

데이터마이닝팀

4팀

김현우
김준서
서희나
김수빈
변석주



CONTENTS

1. 클러스터링

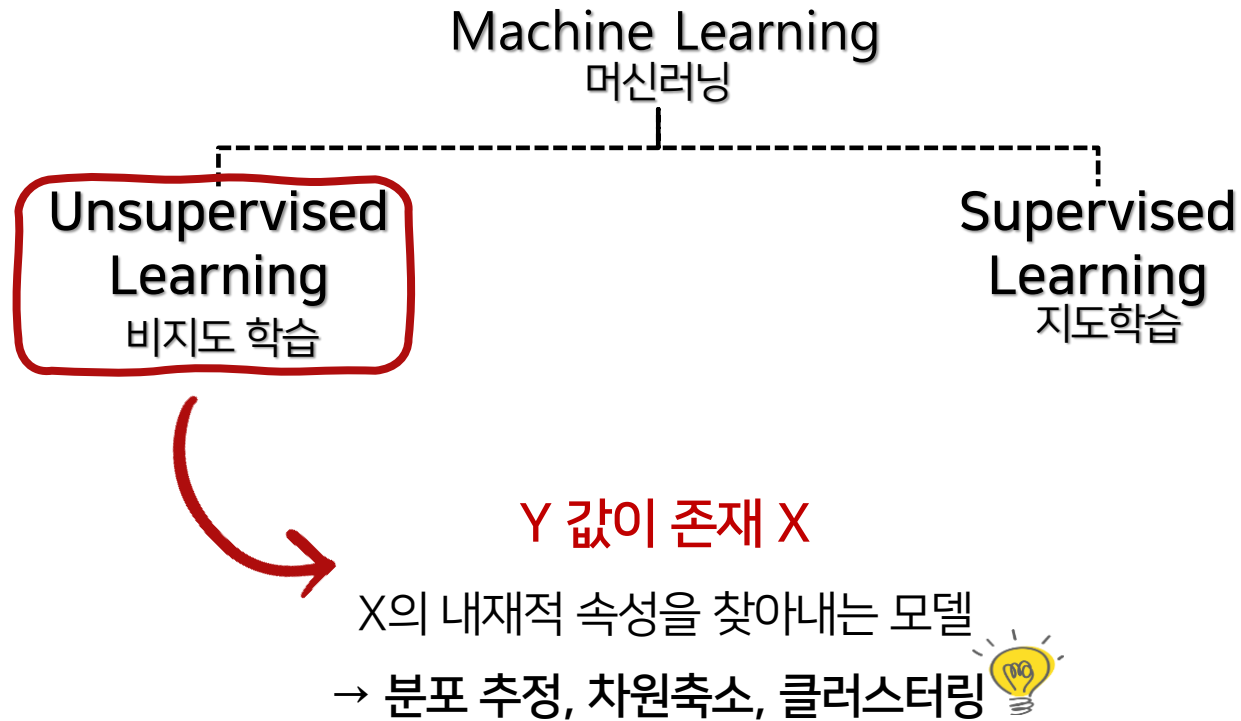
2. 추천시스템

1

클러스터링

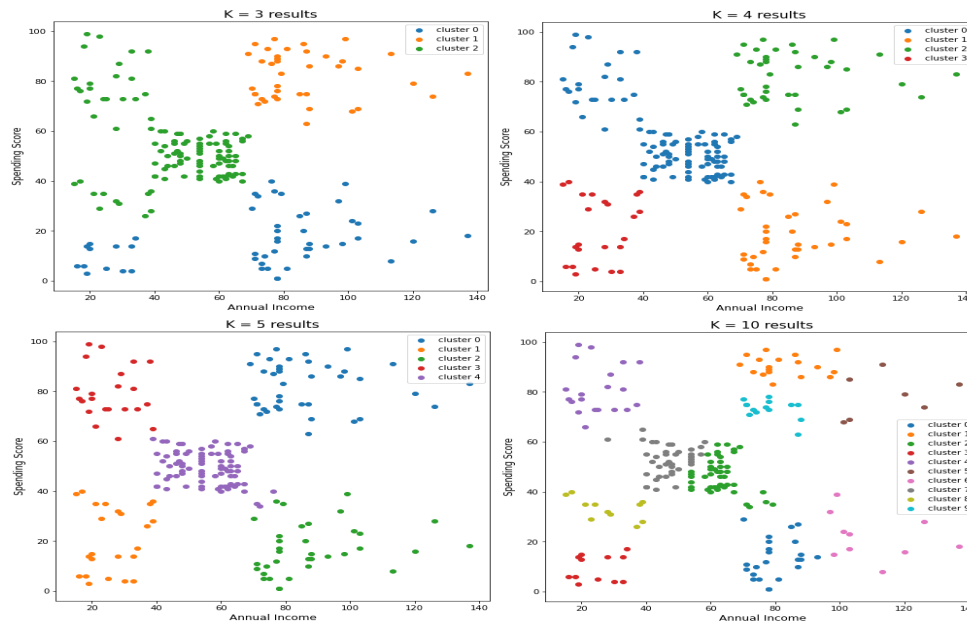
클러스터링

비지도 학습의 대표적 모델



Clustering

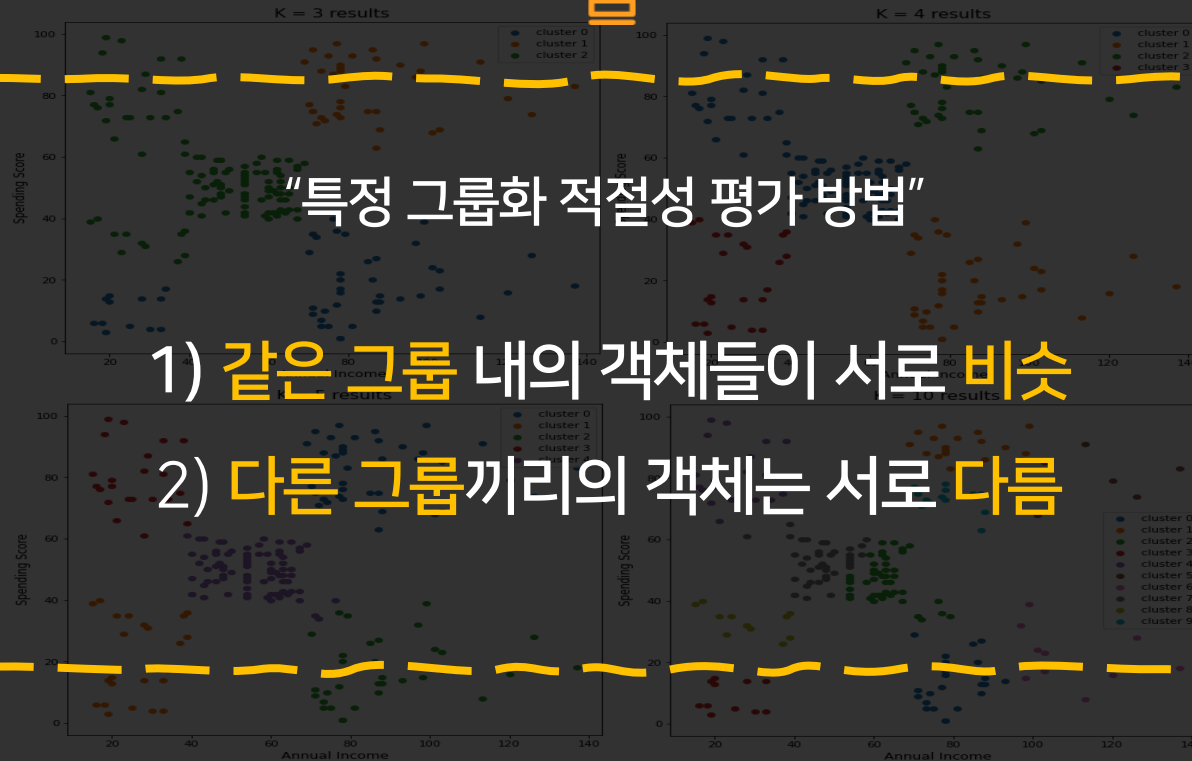
클러스터링



데이터 내에서 군집(그룹)을 찾아내는 것이 목표

Clustering

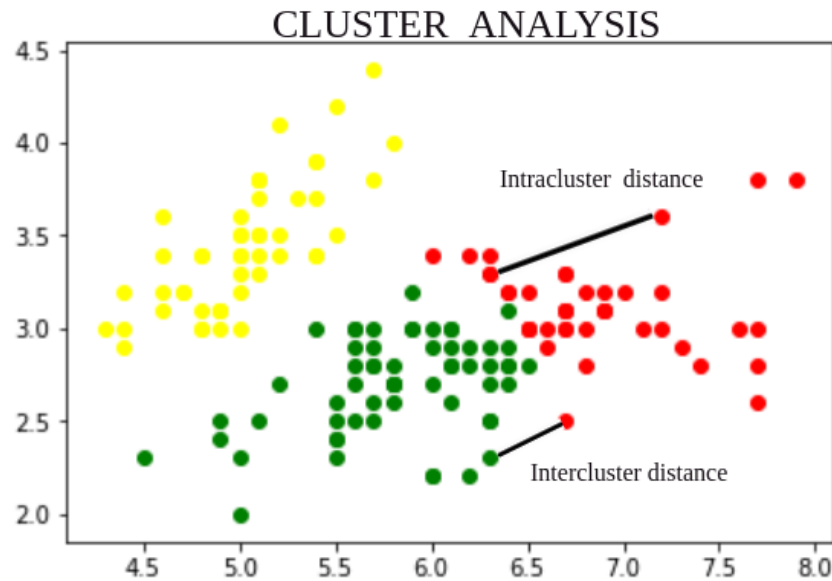
클러스터링



데이터 내에서 군집(그룹)을 찾아내는 것이 목표

Clustering

클러스터링



Intra-cluster distance

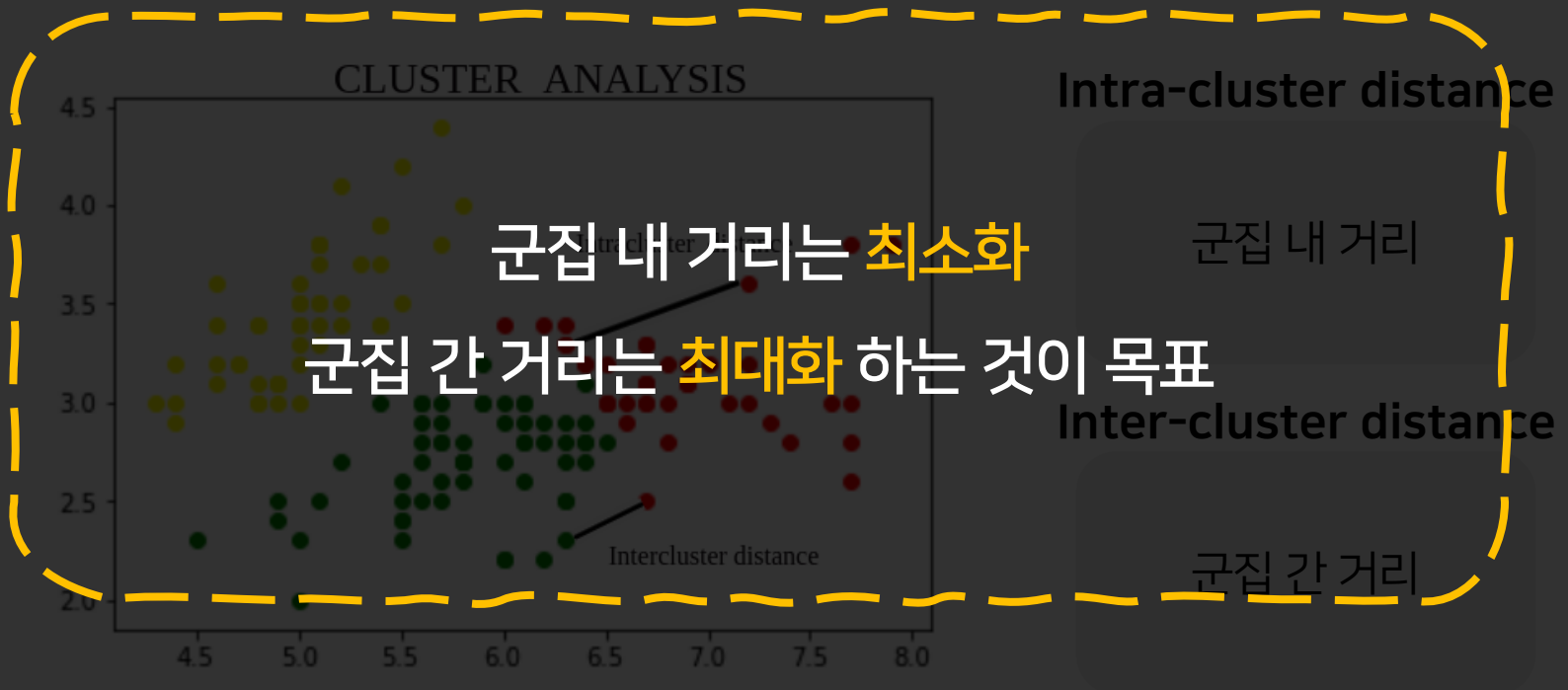
군집 내 거리

Inter-cluster distance

군집 간 거리

Clustering

클러스터링



Clustering

클러스터링

- 클러스터 적절하게 생성 여부
- 몇 개의 클러스터 생성이 적절한지에 대한 여부



이를 측정하는 다양한 지표!

Ex) Dunn family indices, DB index,
semi-partial R-squared SD validity,
Sillhoutte Method, Elbow point Method ...

Clustering

클러스터링

- 클러스터 적절하게 생성 여부
- 몇 개의 클러스터 생성이 적절한지에 대한 여부



이를 측정하는 다양한 지표!

Ex) Dunn family indices, DB index,
semi-partial R-squared SD validity,



Sillhoutte Method, Elbow point Method ...

Silhouette Method

Silhouette 방법

각 데이터 별로 실루엣 계수를 확인하는 방법

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

$a(i)$

군집 내 거리 Intra-cluster variance

객체 i와 같은 군집 안에 속하는 나머지 객체들 간의 거리의 평균

$b(i)$

군집 간 거리 Inter-cluster variance

객체 i와 다른 군집에 속하는 나머지 객체들 간 거리의 평균의 최솟값

Silhouette Method

Silhouette 방법



각 데이터 별로 실루엣 계수를 확인하는 방법

군집 내 거리를 $a(i)$ 는 작게
 군집 간 거리를 $b(i) = \frac{b(i) - a(i)}{\max[a(i), b(i)]}$ 는 크게 이루어질수록 Good!

좋은 군집화 일수록 실루엣 계수는 1에 근접

$a(i)$

군집 내 거리 Intra-cluster variance

객체 i와 객체 i와 같은 군집 안에 속하는 나머지 객체들 간의 거리의 평균

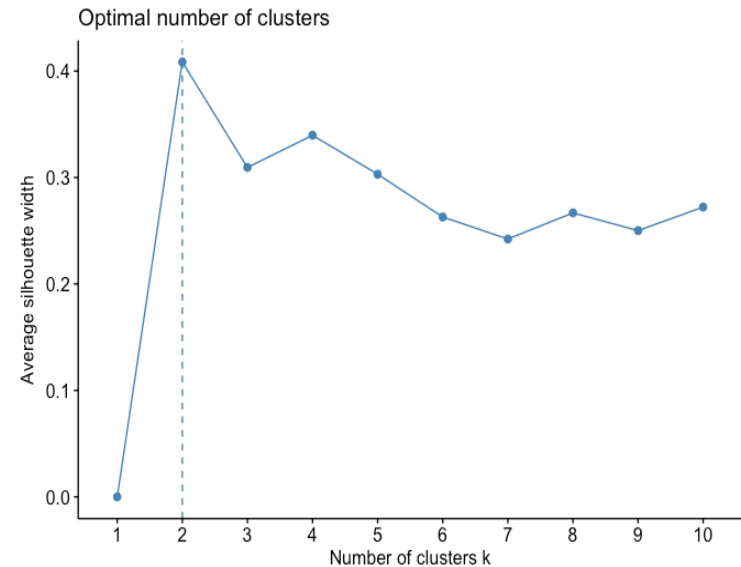
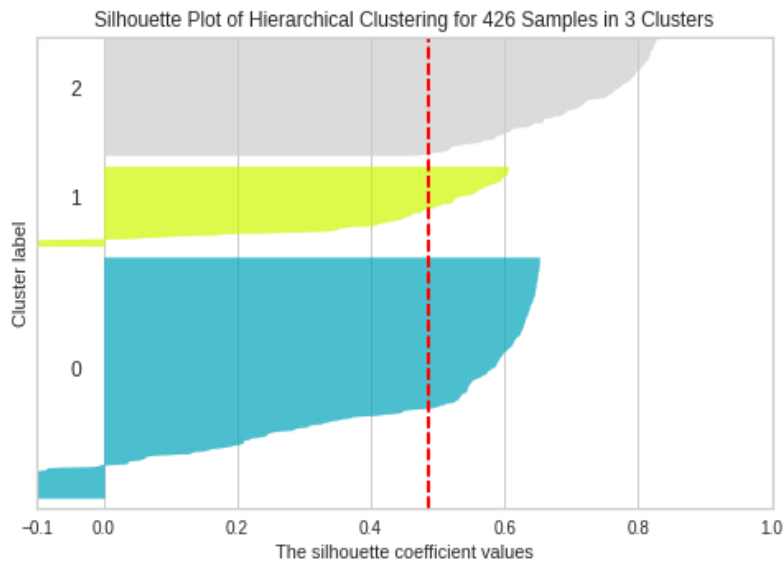
$b(i)$

군집 간 거리 Inter-cluster variance

객체 i와 객체 i와 다른 군집에 속하는 나머지 객체들 간 거리의 평균의 최솟값

Silhouette Method

Silhouette 계수



모든 데이터 포인트에 대해서 실루엣 계수 계산 이후 **평균값** 사용

- 경험적으로 0.5가 넘으면 잘 묶인 클러스터링
- 0.7이 넘으면 정말 잘 묶인 클러스터링이라고 판단

Elbow Point Method

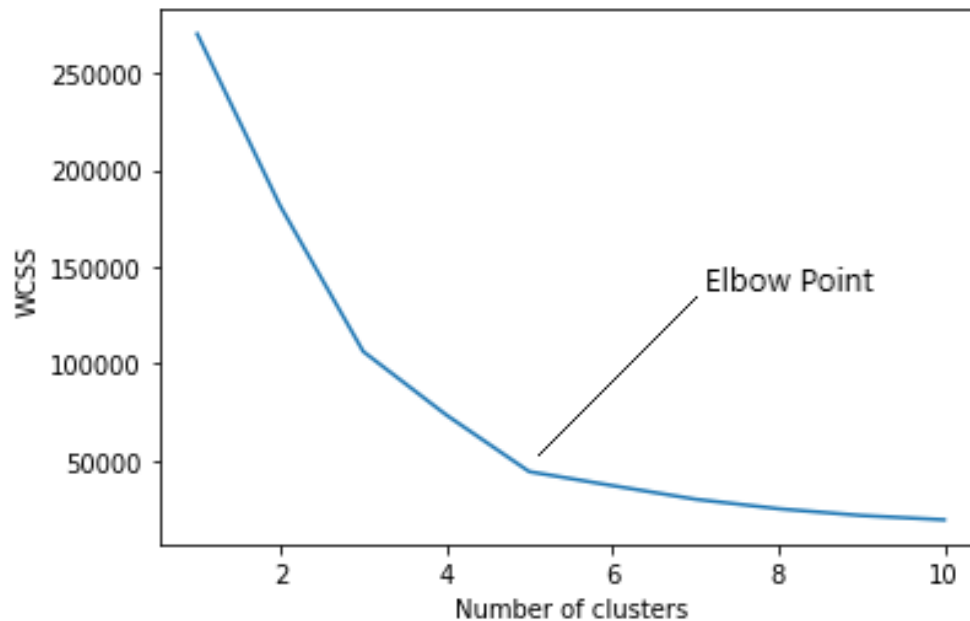
Elbow Point 방법

클러스터 내 RSS가 **최소**가 되도록 클러스터의 중심을 결정해 나가는 방법

- 클러스터 내 중심점과 객체들 간의 거리(RSS)가 최소가 되게 하는 중심점을 고르는 문제
- 클러스터링 개수 증가 → RSS 가 감소함

Elbow Point Method

Elbow Point 방법



오차합이 급격하게 감소하는 시점 → 해당 시점에서 클러스터 개수가 결정됨

클러스터링 종류

비계층적 클러스터링 VS 계층적 클러스터링

비계층적 클러스터링

K-means
K-medoids
DBSCAN

계층적 클러스터링

Hierarchical Clustering

Non-Hierarchical clustering

K-means Clustering

데이터들을 묶어 군집을 생성하는 것을 목표로 함

각 clustering은 중심점을 가지고 있음



데이터 포인트들은 해당 데이터로부터
가장 가까운 중심점을 갖고 있는 군집에 할당됨



이때 클러스터의 개수인 k 를 사전에 정의 내려야 함

"Hyperparameter"

Non-Hierarchical clustering

K-means Clustering

클러스터 위치를 결정하는 데에 쓰이는 군집의 중심으로 데이터의 평균값을 이용

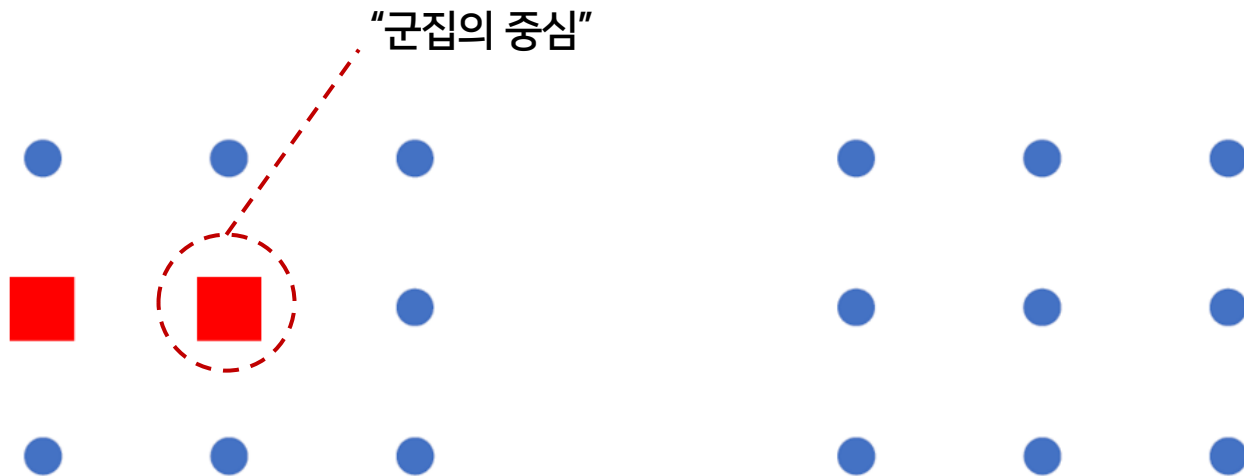
$$WCSS = \sum_{k=1}^K n_k \sum_{C(i)=k} \|x_i - \hat{x}_k\|^2$$

클러스터 내 분산을 의미

결국 이 분산을 최소화하는 것이 목표

Non-Hierarchical clustering

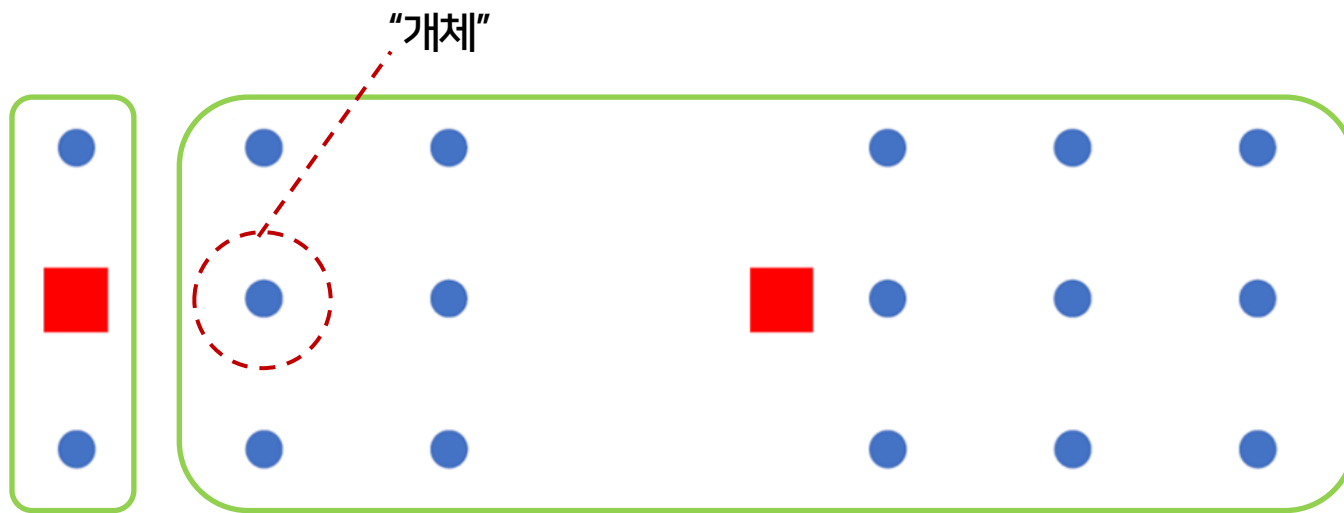
K-means clustering의 학습과정



군집 수를 2로 정하고, 군집의 중심(빨간 점)을 랜덤 초기화

Non-Hierarchical clustering

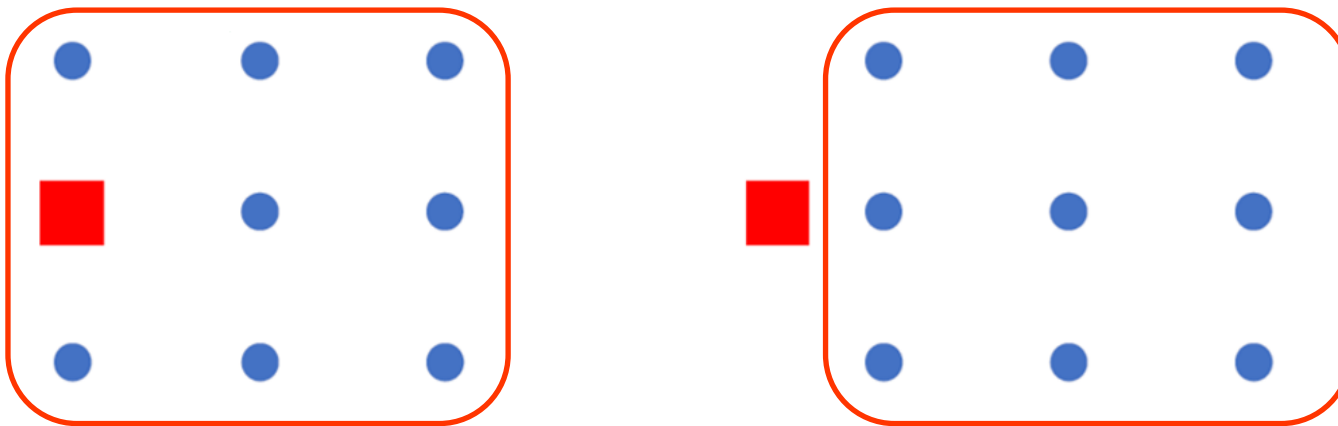
K-means clustering의 학습과정



모든 개체(파란 점)를 그림과 같이 가장 가까운 중심에 군집하도록 할당

Non-Hierarchical clustering

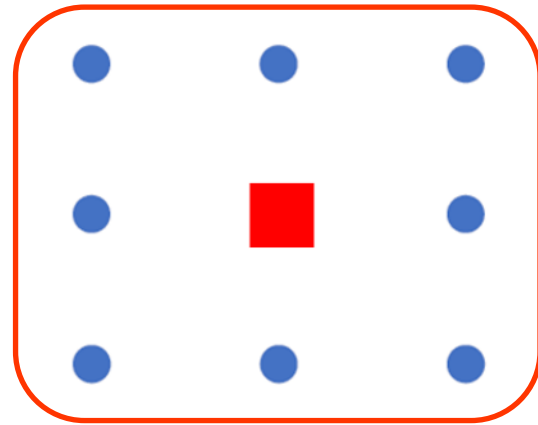
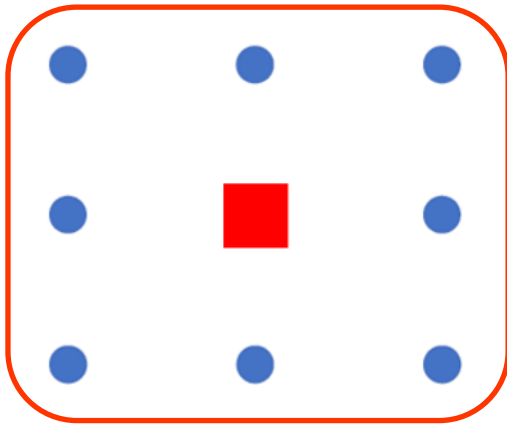
K-means clustering의 학습과정



중심을 군집 경계에 맞게 수정

Non-Hierarchical clustering

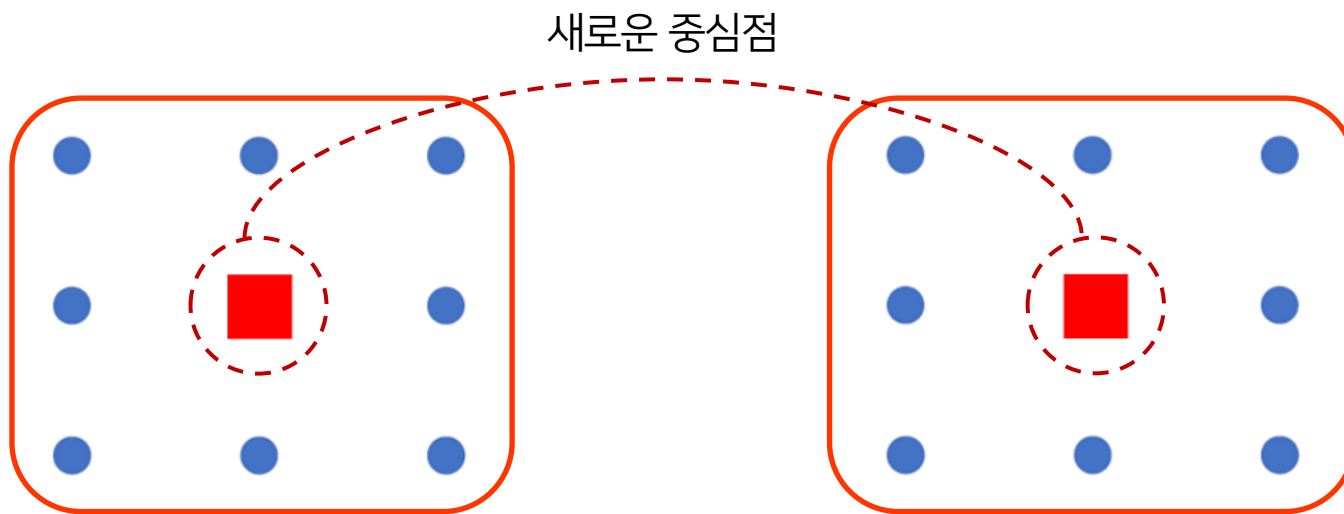
K-means clustering의 학습과정



다시 모든 개체들을 가장 가까운 중심에 군집으로 할당

Non-Hierarchical clustering

K-means clustering의 학습과정



스텝을 반복 해도 결과가 바뀌지 않거나, 정한 반복 수를 채우면 종료

Non-Hierarchical clustering

K-means clustering의 한계점

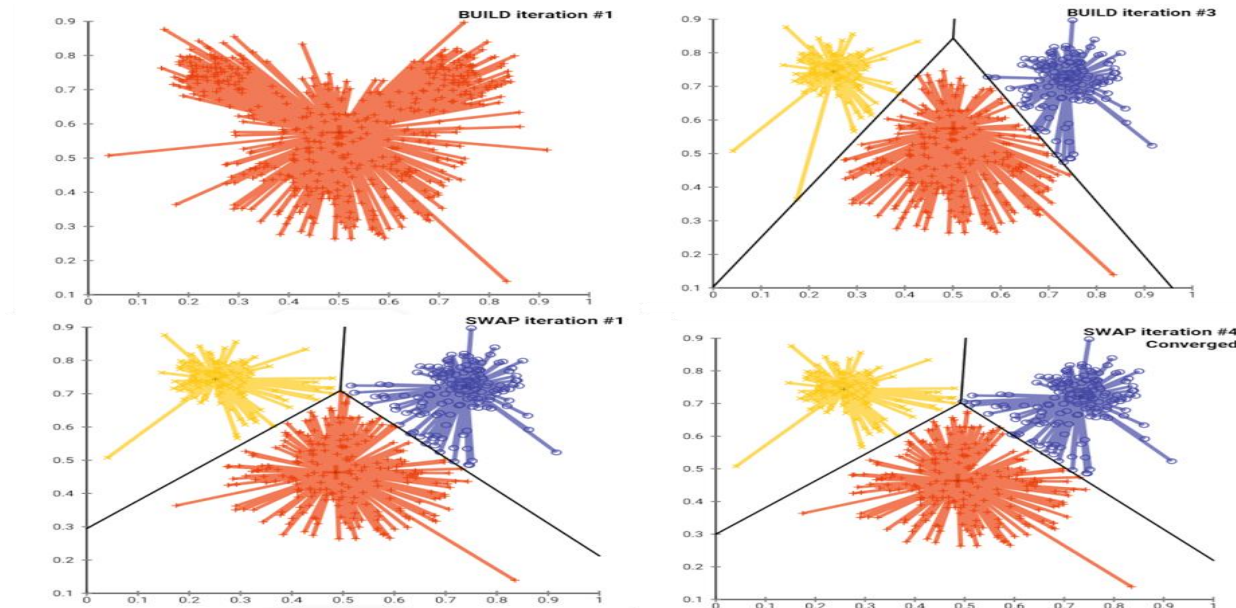
1. 수치형 변수에만 적용 가능
데이터간 **유클리드 거리** 계산 필요

2. Global Optimum이 아닌 **Local Optima**에 빠질 가능성 존재
초기 중심점을 어디로 설정하는지에 따라 최종 군집의 형태가 달라짐

Non-Hierarchical clustering

K-Medoids Clustering: PAM(Partitioning Around Medoids)

데이터의 중앙값으로 중심점을 이동



중앙값은 평균보다 이상치로부터 강건하며 계산이 빠르다는 장점 존재

Density-Based Clustering

밀도를 사용하여 클러스터링을 진행하는 방법

Centroid(중심점) 기반

중심점과의 거리를
바탕으로 클러스터링 진행

- K-Means
- K-Medoids

V/S

Density(밀도) 기반

데이터가 얼마나
뭉쳐 있는가(밀도)를
고려하여 클러스터링 진행

- GMM
- DBSCAN

Density-Based Clustering

밀도를 사용하여 클러스터링을 진행하는 방법



예시를 통해서 비교해보자!

Centroid(중심점) 기반

Density(밀도) 기반

중심점과의 거리를
바탕으로 클러스터링 진행

V/S

데이터가 얼마나
뭉쳐 있는가(밀도)를
고려하여 클러스터링 진행

- K-Means
- K-Medoids

- GMM
- DBSCAN



Density-Based Clustering

밀도를 사용하여 클러스터링을 진행하는 방법



예시를 통해서 비교해보자!

Centroid(중심점) 기반

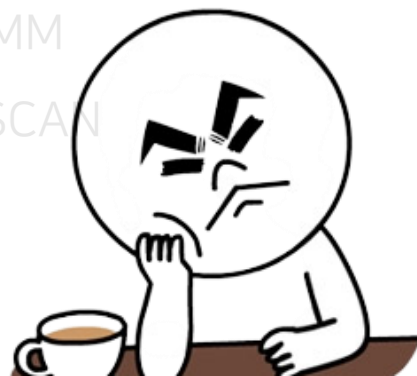
Density(밀도) 기반

중심점과의 거리를
바탕으로 클러스터링 진행

데이터가 얼마나
뭉쳐 있는가(밀도)를
고려하여 클러스터링 진행

- K-Means
- K-Medoids

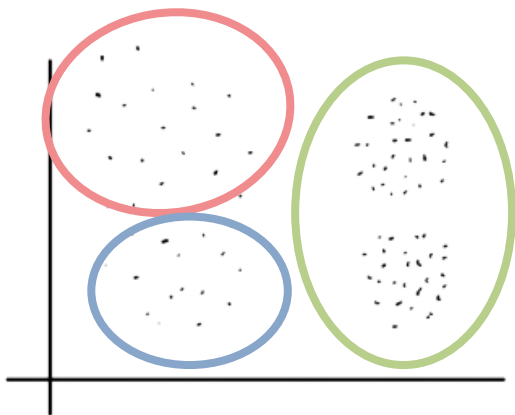
- GMM
- DBSCAN



Density-Based Clustering

밀도를 사용하여 클러스터링을 진행하는 방법

Centroid(중심점) 기반

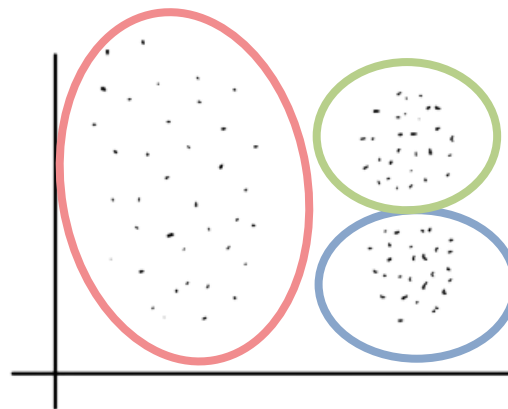


중심점과의 거리 바탕으로
클러스터링 진행

→ 가까이 있는 데이터끼리 묶음

V/S

Density(밀도) 기반



데이터의 뭉쳐 있는 정도 고려
→ 주어진 데이터의 경우

Density-Based Clustering이 더 효과적

Density-Based Clustering

밀도를 사용하여 클러스터링을 진행하는 방법



Centroid(중심점) 기반

Density(밀도) 기반

Density 기반 클러스터링은
이상치 / 일반적인 패턴에서 벗어나는
데이터(noise)들을 클러스터링에 할당 X

→ Anomaly Detection에도 사용 가능

중심점과의 거리 바탕으로

클러스터링 진행

→ 가까이 있는 데이터끼리 묶음

데이터의 밀쳐 있는 정도 고려

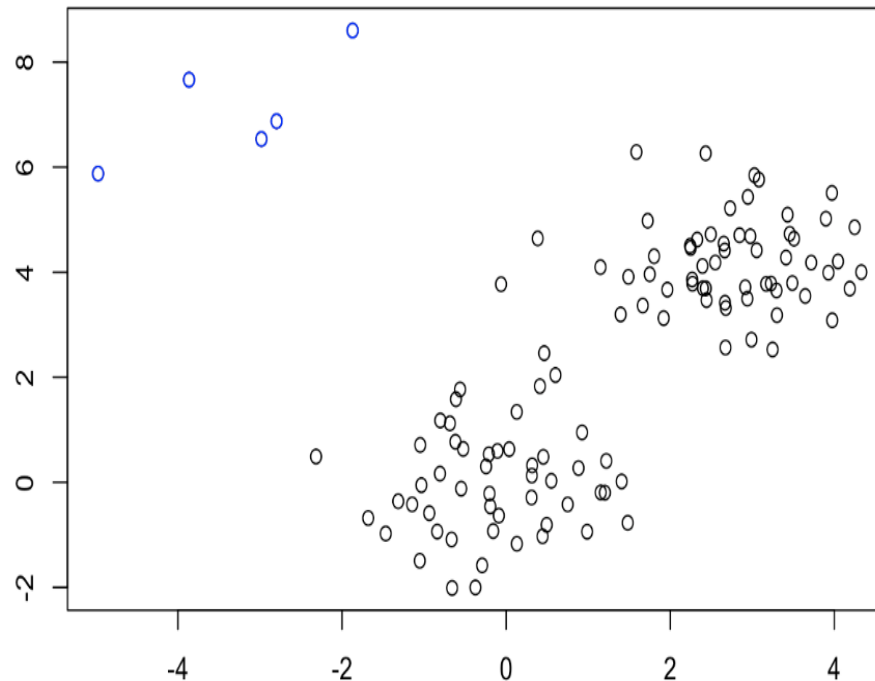
→ 주어진 데이터의 경우

Density-Based Clustering이

더 효과적

Density-Based Clustering

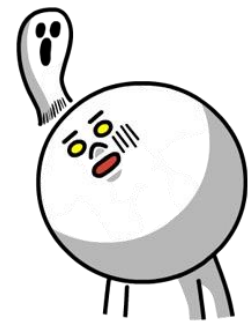
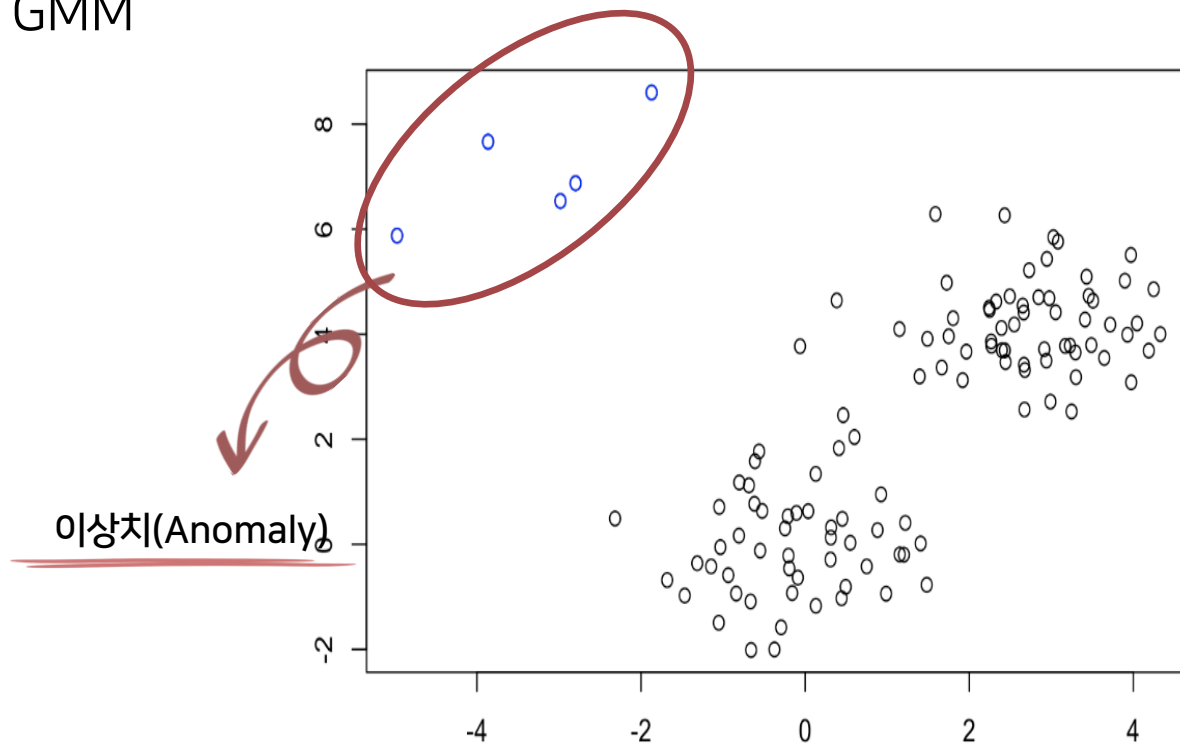
GMM



다음과 같은 데이터가 주어졌다고 가정
이 데이터로부터 **이상치(Anomaly)**를 찾아내고자 함

Density-Based Clustering

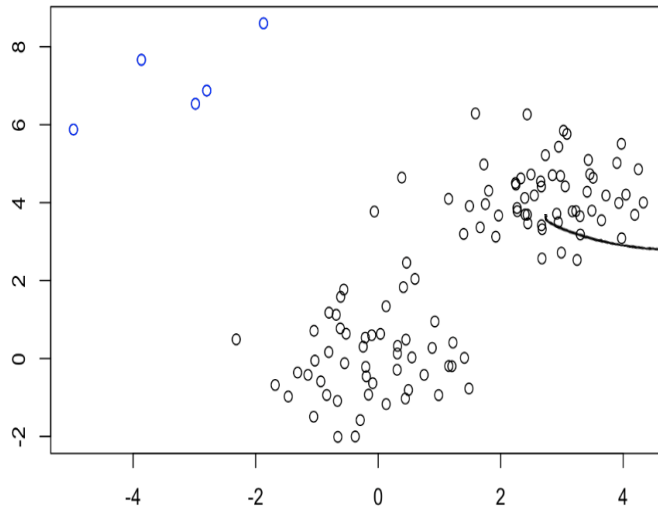
GMM



해당 데이터들은 충분히 **생성될 만하므로** 이를 판별할 수 있는 **명확한 기준**이 필요함

Density-Based Clustering

GMM



$P(x^{(i)})$

이상치(Anomaly)

$$P(x^{(i)}) < \epsilon$$

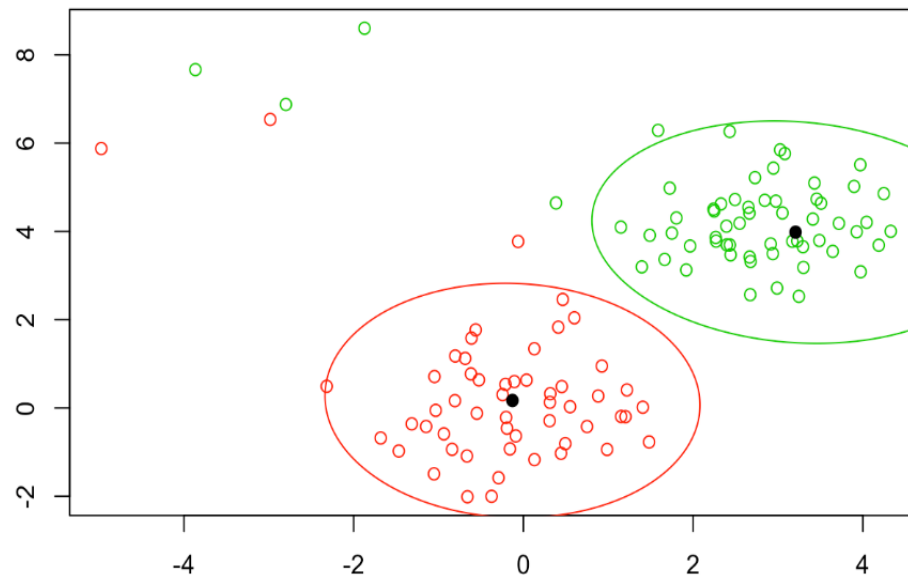
정상 데이터

$$P(x^{(i)}) \geq \epsilon$$

- 명확한 기준 위해 각 데이터가 생성될 확률 $P(x^{(i)})$ 을 모델링
- $P(x^{(i)})$ 가 ϵ 보다 작으면 해당 데이터를 이상치(Anomaly)로 판단

Density-Based Clustering

GMM

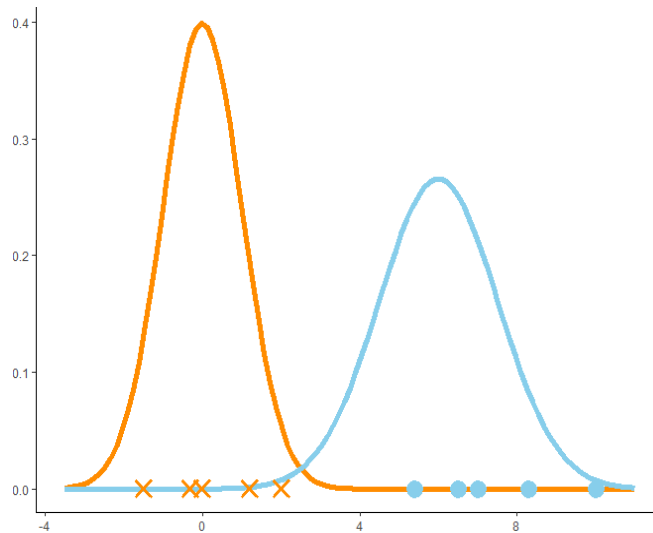


$P(x^{(i)})$ 를 계산하기 위해 GDA에서 처럼 각 데이터들이
2개의 **다변량 정규분포**에 생성되었다고 가정하고자 함

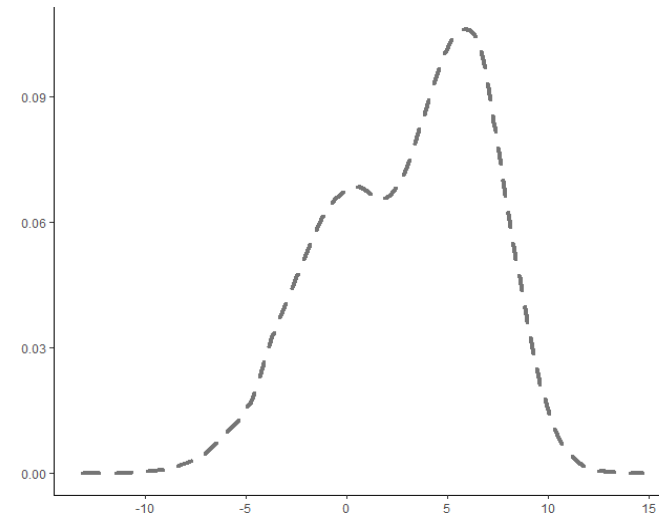


Density-Based Clustering

GMM



GDA



GMM

GMM은 비지도학습

→ GDA와 달리 데이터가 어느 정규분포로부터 생성되었는지 알 수 없음

Density-Based Clustering

GMM

데이터가 어느 정규분포에 의해 형성되었는지 확인하기 위해

① $P(X)$ 가 2개의 정규분포의 혼합모형이라고 가정

② 이 때 각 데이터가 잠재적으로

어떤 정규분포에서 생성되었는지 확률 계산

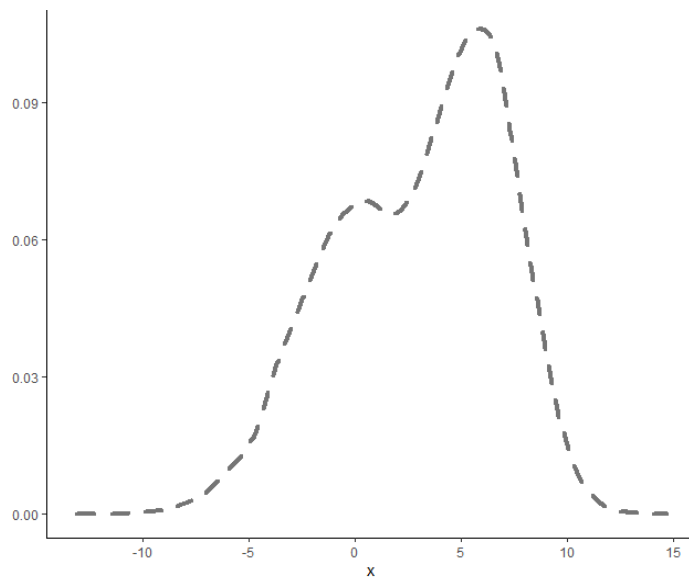
➔ $P(X)$ 를 구할 수 있음

GMM은 비지도학습

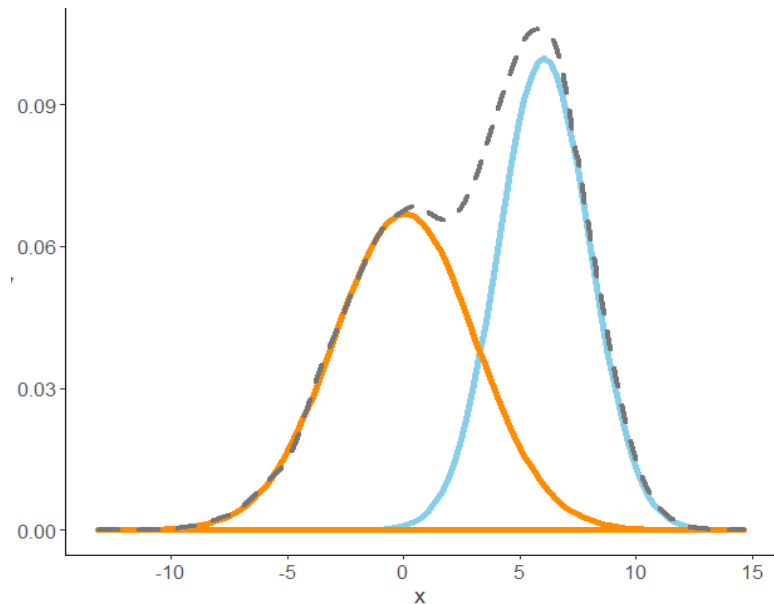
→ GDA와 달리 데이터가 어느 정규분포로부터 생성되었는지 알 수 없음

Density-Based Clustering

GMM



알고리즘



$P(x^{(i)})$ 를 이루는 잠재적인 2개의 정규분포는
경사하강법과 같은 반복적인 알고리즘을 통해 추정 가능

Density-Based Clustering

GMM의 파라미터 추정

GMM의 분포가정

$$z_j \sim \text{multinomial}(\phi_j)$$

$$x^{(i)} | z^{(i)} = j \sim \text{MVN}(\mu_j, \Sigma_j)$$

$P(x^{(i)})$ 를 이루는 잠재적인 2개의 정규분포를 추정하는 알고리즘을 사용하기 위하여 다음과 같이 분포를 가정함

Density-Based Clustering

GMM의 파라미터 추정

GMM의 분포가정

$$z_j \sim \text{multinomial}(\phi_j)$$

$$x^{(i)} | z^{(i)} = j \sim \text{MVN}(\mu_j, \Sigma_j)$$

이때 각 데이터가 잠재적으로 가정한 j번째 정규분포에서 생성될지를

z_j 에 대하여 다항분포를 가정하는 것으로 해결함

Density-Based Clustering

GMM의 파라미터 추정

1

각 데이터 $x^{(i)}$ 에 대한 $w_j^{(i)}$ 를 계산

$$w_j^{(i)} = P(z^{(i)} = j | x^{(i)}) = \frac{P(x^{(i)} | z^{(i)} = j)P(z^{(i)} = j)}{P(x^{(i)})}$$

2

계산한 $w_j^{(i)}$ 을 바탕으로 가정한 분포들의 파라미터 ϕ_j, μ_j, Σ_j 를 업데이트

$$\phi_j = \frac{\sum_{i=1}^m w_j^{(i)}}{m} \quad \mu_j = \frac{\sum_{i=1}^m w_j^{(i)} x^{(i)}}{\sum_{i=1}^m w_j^{(i)}} \quad \Sigma_j = \frac{\sum_{i=1}^m w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^m w_j^{(i)}}$$



1~2

의 과정을 반복



Density-Based Clustering

GMM의 파라미터 추정

GDA MLE 추정량

$$\phi_j = \frac{\sum_{i=1}^m I(y^{(i)} = j)}{m}$$

$$\mu_j = \frac{\sum_{i=1}^m I(y^{(i)} = j) x^{(i)}}{\sum_{i=1}^m I(y^{(i)} = j)}$$

$$\Sigma_j = \frac{\sum_{i=1}^m (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{m}$$

V/S

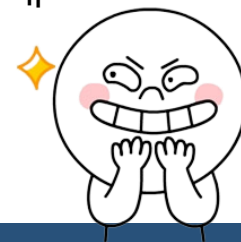
GMM 업데이트 ②단계

$$\phi_j = \frac{\sum_{i=1}^m w_j^{(i)}}{m}$$

$$\mu_j = \frac{\sum_{i=1}^m w_j^{(i)} x^{(i)}}{\sum_{i=1}^m w_j^{(i)}}$$

$$\Sigma_j = \frac{\sum_{i=1}^m (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^m w_j^{(i)}}$$

GMM의 추정 방식과 GDA의 MLE 추정량을 비교해 보았을 때
두 방식에 큰 차이가 없는 것을 확인할 수 있음



Density-Based Clustering

GMM의 파라미터 추정

GDA MLE 추정량

$$\phi_j = \frac{\sum_{i=1}^m I(y^{(i)} = j)}{m}$$

$$\mu_j = \frac{\sum_{i=1}^m I(y^{(i)} = j) x^{(i)}}{\sum_{i=1}^m I(y^{(i)} = j)}$$

$$\Sigma_j = \frac{\sum_{i=1}^m (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{m}$$

V/S

GMM 업데이트 ②단계

$$\phi_j = \frac{\sum_{i=1}^m w_j^{(i)}}{m}$$

$$\mu_j = \frac{\sum_{i=1}^m w_j^{(i)} x^{(i)}}{\sum_{i=1}^m w_j^{(i)}}$$

$$\Sigma_j = \frac{\sum_{i=1}^m (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^m w_j^{(i)}}$$

$w_j^{(i)} = P(z^{(i)} = j | x^{(i)})$ 반복 계산 → **잠재적인 2개의 정규분포**의 파라미터를 얻어냄



GMM은 비지도학습

Density-Based Clustering

GMM의 파라미터



잠재된 분포 추정

GDA MLE 추정량

이처럼 반복적인 알고리즘을 통해

GMM 업데이트 ②단계

$$\phi_j = \frac{\sum_{i=1}^m I(y^{(i)} = j)}{m}$$

잠재되어 있는 분포에 대한 Likelihood를 최대화 가능

$$\mu_j = \frac{\sum_{i=1}^m I(y^{(i)} = j) x^{(i)}}{\sum_{i=1}^m I(y^{(i)} = j)}$$

EM(Expectation-Maximization) 알고리즘!

$$\Sigma_j = \frac{\sum_{i=1}^m (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{m}$$

$$\mu_j = \frac{\sum_{i=1}^m w_j^{(i)} x^{(i)}}{\sum_{i=1}^m w_j^{(i)}}$$

$$\Sigma_j = \frac{\sum_{i=1}^m (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^m w_j^{(i)}}$$

Local Optima로 수렴하기도 하지만

$$w_j^{(i)} = P(z^{(i)} = j | x^{(i)})$$

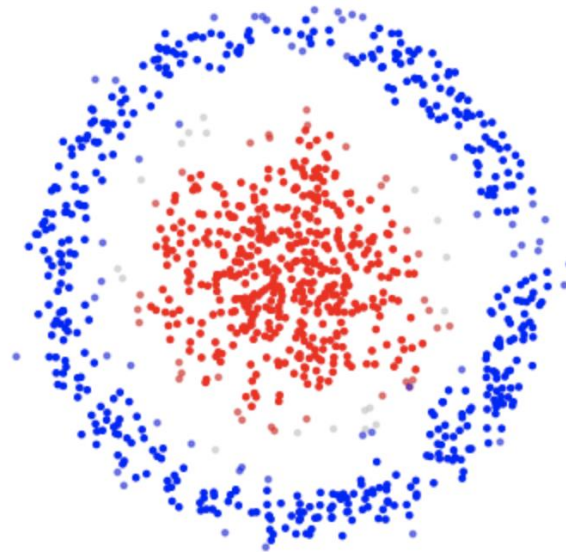
효과적으로 잠재된 분포의 파라미터를 최적화 가능!

GMM은 비지도학습

잠재된 2개의 정규분포의 파라미터를 얻어냄

Density-Based Clustering

DBSCAN



- 밀도 기반의 클러스터링 기법
- 학습 진행 - 군집 내 밀도는 \uparrow 군집에 포함 되지 않은 객체들의 밀도는 \downarrow

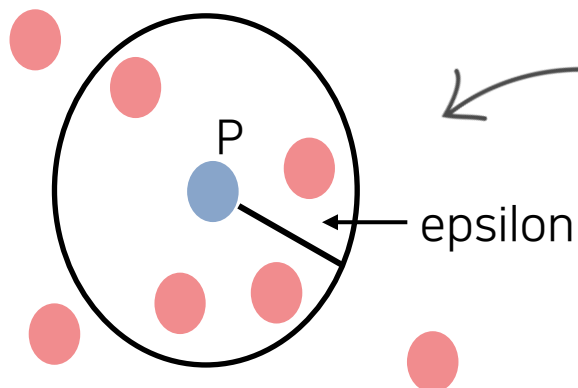
DBSCAN

용어 정리

 ϵ -Neighborhood of a point

$$N_{\epsilon}(p) = \{q \in D \mid \text{dist}(p, q) \leq \epsilon\}$$

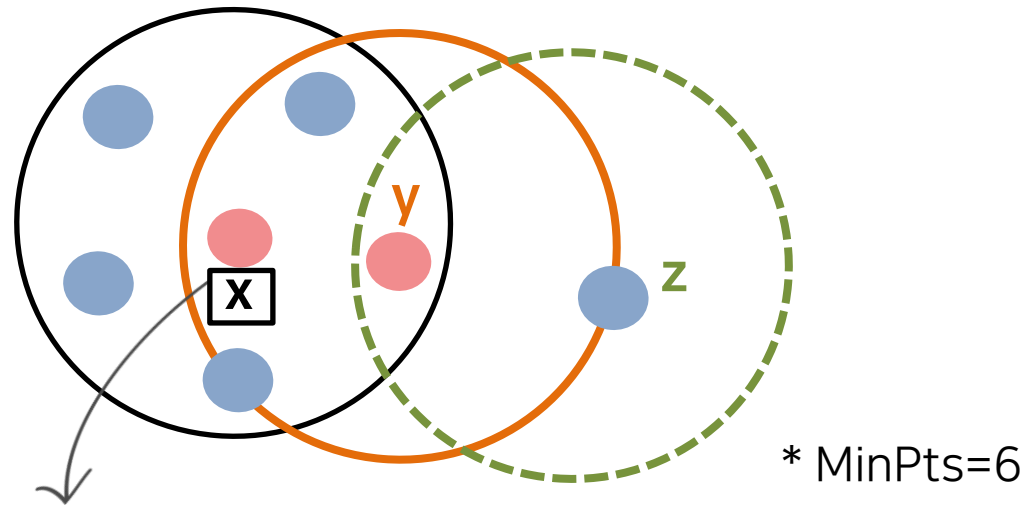
P의 ϵ -neighborhood는 p와의 거리가 ϵ 보다 작은 점 q들의 집합

**minPts**

점 p를 중심으로 군집을 이루려면
P의 ϵ -neighborhood q들이
최소 minPts 이상은 있어야 함

DBSCAN

용어 정리

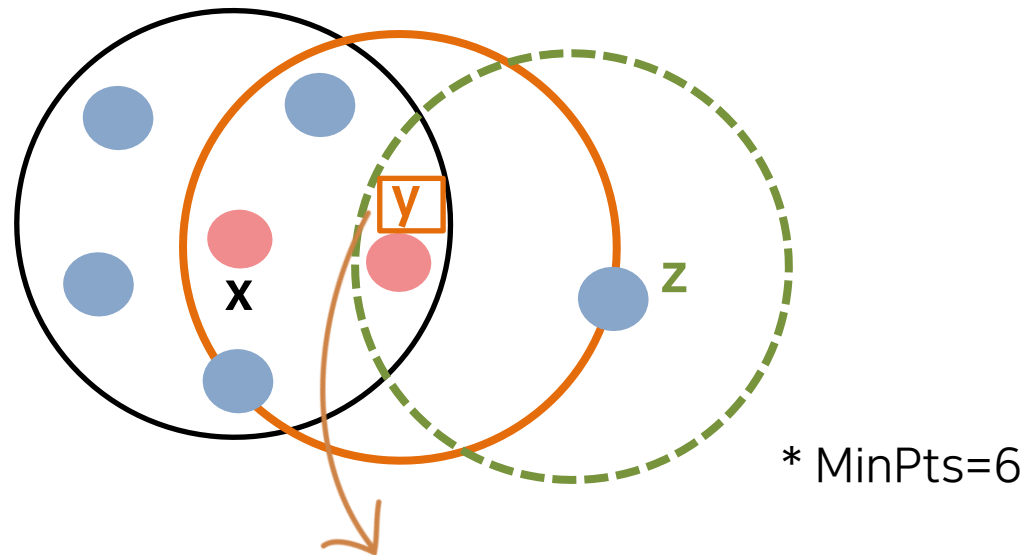
**Core Point**한 점의 ϵ 반경 내에

minPts 이상의 개체가 포함된 점

→ 해당 점을 중심으로 군집 형성 가능

DBSCAN

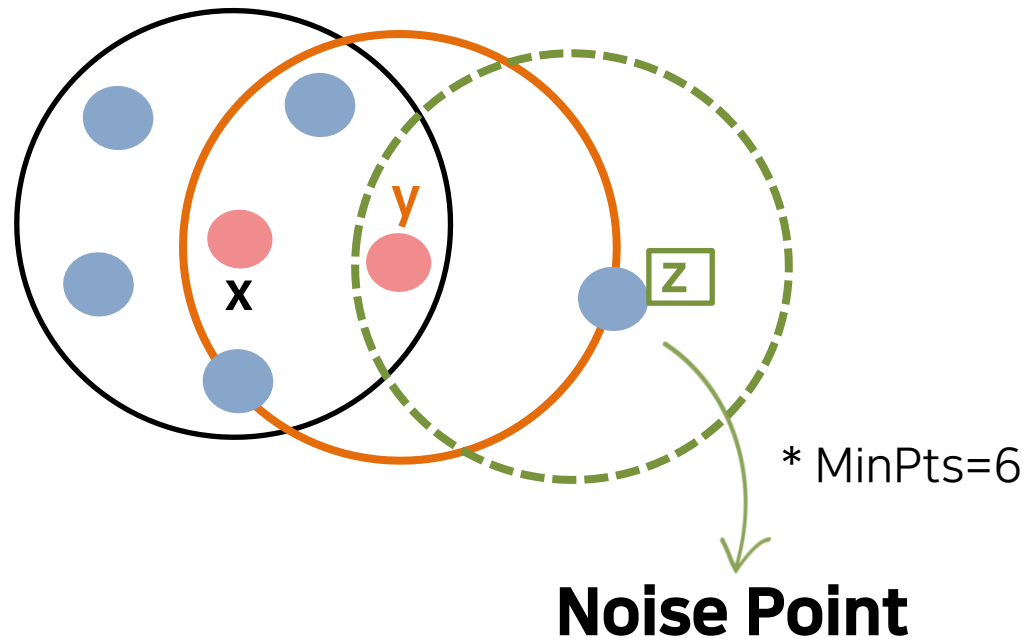
용어 정리

**Border Point**

한 점의 ϵ 반경 내에
minPts보다 적은 수의 개체를 포함하지만
그 중 적어도 하나가 Core point인 경우

DBSCAN

용어 정리

**Noise Point**

Core도 Border도 아닌 point

(minPts 이하의 개체, 주변 점에 core point 없음)

DBSCAN

밀도 관점에서의 연결

Directly Density-Reachable

점 p 가 점 q 로부터 밀도 관점에서 직접적으로 연결 가능하다



Reachability $p \in N_{\epsilon}(q)$

p 는 q 의 epsilon neighborhood에 속해야 함



Core Point Condition $N_{\epsilon}(q) \geq MinPts$

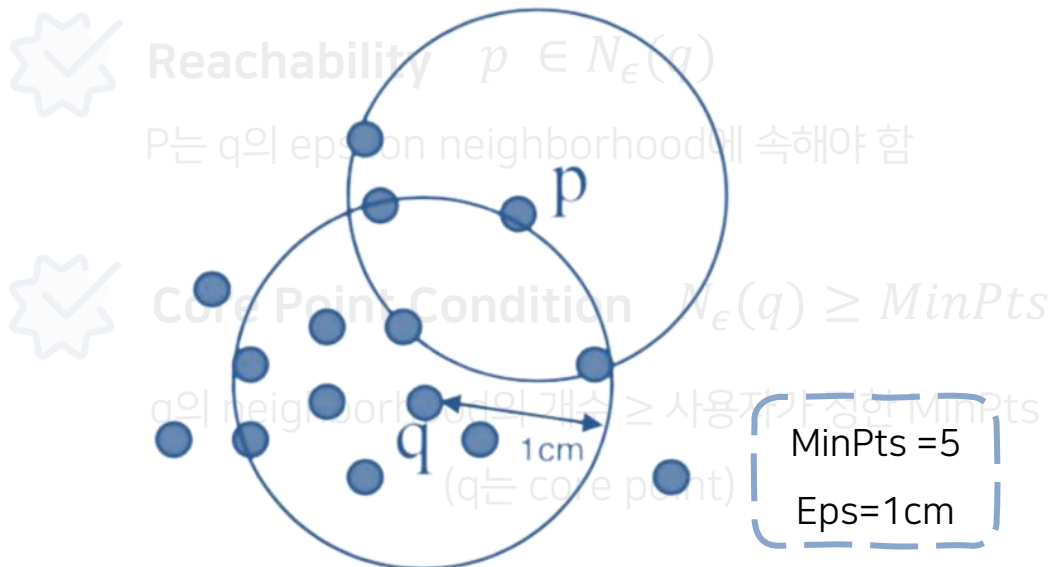
q 의 neighborhood의 개수 \geq 사용자가 정한 MinPts

(q 는 core point)

DBSCAN

밀도 관점에서의 연결

Directly Density-Reachable

점 p 가 점 q 로부터 밀도 관점에서 직접적으로 연결 가능하다

DBSCAN

밀도 관점에서의 연결

Density-Reachable

점 p 가 점 q 로부터 밀도 기반 도달 가능한 관계에 있다

- 점 p 가 점 q 의 ϵ 반경 안에 위치하지 못하더라도
 - 점 p 와 q 사이에 점 p_1, p_2, \dots, p_n 존재
- 모든 점 p_{i+1} 이 p_i 로부터 직접적으로 Directly density-reachable

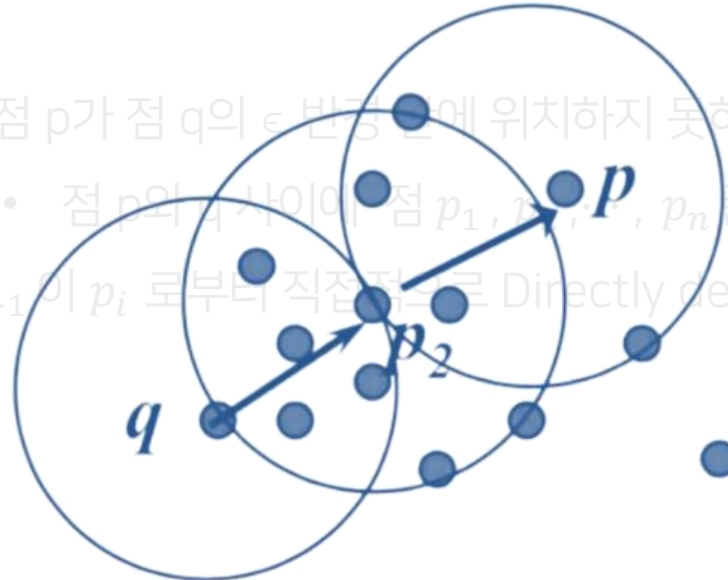
DBSCAN

밀도 관점에서의 연결

Density-Reachable

점 p 가 점 q 로부터 밀도 기반 도달 가능한 관계에 있다

- 점 p 가 점 q 의 ϵ 반경 안에 위치하지 못하더라도
- 점 p 와 q 사이에 점 p_1, p_2, \dots, p_n 존재
- 모든 점 p_{i+1} 이 p_i 로부터 직접적으로 Directly density-reachable



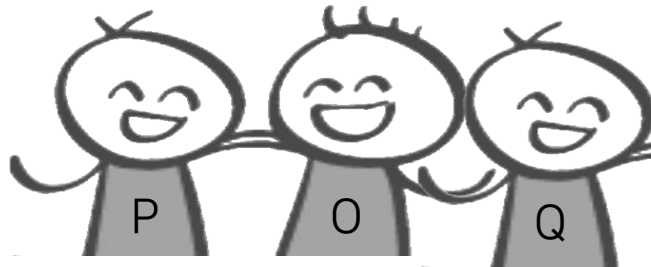
DBSCAN

밀도 관점에서의 연결

Density-Connected

점 p 가 점 q 와 연결된 관계에 있다

두 점 p, q 가 모두 어떤 점 o 로부터 반경 내
MinPts 조건 하에 Density-reachable한 경우

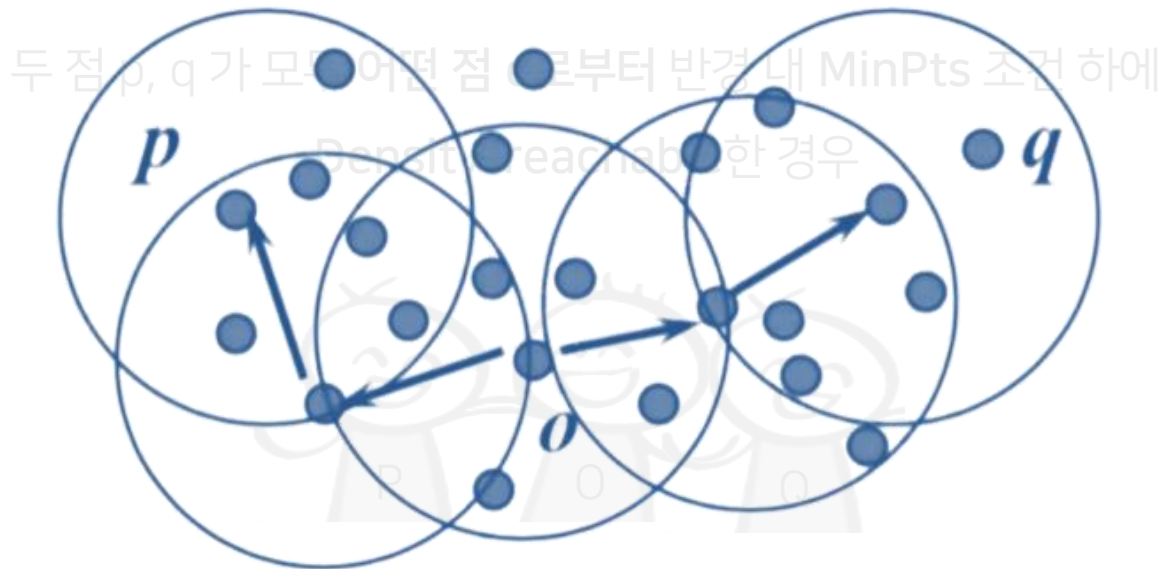


친구의 친구는 친구

DBSCAN

밀도 관점에서의 연결

Density-Connected

점 p 가 점 q 와 연결된 관계에 있다

친구의 친구는 친구

DBSCAN

학습 과정

① 임의의 데이터 포인트 선택

② ϵ 반경 내 minPts 개수 이상의 데이터 존재 여부에 따라
core point와 border point로 데이터를 구분

③ ϵ 반경 안에 있는 코어점들 서로 연결하여 군집 형성
border point들을 어느 하나의 군집에 할당



학습이 끝난 후에도 군집에 속하지 않은 포인트 → 노이즈

DBSCAN

학습 과정



- 하이퍼파라미터 epsilon과 minPts의 적절한 값 알 수 없음
→ heuristic하게 결정해야 함
- Dataset이 바뀔 때마다 적절한 epsilon과 minPts 값도 달라짐
② ϵ 반경 내 minPts 개수 이상의 데이터 존재 여부에 따라
core point와 border point로 데이터를 구분

③ ϵ 반경 안에 있는 코어점들 서로 연결하여 군집 형성
border point들을 어느 하나의 군집에 할당



학습이 끝난 후에도 군집에 속하지 않은 포인트 → 노이즈

DBSCAN

학습 과정



- 하이퍼파라미터 epsilon과 minPts의 적절한 값 알 수 없음
→ heuristic하게 결정해야 함
- Dataset이 바뀔 때마다 적절한 epsilon과 minPts 값도 달라짐

② ϵ 반경 내 minPts 개수 이상의 데이터 존재 여부에 따라

core point와 border point로 데이터를 구분



③ ϵ 반경 안에 있는 코어점들 서로 연결하여 군집 형성

border point들을 어느 하나의 군집에 할당

HDBSCAN

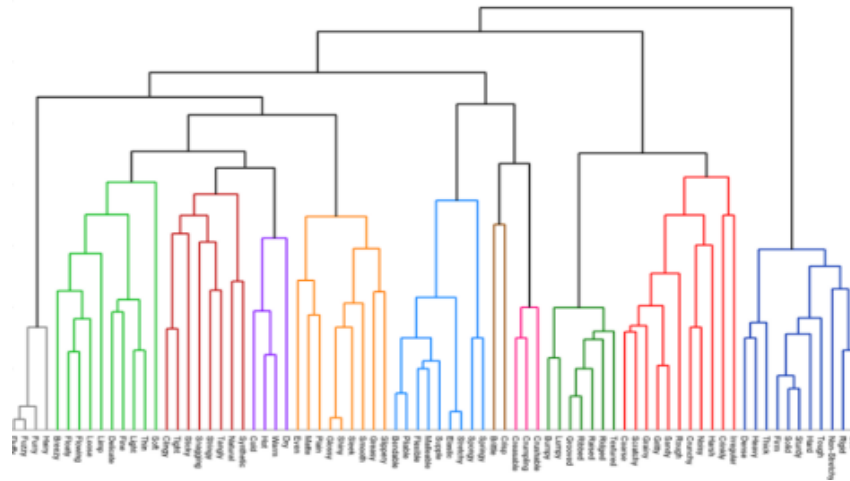
- Dataset의 계층적인 구조 반영

• 학습 epsilon 값들을 필요로 하지 않음 → 노이즈



Hierarchical Clustering

계층적 클러스터링



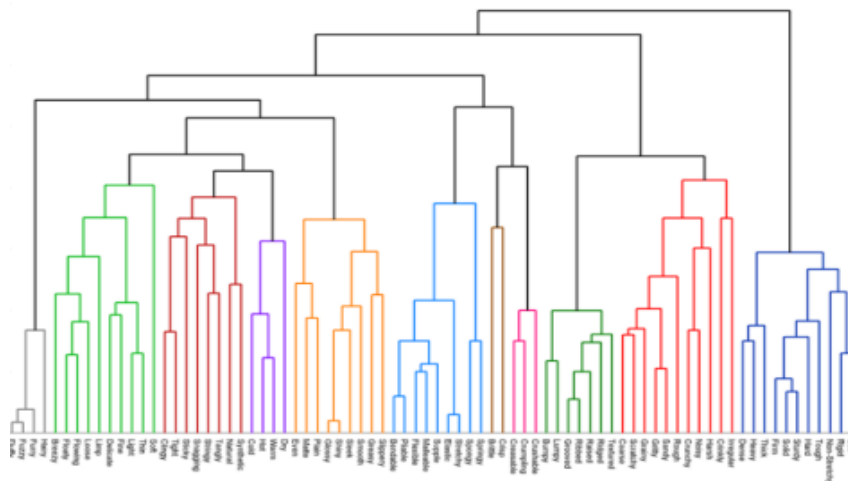
덴드로그램(Dendrogram)

트리모델을 이용해서 개별 개체들을 순차적이고 계층적으로 유사한 개체 혹은 그룹과 함께 클러스터를 만들어 주는 알고리즘

클러스터의 개수를 사전에 정하지 않고도 학습이 수행 가능하다는 장점

Hierarchical Clustering

계층적 클러스터링



덴드로그램(Dendrogram)

트리모델을 이용해서 개별 개체들을 순차적이고 계층적으로

덴드로그램을 수행하기 위해 모든 개체들 간의 유사한 개체들은 그룹화할 수 있는 알고리즘

거리나 **유사도**가 이미 계산되어 있어야 함

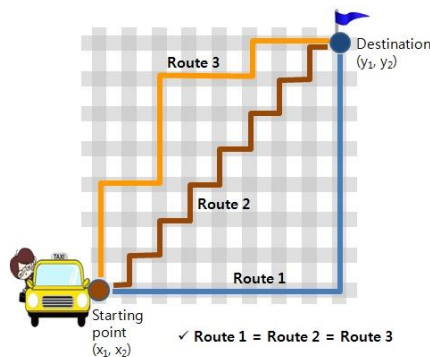
클러스터의 개수를 사전에 정하지 않고도 학습이 수행 가능하다는 장점

Hierarchical Clustering

계층적 클러스터링

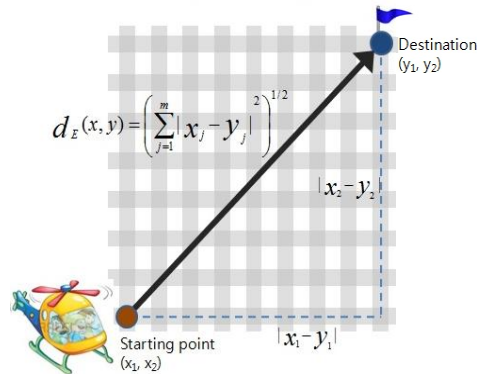
맨해탄 거리(L1)

좌표 축 방향으로
이동할 때 계산되는 거리



유클리드 거리 (L2)

두 관측치 사이의
직선 최단거리



마할라노비스 거리

변수 내 분산, 공분산을
모두 반영해 계산한 거리

$$d_{Mahalanobis}(X, Y)$$

$$= \sqrt{(\vec{X} - \vec{Y})^T \Sigma^{-1} (\vec{X} - \vec{Y})}$$

where Σ^{-1} is the inverse of covariance matrix

Hierarchical Clustering

계층적 클러스터링 | 덴드로그램(Dendrogram)

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

유사도 값을 바탕으로 생성한 거리행렬식

Hierarchical Clustering

계층적 클러스터링 | 덴드로그램(Dendrogram)

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

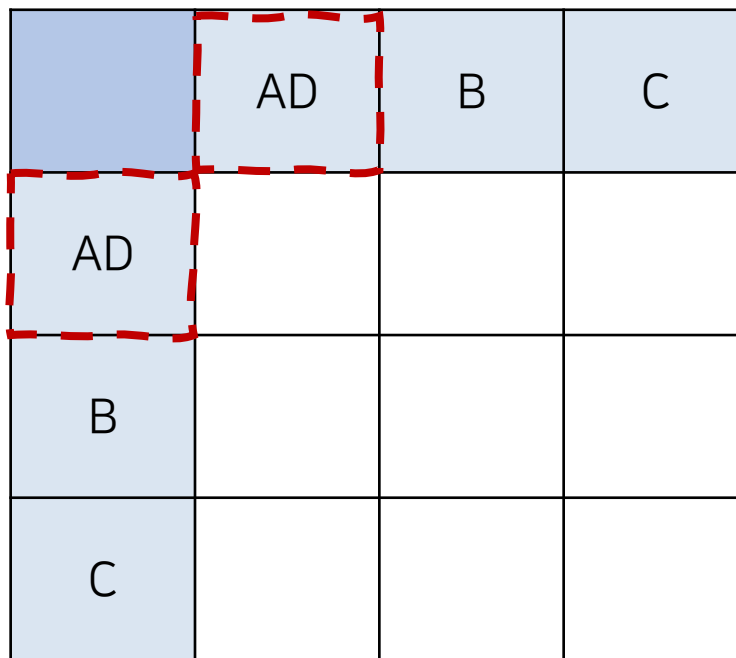
A와 D로 묶기!

서로 가장 가까운 관측치 찾아 묶기

Hierarchical Clustering

계층적 클러스터링 | 덴드로그램(Dendrogram)

A와 D로 묶기!



	AD	B	C
AD			
B			
C			

서로 가장 가까운 관측치 찾아 묶기

Hierarchical Clustering

계층적 클러스터링 | 덴드로그램(Dendrogram)



AD와 B, C사이의 거리는 어떻게 결정해야 할까요?

A와 D로 묶기!

최소기준

최대기준

평균기준

	AD	B	C
AD			
B			
C			

서로 가장 가까운 관측치 찾아 묶기

Hierarchical Clustering

계층적 클러스터링 | 덴드로그램(Dendrogram)

AD와 B, C사이의 거리는 어떻게 결정해야 할까요?

A와 D로 묶기!

최소기준

묶이기 전 각각 개체와
나머지 개체의 거리 중 가장
짧은 거리로 대체



A와 B 사이거리 = 20
D와 B 사이거리 = 25

최대기준

묶이기 전 각각 개체와
나머지 개체의 거리 중
가장 긴 거리로 대체



A와 B 사이거리 = 20
D와 B 사이거리 = 25

평균기준

최소 기준으로 계산된 거리와
최대 기준으로 계산된 거리의
평균으로 대체



$20(\text{A와 B거리}) +$
 $25(\text{D와 B거리}) / 2 = 22.5$

서로 가장 가까운 관측치 찾아 묶기

Hierarchical Clustering

계층적 클러스터링 | 덴드로그램(Dendrogram)

서로 가장 가까운 관측치 찾아 묶기



기준을 결정 후 새로운 행렬에 채워 넣기



위 과정을 반복하여 모든 개체를
하나로 묶으면 학습 종료

Hierarchical Clustering

계층적 클러스터링 | 덴드로그램(Dendrogram)



단점

서로 가장 가까운 관측치 찾아 묶기
단계를 거칠 때 마다 **기준에 맞춰 계산**을 해야함



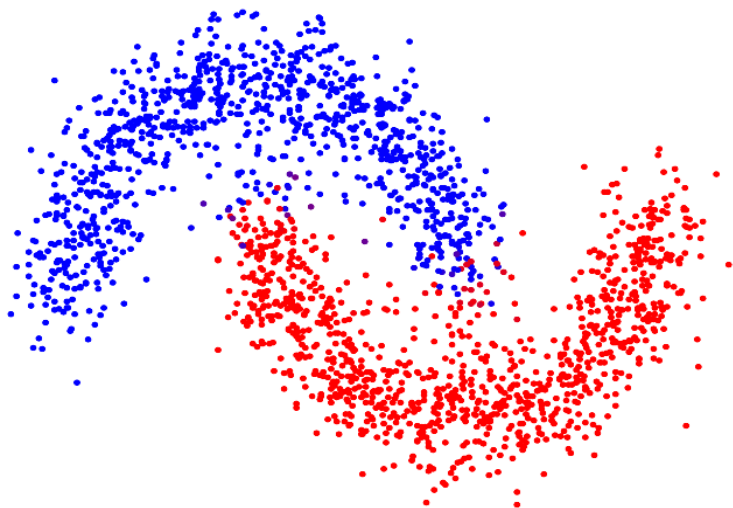
기준을 결정 후 새로운 행렬에 채워 넣기

대용량 데이터의 경우 **많은**
연산 시간과 컴퓨팅 파워가 소모된다.

하나로 묶으면 학습 종료

Other Clustering

Spectral Clustering



장점

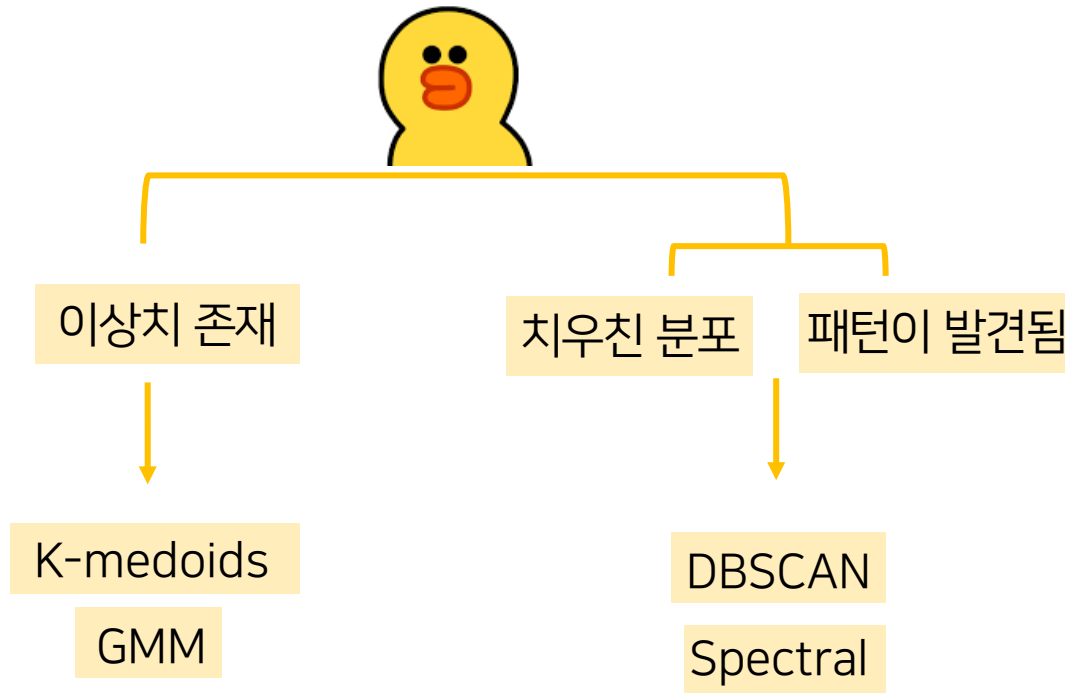
이중 나선 형태에서 효과적



단점

- 긴 연산 시간
- 최적의 클러스터 개수를 결정하기 어려움

Other Clustering



각 클러스터링 기법의 특징을 이해하고 상황에 맞게 적용하는 것이 매우 중요함

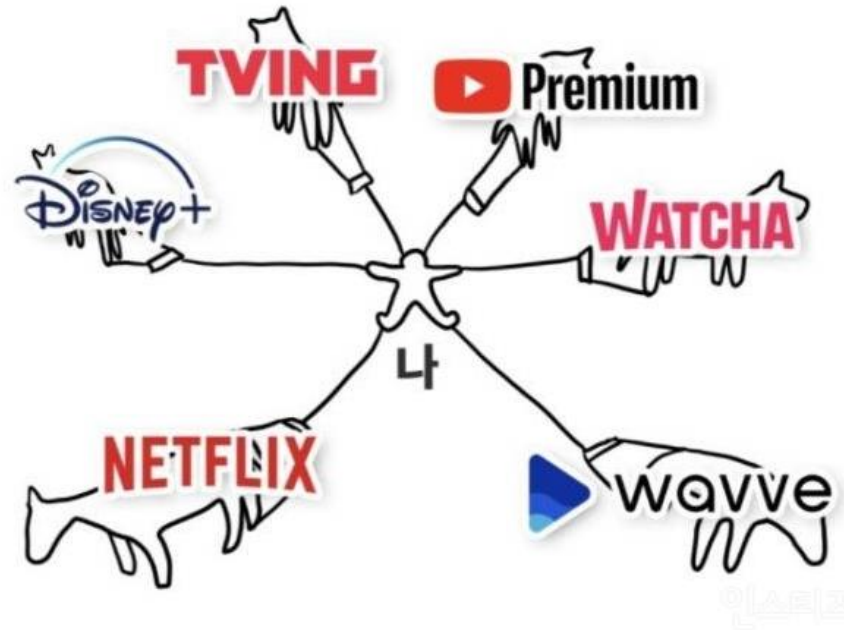


2

추천시스템

Recommendation System

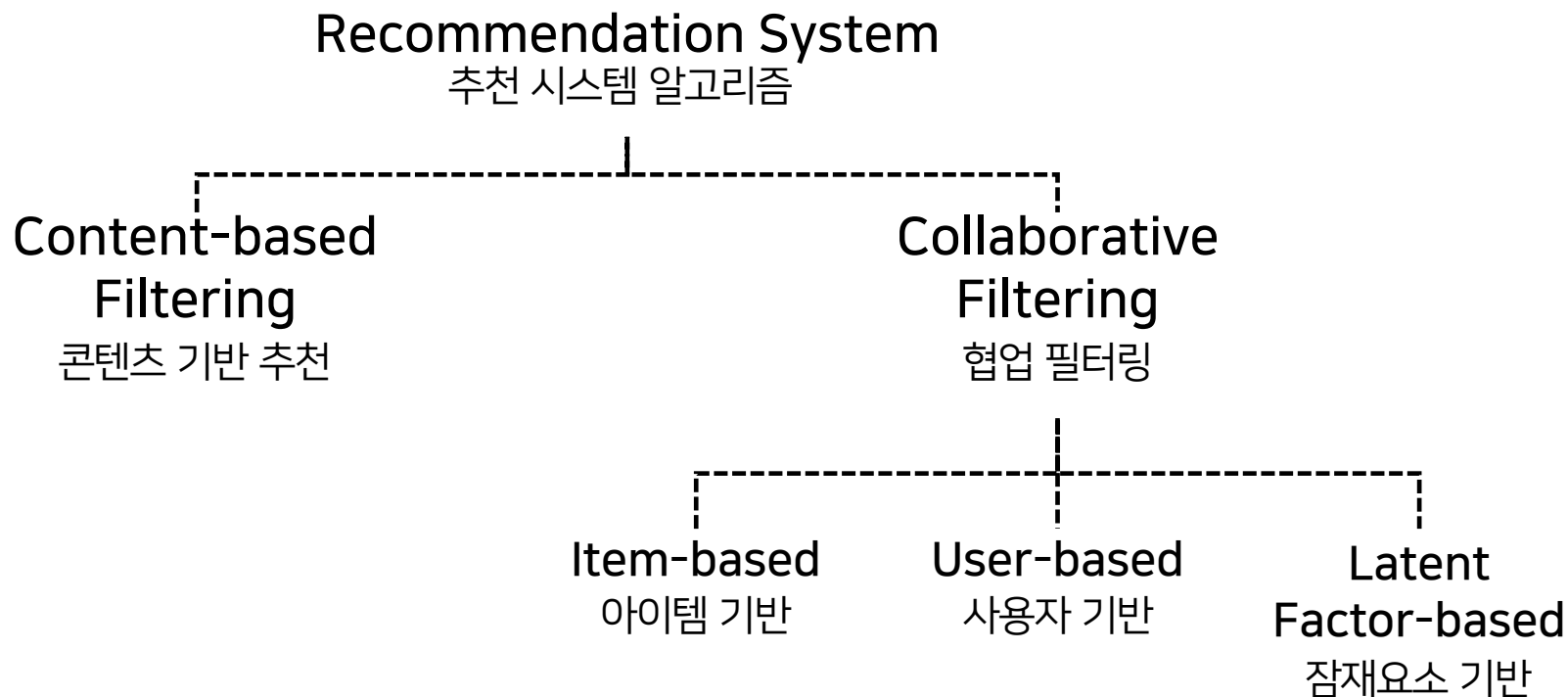
추천 시스템의 필요성



정보량이 증가하고, 관련 플랫폼이 많아지며
개인의 취향에 맞는 콘텐츠를 추천할 필요성 ↑

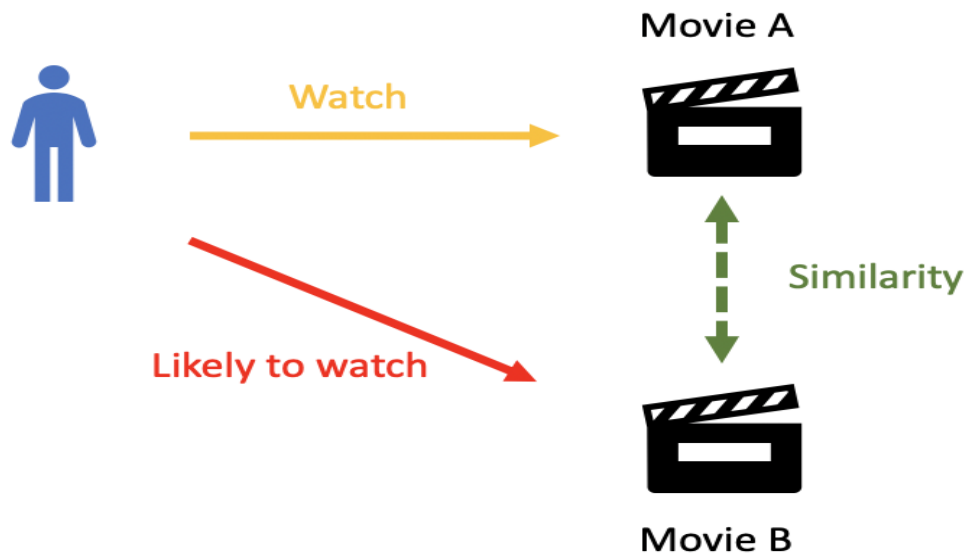
Recommendation System

추천시스템의 종류



Content-Based Filtering

컨텐츠 기반 추천이란?

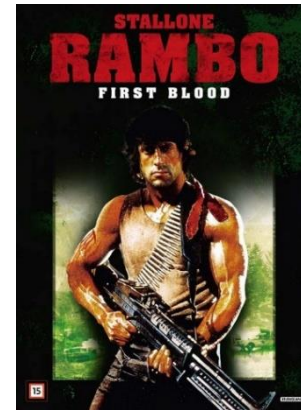
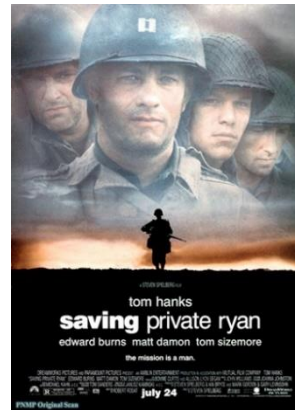
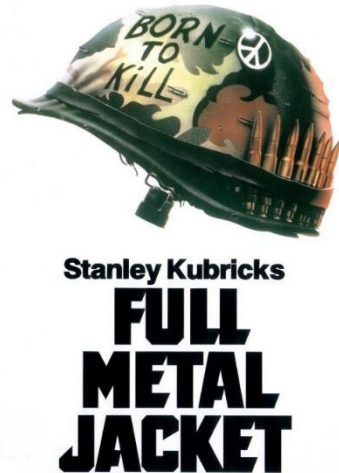


- 해당 **컨텐츠에 대한 정보**만을 이용해 추천을 실시하는 방법
 - 사용자가 과거에 소비했던 컨텐츠 특성을 분석
 - 유사한 특성을 지닌 컨텐츠를 사용자에게 추천

Content-Based Filtering

컨텐츠 기반 추천

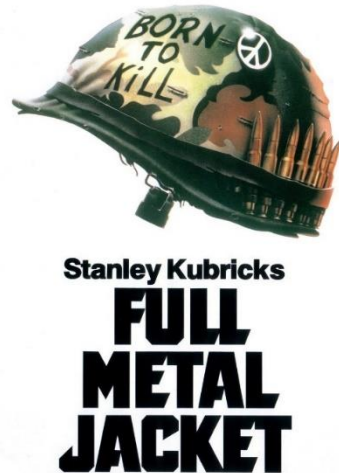
장르 기준 추천



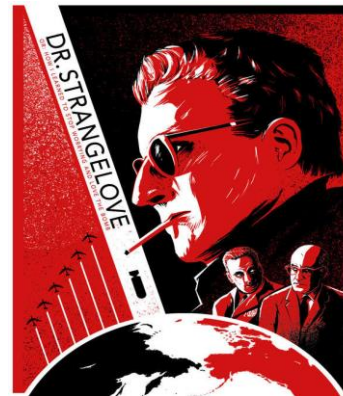
- 사용자의 영화 시청을 바탕으로 다른 영화를 추천한다고 가정
- 사용자가 시청한 영화를 설명하는 메타데이터를 사용하게 될 것
(장르, 배우, 줄거리, 감독 등)

Content-Based Filtering

컨텐츠 기반 추천



감독 기준 추천



- 사용자의 영화 시청을 바탕으로 다른 영화를 추천한다고 가정
- 사용자가 시청한 영화를 설명하는 메타데이터를 사용하게 될 것
(장르, 배우, 줄거리, 감독 등)

Content-Based Filtering

컨텐츠 기반 추천 유사도

기존 사용자가 소비한 콘텐츠와 추천하려는 콘텐츠가
얼마나 유사한지 계산이 필요



- 영화 줄거리가 비슷한 영화를 추천한다고 가정
 - 영화의 줄거리는 **비정형의 텍스트 데이터**
- 합리적인 유사도를 측정하는 것이 매우 어려울 것

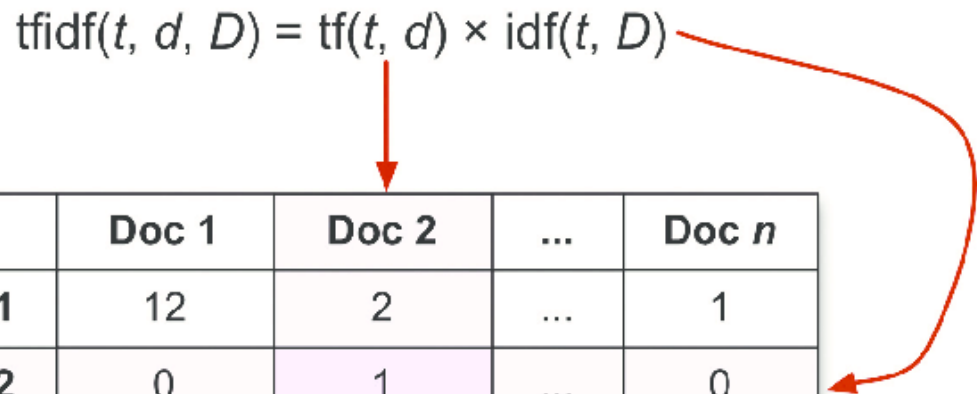


TF-IDF 방법론

Content-Based Filtering

TF-IDF (Term Frequency-Inverse Document Frequency)

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D)$$



	Doc 1	Doc 2	...	Doc n
Term(s) 1	12	2	...	1
Term(s) 2	0	1	...	0
...	
Term(s) n	0	6	...	3

어느 문서에서도 많이 나오는 단어에 **페널티**를 주고
동시에 그 문서를 대표하는 단어들을 추출해내는 방법

Content-Based Filtering

TF-IDF (Term Frequency-Inverse Document Frequency)

$$TF - IDF = TF \times \log \frac{n_D}{1 + n_t}$$

n_D : 전체 문서 수

n_t : 단어 t 가 나온 문서 수

- TF (Term Frequency) : 단어 t 가 하나의 문서에서 나온 빈도수
- IDF (Inverse Document Frequency) : 전체 문서 중 단어 t 가 나온 문서 수의 역수

Content-Based Filtering

TF-IDF (Term Frequency-Inverse Document Frequency)

$$TF - IDF = TF \times \log \frac{n_D}{1 + n_t}$$

IDF 값에 log 취함



전체 문서수가 많을 때 TF-IDF 값이 너무 커지는 것을 방지하기 위함

Content-Based Filtering

TF-IDF (Term Frequency-Inverse Document Frequency)

$$TF - IDF = TF \times \log \frac{n_D}{1 + n_t}$$

IDF의 분모에서 n_t 대신 $1 + n_t$ 사용



나이브 베이즈 모델의
라플라스 평활법과 같은 개념

어떤 단어가 모든 문서에 들어가게 되어
 $\log IDF = \log 1 = 0$ 으로 계산되어($n_D = n_t$)
TF-IDF가 0이 되는 것을 방지하기 위함

Content-Based Filtering

TF-IDF (Term Frequency-Inverse Document Frequency)

$$TF-IDF = TF \times \log \frac{n_D}{1 + n_t}$$

TF-IDF 값이 높은 단어
(t가 문서를 대표)

단어 t가 한 문서에서
자주 등장

다른 문서에서는 많이
등장하지 않아야 함

Content-Based Filtering

TF-IDF (Term Frequency-Inverse Document Frequency)

해당 영화 줄거리에 '베트남 전쟁'이라는 단어가 자주 등장하고,
다른 영화의 줄거리에서는 잘 등장하지 않다고 가정



'베트남 전쟁'



TF-IDF값이 높게 나타남,
중요한 키워드로 발탁

Content-Based Filtering

TF-IDF (Term Frequency-Inverse Document Frequency)



'베트남 전쟁'

컨텐츠에서 중요한 단어(feature)를 추출하여
→ 다른 컨텐츠와 비슷한 컨텐츠를 찾아서 추천을 진행

METAL
JACKET

중요한 키워드로 발탁

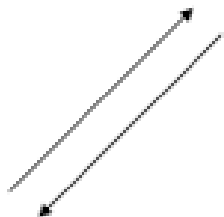
해당 영화 줄거리에 '베트남 전쟁'이라는 단어가 자주 등장하고, 다른 영화의
줄거리에서는 잘 등장하지 않다고 가정

Content-Based Filtering

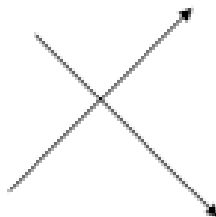
유사도 계산

코사인 유사도

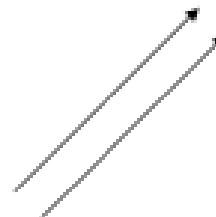
$$\text{cosine similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n (A_i \times B_i)}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$



코사인 유사도 : -1



코사인 유사도 : 0



코사인 유사도 : 1

TD-IDF로 뽑아낸 대표 단어들을 수치형 벡터로 변환(인코딩)

→ 변화된 벡터 간 유사도를 계산 가능

Content-Based Filtering

유사도 계산

코사인 유사도

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



예) 유사한 단어들이 많이 등장하는 줄거리를 가진 영화를 추천하는 알고리즘 구현

→ 이 과정에서 분석자와 개발자의 주관이 개입됨

Content-Based Filtering

콘텐츠 기반 추천의 장점과 한계

장점

cold-start 현상에 크게 구애 받지 않음

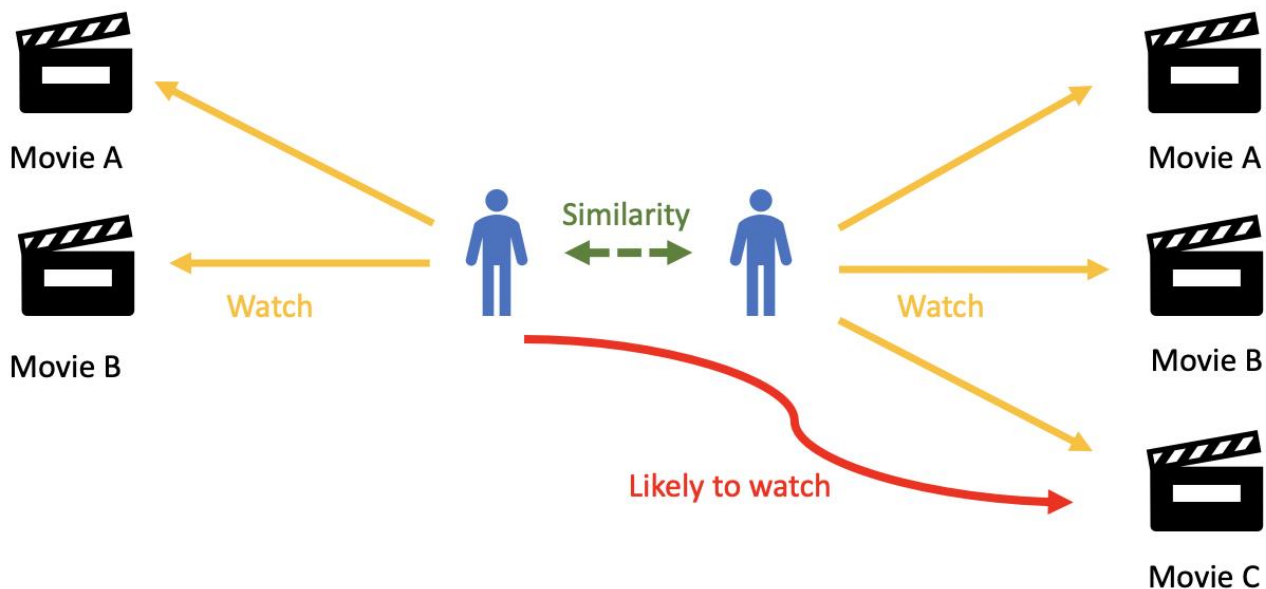
서비스 초반 누적 데이터의 부족으로 제대로 된 추천이 어려운 문제

한계

- ① 새로운 사용자에게는 추천이 불가능 → 고질적인 문제
- ② 메타데이터로부터 주요 feature를 추출하기 어려움

Collaborative Filtering

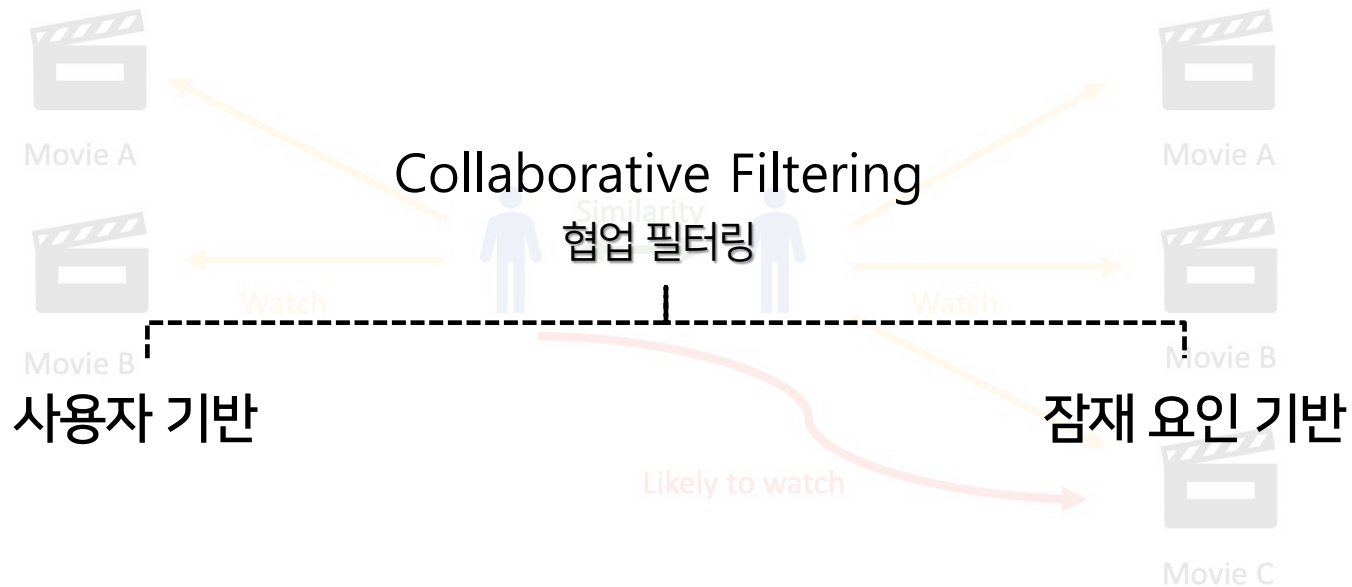
협업 필터링



구입내역, 선호도, 만족도를 기반으로 사용자 혹은 제품 간의 협업(상호작용)을 통하여
비슷한 성향을 가진 사용자가 선호하는 제품 추천

Collaborative Filtering

협업 필터링



구입내역, 선호도, 만족도를 기반으로 사용자 혹은 제품 간의 협업(상호작용)을 통하여
비슷한 성향을 가진 사용자가 선호하는 제품 추천

Collaborative Filtering

사용자 기반 협업 필터링

	탐건:매버릭	헤어질 결심	헌트	한산	외계인
준서	5	4	4	3	1
희나	1	0	1	3	4
수빈	4	4	2	5	3
석주	4	2	3	2	2
현우	5	3	1	2	?

- 사용자와 제품 간의 상호작용 데이터 표현
- 숫자로 표현되기 위해 행렬의 형태 사용

Collaborative Filtering

사용자 기반 협업 필터링

	탐건:매버릭	헤어질 결심	헌트	한산	외계인
준서	5	4	4	3	1
희나	1	0	1	3	4
수빈	4	4	2	5	3
석주	4	2	3	2	2
현우	5	3	1	2	?

사용자가 매긴 제품에 대한 선호도를 바탕으로 시스템 설계

→ 평점행렬 (유저와 아이템 간의 평점 행렬) 사용!

Collaborative Filtering

사용자 기반 협업 필터링

	탑건:매버릭	헤어질 결심	헌트	한산	외계인
준서	5	4	4	3	1
희나	1	0	1	3	4
수빈	4	4	2	5	3
석주	4	2	3	2	2
현우	5	3	1	2	?



준서, 희나, 수빈, 석주 각 영화들에 매긴 평점을 바탕으로
현우가 아직 시청하지 않은 '외계인'에 대한 평점을 예측

Collaborative Filtering

사용자 기반 협업 필터링

피어슨 상관계수를 사용하여 유사도 계산

$$\text{Similarity}(a, b) = \frac{\sum_{p \in P} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P} (r_{b,p} - \bar{r}_b)^2}}$$

Collaborative Filtering

사용자 기반 협업 필터링

	준서	희나	수빈	석주
현우	0.7171372	-0.2714484	0.4265617	0.5606119

Rating Prediction

$$\frac{(0.717 * 1) + (-0.271 * 4) + (0.427 * 3) + (0.561 * 2)}{(0.717 - 0.271 + 0.427 + 0.561)} = 1.42$$



구한 유사도를 가중치로 하여 외계인의 평점을
가중합(weight sum)하여 이를 예측 평점으로 사용

Collaborative Filtering

사용자 기반 협업 필터링



	준서	희나	수빈	석주
--	----	----	----	----

수천명의 사용자와 수십만개의 콘텐츠가 있기 때문에

현실적으로 매우 비효율적인 방법

$$\text{Rating Prediction} = \frac{(0.717 * 1) + (-0.271 * 4) + (0.427 * 3) + (0.561 * 2)}{(0.717 - 0.271 + 0.427 + 0.561)} = 1.42$$



잠재 요소 협업 필터링

이 점수를 바탕으로 콘텐츠 기반 필터링과 마찬가지로 분석자와
개발자의 주관에 따라 추천 진행



Collaborative Filtering

잠재 요소 협업 필터링

잠재 요소 협업 필터링

사용자가 평점을 부여하는데 잠재적으로 고려한 요소가 있다고 가정



사용자-잠재요소, 잠재요소-아이템 관계를 행렬로 표현할 수 있도록
사용자-아이템 관계를 표현해주는 **평점행렬을 분해**하는 방법

Collaborative Filtering

잠재 요소 협업 필터링

	어벤져스	포레스트 검프	매트릭스	엑시트	분노의 질주
준서	6	12	0	12	3
희나	12	10	6	8	10
수빈	14	7	9	4	13
석주	16	4	12	0	16



잠재 요소

- 영화 예시에서 평점을 내리는 기준
- '장르', '연기력', '스토리', '개연성' 등

Collaborative Filtering

잠재 요소 협업 필터링

	어벤져스	포레스트 검프	매트릭스	엑시트	분노의 질주
준서	6	12	0	12	3
희나	12	10	6	8	10
수빈	14	7	9	4	13
석주	16	4	12	0	16

사용자-아이템 관계의 평점행렬이 존재한다고 가정하여
이 중 '장르'가 매우 중요한 요소라고 가정

Collaborative Filtering

잠재 요소 협업 필터링

사용자-장르

	Comedy	Action
준서	3	0
희나	2	2
수빈	1	3
석주	0	4

장르-영화

	어벤져스	포레스트 검프	매트릭스	엑시트	분노의 질주
Comedy	2	4	0	4	1
Action	4	1	3	0	4



두가지 행렬로 분해해서 생각

→ 사용자×장르, 장르×영화 2개의 행렬이 생성

Collaborative Filtering

잠재 요소 협업 필터링과 특이값 분해

$$\begin{array}{c} n \\ \text{---} \\ \boxed{A} \\ \text{---} \\ m \end{array} = \begin{array}{c} m \\ \text{---} \\ \boxed{U} \\ \text{---} \\ m \end{array} \times \begin{array}{c} n \\ \text{---} \\ \boxed{\Sigma} \\ \text{---} \\ m \end{array} \times \begin{array}{c} n \\ \text{---} \\ \boxed{V^T} \\ \text{---} \\ n \end{array}$$

특이값 분해 (Singular Value Decomposition)

- ① 행렬을 특이값 분해했을 때 생기는 대각 행렬에서 특이값으로 차원 축소 진행
- ② 저장공간을 절약함과 동시에 주요한 잠재 요인만을 고려
→ 좀 더 정교하게 추천이 가능해짐



Collaborative Filtering

잠재 요소 협업 필터링의 장점과 한계

장점

연산이나, 평점을 예측하는 방식이 합리적

한계

cold-start문제나 평점행렬의 특성으로 인한
협업 필터링 기반 추천시스템의 근본적인 문제점 해결 못함

Collaborative Filtering

잠재 요소 협업 필터링의 한계

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	4			2	
User 2		5		3	
User 3			3	4	4
User 4	5	2	1	2	

모든 사용자가 자신이 소비한 모든 콘텐츠에 대해서 평점 내리지 않음

→ 대부분의 평점행렬은 **희소 행렬 (Sparse Matrix)** 형태

Collaborative Filtering

잠재 요소 협업 필터링의 한계

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	4			2	
User 2		5		3	
User 3			3	4	4
User 4	5	2	1	2	

모든 사용자가 자신이 소비한 모든 콘텐츠에 대해서 평점 내리지 않음

→ 대부분의 평점행렬은 **희소 행렬 (Sparse Matrix)** 형태

대부분의 원소가 비어 있는 행렬

Collaborative Filtering

잠재 요소 협업 필터링의 한계


	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	4			2	
User 2		5		3	
User 3			3	4	4
User 4	5	2	1	2	

원래 행렬을 그대로 사용

→ 어떤 오차를 줄이는 방향으로 결측치를 채울 수 없음

Collaborative Filtering

잠재 요소 협업 필터링의 한계



	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	4			2	
User 2					
User 3					
User 4	5	2	1	2	

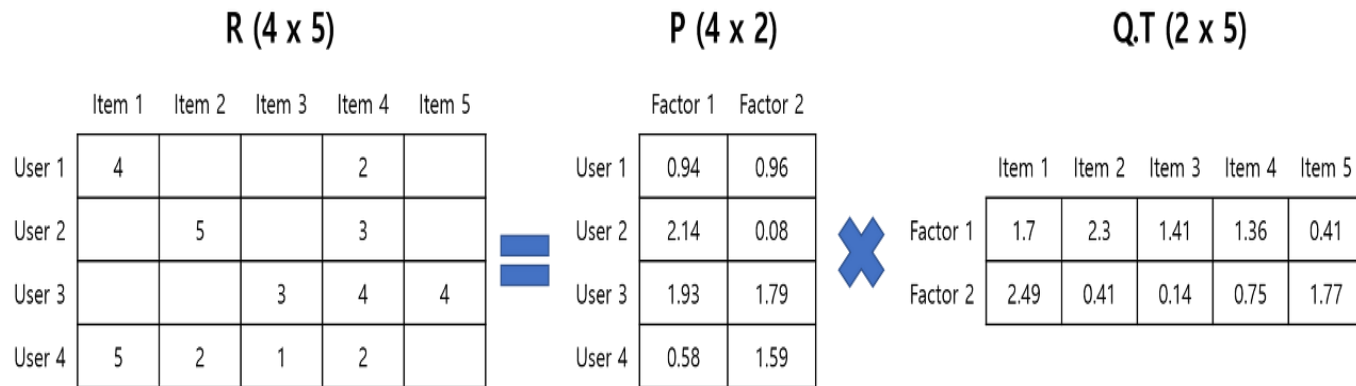
그럼 어떤 방법으로 진행될까요?

원래 행렬을 그대로 사용

→ 어떤 오차를 줄이는 방향으로 결측치를 채울 수 없음

Collaborative Filtering

유저 기반/아이템 기반 협업 필터링



행렬분해를 기반으로 진행

→ 결측치에 대한 예측과 이 예측값의 원래 행렬의 값과

비슷하도록 **최적화**하는 과정을 필요로 함

Collaborative Filtering

유저 기반/아이템 기반 협업 필터링

R (4 x 5)

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	4			2	
User 2		5		3	
User 3			3	4	4
User 4	5	2	1	2	

=

P (4 x 2)

	Factor 1	Factor 2
User 1	0.94	0.96
User 2	2.14	0.08
User 3	1.93	1.79
User 4	0.58	1.59

×

Q.T (2 x 5)

	Item 1	Item 2	Item 3	Item 4	Item 5
Factor 1	1.7	2.3	1.41	1.36	0.41
Factor 2	2.49	0.41	0.14	0.75	1.77

최적화 과정 방법

- ① Stochastic Gradient Descent
- ② Alternating Least Squares



THANK YOU

