

Machine Learning

Clustering

Present by Sangmin Bae

Contents

1

K-Means

2

Mean Shift

3

DBSCAN

4

Gaussian
Mixture Model

5

Hierarchical
Clustering

Part One

K-Means

Clustering (1)

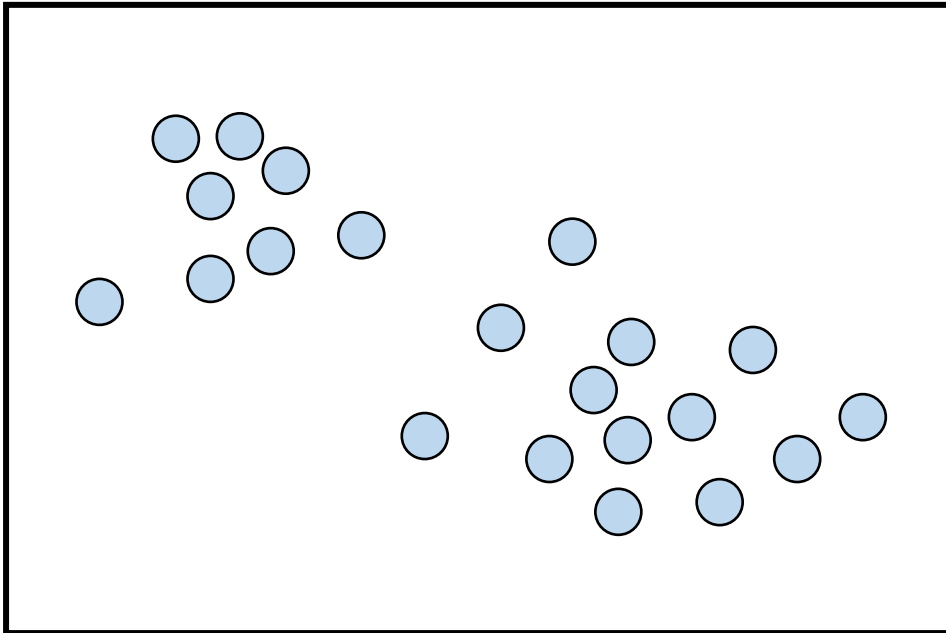
군집화 (clustering)

- 비지도 학습 상황에서, 데이터 샘플들을 별개의 군집(cluster)으로 그룹화 하는 것
- 비지도 학습에서의 분류 알고리즘
- 데이터의 특징에 따라 세분화하는데 사용
- 이상 검출 (anomaly detection)에 사용

Clustering (2)

군집화 (clustering)

- 유사성이 높은 데이터를 동일한 그룹으로 분류
- 서로 다른 군집은 특성이 상이하도록 군집화함
- 클러스터 내부의 분산(**within 분산**) 최소화, 클러스터 간의 분산(**between 분산**) 최대화



Parametric vs Non-Parametric

모수적 추정 vs 비모수적 추정

- **모수적(parametric) 추정 :**
 - 주어진 데이터가 특정 데이터 분포를 따른다고 가정
 - Gaussian Mixture Model(GMM)이 대표적
- **비모수적(non-parametric) 추정:**
 - 데이터가 특정 분포를 따르지 않는다는 가정 하에서 확률 밀도를 추정
 - K-means, Mean Shift, DBSCAN 등의 알고리즘이 있음

K-means Clustering (1)

K-means

- 군집의 **중심점(centroid)** 기반 클러스터링
- 샘플은 가장 가까운 중심점을 가진 군집으로 할당됨
- K-means 알고리즘은 사전에 군집의 수에 대한 하이퍼파라미터 k 를 정해야 사용 가능

$$X = C_1 \cup C_2 \cdots \cup C_k, \quad C_i \cap C_j = \emptyset$$

$$\arg \min_C \sum_{j=1}^k \sum_{x_i \in C_j} \|x_i - c_j\|^2$$

- **EM 알고리즘**을 통해 최적의 군집에 수렴할 때까지 학습함

K-means Clustering (2)

기댓값 최대화 알고리즘 (EM algorithm)

- 최대가능도(maximum likelihood)나 최대사후확률(maximum a posteriori)을 갖는 모수의 추정값을 찾는 반복적인 알고리즘
- EM 알고리즘은 **Expectation 단계**와 **Maximization 단계**로 나뉨
 1. **Expectation(기댓값) 단계:**
현재의 추정된 모수를 통해 샘플을 군집에 할당하는 단계
 2. **Maximization(최대화) 단계:**
로그 가능도(likelihood)의 기댓값을 최대화하는 모수를 추정하는 단계

K-means Clustering (3)

기댓값 최대화 알고리즘 (EM algorithm)

- 특정 분포에 대한 가정이 없는 Non-parametric 추정에서는 가능도의 개념이 없음
- Mean Shift나 DBSCAN은 밀도 추정의 방법으로 학습
- K-means 군집화에서의 EM 알고리즘은,
 - Expectation 단계: 추정하고자 하는 모수는 중심점(centroid)이므로, 샘플을 군집으로 할당하는 단계
 - Maximization 단계: 가능도를 샘플이 군집에 속할 확률로 해석하여, 군집에 할당된 샘플을 바탕으로 새로운 중심점을 계산

Examples of K-means Clustering (1)

EM algorithm in K-means

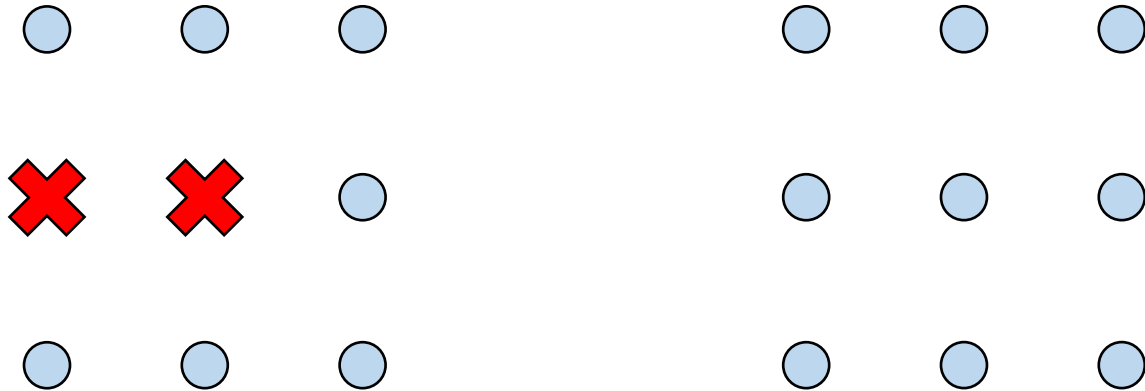
- 군집 수 k 가 2인 상황



Examples of K-means Clustering (2)

EM algorithm in K-means

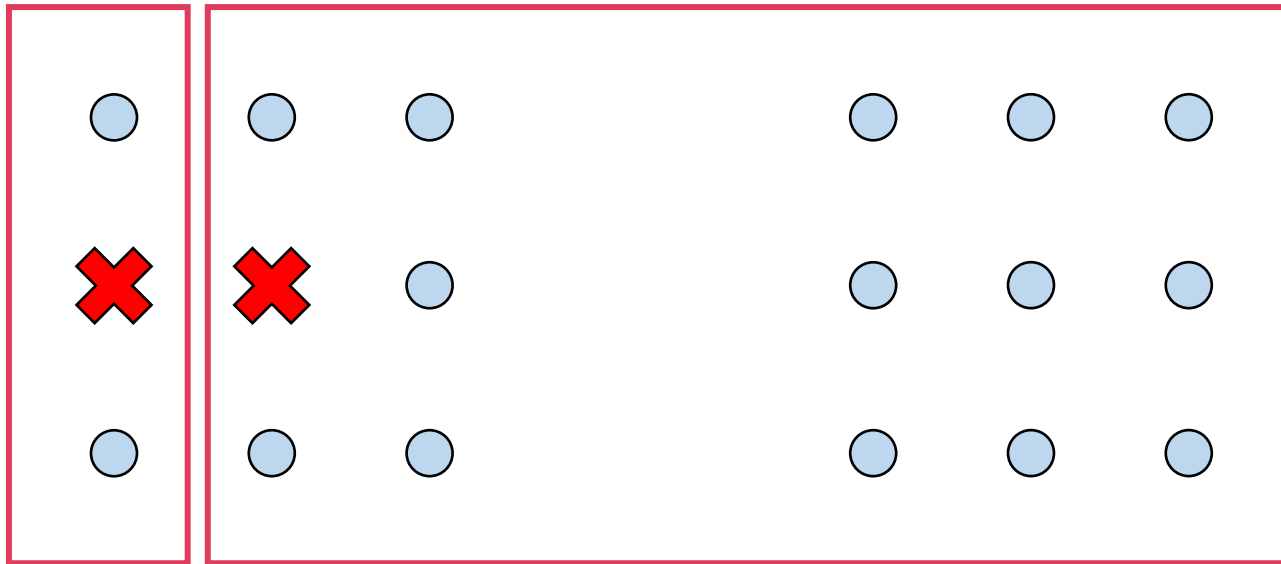
- 처음 군집의 centroid는 랜덤하게 설정함



Examples of K-means Clustering (3)

EM algorithm in K-means

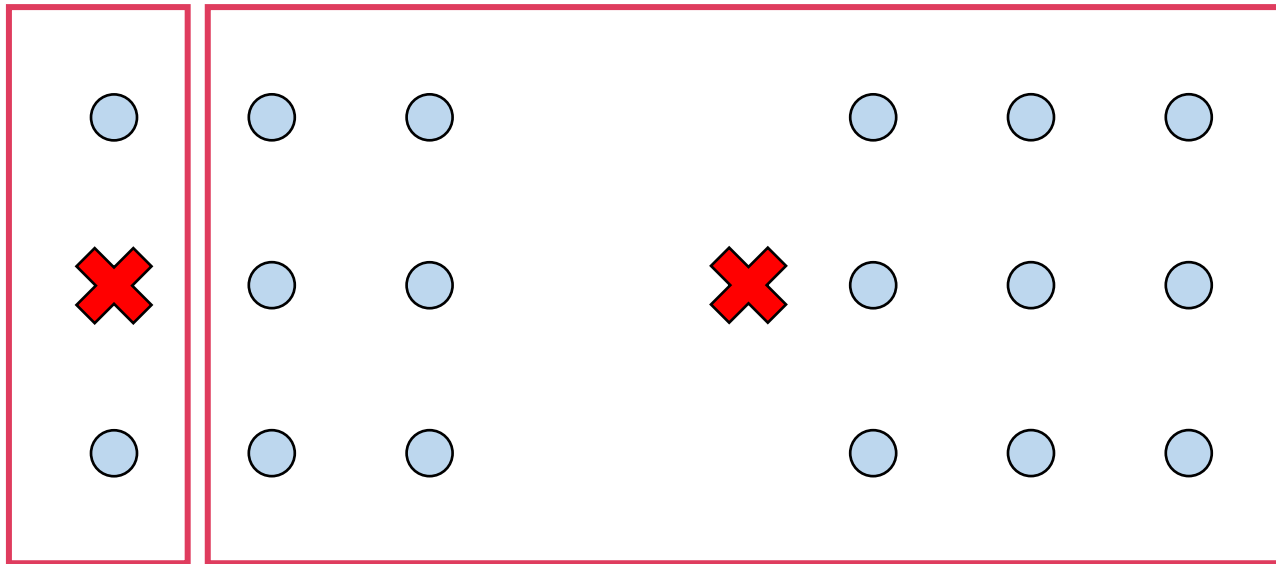
- 샘플들을 **가장 가까운 Centroid에 할당**해 군집을 생성 (Expectation 단계)



Examples of K-means Clustering (4)

EM algorithm in K-means

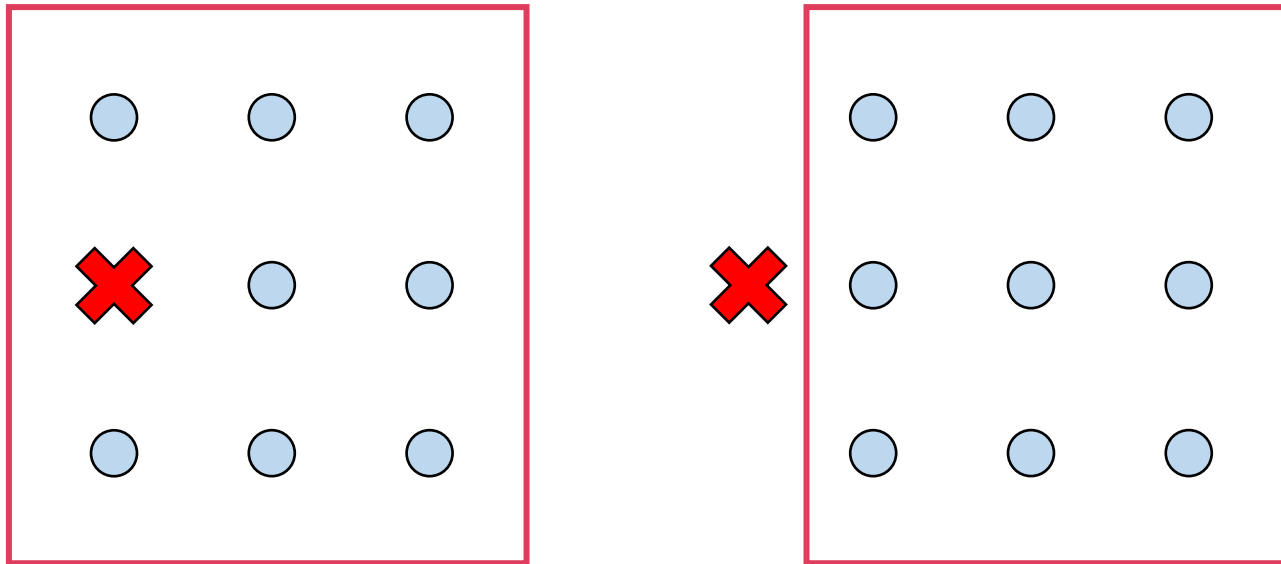
- 군집의 새로운 centroid를 계산 (Maximization 단계)



Examples of K-means Clustering (5)

EM algorithm in K-means

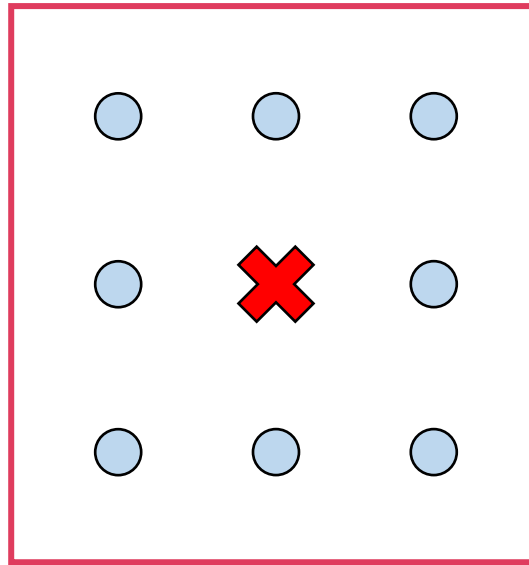
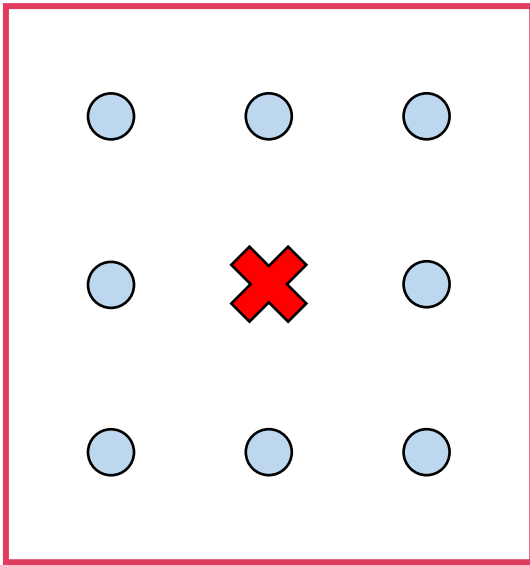
- 샘플들을 **가장 가까운 Centroid에 할당**해 군집을 생성 (Expectation 단계)



Examples of K-means Clustering (6)

EM algorithm in K-means

- 군집의 새로운 centroid를 계산 (Maximization 단계)



Evaluation Metrics of Clustering (1)

실루엣 분석 (silhouette analysis)

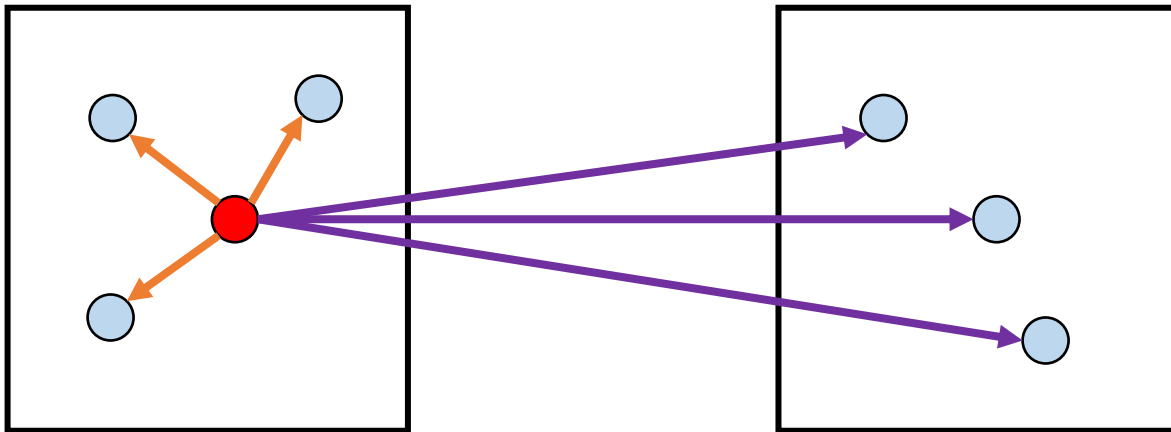
- 군집들이 얼마나 효율적으로 분리되어 있는지를 보여줌
- 각 샘플들이 가지고 있는 실루엣 계수(silhouette coefficient)를 기반으로 함
- 전체 실루엣 계수의 평균값이 클수록, 개별 군집의 평균값의 편차가 작을수록 좋음

Evaluation Metrics of Clustering (2)

실루엣 계수 (silhouette coefficient)

- 가장 가까운 타 군집 하나와의 거리를 통해 계산
- 실루엣 계수는 -1 ~ 1 사이의 값을 가지며, 1에 가까울수록 더 좋은 군집을 의미

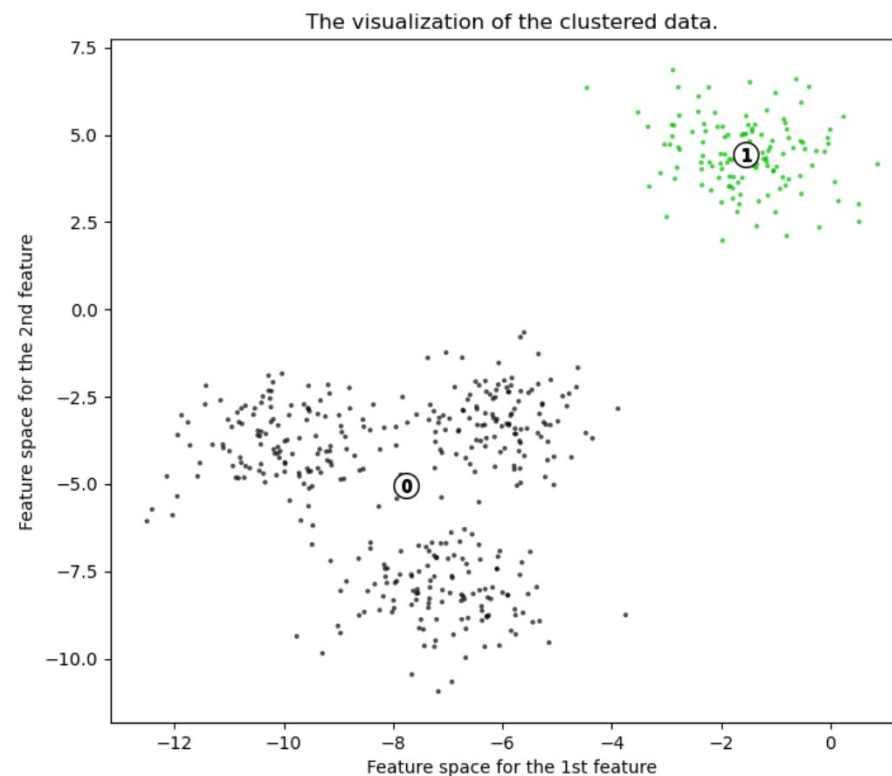
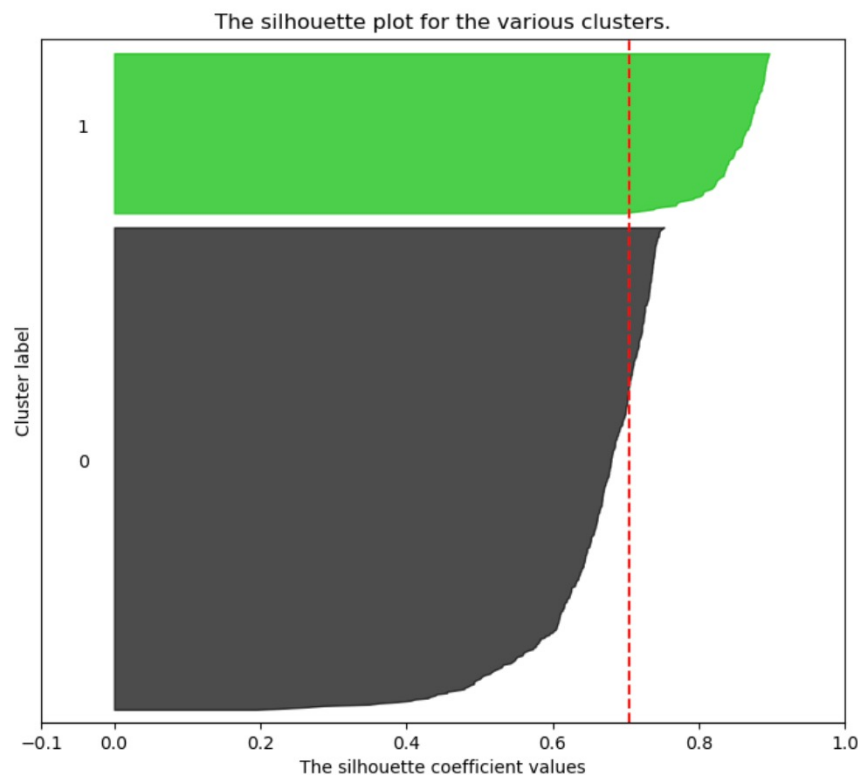
$$s(i) = \frac{\bar{b}_i - \bar{a}_i}{\max(\bar{a}_i, \bar{b}_i)}$$



Evaluation Metrics of Clustering (3)

실루엣 분석 (silhouette analysis)

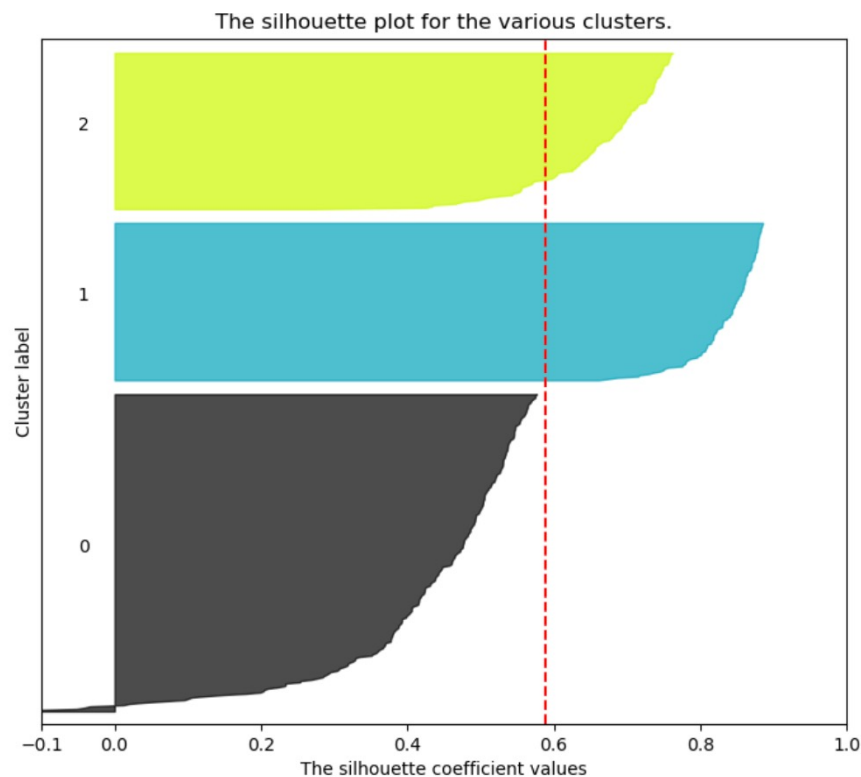
- 실루엣 계수의 평균값이 크고 편차가 작도록, K-means 알고리즘의 군집 개수(k)를 결정함



Evaluation Metrics of Clustering (4)

실루엣 분석 (silhouette analysis)

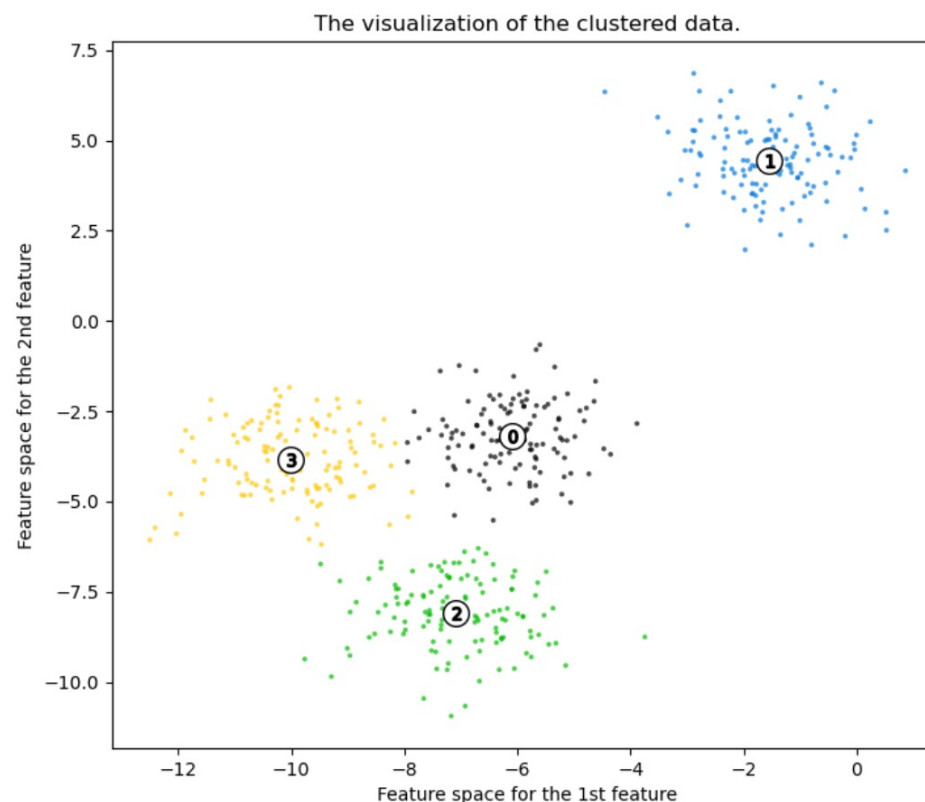
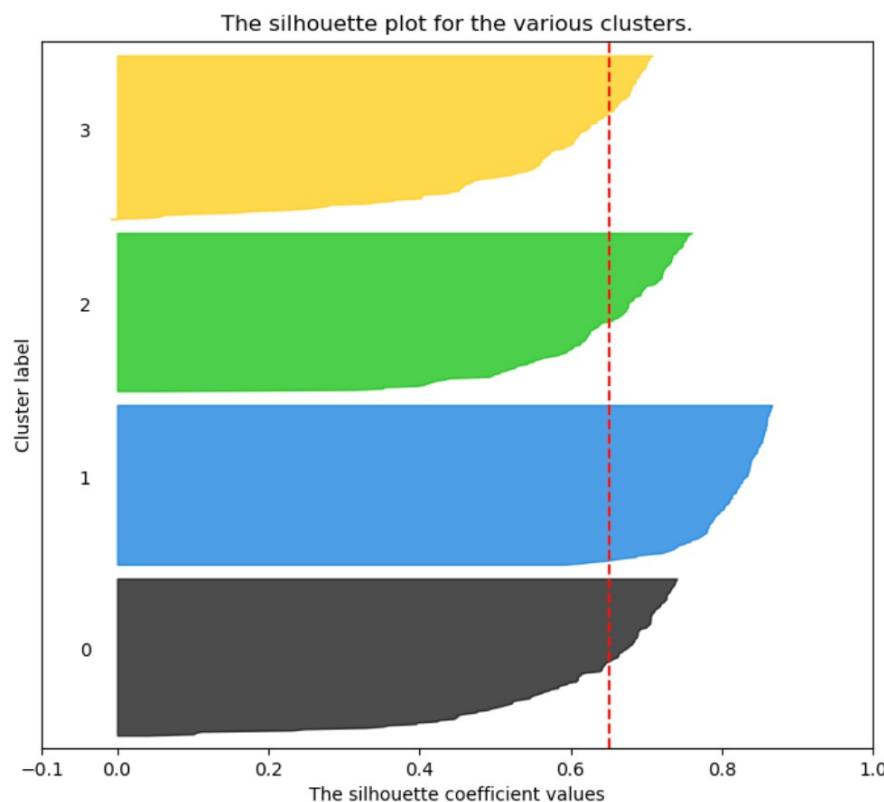
- 실루엣 계수의 평균값이 크고 편차가 작도록, K-means 알고리즘의 군집 개수(k)를 결정함



Evaluation Metrics of Clustering (5)

실루엣 분석 (silhouette analysis)

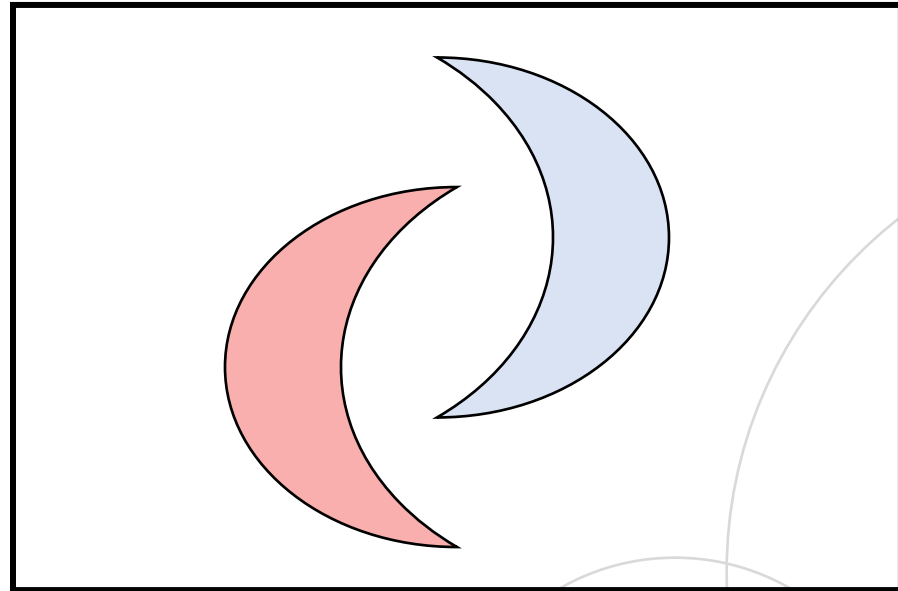
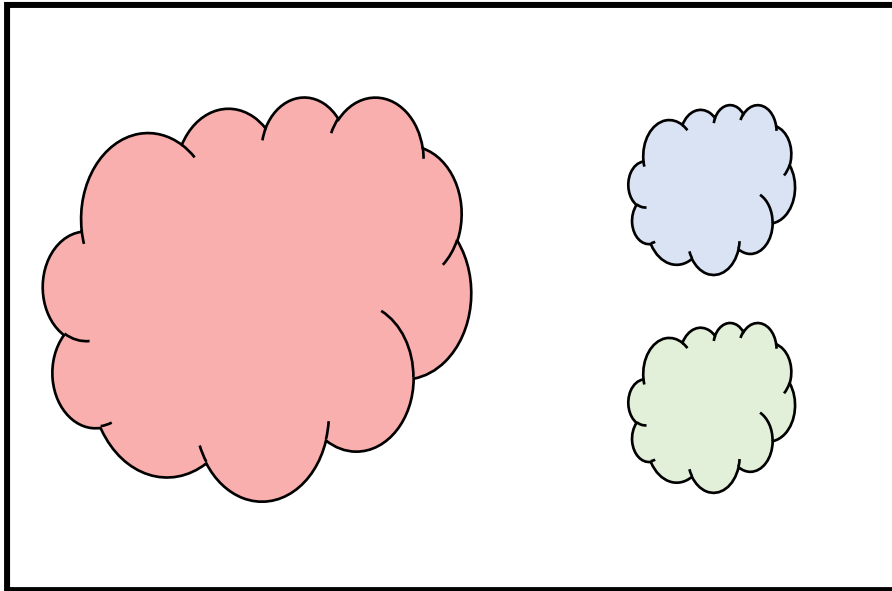
- 실루엣 계수의 평균값이 크고 편차가 작도록, K-means 알고리즘의 군집 개수(k)를 결정함

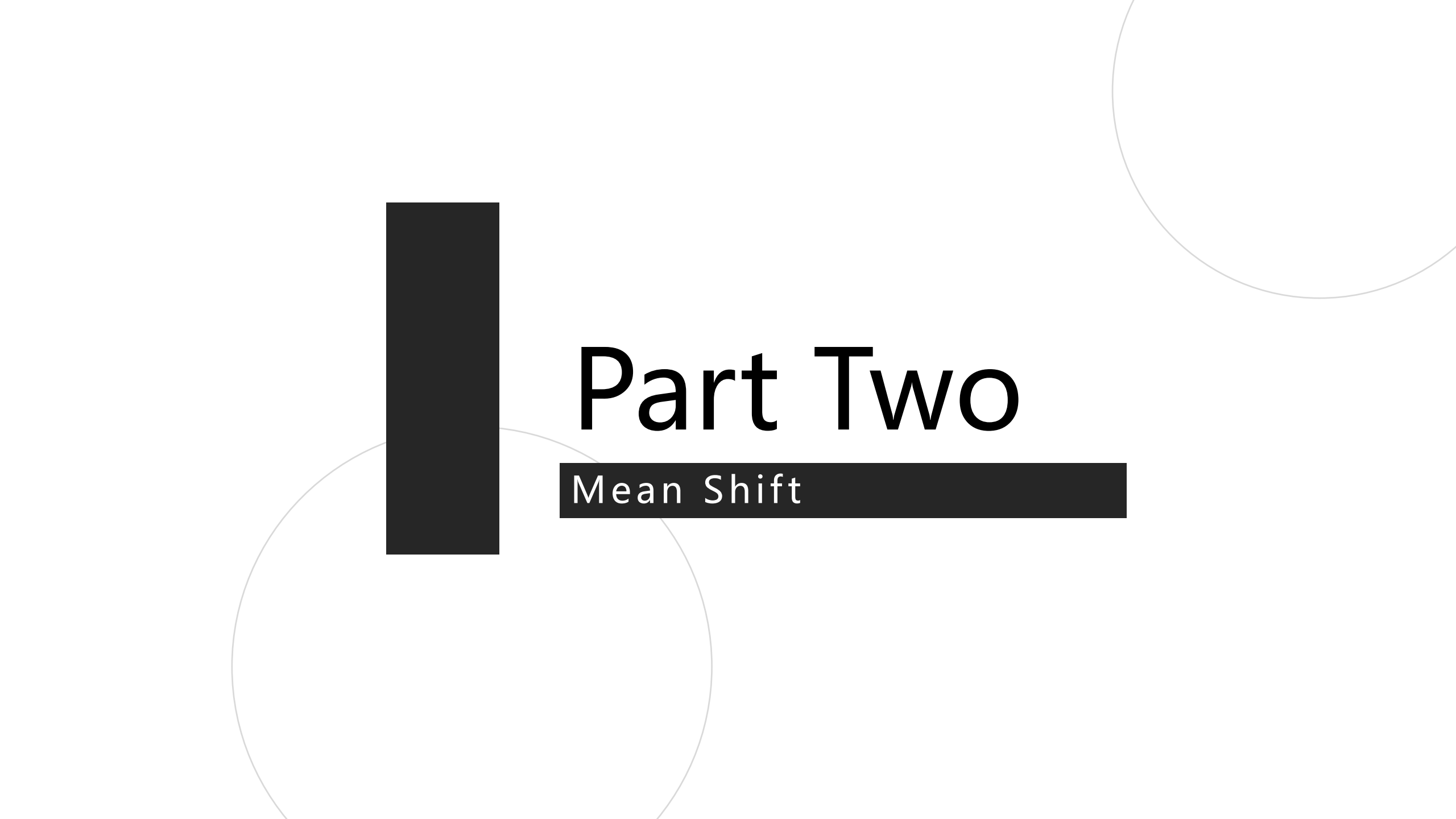


Limitation of K-means Clustering

K-means 한계점

- 군집의 개수, centroid에 대한 초기 설정값에 따라 성능 편차가 심함
- 군집 크기나 밀도가 다를 경우, 학습이 잘 안될 수가 있음
- 데이터 분포가 특이할 경우에도 군집 학습이 어려움





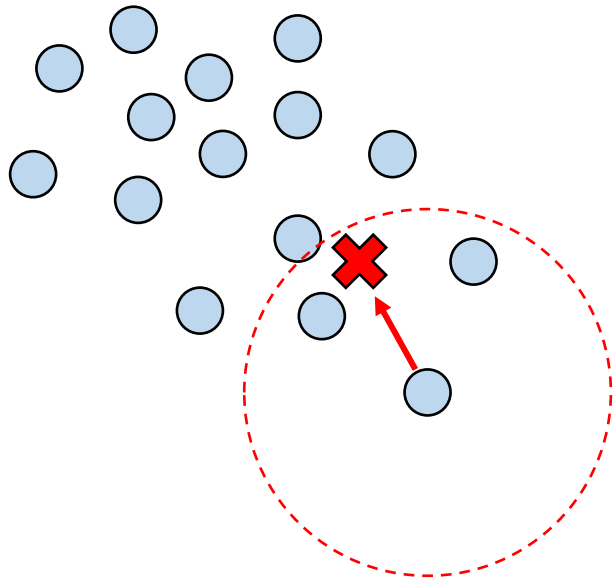
Part Two

Mean Shift

Mean Shift (1)

평균 이동 (mean shift)

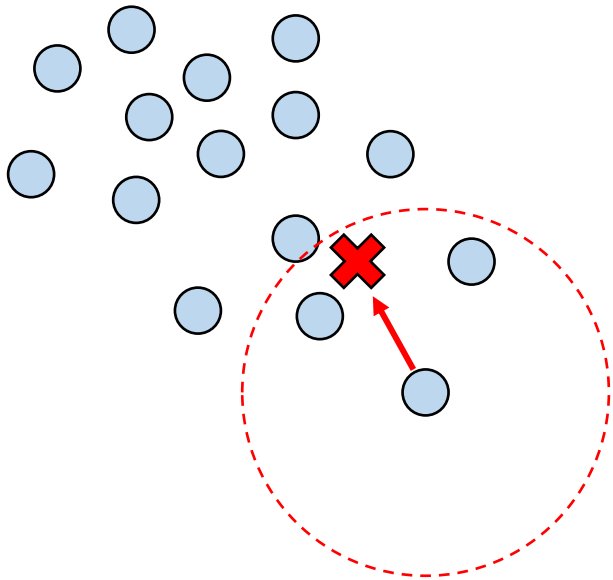
- 각 샘플을 기점으로 **주변에 데이터가 가장 밀집된 곳으로 이동**하는 것을 수렴할 때까지 반복
- 모든 데이터에 대해 수렴 지점을 계산하여, 군집의 개수를 결정
- 각 샘플들을 가장 가까운 중심점을 가진 군집으로 할당



Mean Shift (2)

평균 이동 (mean shift)

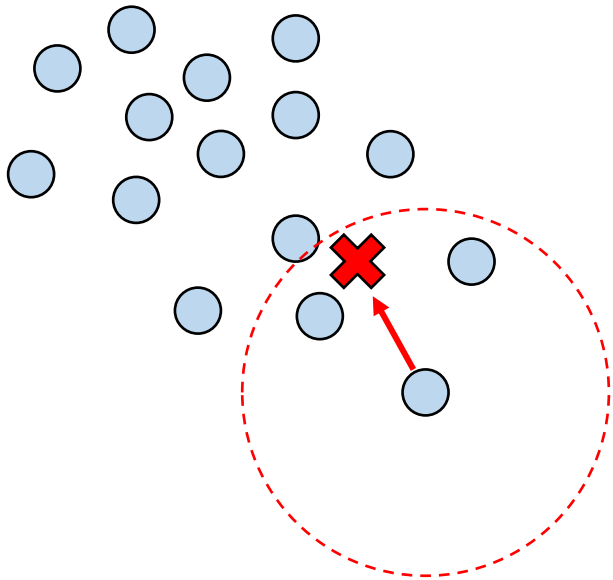
- K-means 알고리즘과 다르게 군집 개수에 대한 하이퍼파라미터가 불필요
- **Sliding window의 크기**를 조절해 주변 어느 정도까지 볼 지 결정해야함



Mean Shift (3)

평균 이동 (mean shift)

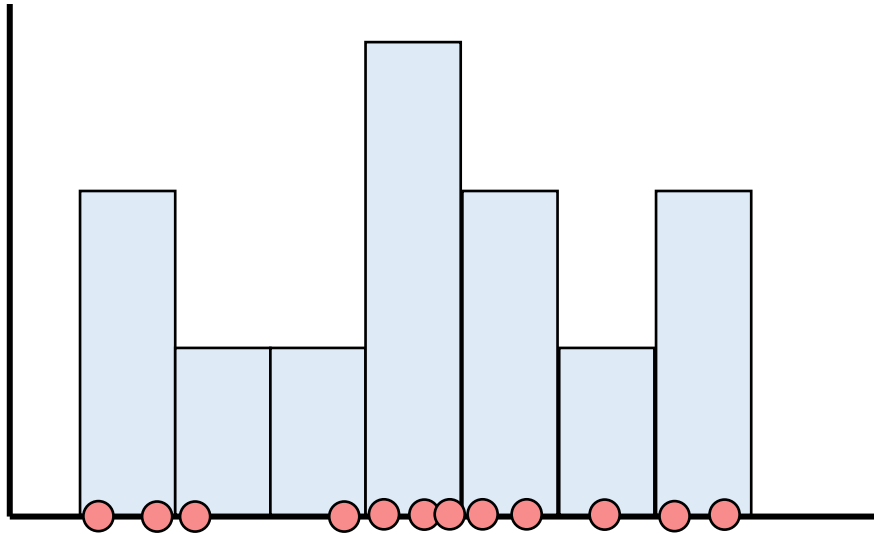
- 비모수(non-parametric) 방법의 모델
- KDE(kernel density estimation)를 통해 밀도가 가장 높은 곳을 찾아냄



Histogram

히스토그램 (histogram)

- 비모수적 밀도 추정을 위해 간단하게 **히스토그램**을 사용할 수 있음
- 하지만, Bin의 경계때문에 **불연속적인** 특징이 있음



Kernel Density Estimation (1)

KDE (kernel density estimation)

- 커널 함수를 통해 어떤 변수의 확률 밀도 함수를 추정하는 방법
- 개별 샘플들에 커널 함수를 적용한 값을 모두 합한 뒤, 데이터 개수로 나누어 확률 밀도 함수를 추정
- $KDE = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right)$
- h 는 커널 함수의 bandwidth 파라미터로, 뾰족한 형태 혹은 완만한 형태일지 결정

Kernel Density Estimation (2)

KDE (kernel density estimation)

- $KDE = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right)$
- 대표적인 커널 함수로 Gaussian 분포 함수가 사용됨

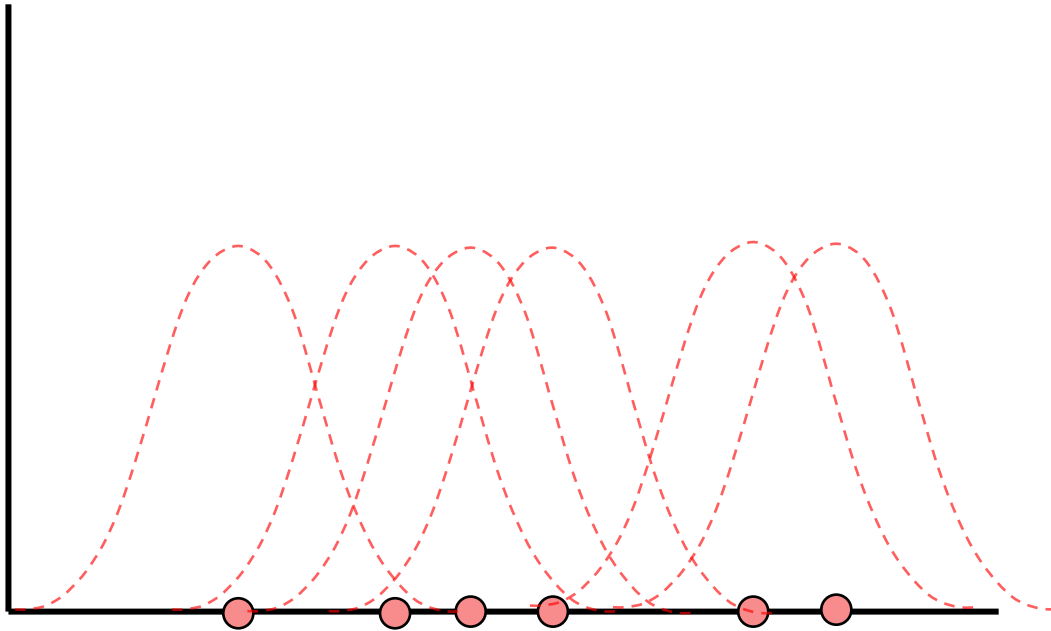
$$KDE = \frac{1}{nh} \sum_{i=1}^n \frac{1}{\sqrt{2\pi}h} e^{\left(-\frac{1}{2}\left(\frac{x-x_i}{h}\right)^2\right)}$$

- 평균은 관측값 x_i 로, 표준편차는 h 로 설정

Kernel Density Estimation (3)

KDE (kernel density estimation)

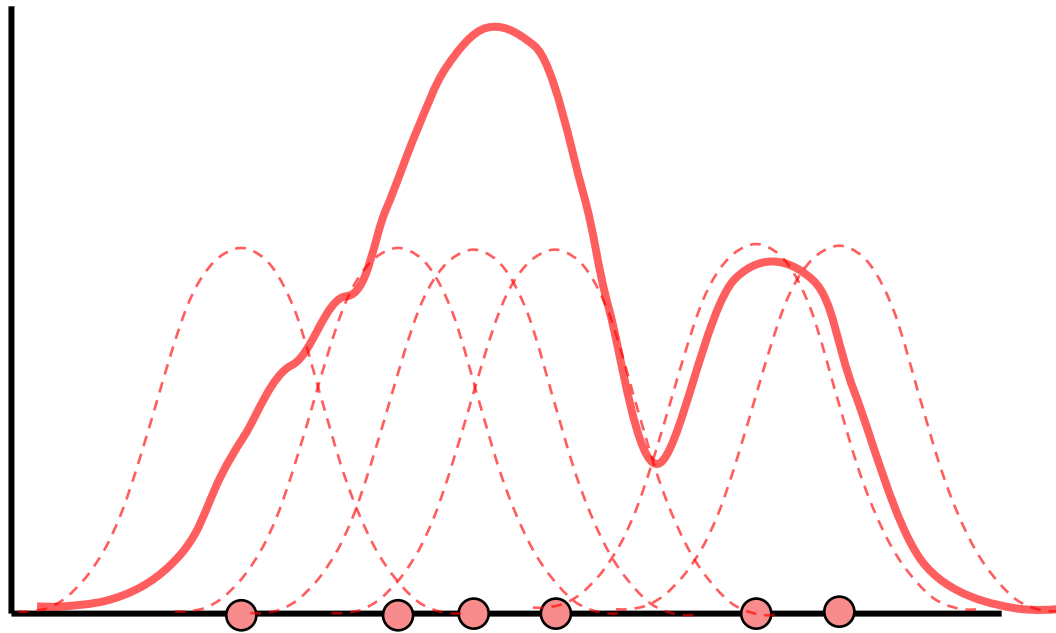
- 개별 샘플들에 커널 함수를 적용한 값을 모두 합한 뒤, 데이터 개수로 나누어 확률 밀도 함수를 추정



Kernel Density Estimation (4)

KDE (kernel density estimation)

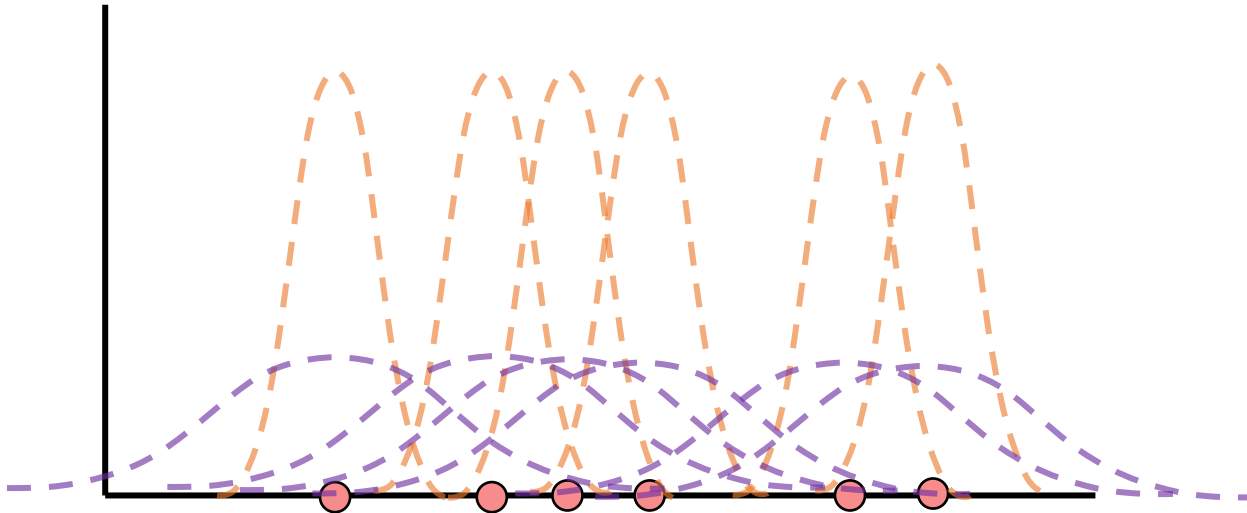
- 개별 샘플들에 커널 함수를 적용한 값을 모두 합한 뒤, 데이터 개수로 나누어 확률 밀도 함수를 추정



Kernel Density Estimation (5)

KDE (kernel density estimation)

- h 값이 작을수록 뾰족한 Gaussian 커널 함수가 생성
-> overfitting 문제 (군집 개수 \uparrow)
- h 값이 크면 편차가 큰 완만한 커널 함수가 생성
-> underfitting 문제 (군집 개수 \downarrow)



Kernel Density Estimation (6)

KDE (kernel density estimation)

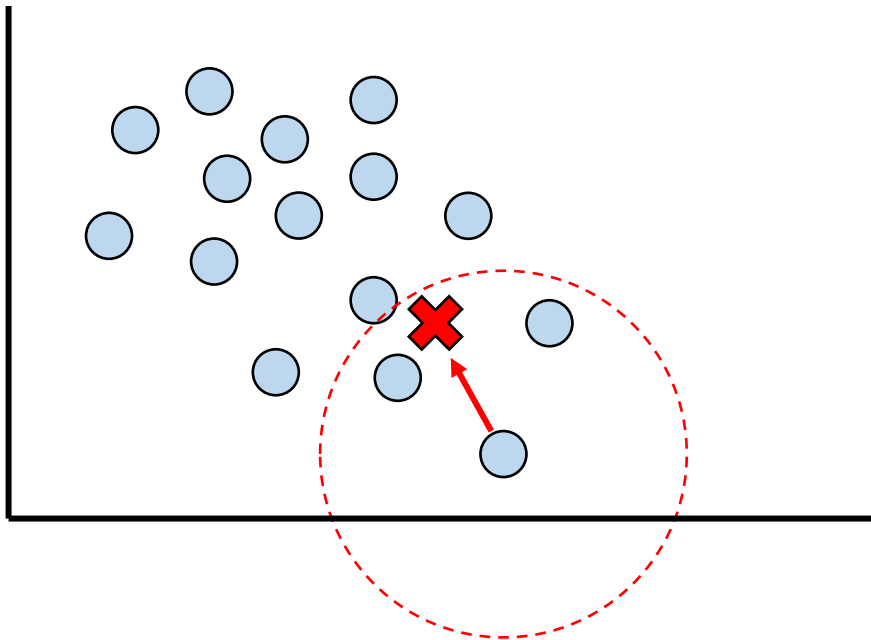
- h 하이퍼파라미터에 대한 최적의 값을 찾아야 함
- 가우시안 커널 함수를 사용할 때의 최적의 bandwidth는 다음과 같음

$$h = \left(\frac{4\sigma^5}{3n} \right)^{\frac{1}{5}} \approx 1.06\sigma n^{-\frac{1}{5}}$$

Kernel Density Estimation (7)

KDE (kernel density estimation)

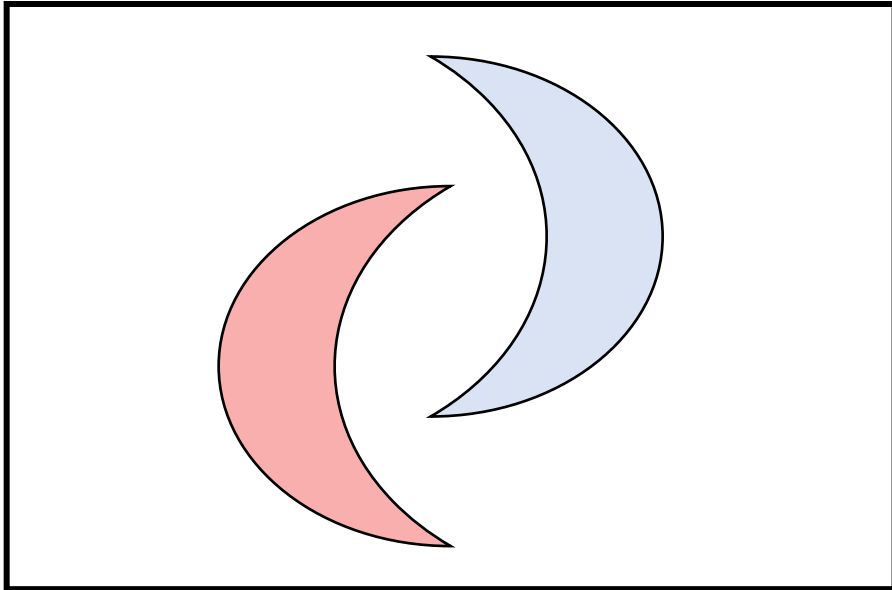
- 2차원 데이터에 커널 함수는 어떻게 적용할까?
- 각 차원으로 데이터를 투영시켜, 커널 함수를 적용해 밀도가 가장 높은 좌표 계산




Limitation of Mean Shift

Mean Shift 한계점

- Sliding window의 크기와 bandwidth h 에 대한 선택이 필요함
- 여전히 데이터 분포가 특이할 경우에 군집 학습이 어려움





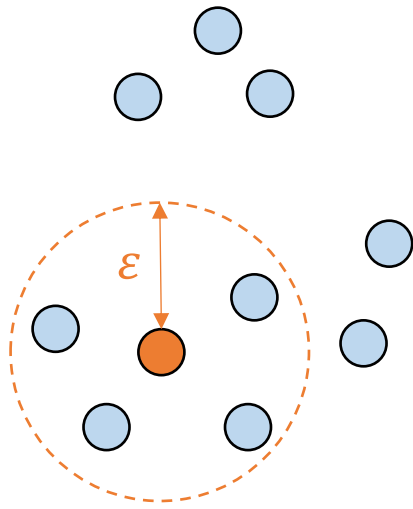
Part Three

DBSCAN

DBSCAN (1)

DBSCAN (density-based spatial clustering of applications with noise)

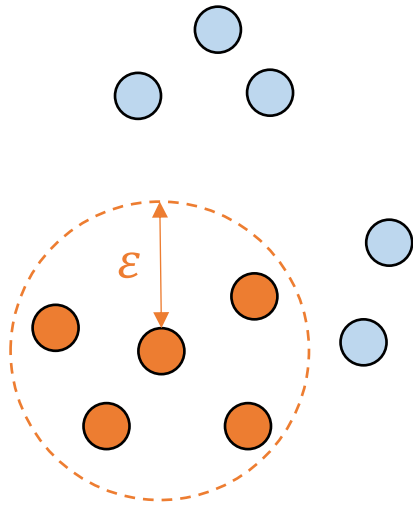
- DBSCAN 또한 밀도가 높은 부분을 중심으로 군집화를 하는 방법론
- 어떤 점을 기준으로 반경 ϵ 내에 샘플이 *minpoints*보다 많으면 같은 군집으로 할당



DBSCAN (2)

DBSCAN (density-based spatial clustering of applications with noise)

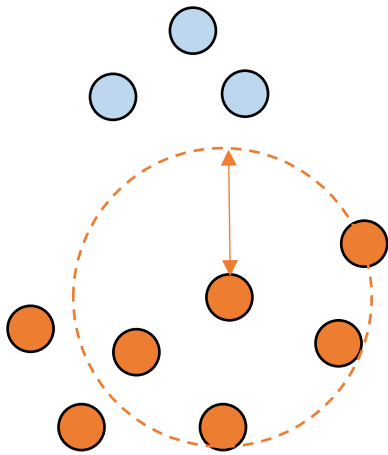
- 어떤 점을 기준으로 반경 ϵ 내에 샘플이 *minpoints* = 3보다 많으면 같은 군집으로 할당



DBSCAN (3)

DBSCAN (density-based spatial clustering of applications with noise)

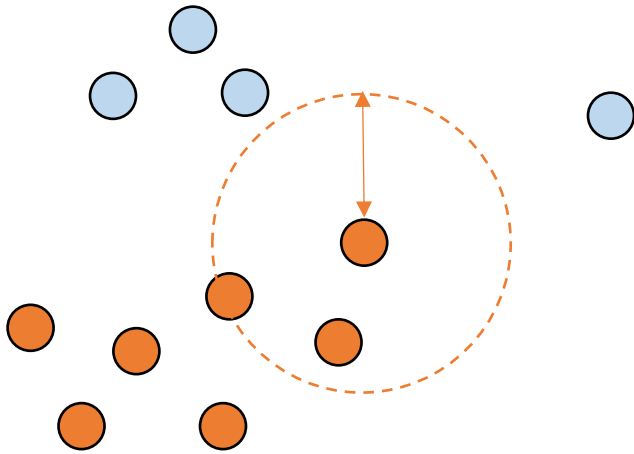
- 어떤 점을 기준으로 반경 ε 내에 샘플이 *minpoints* = 3보다 많으면 같은 군집으로 할당
- 군집으로 할당된 샘플들을 해당 군집의 **corepoint**로 설정해 계속 반복



DBSCAN (4)

DBSCAN (density-based spatial clustering of applications with noise)

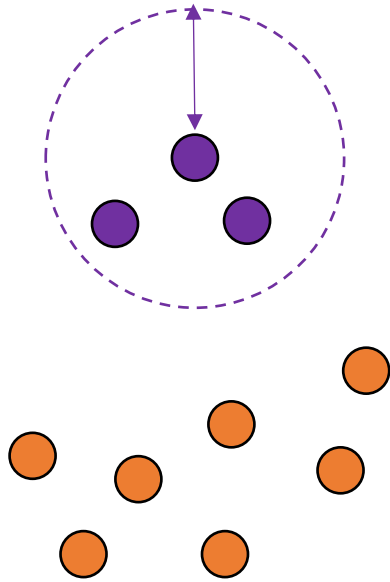
- 어떤 점을 기준으로 반경 ε 내에 샘플이 *minpoints* = 3보다 많으면 같은 군집으로 할당
- *minpoints* 개수를 만족 못하는 **borderpoint** 샘플(군집으로 할당은 됐지만, corepoint가 될 수 없는 샘플)이 나올 경우 멈춤



DBSCAN (5)

DBSCAN (density-based spatial clustering of applications with noise)

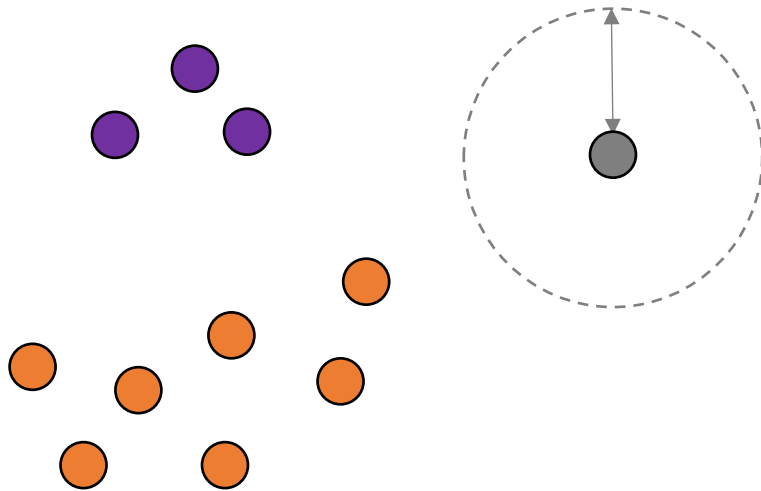
- 어떤 점을 기준으로 반경 ε 내에 샘플이 *minpoints* = 3보다 많으면 같은 군집으로 할당
- 이를 모든 데이터 샘플에 대해 진행하여 **cluster point**와 **noise point**를 구분



DBSCAN (6)

DBSCAN (density-based spatial clustering of applications with noise)

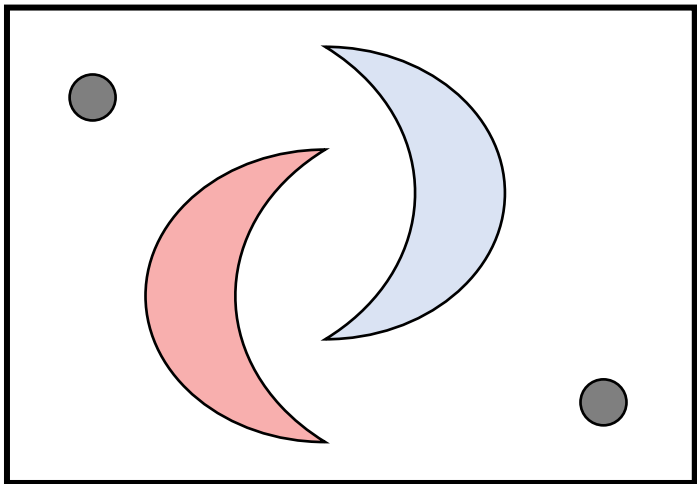
- 어떤 점을 기준으로 반경 ε 내에 샘플이 *minpoints* = 3보다 많으면 같은 군집으로 할당
- 이를 모든 데이터 샘플에 대해 진행하여 **cluster point**와 **noise point**를 구분



Pros and Cons of DBSCAN

DBSCAN 장점과 단점

- (장점) 다양한 형태의 군집 유형을 구분할 수 있음
- (장점) 아웃라이어(noise point)를 찾아낼 수 있음
- (단점) 군집의 개수 설정에선 자유롭지만, ϵ 과 *minpoints*에 대한 설정이 필요
- (단점) 연산량이 크기 때문에 시간이 오래 걸림





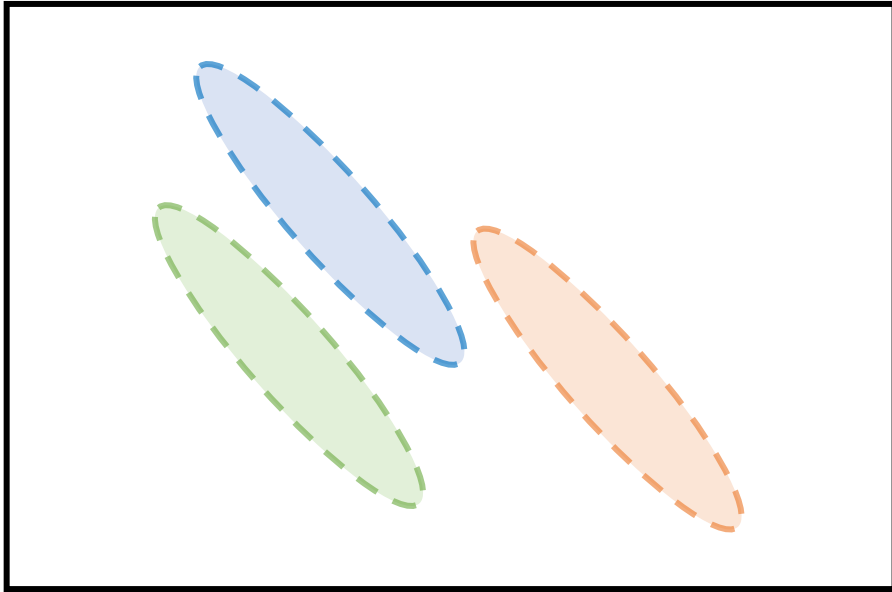
Part Four

Gaussian Mixture Model

Gaussian Mixture Model (1)

GMM (gaussian mixture model)

- 모수적 추정의 방법론으로, 주어진 데이터를 K개의 Gaussian 분포의 혼합으로 가정
- EM 알고리즘을 통해 모델을 학습함



Gaussian Mixture Model (2)

GMM (gaussian mixture model)

- LDA에서의 베이지스 분류기와 매우 비슷
- 다만, 비지도 학습이기 때문에 Y 라벨을 Z 클러스터 군집으로 대체하여 표현

$$\begin{aligned} P(Z = k | X = x) &= \frac{P(X = x | Z = k)P(Z = k)}{\sum_{j=1}^K P(X = x | Z = j)P(Z = j)} \\ &= \frac{\pi_k f_k(x)}{\sum_{j=1}^K \pi_j f_j(x)} = \frac{\pi_k N(x | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x | \mu_j, \Sigma_j)} \end{aligned}$$

Gaussian Mixture Model (3)

GMM (gaussian mixture model)

- 비지도 학습이기 때문에 LDA와 다르게 μ_k, Σ_k 뿐만 아니라, π_k 에 대한 모수 추정 필요
- EM 알고리즘은 **Expectation 단계**와 **Maximization 단계**로 나뉨
 1. Expectation(기댓값) 단계:
현재의 추정된 모수를 통해 샘플을 군집에 할당하는 단계
 2. Maximization(최대화) 단계:
로그 가능도(likelihood)의 기댓값을 최대화하는 모수를 추정하는 단계

Expectation in GMM

Expectation in GMM

- 현재 추정된 모수를 통해 샘플을 군집에 할당하는 단계
- **Responsibility(책임값)**를 계산하여, 샘플마다 가장 큰 값을 도출하는 군집으로 할당

$$\arg \max_k \gamma(z_{ik}) = \arg \max_k p(z_k = 1 | x_i) = \frac{\pi_k N(x_i | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x_i | \mu_j, \Sigma_j)}$$

Maximization in GMM (1)

Maximization in GMM

- 로그 가능도(likelihood)의 기댓값을 최대화하는 모수를 추정하는 단계
- 먼저 GMM의 우도 확률(가능도)을 $p(X|\pi, \mu, \Sigma)$ 로 정의
- 단조 증가 함수의 로그 함수를 사용해 로그 가능도 함수를 정의

$$\ln p(X|\pi, \mu, \Sigma) = \ln \left\{ \prod_{i=1}^N p(x_i|\pi, \mu, \Sigma) \right\} = \sum_{i=1}^N \ln p(x_i|\pi, \mu, \Sigma)$$

Maximization in GMM (2)

Maximization in GMM

- 클러스터 변수 z_k 와 marginal 확률을 이용해 로그 가능도 정의

$$\sum_{i=1}^N \ln p(x_i | \pi, \mu, \Sigma) = \sum_{i=1}^N \ln \sum_{k=1}^K p(x_i, z_k | \pi, \mu, \Sigma)$$

- $p(x, z) = p(z) p(x|z)$ 의 성질을 이용해, 다음과 같이 도출

$$\sum_{i=1}^N \ln \sum_{k=1}^K p(z_k | \pi, \mu, \Sigma) p(x_i | z_k, \pi, \mu, \Sigma) = \sum_{i=1}^N \ln \sum_{k=1}^K \pi_k N(x_i | \mu_k, \Sigma_k)$$

Maximization in GMM (3)

Maximization in GMM

- 각 파라미터에 대해 **편미분했을 때 0이 되는 지점**을 찾음

$$\begin{aligned} L &= \sum_{i=1}^N \ln \sum_{k=1}^K \pi_k N(x_i | \mu_k, \Sigma_k) \\ &= \sum_{i=1}^N \ln \sum_{k=1}^K \pi_k \frac{1}{(2\pi)^{p/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)} \end{aligned}$$

Maximization in GMM (4)

Maximization in GMM

- 각 파라미터에 대해 **편미분했을 때 0이 되는 지점**을 찾음

$$\frac{\partial L}{\partial \mu_k} = \sum_{i=1}^N \frac{\pi_k N(x_i | \mu_k, \Sigma_k)}{\sum_{k=1}^K \pi_k N(x_i | \mu_k, \Sigma_k)} \Sigma_k^{-1} (x_i - \mu_k) = 0$$

$$\Rightarrow \sum_{i=1}^N \gamma(z_{ik})(x_i - \mu_k) = 0$$

$$\Rightarrow \mu_k = \frac{\sum_{i=1}^N \gamma(z_{ik})x_i}{\sum_{i=1}^N \gamma(z_{ik})}$$

Maximization in GMM (5)

Maximization in GMM

- 각 파라미터에 대해 **편미분했을 때 0이 되는 지점**을 찾음

$$\frac{\partial L}{\partial \Sigma_k} = \sum_{i=1}^N \frac{\pi_k N(x_i | \mu_k, \Sigma_k)}{\sum_{k=1}^K \pi_k N(x_i | \mu_k, \Sigma_k)} \left\{ \frac{1}{2} \Sigma_k^{-1} (x_i - \mu_k)(x_i - \mu_k)^T \Sigma_k^{-1} - \frac{1}{2} \Sigma_k^{-1} \right\} = 0$$

$$\Rightarrow \sum_{i=1}^N \gamma(z_{ik}) (\Sigma_k^{-1} (x_i - \mu_k)(x_i - \mu_k)^T - 1) = 0$$

$$\Rightarrow \Sigma_k = \frac{\sum_{i=1}^N \gamma(z_{ik}) (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^N \gamma(z_{ik})}$$

Maximization in GMM (5)

Maximization in GMM

- 각 파라미터에 대해 **편미분했을 때 0이 되는 지점**을 찾음
- π_k 파라미터는 클러스터 k 에 속할 확률로, **$\sum_{k=1}^K \pi_k = 1$** 의 조건식 필요
- 따라서, 제약조건을 **라그랑주 승수 벡터**로 대체

$$L_\lambda = \sum_{i=1}^N \ln \sum_{k=1}^K \pi_k N(x_i | \mu_k, \Sigma_k) + \lambda (1 - \sum_{k=1}^K \pi_k)$$

Maximization in GMM (6)

Maximization in GMM

- 각 파라미터에 대해 **편미분했을 때 0이 되는 지점**을 찾음

$$\frac{\partial L_{\lambda}}{\partial \pi_k} = \sum_{i=1}^N \frac{N(x_i | \mu_k, \Sigma_k)}{\sum_{k=1}^K \pi_k N(x_i | \mu_k, \Sigma_k)} - \lambda = 0$$

$$\Rightarrow \sum_{k=1}^K \sum_{i=1}^N \frac{\pi_k N(x_i | \mu_k, \Sigma_k)}{\sum_{k=1}^K \pi_k N(x_i | \mu_k, \Sigma_k)} - \lambda \sum_{k=1}^K \pi_k = 0$$

$$\Rightarrow \sum_{k=1}^K \sum_{i=1}^N \gamma(z_{ik}) - \lambda = 0 \Rightarrow \lambda = N$$

Maximization in GMM (7)

Maximization in GMM

- 각 파라미터에 대해 **편미분했을 때 0이 되는 지점**을 찾음

$$\frac{\partial L_{\lambda}}{\partial \pi_k} = \sum_{i=1}^N \frac{N(x_i | \mu_k, \Sigma_k)}{\sum_{k=1}^K \pi_k N(x_i | \mu_k, \Sigma_k)} - N = 0$$

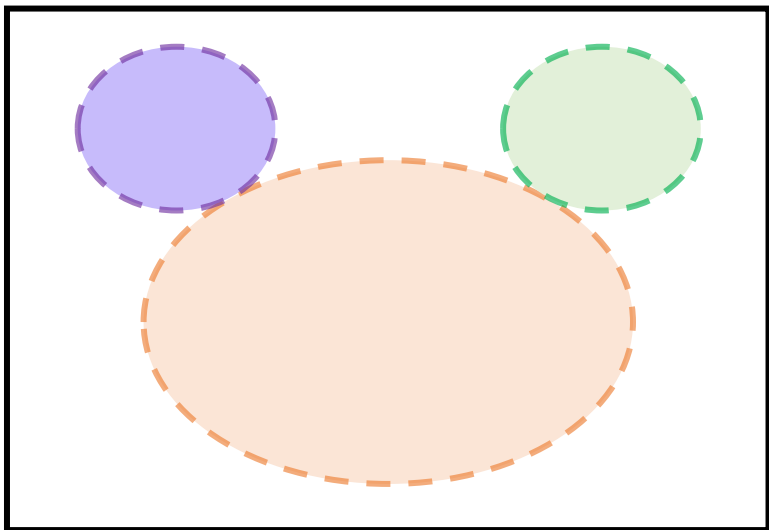
$$\Rightarrow \sum_{i=1}^N \frac{\pi_k N(x_i | \mu_k, \Sigma_k)}{\sum_{k=1}^K \pi_k N(x_i | \mu_k, \Sigma_k)} - N\pi_k = 0$$

$$\Rightarrow \pi_k = \frac{1}{N} \sum_{i=1}^N \gamma(z_{ik})$$

Pros and Cons of GMM

GMM 장점과 단점

- (장점) 각 유형별 데이터의 밀도가 일정하지 않거나, 경계가 모호해도 군집화가 잘됨
- (단점) 클러스터 개수 K 에 대한 설정이 필요
- (단점) 데이터가 정규 분포의 조합으로 표현된다는 가정이 어긋나면, 성능이 떨어짐
- (단점) 연산량이 크기 때문에 대량의 데이터에 사용하기 어려움





Part Five

Hierarchical Clustering

Hierarchical Clustering (1)

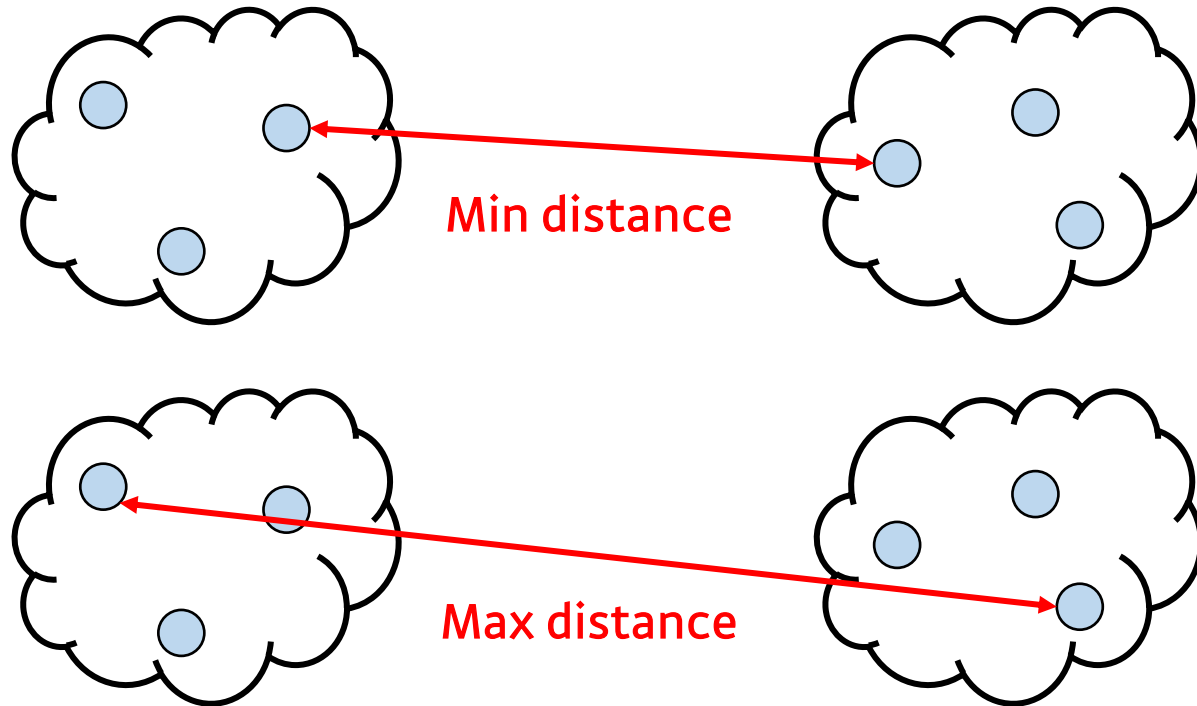
계층적 군집화 (hierarchical clustering)

- 계층적 군집화는 두 가지 방법으로 나눌 수 있음
- 하나의 클러스터로부터 시작해서 모든 클러스터가 하나의 원소를 가질 때까지 쪼개는 **Divisive**(top-down approach) 방법
- 각각의 샘플을 원소로 가지는 클러스터들로부터 전체를 포함하는 하나의 클러스터가 될 때까지 합쳐가는 **Agglomerative**(bottom-up approach) 방법

Hierarchical Clustering (2)

계층적 군집화 (hierarchical clustering)

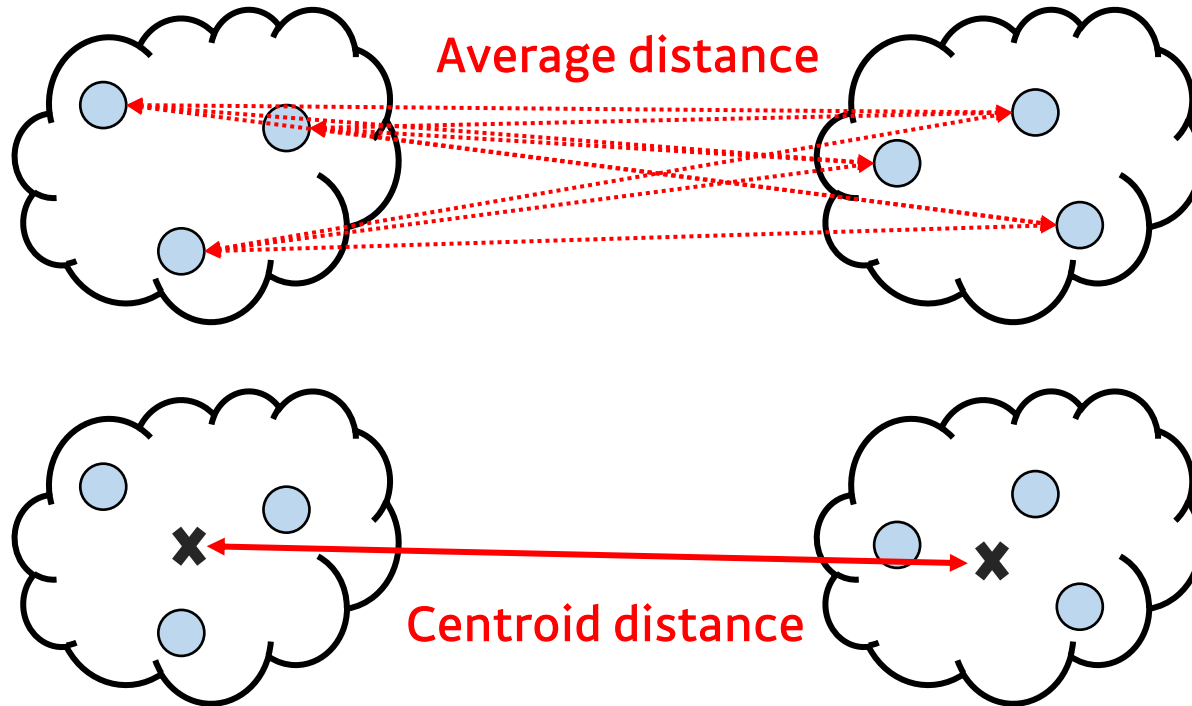
- 군집-군집 간 거리 계산을 통해 합치거나 나눔



Hierarchical Clustering (3)

계층적 군집화 (hierarchical clustering)

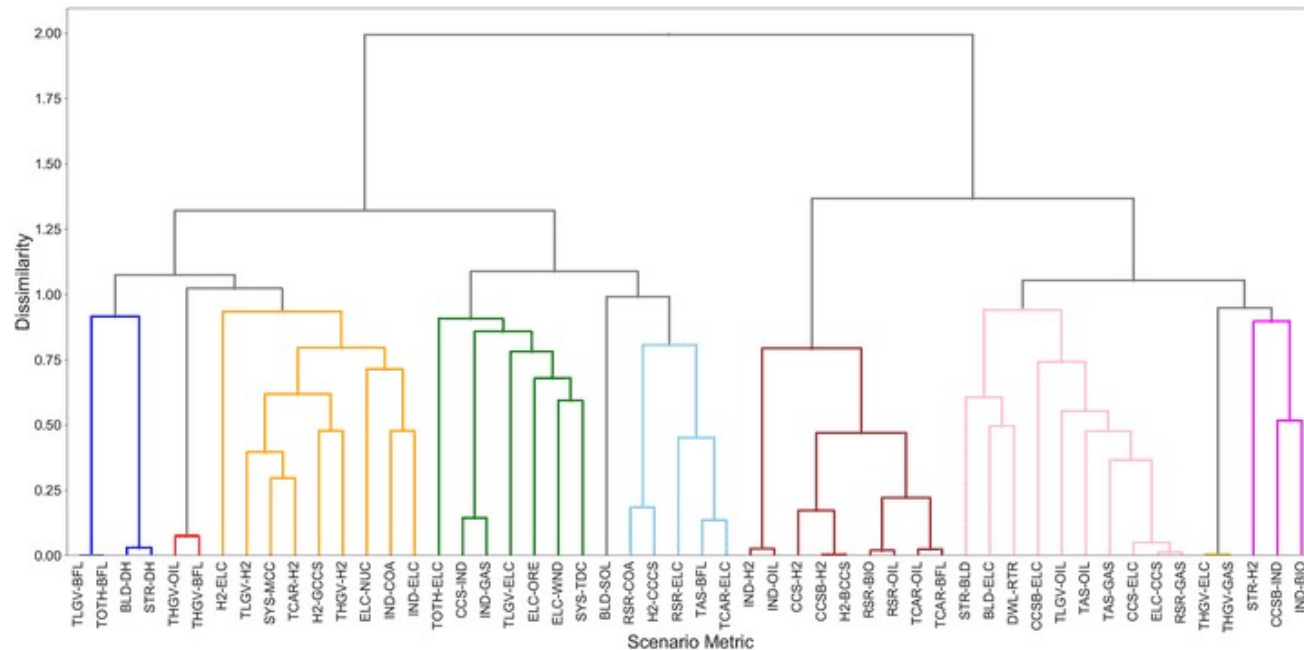
- 군집-군집 간 거리 계산을 통해 합치거나 나눔



Hierarchical Clustering (4)

계층적 군집화 (hierarchical clustering)

- 데이터 세분화에 보다 적합한 방법론
- 사전에 클러스터의 수를 정하지 않아도 학습이 가능함
- 덴드로그램(dendrogram)으로 개체들이 결합되는 순서를 시각화함





Thank you



Clustering