

# 인공지능개론

## 기계학습

# K-nearest neighbor

## K-근접이웃 (K-nearest neighbor, KNN) 알고리즘

(입력, 결과)가 있는 데이터들이 주어진 상황에서, 새로운 입력에 대한 결과를 추정할 때

결과를 아는 **최근접한 k개의 데이터**에 대한 결과정보를 이용하는 방법

- Linear regression, logistic regression
- Support vector machine

→ Model-based Learning

$$X_{new} \rightarrow f(X) \rightarrow Y_{new}$$

②

①

③

- K-nearest neighbor, Locally weighted regression

→ Instance-based learning

$$X_{new} \rightarrow X_S \rightarrow Y_{new}$$

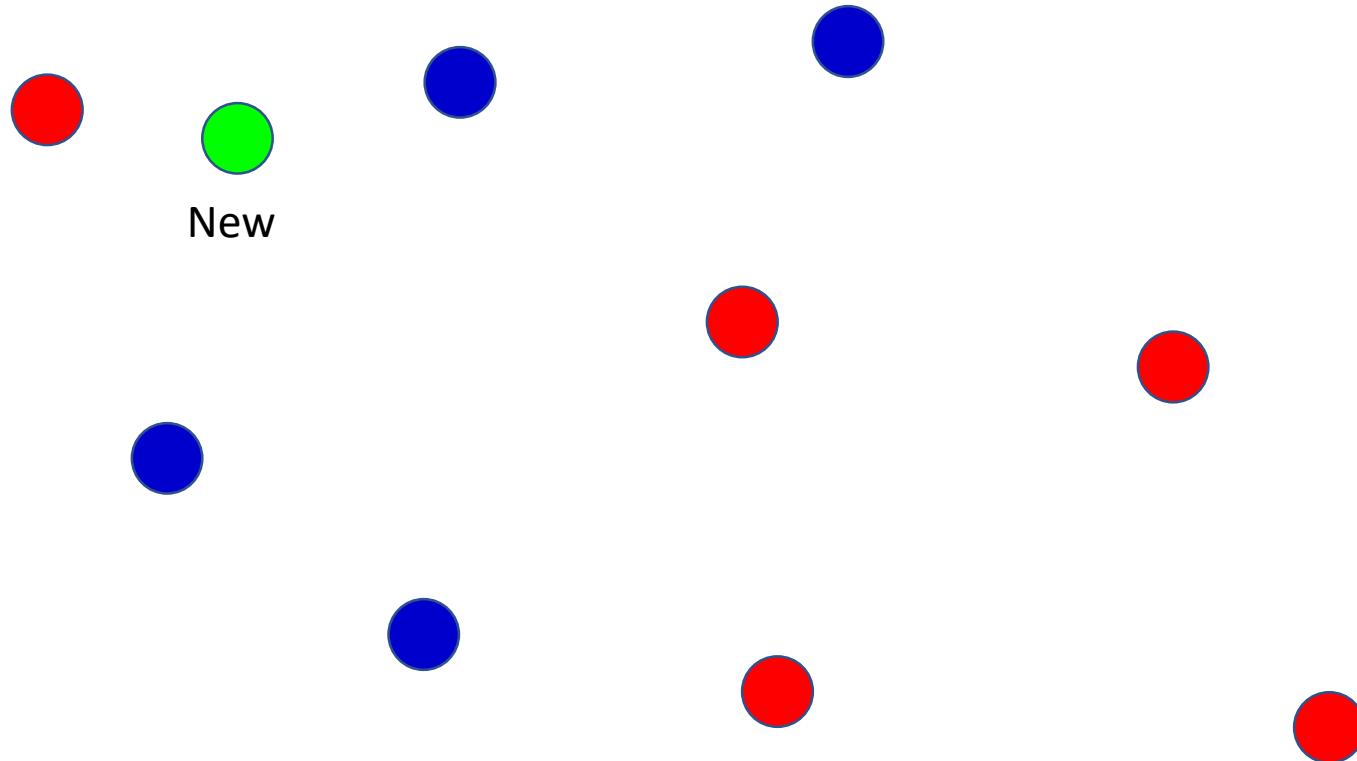
①

②

③

# *K-nearest neighbor*

1-nearest vs 3-nearest



# *K-nearest neighbor*

x1	x2	Class	distance	
3	3.5	1	1.5	1 <sup>st</sup> nearest
2	4.5	1	2.69	3 <sup>rd</sup> nearest
1	1	2	2.24	2 <sup>nd</sup> nearest
6	5.5	2	4.61	5 <sup>th</sup> nearest
4	5	1	3.16	4 <sup>th</sup> nearest

3	2	???
---	---	-----

# *K-nearest neighbor*

---

## - Instance-based Learning

→ 관측치 (instance)를 이용한 예측

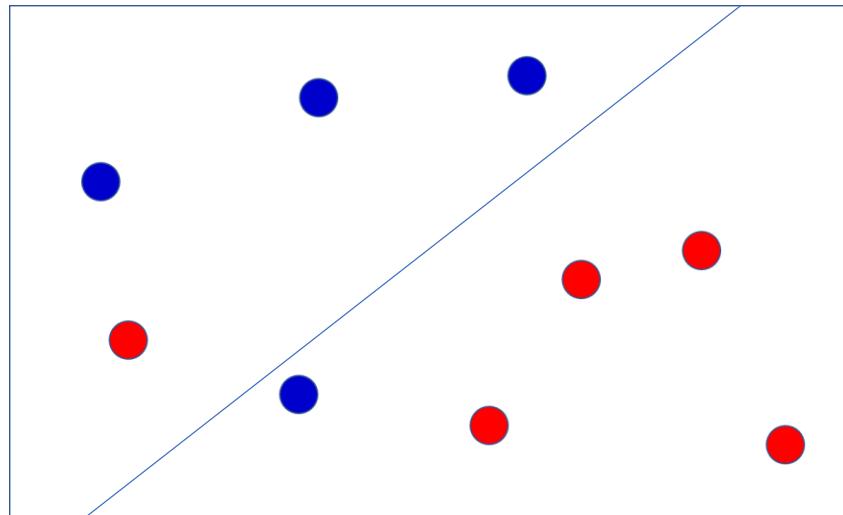
## - Memory-based Learning

→ 모든 학습데이터를 메모리에 저장

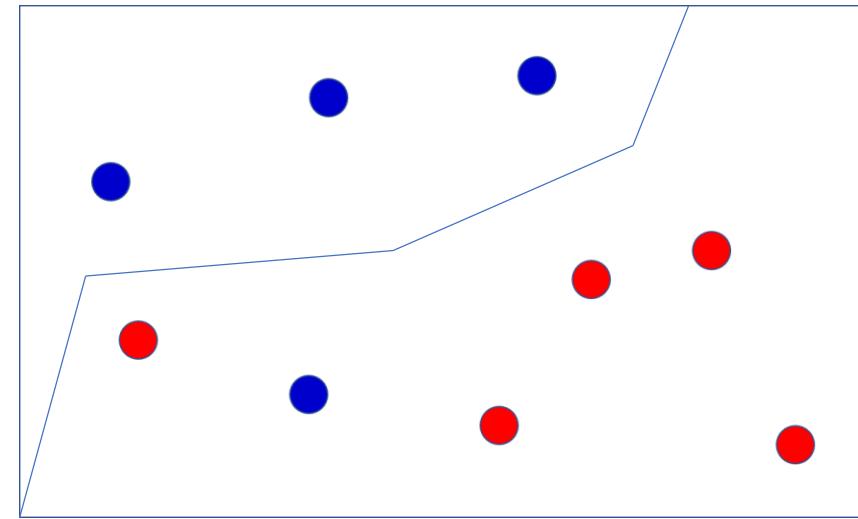
## - Lazy Learning

→ 학습단계에서 실질적 학습 없이 데이터만 저장  
→ 학습데이터가 많을 시 메모리 문제

# *K-nearest neighbor : classification*



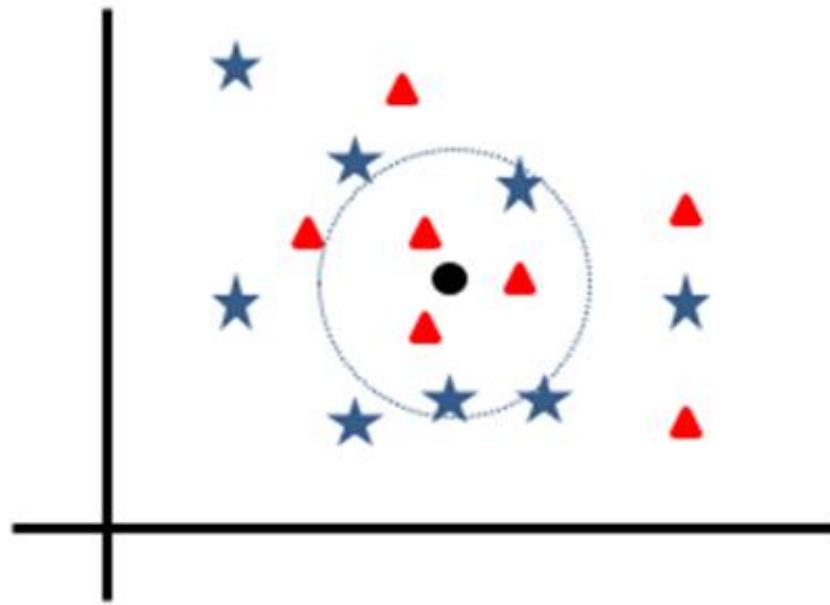
Linear boundary



KNN (k=4) boundary

# *K-nearest neighbor : classification*

## KNN 분류기법



- 출력이 범주형 값
- 다수결 투표(majority voting) : 개수가 많은 범주 선택

# K-nearest neighbor : classification

유전자 정보					환자 상태	새로운 관측치와의 거리
사람	유전자 A	유전자 B	유전자 C	유전자 D		
A	2.54	4.33	3.99	2.57	정상	1.54
B	3.12	3.87	3.84	3.04	정상	0.76
C	2.76	4.17	5.63	3.28	정상	2.00
D	3.87	3.56	4.25	3.65	질병	0.78
E	3.55	3.91	2.68	4.22	질병	1.28
F	4.12	2.86	3.30	3.71	질병	1.31

G	3.24	3.68	3.82	3.77	?	k = 1 k = 3
---	------	------	------	------	---	----------------

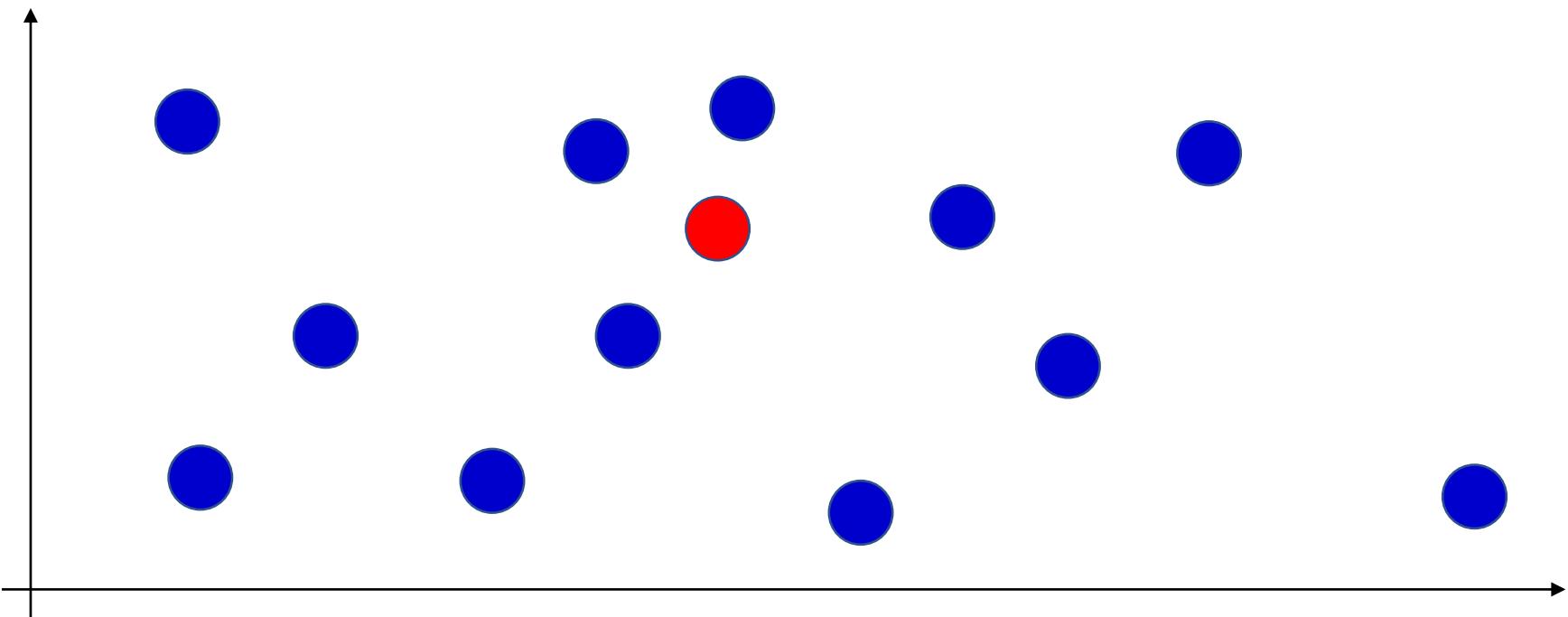
# *K-nearest neighbor : classification*

---

KNN 분류 알고리즘

- ① 분류하고자 하는 관측치  $x$  선택
- ②  $x$ 와 인접한 순서대로  $k$ 개의 데이터 탐색
- ③ 탐색된 학습 데이터  $k$ 개로 부터 majority class  $Y$  정의
- ④  $Y$ 를  $x$ 의 분류 값으로 반환

# *K-nearest neighbor : prediction*



$k$  = number of nearest neighbors

# *K-nearest neighbor : prediction*

기존 선호도						선호도	
	LOL	FIFA	Star Craft	Lineage	Kart Rider	Battle Ground	Distance
A	7.5	7.5	7	9.5	8.5	5.0	3.28
B	7.5	7.0	7.5	8.0	8.0	6.0	2.40
C	8.0	7.0	8.0	8.0	8.5	8.5	2.12
D	8.5	8.0	9.5	7.5	6.0	7.0	2.65
E	10.0	9.5	9.0	7.5	7.5	10.0	1.87
F	9.0	9.0	8.0	8.0	8.0	9.0	1.12
G	9.0	8.5	8.0	7.0	8.0		

# *K-nearest neighbor : prediction*

---

KNN 예측 알고리즘

- ① 예측하고자 하는 관측치  $x$  선택
- ②  $x$ 와 인접한 순서대로  $k$ 개의 데이터 탐색
- ③ 탐색된 학습 데이터  $k$ 개로 부터 평균을  $x$ 의 예측값으로 반환

# *K-nearest neighbor : parameters*

---

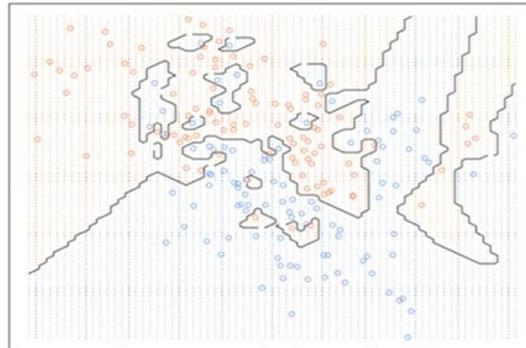
- **K**

→ 인접 학습 데이터를 몇 개까지 고려할 것인지?

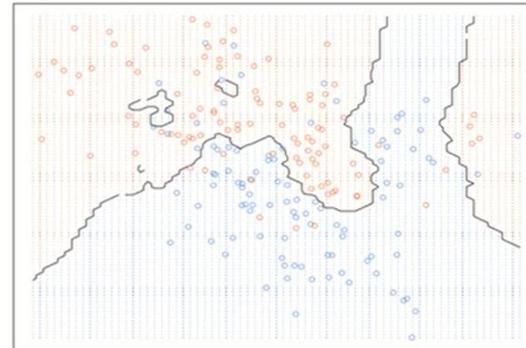
- **Distance**

→ 데이터 간 거리는 어떻게 정의하고 측정할 것인지?

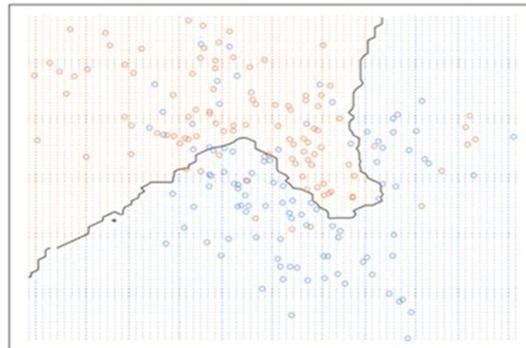
# *K-nearest neighbor : parameters*



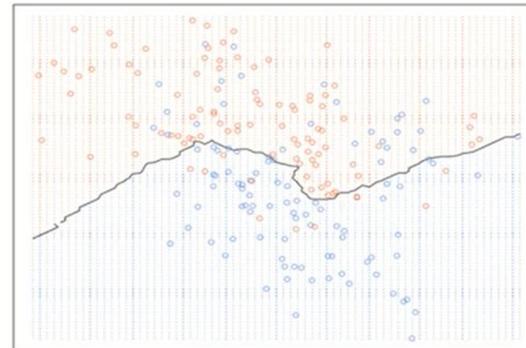
1-nearest neighbor



5-nearest neighbor



15-nearest neighbor



50-nearest neighbor

$K$ 가 작을 경우 : overfitting

$K$ 가 클 경우 : underfitting

# *K-nearest neighbor : parameters*

---

How to choose K?

- Classification

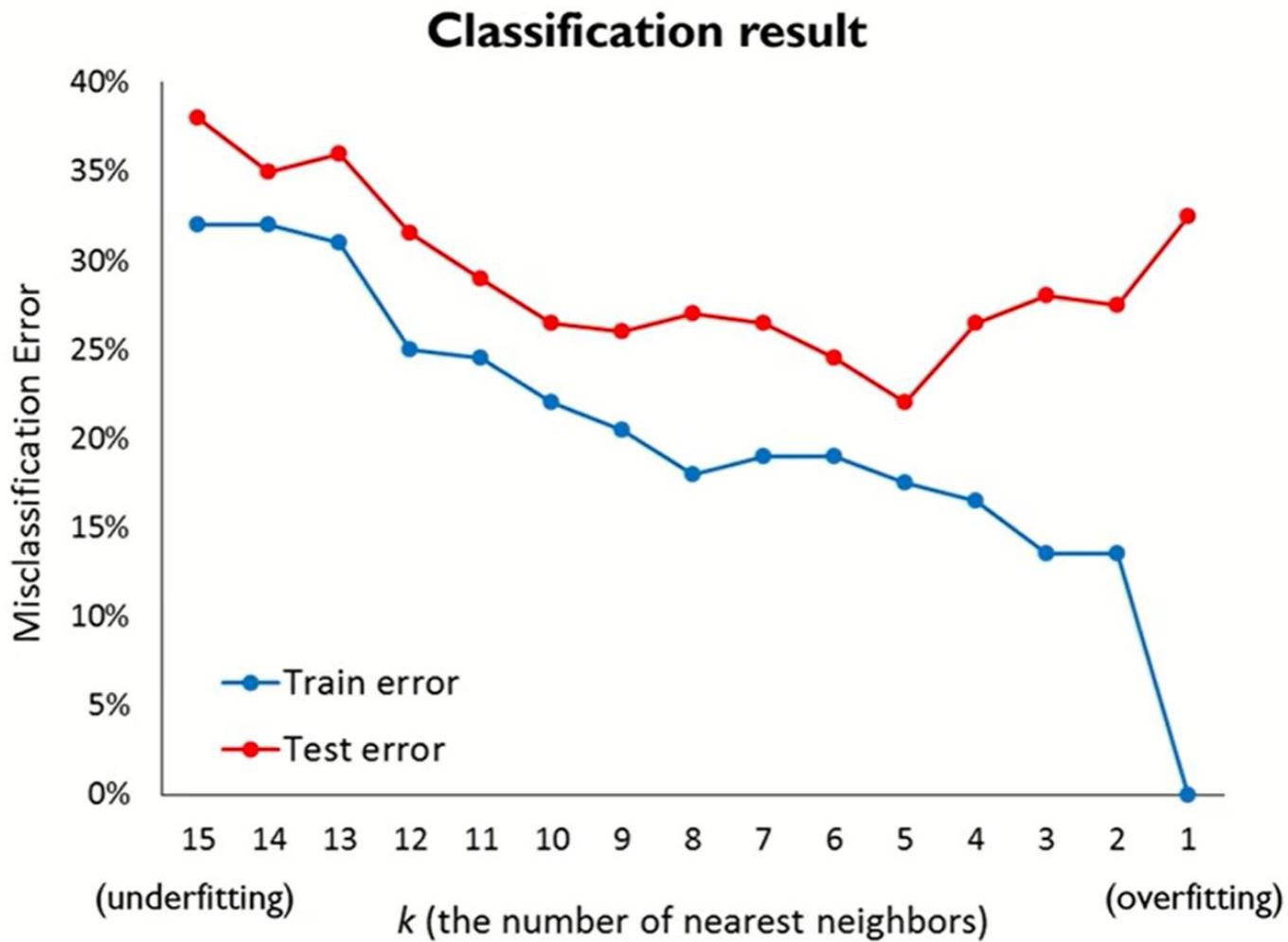
$$MisclassError_k = \frac{1}{k} \sum_{i=1}^k I(c_i \neq \hat{c}_i) \text{ for } k = 1, 2, \dots, k^*$$

$I(\cdot)$ : Indicator Function

- Prediction

$$SSE_k = \sum_{i=1}^k (y_i - \hat{y}_i)^2 \text{ for } k = 1, 2, \dots, k^*$$

# *K-nearest neighbor : parameters*



# *K-nearest neighbor : parameters*

---

## - 거리 측도 방법

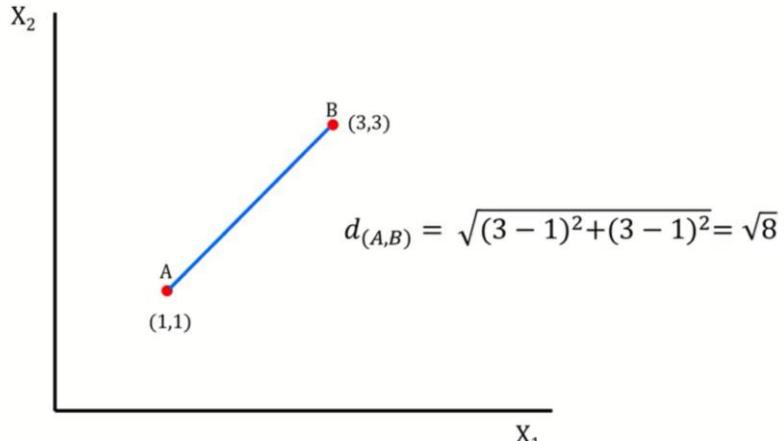
- Euclidean Distance
- Manhattan Distance
- Mahalanobis Distance
- Correlation Distance

## - 정규화 or 표준화

- 특정 변수들이 거리를 결정하는 것을 방지

# *K-nearest neighbor : Euclidean distance*

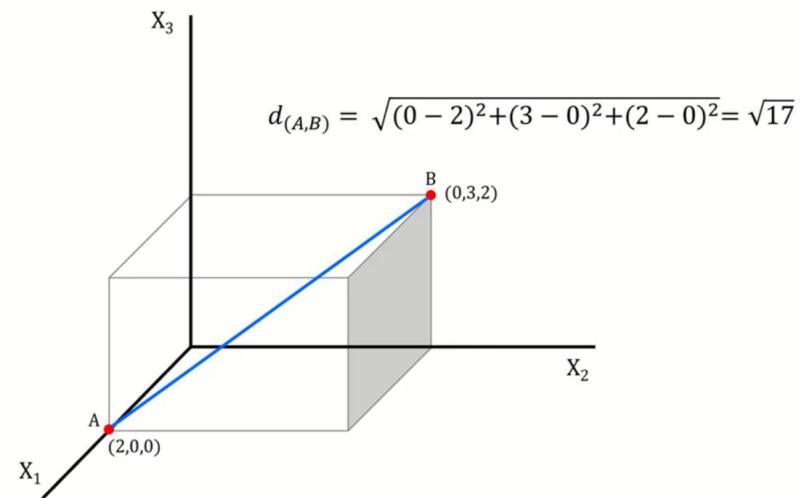
$$d_{(X,Y)} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$



$$A = (a_1, a_2, \dots, a_p)$$

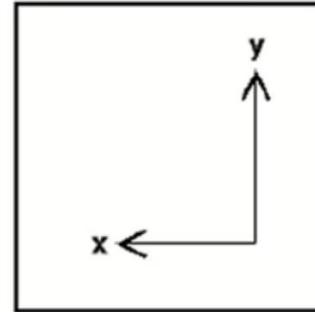
$$B = (b_1, b_2, \dots, b_p)$$

$$d_{(A,B)} = \sqrt{(a_1 - b_1)^2 + \dots + (a_p - b_p)^2} = \sqrt{\sum_{i=1}^p (a_i - b_i)^2}$$

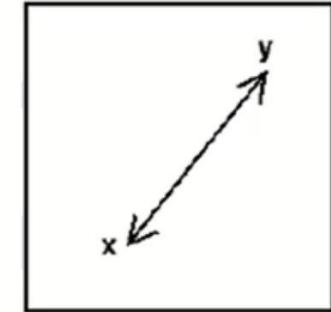


# *K-nearest neighbor : Manhattan distance*

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



Manhattan



Euclidean



# *K-nearest neighbor : Mahalanobis distance*

---

$$d_{Mahalanobis}(X, Y) = \sqrt{(X - Y)^T \Sigma^{-1} (X - Y)},$$

$\Sigma^{-1}$ : inverse of covariance matrix

- 변수들의 공분산을 고려하여 거리를 계산하는 방식
- Covariance matrix가 identity matrix이면 Euclidean distance

# *K-nearest neighbor : Mahalanobis distance*

---

$$\sqrt{(X - Y)^T \Sigma^{-1} (X - Y)} = c \text{ (c is Mahalanobis distance)}$$

$$\rightarrow (X - Y)^T \Sigma^{-1} (X - Y) = c^2$$

Let  $X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ ,  $Y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$ ,  $\Sigma^{-1} = \begin{pmatrix} s_{11}^{-1} & s_{12}^{-1} \\ s_{21}^{-1} & s_{22}^{-1} \end{pmatrix}$ , then

$$\rightarrow (x_1 - y_1)^2 s_{11}^{-1} + 2(x_1 - y_1)(x_2 - y_2)s_{12}^{-1} + (x_2 - y_2)^2 s_{22}^{-1} = c^2 \quad (\because s_{12}^{-1} = s_{21}^{-1})$$

Let  $Y = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ , then

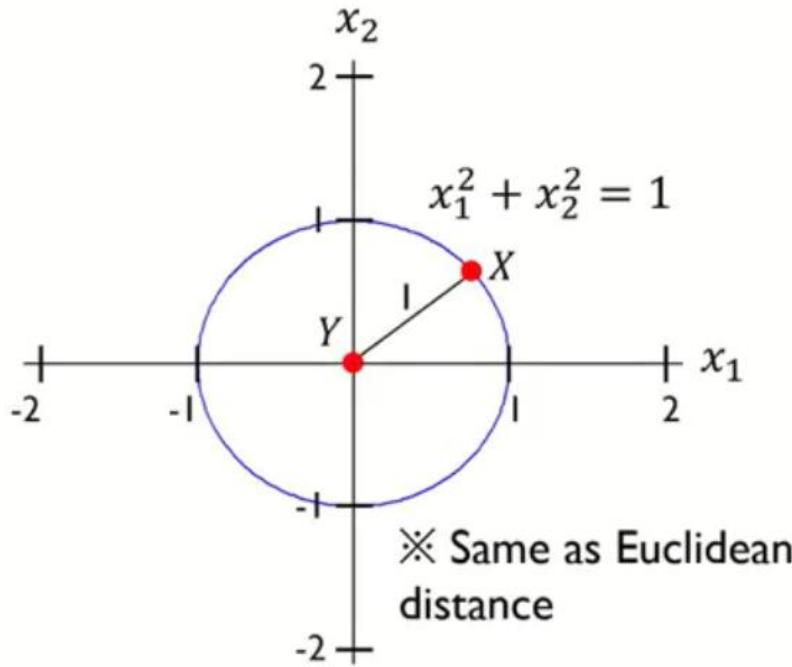
$$\rightarrow x_1^2 s_{11}^{-1} + 2x_1 x_2 s_{12}^{-1} + x_2^2 s_{22}^{-1} = c^2$$

which is a general equation of the ellipse.

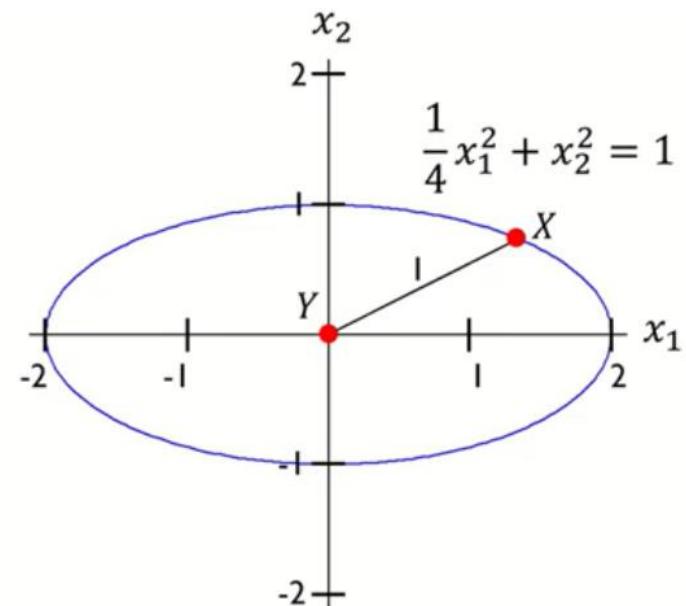
# *K-nearest neighbor : Mahalanobis distance*

$$\Sigma = \Sigma^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

(identity matrix)



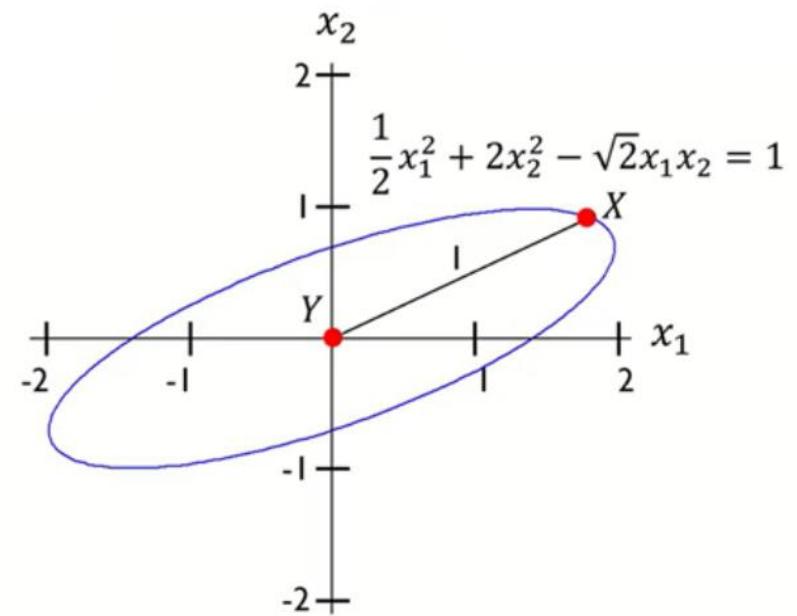
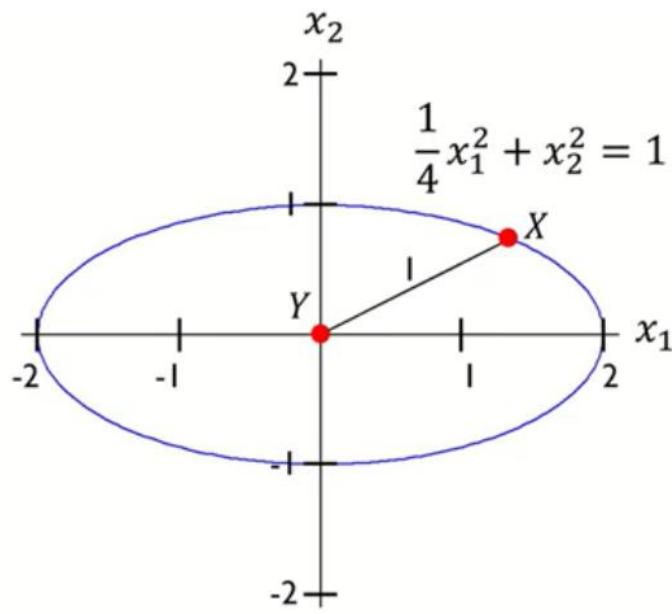
$$\Sigma = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}, \Sigma^{-1} = \begin{pmatrix} 1/4 & 0 \\ 0 & 1 \end{pmatrix}$$



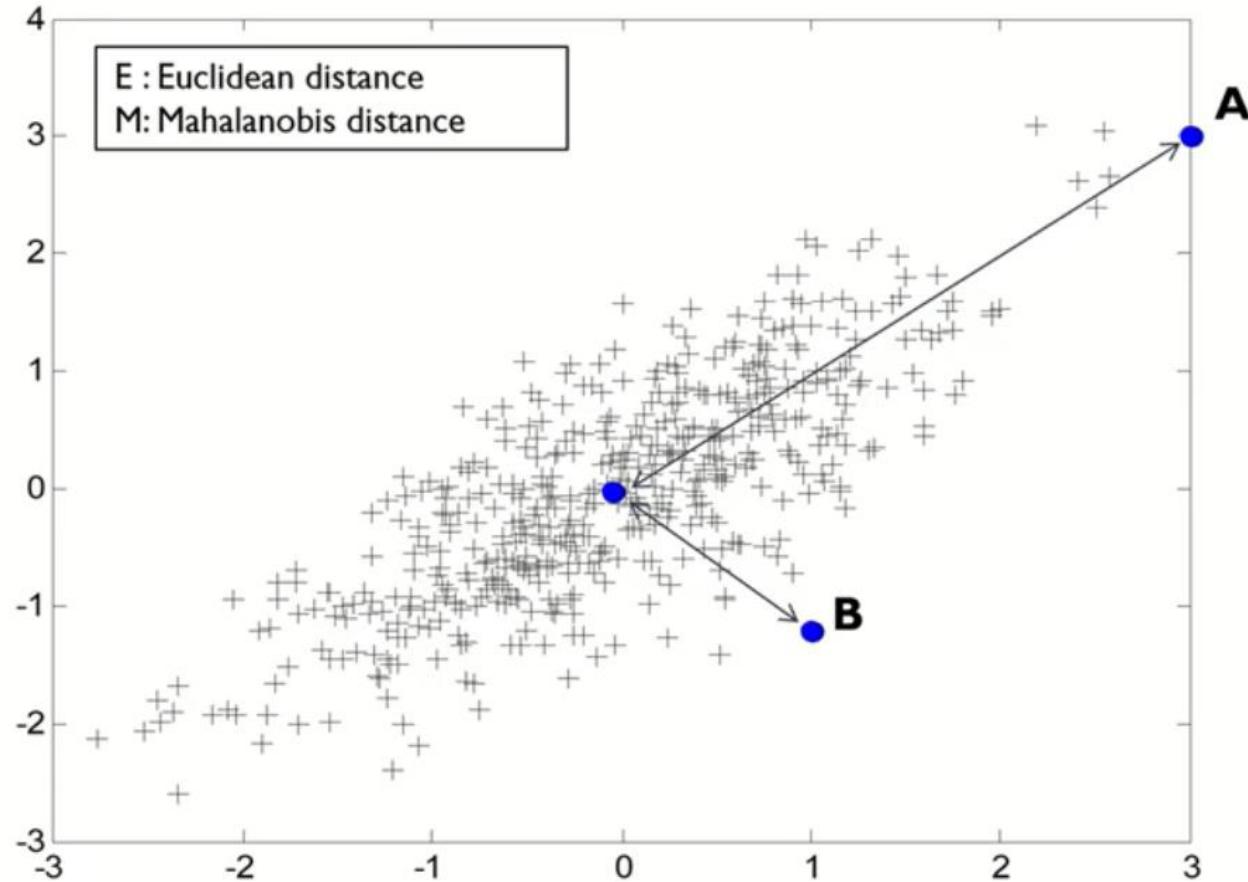
# *K-nearest neighbor : Mahalanobis distance*

$$\Sigma = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}, \Sigma^{-1} = \begin{pmatrix} 1/4 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 4 & \sqrt{2} \\ \sqrt{2} & 1 \end{pmatrix}, \Sigma^{-1} = \begin{pmatrix} 1/2 & -\sqrt{1/2} \\ -\sqrt{1/2} & 2 \end{pmatrix}$$



# *K-nearest neighbor : Mahalanobis distance*

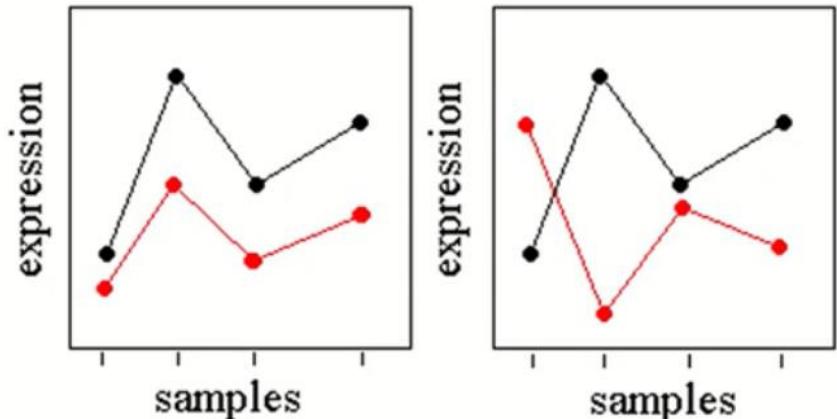


# *K-nearest neighbor : Correlation distance*

$$d_{Corr(X,Y)} = 1 - r$$

where  $r = \sigma_{XY}$

$$-1 \leq r \leq 1$$



- Pearson correlation을 거리측도로 사용, 데이터 패턴의 유사도 반영 가능

$$d_{Spearman(X,Y)} = 1 - \rho,$$

$$\text{where } \rho = 1 - \frac{6 \sum_{i=1}^n (rank(x_i) - rank(y_i))^2}{n(n^2 - 1)}$$

- 데이터의 rank를 이용하여 계산

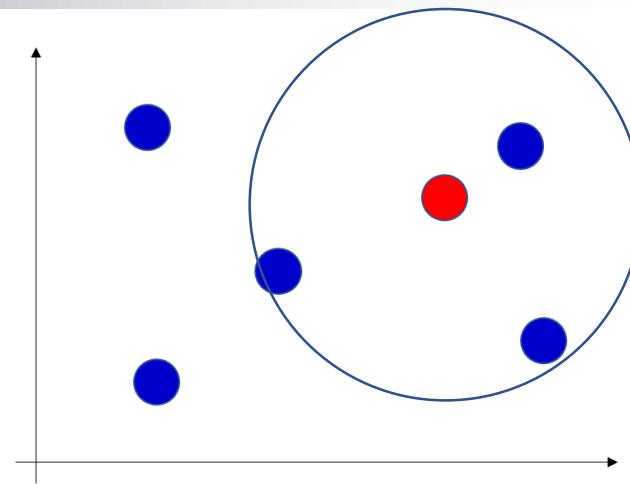
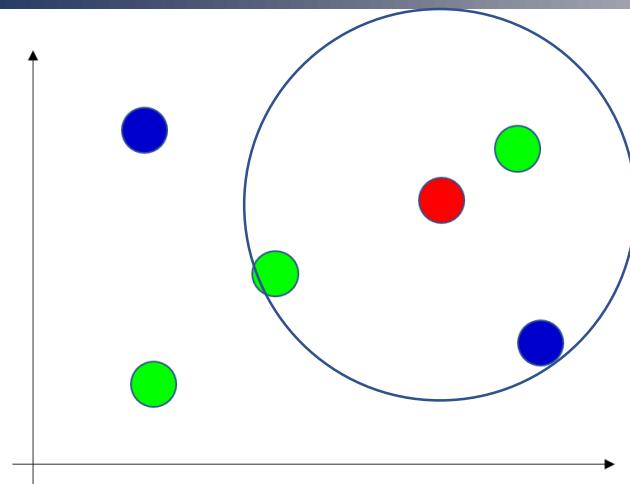
계절 평균 낮 최고 기온

지역	봄	여름	가을	겨울
서울	17.06	28.43	19.07	3.50
뉴욕	16.32	28.22	18.37	5.43
시드니	22.23	17.03	21.90	25.63

지역 별 계절 기온 순위

지역	봄	여름	가을	겨울
서울	3	1	2	4
뉴욕	3	1	2	4
시드니	2	4	3	1

# *K-nearest neighbor : Weighted KNN*



## - Classification

$$\hat{y}_{new} = \frac{\sum_{i=1}^k w_i y_i}{\sum_{i=1}^k w_i} \quad \text{where } w_i = \frac{1}{d_{(new, x_i)}^2}$$

## - Prediction

$$\hat{c}_{new} = \max_c \sum_{i=1}^k w_i I(w_i \in c)$$

# *K-means clustering*

---

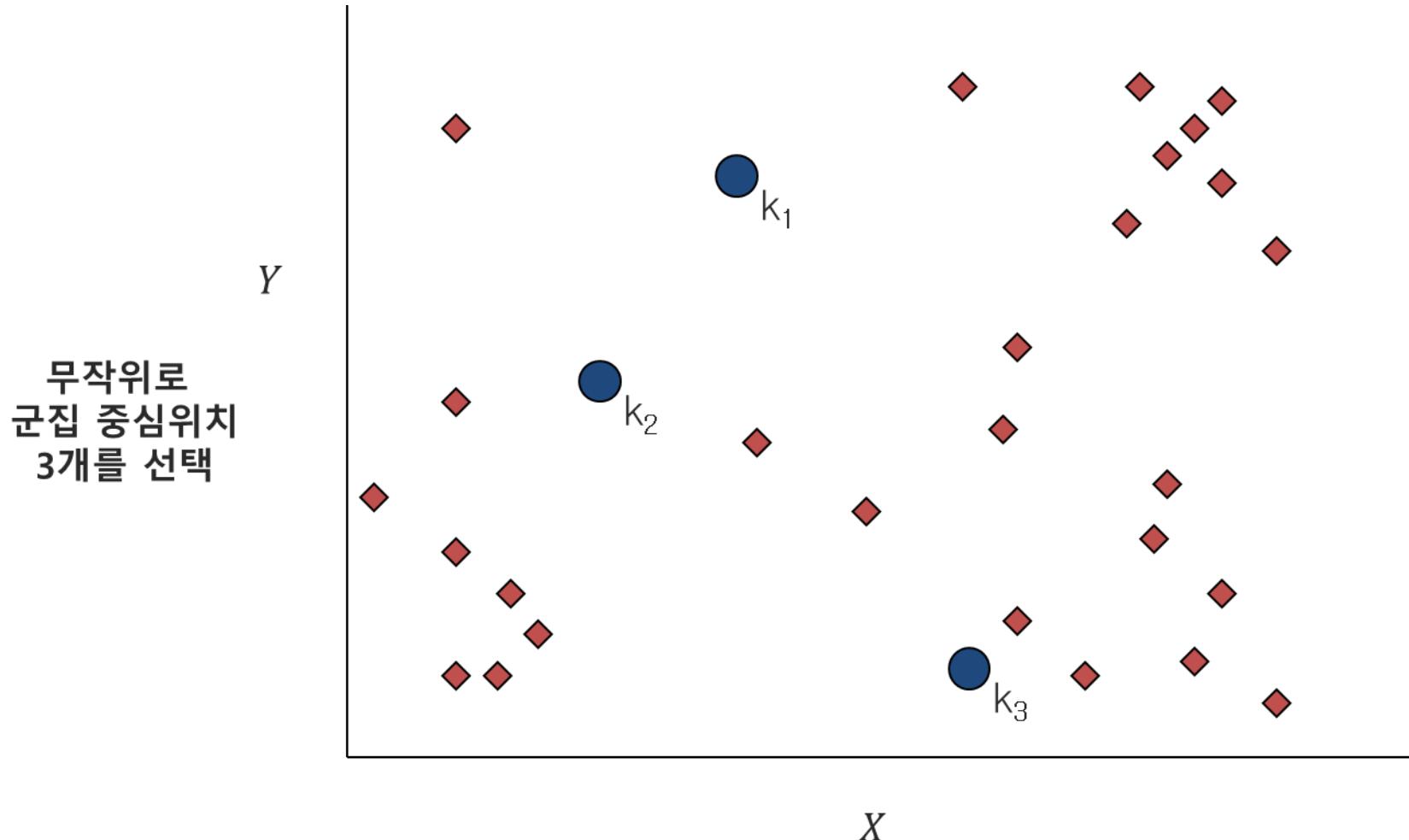
- 군집화(clustering) 알고리즘

- 데이터를 유사한 것들끼리 모우는 것
- 군집 간의 유사도(similarity)는 크게, 군집 내의 유사도는 작게

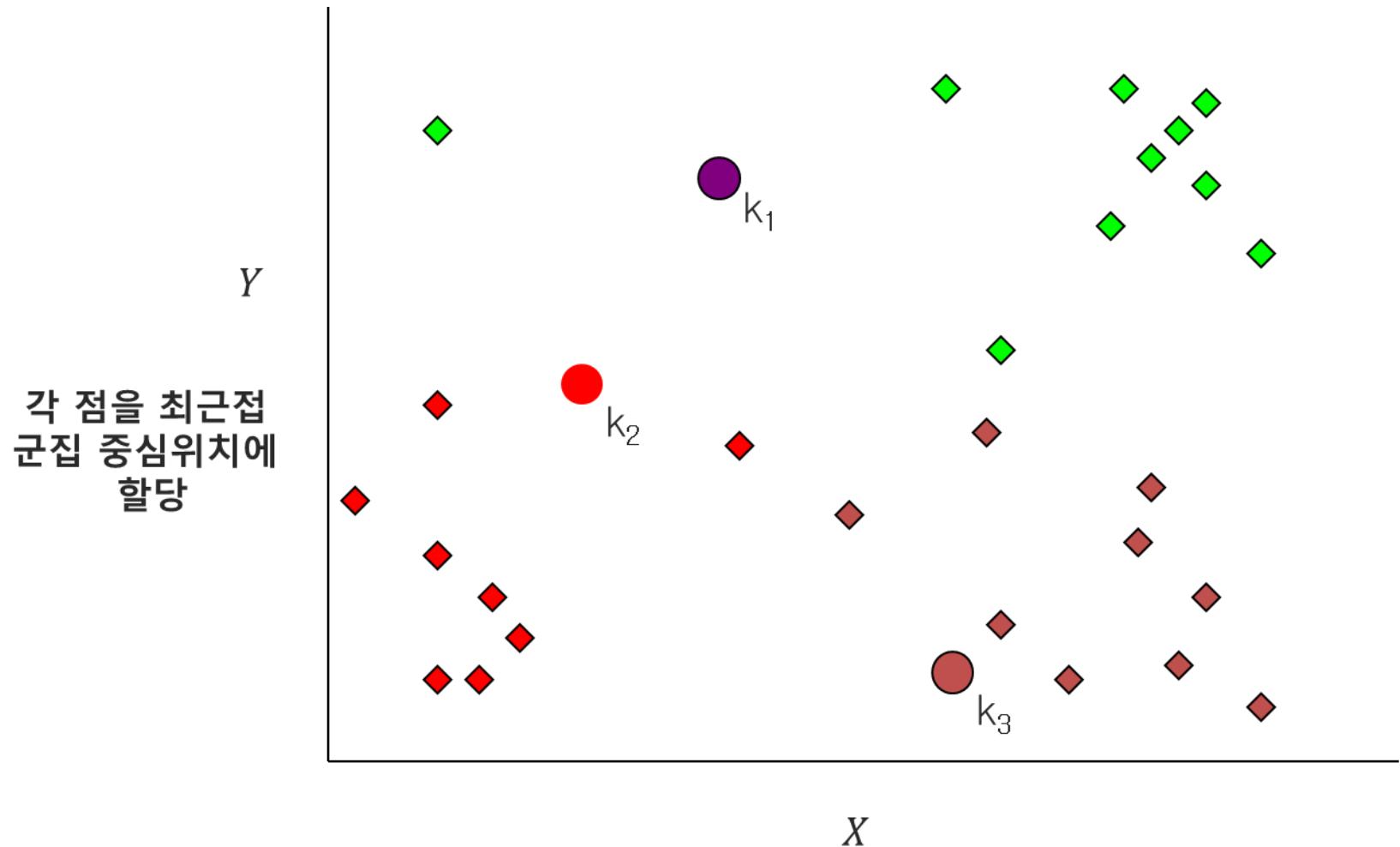
## k-means 알고리즘

1. 군집의 중심 위치 선정
2. 군집 중심을 기준으로 군집 재구성
3. 군집별 평균 위치 결정
4. 군집 평균 위치로 군집 중심 조정
5. 수렴할 때까지 2-4 과정 반복

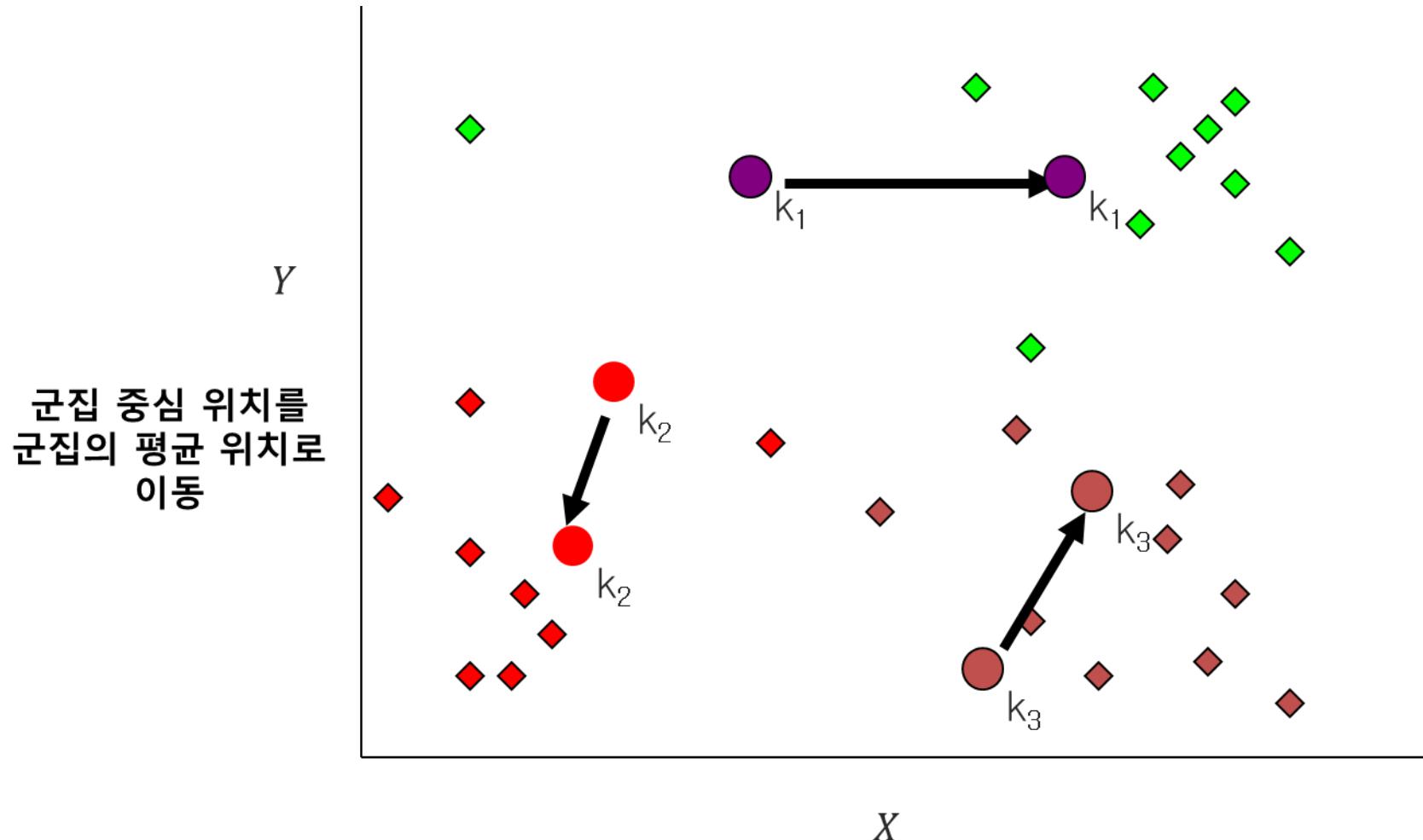
# *K-means clustering*



# *K-means clustering*

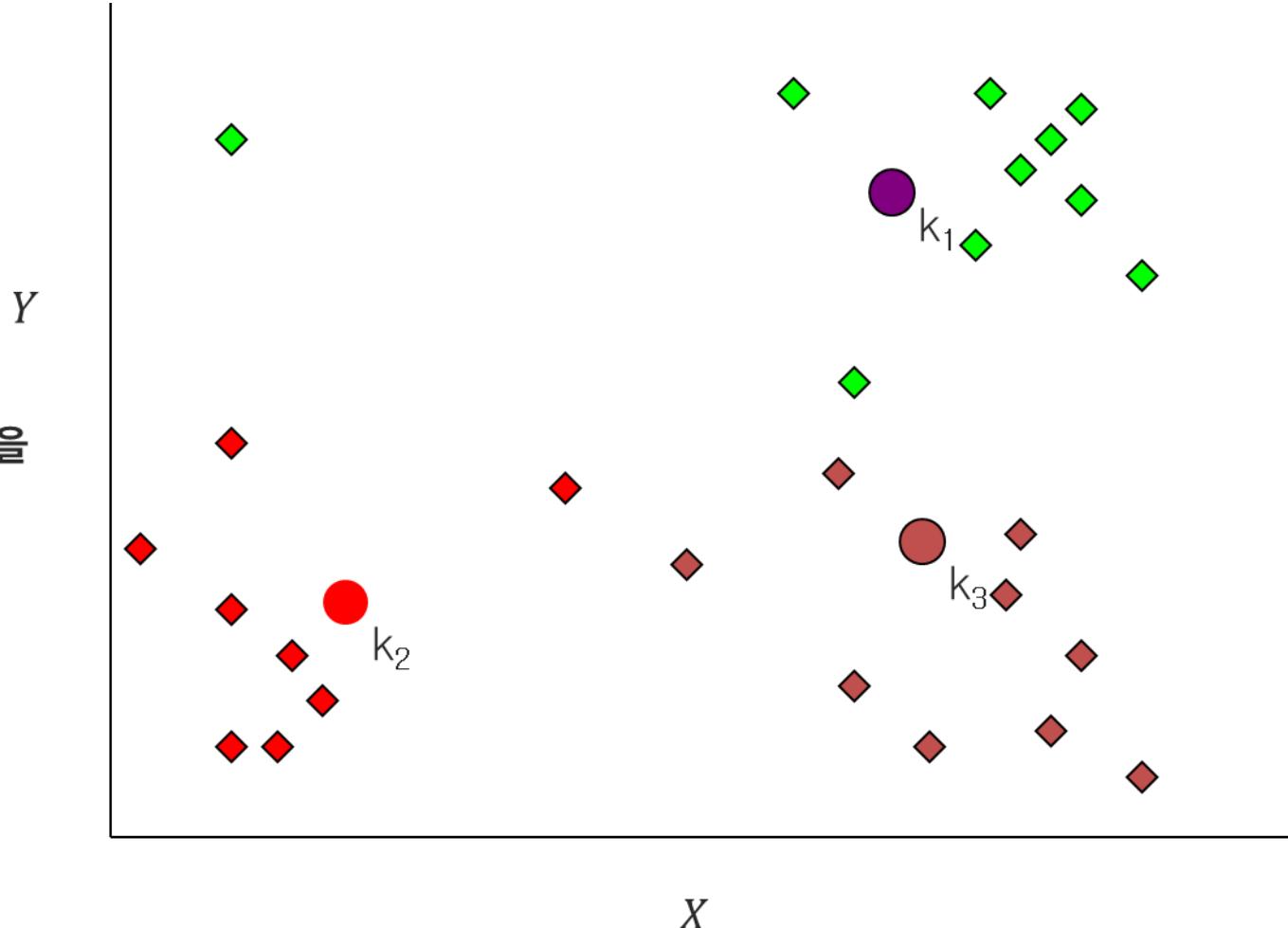


# *K-means clustering*

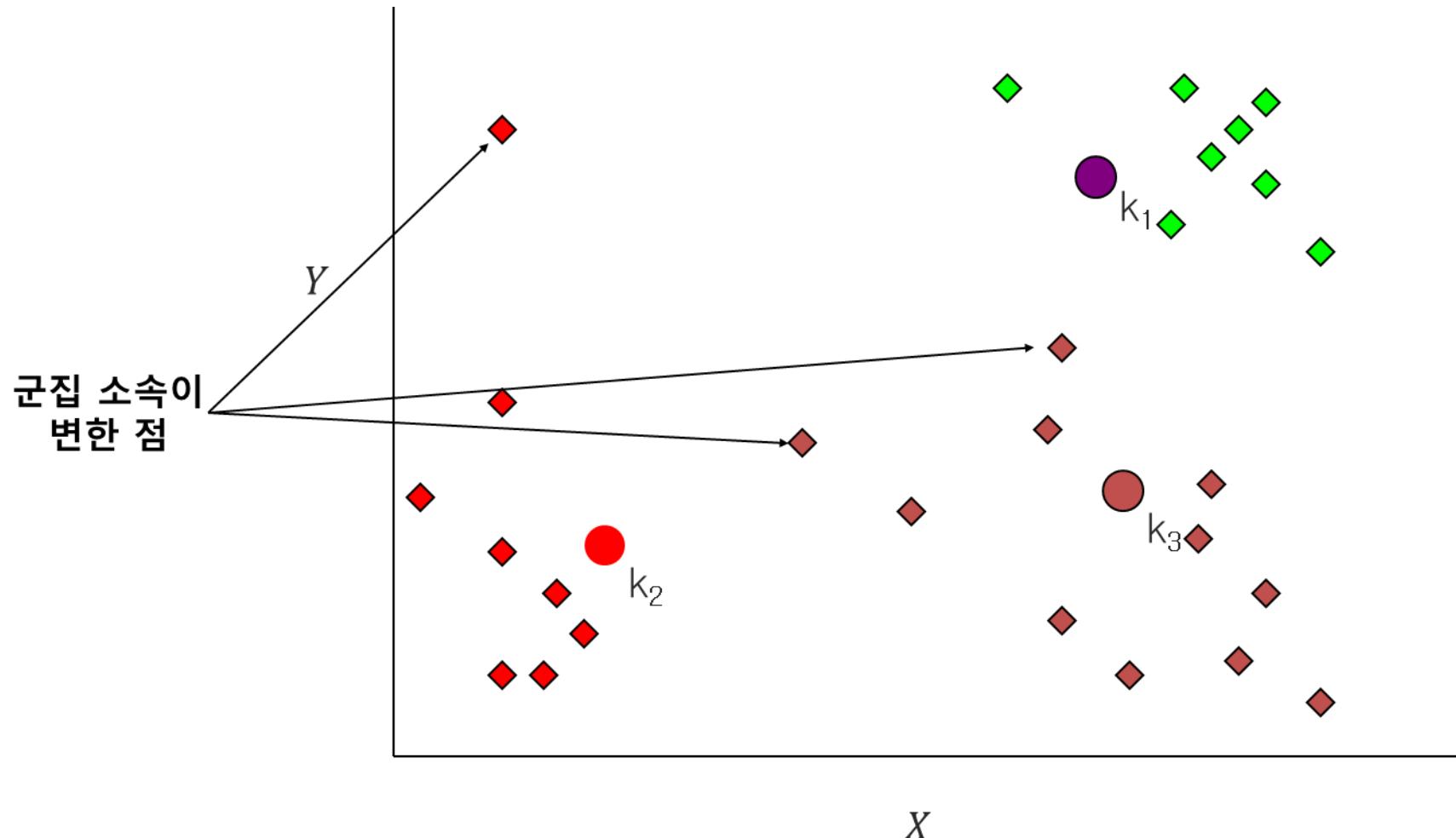


# *K-means clustering*

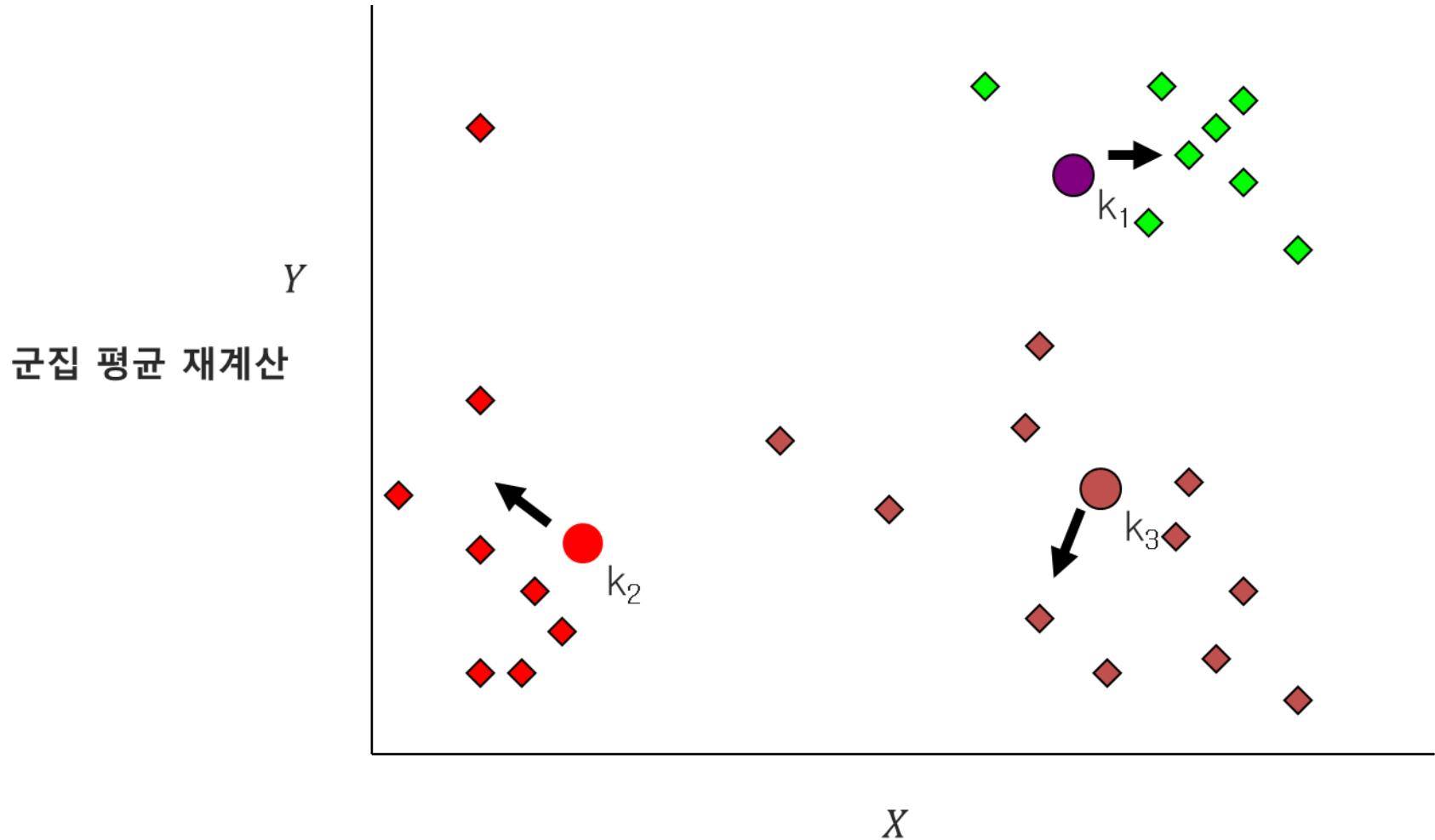
새로운 군집 중심을  
기준으로 각 점의  
소속을 재할당



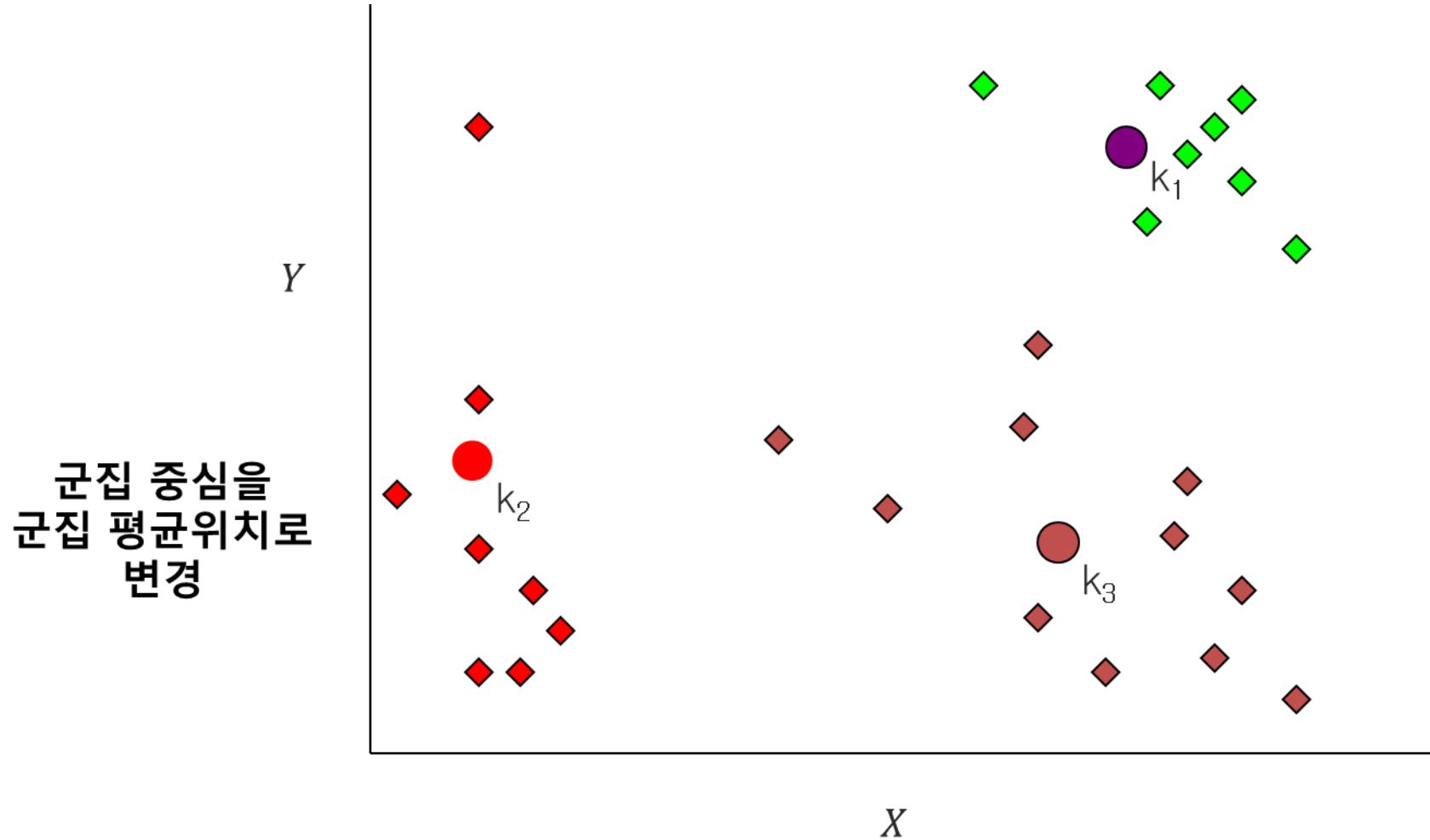
# *K-means clustering*



# *K-means clustering*



# *K-means clustering*



# *K-means clustering*

---

## k-means 알고리즘

- $i$  번째 클러스터의 중심을  $\mu_i$ , 클러스터에 속하는 점의 집합  $S_i$ 을 라고 할 때,  
전체 분산

$$V = \sum_{i=1}^k \sum_{j \in S_i} |x_j - \mu_i|^2$$

- 분산값  $V$ 을 최소화하는  $S_i$ 를 찾는 것이 알고리즘의 목표

### - 과정

1. 우선 초기의  $\mu_i$ 를 임의로 설정
2. 다음 두 단계를 클러스터가 변하지 않을 때까지 반복
  - I. 클러스터 설정: 각 점에 대해, 그 점에서 가장 가까운 클러스터를 찾아 배당한다.
  - II. 클러스터 중심 재조정:  $\mu_i$ 를 각 클러스터에 있는 점들의 평균값으로 재설정해준다.

### - 특성

- 군집의 개수  $k$ 는 미리 지정
- 초기 군집 위치에 민감

# *K-means clustering*

❖ 초기 중심값에 대해 민감한 군집화 결과

