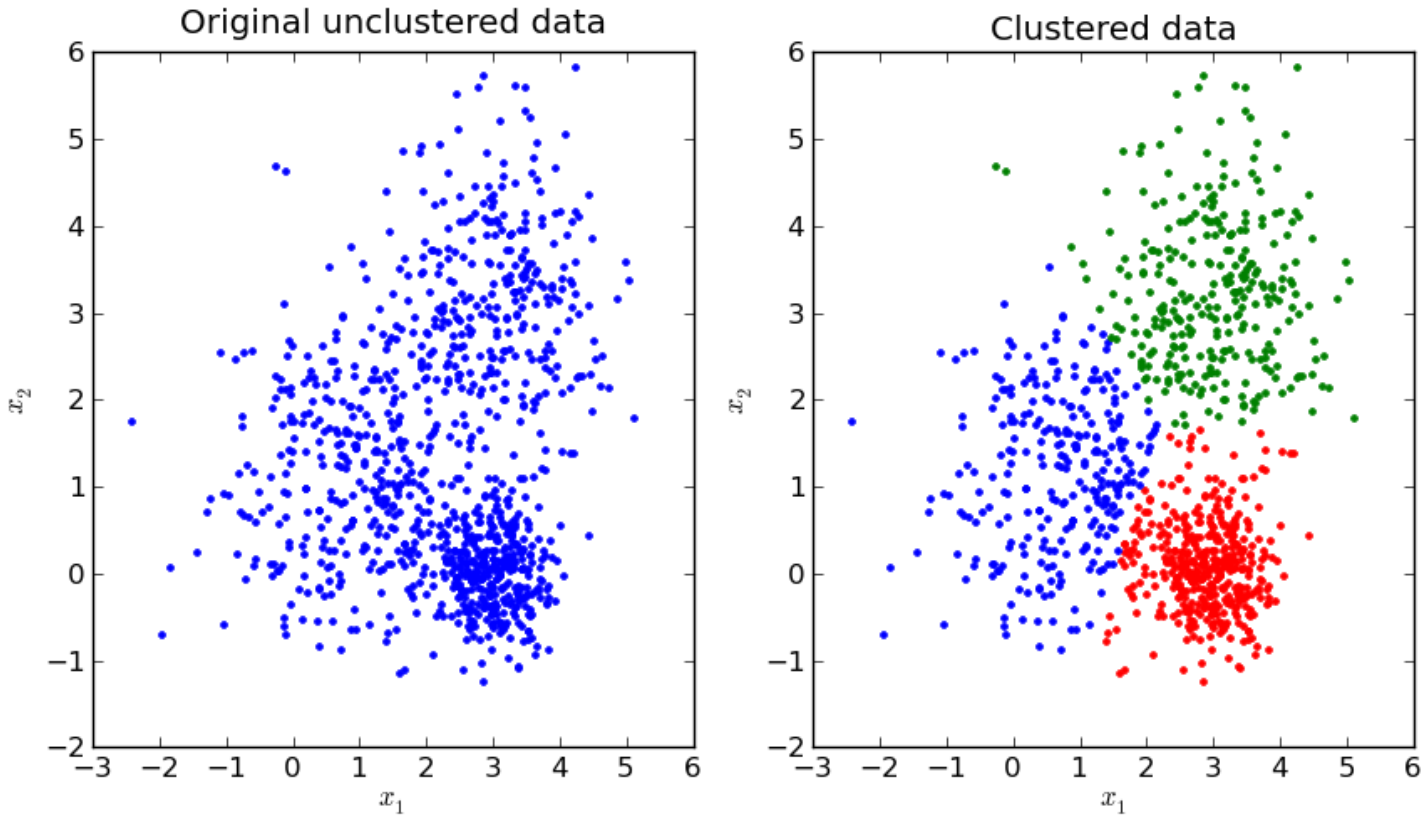
DATA
SCIENCE LAB

Clustering

23.02.09 / 8기 한예림

CONTENTS

0. INTRO



데이터셋의 특성을 어떻게 하면 효과적으로 파악할 수 있을까?

1. Clustering

군집화 (Clustering)

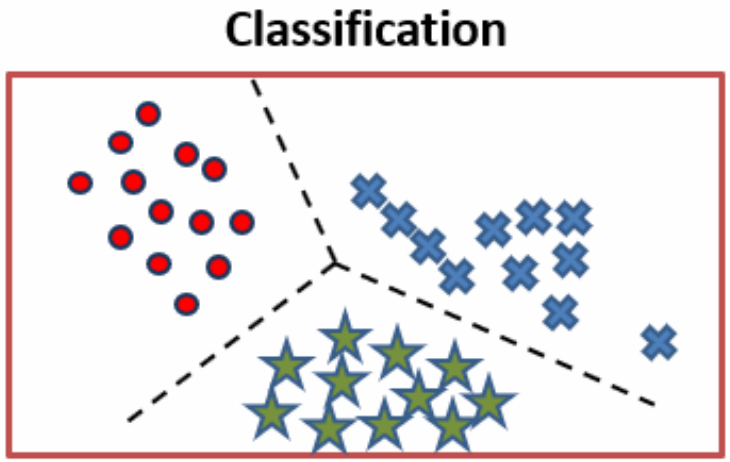
레이블이 없는 데이터 집합을 유사한 데이터들의 그룹으로 나누는 것

- 같은 클러스터 내의 객체들은 유사해야 한다.
  - minimize the inner-cluster variance
- 서로 다른 클러스터의 객체들은 달라야 한다.
  - maximize the inter-cluster variance
- 군집 (Cluster): 나누어진 데이터 그룹

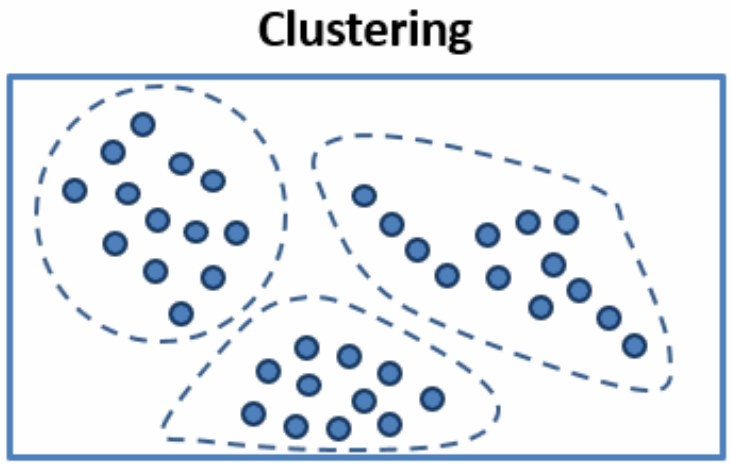
비지도학습의 한 형태

1. Clustering

Clustering is DIFFERENT from Classification!



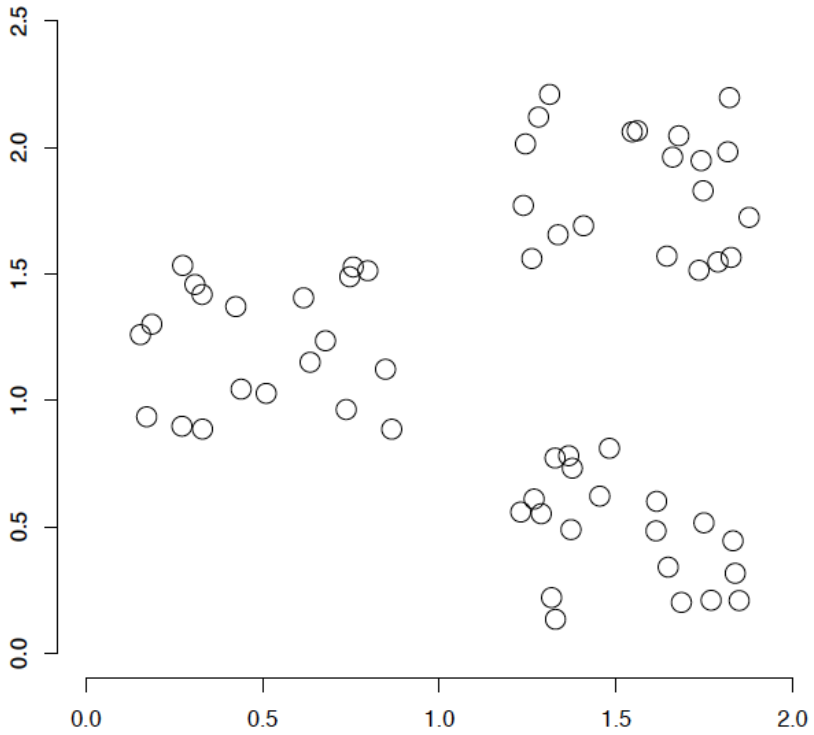
Supervised learning



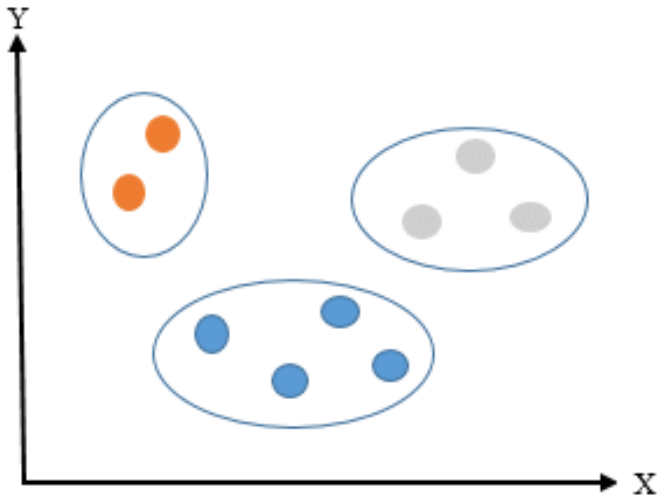
Unsupervised learning

1. Clustering

How can we design an algorithm to make clusters?

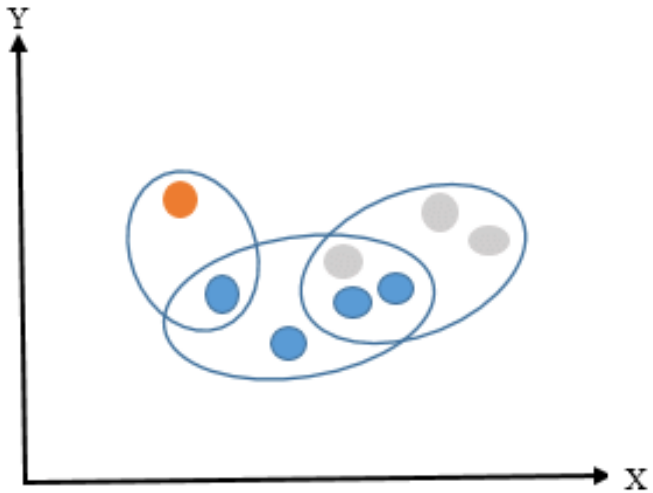


1. Clustering



Hard Clustering

one sample ∈ one cluster



Soft Clustering

one sample ∈ multiple cluster

1. Clustering

Algorithms

Flat algorithms

- 전체 데이터의 영역을 특정 기준에 의해 한 번에 구분하는 클러스터링
- Partitioned
  - K-means clustering
  - Gaussian Mixture Models (GMM) …

Hierarchical algorithms

- 개체들을 가까운 집단부터 계층적으로 차근차근 묶어 나가는 클러스터링
- Bottom-up (agglomerative)
- Top-down (divisive)

DBSCAN

2. K-means Clustering

K-means Clustering

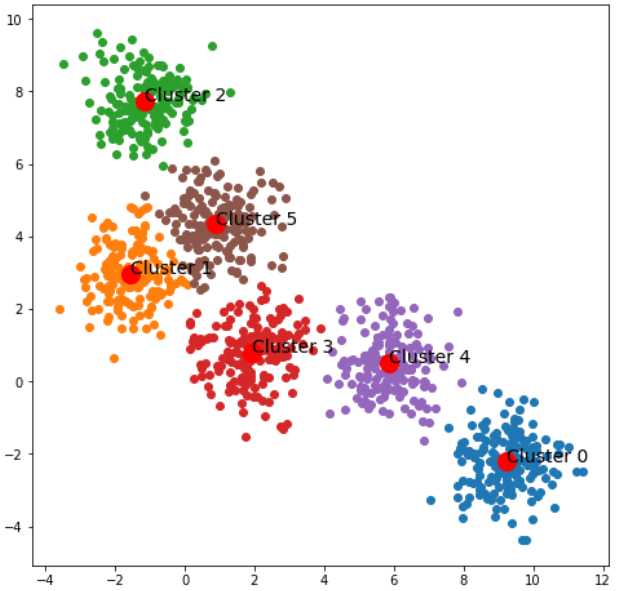
주어진 데이터를 k개의 클러스터를 중심으로 묶는 알고리즘

(1) 임의로 k개의 중심점 (centroid)을 설정

- k개의 중심점은 곧 k개의 클러스터

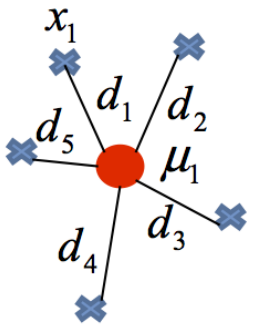
(2) 각 개체는 가장 가까운 중심에 할당되어 하나의 클러스터를 형성

(3) 각 클러스터에 할당된 포인트들의 평균 좌표를 이용해 중심점을 반복적으로 업데이트

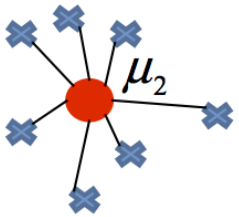


2. K-means Clustering

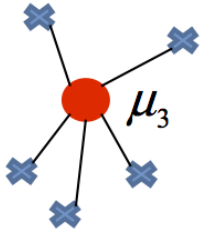
Centroid
Sample



$$\sum_{x_j \in S_2} d_j^2$$



$$\sum_{x_j \in S_1} d_j^2 = d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2$$



$$\sum_{x_j \in S_3} d_j^2$$

$$\min_S E(\mu_i) = \sum_{x_j \in S_1} d_j^2 + \sum_{x_j \in S_2} d_j^2 + \sum_{x_j \in S_3} d_j^2$$

$$X = S_1 \cup S_2 \cdots \cup S_K, \qquad S_i \cap S_j = \emptyset$$

$$argmin \sum_{i=1}^{K} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$$

S Sets of observations
k Number of clusters
x Observation data point
μ_i Mean of points in S_i

각 객체와 그룹 또는 그룹의 중심 간 유클리디안 거리 합이 최소가 되는 방향으로 군집화

2. K-means Clustering

EM Algorithm을 기반으로 작동

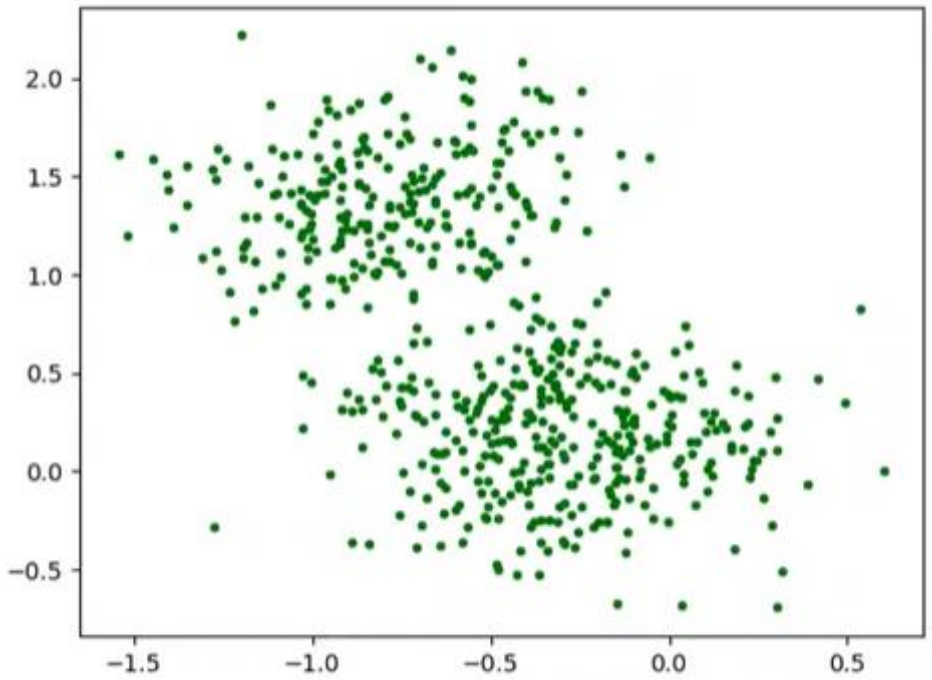
- EM Algorithm: 잠재 변수가 포함된 우도 함수를 최적화하여 모수 (parameter)의 최대 가능도 추정치를 도출
  - 잠재 변수 (latent variable) : 직접적으로 관찰 또는 측정이 불가능한 변수

반복
- Expectation Step: log likelihood의 기댓값을 계산하는 단계 $\mathbb{E}_{q(\mathbf{z})} \ln p(\mathbf{x}, \mathbf{z}|\theta_n)$
- Maximization Step: 기댓값을 최대화하는 모수의 추정값을 구하는 단계 $\theta_{n+1} = \arg\max_\theta \mathbb{E}_{q(\mathbf{z})} \ln p(\mathbf{x}, \mathbf{z}|\theta_n)$

- 1) 각 클러스터 중심의 위치 2) 각 개체가 속하는 클러스터 를 찾아야 함 (잠재 변수 존재)

- EM Algorithm을 통해 각 클러스터의 모수를 찾고자 함

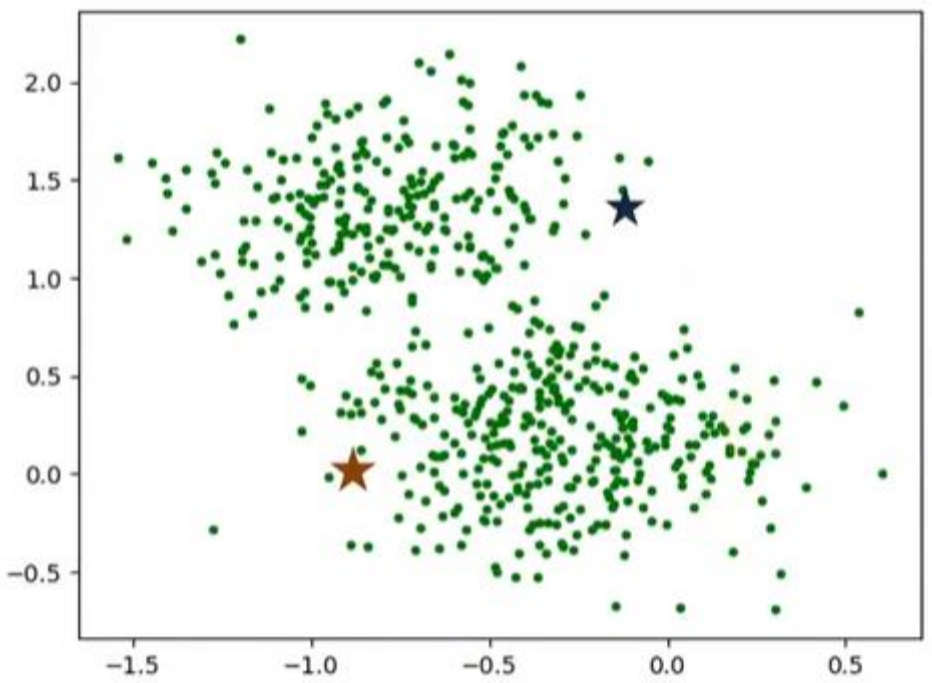
2. K-means Clustering



Data

2. K-means Clustering

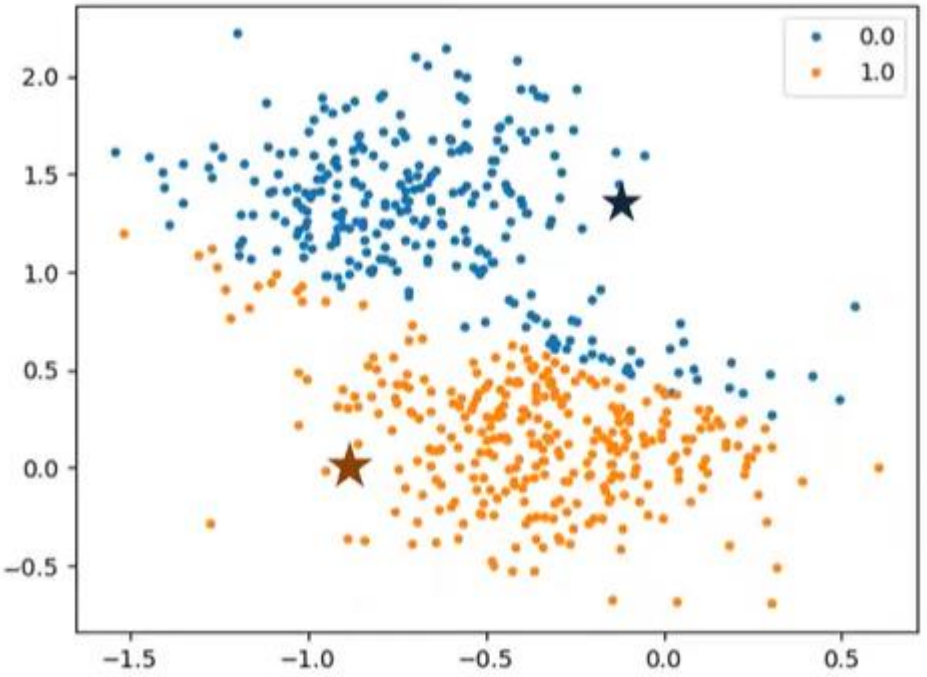
1) E(expectation)-step



Arbitrary means Z are set

2. K-means Clustering

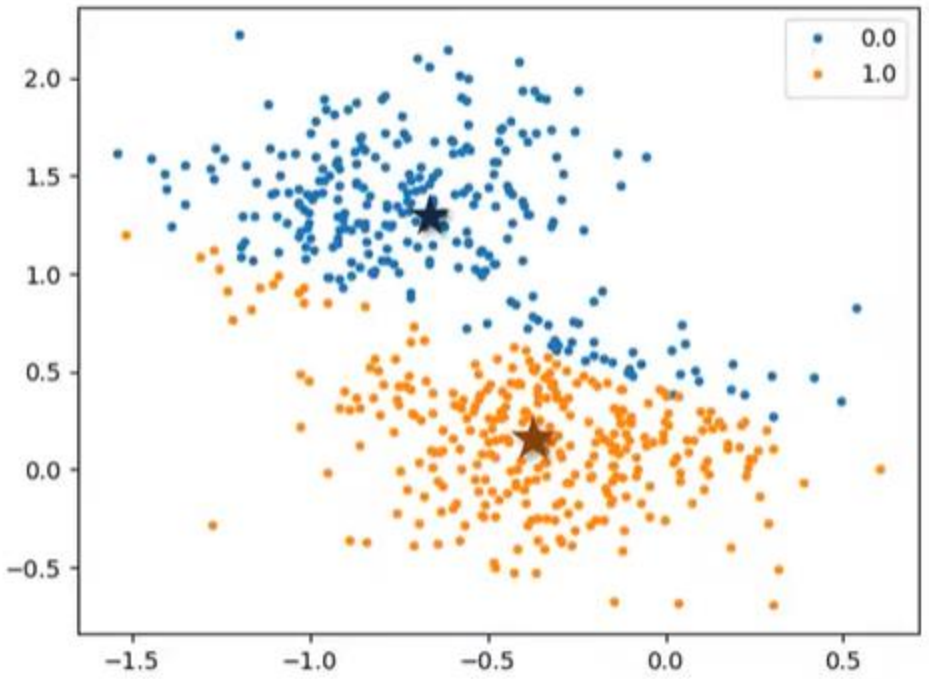
2) M(maximization)-step



Inputs are mapped to the nearest Z

2. K-means Clustering

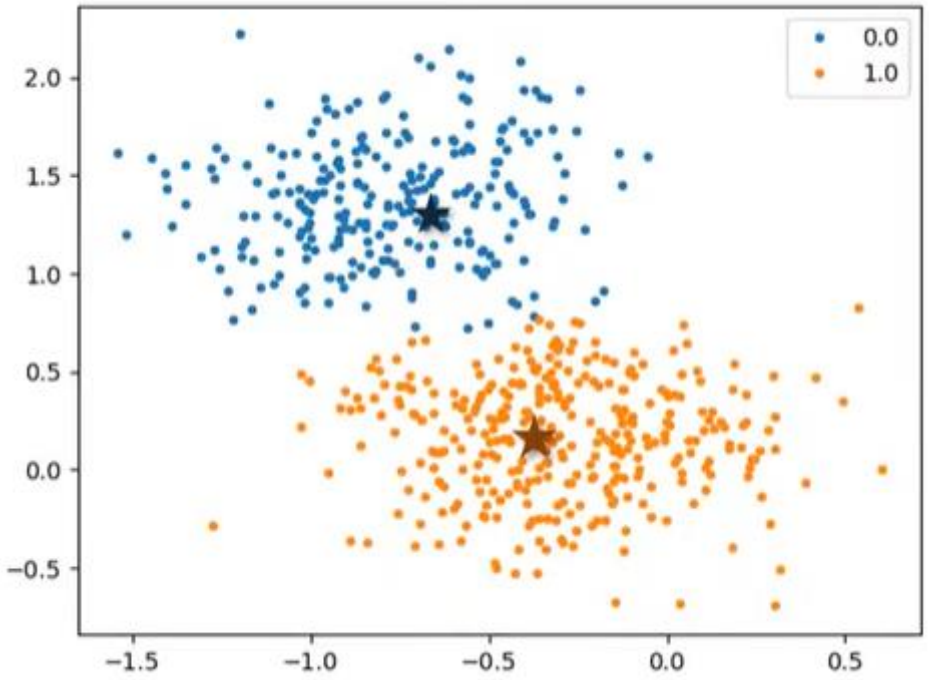
1) E(expectation)-step



Means Z are updated to the mean in each cluster

2. K-means Clustering

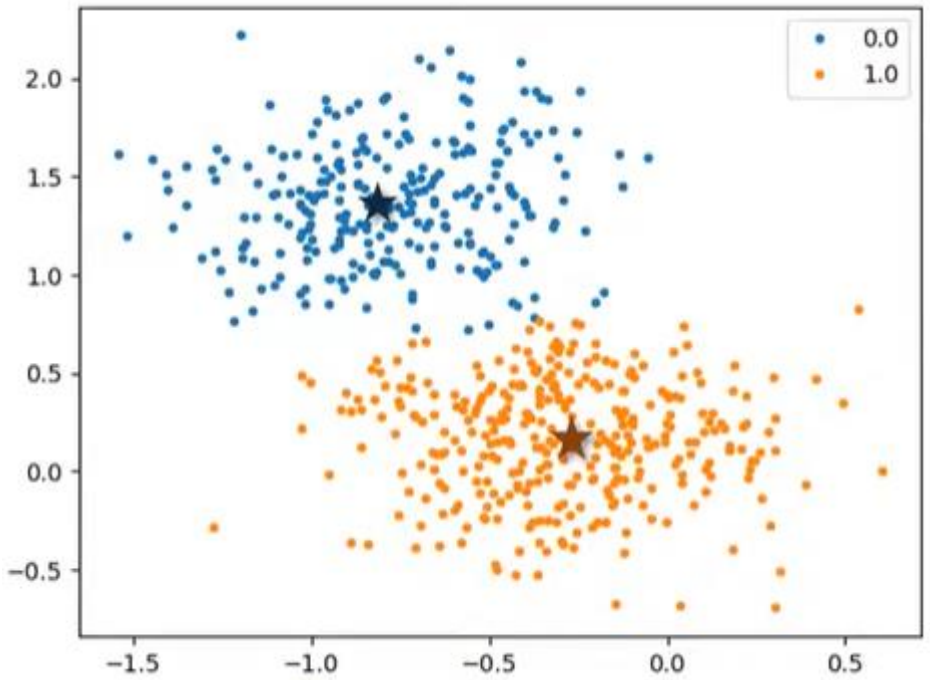
2) M(maximization)-step



Inputs are mapped to updated Z

2. K-means Clustering

1) E(expectation)-step



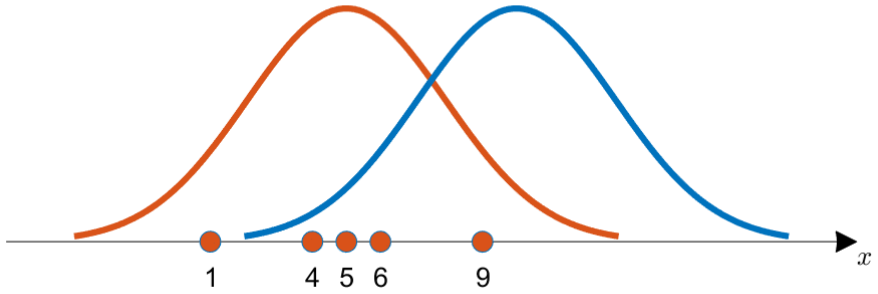
Means Z are updated again to the mean in each cluster

2. K-means Clustering

input들을 계속 업데이트되는 클러스터에 할당하는 작업은 결국 MLE와 같음.

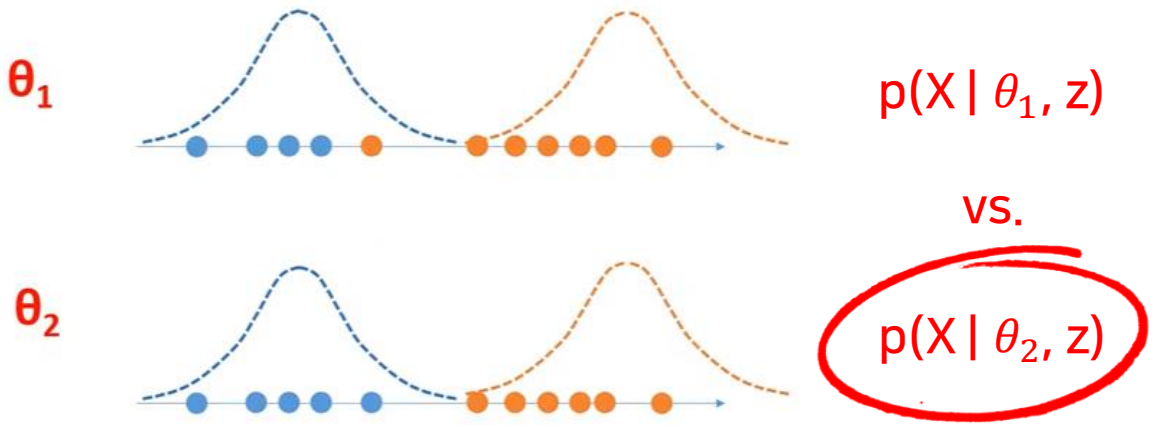
- MLE (Maximum Likelihood Estimation): 최대우도법. 확률밀도함수 p(X | θ)를 최대화하는 θ 값을 찾는 것이 목표

- MLE의 핵심



데이터들은 주황색 커브로부터 나왔을 가능성이 더 크다

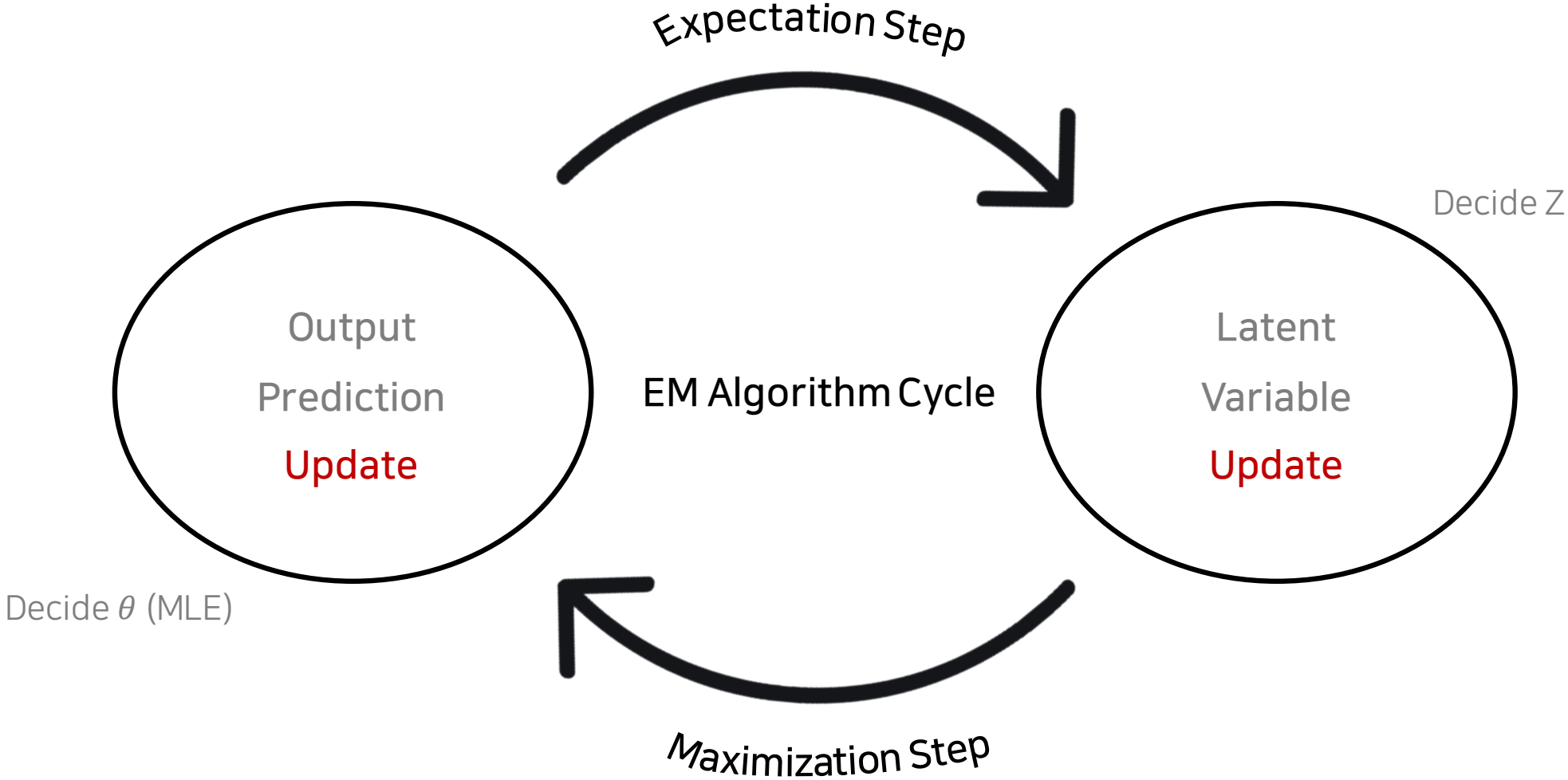
* 해당 예시에서의 z = (Z_0, Z_1) 은 정해진 값이다.



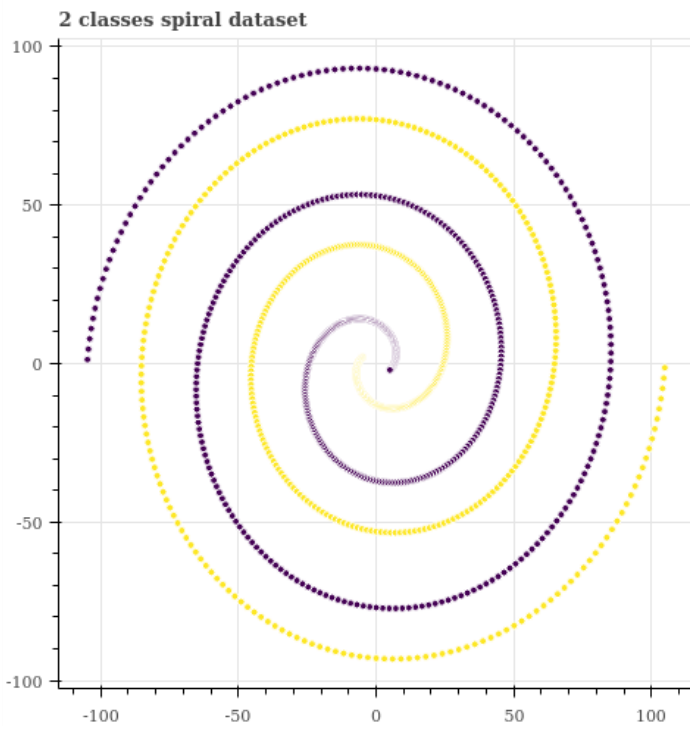
θ_2의 클러스터링 결과가 θ_1보다 좋음

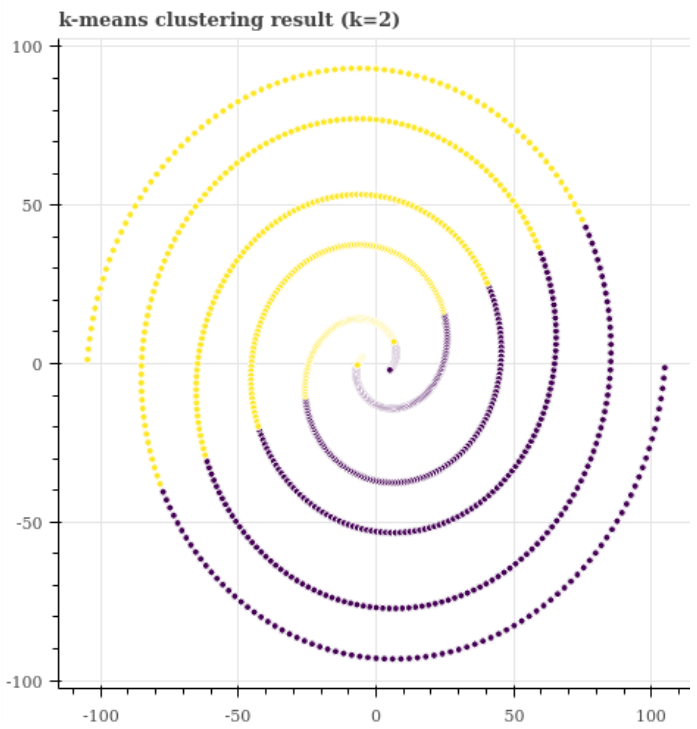
이는 곧 클러스터링도 θ_2의 방향으로 나아가야 함을 의미

2. K-means Clustering



2. K-means Clustering





원점 근방에서 시작해 두 개의 나선으로 이루어진 데이터

- 초기에 정한 **중심점에 따라** 클러스터링 결과가 결정됨
  (중심점 영향 ↑)
- **클러스터 수 k가 커질수록** 순도 높은 결과 도출

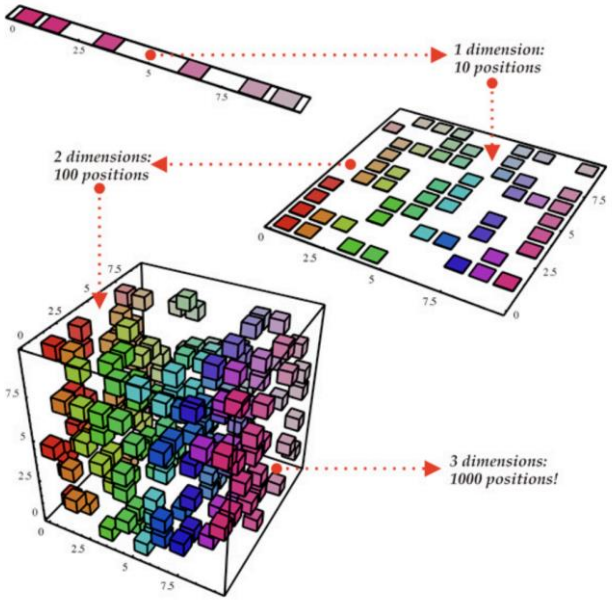
단, 각 군집의 모양이 **구 형태로** convex할 때에만 좋은 결과를 도출

2. K-means Clustering

유클리디안 거리로 거리 측정… 너무 차원이 높으면 클러스터링 성능 ↓

Curse of Dimensionality

차원의 저주. 차원이 증가하면서 학습 데이터 수가 차원 수보다 적어져 성능이 저하하는 현상



고차원 데이터의 특징

- 정보량이 많다.
- 데이터의 차원이 클수록 공간도 커져 필요한 데이터의 수가 지수적으로 증가한다.

2. K-means Clustering

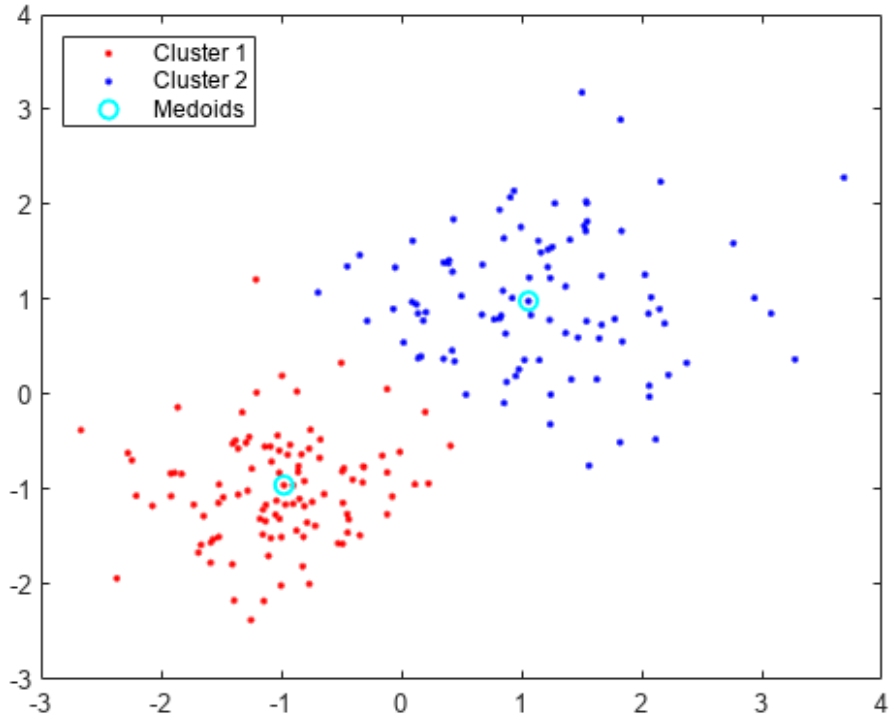
K-means는 outlier에 민감

- '평균'의 한계

K-medoids Clustering

- 클러스터를 대표할 수 있는 **실제 점** 하나를 중심점으로 잡음

=> robust to outliers



- 중앙자 (Medoid): 클러스터의 비유사성의 평균이 클러스터 내의 모든 객체에 대하여 최소가 되는 객체

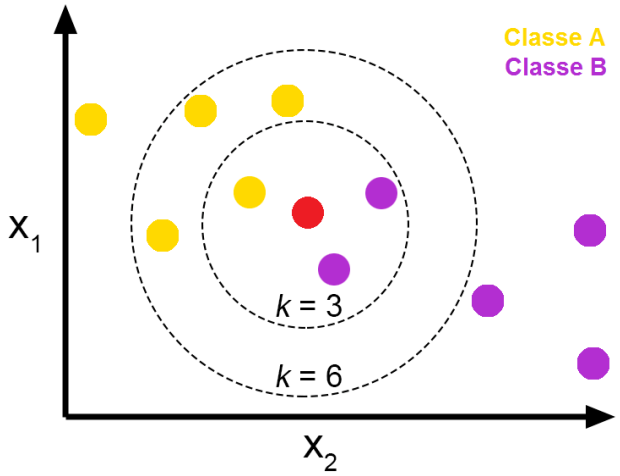
2. K-means Clustering

KNN vs. K-means

두 방식 모두 K개의 점을 지정해 거리를 기반으로 구현되는 알고리즘. 그러나 둘은 목적부터가 다르다.

KNN (K-nearest Neighbors Algorithm)

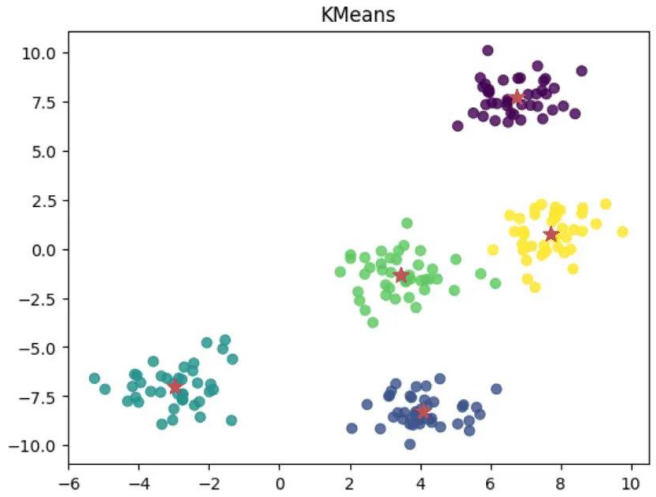
해당 데이터와 가까이 있는 K개의 데이터를 확인한 뒤
더 많은 데이터가 포함되어 있는 범주로 분류



- 지도학습

K-means Clustering

주어진 데이터를 k개의 클러스터를 중심으로 군집화



- 비지도학습

3. Hierarchical Clustering

Hierarchical Clustering

계층적 트리 모형을 이용해 개별 객체들을 유사한 객체와 통합하는 알고리즘

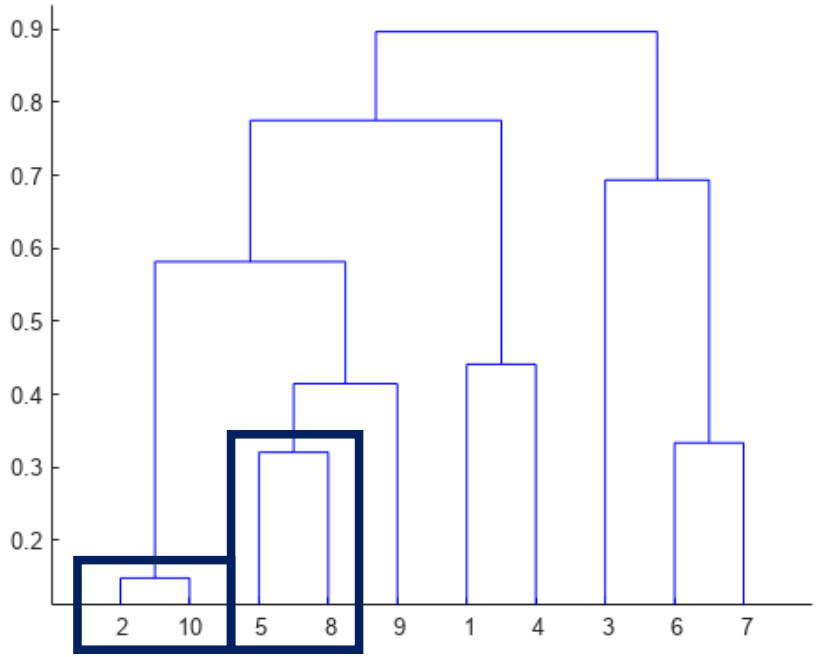
덴드로그램을 통해 시각화 가능

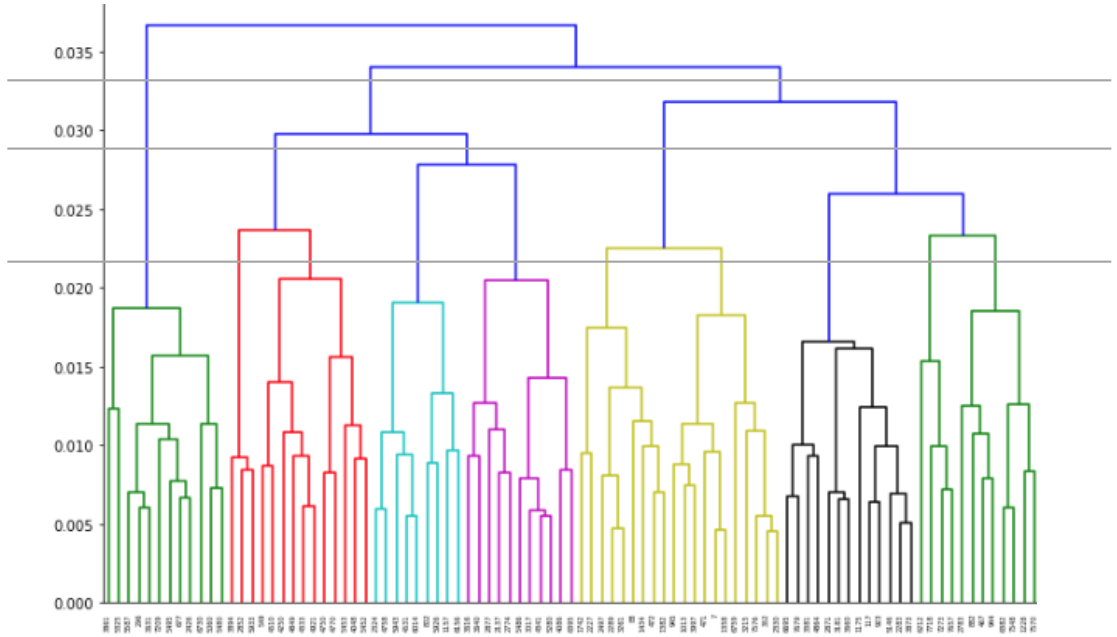
- 덴드로그램 (Dendrogram): 개체들이 결합되는 순서를 나타내는 트리 형태의 구조

사전에 군집의 수를 정하지 않음

- 덴드로그램을 만든 뒤 적절한 수준에서 자르면 그에 해당하는 군집화 결과가 생성됨

3. Hierarchical Clustering





- 덴드로그램의 높이는 객체 간의 거리를 의미한다.

3. Hierarchical Clustering

Agglomerative Hierarchical Clustering

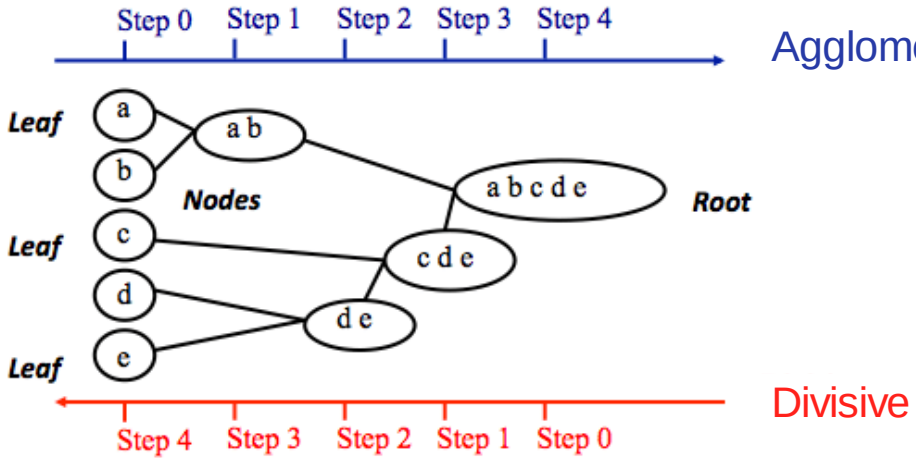
(각 개체를 하나의 군집으로 간주)

(1) 모든 개체들 사이의 거리에 대한 유사도 행렬 계산

(2) 거리가 인접한 관측치끼리 클러스터 형성

(3) 유사도 행렬 업데이트

… 하나의 클러스터로 합쳐질 때까지 병합

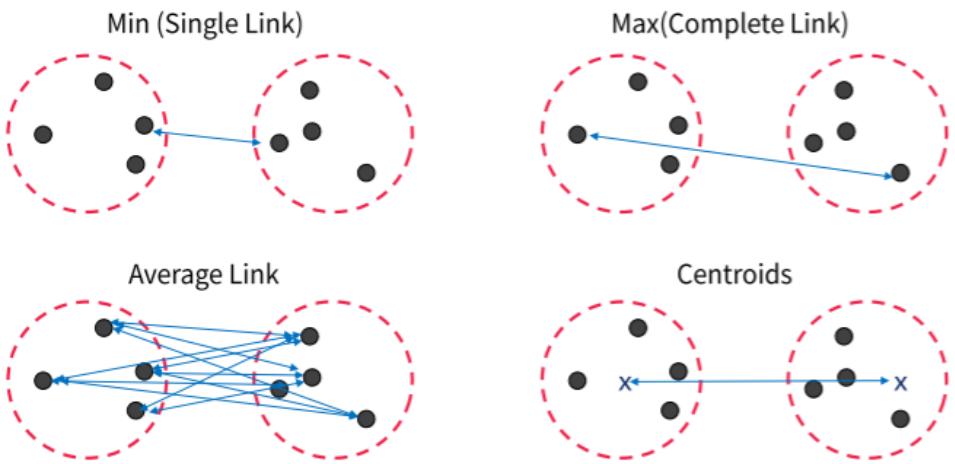


- Bottom-up (agglomerative): 각 데이터 포인트를 순차적으로 병합하여 클러스터링
- Top-down (divisive) : 전체를 하나의 클러스터로 보고 분할해 나가는 클러스터링

3. Hierarchical Clustering

Distance in Similarity Matrix

1) Euclidean



유사도 행렬 계산 시 사용할 거리 지정 가능

- Single (최단연결법): 두 클러스터 간 **가장 가까운 거리**를 사용
- Complete (최장연결법): 두 클러스터 간 **가장 먼 거리**를 사용
- Average (평균연결법): 클러스터 내 모든 데이터와 다른 클러스터 내 모든 데이터 사이의 **거리 평균**을 사용
- Centroids (중심연결법): 두 클러스터의 **중심점 거리**를 사용

3. Hierarchical Clustering

2) Ward's Linkage

두 군집이 합쳐졌을 때의 **오차제곱합 (SSE)의 증가분**을 기반으로 계산

$$Ward_d = \sum_{i \in A \cup B} ||x_i - m_{A \cup B}||^2 - \{\sum_{i \in A} ||x_i - m_A||^2 + \sum_{i \in B} ||x_i - m_B||^2\}$$

(군집을 하나로 묶었을 때) 모든 개체의 중간에 해당하는 점과 각 개체 사이의 거리를 제곱하여 합한 값

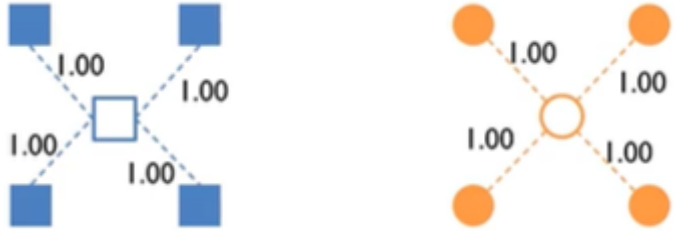
각 군집 내에서 객체들의 중간에 해당하는 점과 각 개체 사이의 거리를 제곱하여 합한 값

- 클러스터 내의 분산을 가장 작게 증가시키는 두 클러스터를 합침
  - 모든 클러스터 내의 분산을 가장 작게 만드는 것이 목표
- 크기가 비교적 비슷한 클러스터를 생성할 수 있음

3. Hierarchical Clustering

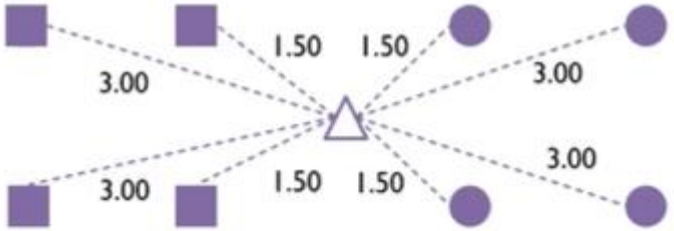
2) Ward's Linkage

- SSE before merge:



$$1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 = 8$$

- SSE after merge:

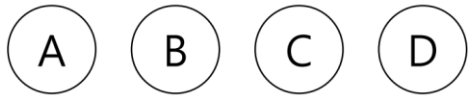


$$4 * 1.5^2 + 4 * 3^2 = 45$$

- Ward distance: 45 – 8 = 37

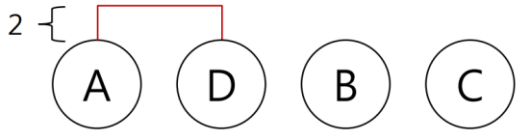
3. Hierarchical Clustering

1)



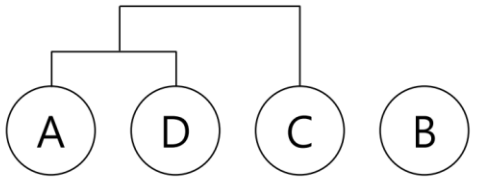
	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

2)



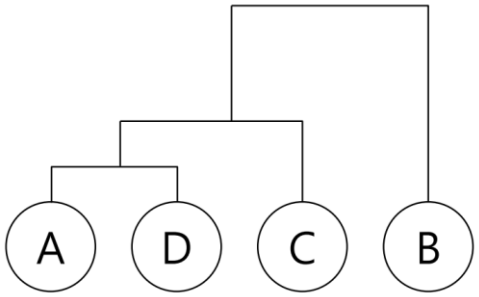
	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

3)



	ADC	B		
ADC		10		
B				

4)



	AD CB			
AD CB				

3. Hierarchical Clustering

Features

군집의 수를 미리 정하지 않아도 됨

random point에서 시작하지 않아 항상 동일한 결과가 나옴

- 클러스터의 수는 덴드로그램을 자르는 위치에 따라 상이하겠지만 결과는 같다

전체적인 군집 파악 가능

but 데이터가 큰 경우 연산 시간이 굉장히 오래 걸림

- 모든 데이터 간 거리 계산 … 데이터 개수가 커지면 연산이 기하급수적으로 늘어남

4. DBSCAN

DBSCAN

점이 세밀하게 몰려 있어 밀도가 높은 부분을 클러스터링하는 알고리즘

- Density-Based Spatial Clustering of Applications with Noise

- 한 점을 기준으로 반경 x내에 점이 n개 이상 있으면 하나의 군집으로 인식

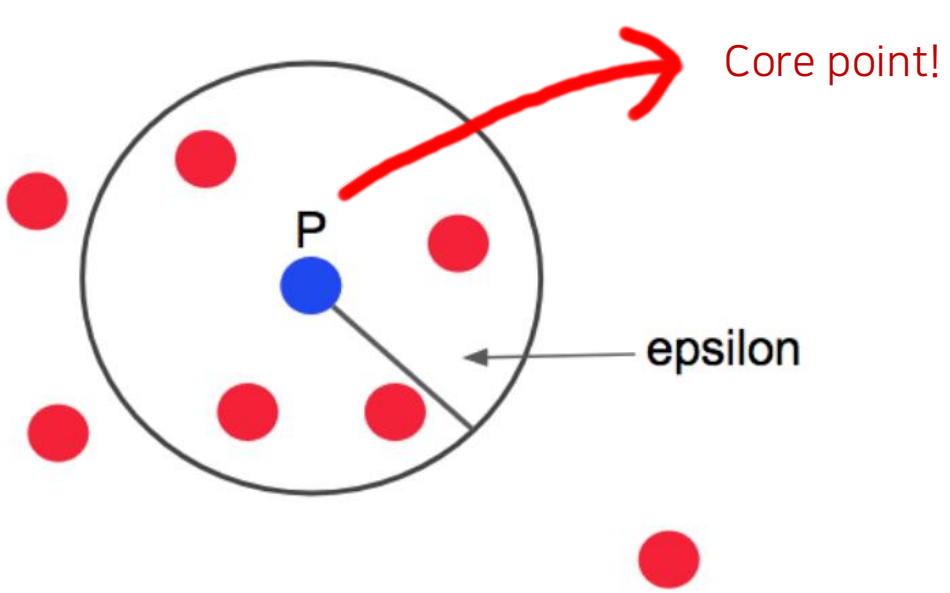
- 1) 점으로 부터의 반경 eps (epsilon) 2) eps 내에 필요한 점의 최소 개수 minPts 를 찾아야 함

4. DBSCAN

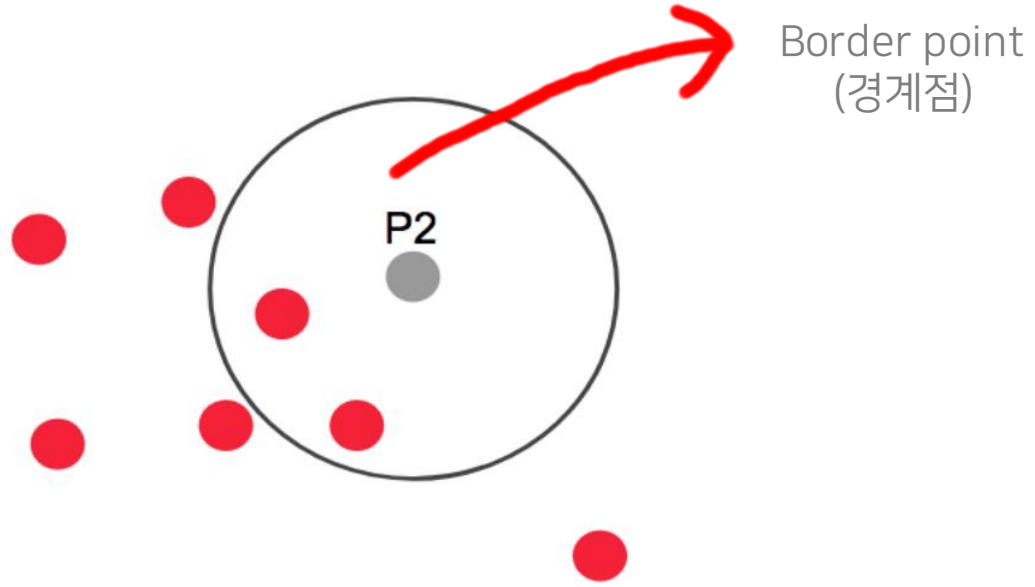
점 p에서부터 거리 e(epsilon)내에 점이 m(minPts)개 있으면 하나의 군집으로 인식한다고 해보자.

➡ 조건을 만족하는 점 p를 core point (중심점) 이라고 한다.

(minPts = 4)



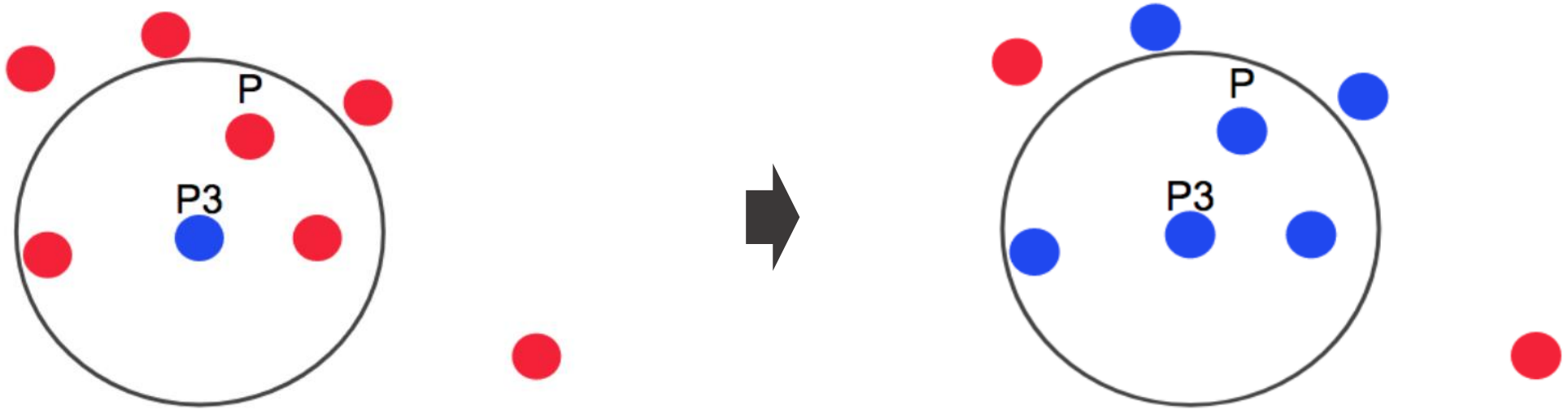
군집 생성 가능



군집 생성 불가능

4. DBSCAN

(minPts = 4)

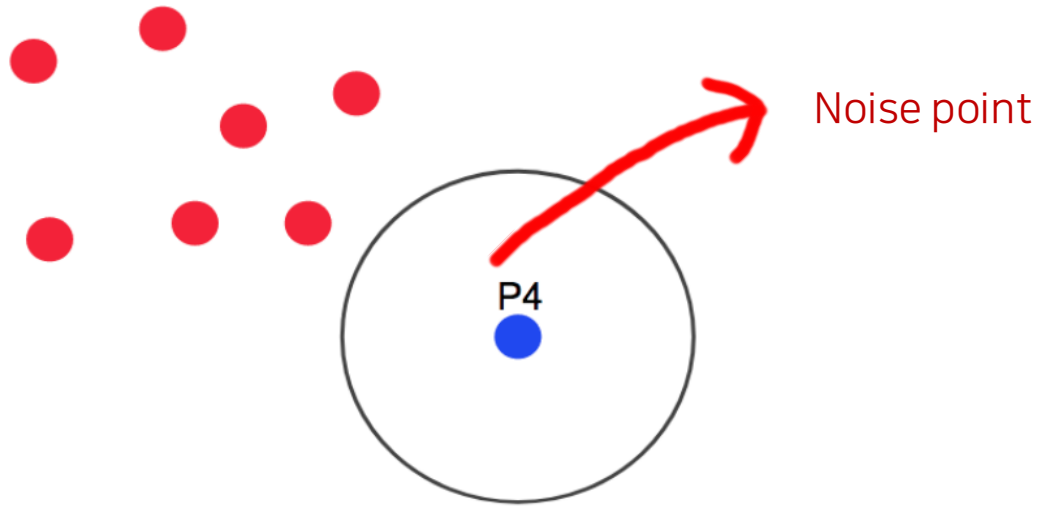


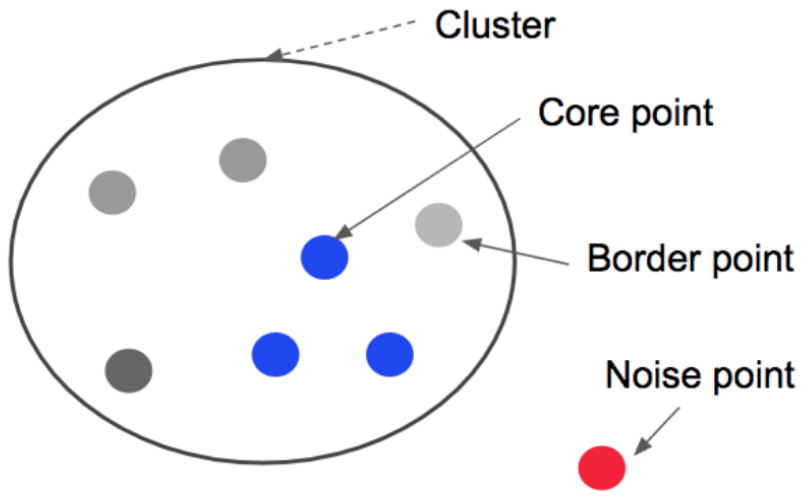
P3를 중심으로 하는 반경 내에 다른 core point P가 포함되어 있다면 …

➡ core point P와 P3는 연결되어 있다고 보고 **하나의 군집**으로 묶임

4. DBSCAN

(minPts = 4)





어느 군집에도 속하지 않는 outlier는 noise point가 된다.

4. DBSCAN

*in “thinnest” cluster

Heuristic Approach to Determine eps and minPts

- Heuristic: problem solving or self-discovery to reach solutions that are not guaranteed to be optimal, but are sufficient for reaching an immediate goal

- sorted k-dist graph

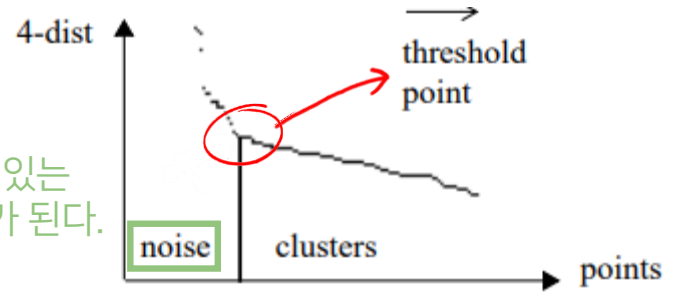


figure 4: sorted 4-dist graph for sample database 3

높은 k-dist값을 갖고 있는 모든 포인트는 noise가 된다.

*threshold point에서의 k-dist값이 eps가 된다.

1) minPts의 개수를 k라고 하면, KNN
하나의 점으로부터 k번째로 가까운 점과의 거리 k-dist를 구한다.
   - 점 p에 대한 KNN 거리를 d라 하자. p의 d-neighborhood는 모든 점 p에 대해 k+1 개 이상의 점을 포함할 것이다.
2) k-dist를 내림차순으로 정렬하면 데이터베이스의 밀도 분포에 대한 정보를 얻을 수 있다.
3) eps를 k-dist(p), minPts를 k로 설정하면 k-dist보다 작거나 같은 모든 포인트는 core point가 된다.
4) x축은 모든 포인트에 대해 k-dist를 내림차순 정렬한 포인트, y축은 각 포인트에 대한 k-dist 값이다.

"A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”, Martin Ester, Hans-Peter Kriegel, Jorg Sander, Xiaowei Xu, 1996.

4. DBSCAN

Heuristic Approach to Determine eps and minPts

minPts가 너무 작다면 …

- noise로 구분되어야 할 점들도 core point나 border point로 잘못 구분될 수 있음
- 불필요한 군집 생성

논문에 의하면,

- 2차원 데이터에 대해 실험한 결과 minPts > 4 일 경우 k-dist graph는 의미 있는 변동을 하지 않는다.
- minPts가 커질수록 연산량이 상당히 커지게 됨.

➡ 2차원 데이터에서는 minPts = 4로 하자!

"A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Martin Ester, Hans-Peter Kriegel, Jorg Sander, Xiaowei Xu, 1996.

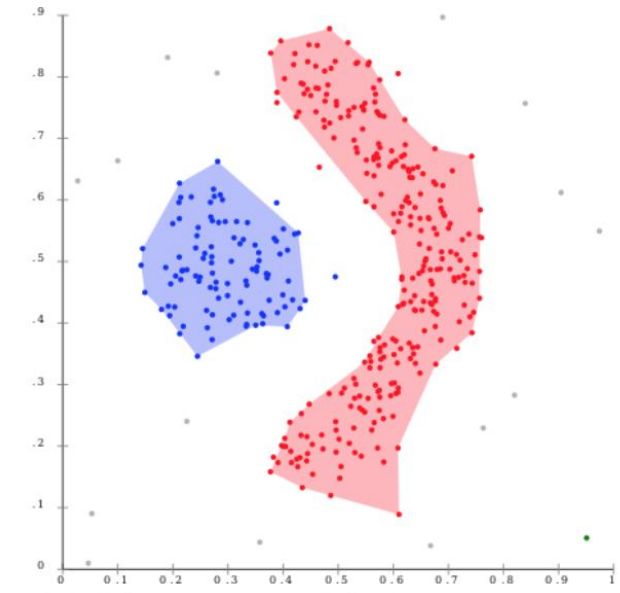
4. DBSCAN

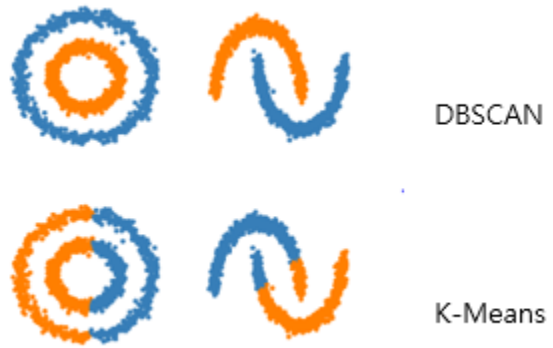
Features

군집의 수를 미리 정하지 않아도 됨

객체의 밀도에 따라 클러스터를 서로 연결하기 때문에
기하학적인 모양의 군집도 생성 가능

noise에 강함 (outlier 검출)

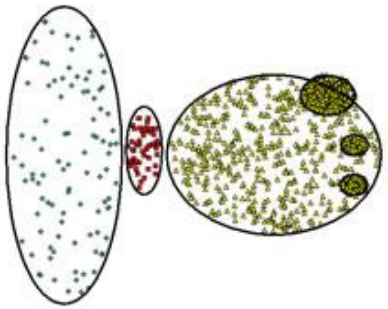




4. DBSCAN

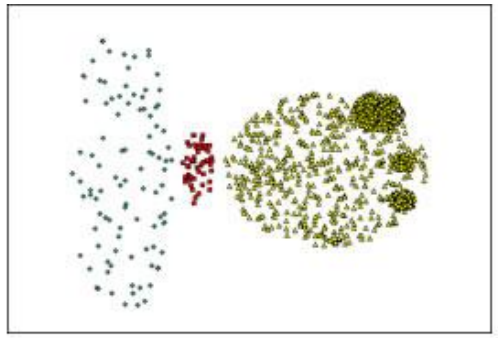
단, '밀도 기반'의 군집화

- 밀도가 높은 곳에만 집중하기 때문에 다른 밀도 분포를 가진 데이터의 클러스터링 결과 ↓

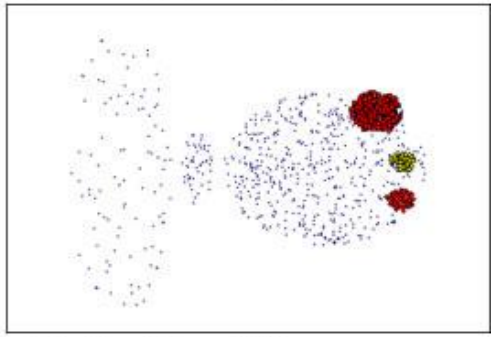


Original Points

- Varying densities
- High-dimensional data



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

5. Issues for Clustering

Things to consider…

클러스터링을 위한 척도

- 유사도(거리)에 대한 개념 필요
- 정규화 (Normalization)

클러스터 개수

클러스터링 결과를 평가할 수 있는 지표

… 등등

5. Issues for Clustering

Proximity measures

데이터 포인트 간 비유사성 (dissimilarity)을 계산하는 지표

* L2 norm

(1) Euclidean Distance

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

두 관측치 사이의 직선,
즉 최단 거리를 의미

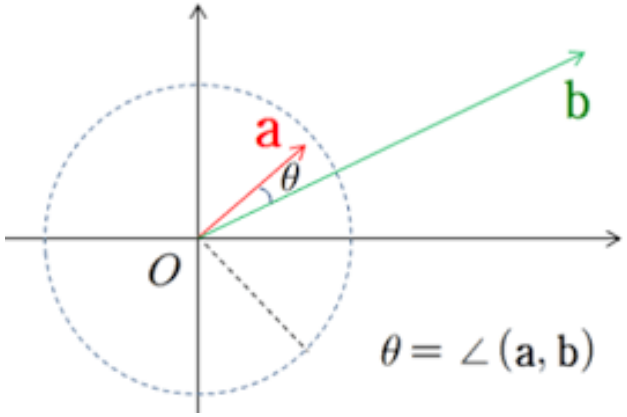
* L1 norm

(2) Manhattan Distance

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

이동이 불가능한 요소를
배제하고 계산

5. Issues for Clustering



… 만약 이와 같은 상황이라면?

(3) Cosine Similarity

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

두 데이터(벡터)가 이루는 사잇각 θ로 유사도를 측정하는 방식

- 사잇값은 벡터의 내적(inner product)으로부터 정의되므로 θ의 코사인 값으로 유사도 측정

코사인 값이 크면 코사인 함수의 성질에 의해 사잇각은 작아지고, 유사도는 높아진다. “pattern”

5. Issues for Clustering

Then, how to validate the clustering?

Elbow method

일반적으로 elbow point에서 최적의 클러스터 수 'k'가 결정됨

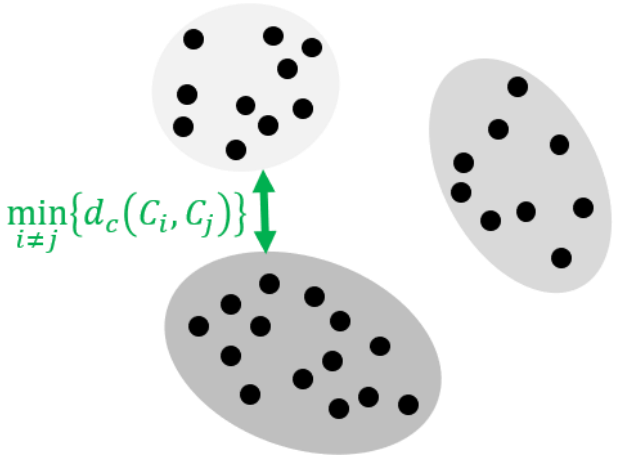
- elbow point를 넘어서면 k가 증가하더라도 함숫값의 significant reduction을 불러오지 않는다

y축에는 다양한 지표가 올 수 있음

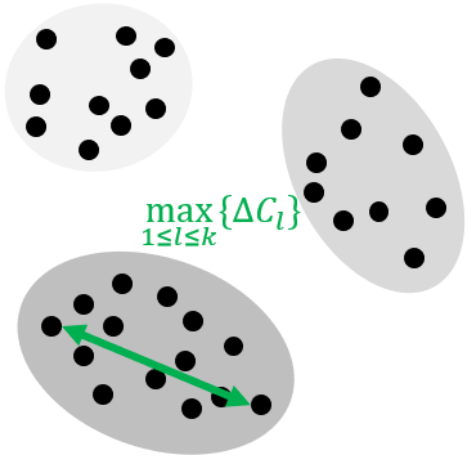
- 내부평가: Dunn Index, SSE, Silhouette …
  - 최적의 군집 개수를 평가
- 외부평가: Rand Index, Jaccard Coefficient …
  - 이미 정해진 정답을 바탕으로 클러스터링의 성능을 평가

5. Issues for Clustering

Finding optimal 'k'



*between clusters



*within a cluster

Dunn Index

클러스터 내의 최대 거리에 대한 클러스터 간의 최소 거리의 비

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

- 인덱스 값이 클수록 클러스터링이 잘 되었다고 판단

5. Issues for Clustering

SSE

각 군집의 중심에서 해당 군집에 속해있는 관측치들의 거리 제곱의 합

$$SSE = \sum_{i=1}^{k} \sum_{x \in c_i} dist(x, c_i)^2$$

- 군집 내에서 중심과 관측치들의 거리는 작을수록 좋음. SSE는 작을수록 좋다.

그러나… 클러스터링의 핵심 목표는 1) 군집 내 분산 최소화
2) 군집 간 분산 최대화

➡ 군집 간 분산 최대화는 어떻게 고려할까?

5. Issues for Clustering

No clear Elbow point?

(i) $a(i) > 0.\ b(i) = 0.$ $\therefore s(i) = -1$

(ii) $a(i) = 0.\ b(i) > 0.$ $\therefore s(i) = 1$

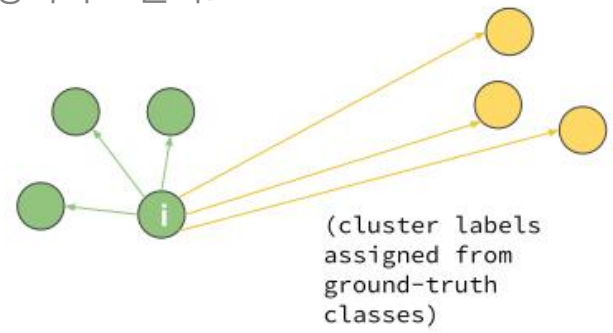
Silhouette Coefficient!

각 군집 간의 거리가 얼마나 **효율적으로 분리**되어 있는지를 나타내는 지표

$s(i) > 0.5$ 면 군집화 결과가 타당하다고 본다.

*클수록 좋다.

$b(i)$: i번째 관측치로부터 **다른 군집 내**에 있는 모든 개체들 사이의 평균 거리 중 **최솟값**



For a single point, i

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

$a(i)$: i번째 관측치로부터 **같은 군집 내**에 있는 모든 개체들 사이의 평균 거리

*작을수록 좋다.

범위가 정해져 있지 않아 무한대까지 갈 수 있으므로 scaling 차원에서 max로 나눠준다.

$-1 \le s(i) \le 1$

클러스터 구분이 어려운, 즉 군집화가 잘 안 된 경우

군집화가 잘 되었을 경우

*만약 클러스터 안에 하나의 객체만 존재하는 경우, 해당 객체의 실루엣 계수는 0으로 본다.

5. Issues for Clustering

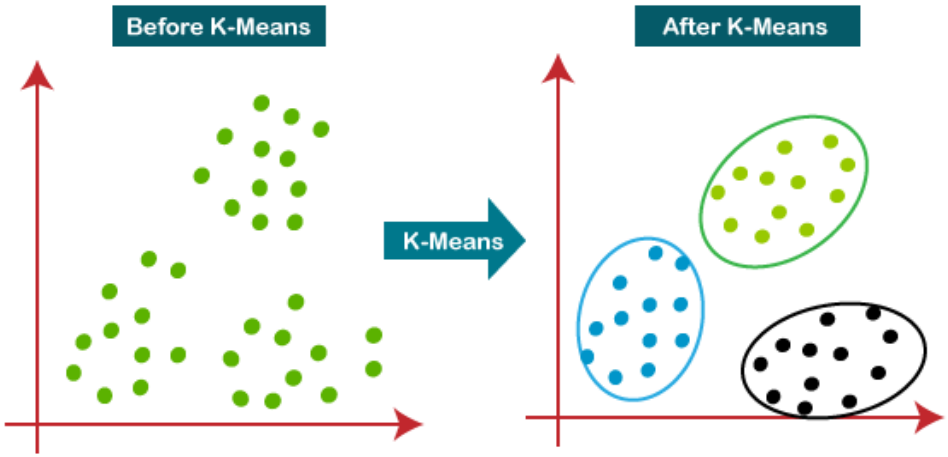
데이터의 패턴이 결과에 영향을 미치는 비지도학습

➡ 반드시 정규화가 선행되어야 한다.

Feature Normalization

데이터를 특정 구간으로 바꾸는 척도법

- 차원의 영향 제거
- 값을 일정한 범위 내로 통일

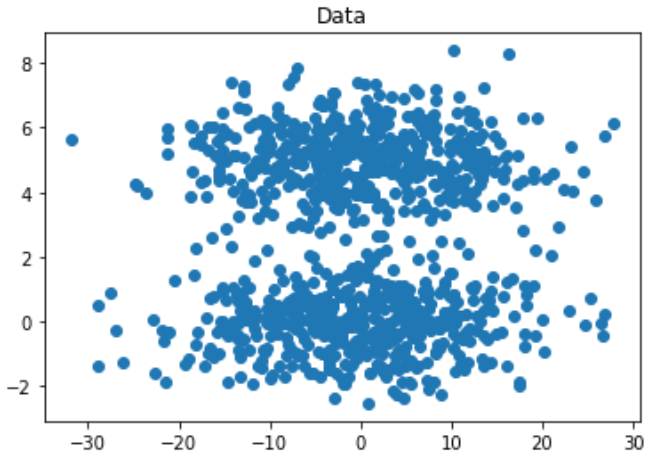


<K-means Clustering>

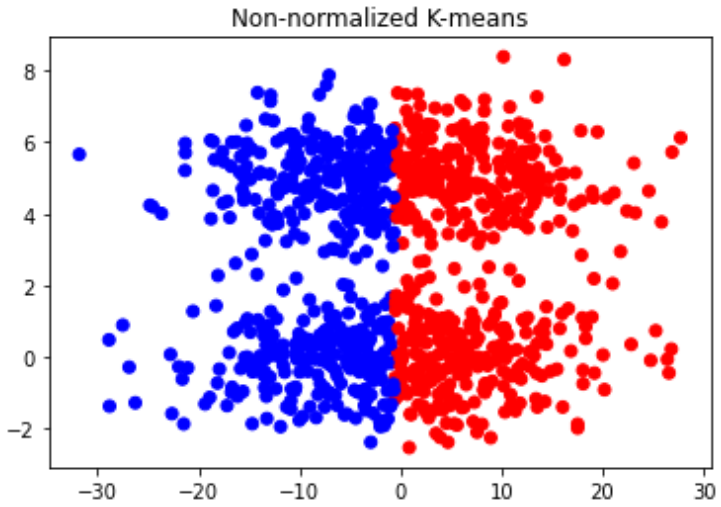
상대적인 거리는 유지하면서 일정한 범위 내에 있도록 변환

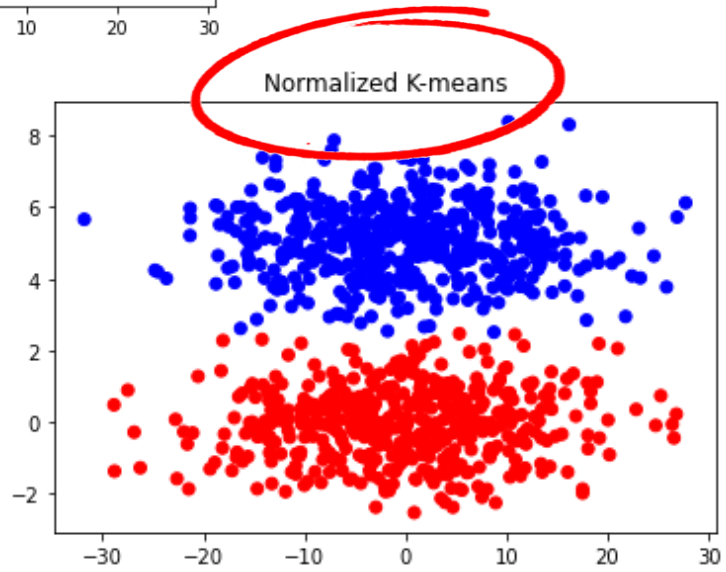
5. Issues for Clustering

Importance of Normalization



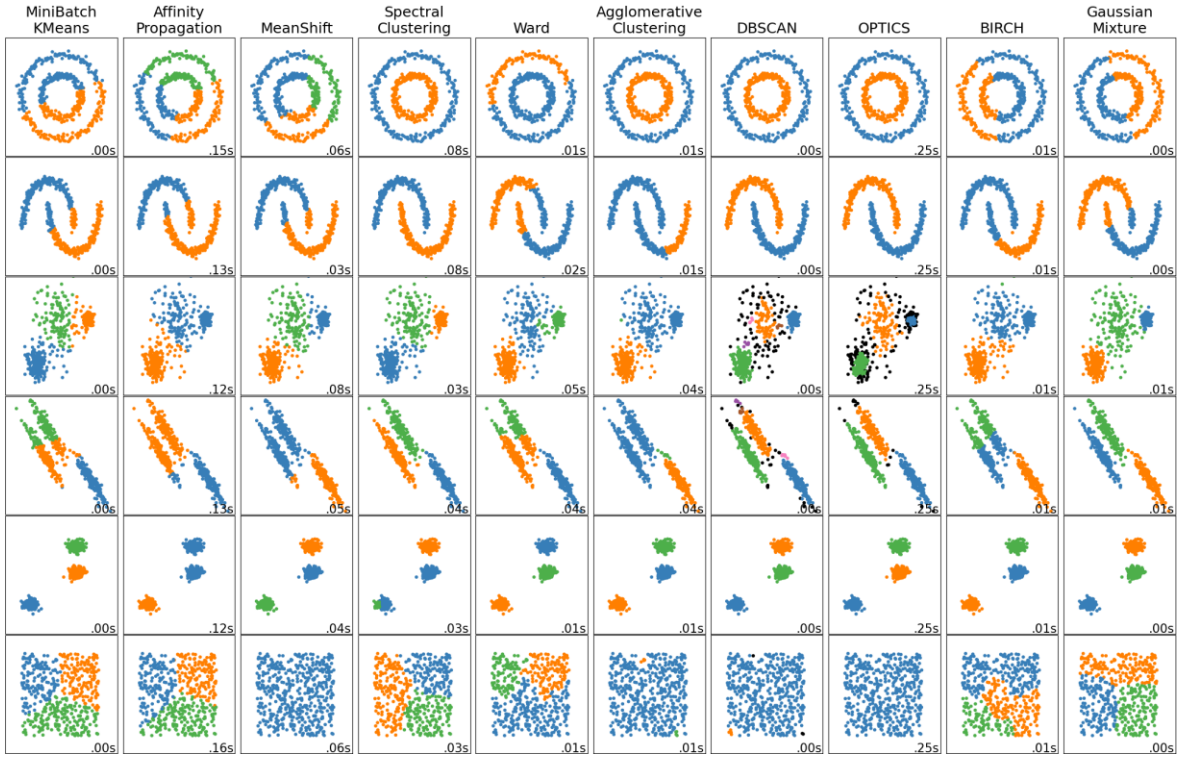
정규화된 데이터셋에서
클러스터링의 결과가 더 잘 나옴





6. Summary

다양한 군집화 알고리즘을 알아야 하는 이유?



해당 데이터에 알맞은 클러스터링을 진행하기 위해

- 데이터의 형태에 따라 방법론 별 군집화 결과가 다르게 나타남

6. Summary

Before clustering…

데이터셋에 대한 정확한 이해가 중요

- Proximity measures
- Normalization
- Algorithm
- Missing values
- Outlier

… 등등 여러 사항을 고려해야 한다!

6. Summary

Recap

- **Why Clustering?**

  주어진 데이터를 **효과적으로 파악**하기 위해

  - 데이터가 얼마나, 어떻게 유사하고 구조는 어떠한가?
  - 비가시적인, 혹은 알지 못했던 새로운 군집 발견

- **Algorithms**

K-means	Hierarchical	DBSCAN
▪ 방대한 양의 자료, 빠른 연산 ▪ 일반적으로는…	▪ 다양한 형태의 거리 지표 사용 ▪ 모든 객체 군집 확인	▪ outlier가 있는 경우 ▪ Non-flat geometry ▪ 다양한 형태의 클러스터 모양

DATA SCIENCE LAB

발표자 한예림
E-mail: yerim.han@yonsei.ac.kr