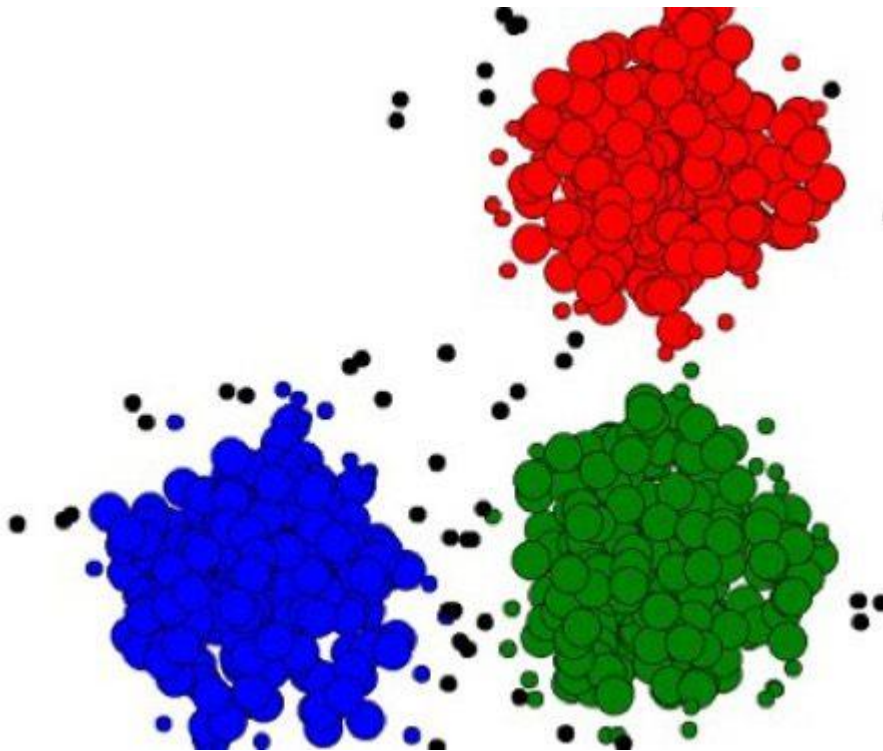


군집화

# 군집화(Clustering)

- 데이터 포인트들을 별개의 군집으로 그룹화 하는 것을 의미한다.
- 유사성이 높은 데이터들을 동일한 그룹으로 분류하고 서로 다른 군집들이 상이성을 가지도록 그룹화



# 군집화 활용 분야

- 고객, 마켓, 브랜드, 사회 경제 활동 세분화(Segmentation)
- Image 검출, 세분화, 트래킹
- 이상 검출(Anomaly Detection)

# 군집화 알고리즘

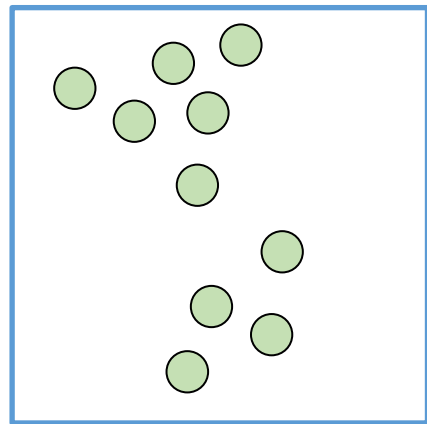
- K-Means
- Mean Shift
- Gaussian Mixture Model
- DBSCAN

# K-Means

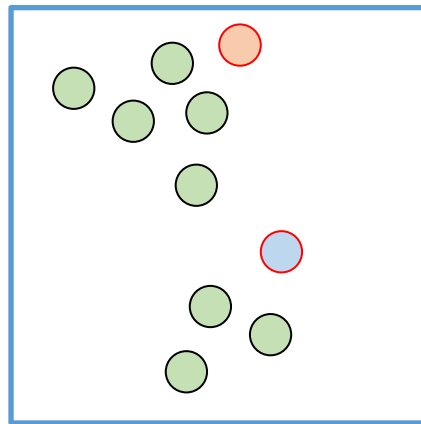
# 특성 추출(Feature Extraction)

- 군집 중심점(Centroid) 기반 클러스터링

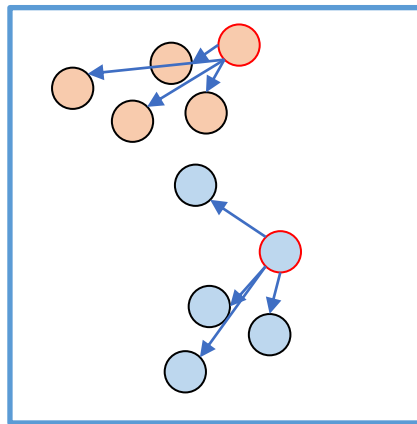
원본 데이터



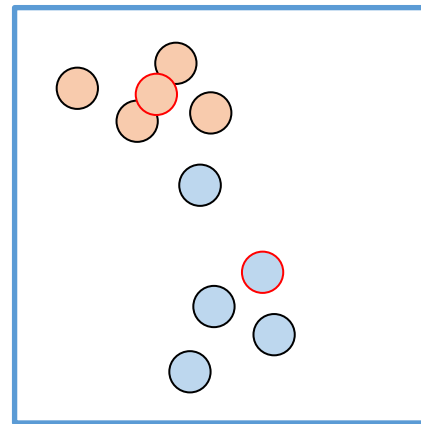
2개의 군집 중심점 설정



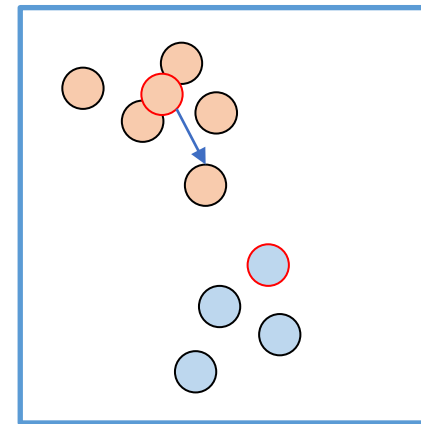
각 데이터들은 가장 가까운 중심점에 소속



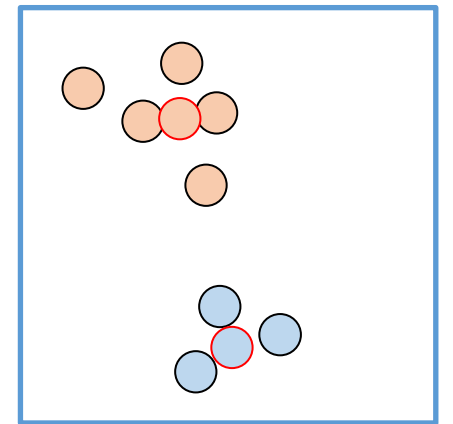
중심점에 할당된 데이터들의 평균 중심으로 중심점 이동



각 데이터들은 이동된 중심점 기준으로 가장 가까운 중심점에 소속

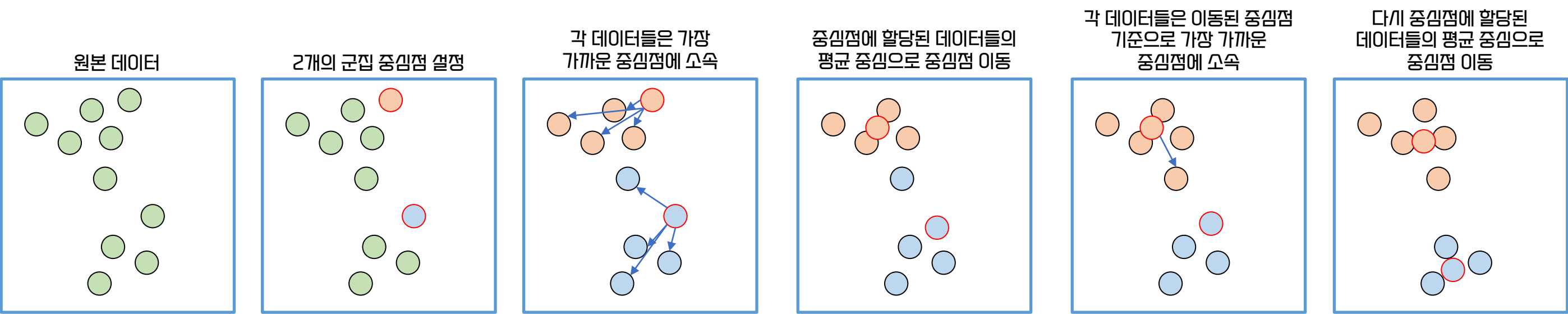


다시 중심점에 할당된 데이터들의 평균 중심으로 중심점 이동



# 특성 추출(Feature Extraction)

- 군집 중심점(Centroid) 기반 클러스터링



중심점을 이동하였지만 데이터들의 중심점 소속 변경이 없으면 군집화 완료

# K-Means의 장단점

- 장점
  - 일반적인 군집화에서 가장 많이 사용되는 알고리즘
  - 알고리즘이 쉽고 간결하다
  - 대용량 데이터에도 활용이 가능하다.
- 단점
  - 거리 기반 알고리즘으로 속성의 개수가 매우 많을 경우 군집화 정확도가 떨어지게 된다.
    - 이를 위해 보통 PCA로 차원 축소를 적용해야 할 수도 있다.
  - 반복을 수행하는데, 반복 횟수가 많을 경우 수행시간이 느려진다.
  - 이상치 데이터에 취약하다.



# 사이킷런 KMeans 클래스

- 사이킷런 패키지는 K-평균을 구현하기 위해 KMeans 클래스를 제공. KMeans 클래스는 다음과 같은 초기화 파라미터를 가지고 있다.

```
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')
```

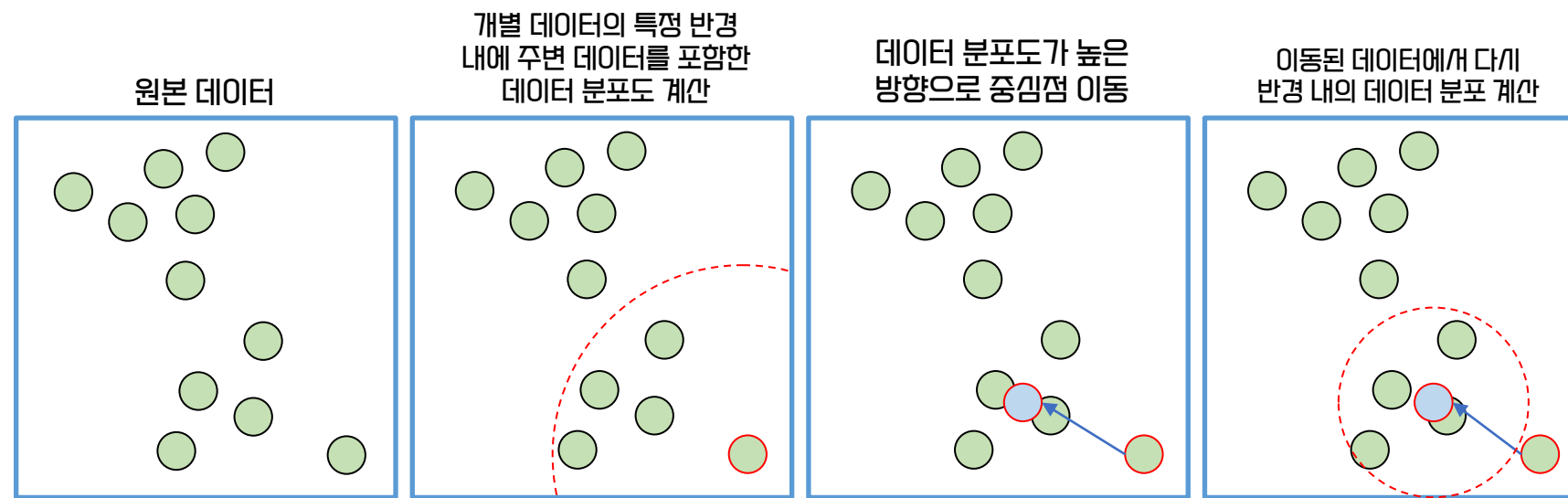
- KMeans 초기화 파라미터 중 가장 중요한 파라미터는 n\_clusters이다. 군집의 개수, 즉 군집 중심점의 개수를 지정한다
- max\_iter는 중심점 이동의 최대 반복 횟수이며, 이 횟수 이전에 모든 데이터의 중심점 이동이 없으면 종료한다.
- labels\_ : 각 데이터 포인트가 속한 군집 중심점 레이블이다.
- cluster\_centers\_ : 각 군집 중심점 좌표(Shape는 [군집 개수, 피쳐 개수]). 이를 이용해 군집 중심점 좌표가 어디인지 시각화 가능

# MeanShift

# Mean Shift 군집화

- Mean Shift는 KDE(Kernel Density Estimation)를 이용하여 데이터 포인트들이 데이터 분포가 높은 곳으로 이동하면서 군집화를 수행한다.
- 별도의 군집화 개수를 지정하지 않으며, Mean Shift는 데이터 분포도에 기반하여 자동으로 군집화 개수를 정한다.

# Mean Shift 수행 절차



특정 데이터가 반경 내의 데이터 분포  
확률 밀도가 가장 높은 곳으로 이동 할 때  
주변 데이터들과의 거리 값을 커널  
함수로 입력한 뒤 그 반환값을 현재  
위치에서 갱신하면서 이동

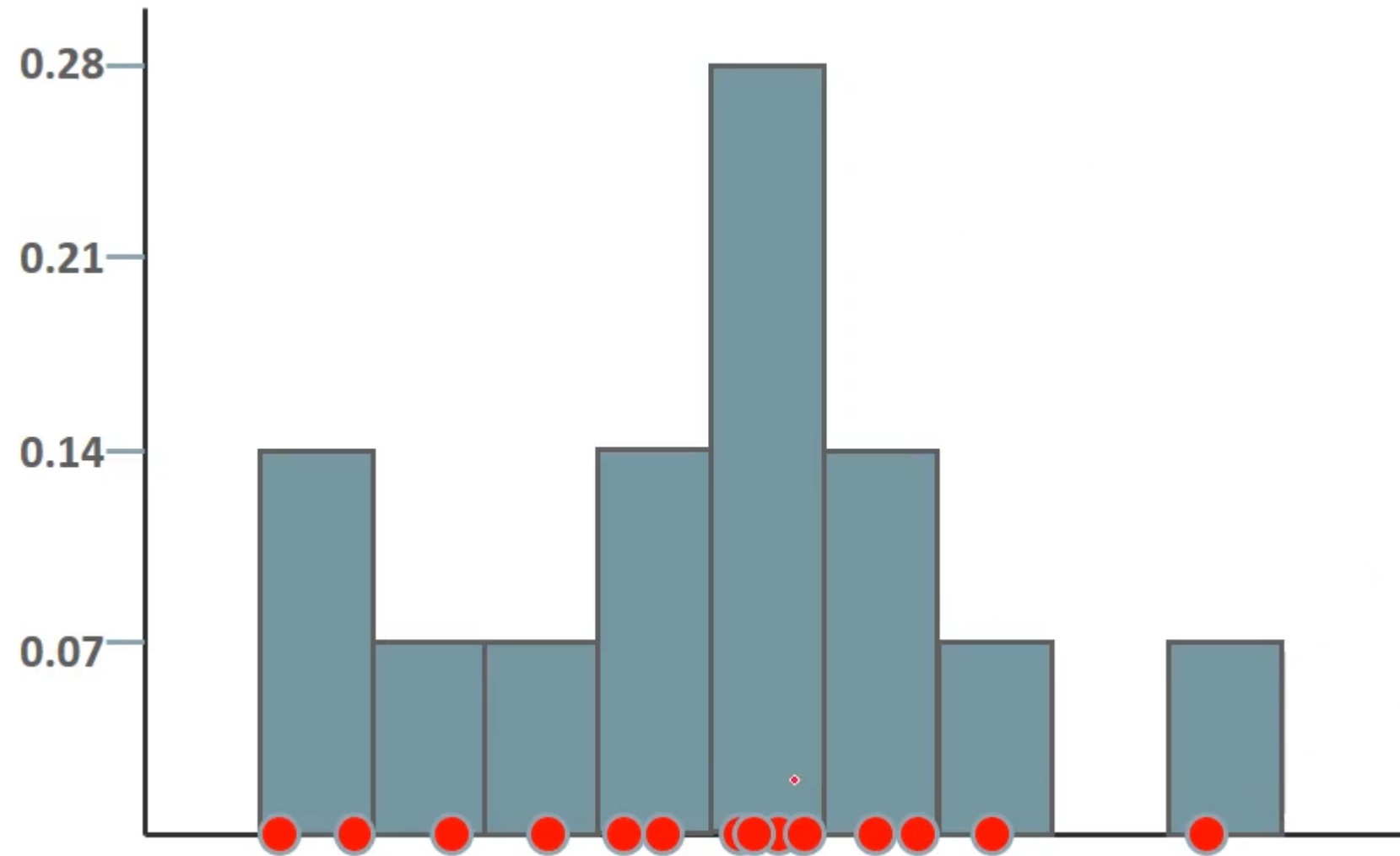
# KDE(Kernel Density Estimation)

- KDE는 커널(Kernel) 함수를 통해 어떤 확률 변수의 확률밀도 함수를 추정하는 방식이다.
- 관측된 데이터 각각에 커널 함수를 적용한 값을 모두 더한 뒤 데이터 건수로 나누어 확률 밀도 함수를 추정한다.
- 확률 밀도 함수(PDF : Probability Density Function) : 확률 변수의 분포를 나타내는 함수로써, 대표적으로 정규 분포, 감마 분포, t-분포 등이 있다.
- 확률 밀도 함수를 알게 되면 특정 변수가 어떤 값을 갖게 될지의 확률을 알게 됨을 의미한다. 즉 확률 밀도 함수를 통해 변수의 특성( 예를 들어 정규 분포의 경우 평균, 분산), 확률 분포 등 변수의 많은 요소를 알 수 있게 된다.

# 확률 밀도 추정 방법

- 모수적(Parametric) 추정
  - 데이터가 특정 데이터 분포(예를 들어 가우시안 분포)를 따른다는 가정 하에 데이터 분포를 찾는 방법.  
Gaussian Mixture 등이 여기에 포함된다.
- 비모수적(Non-Parametric) 추정
  - 데이터가 특정 분포를 따르지 않는다는 가정 하에서 밀도를 추정하는 방법
  - 관측된 데이터 만으로 확률 밀도를 찾는 방법으로 대표적으로 KDE가 있다.

# 비모수적 밀도 추정 - Histogram



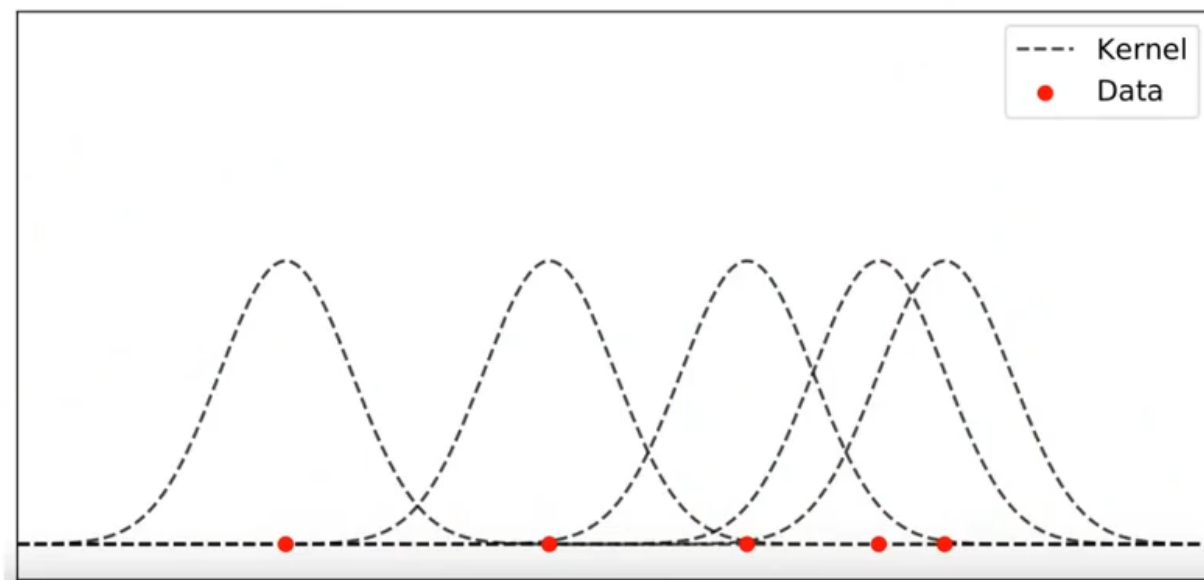
## 히스토그램 밀도 추정의 문제점

- Bin의 경계에서 불연속성이 나타난다.
- Bin의 크기에 따라 히스토그램이 달라진다.

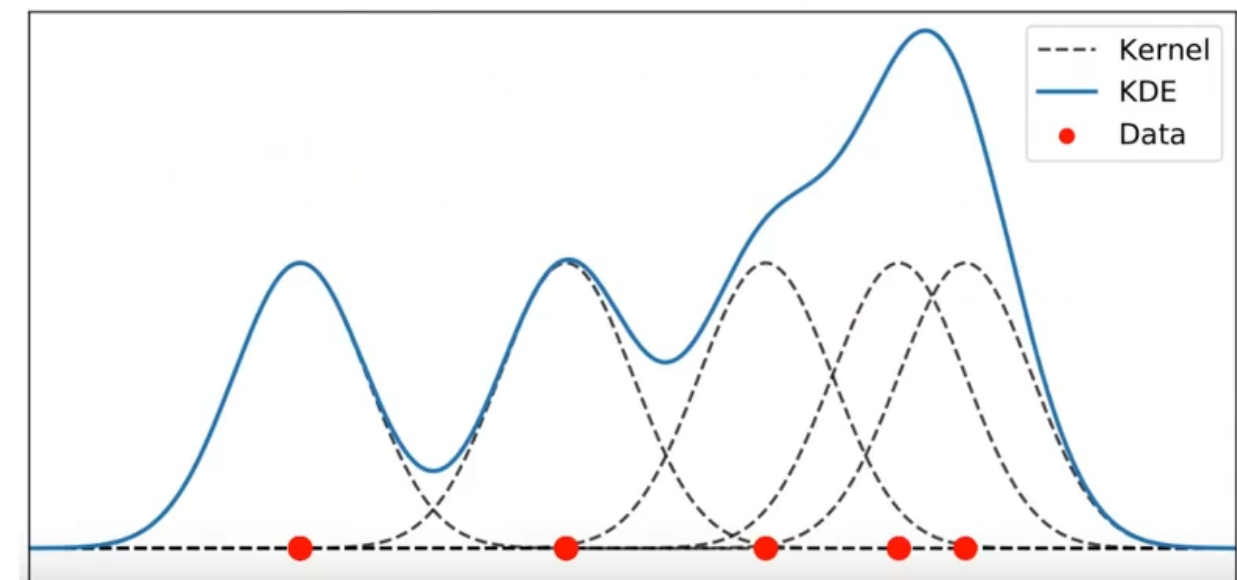
# 비모수적 밀도 추정 - KDE

- KDE는 개별 관측 데이터들에 커널함수를 적용한 뒤, 커널 함수들의 적용값을 모두 합한 뒤에 개별 관측 데이터의 건수로 나누어서 확률밀도 함수를 추정하는 방식이다. 대표적인 커널함수로 가우시안 분포함수가 사용된다.

개별 관측 데이터에 가우시안 커널 함수 적용



가우시안 커널 함수 적용 후 합산





# KDE와 가우시안 커널함수

- KDE는 아래와 같은 커널함수 식으로 표현된다.  $K$ 는 커널함수,  $x$ 는 random값,  $x_i$ 는 관측값,  $h$ 는 bandwidth

- $KDE = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i) = \frac{1}{nh} \sum_{i=1}^n K_h\left(\frac{x - x_i}{h}\right)$

- 대표적인 커널함수  $K_h$ 는 가우시안 분포이다.

- $f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

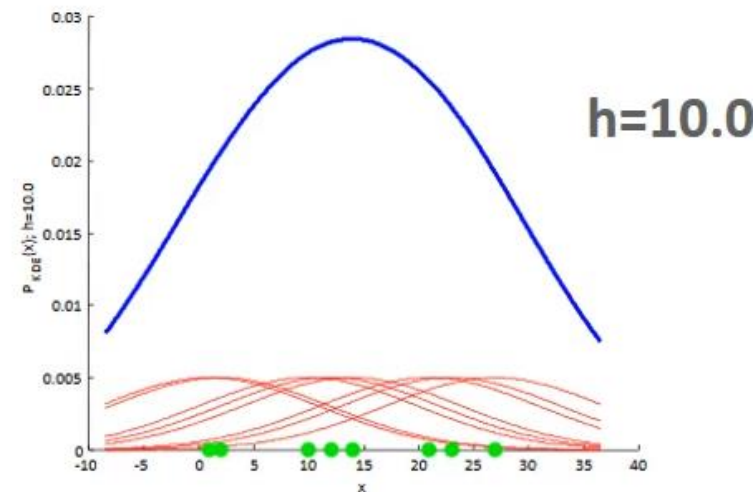
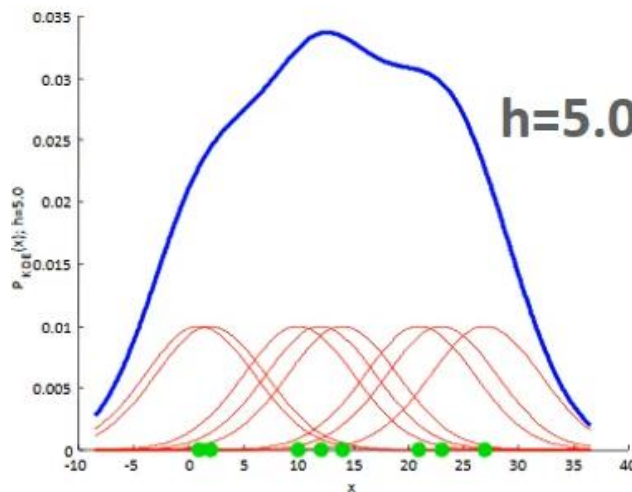
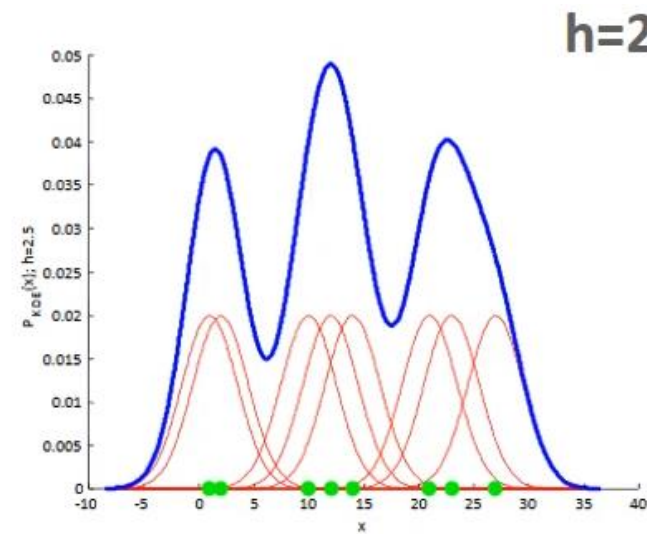
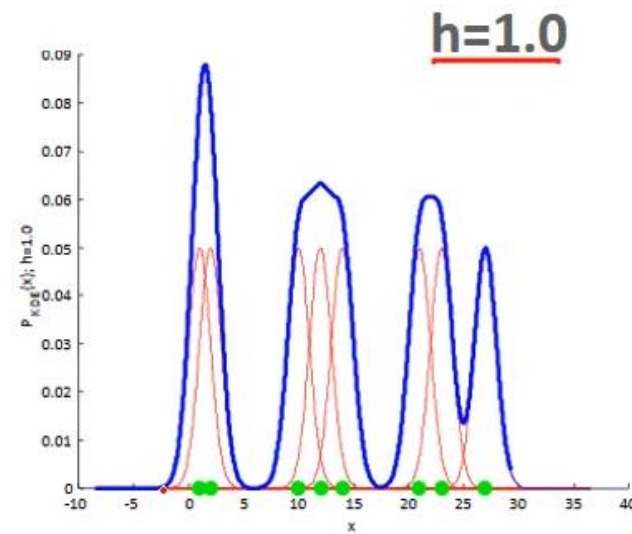
- 가우시안 커널함수를 적용한 KDE는 다음과 같다. 이 경우 관측값  $x_i$ 는 평균, bandwidth  $h$ 는 표준편차와 동일.

- $KDE = \frac{1}{nh} \sum_{i=1}^n \frac{1}{\sqrt{2\pi h}} e^{-\frac{1}{2}\left(\frac{x - x_i}{h}\right)^2}$

- 참고로 가우시안 커널함수를 적용할 경우 최적의 bandwidth는 다음과 같다.

- $h = \left(\frac{4\sigma^5}{3n}\right)^{\frac{1}{5}} \approx 1.06\sigma n^{-\frac{1}{5}}$ . 단  $n$ 은 샘플 데이터의 개수,  $\sigma$ 는 샘플 데이터의 표준편차

# Bandwidth에 따른 KDE의 변화



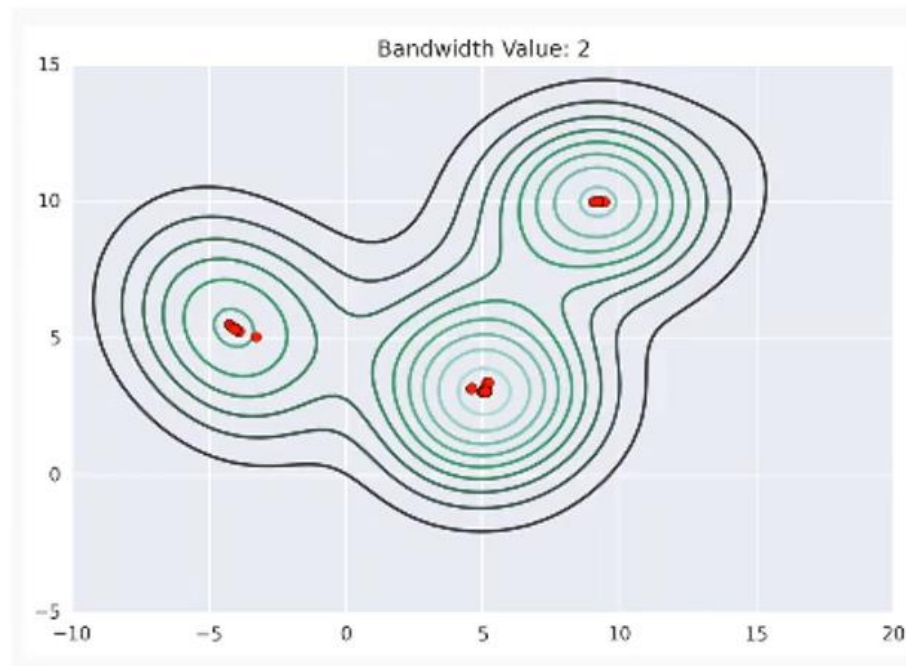
작은  $h$  값은 좁고 뾰족한 KDE로  
변동성이 큰 확률밀도함수를  
추정한다.(과대적합)

큰  $h$  값은 과도하게 부드러운 KDE로  
단순화된 확률밀도함수를  
추정한다.(과소적합)

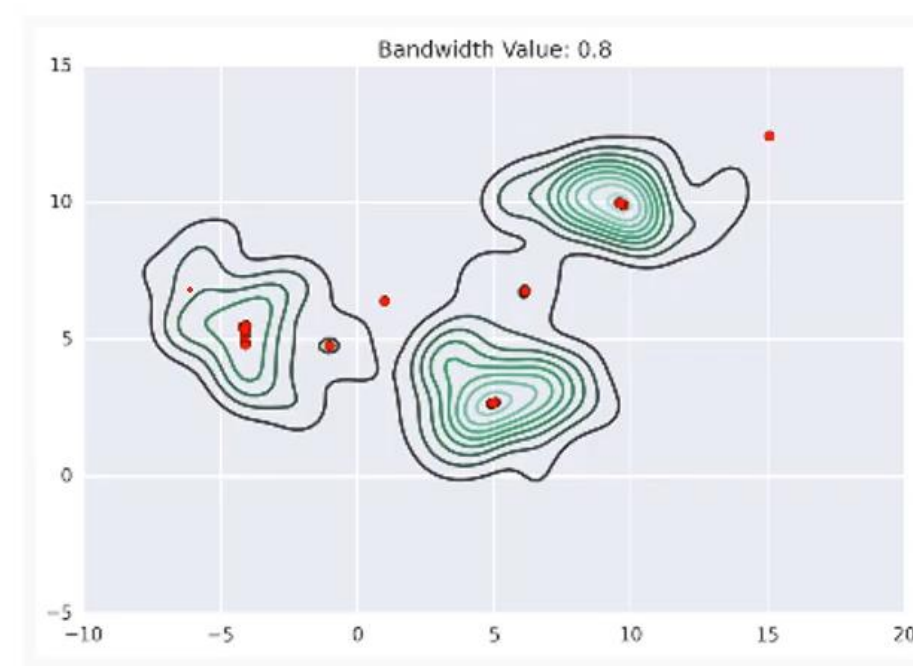
# Bandwidth에 따른 KDE의 변화

- Mean Shift는 Bandwidth가 클수록 적은 수의 클러스터링 중심점을, Bandwidth가 작을 수록 많은 수의 클러스터링 중심점을 가지게 된다. 또한 Mean Shift는 군집의 개수를 지정하지 않고, bandwidth의 크기에 따라 군집을 수행한다.

Bandwidth: 2



Bandwidth: 0.8



# 사이킷런 Mean Shift

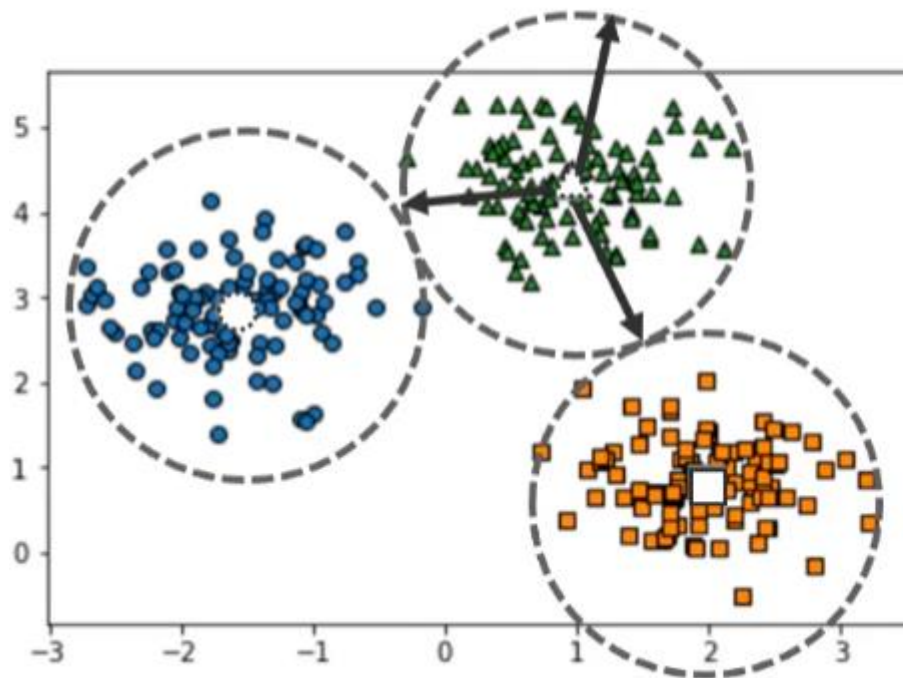
- 사이킷런은 Mean Shift 군집화를 위해 MeanShift 클래스를 제공한다.
- MeanShift 클래스의 가장 중요한 초기화 파라미터는 bandwidth 이며 해당 파라미터는 밀도 중심으로 이동할 때 사용되는 커널 함수의 bandwidth이다. 이 bandwidth를 어떻게 설정하느냐에 따라 군집화 성능이 달라지게 된다.
- 최적의 bandwidth 계산을 위해 사이킷런은 estimate\_bandwidth() 함수를 제공한다.

**GMM**

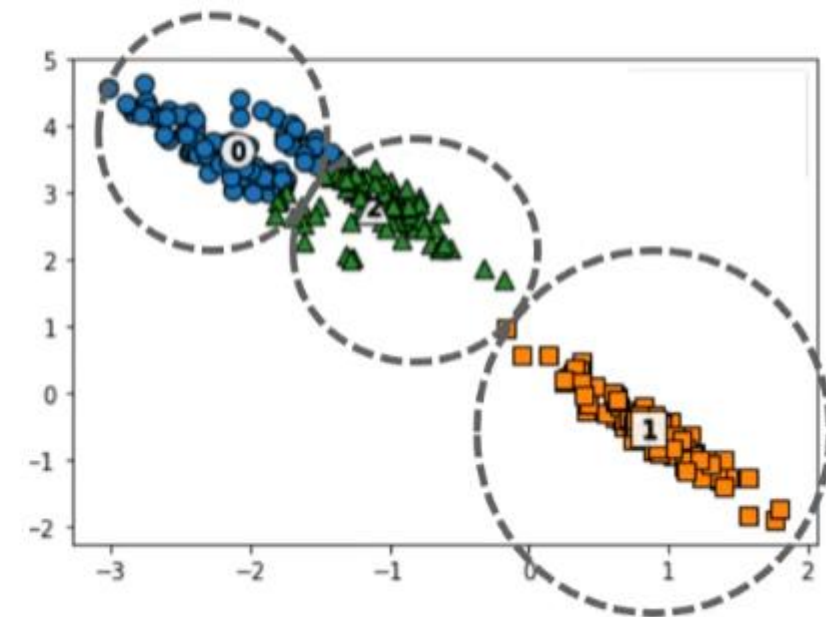
# GMM - 거리기반 K-Means의 문제점

- K-Means는 특정 중심점을 기반으로 거리적으로 퍼져있는 데이터 세트에 군집화를 적용하면 효율적이다. 하지만 K-Means는 이러한 데이터 분포를 가지지 않는 데이터 세트에 대해서는 효율적인 군집화가 힘들다.

Kmeans로 효율적인 군집화 가능

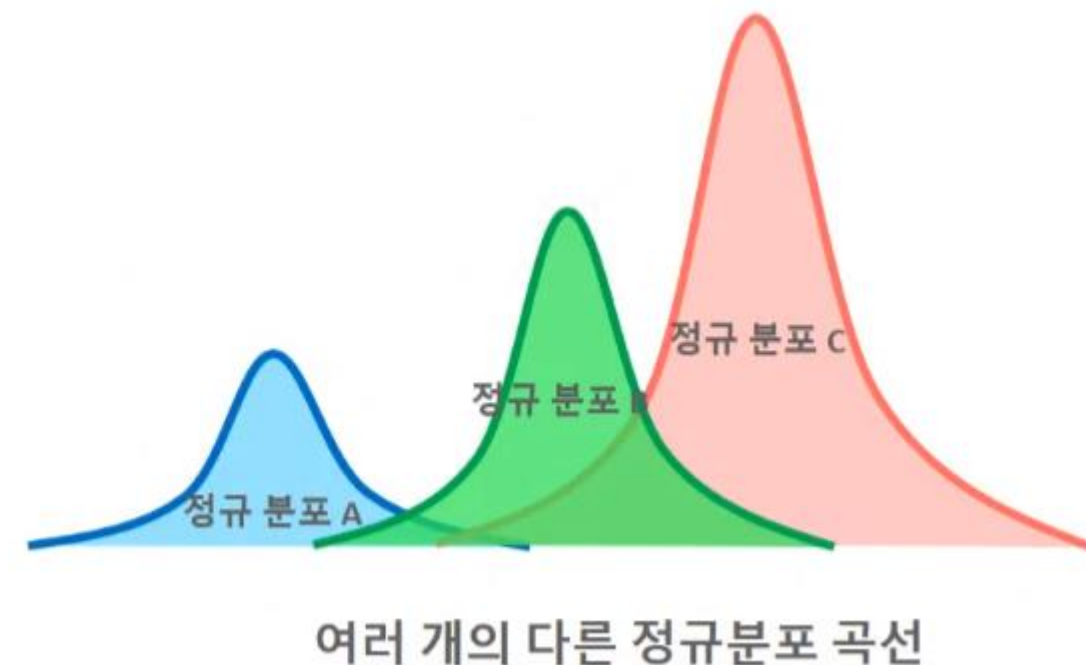
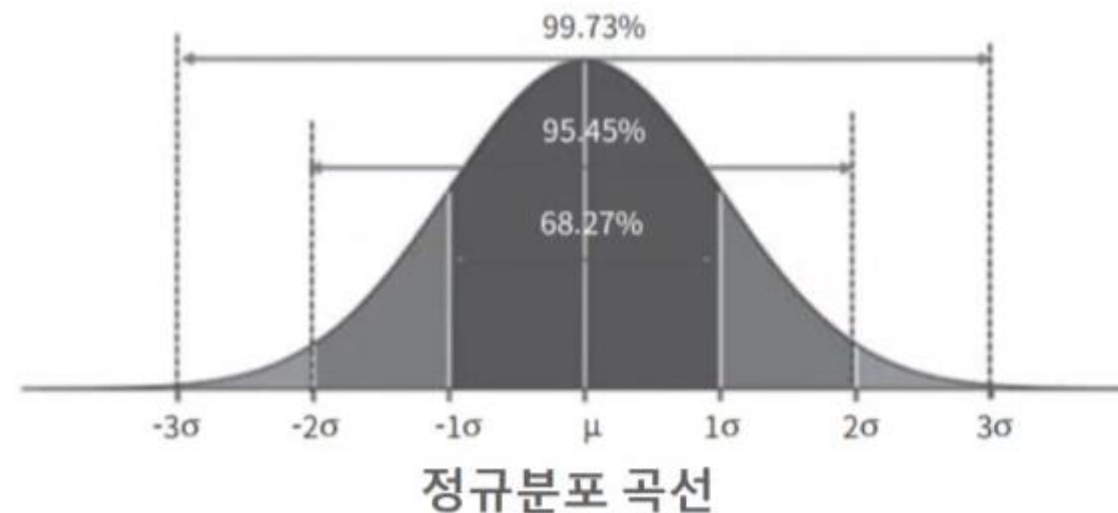


Kmeans로 군집화가 어려운 데이터



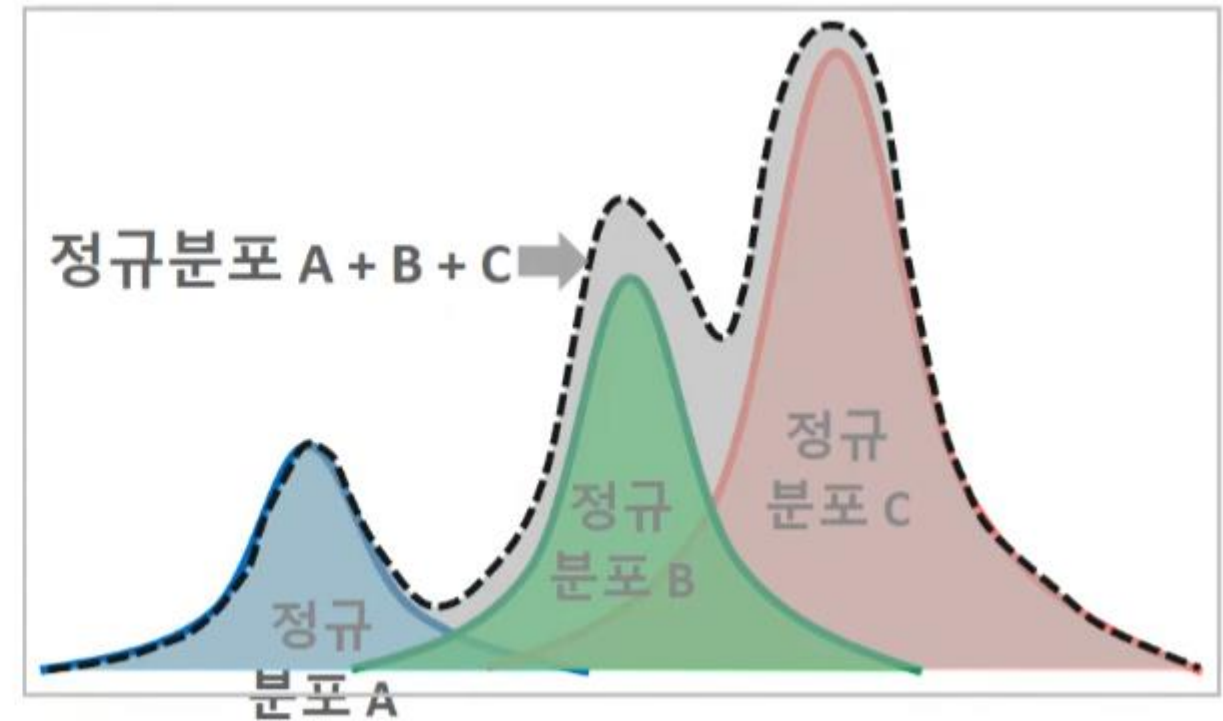
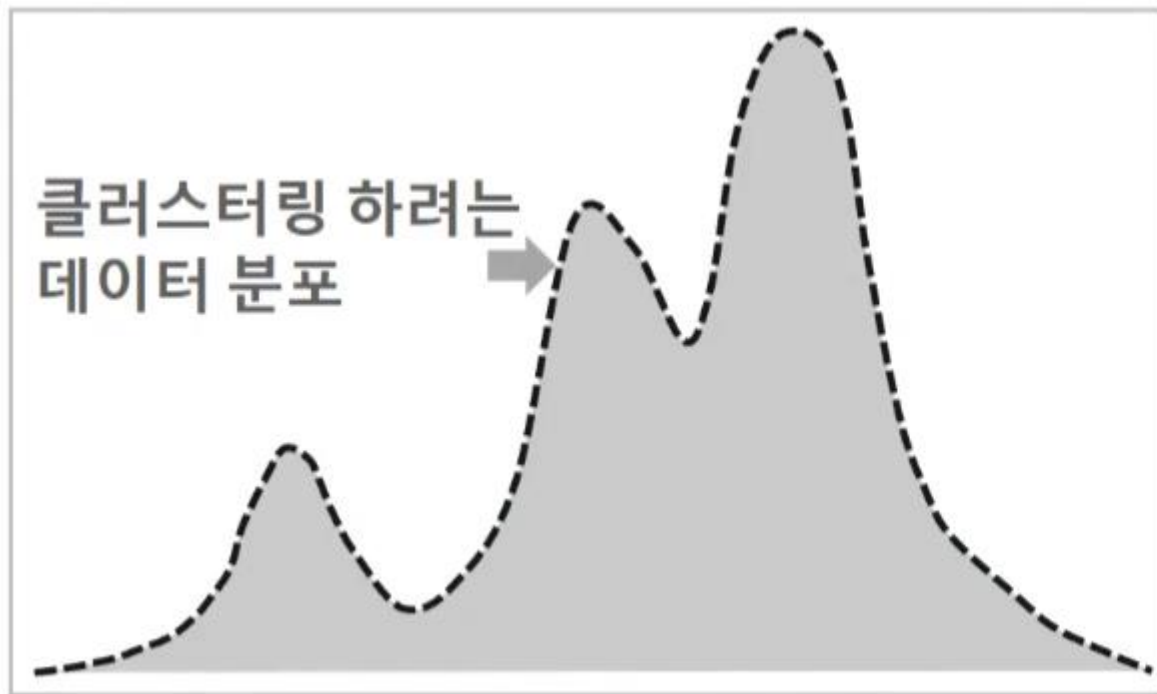
# GMM(Gaussian Mixture Model) 개요

- GMM 군집화는 군집화를 적용하고자 하는 데이터가 여러 개의 다른 가우시안 분포(Gaussian Distribution)를 가지는 모델로 가정하고 군집화를 수행하게 된다.
- 가령 1000개의 데이터 세트가 있다면 이를 구성하는 여러 개의 정규 분포 곡선을 추출하고, 개별 데이터가 이 중 어떤 정규 분포에 속하는지 결정하는 방식이다.





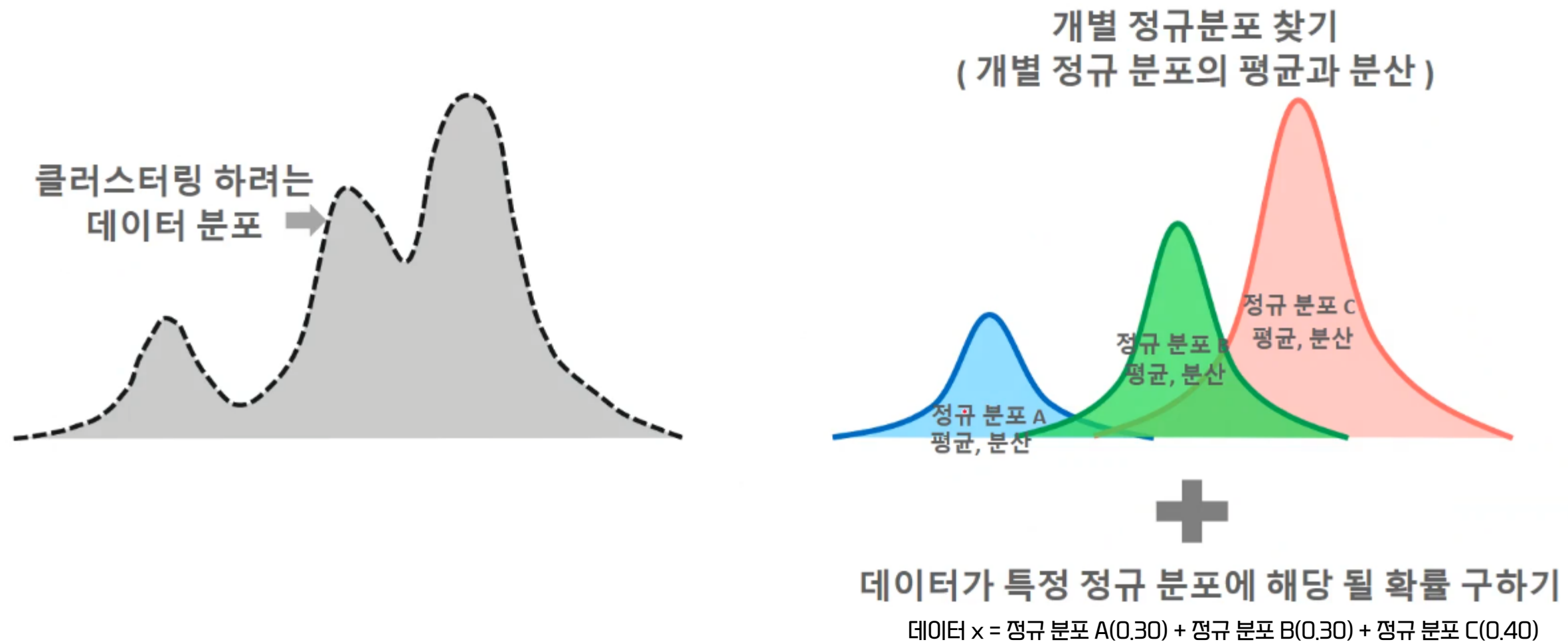
# 새로 다른 정규분포로 결합된 원본 데이터 분포





# GMM 모수(Parameter) 추정

- GMM 모수 추정은 개별 정규 분포들의 평균과 분산, 그리고 데이터가 특정 정규 분포에 해당 될 확률을 추정



# GMM 모수 추정을 위한 EM(Expectation and Maximization)

- Expectation
  - 개별 데이터 각각에 대해/개 특정 정규 분포에 소속될 확률을 구하고, 가장 높은 확률을 가진 정규 분포에 소속(최초시에는 데이터를 임의로 특정 정규 분포로 소속)
- Maximization
  - 데이터들이 특정 정규분포로 소속되면 다시 해당 정규분포의 평균과 분산을 구한다.
  - 해당 데이터가 발견될 수 있는 가능도를 최대화(Maximum likelihood) 할 수 있도록 평균과 분산(모수)를 구한다.

개별 정규분포의 모수인 평균과 분산이 더 이상 변경되지 않고 개별 데이터들의 이진 정규분포 소속이 더 이상 변경되지 않으면 그것으로 최종 군집화를 결정하고 그렇지 않으면 계속 Expectation과 Maximization을 반복

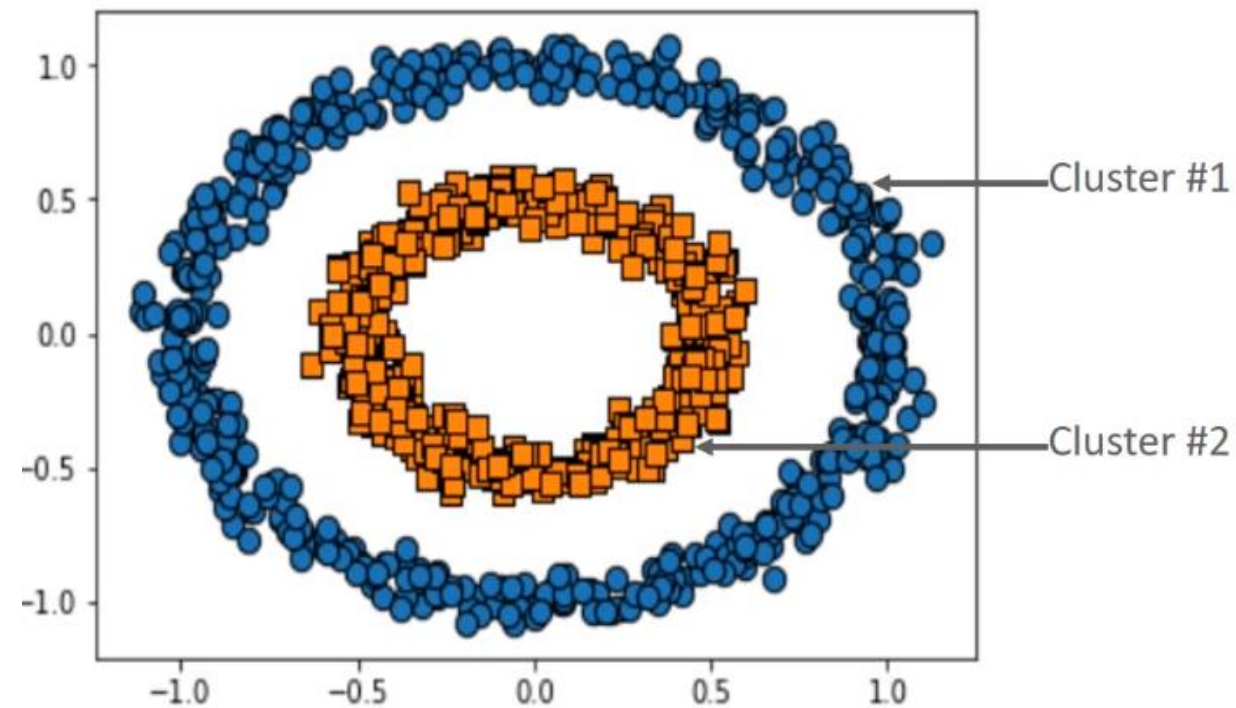
# 사이킷런 GaussianMixture

- 사이킷런은 GMM 군집화를 위해 GaussianMixture 클래스를 제공한다.
- GaussianMixture 클래스의 주요 생성자 파라미터는 `n_components` 이며, 이는 Mixture Model의 개수, 즉 군집화 개수를 의미한다.

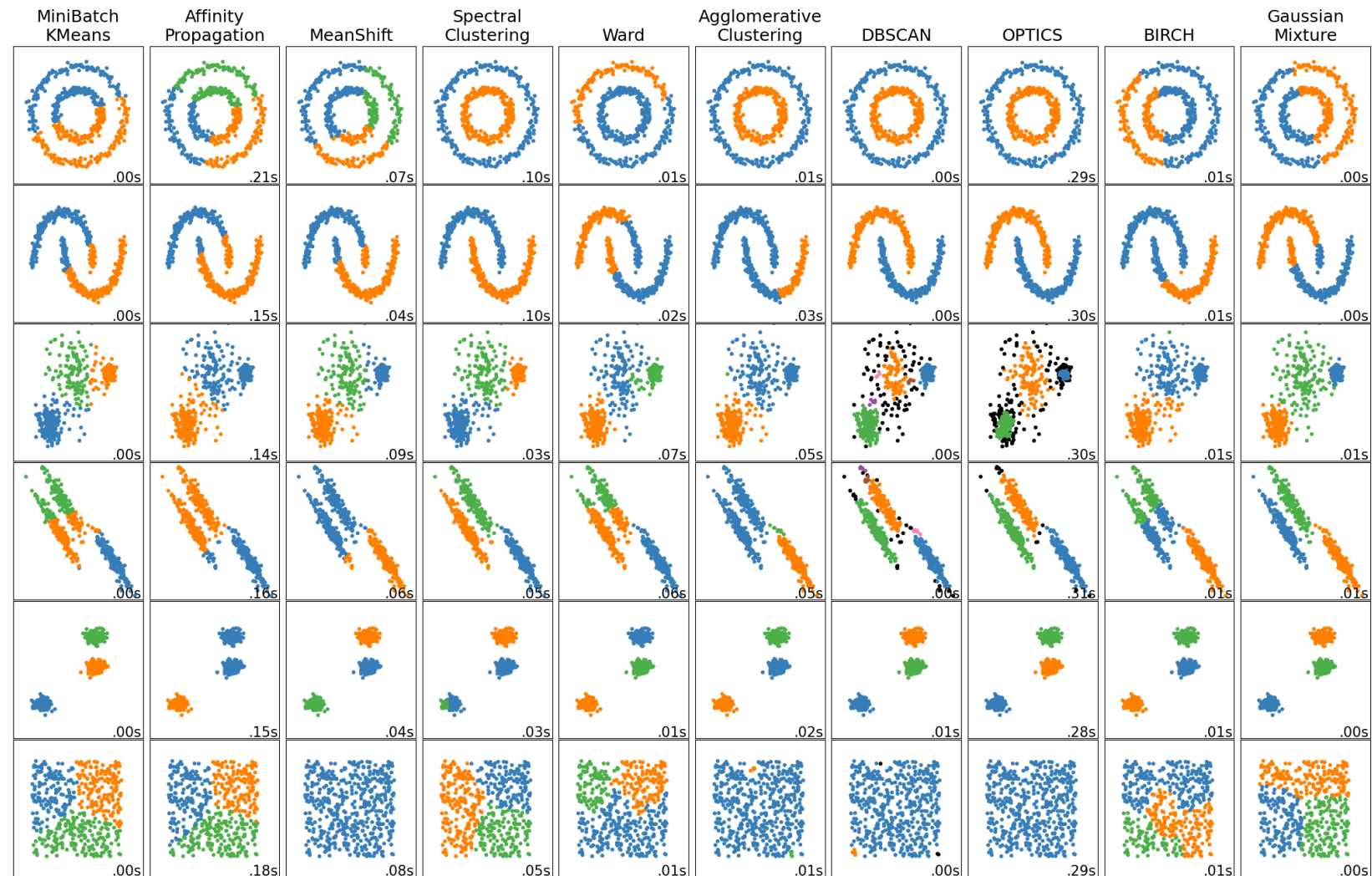
**DBSCAN**

# DBSCAN(Density Based Spatial Clustering of Applications with Noise)

- DBSCAN은 특정 공간 내에 데이터 밀도 차이를 기반 알고리즘으로 하고 있어서 기하학적 분포도를 가진 데이터 세트에 대해서도 군집화를 잘 수행할 수 있다.
- 알고리즘이 데이터 밀도 차이를 자동으로 감지하여 군집을 생성하므로 사용자가 군집 개수를 지정할 수 없다.



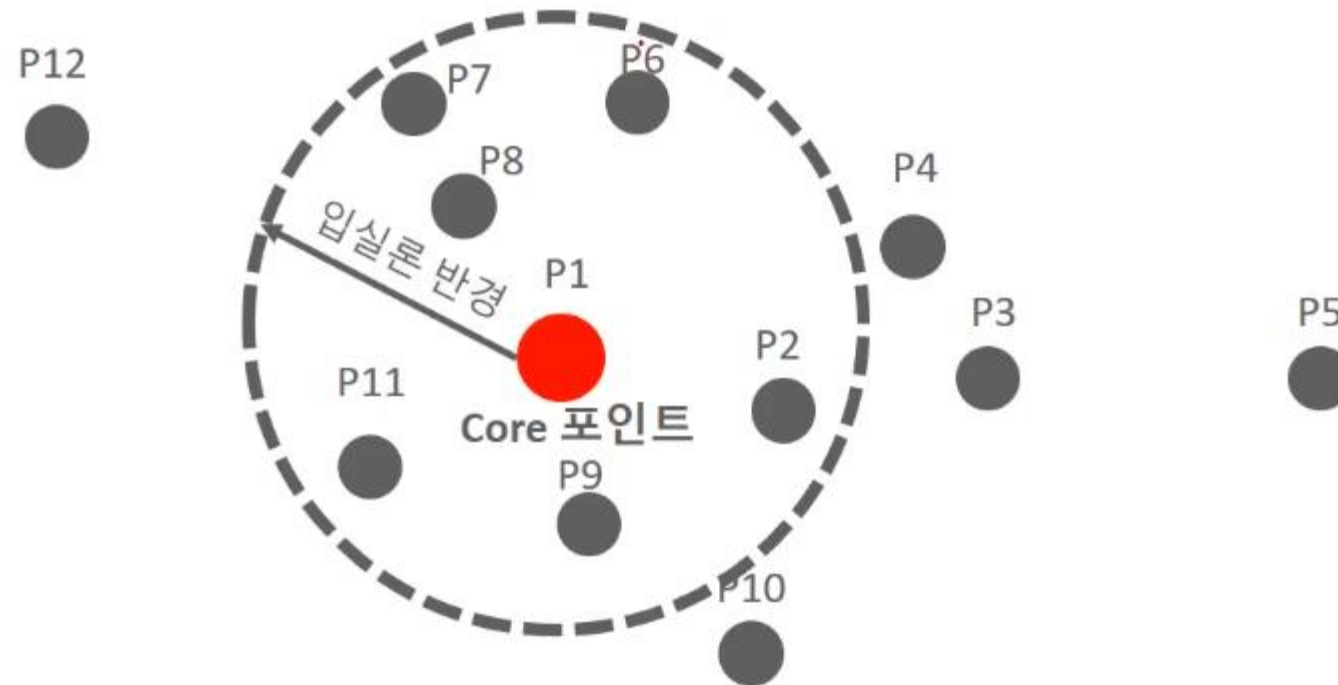
# 군집화 알고리즘 비교



- DBSCAN은 데이터의 밀도가 자주 변하거나, 아예 모든 데이터의 밀도가 크게 변하지 않으면 군집화 성능이 떨어지게 된다.
- 피쳐의 개수가 많으면 군집화 성능이 떨어지게 된다.

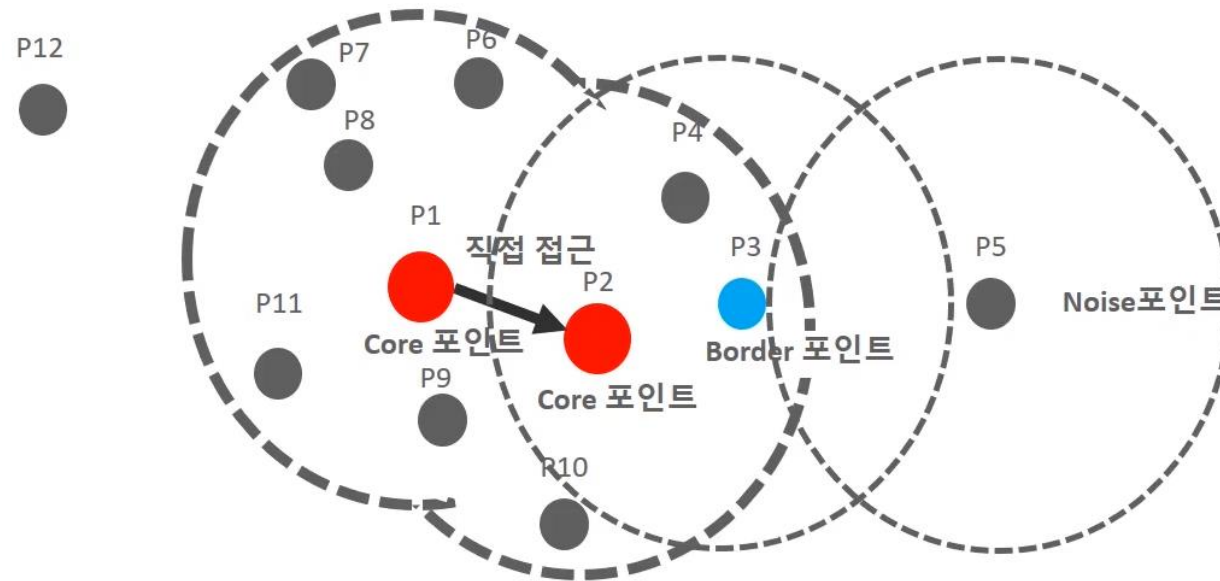
# DBSCAN 구성 요소

- DBSCAN을 구성하는 가장 중요한 두 가지 파라미터는 입실론(epsilon)으로 표기하는 주변 영역과 이 입실론 주변 영역에 포함되는 최소 데이터의 개수 min\_points
  - 입실론 주변 영역(epsilon) : 개별 데이터를 중심으로 입실론 반경을 가지는 원형의 영역
  - 최소 데이터 개수(min points) : 개별 데이터의 입실론 주변 영역에 포함되는 타 데이터의 개수



# DBSCAN 구성 요소

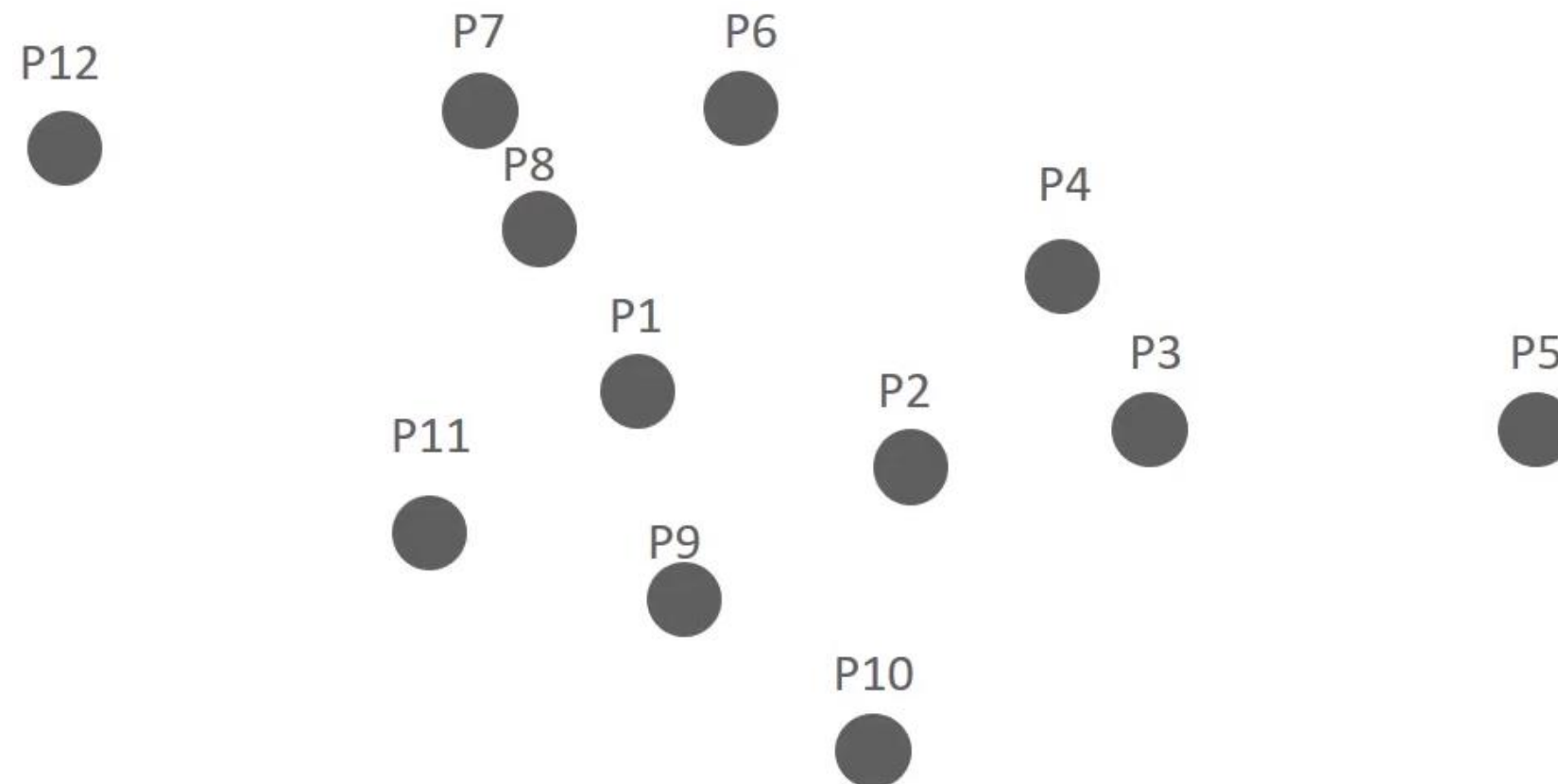
- 입실론 주변 영역 내에 포함되는 최소 데이터 개수를 충족시키지 못하는 데이터 포인트를 다음과 같이 정의한다.
  - 핵심 포인트(Core Point) : 주변 영역 내 최소 데이터 개수 이상의 타 데이터를 가지고 있을 경우 해당 데이터
  - 이웃 포인트(Neighbor Point) : 주변 영역 내에 위치한 타 데이터
  - 경계 포인트(Border Point) : 주변 영역 내에 최소 데이터 개수 이상의 이웃 포인트를 가지고 있지 않지만 핵심포인트를 이웃 포인트로 가지고 있는 데이터
  - 잡음 포인트(Noise Point) : 최소 데이터 개수 이상의 이웃 포인트를 가지고 있지 않으며, 핵심 포인트도 이웃 포인트로 가지고 있지 않는 데이터





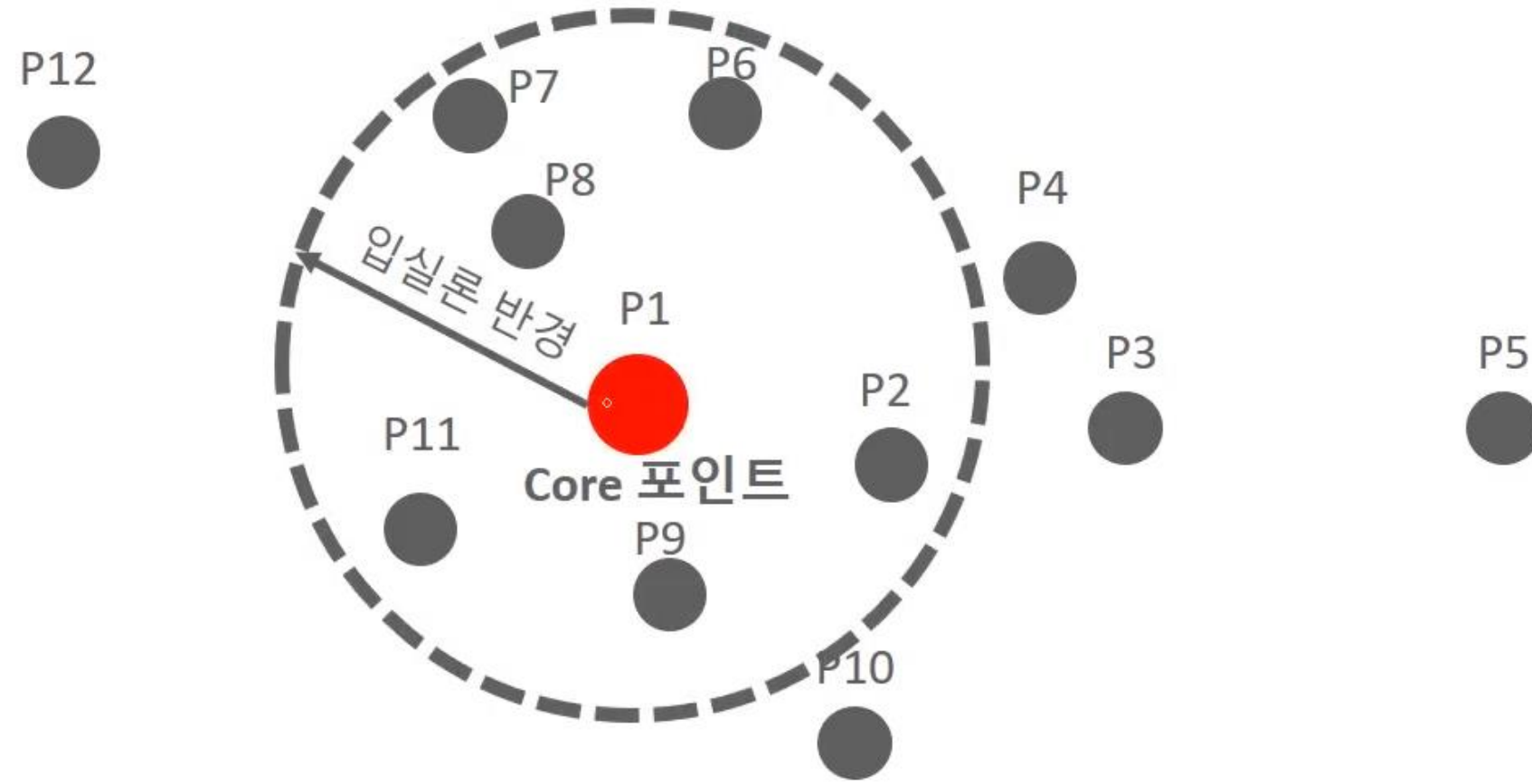
# DBSCAN 절차 1

- P1에서 P12 까지 12개의 데이터 세트에 대해서 DBSCAN 군집화를 적용하는 예시이다. 특정 임실론 반경 내에 포함될 최소 데이터 세트를 6개로(자기 자신 포함) 가정한다.



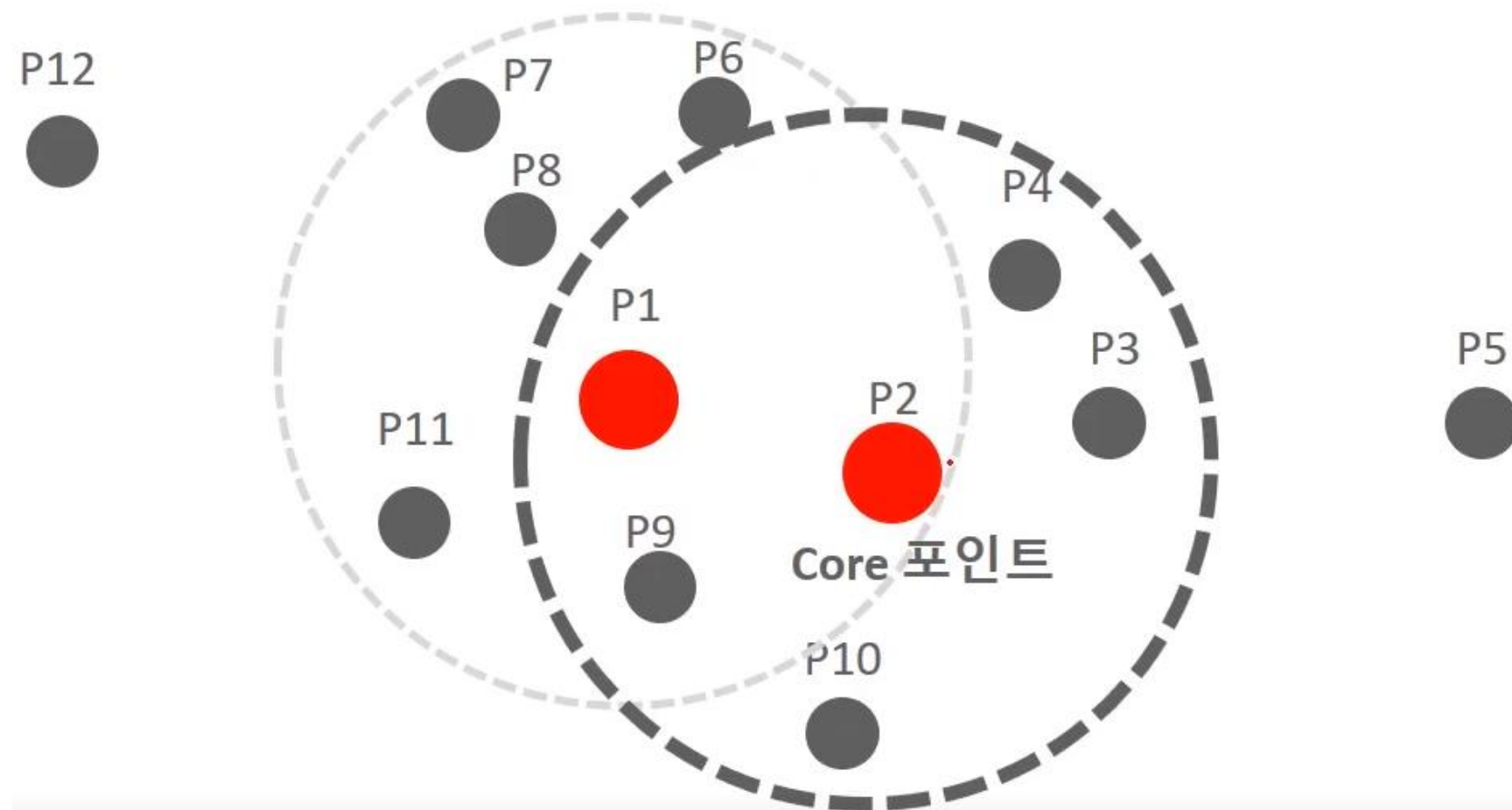
## DBSCAN 절차 2

- P1 데이터를 기준으로 임실론 반경 내에 포함된 데이터가 7개(자기 자신은 P1, 이웃 데이터 P2, P6, P7, P8, P9, P11)로 최소 데이터 5개 이상을 만족하므로 P1 데이터는 핵심 포인트(Core Point)가 된다.



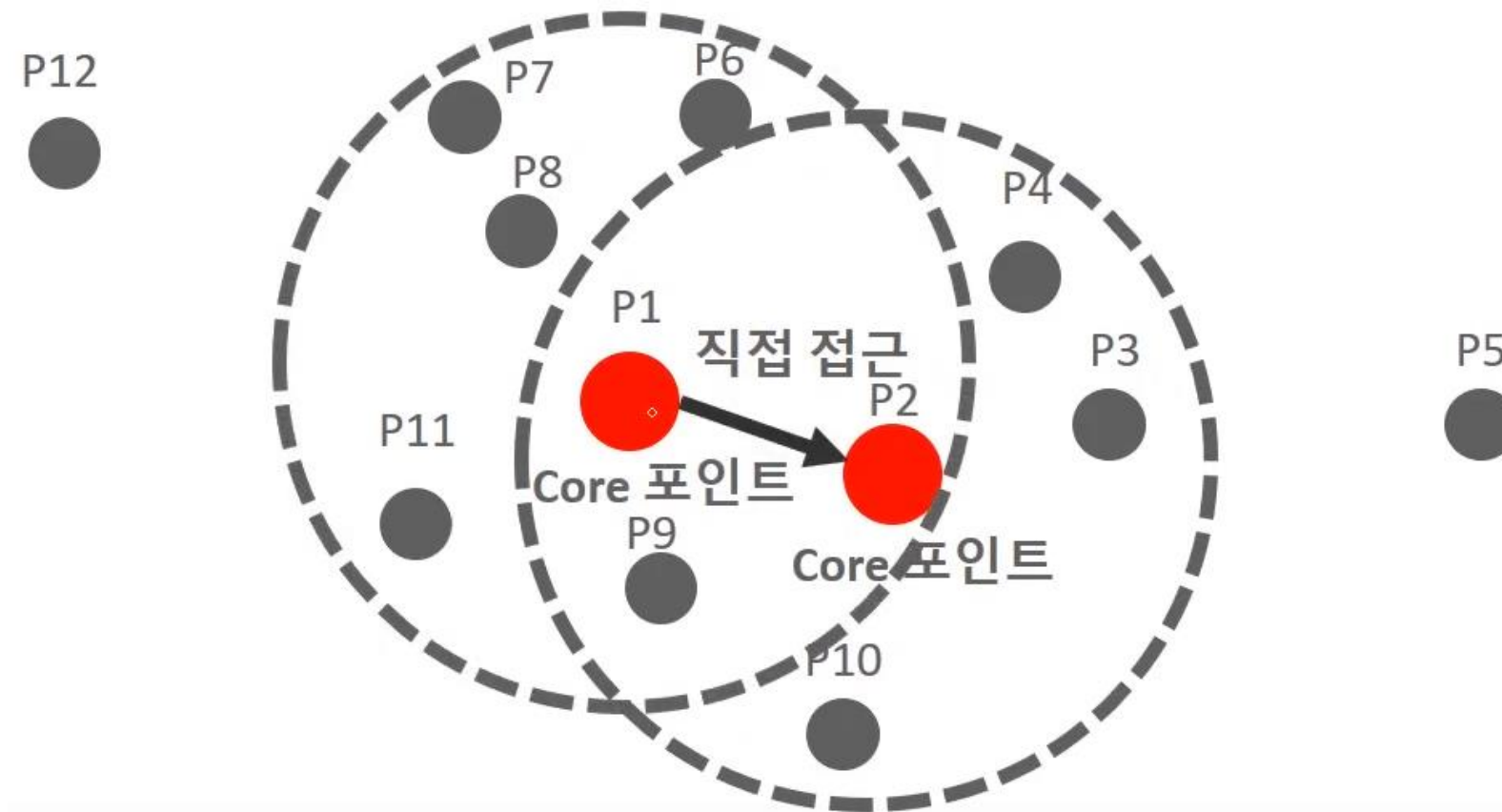
# DBSCAN 절차 3

- P2 데이터 포인트 역시 반경 내에 6개의 데이터(P2, P1, P3, P4, P9, P10)를 가지고 있으므로 핵심 포인트가 된다.



## DBSCAN 절차 4

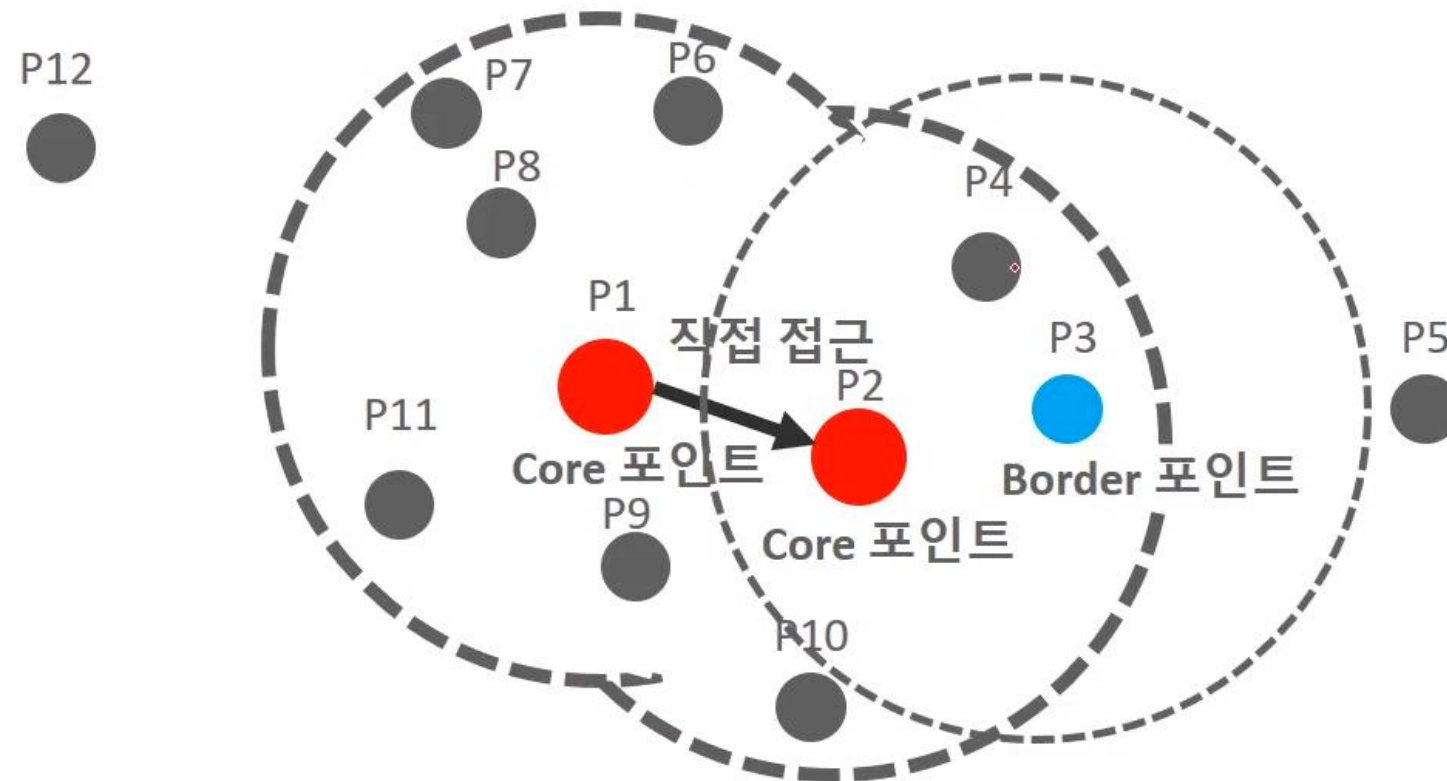
- 핵심 포인트 P1의 이웃 데이터 포인트 P2 역시 핵심 포인트일 경우 P1에서 P2로 연결해 직접 접근이 가능





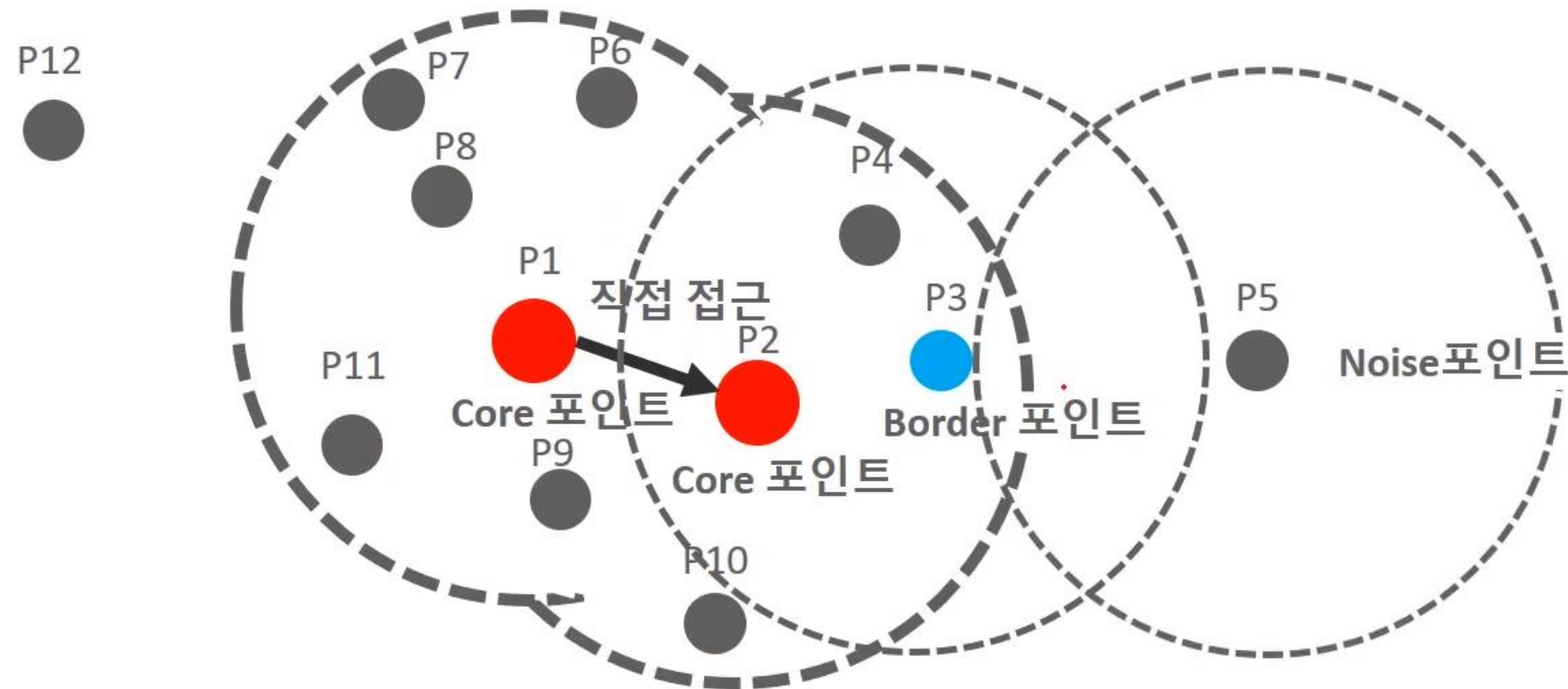
# DBSCAN 절차 6

- P3의 경우 반경 내에 포함되는 이웃 데이터는 P2, P4로 2개이므로 군집으로 구분할 수 있는 핵심 포인트가 될 수 없다.
- 하지만 이웃 데이터 중 핵심 포인트인 P2를 가지고 있는데, 이처럼 자신은 핵심 포인트가 아니지만 이웃 데이터로 핵심 포인트를 가지고 있는 데이터를 경계 포인트라고 한다. 경계 포인트는 군집의 외곽을 형성한다.



# DBSCAN 절차 7

- P5와 같이 반경 내에 최소 데이터를 가지고 있지 않고, 핵심 포인트 또한 이웃 데이터로 가지고 있지 않는 데이터를 잡음 포인트(Noise Point)라고 한다.



# 사이킷런 DBSCAN

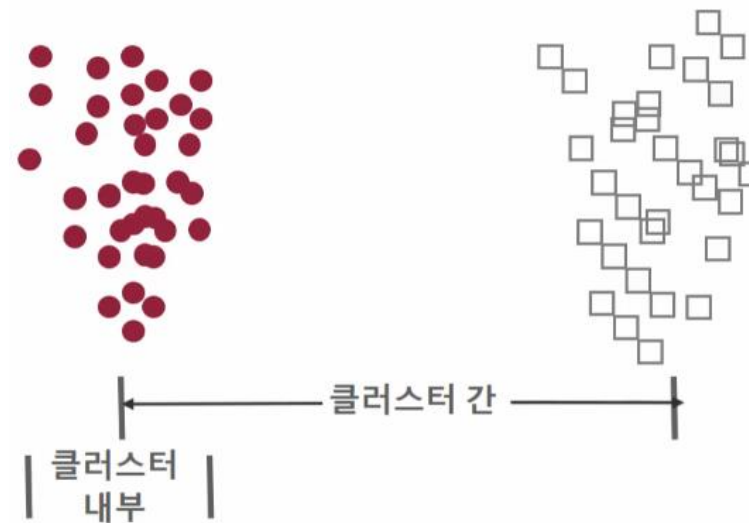
- 사이킷런은 DBSCAN 클래스를 통해 DBSCAN 알고리즘을 지원한다.
- 주요 하이퍼 파라미터는 `eps`, `min_samples` 이다.
  - `eps` : 입실론 주변 영역의 반경을 의미한다.
  - `min_samples` : 핵심 포인트가 되기 위해 입실론 주변 영역 내 포함되어야 할 데이터의 최소 개수를 의미한다.
    - 자기 자신의 데이터를 포함해 `min_samples + 1`이 된다.



## 군집화 성능 평가

# 군집 평가. 실루엣 분석

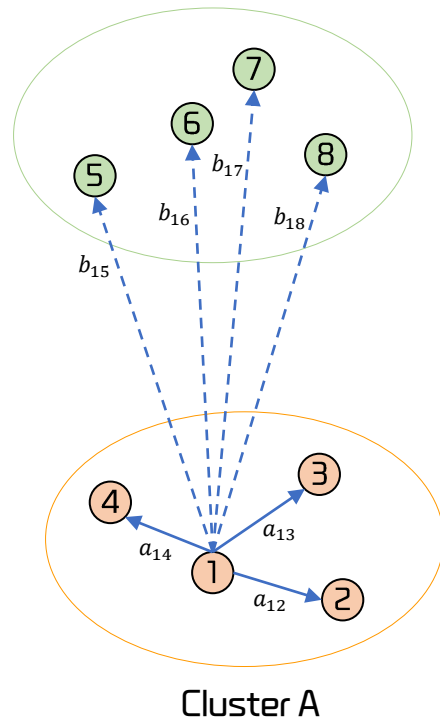
- 실루엣 분석은 각 군집 간의 거리가 얼마나 효율적으로 분리되어 있는지를 나타낸다.
- 실루엣 분석은 개별 데이터가 가지는 군집화 지표인 실루엣 계수(Silhouette Coefficient)를 기반으로 한다.
- 개별 데이터가 가지는 실루엣 계수는 해당 데이터가 같은 군집 내의 데이터와 얼마나 가깝게 군집되어 있고, 다른 군집에 있는 데이터와는 얼마나 멀리 분리되어 있는지를 나타내는 지표이다.



다른 군집과의 거리는 떨어져 있고 동일  
군집끼리의 데이터는 서로 가깝게

# 실루엣 계수(Silhouette Coefficient)

Cluster B  
(클러스터 A의 1번 데이터에서 가장 가까운 타 클러스터)



$$\text{실루엣 계수 } s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

- $a_{ij}$ 는  $i$ 번째 데이터에서 자신이 속한 클러스터 내의 다른 데이터 포인트 까지의 거리이다. 즉  $a_{12}$ 는 1번 데이터에서 2번 데이터 까지의 거리이다.
- $a_i$ 는  $i$ 번째 데이터에서 자신이 속한 클러스터 내의 다른 데이터 포인트들의 거리 평균.
  - $a_i = \text{평균}(a_{12}, a_{13}, a_{14})$
- $b_i$ 는  $i$ 번째 데이터에서 가장 가까운 타 클러스터 내의 다른 데이터 포인트들의 거리 평균.
  - $b_i = \text{평균}(b_{15}, b_{16}, b_{17}, b_{18})$
- 두 군집 간의 거리가 얼마나 떨어져 있는가의 값은  $b_i - a_i$ 이며 이 값을 정규화 하기 위해  $\max(a_i, b_i)$  값으로 나눈다.
- 실루엣 계수는 -1에서 1 사이의 값을 가지며, 1로 가까워질수록 근처의 군집과 더 멀리 떨어져 있다는 것이고, 0에 가까울수록 근처의 군집과 가까워진다는 것이다. 음수는 아예 다른 군집에 데이터 포인트가 할당 됐음을 의미한다.

# 사이킷런 실루엣 분석 API와 좋은 군집의 기준

- `sklearn.metrics.silhouette_samples(X, labels, metric='euclidean', **kwargs)`
  - 인자로 X Feature 데이터 세트와 각 피쳐 데이터 세트가 속한 군집 레이블 값인 labels 데이터를 입력해주면 각 데이터 포인트의 실루엣 계수를 반환한다.
- `sklearn.metrics.silhouette_score(X, labels, metric='euclidean', sample_size=None, **kwargs)`
  - 인자로 X Feature 데이터 세트와 각 피쳐 데이터 세트가 속한 군집 레이블 값인 labels 데이터를 입력해주면 전체 데이터의 실루엣 계수 값을 평균해 반환한다. 즉 `np.mean(silhouette_samples())`와 같다.
  - 일반적으로 이 값이 높을수록 군집화가 어느정도 잘 됐다고 판단할 수 있다. 하지만 무조건 이 값이 높다고 해서 군집화가 잘 됐다고 판단할 수는 없다.
- 실루엣 분석에 기반한 좋은 군집 기준은 `silhouette_score` 함수의 값이 0 ~ 1 사이의 값을 가질 때, 1에 가까울 수록 좋다.
- 하지만 전체 실루엣 계수의 평균값과 더불어 **개별 군집의 평균값의 편차가 크지 않아야 한다**. 즉 개별 군집의 실루엣 계수 평균값이 전체 실루엣 계수의 평균값에서 크게 벗어나지 않는 것이 중요하다. 만약 실루엣 계수의 평균값은 높지만, 특정 군집의 실루엣 계수 평균값만 유난히 높고, 다른 군집들의 실루엣 계수 평균값이 낮으면 좋은 군집화가 되지 못한다.