

① Model-based Learning

- 선형/비선형모델 (e.g., linear regression, logistic regression)

- Neural network

(2)

- 의사 결정 나무

(1)

- Support vector machine

(3)



→ 데이터로부터 모델을 생성하여 분류 / 예측 진행

② Instance-based Learning

- K-nearest neighbor

- Locally weighted regression

(1)



(2)

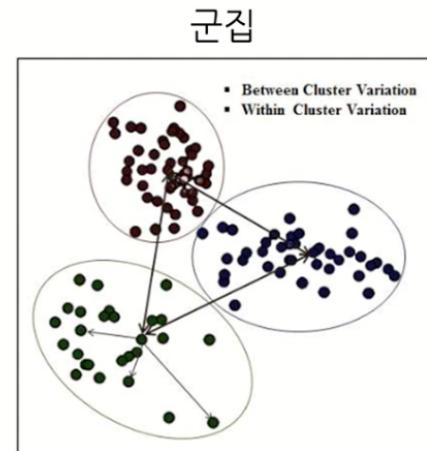
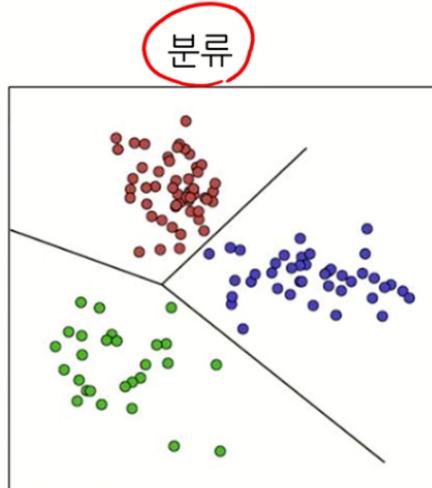
(3)

→ 별도의 모델 생성 없이 인접 데이터를 분류 / 예측에 사용

군집화 개념

❖ 분류 (Classification) vs. 군집화 (Clustering)

- 분류: 사전 정의된 범주가 있는 (labeled) 데이터로부터 예측 모델을 학습하는 문제 (지도학습; Supervised learning)
- 군집화: 사전 정의된 범주가 없는 (unlabeled) 데이터에서 최적의 그룹을 찾아나가는 문제 (비지도학습; Unsupervised learning)



분류: Y가 있다. X들을 갖고, y의 범주를 잘 나눌 수 있도록, decision boundary

분류 모델은 새로운 점이 빨, 초, 파인지 분류할 수 있게 한다.

군집: y가 없다. 예측하려 하는 대상이 없다. 비슷한 관측치끼리 서로 묶는다. 예측할 수 있는

[참고] adp 필기책

1. 계층적 군집 (hierarchical) : n개의 군집으로 시작해서, 군집의 개수를 줄여가는 방법이다

① 합병형(군집간 거리 척도/연결법, linkage method)

- 단일 최단 연결법
- 평균연결법
- Ward연결법
- 완전 최장 연결법
- 중심연결법

② 분리형(top-down)

- 다이아나 방법

2. 분할적 군집 (partitional)

프로토타입

- K-means
- K-centroid
- K- median
- K-medoid
- Fuzzy

<https://syj9700.tistory.com/41>

분포기반

- Mixture distribution
- EM(Expectation – maximization) 알고리즘
<https://people.duke.edu/~ccc14/sta-663/EMAlgorithm.html>
<https://syj9700.tistory.com/39>
- SOM(self organizing map)

밀도기반

어느 점을 기준으로 주어진 반경 내에 최소 개수만큼의 데이터들은 가질 수 있도록 함으로써 특정 밀도 함수 혹은 밀도에 의해 군집을 형성해 나가는 방법

- DBSCAN

<https://syj9700.tistory.com/40>

- OPTICS
- DENCLUE

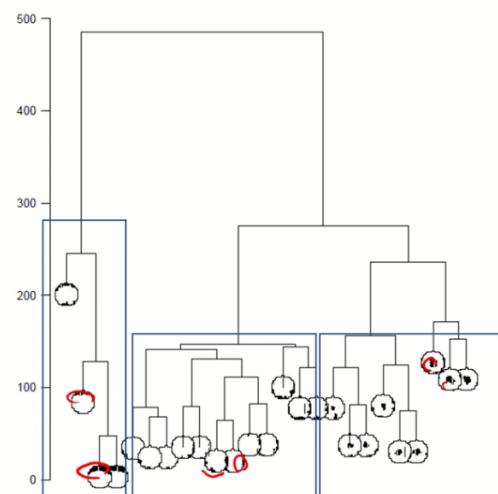
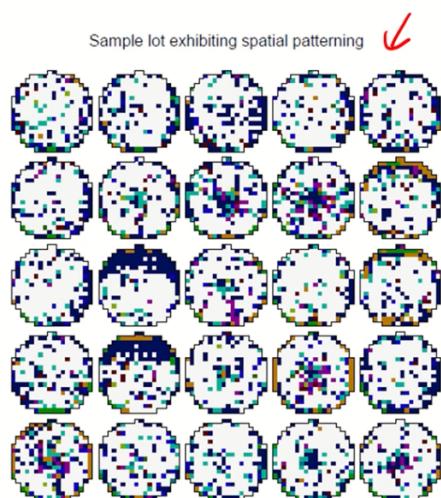
Grid based (STING, WaveCluster, Cliqe)

- 군집의 활용 및 적용 사례
- 효율적인 광고를 위해 비슷한 특징을 갖고 있는 고객들끼리 묶는다
 비슷한 성격을 갖고 있는 특히 문서끼리 묶는다
 서울시 오존농도 25개의 관측치, 비슷한 패턴을 갖는 구끼리 묶어서 분석한다

군집화 적용사례

❖ 군집화 적용 사례

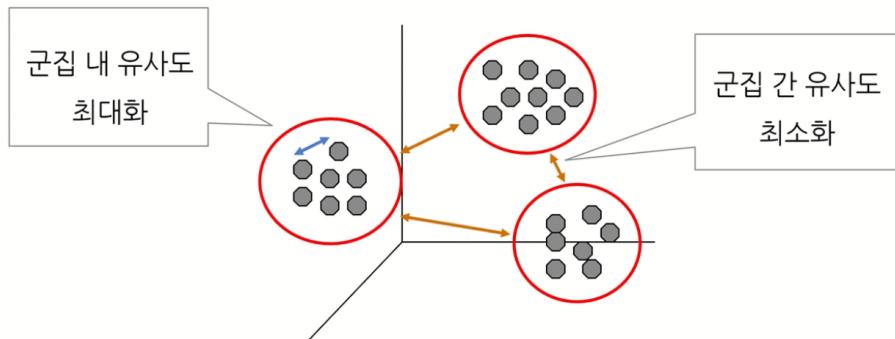
- 웨이퍼 Fail bit map 군집화



군집화 개념

❖ 군집화 기준

- 동일한 군집에 소속된 관측치들은 서로 유사할수록 좋음
- 상이한 군집에 소속된 관측치들은 서로 다를수록 좋음



군집화 수행 시 주요 고려사항

- ❖ 어떤 거리 척도를 사용하여 유사도를 측정할 것인가?
- ❖ 어떤 군집화 알고리즘을 사용할 것인가?
- ❖ 어떻게 최적의 군집 수를 결정할 것인가?
- ❖ 어떻게 군집화 결과를 측정/평가할 것인가?

비슷한 거 끼리는 같은 군집, 다른거 끼리는 다르게 묶어야 하기 때문에 “유사도”를 측정해야한다.
유사도라는 것은 거리와 닮아있다. 거리가 멀수록 유사도는 떨어지는 반대되는 개념이기 때문이다.

: $1-d$, $1/d$

설정한 유사도계산법으로, 어떻게 군집화를 할 것인지

(예를 들면, 숫자라는 유사도를 설정했는데, 어디서 어디까지를 묶음으로 할지 알고리즘은 군집화 알고리즘)

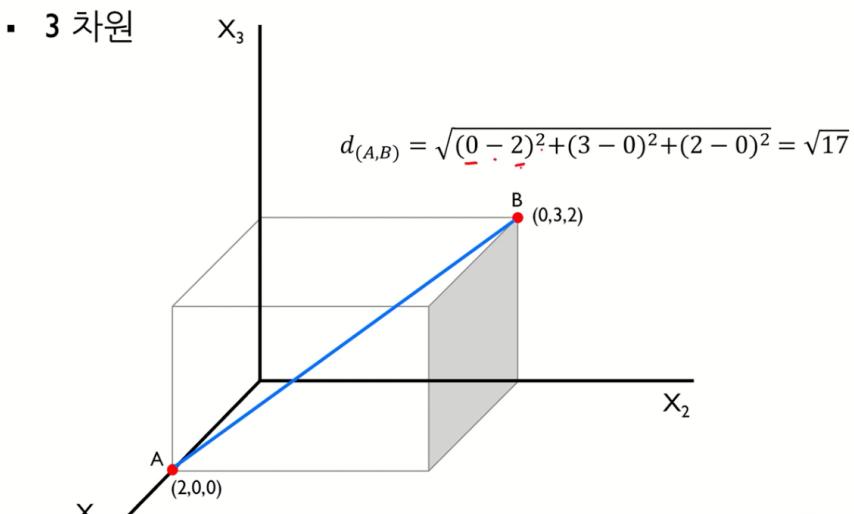
대표적인 거리측도

- Euclidean Distance
- Manhattan Distance
- Mahalanobis Distance
- Correlation Distance
 - Pearson Correlation
 - Spearman Rank Correlation

거리측도 (1-유사도)

- 다양한 거리측도 (Distance measure) 존재
(e.g., Euclidean distance, Correlation distance, ...)
- 데이터 내 변수들이 각기 다른 데이터 범위, 분산 등을 가질 수 있으므로, 데이터 정규화 (혹은 표준화)를 통해 이를 맞추는 것이 중요함
 - 거리를 계산할 때, 단위가 큰 특정 변수(들)가 거리를 결정하는 것 방지
 - 예) 키(1.5m~1.8m), 몸무게(90lb~300lb), 연봉(20,000,000원~100,000,000원)

❖ 유클리디안 거리



Knn 거리에 대해서 자세히 설명함

❖ 유클리디안 거리

- p 차원

$$A = (a_1, a_2, \dots, a_p)$$
$$B = (b_1, b_2, \dots, b_p)$$

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

❖ 맨하탄 거리 (Manhattan Distance)



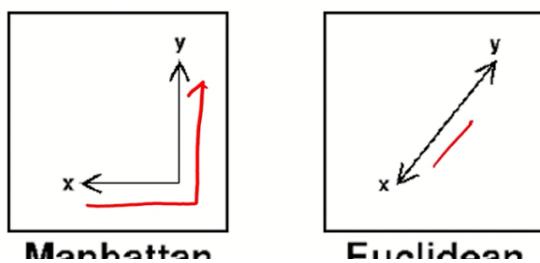
미국에서는 가로질러 갈수가 없다. 블록으로 되어있음. 맨하탄에서 계산되는 거리와 비슷해서, 맨하탄 거리라고 부른다. 맨하탄 거리는 격자가 있다고 가정한다. 수식은 간단한다. 차이 절대값의 합이다.

군집화: 유사도 척도

❖ 맨하탄 거리 (Manhattan Distance)

- X에서 Y로 이동 시 각 좌표축 방향으로만 이동할 경우에 계산되는 거리

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



❖ 마할라노비스 거리

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

where Σ^{-1} = Inverse of covariance matrix

- 변수 내 분산, 변수 간 공분산을 모두 반영하여 X, Y 간 거리를 계산하는 방식
- 데이터의 covariance matrix가 identity matrix인 경우는 Euclidean distance와 동일함

가운데 공분산만 없으면, 유클라디안 거리와 동일하다.

분산을 역행렬을 취했다는 것은, 분산이 크면 작은 값을 곱하고, 분산이 작으면 큰 값을 곱하게 된다. 단위행렬은 1과 동일하다. 그래서 단위행렬이 공분산 되는 경우는 마할라노비스=유클라디안 거리

변수가 여러개 있는 경우에, 두 점사이의 공분산을 고려한다

Mahalanobis Distance

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

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

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

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

It can be considered as the squared Mahalanobis distance between a certain point X , and the fixed point Y .

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

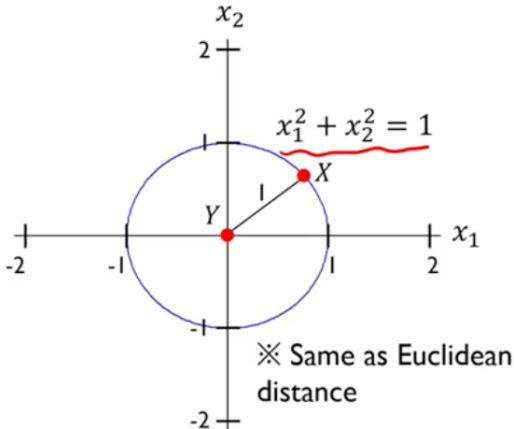
$$\rightarrow x_1^2 s_{11}^{-1} + 2x_1 x_2 s_{12}^{-1} + x_2^2 s_{22}^{-1} = c^2$$

which is a general equation of the ellipse. *证毕.*

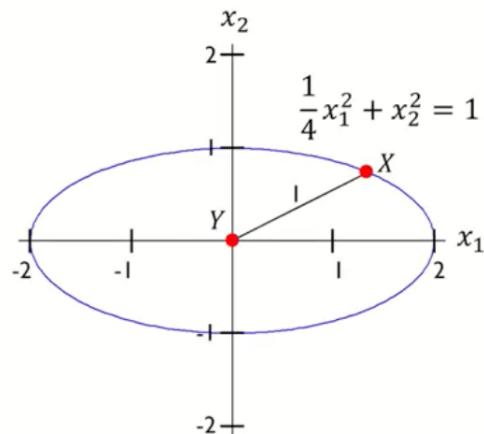
Mahalanobis Distance

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

(identity matrix)

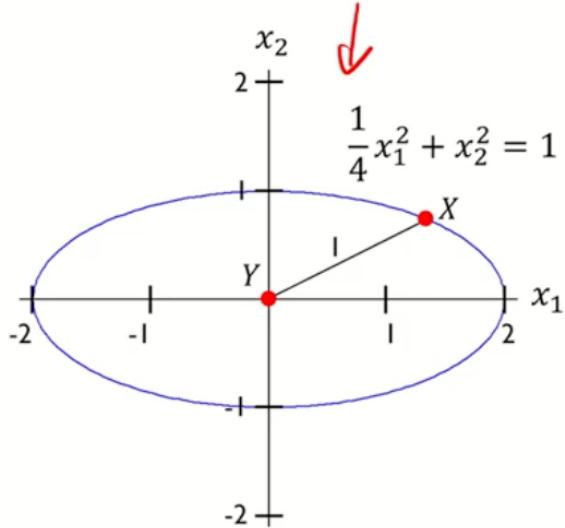


$$\Sigma = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}, \Sigma^{-1} = \begin{pmatrix} 1/4 & 0 \\ 0 & 1 \end{pmatrix}$$

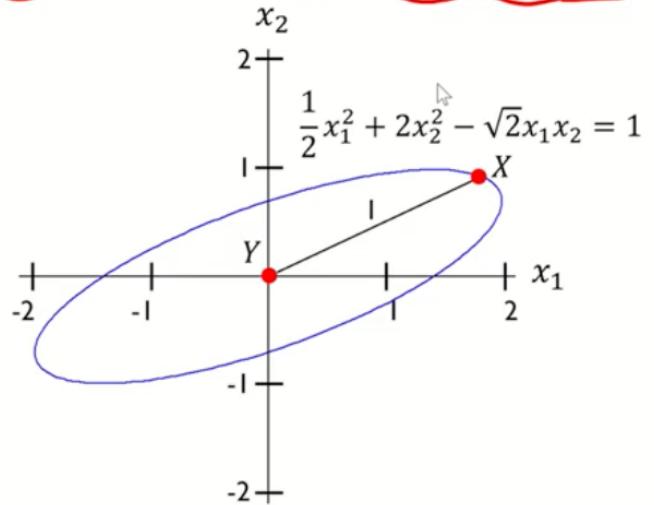


Mahalanobis Distance

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



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



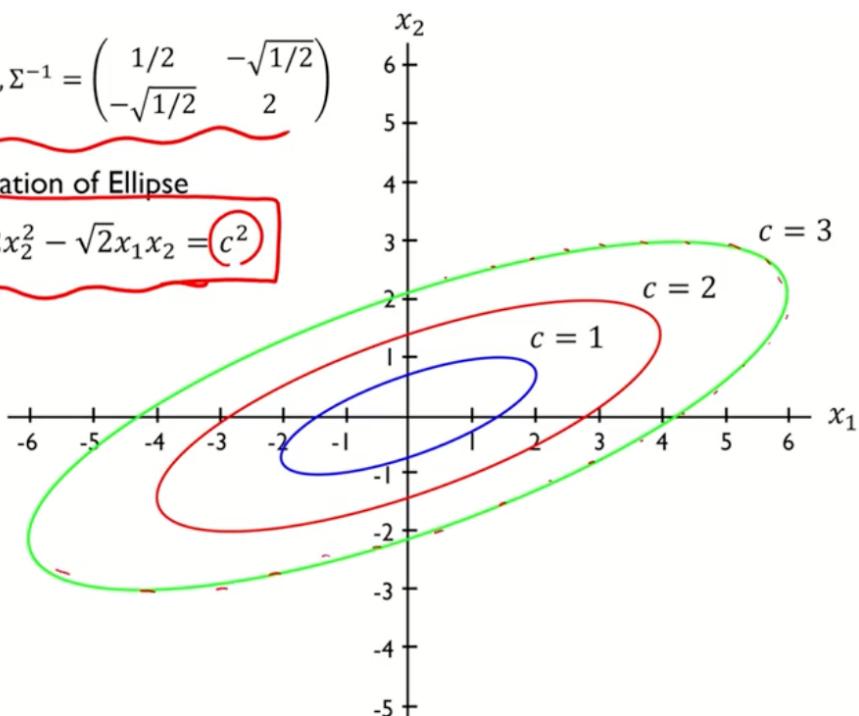
공분산이 고려되면, 휘어진 타원이 된다.

만약 마지막 예제를 더 확인해보면

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

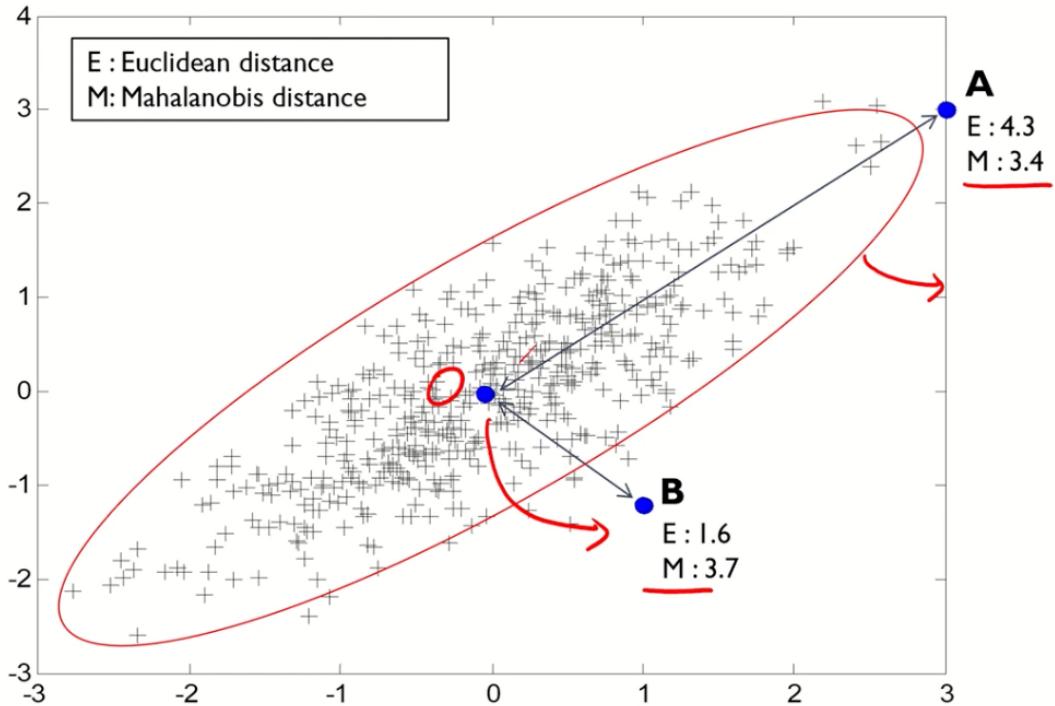
Equation of Ellipse

$$\frac{1}{2}x_1^2 + 2x_2^2 - \sqrt{2}x_1x_2 = c^2$$



타원 위에 있는 점들은 마할라노비스에서 동일한 거리를 갖게 된다.

❖ 마할라노비스 거리



B는 상관관계에 반하는 방향으로 가고 있기 때문에, 마할라노비스 관점에서보면 거리가 더 멀어진다.

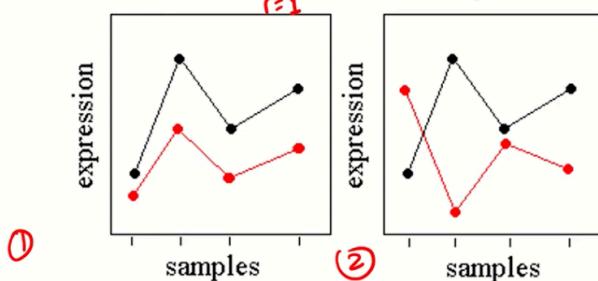
❖ 상관계수 거리

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

$$d_{\text{corr}}(X, Y) = 1 - r$$

where $r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2} \sqrt{\sum (y_i - \bar{y})^2}}$

$$r=1 \quad 0 \leq d_{\text{corr}} \leq 2 \quad r=-1$$



- 데이터 간 Pearson correlation을 거리 척도로 직접 사용하는 방식으로, 데이터 패턴의 유사도 / 비유사도를 반영할 수 있음

1. 전체적인 패턴이 동일하다.
2. 전체적인 패턴이 반대이다.

Spearman Rank Correlation Distance

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

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

- ρ 를 Spearman correlation이라 하며, 이는 데이터의 rank를 이용하여 correlation distance를 계산하는 방식임

- ρ 의 범위는 -1 부터 1로, Pearson correlation과 동일

The diagram illustrates the process of ranking data. On the left, a table shows raw data for four cities (Seoul, New York, Sydney) across four seasons (Spring, Summer, Autumn, Winter). A red bracket above the table indicates the '계절 평균 낮 최고 기온' (Seasonal average low highest temperature). An arrow points from this table to the right, where another table shows the '지역 별 계절 기온 순위' (Seasonal temperature ranking by region). In this second table, Seoul and New York have identical rankings (3 for Spring, 1 for Summer, 2 for Autumn, 4 for Winter), while Sydney has a different ranking (2 for Spring, 4 for Summer, 3 for Autumn, 1 for Winter).

지역	계절 평균 낮 최고 기온			
	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	지역 별 계절 기온 순위			
	봄	여름	가을	겨울
서울	3	1	2	4
뉴욕	3	1	2	4
시드니	2	4	3	1

$$\rho = 1 - \frac{6\{(3-3)^2 + (1-1)^2 + (2-2)^2 + (4-4)^2\}}{4(4^2 - 1)} = 1 \quad d_{(서울, 뉴욕)} = 1 - 1 = 0$$

$$\rho = 1 - \frac{6\{(3-2)^2 + (1-4)^2 + (2-3)^2 + (4-1)^2\}}{4(4^2 - 1)} = -1 \quad d_{(서울, 시드니)} = 1 - (-1) = 2$$

순위데이터 관점에서, 서울과 뉴욕은 같은 군집이 될것이다. ($D=0$)

❖ 어떤 군집화 알고리즘을 사용할 것인가?

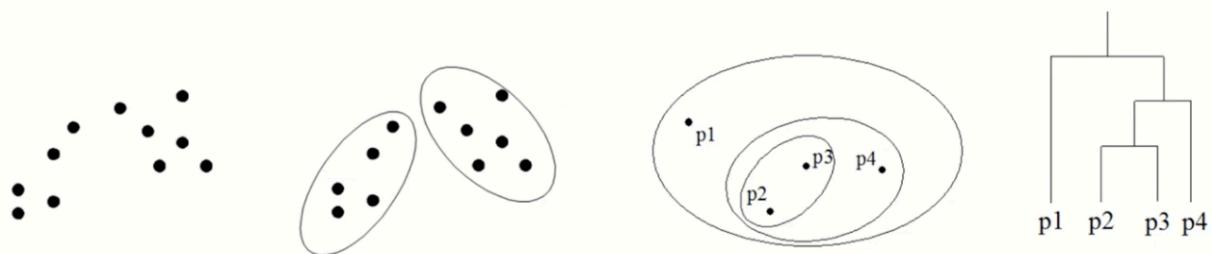
❖ 군집화 알고리즘의 종류

- 계층적 군집화

- 개체들을 가까운 집단부터 차근차근 떠나가는 방식
- 군집화 결과 뿐만 아니라 유사한 개체들이 결합되는 **dendrogram** 생성

- 분리형 군집화

- 전체 데이터의 영역을 특정 기준에 의해 동시에 구분
- 각 개체들은 사전에 정의된 개수의 군집 중 하나에 속하게 됨



계층적 군집화는 하나하나 군집을 해나간다.

분리형 군집화는 하나하나가 아니라, 데이터를 동시에 구분한다

❖ 어떤 군집화 알고리즘을 사용할 것인가?

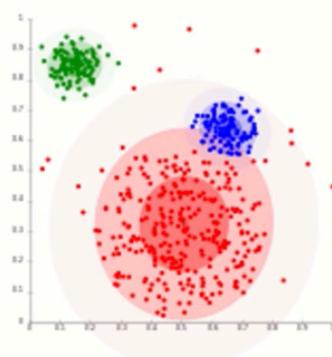
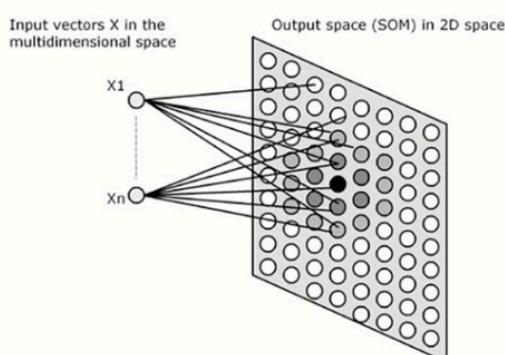
❖ 군집화 알고리즘의 종류

③. 자기조직화 지도

- 2차원의 격자에 각 개체들이 대응하도록 인공신경망과 유사한 학습을 통해 군집 도출

④. 분포 기반 군집화

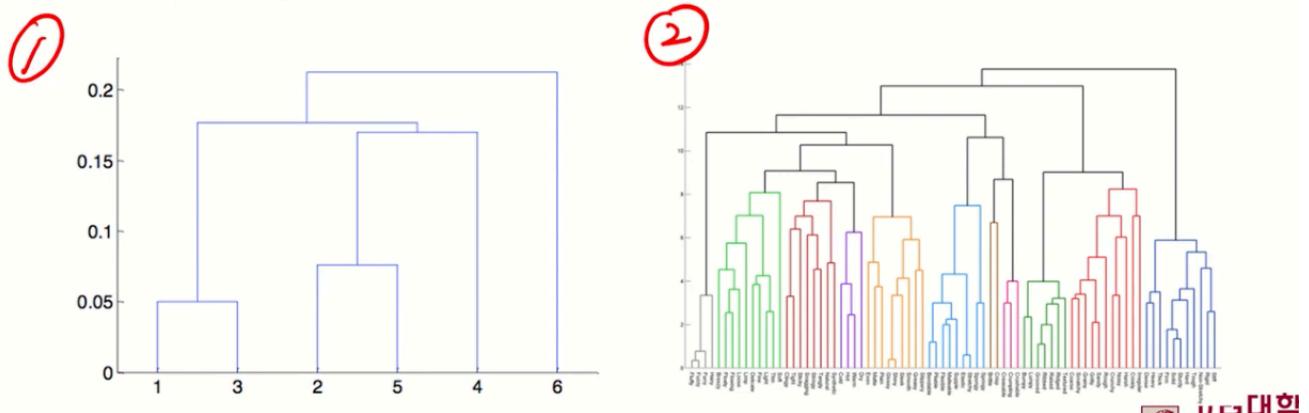
- 데이터의 분포를 기반으로 높은 밀도를 갖는 세부 영역들로 전체 영역을 구분



- [딥러닝 클러스터링] https://www.youtube.com/watch?v=eLZeodQ_v2M

❖ 계층적 군집화

- 계층적 트리모형을 이용하여 개별 개체들을 순차적/계층적으로 유사한 개체/군집과 통합
- 덴드로그램(Dendrogram)을 통해 시각화 가능
 - ✓ 덴드로그램: 개체들이 결합되는 순서를 나타내는 트리형태의 구조
- 사전에 군집의 수를 정하지 않아도 수행 가능
 - ✓ 덴드로그램 생성 후 적절한 수준에서 자르면 그에 해당하는 군집화 결과 생성



1-3 관측치가 가장 유사하고, 2-5가 가장 유사하고.. 6번째 관측치는 따로 떨어져 있다. 나머지 5개와는 서로 다른 관측치인 6일것으로 생각할 수 있다.

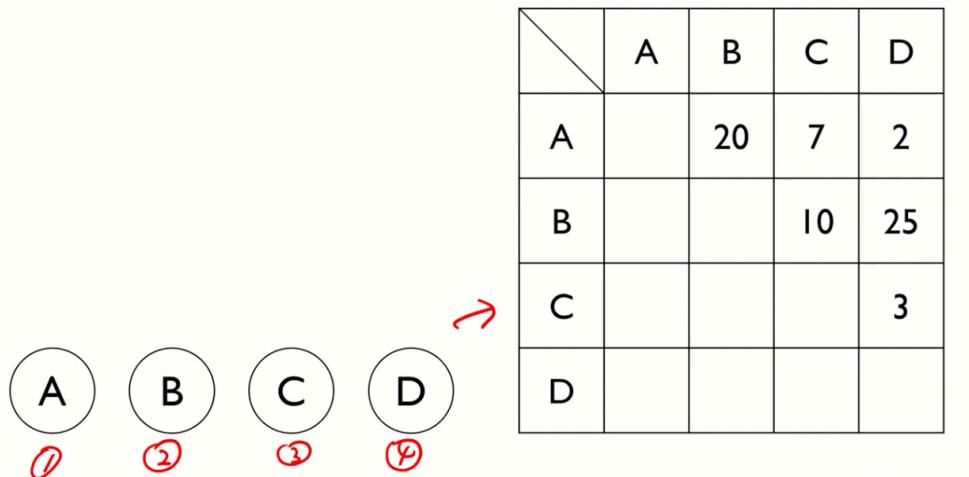
1-3이 2-5보다 더 비슷한 군집이다. Y축의 값은 normalized된 값이고, 상대성을 갖는다.

덴드로그램, clustering tree라고도 부르지만, 의사결정나무와는 완전히 다르다.

의사결정 나무는 y의 균일한 방향으로 묶어가지만, 여기서는 y가 없이, 관측치의 유사도로 묶어가는 것이기 때문이다.

❖ 계층적 군집화 수행 예시

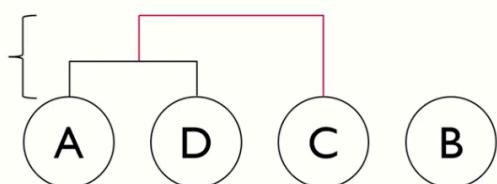
- 모든 개체들 사이의 거리에 대한 유사도 행렬 계산



가장 거리가 가까운 것 – a,d

❖ 계층적 군집화 수행 예시

- 모든 개체들 사이의 거리에 대한 유사도 행렬 계산
- 거리가 인접한 관측치끼리 군집 형성
- 유사도 행렬 업데이트
- 위의 과정 반복



	AD	B	C	
AD		20	3	
B			10	
C				

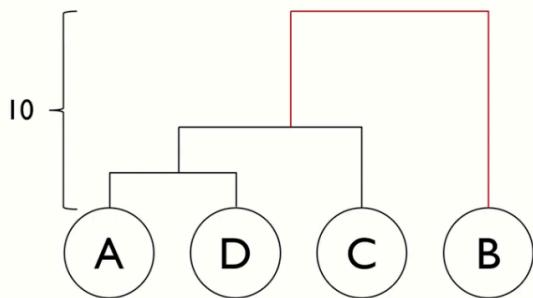
하나의 개체가 아니라 두 군집사이의 거리는 어떻게 계산할까?

만약 1번 군집에 (a,b,c) 있고, 2번 군집에 (d,e)가 있을때

가장 가까운 거리로 둘 수도 있고, 가장 큰 거리를 둘 수도 있고, 6개 case의 거리를 계산해 평균을 계산 할 수도 있다. 또는 각각 군집의 대표를 선정해서 거리를 둘 수도 있다.

❖ 계층적 군집화 수행 예시

- 모든 개체들 사이의 거리에 대한 유사도 행렬 계산
- 거리가 인접한 관측치끼리 군집 형성
- 유사도 행렬 업데이트
- 위의 과정 반복

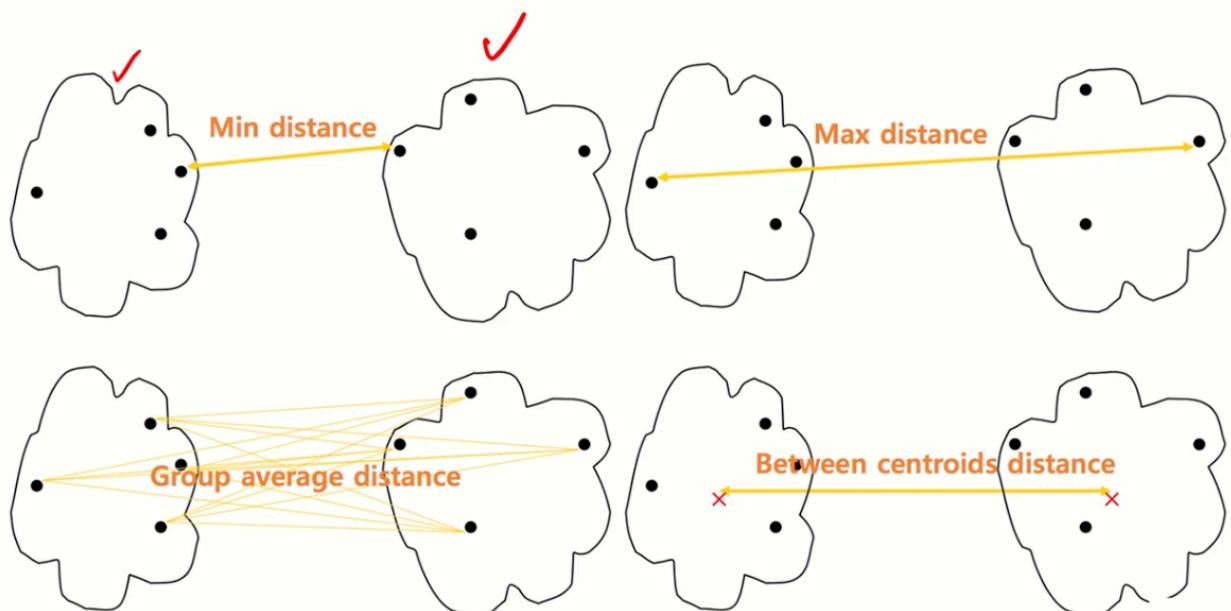


	AD	C	B		
	AD	C		10	
	B				

설정하는 높이에 따라서 군집의 개수가 달라질 수 있다. (y축으로 그어볼때)

▪ 핵심 수행 절차: 두 군집 사이의 유사성/거리 측정

- ✓ Min (단일연결), max (완전연결), group average (평균연결), between centroid, Ward's, ...



계층적 군집화

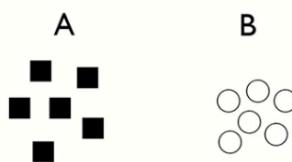


- ❖ Ward's method: Distance between two clusters, A and B, is how much the sum of squares will increase when they are merged.

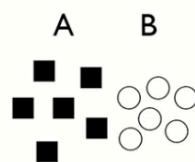
$$\text{Ward Distance} = \sum_{i \in A \cup B} \|x_i - m_{A \cup B}\|^2 - \left\{ \sum_{i \in A} \|x_i - m_A\|^2 + \sum_{i \in B} \|x_i - m_B\|^2 \right\}$$

m_A is the center of cluster A.

Ward's distance can be considered as the merging cost of combining the clusters A and B



$$\text{Ward's distance} = 10 - (3+2) = 5$$



$$\text{Ward's distance} = 7 - (3+2) = 2$$

계층적 군집화에서 비교적 정확하게 나오는 방법이다.

1번은 Ab 는 서로 떨어져 있고, 2번에는 좀 더 붙어있다. 결과가 1>2이어야 할것이다.

식의 의미는 아래와 같다.

A,B가 두개의 군집이지만, 하나의 군집이라고 보고 중심과의 거리에서
A의 군집으로부터 중심과의 거리, B군집으로부터 중심과의 거리를 뺀다

- ❖ 어떤 군집화 알고리즘을 사용할 것인가?

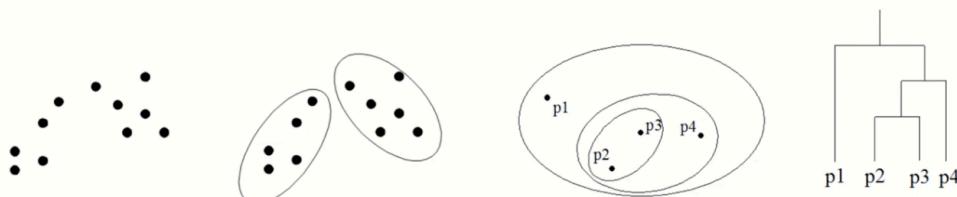
- ❖ 군집화 알고리즘의 종류

- 계층적 군집화

- 개체들을 가까운 집단부터 차근차근
묶어나가는 방식
- 군집화 결과 뿐만 아니라 유사한 개체
들이 결합되는 dendrogram 생성

- 분리형 군집화

- 전체 데이터의 영역을 특정 기준에
의해 동시에 구분
- 각 개체들은 사전에 정의된 개수의
군집 중 하나에 속하게 됨



K-평균 군집화 (K-Means Clustering)

❖ K-평균 군집화

- 대표적인 분리형 군집화 알고리즘
 - ✓ 각 군집은 하나의 중심(centroid)을 가짐
 - ✓ 각 개체는 가장 가까운 중심에 할당되며, 같은 중심에 할당된 개체들이 모여 하나의 군집을 형성
 - ✓ 사전에 군집의 수 K가 정해져야 알고리즘을 실행할 수 있음

$$X = C_1 \cup C_2 \dots \cup C_k, C_i \cap C_j = \emptyset, i \neq j$$

$$\operatorname{argmin}_c \sum_{i=1}^K \sum_{x_j \in C_i} \|x_j - c_i\|^2$$

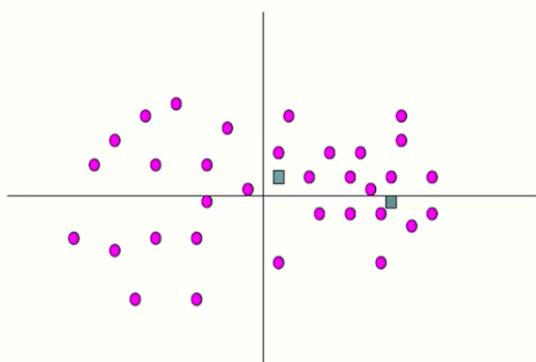
X = 우리가 갖고 있는 데이터이다. Y가 없다!

K개의 군집에 대한 합집합이 x data, overlapping 된 곳이 없다.

군집의 개수를 임의로! (사실 실제상황에서는, 어렵겠지 알 수 있을 때가 꽤 많다. 불량품/양품, 생존/죽음)

❖ K-평균 군집화 수행 예시 (K=2)

1. 2개의 중심을 임의로 생성

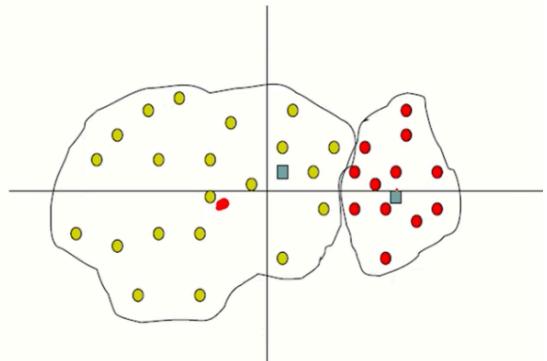


각각의 관측치로부터, 다른 데이터와의 거리를 계산한다.

하나의 핑크색 점이 1, 2(회색점)과의 중심을 갖고 계산해서, 군집을 그려볼 수 있다.

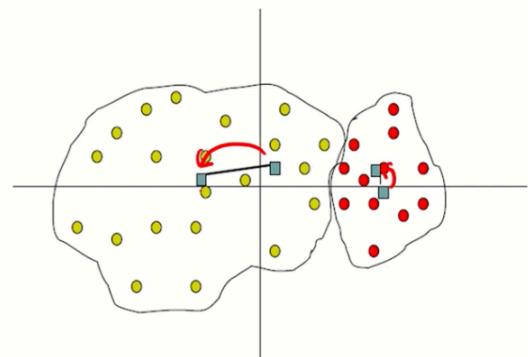
❖ K-평균 군집화 수행 예시 (K=2)

1. 2개의 중심을 임의로 생성
2. 생성된 중심을 기준으로 모든 관측치에 군집 할당



❖ K-평균 군집화 수행 예시 (K=2)

1. 2개의 중심을 임의로 생성
2. 생성된 중심을 기준으로 모든 관측치에 군집 할당
3. 각 군집의 중심을 다시 계산

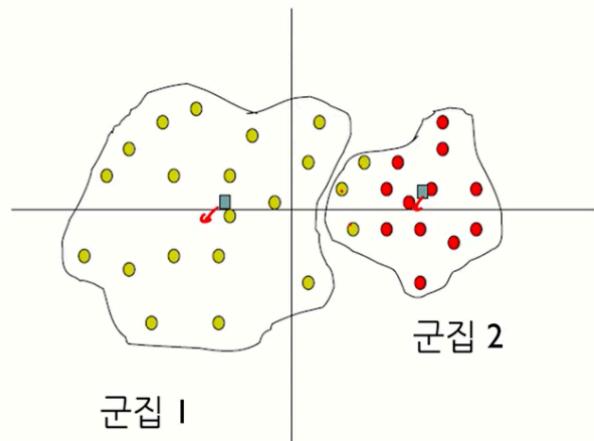


중심(회색점)이 이동했을 때, 더 가까운 중심점이 이동되는 점들이 있다. (빨간 군집과 가까운 노란색점)

❖ K-평균 군집화 수행 예시 ($K=2$)

1. 2개의 중심을 임의로 생성
2. 생성된 중심을 기준으로 모든 관측치에 군집 할당
3. 각 군집의 중심을 다시 계산
4. 중심이 변하지 않을 때까지

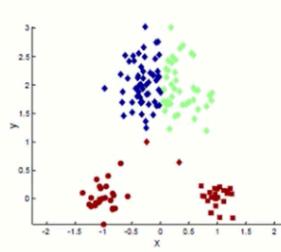
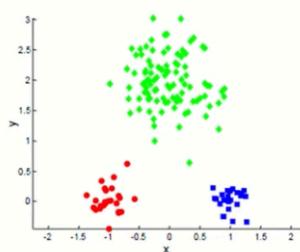
위의 과정을 반복



K-평균 군집화

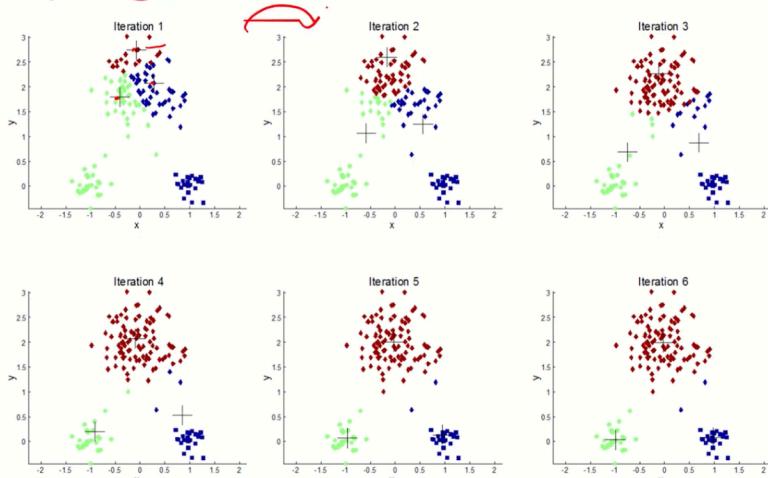
❖ K-평균 군집화 수행 절차

1. 초기 중심을 K개 임의로 생성
 2. 개별 관측치로부터 각 중심까지의 거리를 계산 후, 가장 가까운 중심이 이루는 군집에 관측치 할당
 3. 각 군집의 중심을 다시 계산
 4. 중심이 변하지 않을 때까지 2, 3의 과정을 반복
- ✓ 초기 중심은 종종 무작위로 설정됨: 군집화 결과가 초기 중심 설정에 따라 다르게 나타나는 경우가 발생할 수도 있음

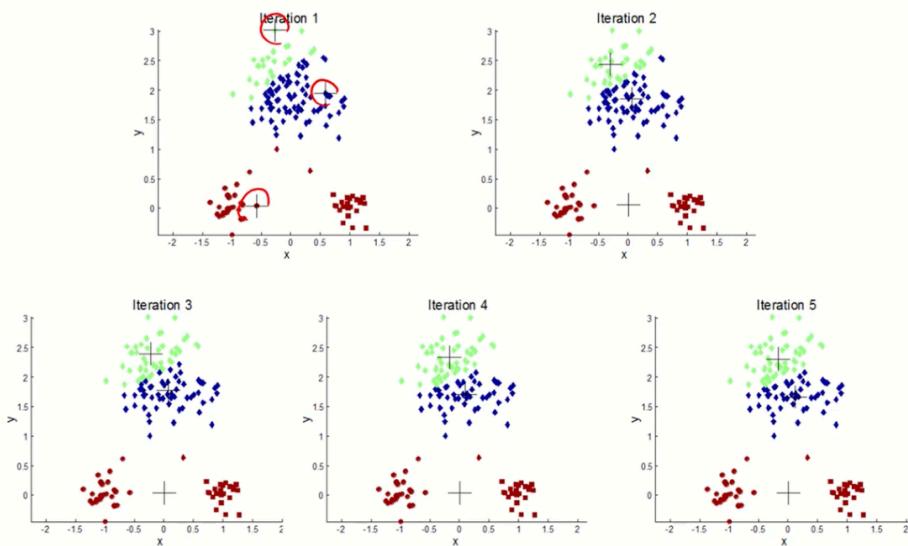


- ◆ 초기 중심 설정이 최종 결과에 어떤 영향을 미치는가?

바람직한 결과



바람직하지 않은 결과



초기 중심점을 반복적으로 하고, 여러번으로 나온 군집의 결과로 실행한다.

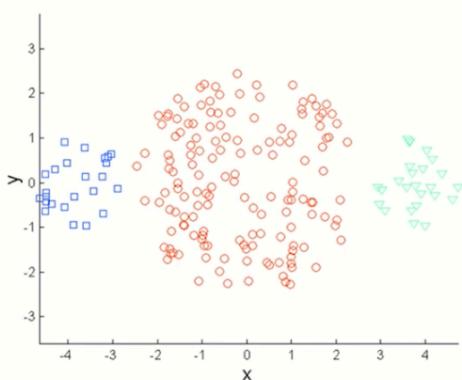
정확한 군집의 개수는 알지 못하지만, 먼저 계층 군집을 수행한다음에 k개수를 설정한다.

데이터의 분포(정규분포등 알고있을 경우) 초기 중심으로 설정에 활용한다.

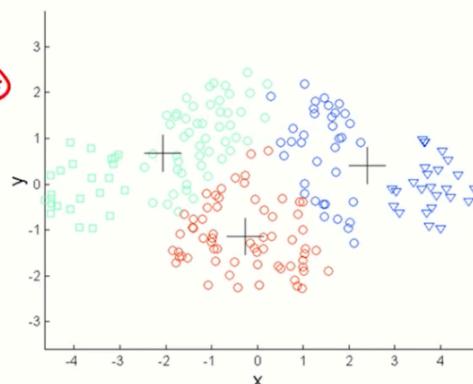
❖ 무작위 초기 중심 설정의 위험을 피하고자 다양한 연구 존재

- 반복적으로 수행하여 가장 여러 번 나타나는 군집을 사용
- 전체 데이터 중 일부만 샘플링하여 계층적 군집화를 수행한 뒤 초기 군집 중심 설정
- 데이터 분포의 정보를 사용하여 초기 중심 설정
- 하지만 많은 경우 초기 중심 설정이 최종 결과에 큰 영향을 미치지는 않음
- 문제점1: 서로 다른 크기의 군집을 잘 찾아내지 못함

정답



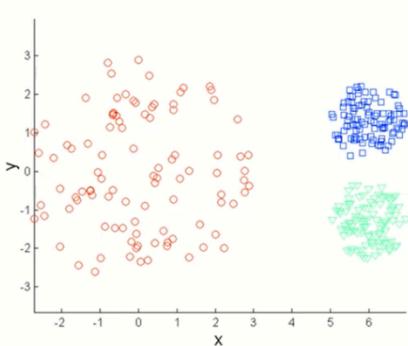
K-평균 군집화 결과



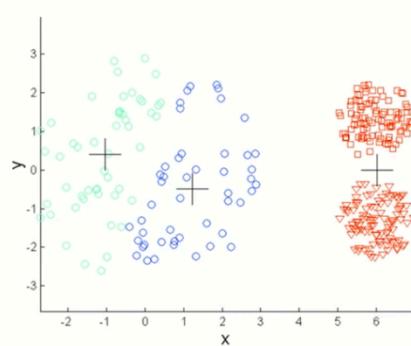
❖ K-평균 군집화의 문제점

- 문제점2: 서로 다른 밀도의 군집을 잘 찾아내지 못함

정답



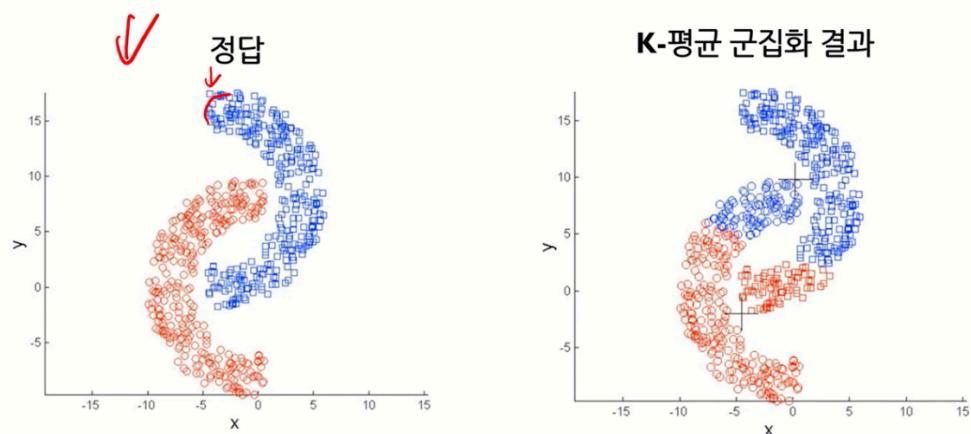
K-평균 군집화 결과



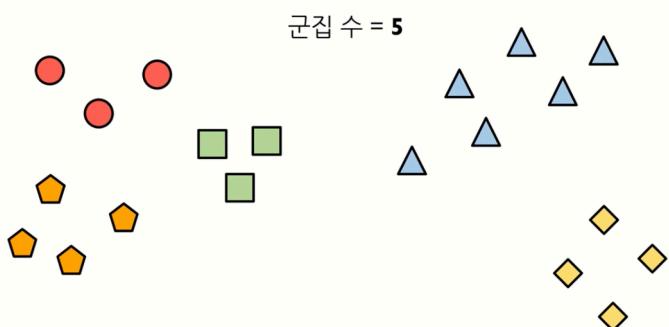
군집이 지역적 패턴이 존재하는 경우. 보는 위치에 따라 데이터가 볼록, 오목, 평평... 다른 것이다.

지역적 패턴이 존재한다는 것은, 어디서 보는지에 따라서 패턴이 다른 것이다.
 반대의 예로 원은 지역적 패턴이 전혀 존재하지 않는다.
 우리가 소위 생각하는 거리접근법(유클리디안 거리)이 갖고 있는 문제점이기도 하다.

- Geodesic distance (local pattern 있을 때 사용할 수 있는 거리)
- ❖ K-평균 군집화의 문제점
 - 문제점3: 지역적 패턴이 존재하는 군집을 판별하기 어려움

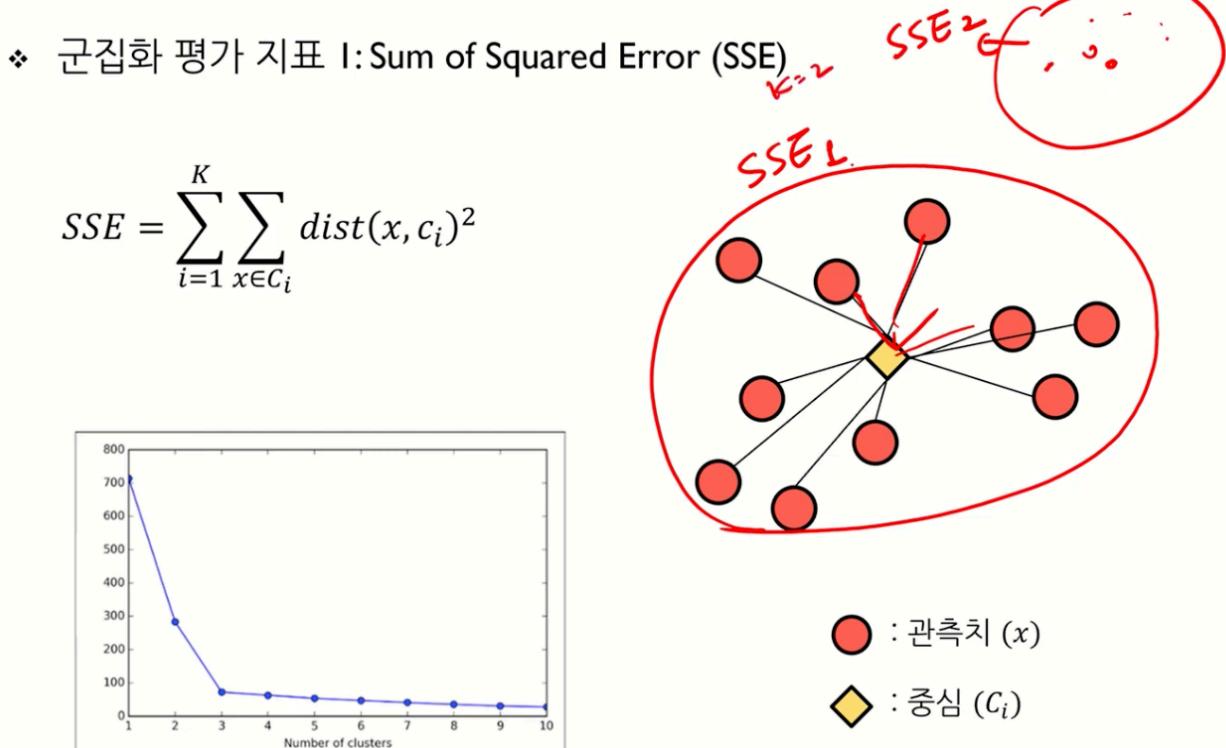
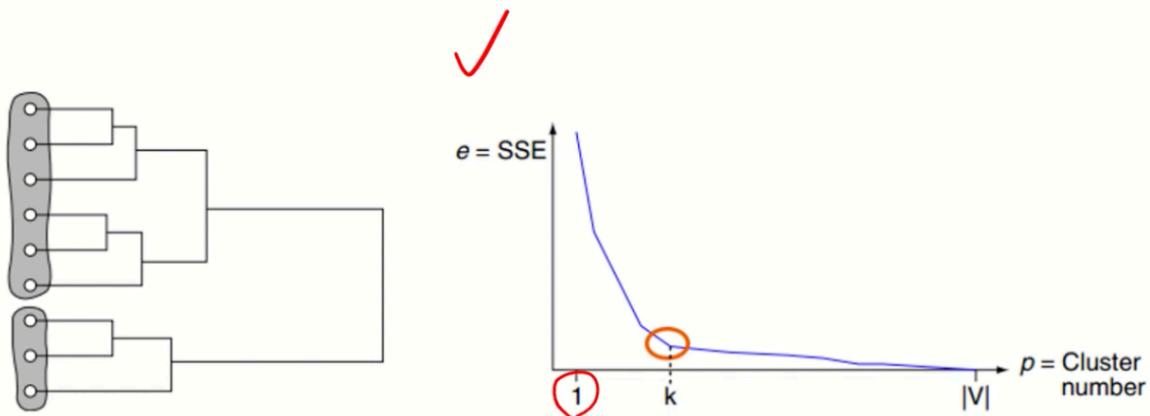


- ❖ 어떻게 최적의 군집 수를 결정할 것인가?
 - 예시) 20개의 관측치가 존재할 때, 최적의 군집 수는?



객관적으로 군집 결과에 대한 평가를 수치화 할 수 없을까?

- ❖ 어떻게 최적의 군집 수를 결정할 것인가?
 - 다양한 군집 수에 대해 성능 평가 지표를 도시하여 최적의 군집 수 선택
 - Elbow point에서 최적 군집 수가 결정되는 경우가 일반적



직관적, 간단하고 좋지만 SSE는 군집내 거리 최소화를 잘 반영하지만, 군집간의 거리는 전혀 반영되어있지 않는다는 단점이 있다. 이러한 단점을 보완한 것이 실루엣 거리다.

$b(i), a(i)$

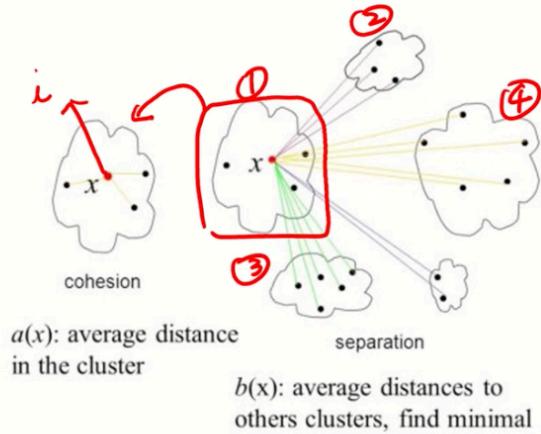
❖ 군집화 평가 지표 2: Silhouette 통계량

- ❖ $a(i)$: 관측치 i 로부터 같은 군집 내에 있는 모든 다른 개체들 사이의 평균 거리
- ❖ $b(i)$: 관측치 i 로부터 다른 군집 내에 있는 개체들 사이의 평균 거리 중 최솟값
- ❖ 일반적으로 \bar{S} 의 값 0.5보다 크면 군집 결과가 타당하다고 볼 수 있음
- ❖ -1에 가까우면 군집이 전혀 되지 않음

$|C=4|$

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad -1 \leq s(i) \leq 1$$

$$\bar{S} = \frac{1}{n} \sum_{i=1}^n S(i)$$

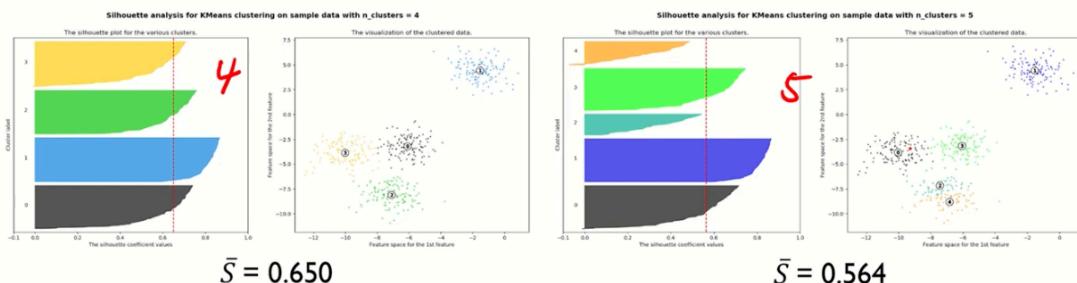
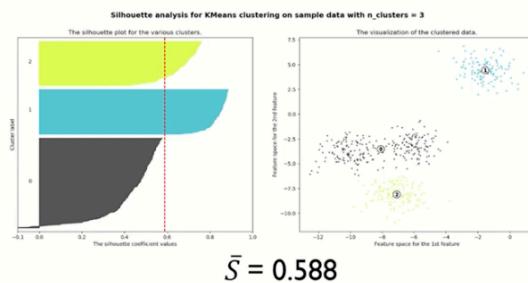
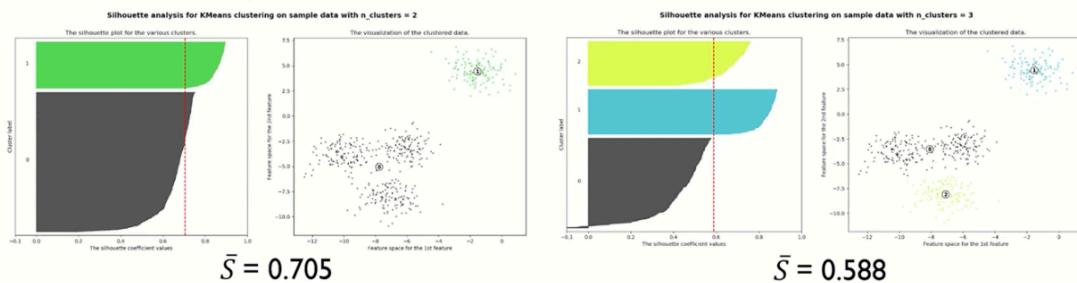


<https://www.mtechprojects.org/silhouette-coefficient-projects.html>

KOREA 고려대학교

군집화: 결과 측정 및 평가

$$K=2 \quad \left[s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \right] \quad 3$$



KOREA 고려대
KOREA UNIV

군집이 2개가 좋은지 3개가 좋은지..

Scale된 값을 만들기 위해서, 1에 가까울 수록 좋고, -1에 가까울 수록 좋지 않다.

https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

원쪽 그래프에서, x축이 실루엣 값이다.

색깔은 군집을 얘기하고, y축의 값이 개체 값을 의미한다

실루엣으로 평가할 때 많은 경우에 $k=2$ 에서 높게 나오기 때문에, second best k 값을 본다.

	Kmeans	Knn
분류/군집	Clustering	Classifier
K의미	임의의 군집들의 중심을 잡기 위한 개수	K개를 참조하여 분류하기