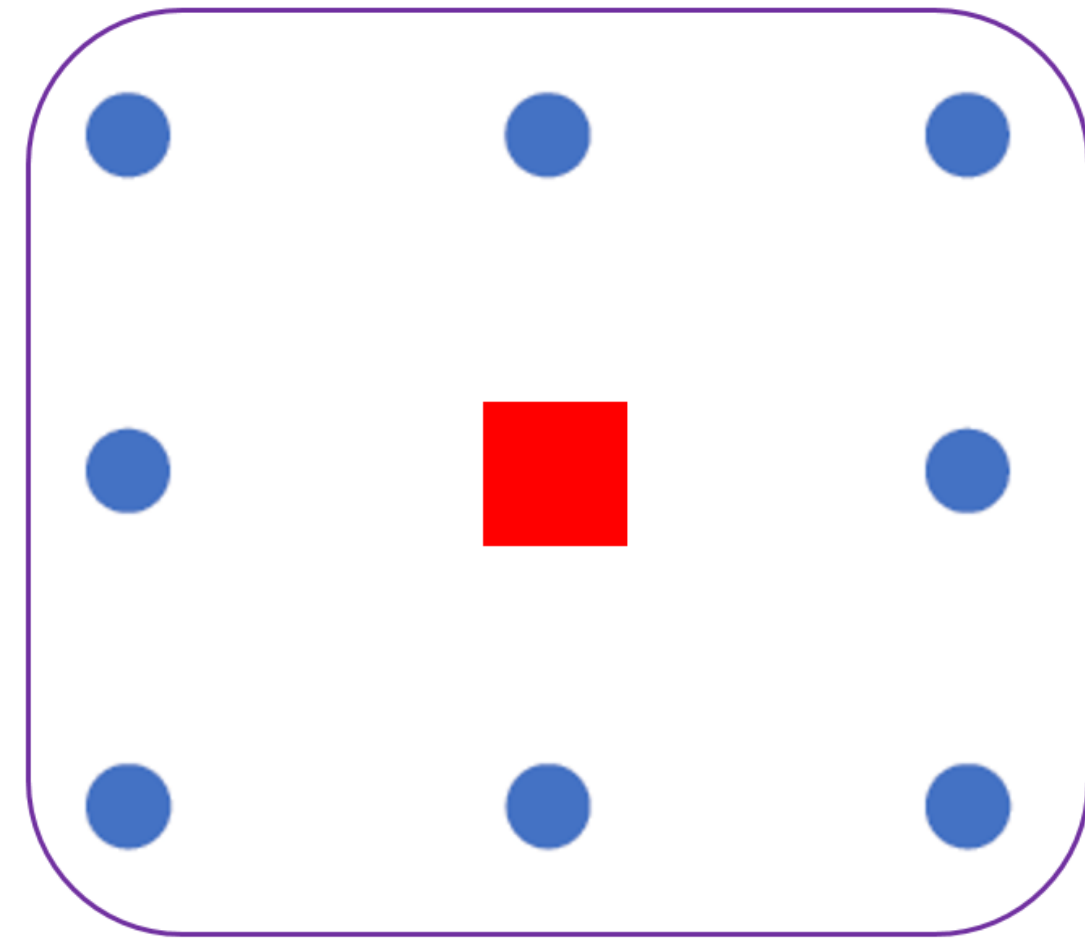
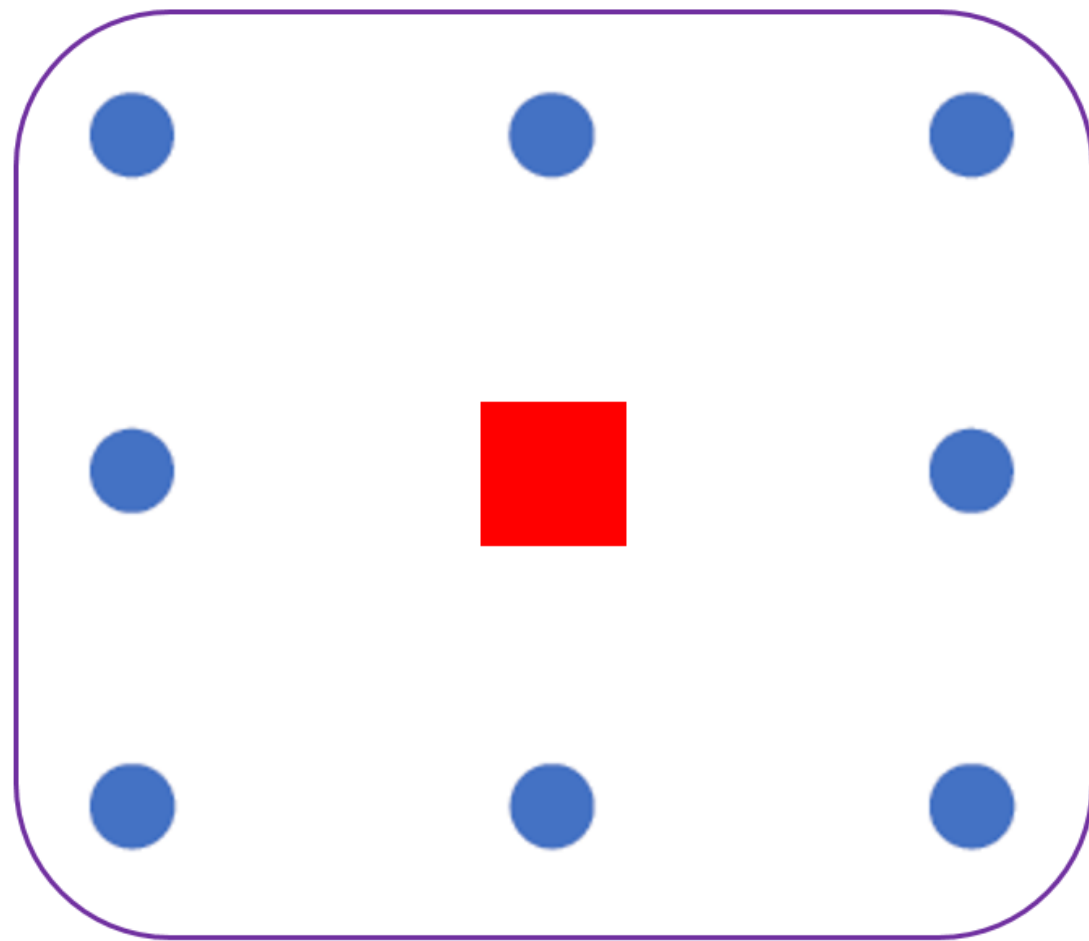
- E스텝

모든 점들(파란색 점)을 가장 가까운 무게중심을 기준으로 클러스터링



2-1. K-means

- M스텝
클러스터의 무게중심을 업데이트
- E, M스텝을 반복 적용해도 결과가 바뀌지 않거나(=해가 수렴), 사용자가 정한 반복수를 채우게 되면 학습이 끝남



2-1. K-means

- 손실함수

K-means 알고리즘은 결국 아래의 손실함수를

$$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K) = \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - \mu_{c^{(i)}}\|^2$$

E스텝에서는

$c^{(1)}, c^{(2)}, \dots, c^{(i)}$ 에 대해서 최소화하고,

M스텝에서는

$\mu_1, \mu_2, \dots, \mu_K$ 에 대해서 최소화하는 것

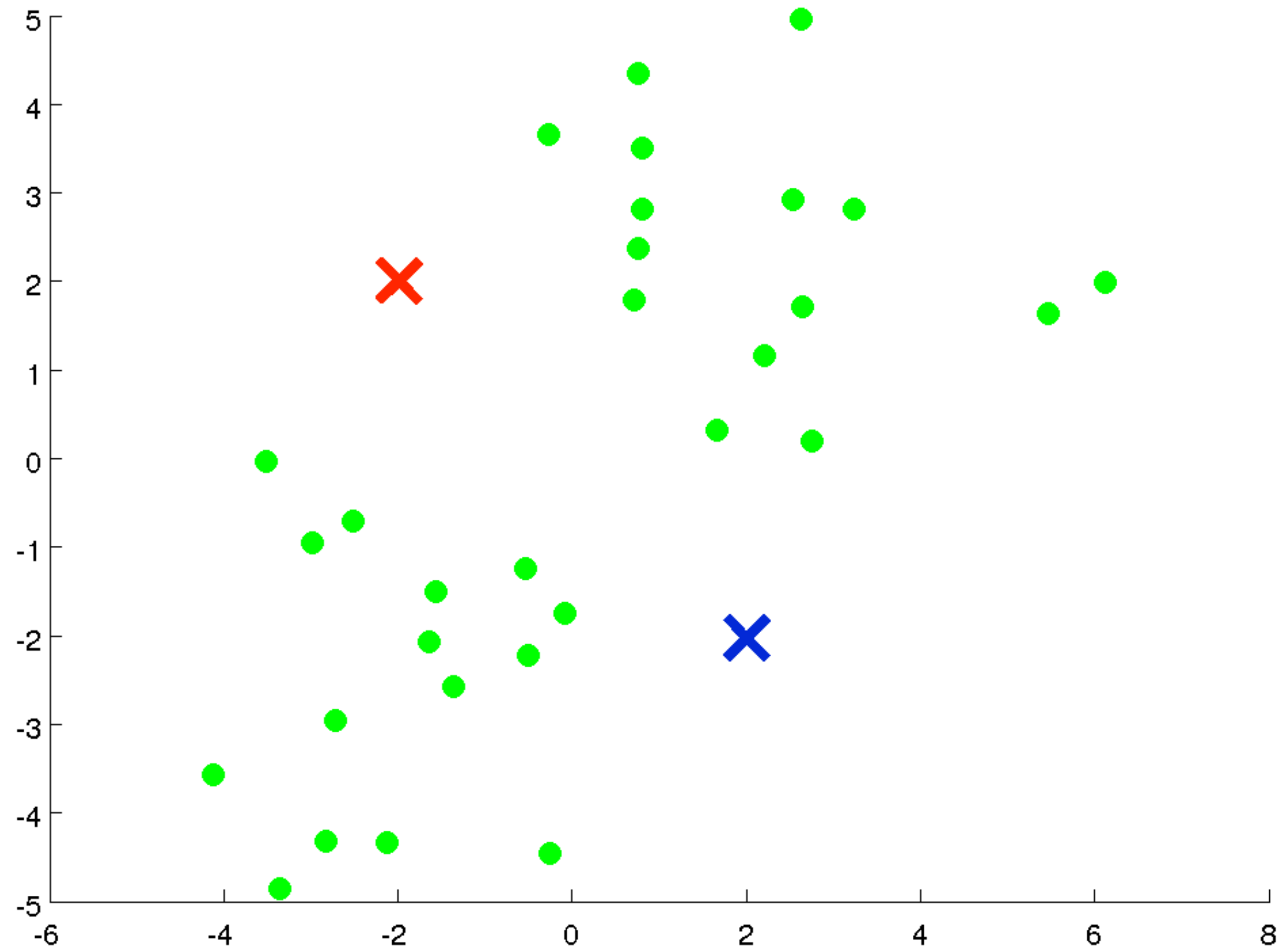
$x^{(i)}$: i 번째 data

$c^{(i)}$: $x^{(i)}$ 가 있는 그룹

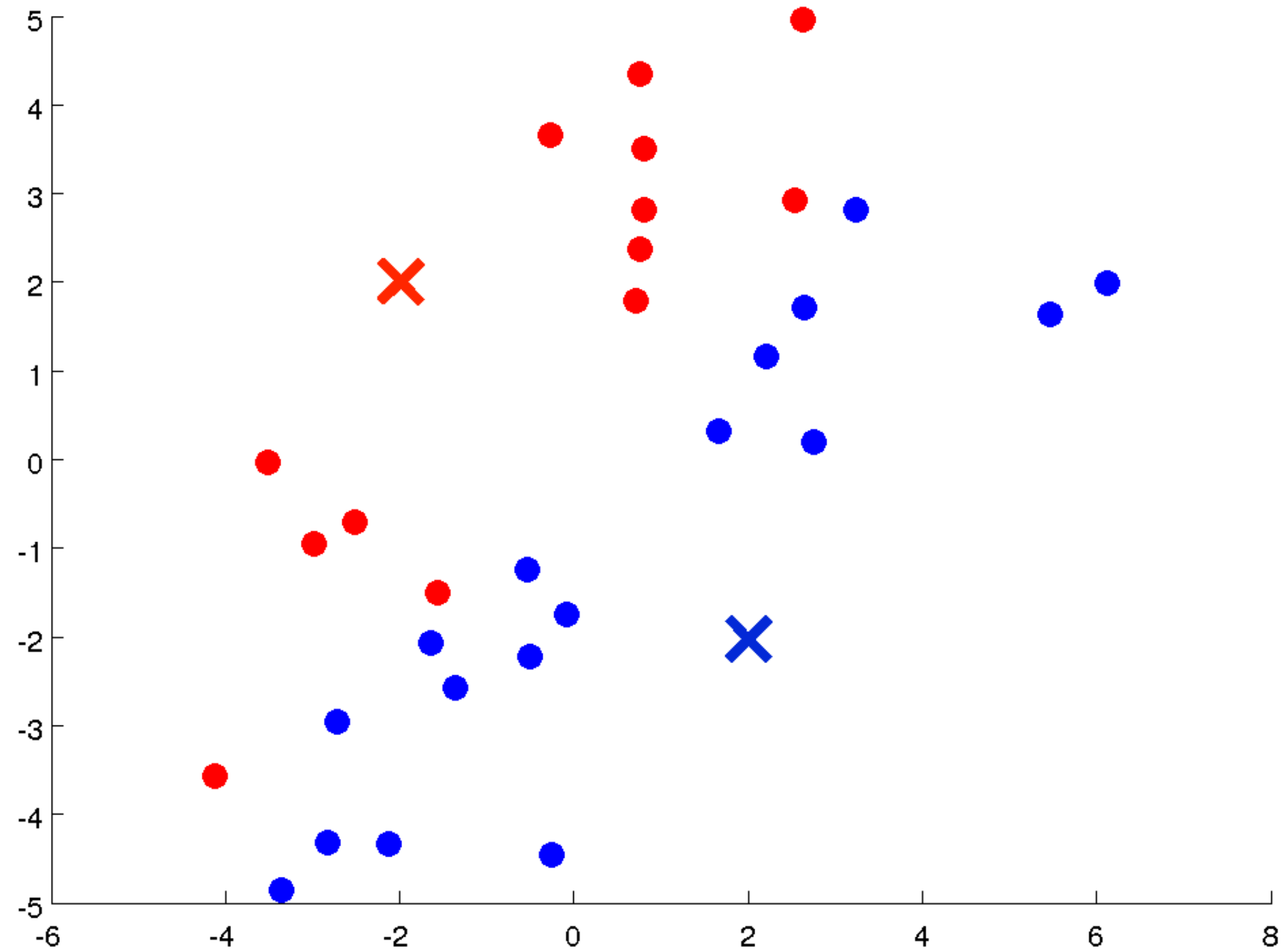
μ_j : j 번째 그룹의 무게 중심

$\mu_{c^{(i)}}$: $x^{(i)}$ 가 속한 그룹의 무게 중심

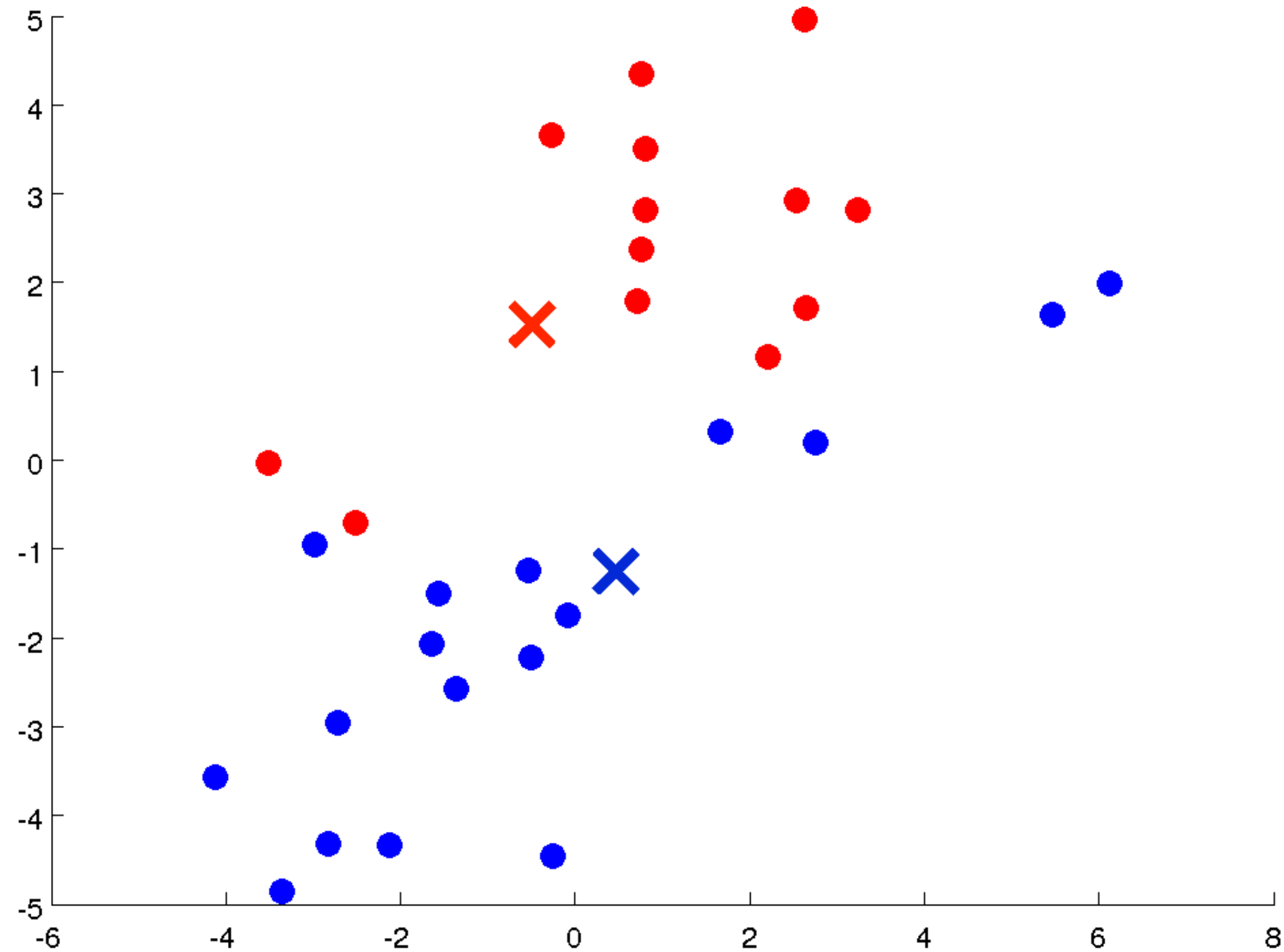
2-1. K-means



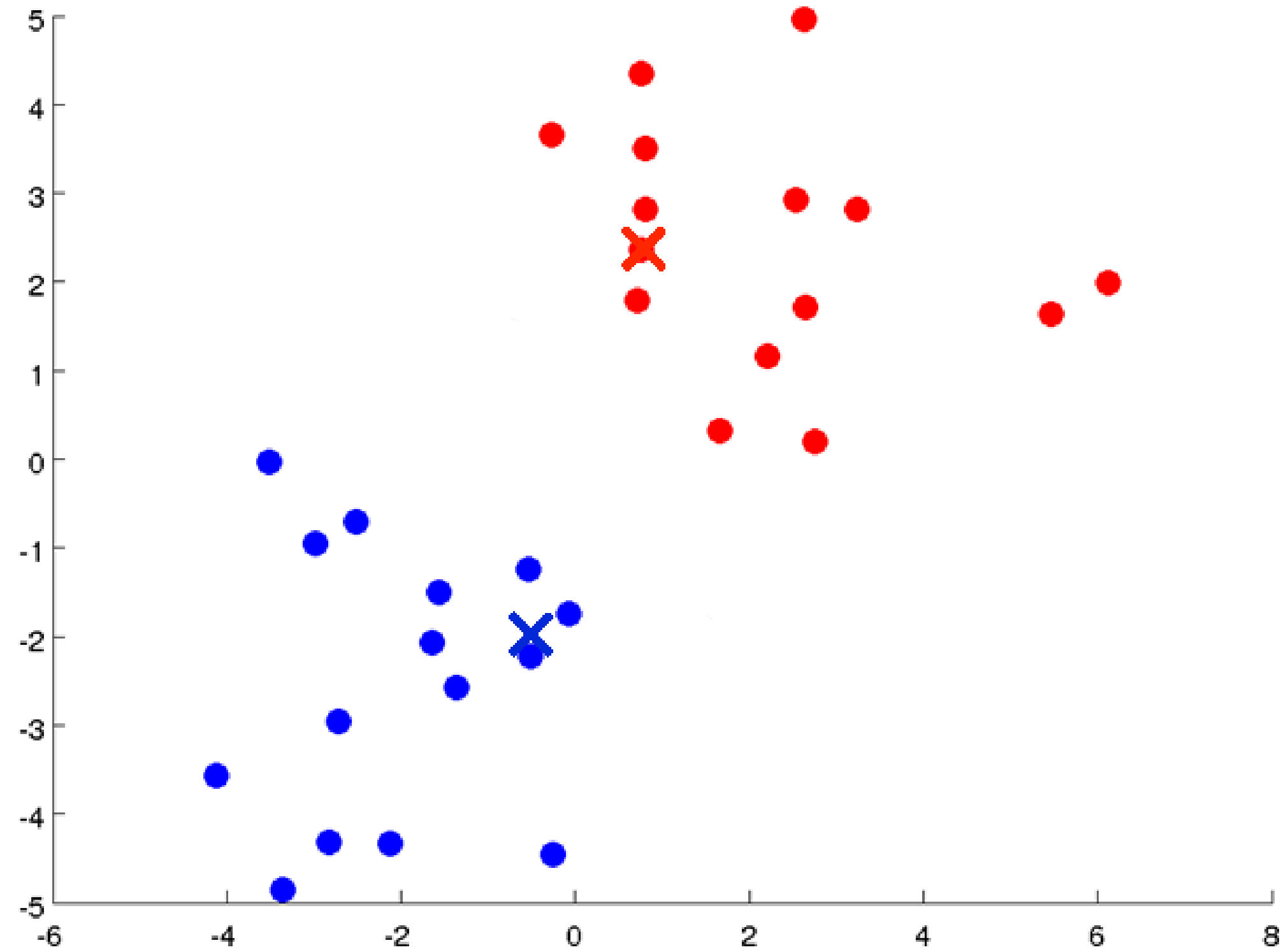
2-1. K-means



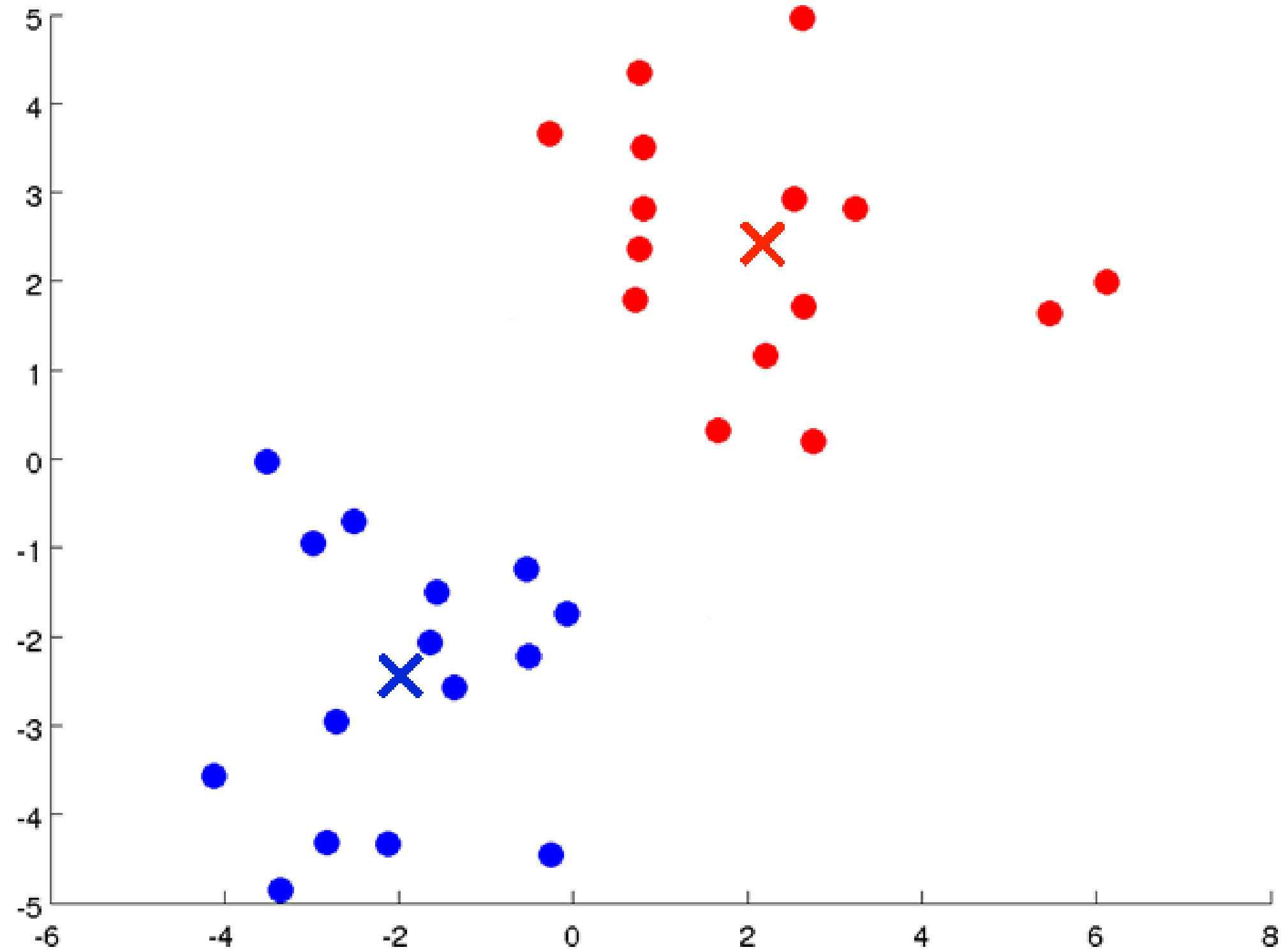
2-1. K-means



2-1. K-means



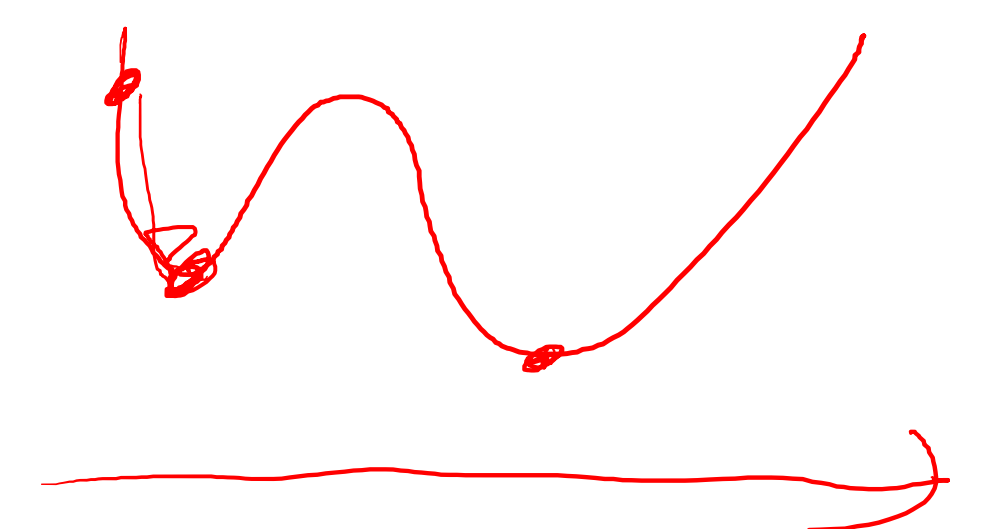
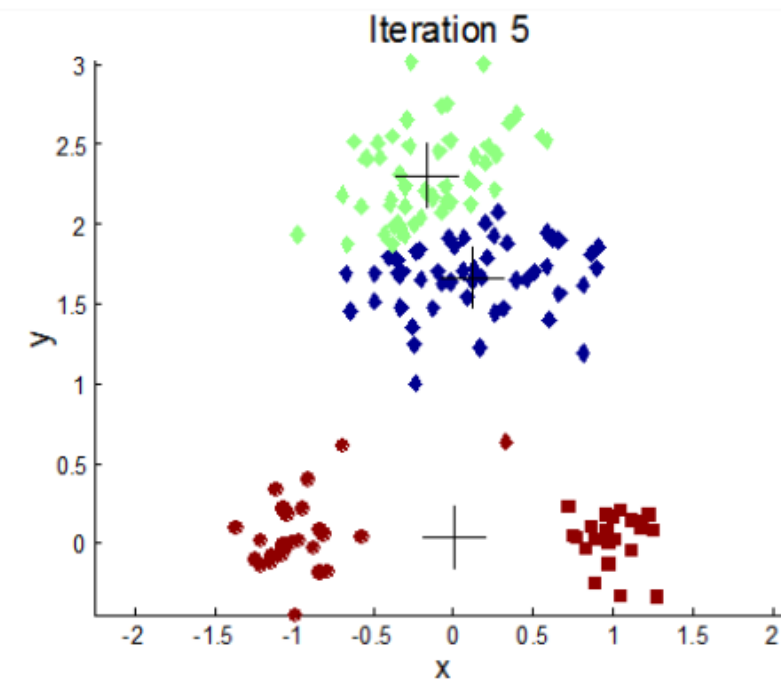
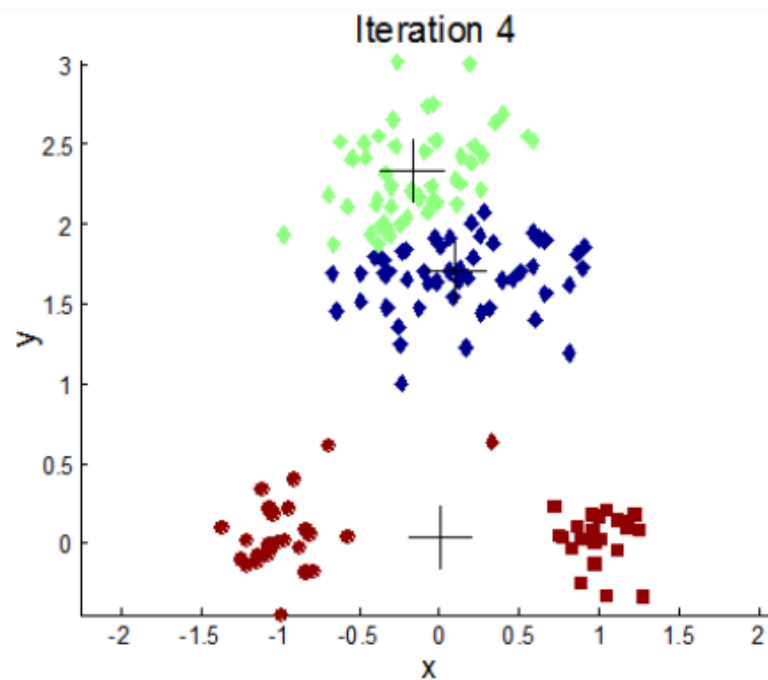
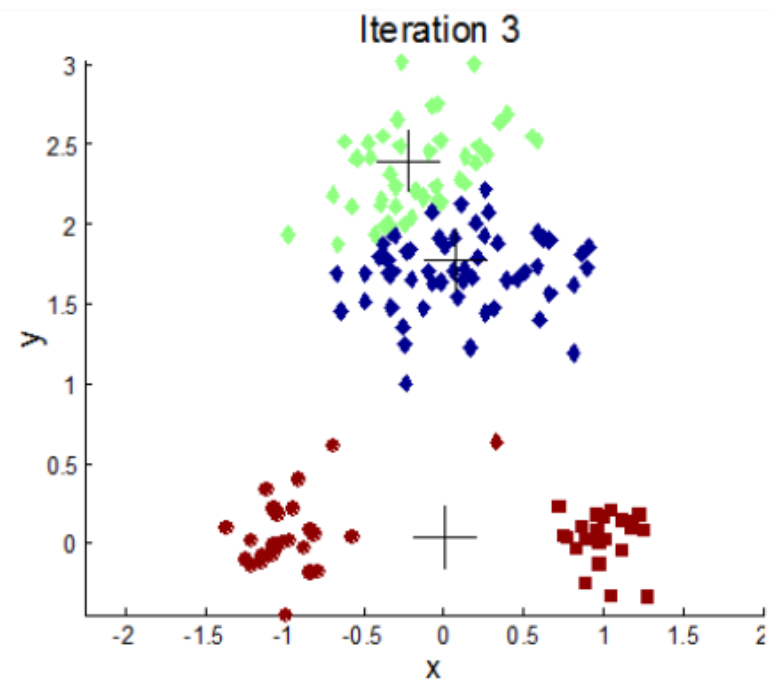
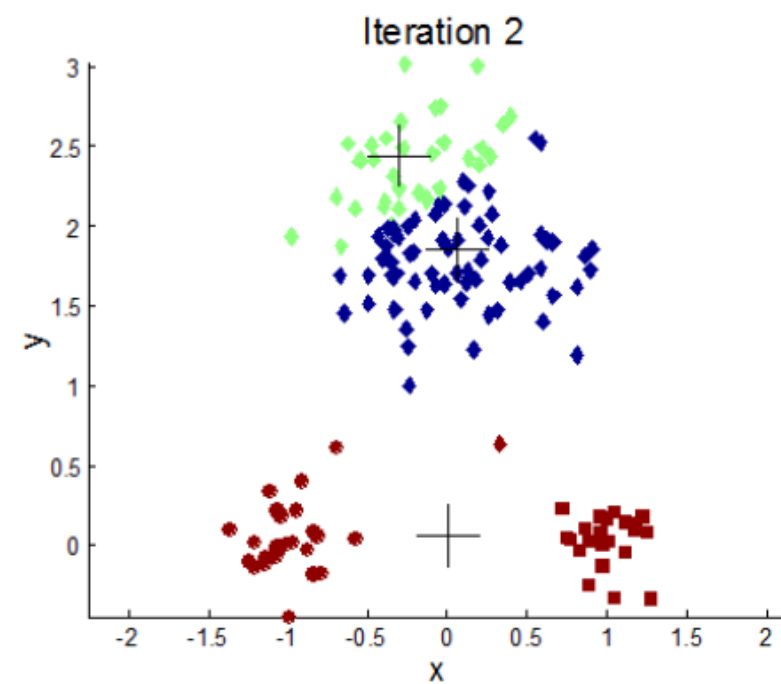
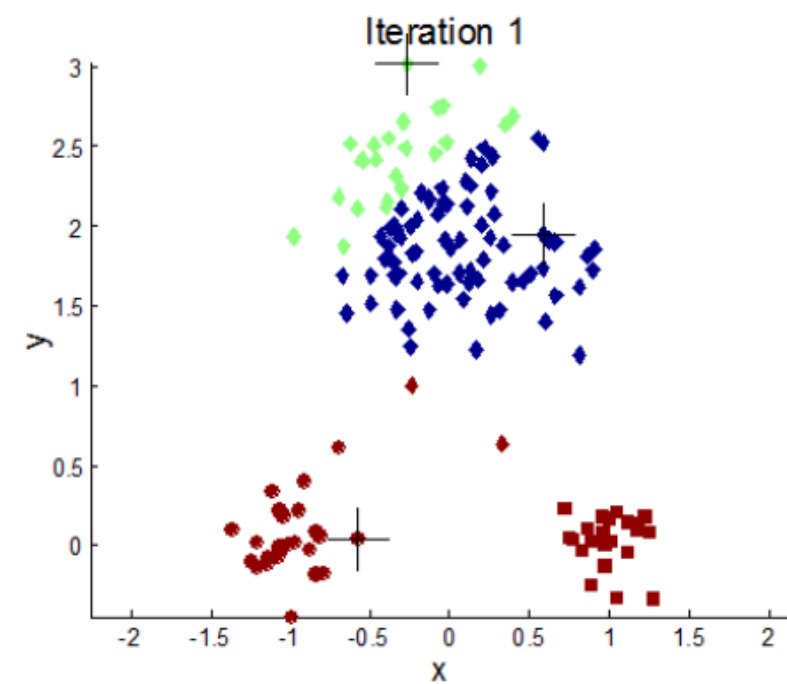
2-1. K-means



2-1. K-means

- **Local Minima**

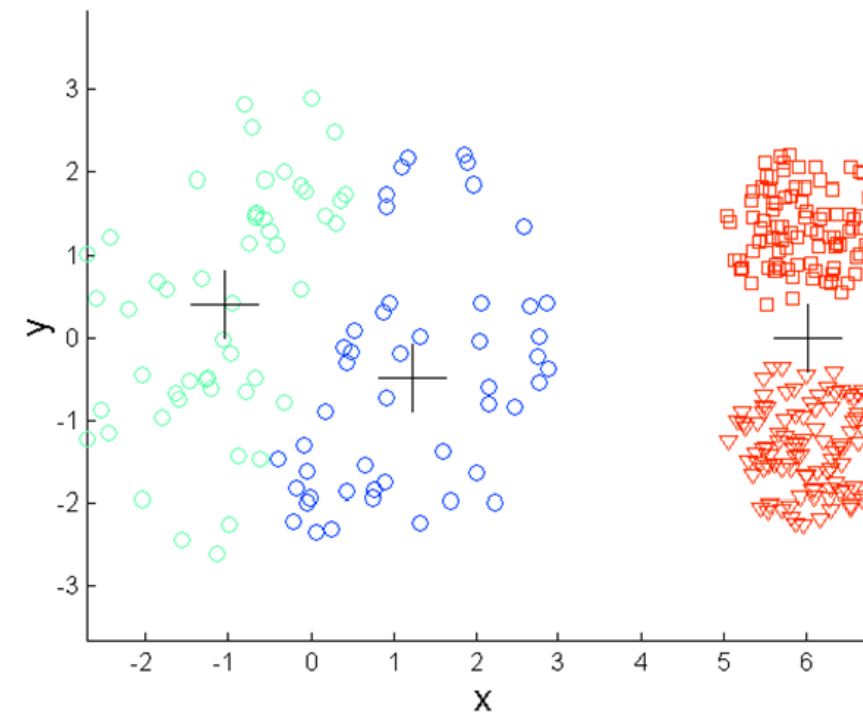
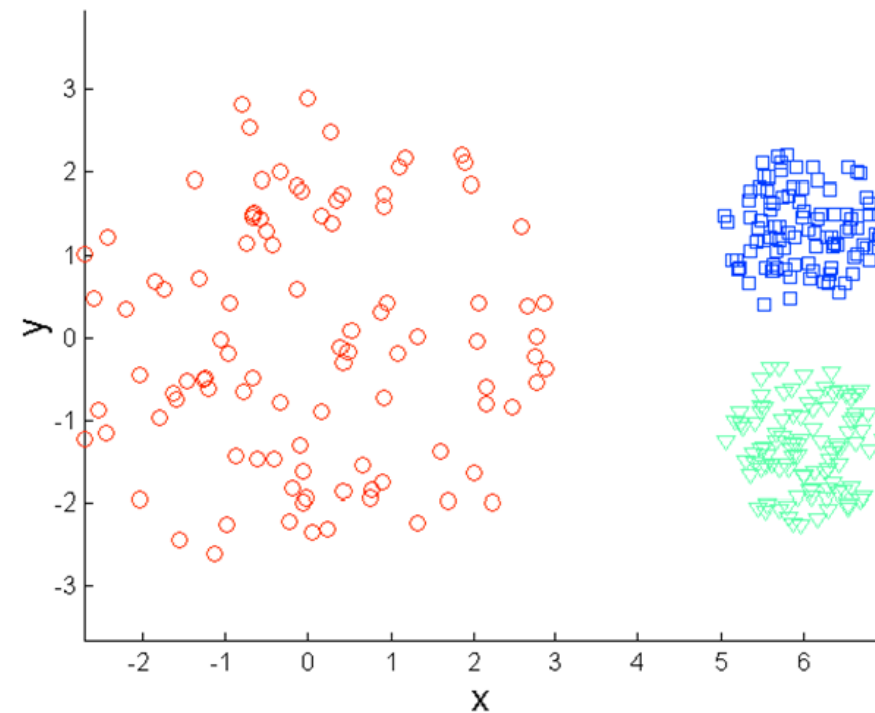
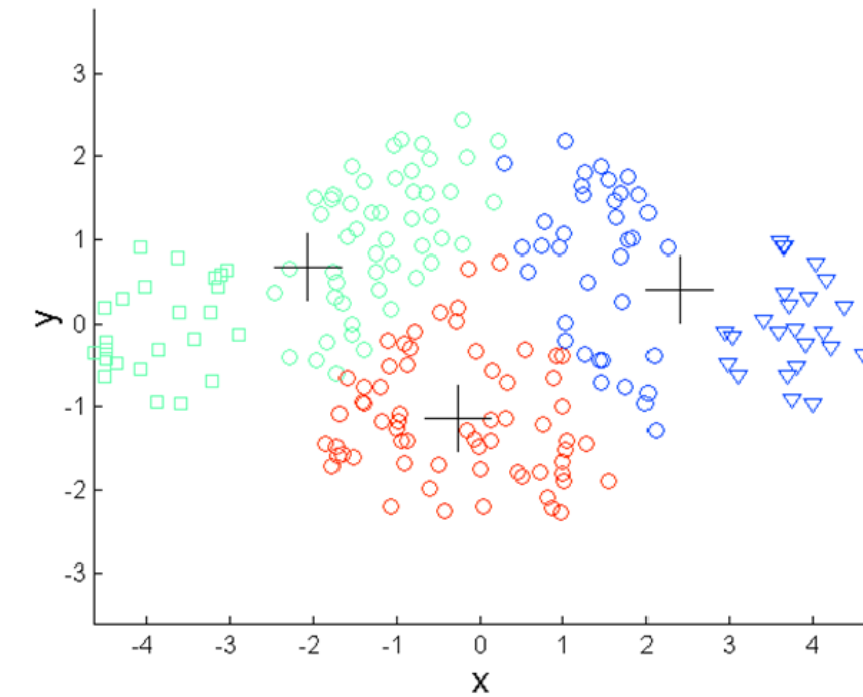
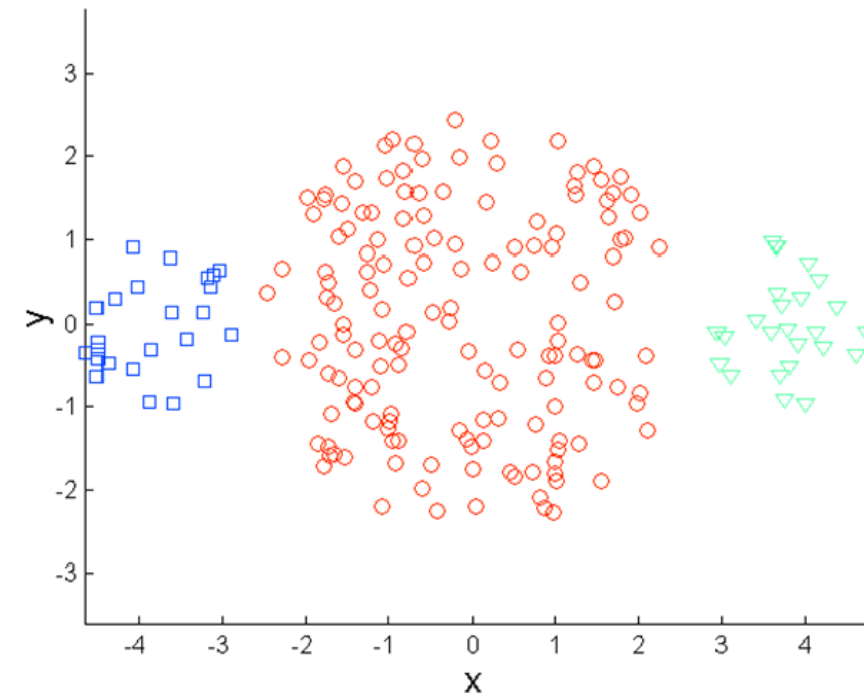
초기값 위치에 따라 원하는 결과가 나오지 않을 수 있음



2-1. K-means

- **Local Minima**

클러스터의 크기나 밀도가 다를 경우 원치않는 결과가 나올수 있음

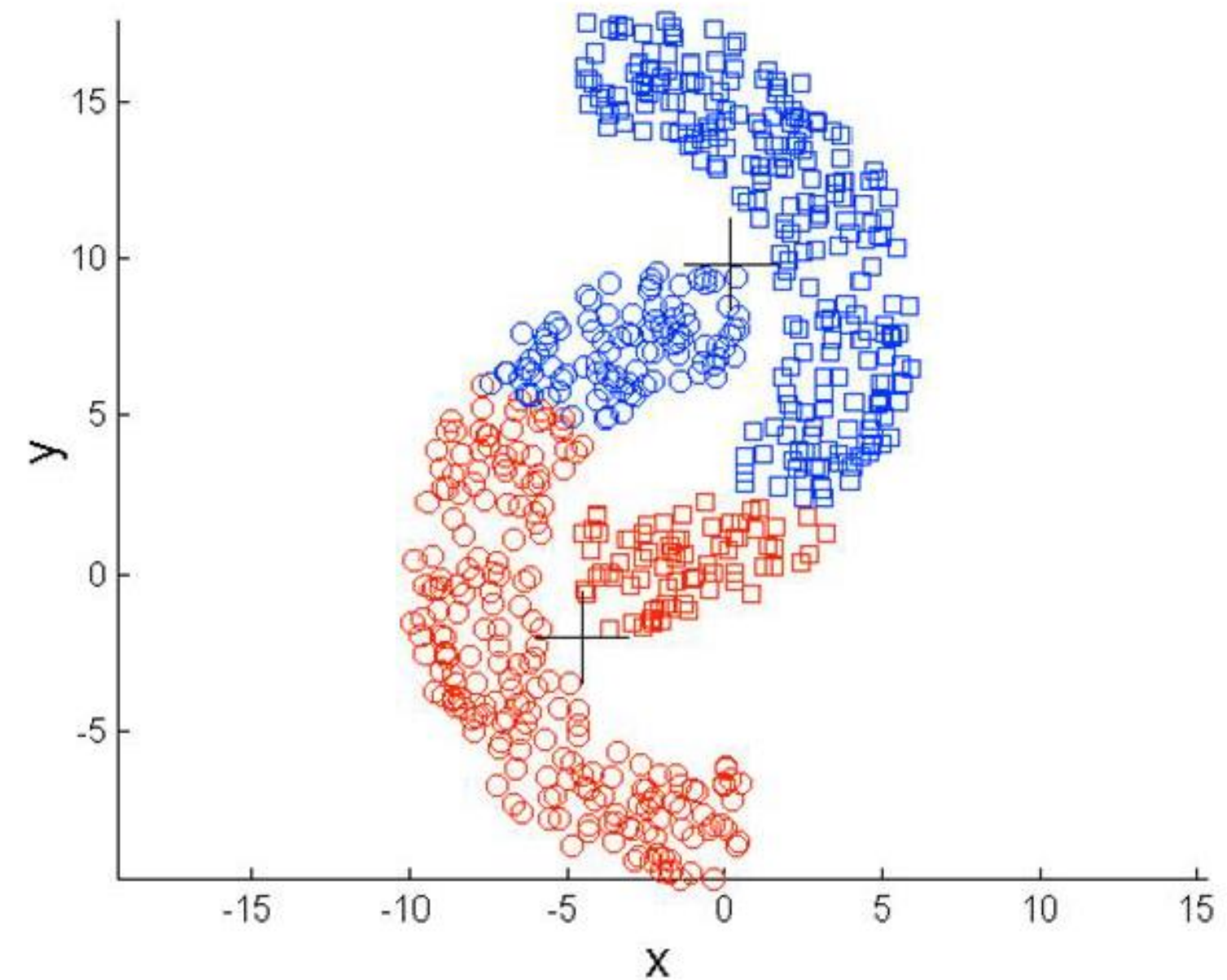
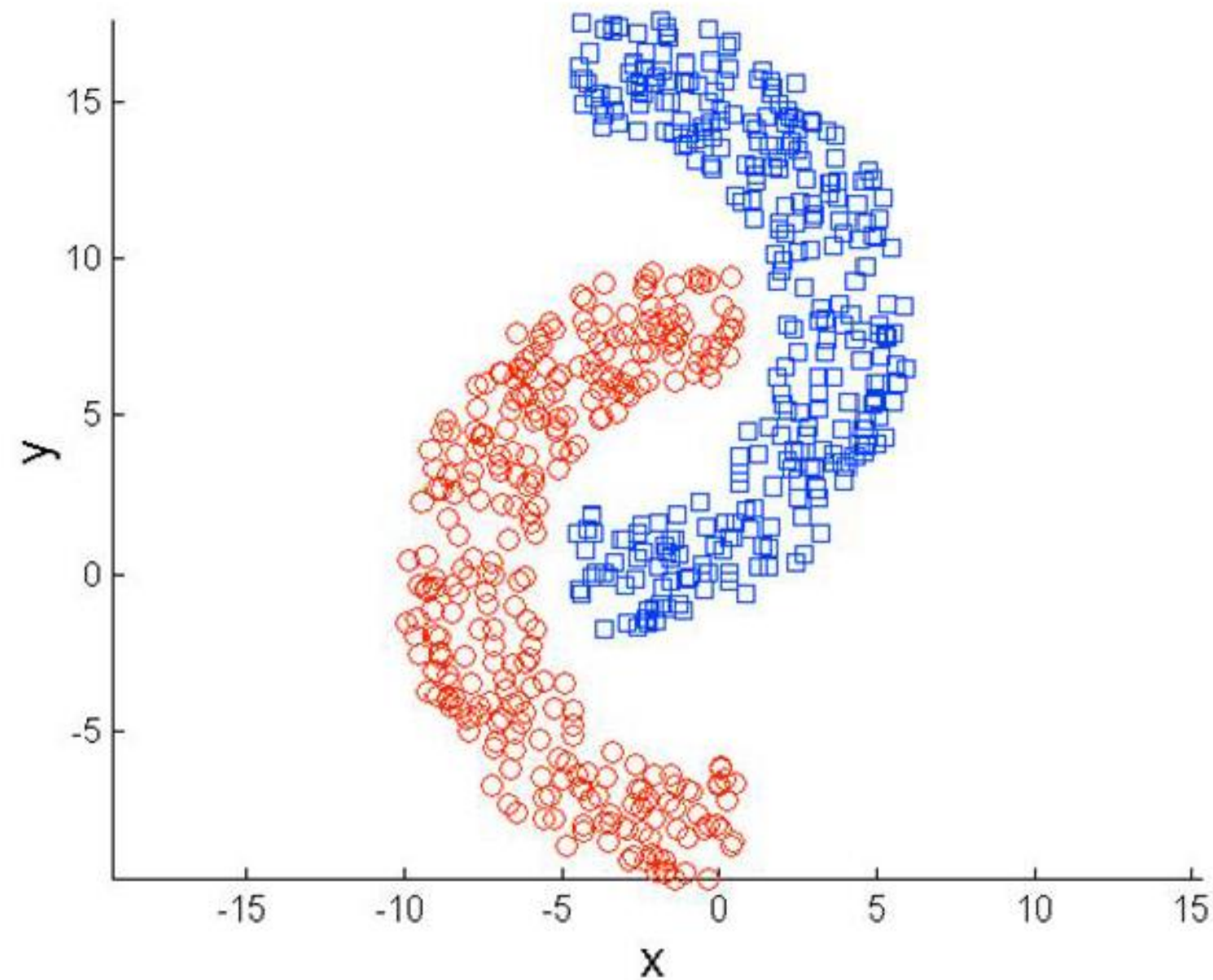


2-1. K-means

$$\underbrace{\|x^i - \mu\|^2}$$

- **Local Minima**

데이터의 분포가 특이한 경우에도 잘 작동하지 않을 수 있음



2-1. K-means

- 시간 복잡도(Time Complexity)

Big O notation

$$f(n) = O(g(n))$$

이라는 것은 충분히 큰 모든 자연수 n 에 대해
 $|f(n)| \leq C|g(n)|$ 가 성립한다는 뜻

- 시간복잡도가 $O(n)$ 으로 가벼운 편
- 실제 문제에 적용할 때는 여러번 클러스터링을 수행해 가장 빈번히 등장하는 군집에 할당하는 majority voting 방법을 쓰는 경우가 많음



n : 데이터 개수

$g(n)$: 걸리는 시간

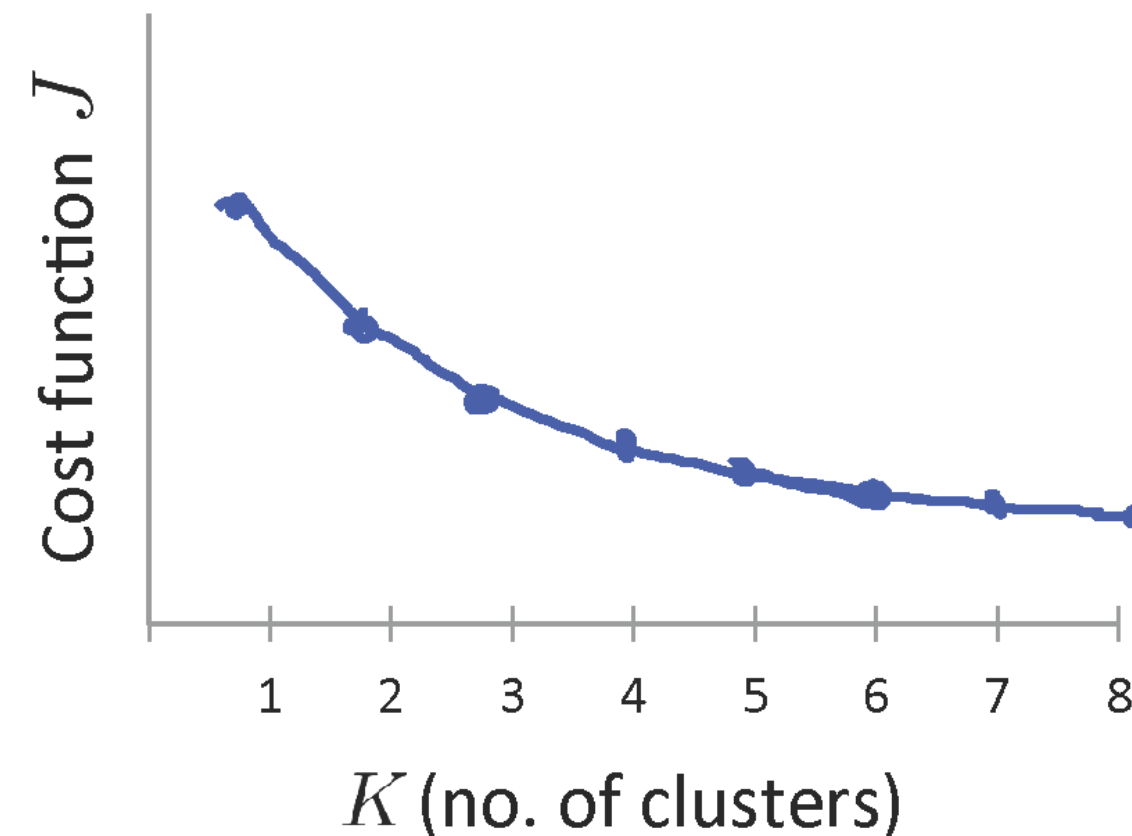
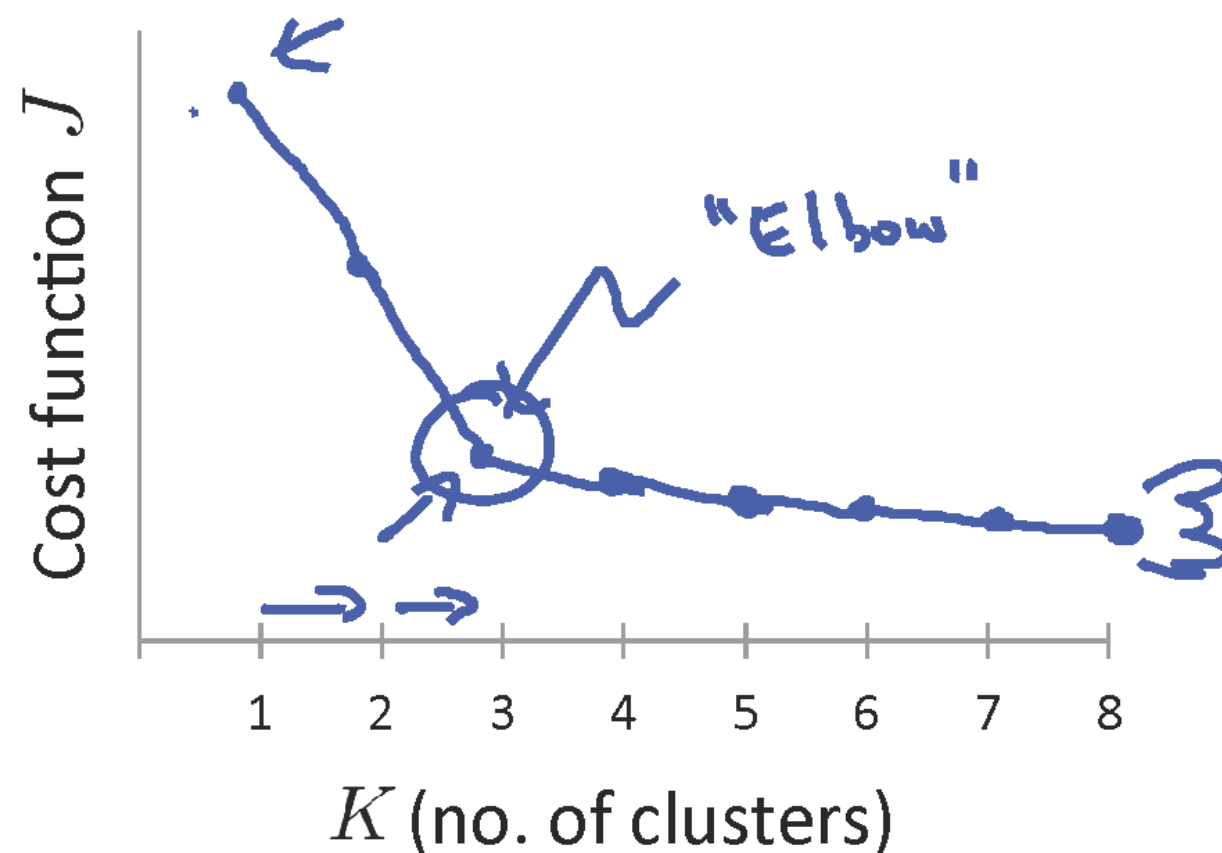
2-1. K-means

- 하이퍼 파라미터 K 정하기

Elbow Method

다양한 K 의 값을 시도해 보고

그림과 같이 꺾이는 점이 있는 경우 최적이라고 판단

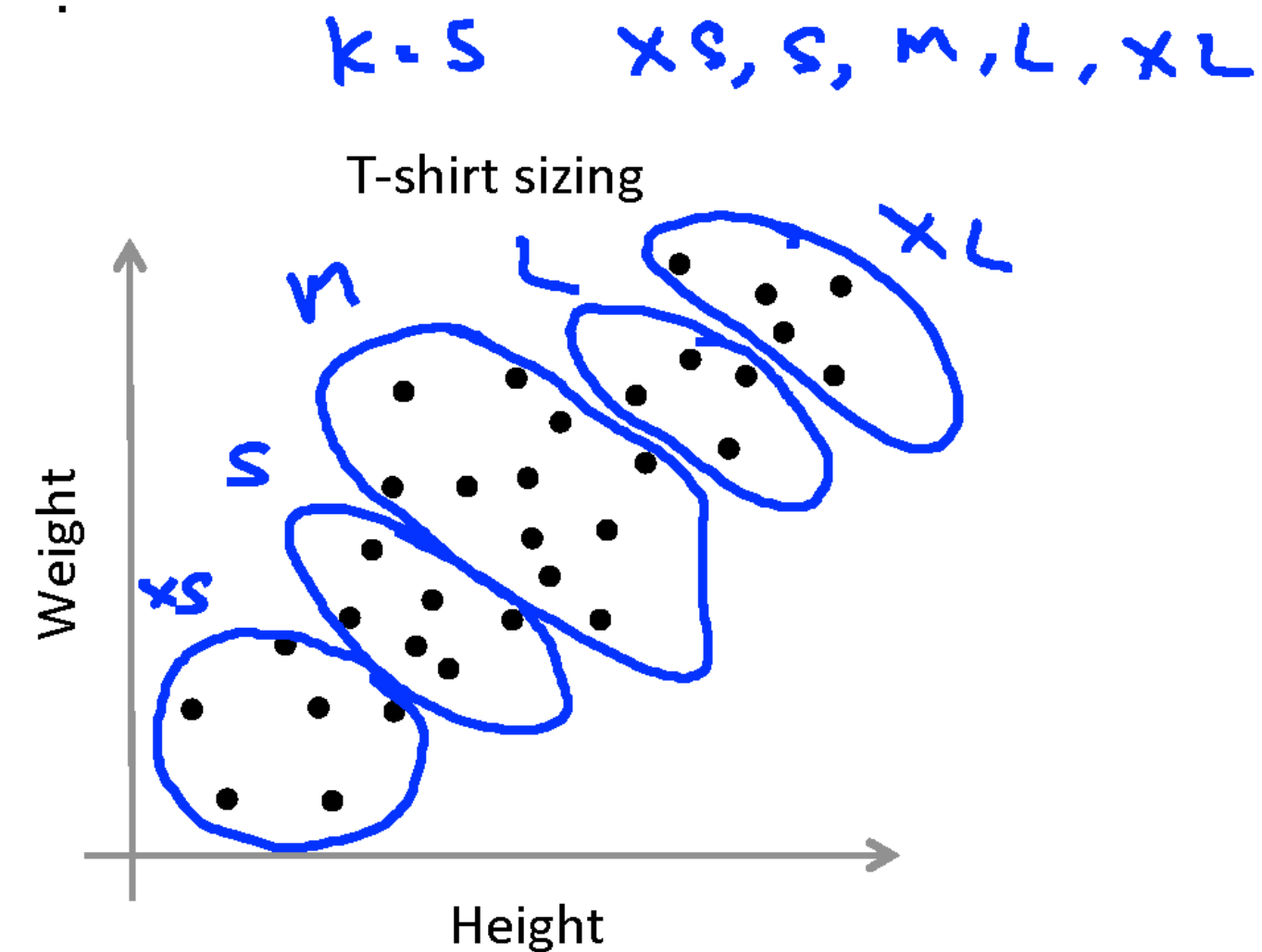
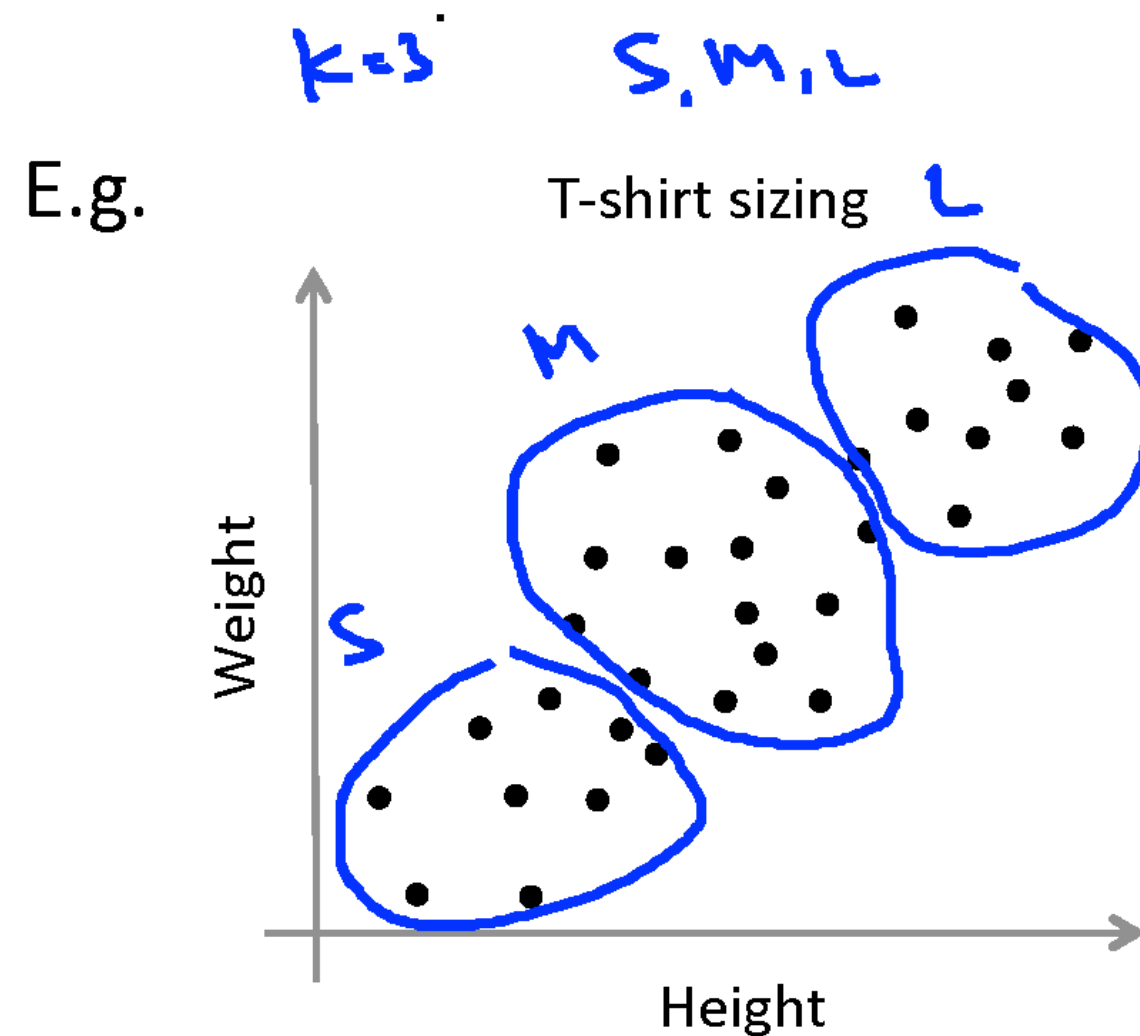


2-1. K-means

밀도기반 클러스터링 (DBSCAN)
GMM

- 하이퍼 파라미터 K 정하기

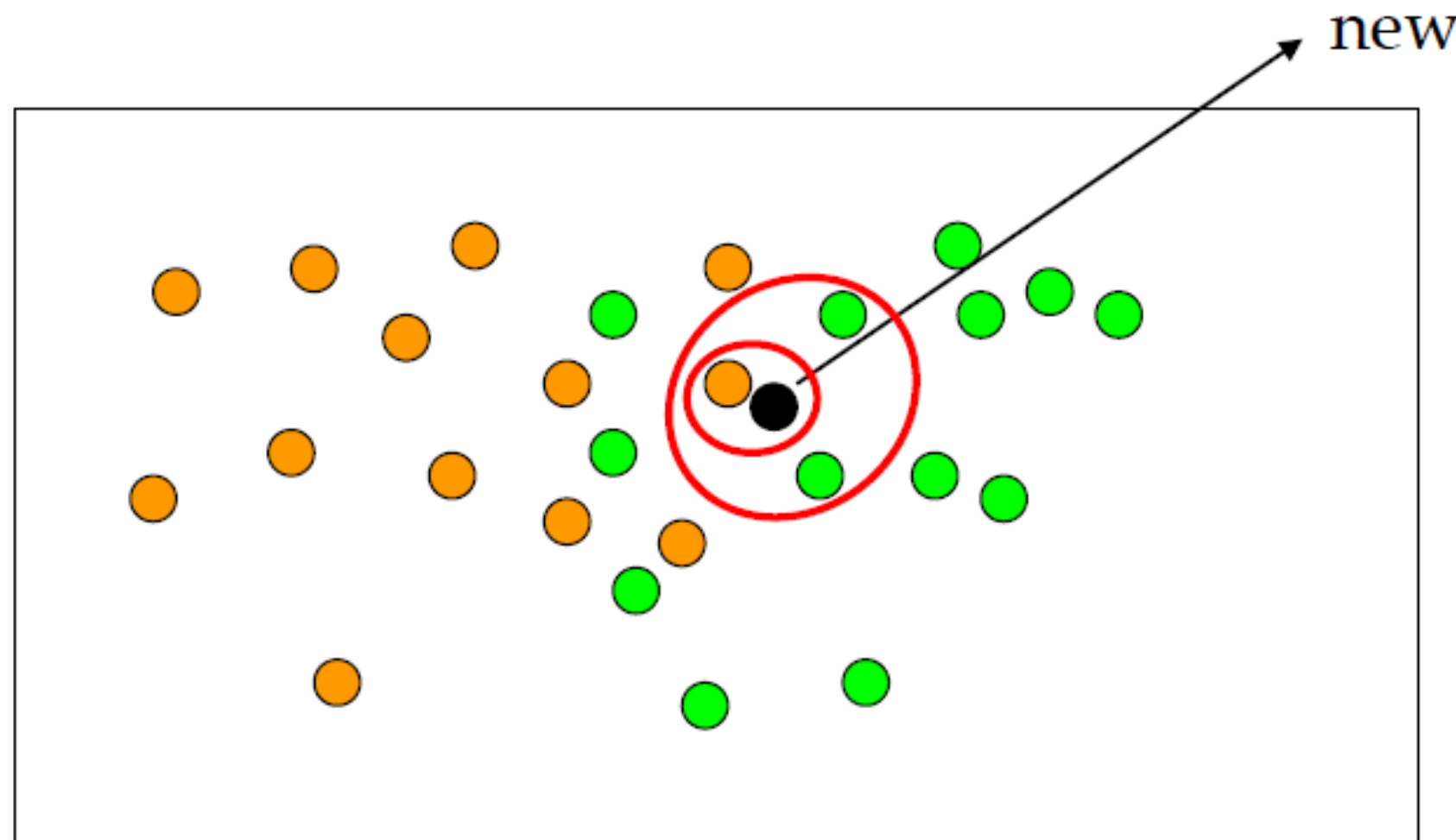
티셔츠 사이즈 분류와 같이 필요와 목적에 맞게 정할수도 있음



2-2. KNN

- 새로운 데이터가 주어졌을 때 기존 데이터 가운데 가장 가까운 k개 이웃의 정보로 새로운 데이터를 예측하는 방법론 (Supervised)
- 레이지(lazy) 모델 – 딱히 학습이라고 할 만한 것이 없음
- 그림에서 검은색 점의 범주 정보는 주변 이웃들을 가지고 추론
- k가 1이면 오렌지색, k가 3라면 녹색으로 분류

x, y

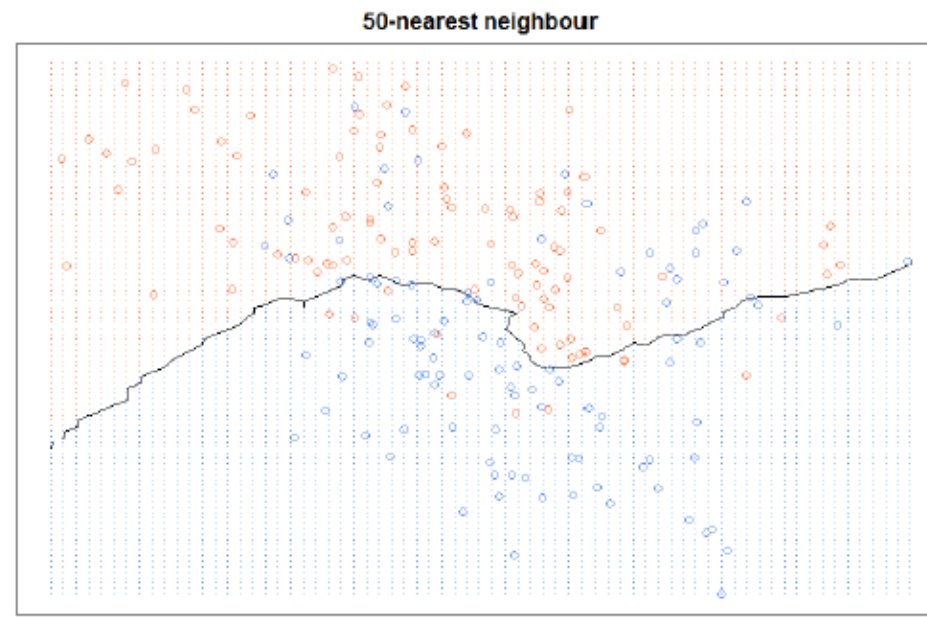
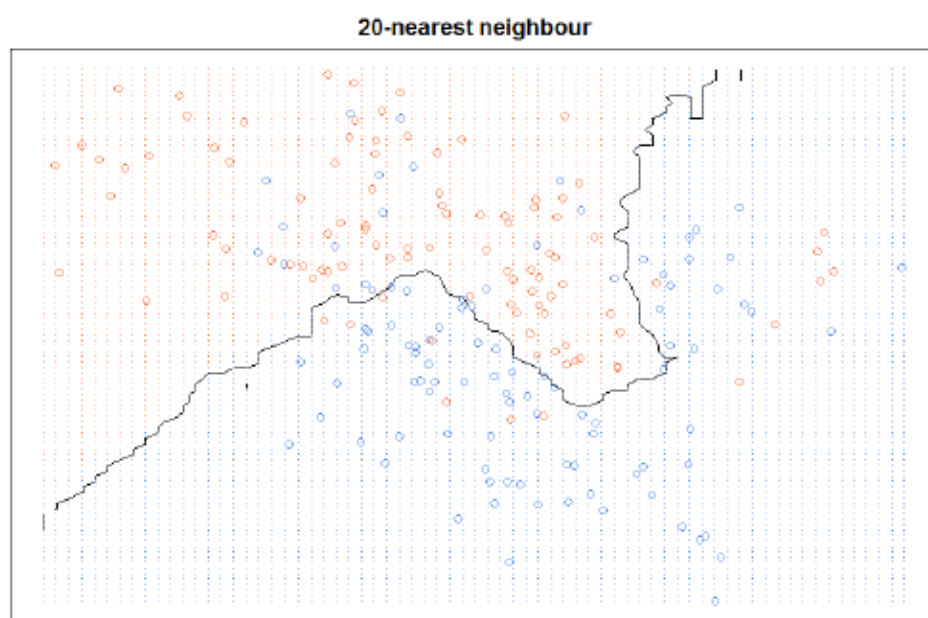
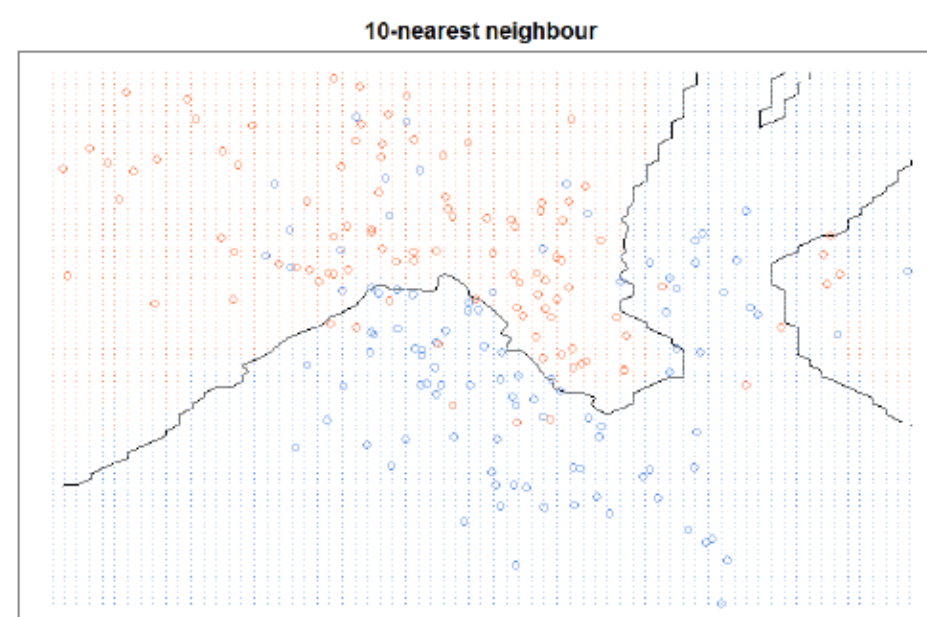
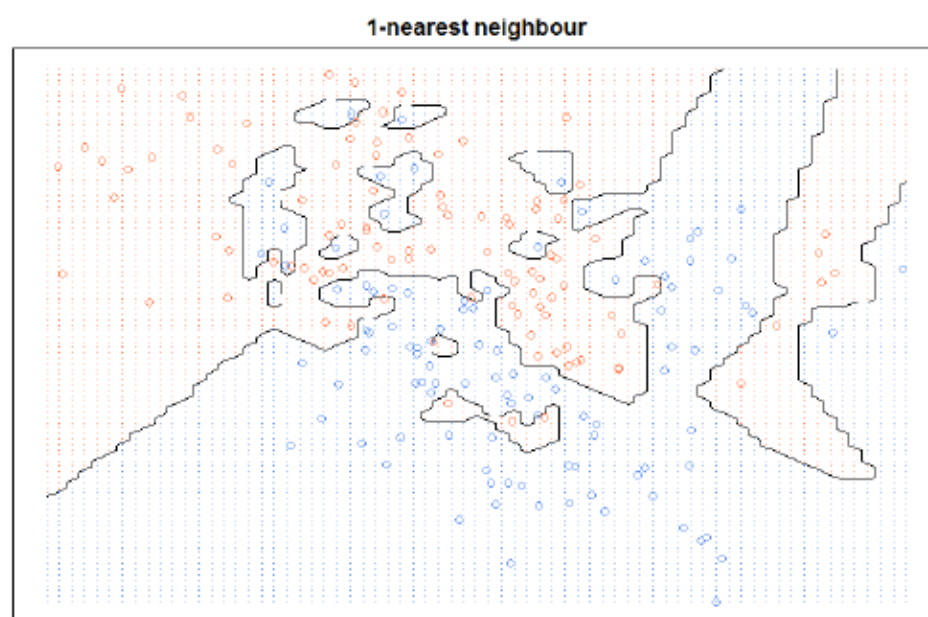


x_1 x_2
 (x_1, y_1) (x_2, y_2)

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

2-2. KNN

- 하이퍼 파라미터는 탐색할 이웃 수(k), 거리 측정방법의 두가지
- k 가 작을 경우 오버피팅,
반대로 매우 클 경우 언더피팅이 일어나는 경향이 있음

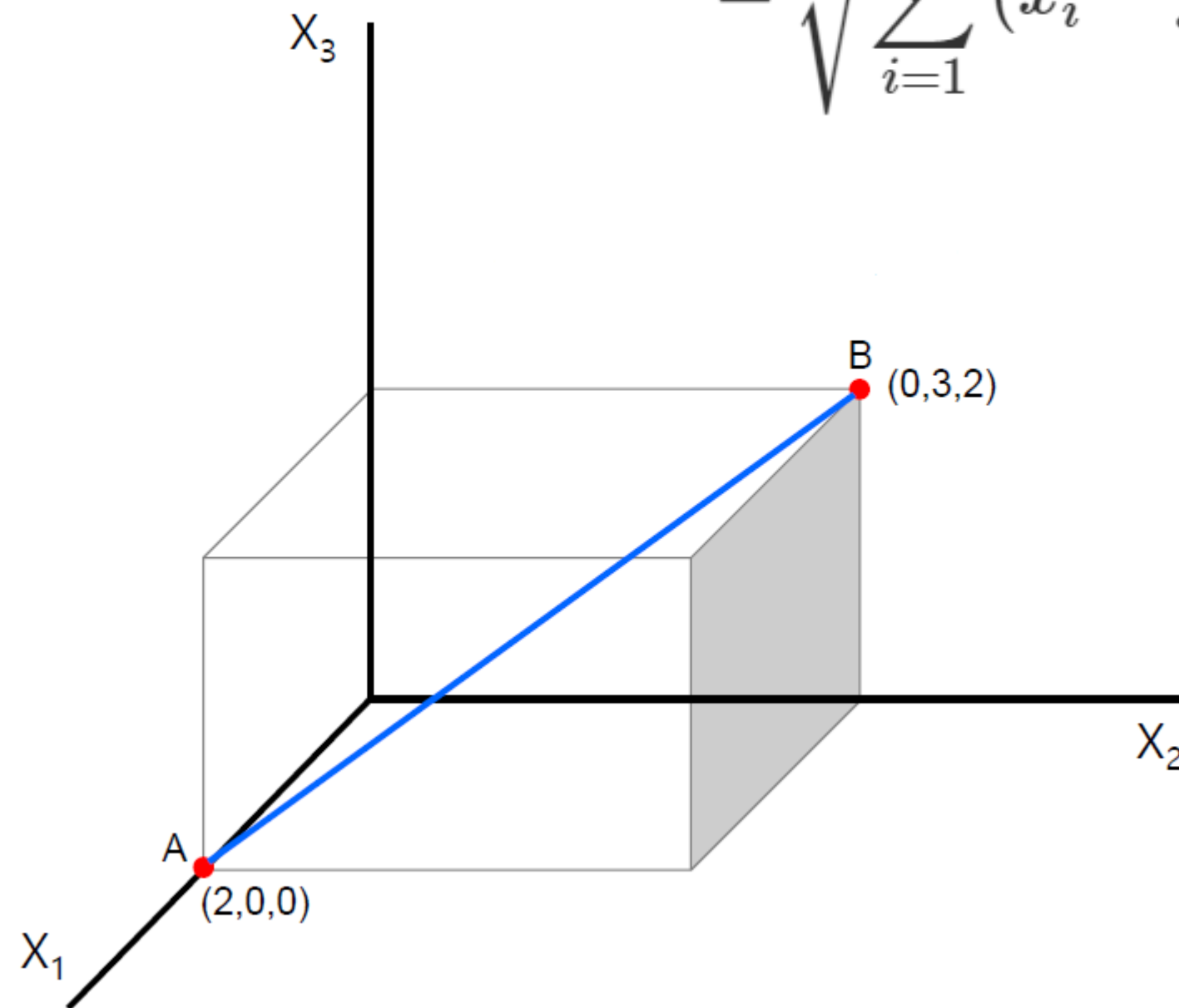


2-2. KNN

- 거리지표

유클리드 거리

$$\begin{aligned}d_{(X,Y)} &= \sqrt{(x_1 - y_1)^2 + \dots + (x_n - y_n)^2} \\&= \sqrt{\sum_{i=1}^n (x_i - y_i)^2}\end{aligned}$$



10 $k=3$

2-2. KNN

• 맨해튼 거리 : 데이터의 분산

• 거리지표 $k=3$

데이터들간의 분산

맨해튼 거리

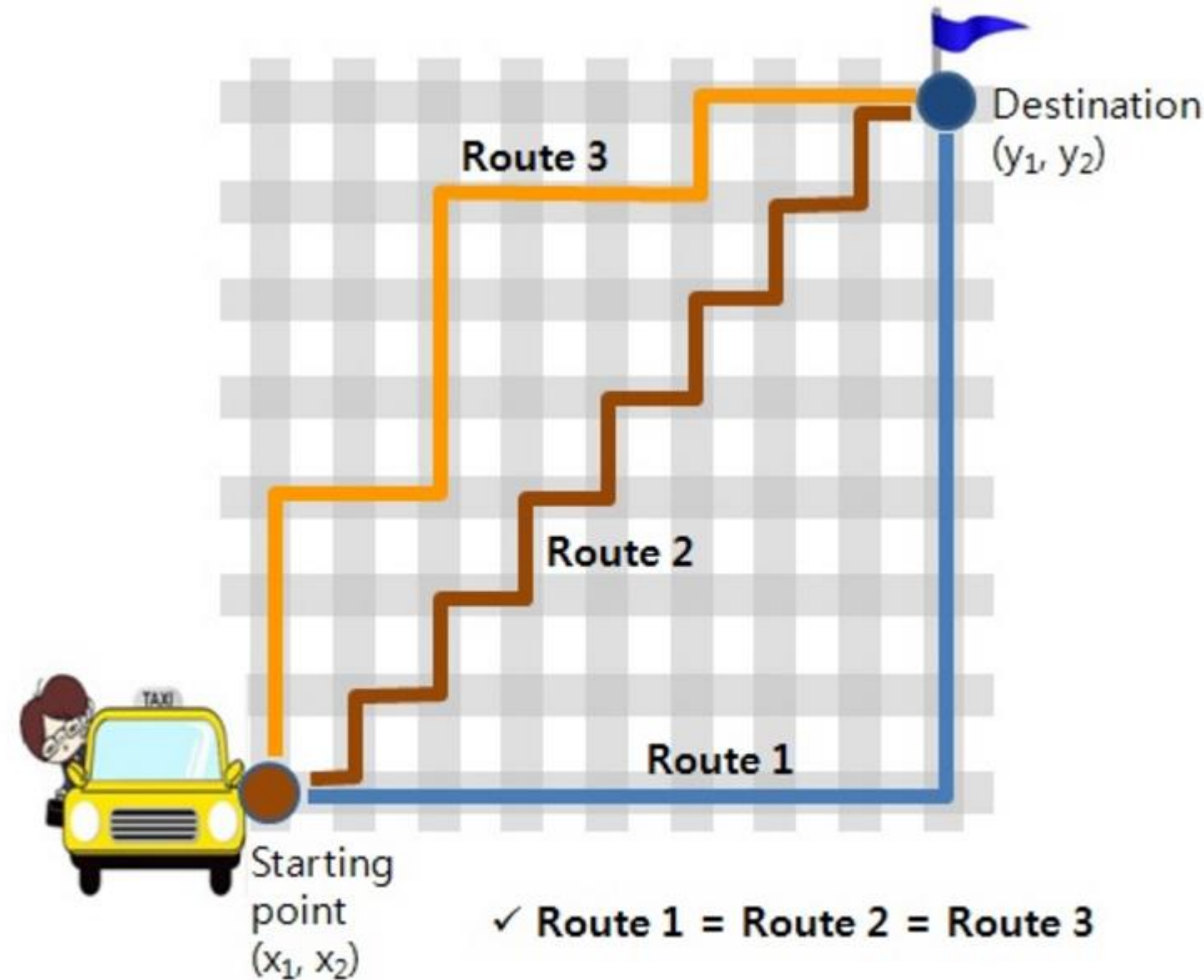
$$d_{\text{Manhattan}}(X,Y) = \sum_{i=1}^n |x_i - y_i|$$

(공분산)

+) 노이즈 감동하

-) 데이터 분포가 대
각각 2등

-) 계산 문제 걸림



[R 분석과 프로그래밍] <http://rfriend.tistory.com>

• Correlation

• Rank - correlation

• 데이터 normalization

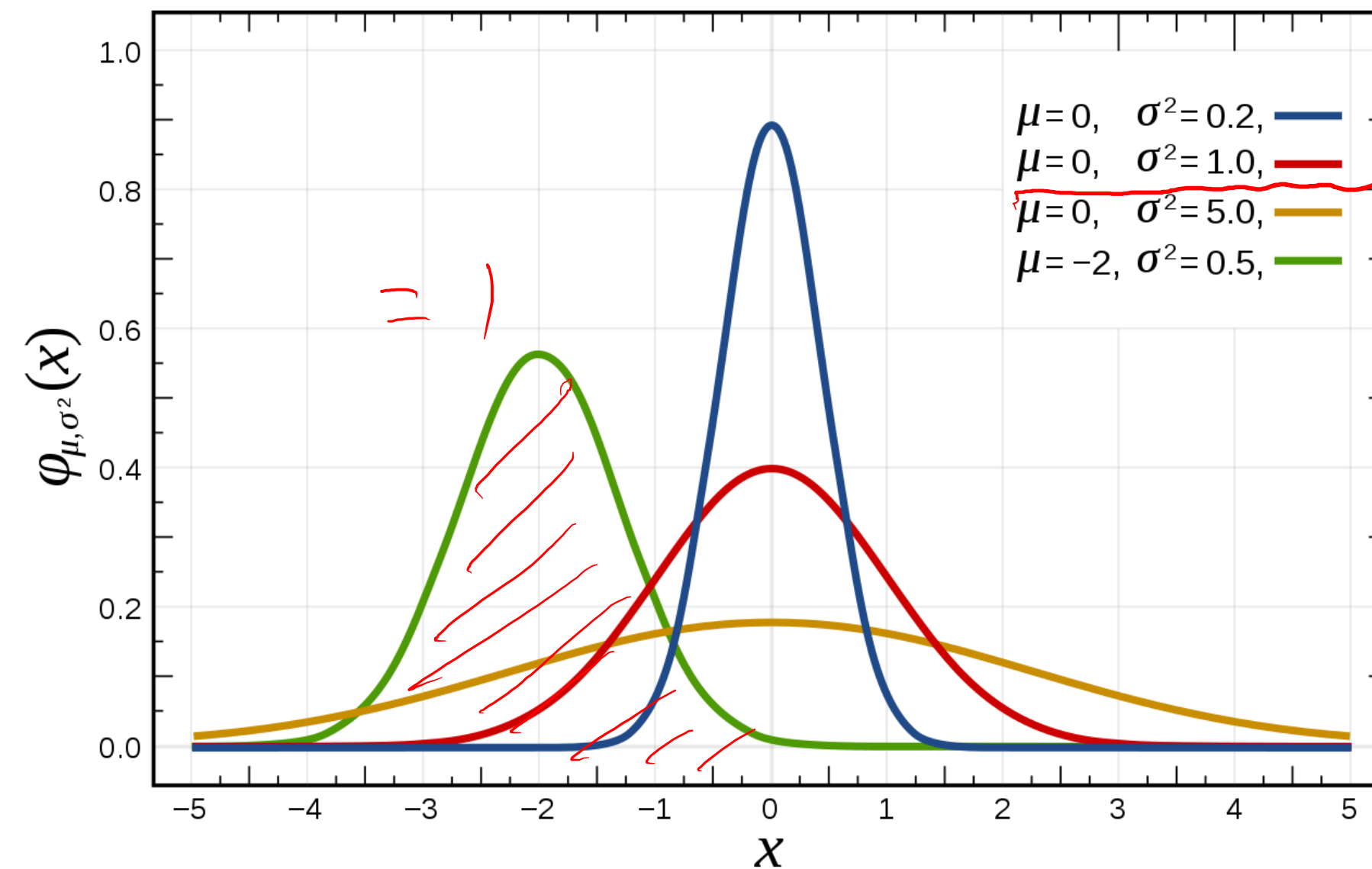
필수

$\frac{x_i - \mu_i}{\sigma_i}$

S_i

Gaussian Mixture Model (GMM)

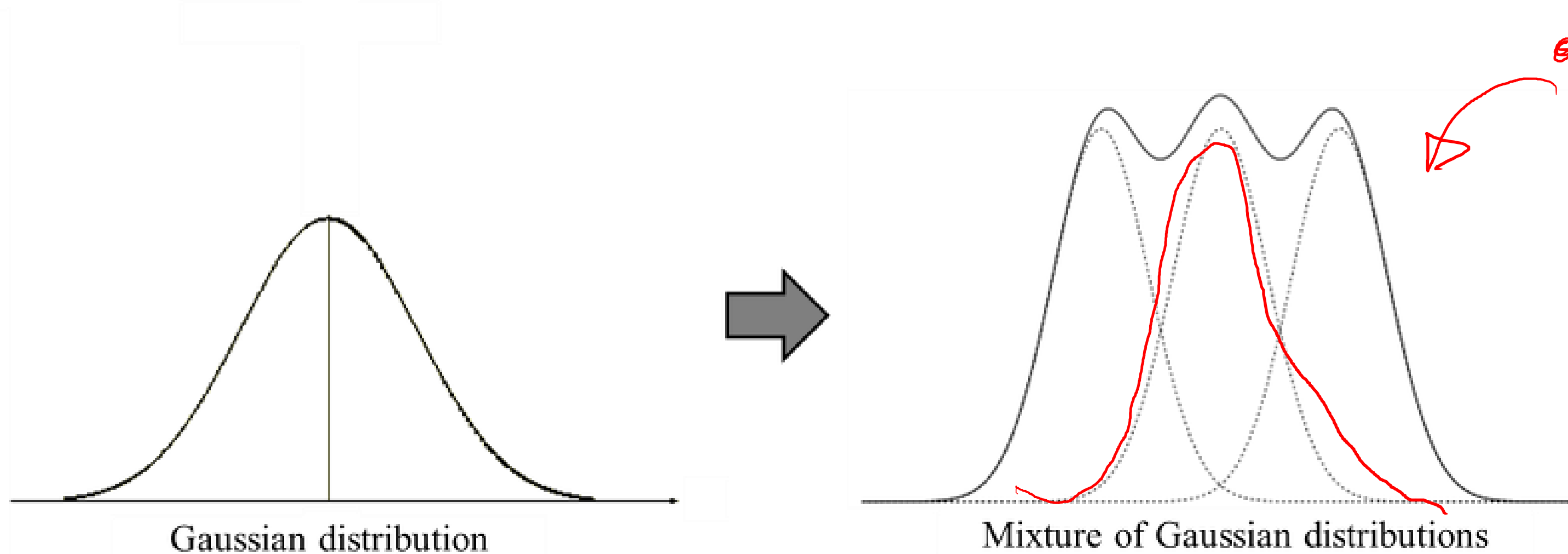
Gaussian Mixture Model (GMM)



- Gaussian 분포 (정규분포): $y = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$, μ : 평균 σ : 분산
- $\mu = 0$ 이고 $\sigma = 1$ 이면 표준 정규분포

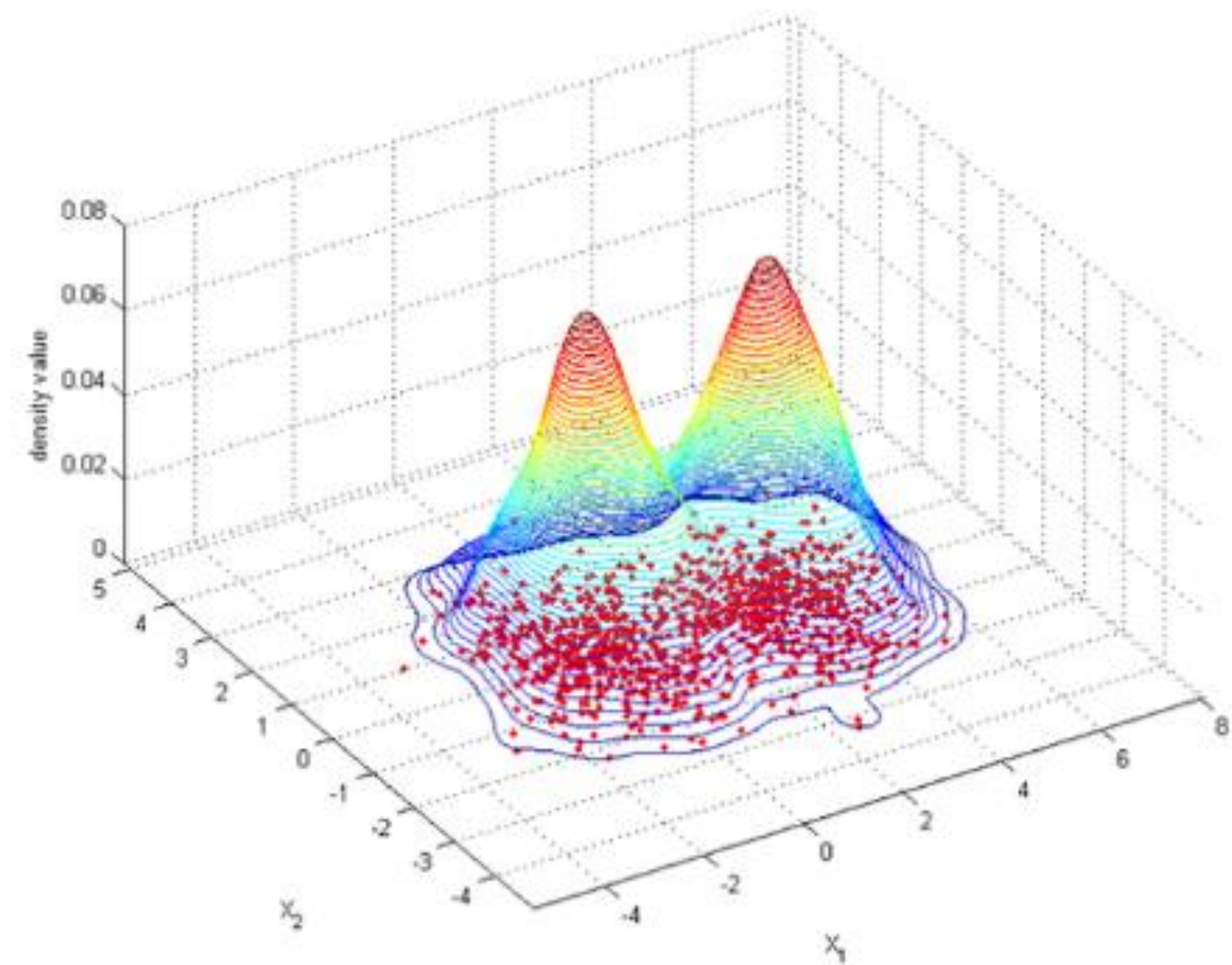
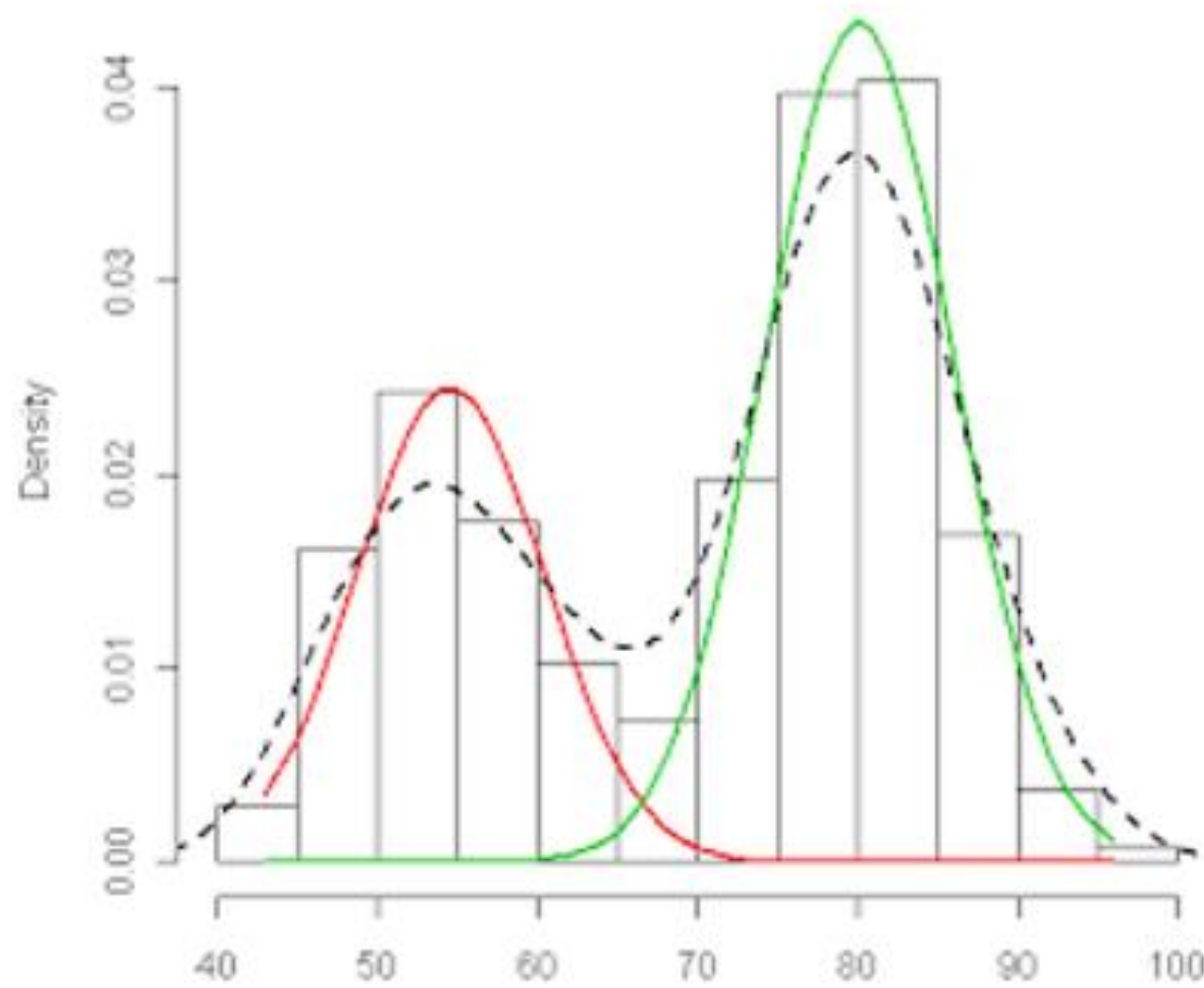
2-3. GMM과 EM 알고리즘

- GMM은 전체 데이터의 확률분포가 여러개의 정규분포의 조합으로 이루어져 있다고 가정하고 각 분포에 속할 확률이 높은 데이터끼리 클러스터링 하는 방법



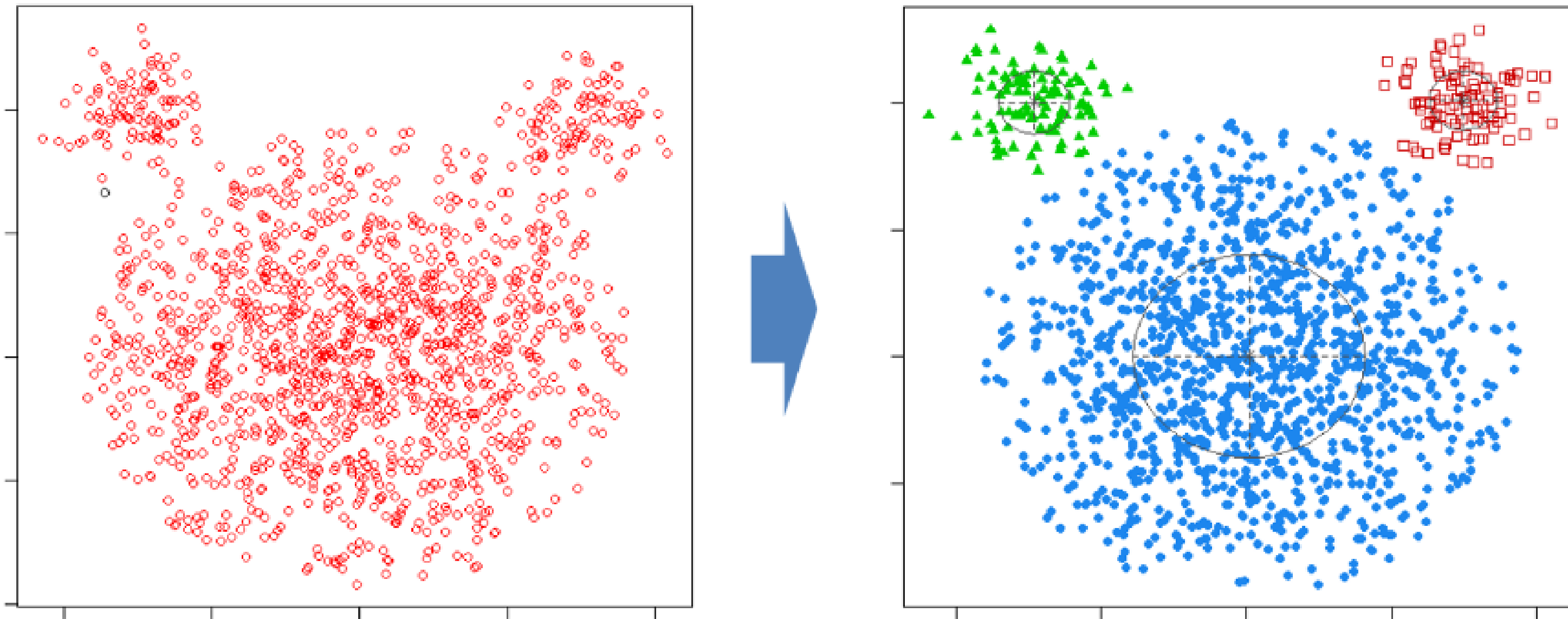
2-3. GMM과 EM 알고리즘

- 다변수 정규분포를 가정할수도 있음



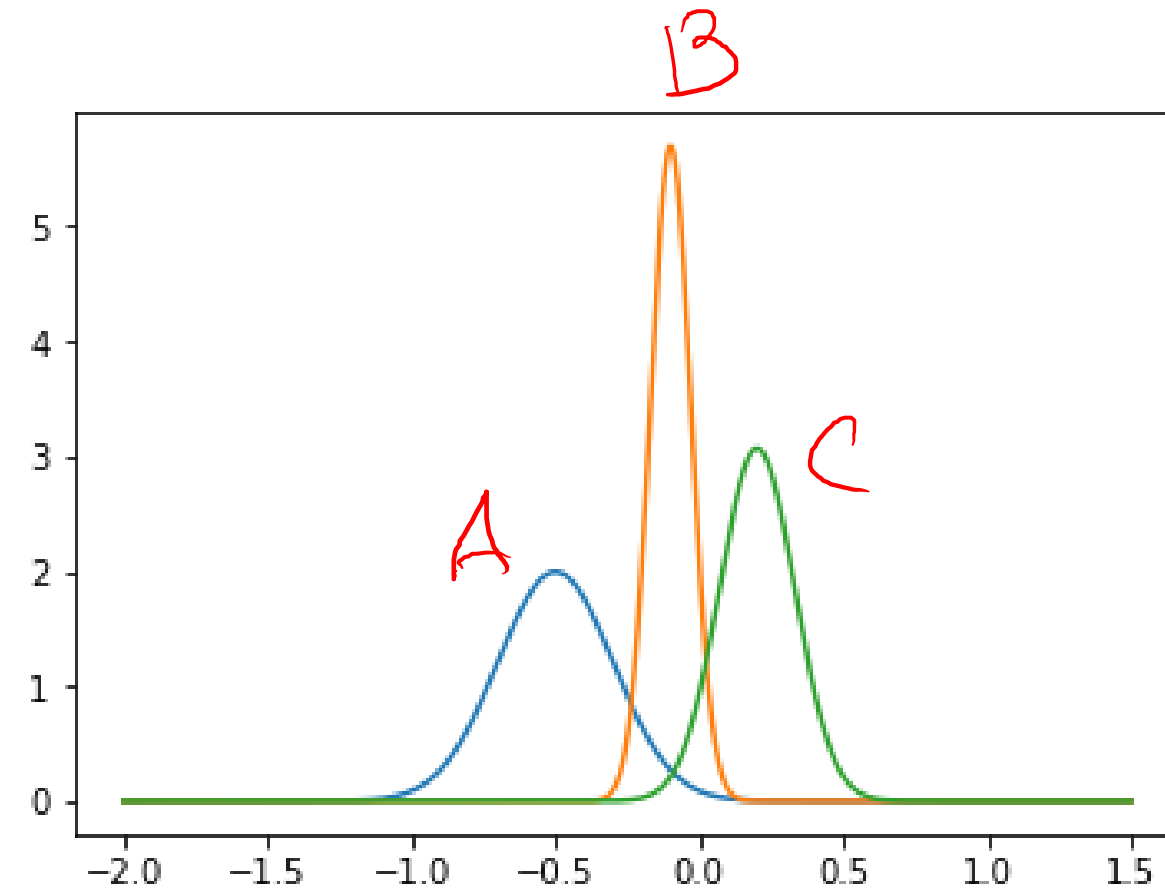
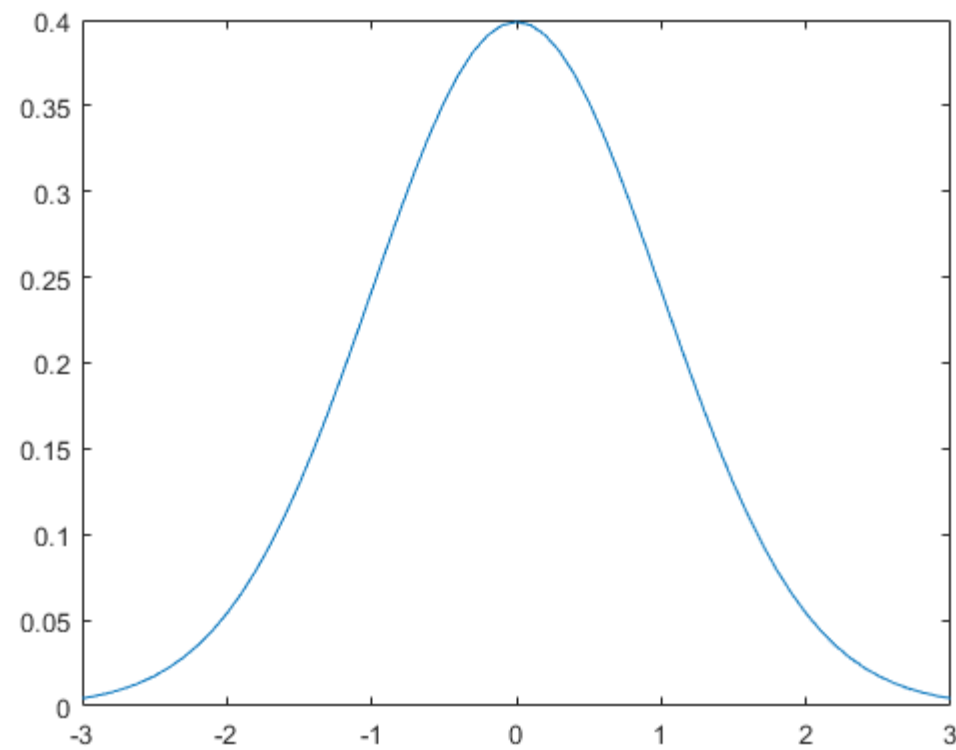
2-3. GMM과 EM 알고리즘

- K-means 등의 클러스터링 알고리즘으로 잘 묶을수 없었던 아래 데이터에서도 잘 작동함



Gaussian Mixture Model (GMM)

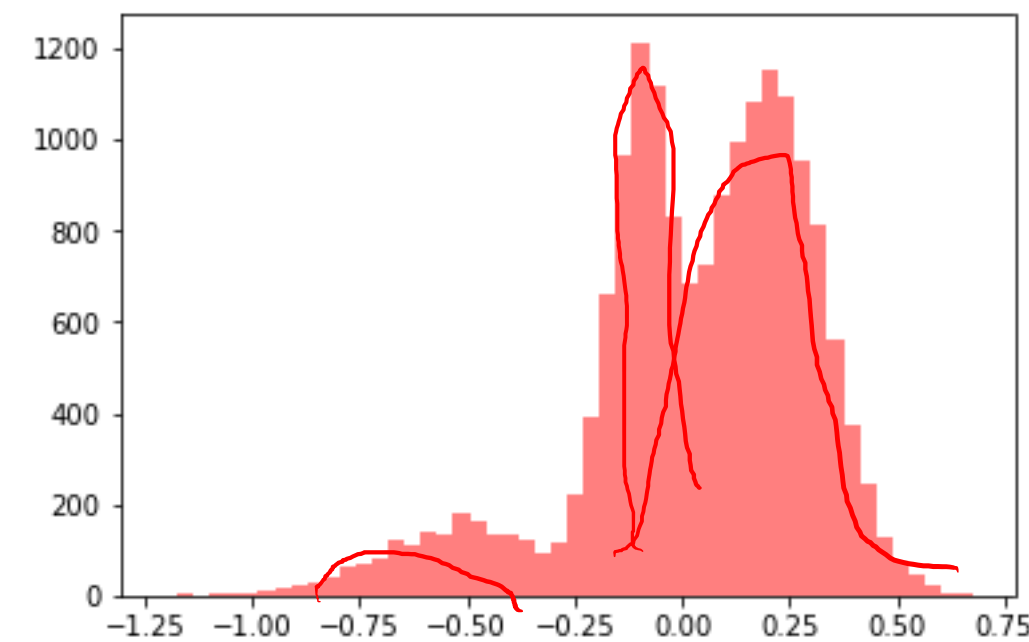
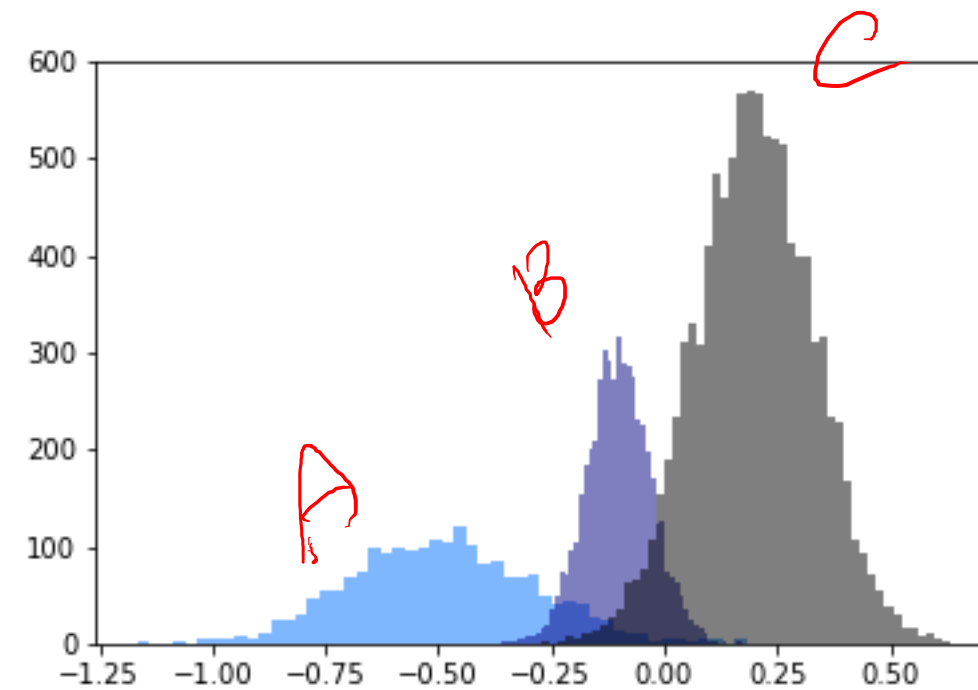
- 데이터들이 여러 개의 정규분포에서 나왔다고 가정
- 만약 A, B, C 라는 세 종류의 정규분포에서 데이터들이 생성되었다고 한다면?



Gaussian Mixture Model (GMM)

$$y = \frac{1}{\sigma} e^{-\frac{1}{2\sigma^2} (x - \mu)^2}$$

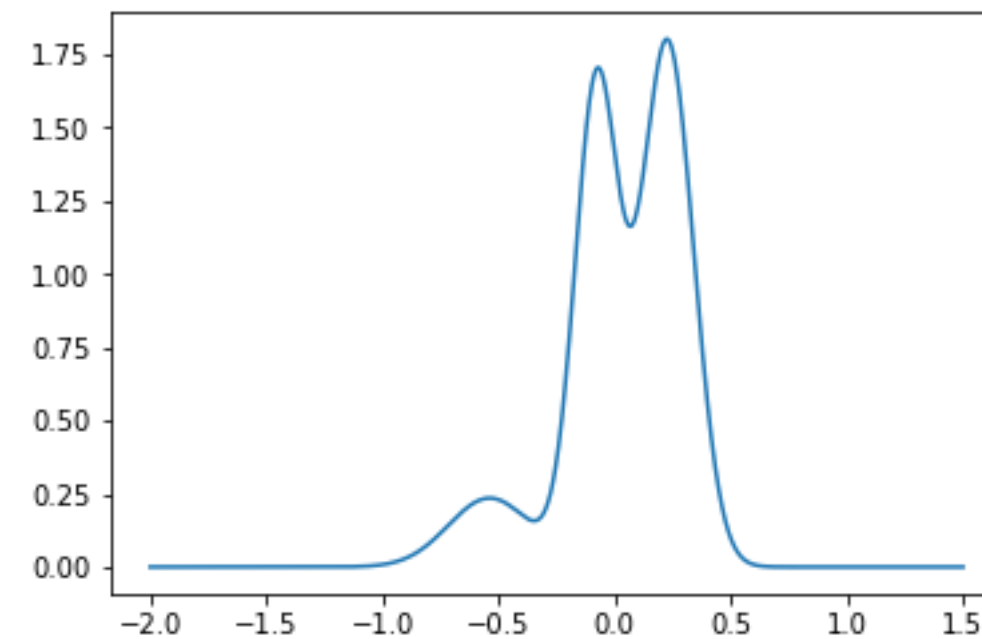
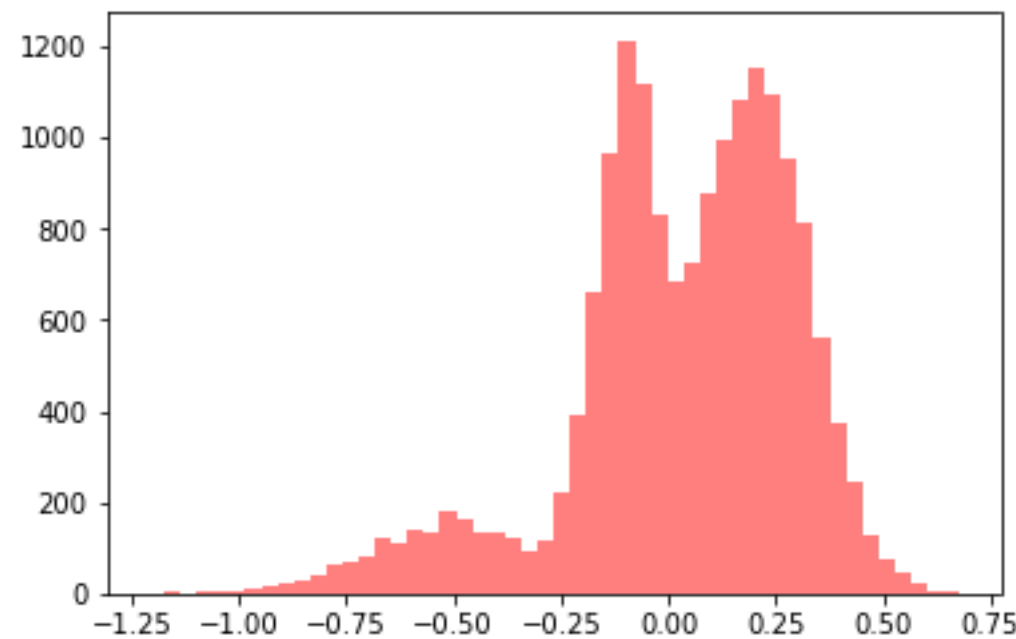
- 각 정규분포 A, B, C에서 2000개, 5000개, 10,000개 씩 임의로 뽑은 뒤 분포를 그려볼 수 있다 (왼쪽 그림, 색깔로 구분)
- 하지만 실제 우리가 보는 것은 오른쪽 그림
- 오른쪽 그림에 GMM을 적용하여 3개의 정규분포의 **평균**, **분산**, 임의의 데이터가 확률적으로 어디에 속해있는지(**Weight**) 를 구해야 함



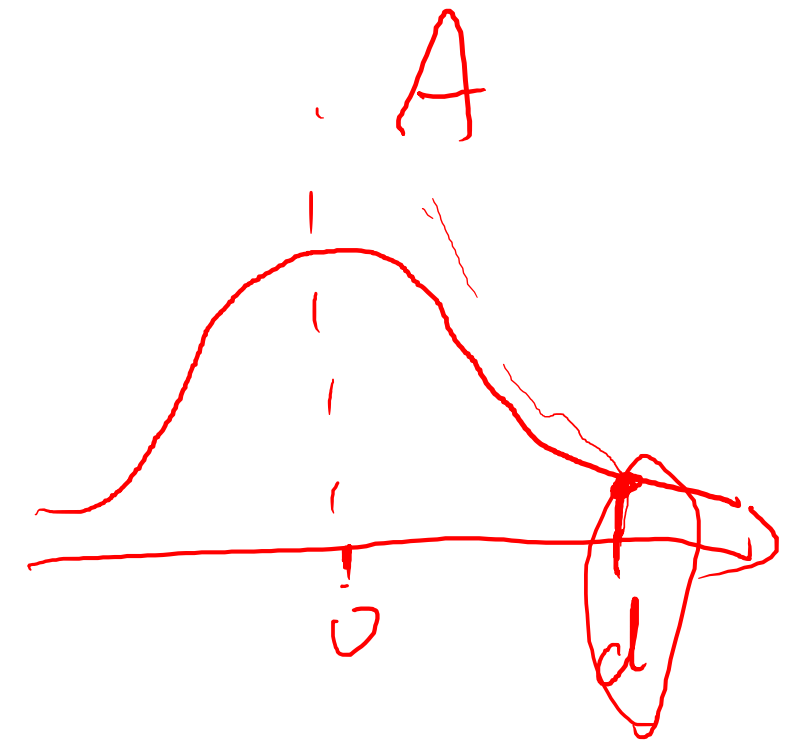
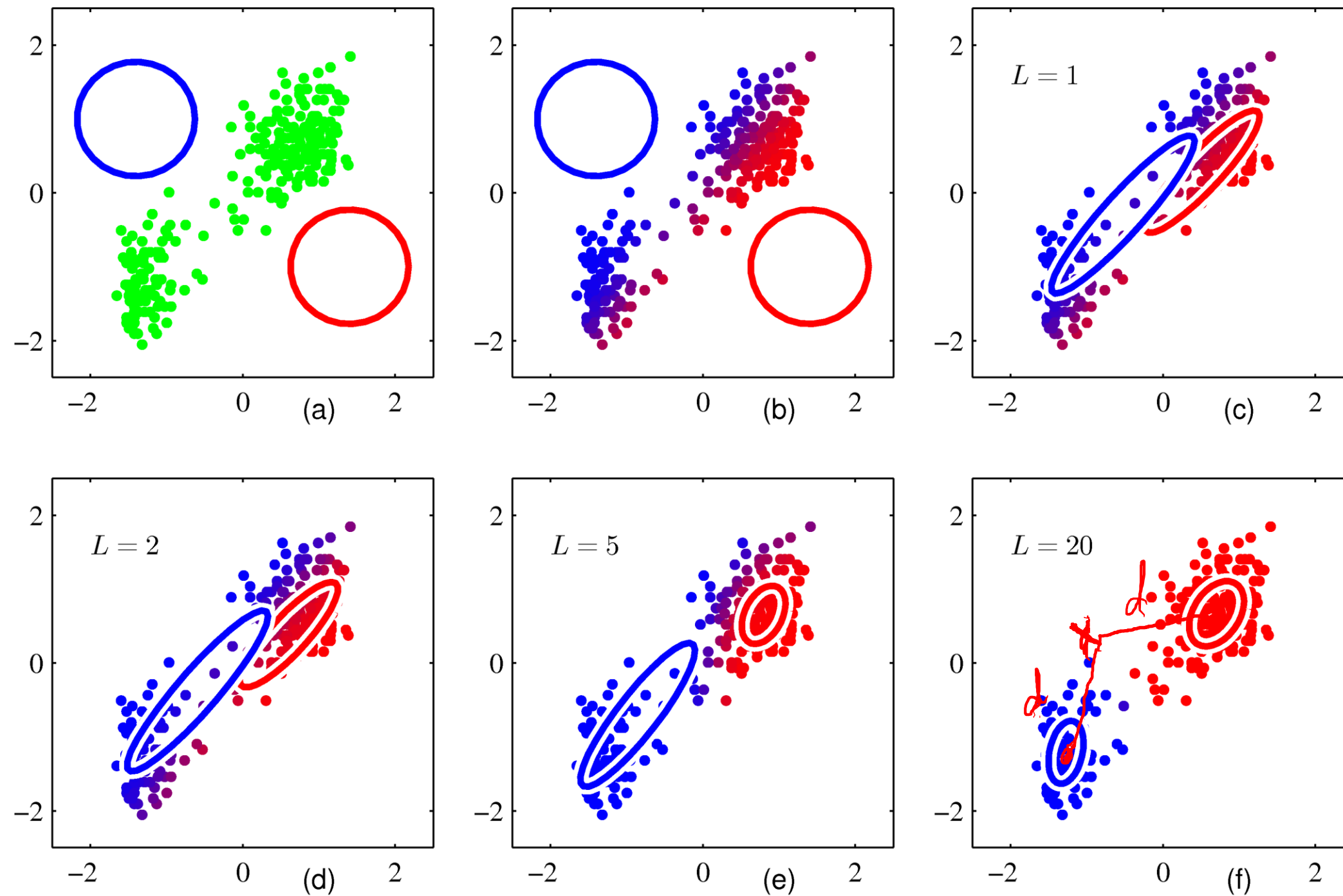
1 A?
2 B?
3 C?

Gaussian Mixture Model (GMM)

- 3개의 정규분포의 평균, 분산, 임의의 데이터가 확률적으로 어디에 속해있는지(Weight)에 대한 값들을 **모수**라고 부름
- Weight: 3개의 정규분포 중 데이터가 확률적으로 어디에 속해 있는가 (잠재변수)
- **Expectation Maximization (EM)** 알고리즘을 이용하여 구한다



Gaussian Mixture Model (GMM)



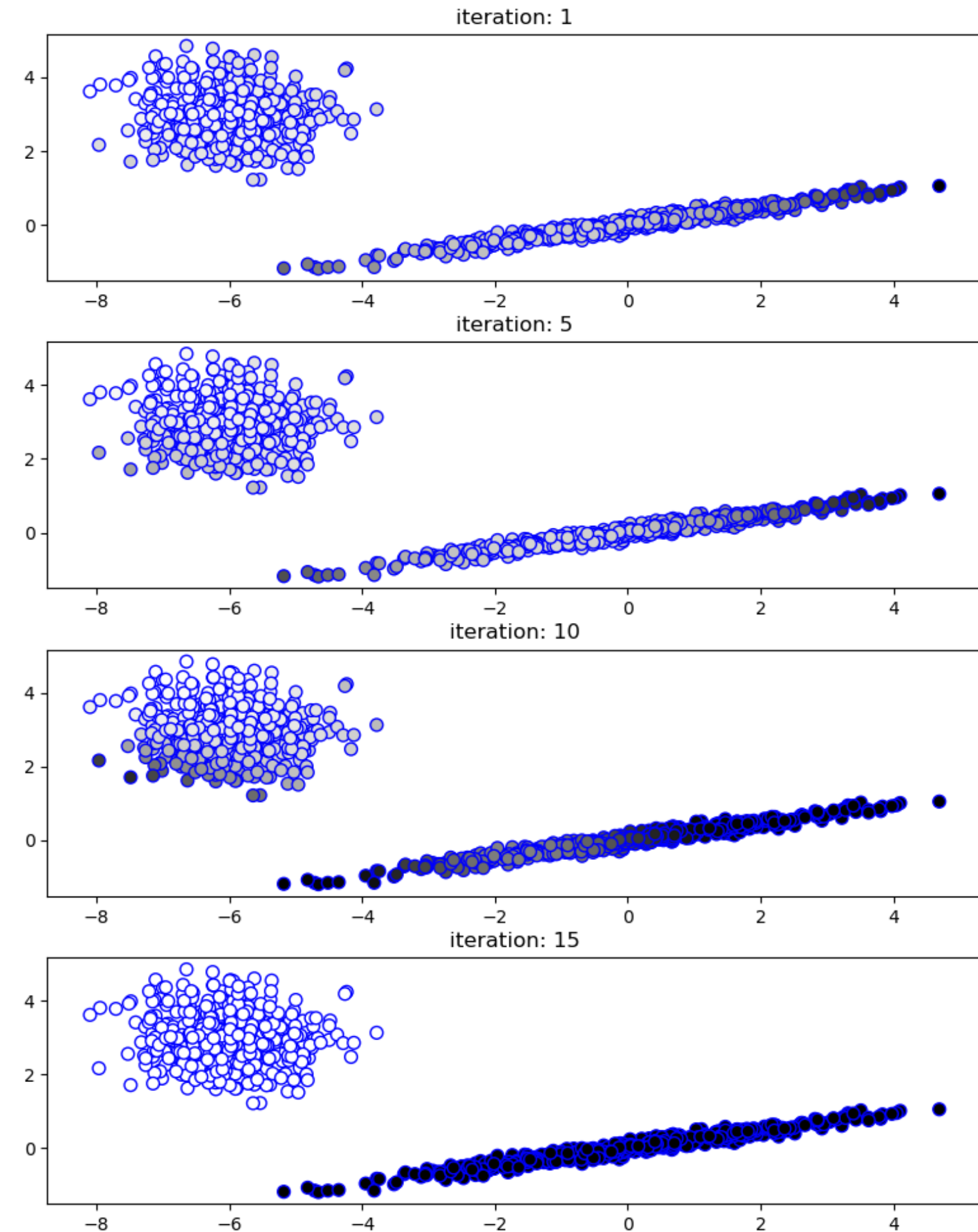
EM 알고리즘을 이용한 Clustering

Gaussian Mixture Model (GMM)

- 데이터를 어떻게 cluster 구분?
 - 어떤 Gaussian 분포에서 나온 데이터 인지 예측하자
 - EM 알고리즘을 통해 **모수** (평균, 분산)와 **responsibility** 를 예측
 - 데이터에 대해 responsibility 값이 가장 큰 카테고리를 찾아낸다!
- 데이터가 각 정규분포에서 가지는 확률 값이 가장 큰 분포에 할당

2-3. GMM과 EM 알고리즘

- iteration이 증가함에 따라
분류가 진행됨을 확인할 수 있음



2-3. GMM과 EM 알고리즘

- GMM의 parameter는 아래의 [알고리즘 1]과 같이

추정

Algorithm 1: EM algorithm for GMM

Input : a given data $X = \{x_1, x_2, \dots, x_n\}$

Output : $\pi = \{\pi_1, \pi_2, \dots, \pi_K\}$,

$\mu = \{\mu_1, \mu_2, \dots, \mu_K\}$,

$\Sigma = \{\Sigma_1, \Sigma_2, \dots, \Sigma_K\}$

1 Randomly initialize π, μ, Σ

2 **for** $t = 1 : T$ **do**

3 // E-step

4 **for** $n = 1 : N$ **do**

5 **for** $k = 1 : K$ **do**

6 $\gamma(z_{nk}) = \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x_n | \mu_j, \Sigma_j)}$

7 **end**

8 **end**

9 // M-step

10 **for** $k = 1 : K$ **do**

11 $\mu_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) x_n}{\sum_{n=1}^N \gamma(z_{nk})}$

12 $\Sigma_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) (x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^N \gamma(z_{nk})}$

13 $\pi_k = \frac{1}{N} \sum_{n=1}^N \gamma(z_{nk})$

14 **end**

15 **end**

2-3. GMM과 EM 알고리즘

- 데이터에 대한 분류는 [알고리즘 2]와 같이 수행됨

Algorithm 2: GMM classification

Input : a given data $X = \{x_1, x_2, \dots, x_n\}$,

$$\pi = \{\pi_1, \pi_2, \dots, \pi_K\},$$

$$\mu = \{\mu_1, \mu_2, \dots, \mu_K\},$$

$$\Sigma = \{\Sigma_1, \Sigma_2, \dots, \Sigma_K\}$$

Output: class labels $y = \{y_1, y_2, \dots, y_N\}$ for X

1 **for** $n = 1 : N$ **do**

2 $y_n = \arg \max_k \gamma(z_{nk})$

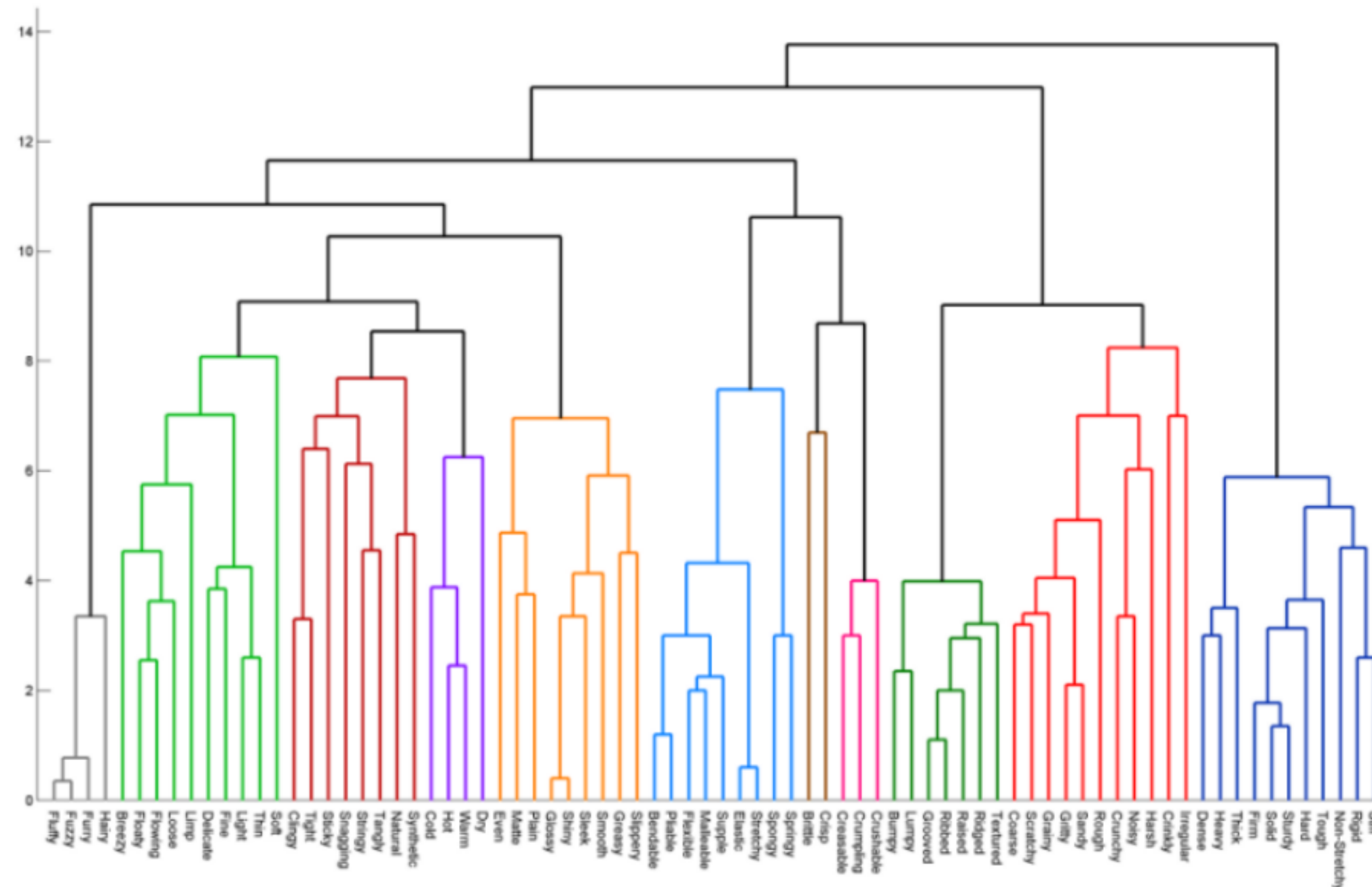
3 **end**

2-4. 계층적 클러스터링(HC)

- 계층적 트리 모형을 이용해 개별 개체들을 순차적, 계층적으로 유사한 개체 내지 그룹과 통합하여 군집화를 수행하는 알고리즘
- 클러스터의 수를 사전에 정하지 않아도 학습 수행가능

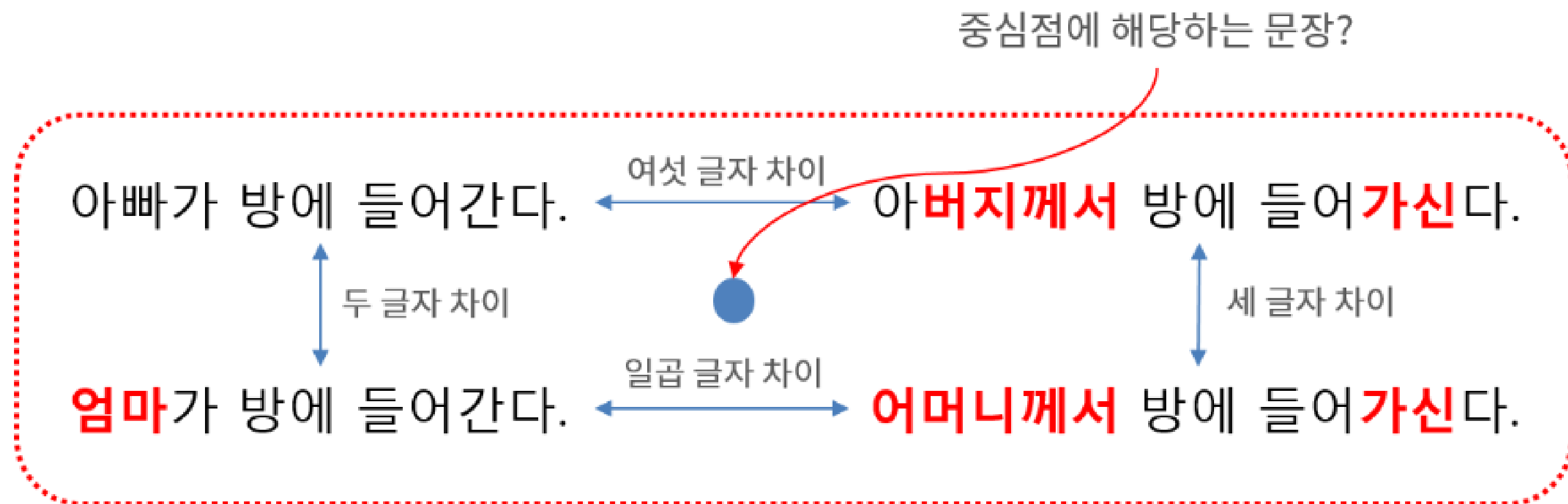
2-4. 계층적 클러스터링(HC)

- 개체들이 결합되는 순서를 나타내는 트리형태의 구조인 덴드로그램(Dendrogram)을 생성한 후 적절한 수준에서 트리를 자르면 클러스터링이 완료됨



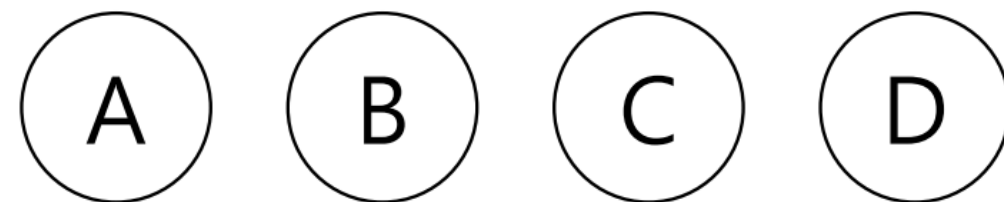
2-4. 계층적 클러스터링(HC)

- 데이터 각 쌍에 대해서는 유사도를 측정할 수 있지만 평균이나 분산 등을 구할 수 없는 경우에 사용할 수 있는 기법



2-4. 계층적 클러스터링(HC)

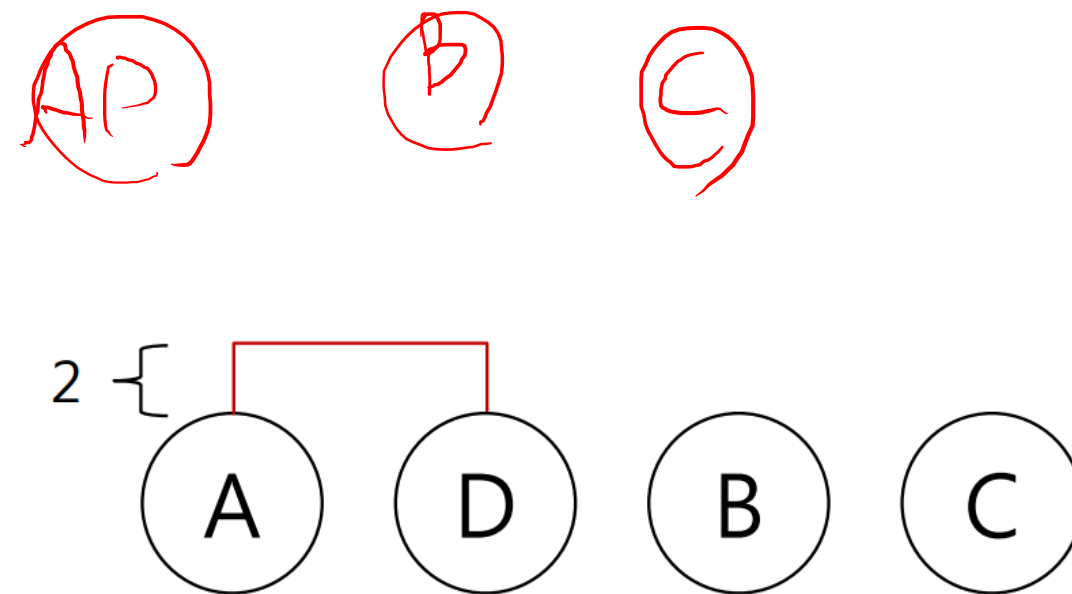
- HC를 수행하려면 모든 개체들 간 거리(distance)나 유사도(similarity)가 이미 계산되어 있어야 함
- 그림처럼 거리 행렬을 이미 구해놨다고 가정



	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

2-4. 계층적 클러스터링(HC)

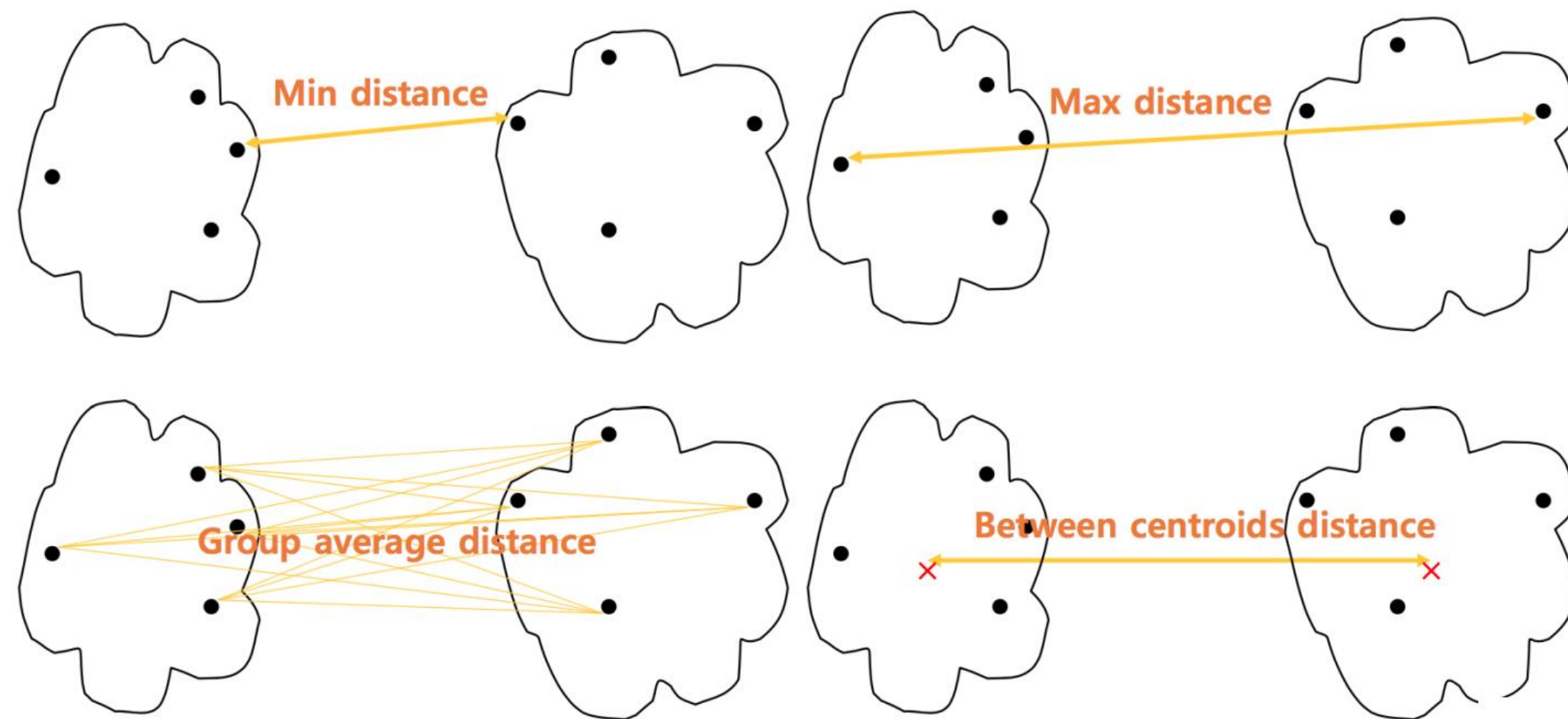
- 거리가 가장 짧은 것이 2이고 이에 해당하는 개체는 A와 D이므로 먼저 A와 D를 하나의 클러스터로 묶음
- 덴드로그램의 높이는 관측치간 거리(2)가 되도록 함



	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

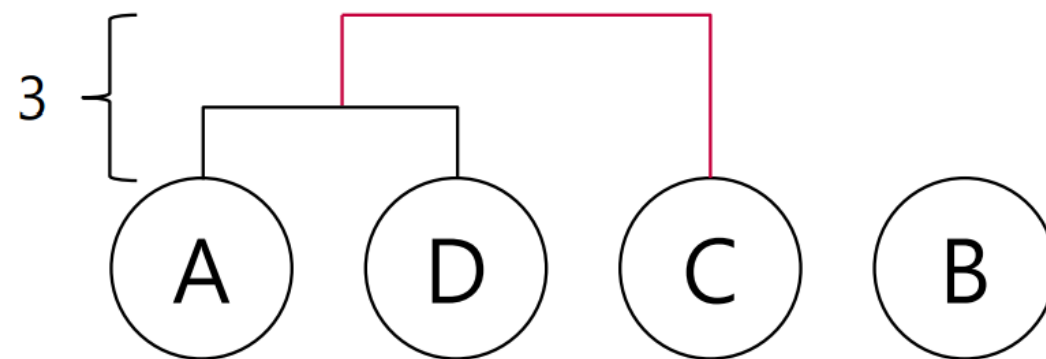
2-4. 계층적 클러스터링(HC)

- 여기서 A와 D를 한 군집으로 엮었으니 거리행렬을 바꿔주어야 함
즉 개체-개체 거리를 군집-개체 거리로 계산해야 함
- 군집-개체, 혹은 군집-군집 간 거리는 어떻게 계산?
아래 그림처럼 여러가지 선택지가 존재



2-4. 계층적 클러스터링(HC)

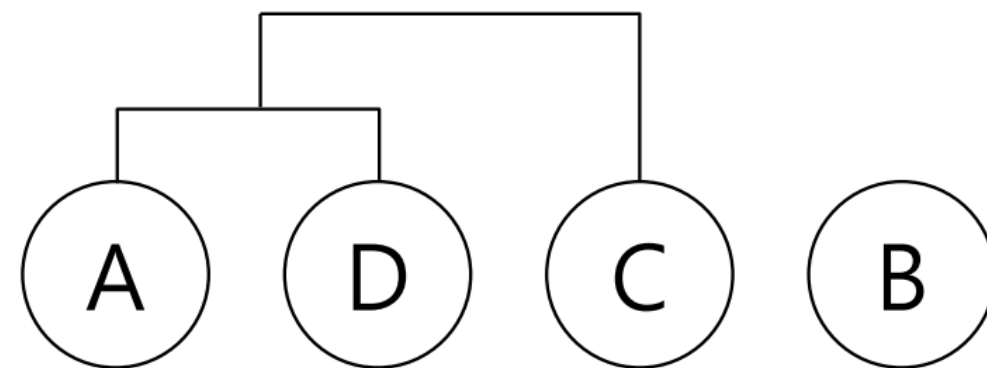
- 위 네 가지 방법 가운데 하나를 택했다고 가정
- 거리행렬을 업데이트했더니 AD와 C가 가장 인접



	AD	B	C	
AD		20	3	
B			10	
C				

2-4. 계층적 클러스터링(HC)

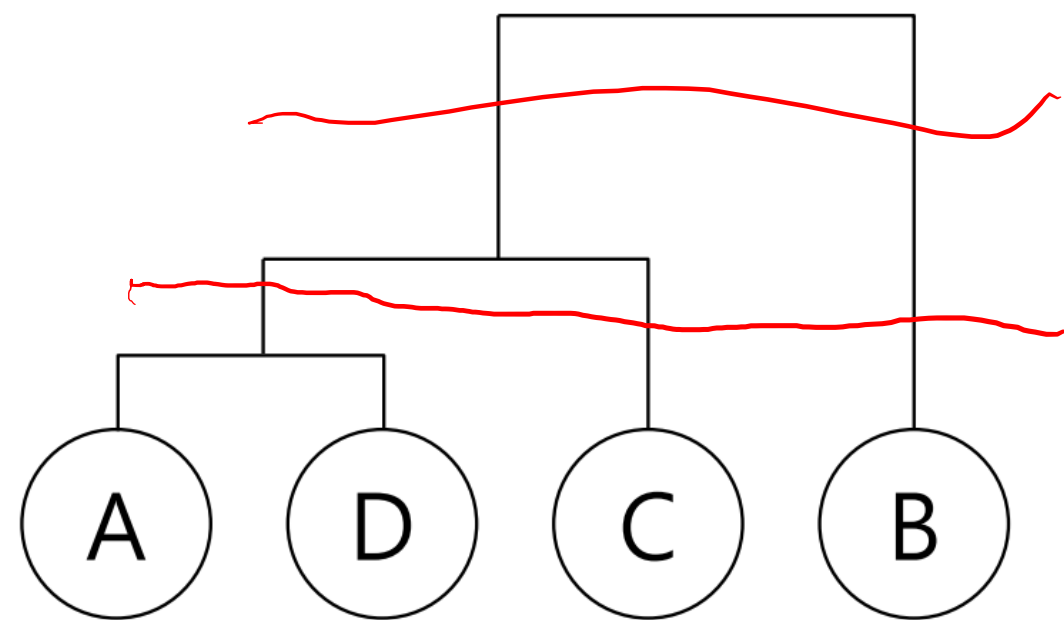
- 그 다음 과정을 수행



	ADC	B		
ADC		10		
B				

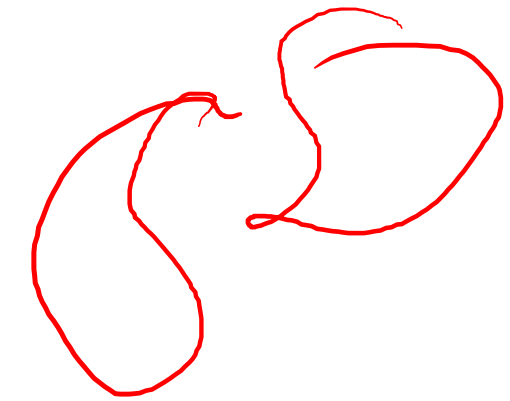
2-4. 계층적 클러스터링(HC)

- 분석 대상 관측치가 하나도 없으면 학습이 종료됨



	AD CB			
AD CB				

2-5. 알고리즘 비교



- K-means, GMM, 계층 클러스터링의 특성 및 장단점

	사용되는 유사도 측정 방식	하이퍼 파라미터	장점	단점 (한계점)
k 평균	좌표 기반 거리 구하기 (유클리드, 마할라노비스, 맨하탄)	<ul style="list-style-type: none">• 군집 개수• 종료 조건	<ul style="list-style-type: none">• 알고리즘이 쉽고 직관적임• 분산 시스템을 이용한 대용량 데이터 처리 가능• 각 유형의 특징 파악 용이	<ul style="list-style-type: none">• 아웃라이어에 민감• 유형별 데이터의 분산이 비슷하고 구형으로 분포되어 있지 않으면 결과가 좋지 못함
GMM	좌표 기반 거리 구하기 (유클리드, 마할라노비스, 맨하탄)	<ul style="list-style-type: none">• 군집 개수• 종료 조건	<ul style="list-style-type: none">• 확률 분포의 차이를 고려하여 군집을 묶는 방식이기 때문에 k 평균 알고리즘에 비해 좀 더 통계적으로 엄밀한 결과를 얻을 수 있음	<ul style="list-style-type: none">• 계산량이 많기 때문에 대량의 데이터에 사용하기 어려움• 유형들의 분포가 정규 분포와 차이가 크다면 결과가 좋지 못함
계층 클러스터링	모든 유사도 측정 방식 사용 가능	<ul style="list-style-type: none">• 군집 개수• 군집 간의 거리 측정 방법	<ul style="list-style-type: none">• 개체간의 거리는 구할 수는 있지만 군집의 평균을 구할 수 없는 데이터에 대해서도 적용 가능	<ul style="list-style-type: none">• 계산량이 많기 때문에 대량의 데이터에 사용하기 어려움

참고자료

1. 클러스터링

1. <https://ratsgo.github.io/machine%20learning/2017/04/16/clustering/>

2. <https://brunch.co.kr/@gimmesilver/40>

3. Andrew Ng 강의 동영상

2. K-means

<https://ratsgo.github.io/machine%20learning/2017/04/19/KC/>

3. KNN

<https://ratsgo.github.io/machine%20learning/2017/04/17/KNN/>

4. GMM

<https://untitledtblog.tistory.com/133>

<https://datascienceschool.net/view-notebook/c2934b8a2f9140e4b8364b266b1aa0d8/> - 코드

5. HC

`/* elice */`

문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

medium.com/elice