



Introduction to Artificial Intelligence [AICS223]

Unsupervised Learning: Dimension Reduction (W10)

Prof. Mee Lan Han (aeternus1203@gmail.com)

고려대학교

인공지능사이버보안학과



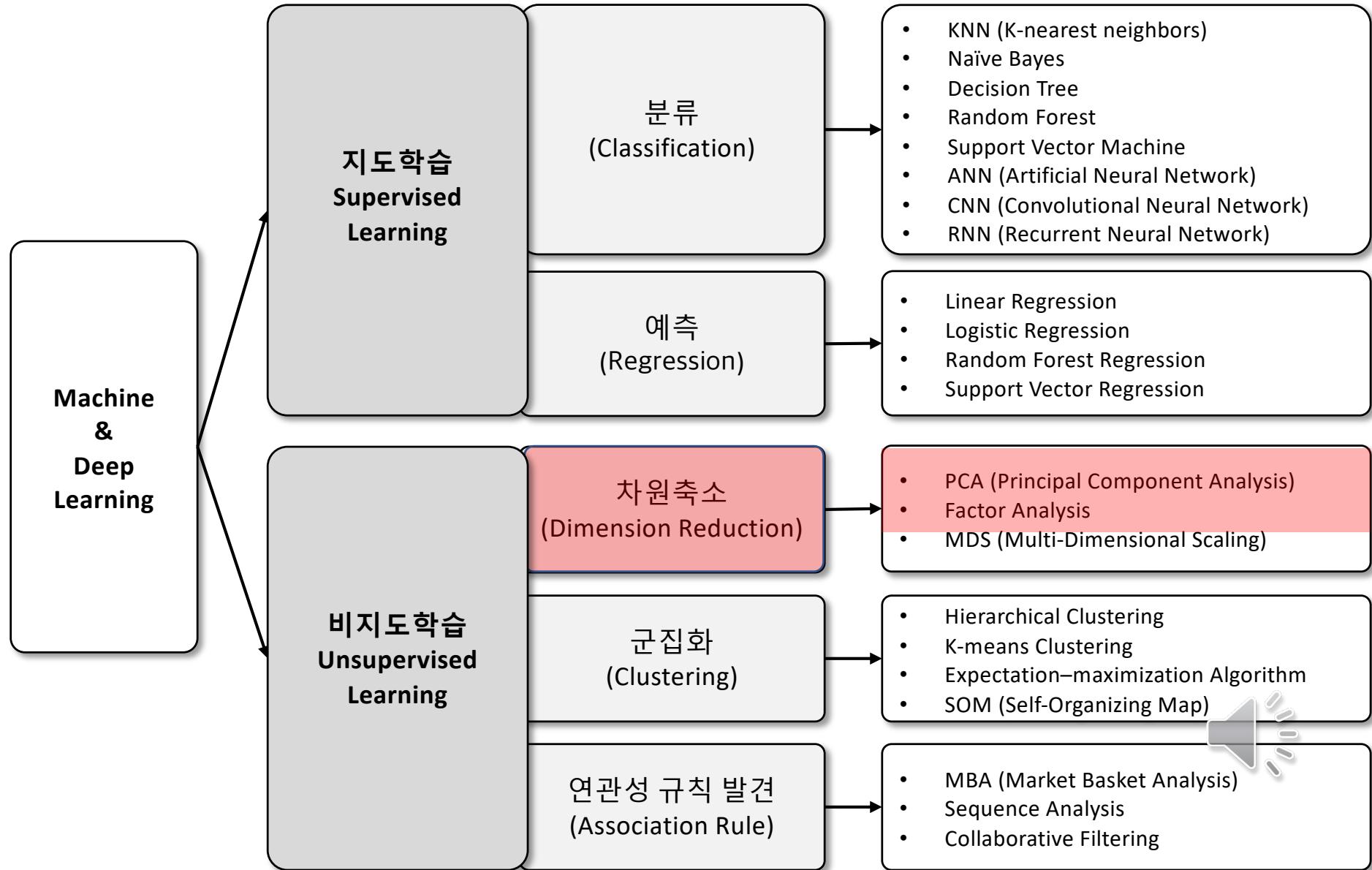
KOREA
UNIVERSITY

CONTENTS

- 차원의 저주 (Curse of dimensionality)
- 차원 축소 (Dimension Reduction)
 - PCA (Principal Component Analysis)
 - Factor Analysis



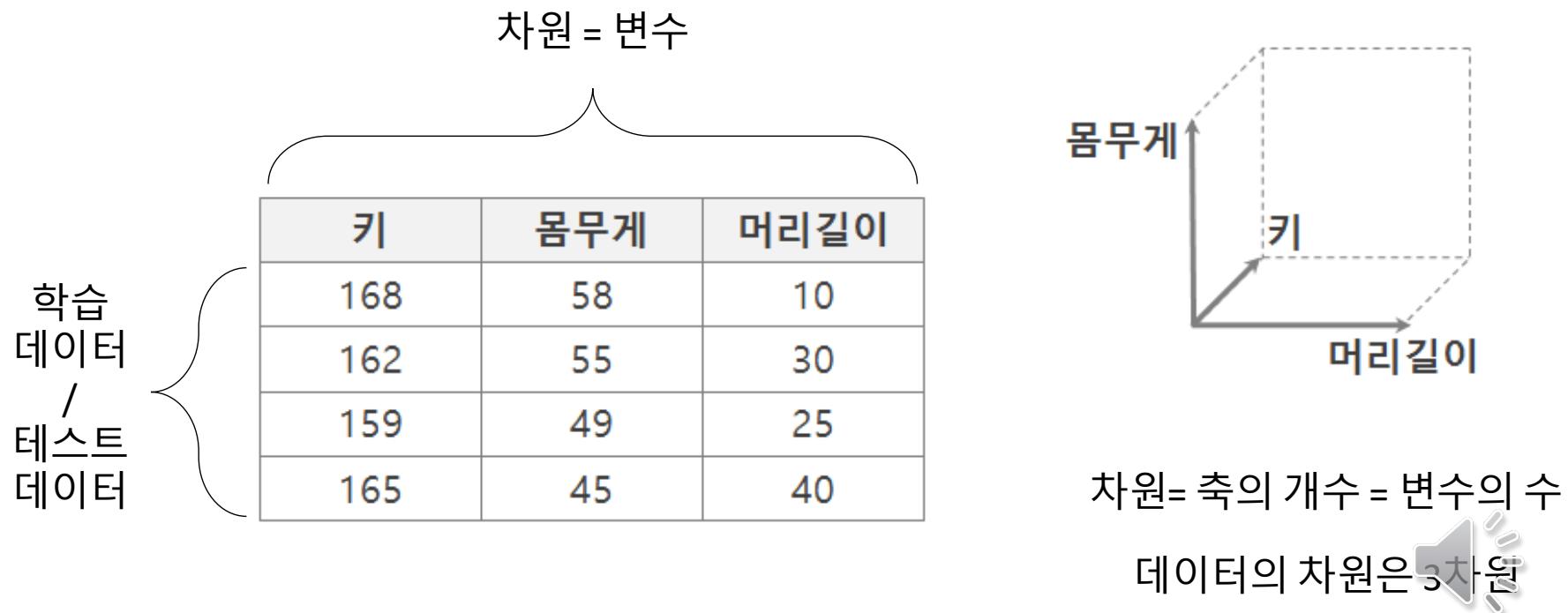
Machine Learning/Deep Learning



Curse of dimensionality

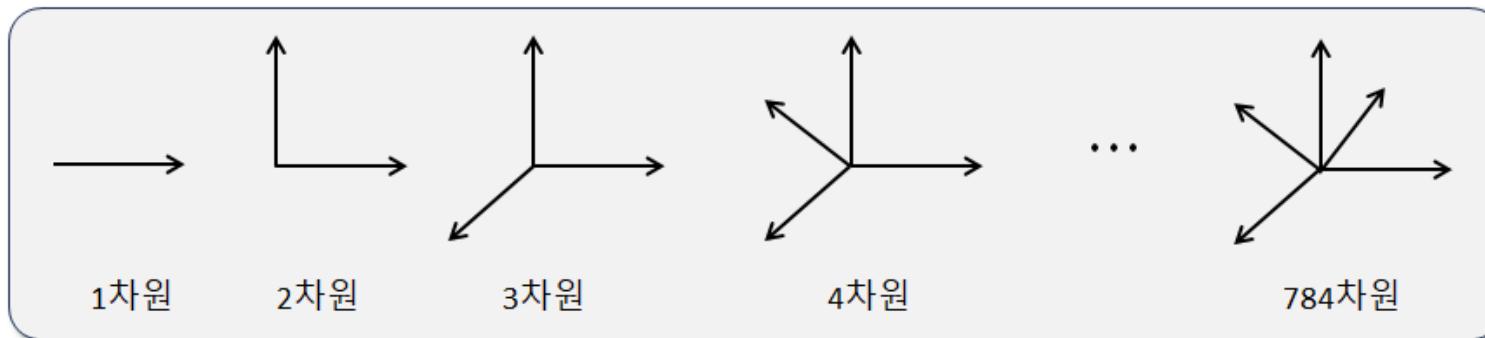
■ Dimension

- 공간 내에 있는 점 등의 위치를 나타내기 위해 필요한 축의 개수
- 차원의 수 증가 = 변수의 수 증가 = 데이터를 표현하는 공간 증가



Curse of dimensionality

■ Dimension



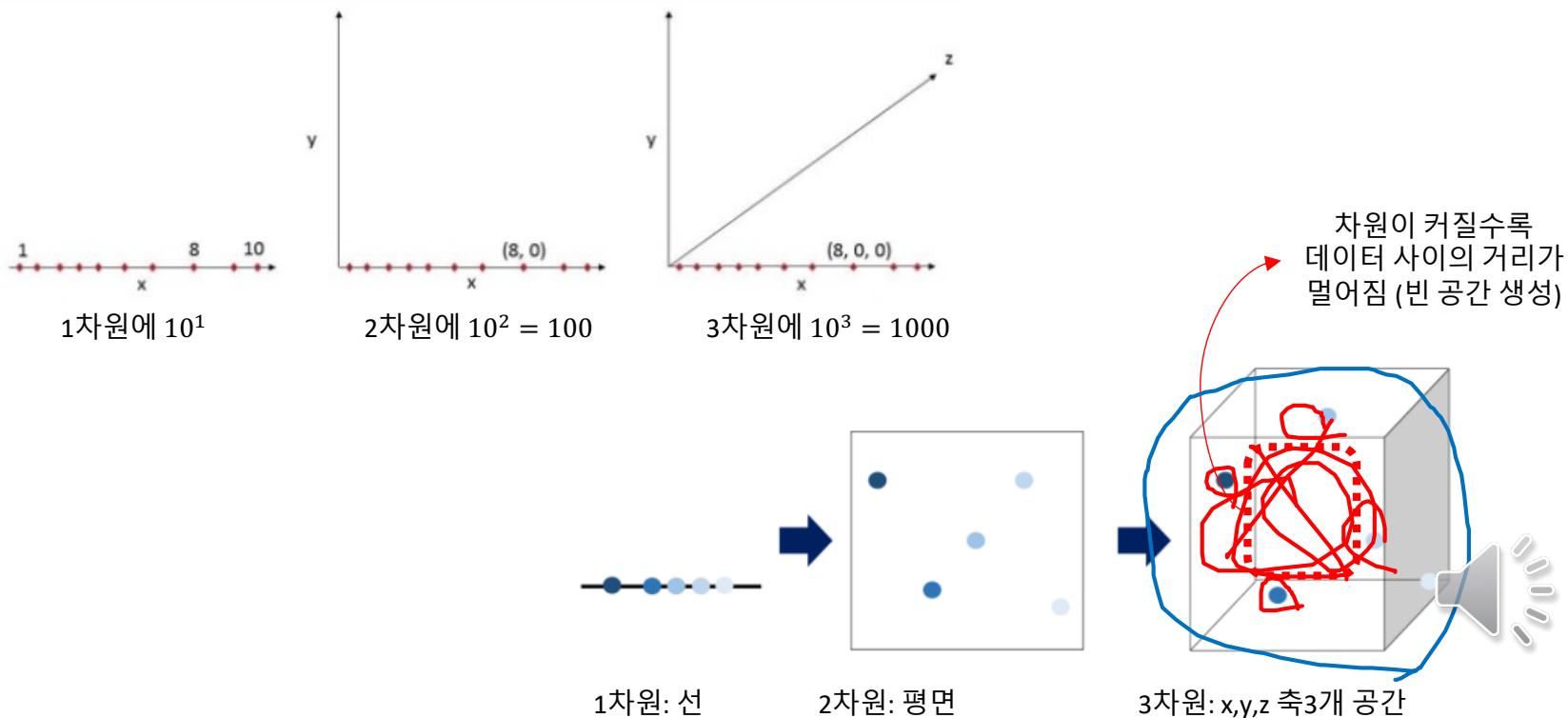
- 3차원 **Location**, $x = (\text{위도}, \text{경도}, \text{고도})$
 - 4차원 **Iris**, $x = (\text{꽃받침 길이}, \text{꽃받침 너비}, \text{꽃잎 길이}, \text{꽃잎 너비})$
 - 784차원 MNIST: Modified National Institute of Standards and Technology database
손으로 쓴 숫자들로 이루어진 대형 데이터베이스, 28×28 행렬
MNIST, $x = (\text{화소1}, \text{화소2}, \dots, \text{화소784})$
(2차원 배열 $[28, 28]$ 의 이미지 데이터를 1차원으로 바꾸면 $28 \times 28 = 784$)



Curse of dimensionality

■ 차원의 저주 (Curse of dimensionality)

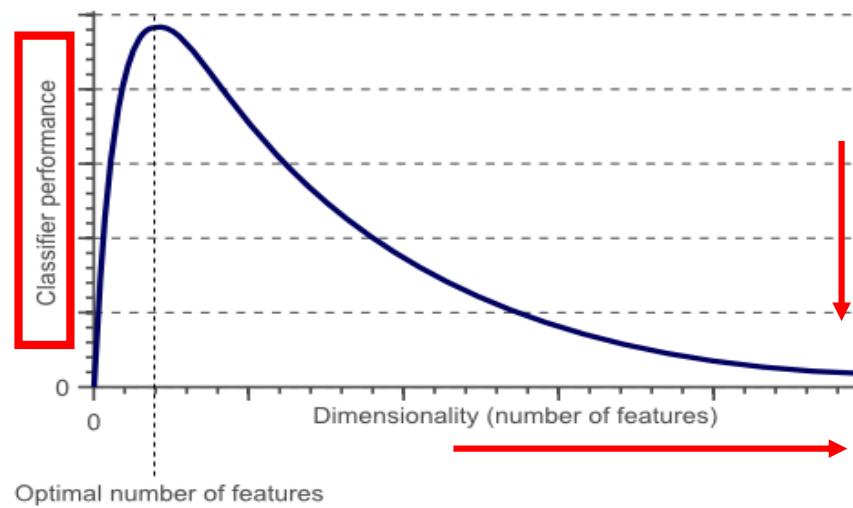
- 수학적 공간 차원 (변수, Feature)이 3차원 이상으로 커지면서, 고차원 데이터를 분석하거나 구성하는 과정에서 발생하는 현상
- 데이터의 수 일정 & 차원 증가 -> 빈 공간 생성 (데이터 밀도 낮아짐 = 성김 현상 (Sparsity) 발생)
- 빈 공간이 생성되었다는 것은 정보 없음을 의미함



Curse of dimensionality

■ 차원의 저주 (Curse of dimensionality)

- 적은 데이터 (정보없음)로 높은 차원의 공간을 표현할 경우 데이터 간의 거리 또한 증가
 - 과적합 (Overfitting) 발생
- 차원이 증가하여 학습 데이터 수가 차원의 수보다 작아질 경우 성능이 저하
(관측치 수 200개, Feature의 수 3000개)
- 일정 차원을 넘으면 분류 알고리즘 (모델)의 성능이 0으로 수렴



- 변수 (차원)이 많을 경우 개별 변수 간 상관 관계가 높을 가능성 존재
 - 양/음의 상관관계가 높을 경우, 변수 하나만 남기고 나머지는 삭제

Curse of dimensionality

■ 차원의 저주 (Curse of dimensionality)

- 같은 데이터 수를 가지고 차원을 높였을 경우?
 - 같은 데이터 수로 차원을 늘려도 정보량은 희박함 (밀집도 낮음, [Data Density](#))
 - 정보량이 희박한 상황에서 모델링을 하면 제대로 된 예측 값 도출이 안됨

차원	공간	학습 데이터	밀집도
1	10	8	80%
2	100 \leftarrow x 10	8	8% \leftarrow % 10
3	1000 \leftarrow	8	0.8% \leftarrow



Curse of dimensionality

■ 차원의 저주 (Curse of dimensionality)

□ 차원축소를 하는 이유

① 데이터 처리를 하기 위한 비용, 시간, 자원, 용량의 문제

- 분석시간 증가
- 불필요한 변수의 저장

② 차원이 높을 경우 발생하는 과적합 문제 해소

- 모델 복잡도 증가
- 민감도 증가 (노이즈가 커질 여지가 많음)

③ 높은 차원의 데이터셋은 시각화 분석이 어려움

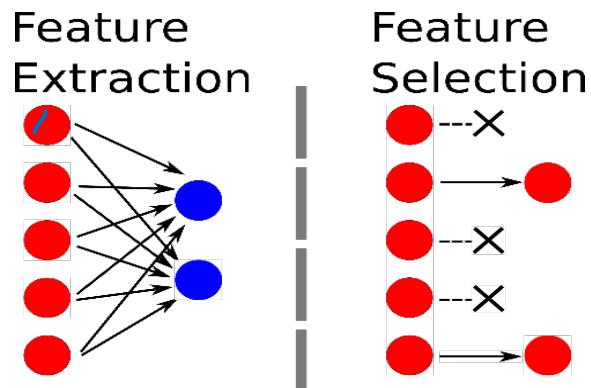
- 차원이 적을수록 내부 구조 파악이 수월함 & 직관적으로 시각화 용이



Dimension Reduction

■ 차원 축소

- 많은 변수 (Feature)로 구성된 다차원의 데이터 셋 -> 차원을 축소시킴
- 차원 축소를 통해 새로운 차원의 데이터 세트를 생성
- 데이터 밀집도↑, 학습 데이터 셋 크기가 상대적으로 커짐 ↑
- 차원 축소 방법
 - 피처 선택 (Feature Selection)
 - 피처 추출 (Feature Extraction)



Dimension Reduction

■ 차원 축소

□ 차원 축소 방법

피처 선택 (Feature Selection)

- 모든 Features 중 필요한 것들만 선택하는 기법
- 중첩되는 변수 검색 시 상관 분석(Correlation)을 통해, 상관계수가 +/-로 높은 경우 하나만 선택

피처 추출 (Feature Extraction)

- 높은 차원의 Raw Features를 더 필요한 요소로 추출하는 기법
- 기존의 Feature로 새로운 Feature 생성
 - ✓ 주성분분석 (PCA) / 인자분석 (Factor Analysis)

□ 차원 축소의 장점

- 차원 축소를 3차원 이하로 할 경우, 시각적으로 데이터를 압축하여 표현하는데 용이
- 학습 데이터의 크기가 줄어들기에 학습에 필요한 처리 능력 (컴퓨터 성능, 작업 시간) 도 줄일 수 있음



Dimension Reduction

■ 차원 축소를 위한 접근 방법

- Projection (투영)
- Manifold Learning (매니폴드 학습)

■ 차원 축소 알고리즘

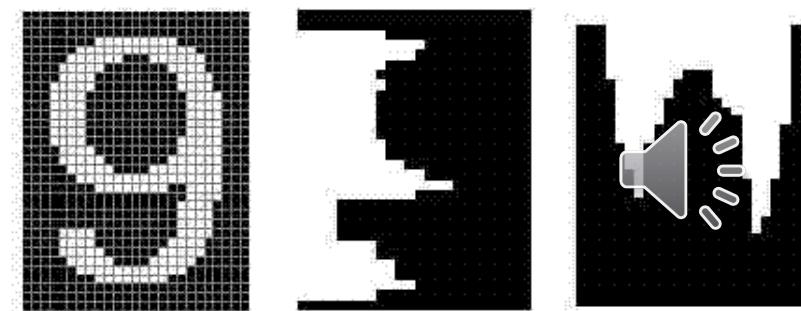
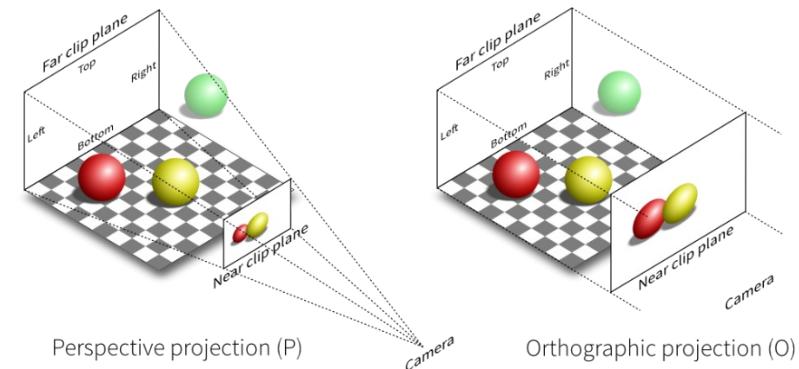
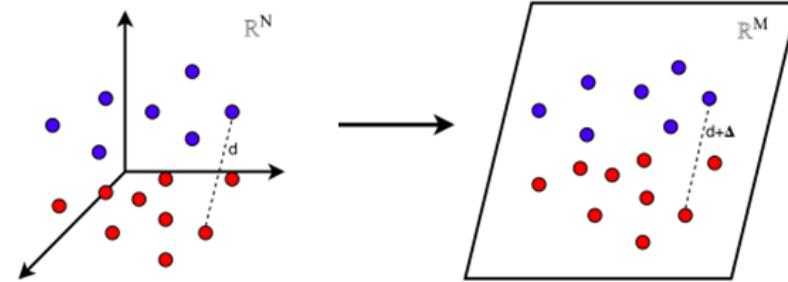
- PCA (Principal Component Analysis, 주성분 분석)
 - 데이터의 분산을 데이터의 특성으로 정의
- Factor Analysis (인자 분석)
- LDA (Linear Discriminant Analysis, 선형판별분석)
 - 차원축소와 함께 데이터 분류도 수행
- MDS (Multidimensional Scaling, 다차원 척도법)
 - 데이터 객체 사이의 거리를 데이터의 특성으로 정의
- Isomap



Dimension Reduction

■ Projection (투영)

- 학습 데이터 셋이 고차원 공간에서 저차원 부분 공간에 위치
- 고차원의 데이터의 특성 중 일부 특성으로 데이터를 표현
- 3차원 공간상의 데이터를 2차원 부분 공간으로 투영
 - > 2차원 데이터셋으로 생성
- 이진 영상을 수평이나 수직 방향으로 투영
 - 수평이나 수직 방향의 라인상에 존재하는 픽셀값이 1인 픽셀의 개수를 세는 것



Dimension Reduction

■ Manifold Learning (다양성 학습)

□ Manifold: 고차원 공간에 내재한 저차원 공간

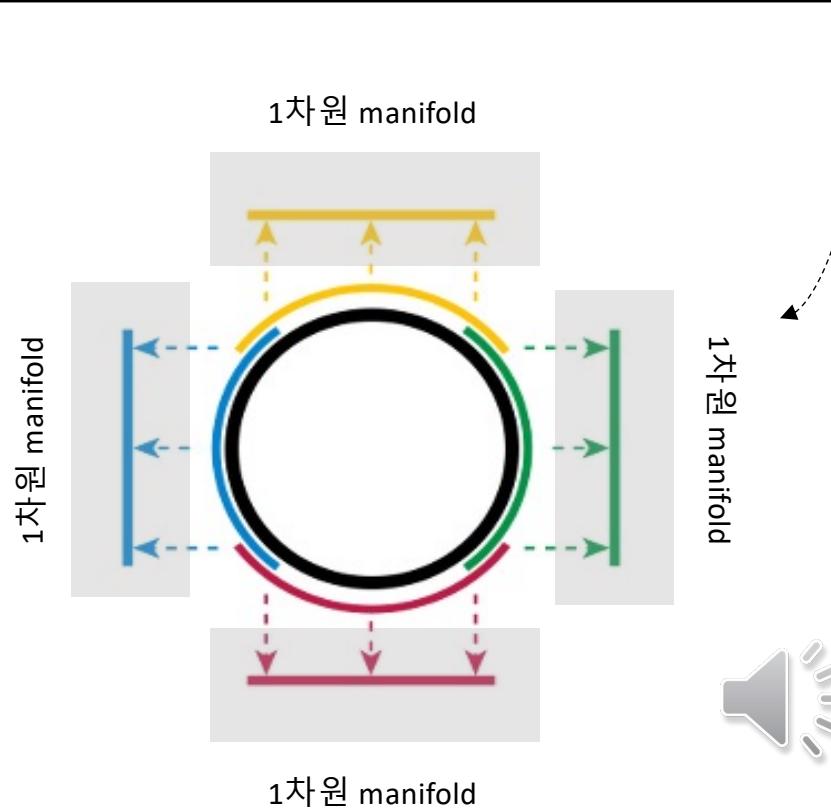
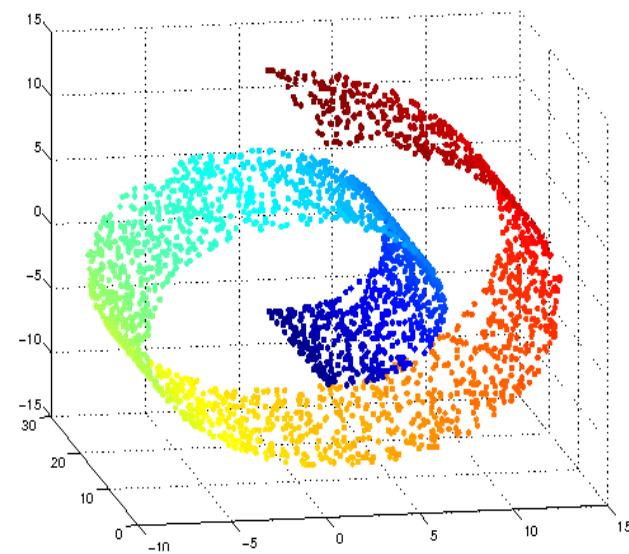
- Manifold는 일반적으로 비선형 구조
- 특정한 점을 중심으로 인근만 살펴보면 선형 구조에 가까움



Dimension Reduction

■ Manifold Learning (다양성 학습)

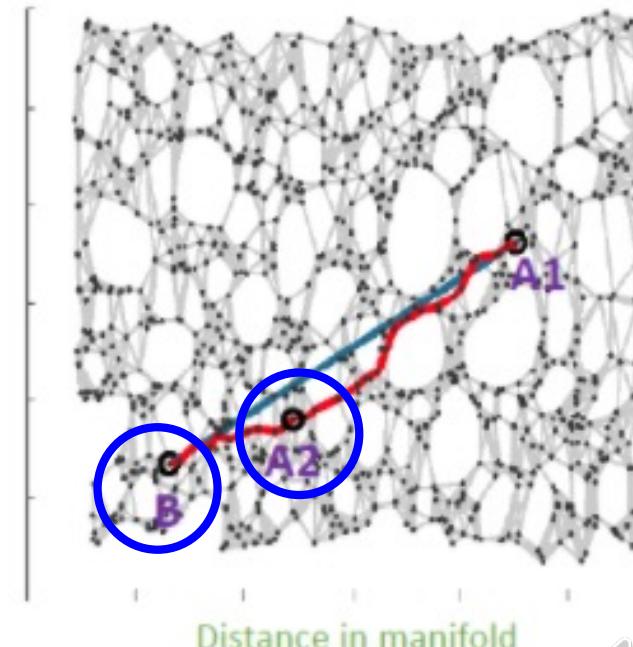
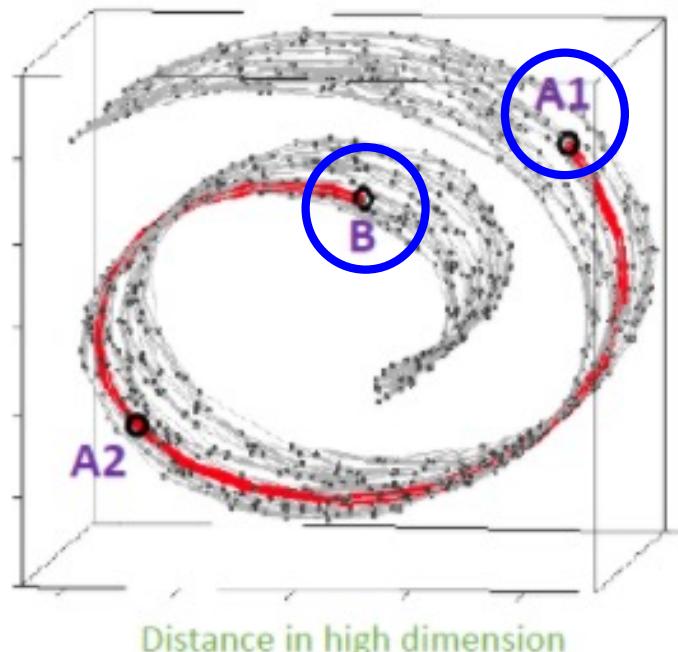
- 고차원 데이터를 데이터 공간에 뿌리면 sample들을 잘 아우르는 subspace가 있을 것이라는 가정에서 학습을 진행하는 방법
- 고차원 데이터를 잘 표현해주는 manifold를 통해, 샘플 데이터의 중요한 특징을 파악
- 모든 점에 대해서 국소적으로 직선과 같은 구조를 가지는 1차원 manifold 라 할 수 있음



Dimension Reduction

■ Manifold Learning (다양성 학습)

- 고차원 공간에서 A1과 B는 가까운 거리에 존재하지만, 의미론적으로는 다를 수 있음
- 차원의 저주로 인해 고차원에서의 유의미한 거리 측정이 어려움

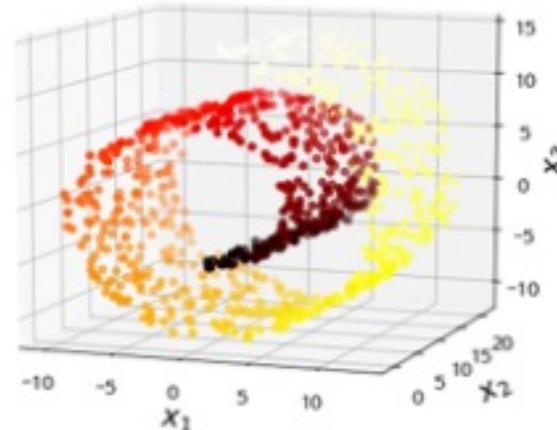


차원 축소 후 B는 A2와 거리상 가까운 것을 알 수 있음

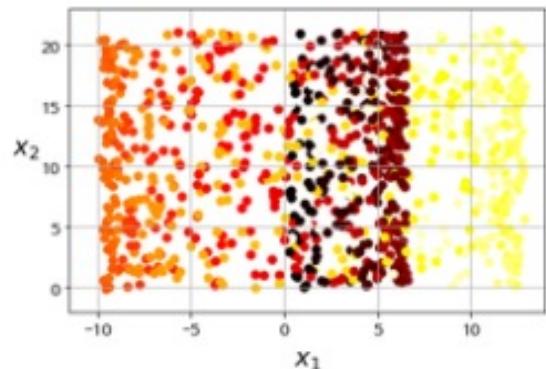
Dimension Reduction

■ Manifold Learning (다양성 학습)

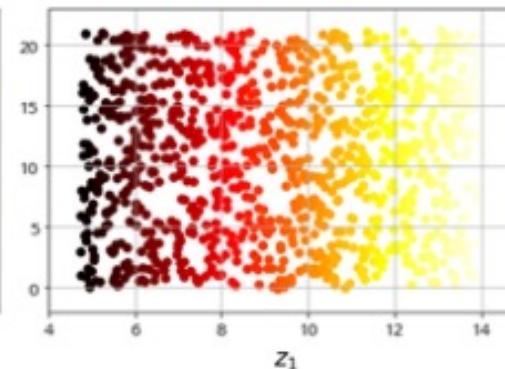
- 대부분의 차원 축소 알고리즘이 이러한 manifold 를 모델링하는 방식으로 동작



고차원 (3차원)



저차원 (2D 투영)



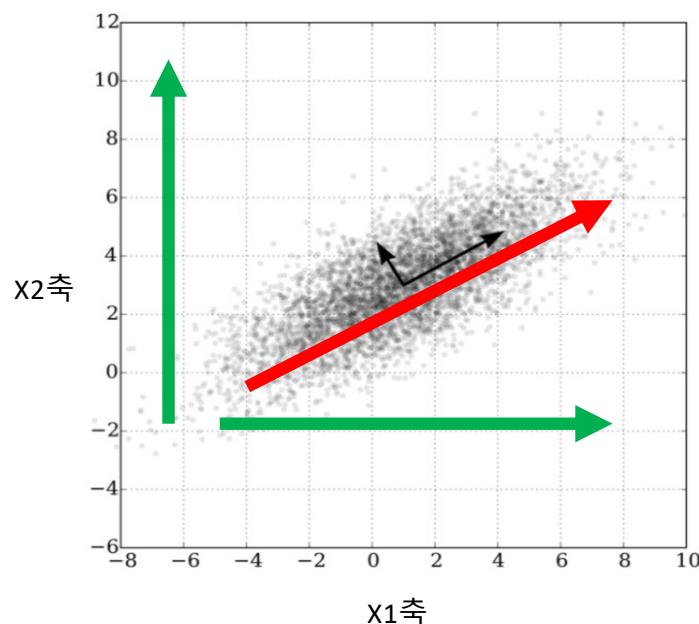
2D unrolling

Dimension Reduction

■PCA (Principal Component Analysis)

- 주성분 분석

- 가장 대표적인 차원 축소 방법
- 데이터의 분산을 데이터의 특성으로 정의하고 이를 최대한 보존하기 위한 방법론
- 특정 기저에 모든 데이터가 사영 되었을 시, 사영된 데이터의 퍼진 정도가 최대인 기저 (분산이 가장 큰 방향 벡터)를 추출함
- 여러 변수 간에 존재하는 상관관계 측정 후 데이터를 대표하는 주성분을 추출해 차원을 축소함



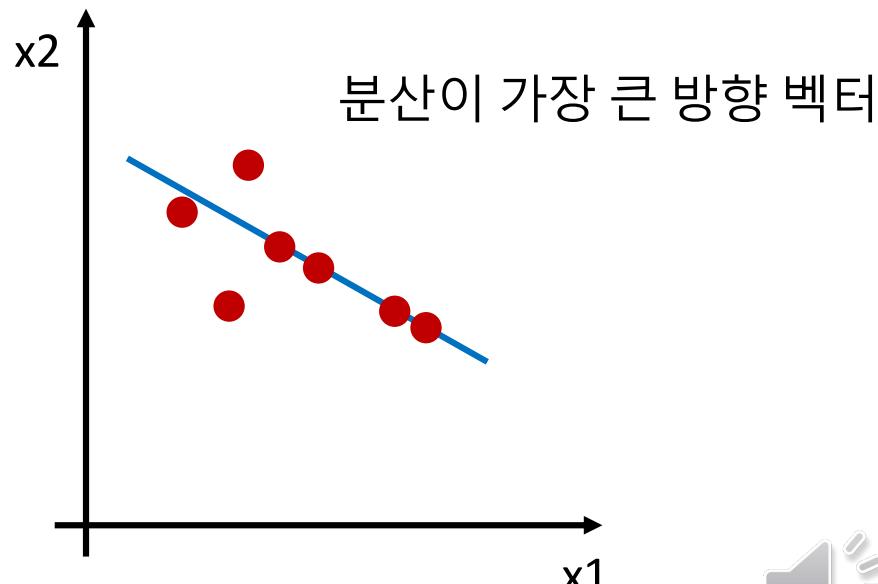
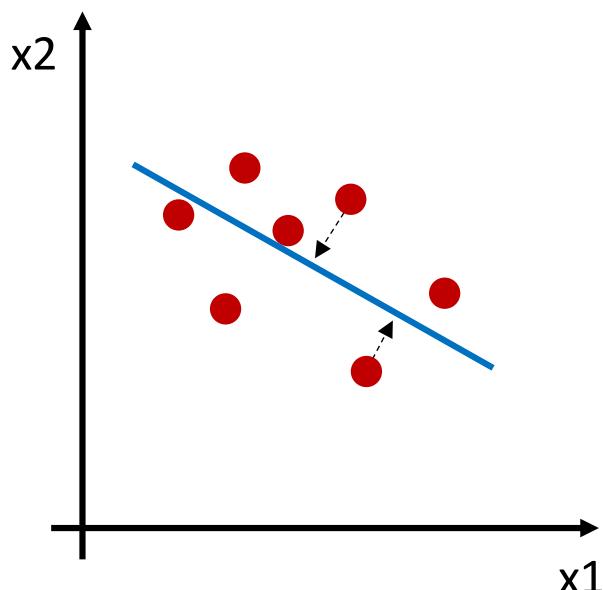
- 가장 높은 분산을 가지는 데이터의 축을 찾아 차원을 축소함
- x, y 축 보다는 **빨간색화살표 방향의 축**이 데이터를 더 잘 표현함
-> **주성분**이라 할 수 있음



Dimension Reduction

■PCA (Principal Component Analysis) 동작원리

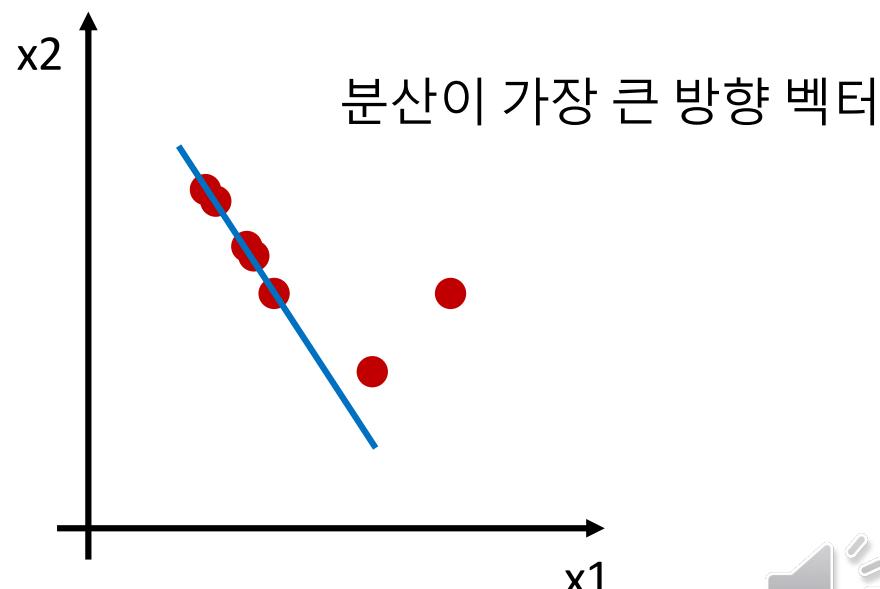
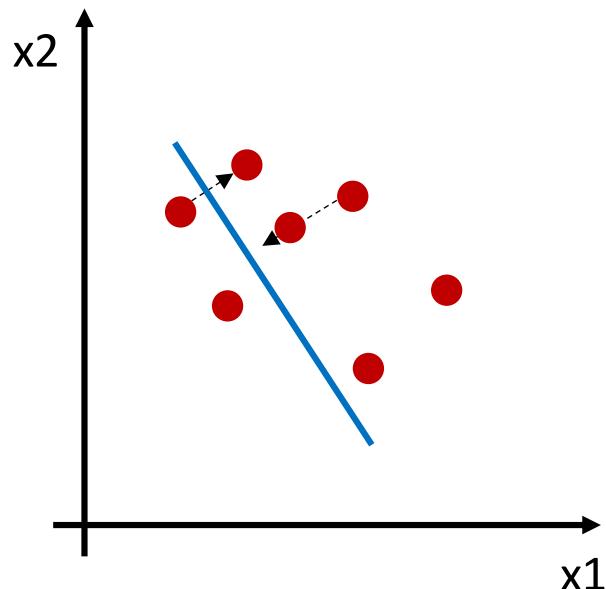
- 정보의 유실을 막으면서 차원을 축소
- 분산이 가장 넓은 곳을 찾아 직선으로 표시
- 데이터 특성에 대한 정보 손실을 최소화



Dimension Reduction

■PCA (Principal Component Analysis) 동작원리

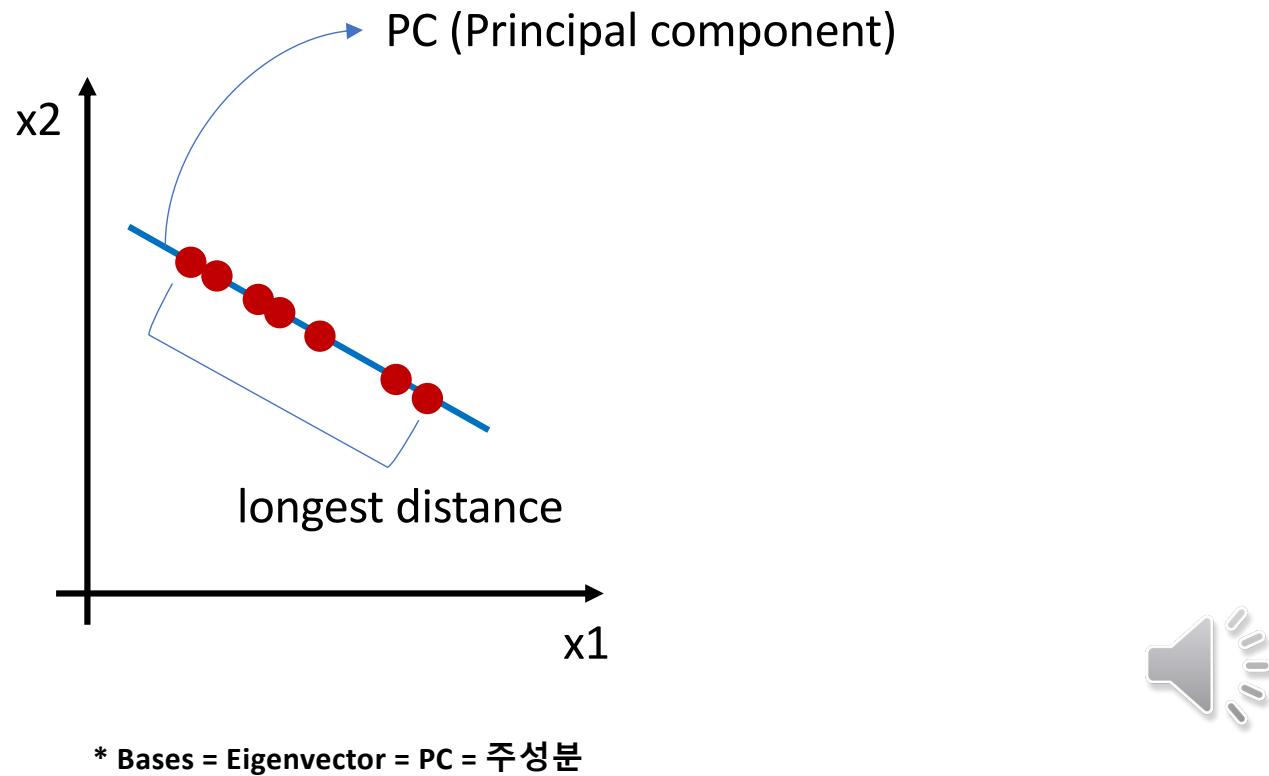
- 데이터 특성에 대한 정보 손실?



Dimension Reduction

■PCA (Principal Component Analysis) 동작원리

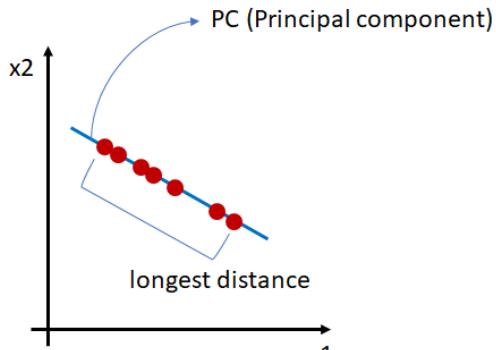
- 각 데이터가 최소로 겹치도록 PC (주성분) 직선을 긋게 됨
- 방향 (벡터)는 보존하고 스케일만 변화하는 벡터
- x_1, x_2 축으로 차원의 축소된 거리 < PC 선 상의 거리 (어떤 의미?)



Dimension Reduction

■PCA (Principal Component Analysis) 동작원리

- 각 데이터가 최소로 겹치도록 **PC (주성분)** 직선을 긋게 됨
- 방향 (벡터)는 보존하고 스케일만 변화하는 벡터
- x1, x2 축으로 차원의 축소된 거리 < PC 선 상의 거리** (어떤 의미?)



* Bases = Eigenvector = PC = 주성분

*공분산:

둘 이상의 변량 (Variance)이 연관성을 가지며 분포하는 모양을 전체적으로 나타낸 분산

- 각 데이터들이 가지고 있는 Feature (x_1, x_2)들의 공분산 행렬 (Covariance Matrix)에 있는 Eigenvector
 - 차원의 수만큼 Eigenvector가 존재 (주성분)
 - 2차원은 2개의 Eigenvector
- * Bases = Eigenvector = PC = 주성분

square matrix

$$Ax = \lambda x$$

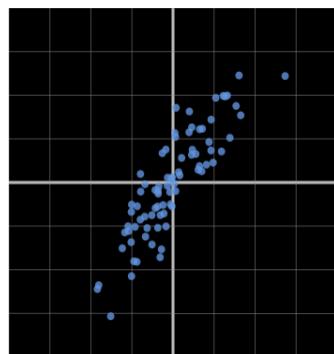
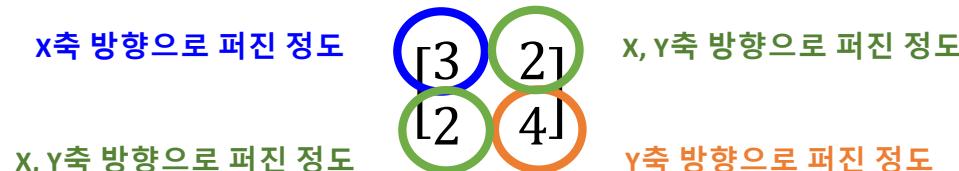
eigenvalue
eigenvector

$$\begin{matrix} A & x = \lambda x \\ \begin{pmatrix} 4 & 2 \\ 3 & 5 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \end{pmatrix} = 7 \begin{pmatrix} 2 \\ 3 \end{pmatrix} \end{matrix}$$

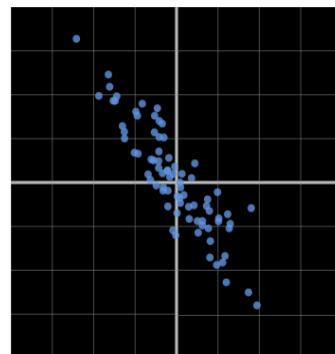
행렬 A (정방행렬, 공분산행렬)에 따라 어떤 선형행렬로 변환되는지 결정

Dimension Reduction

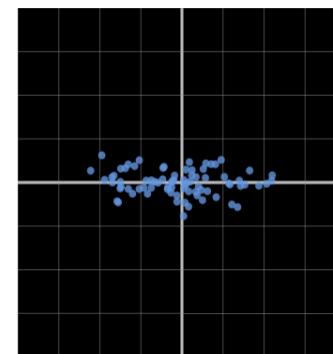
■PCA (Principal Component Analysis) 동작원리



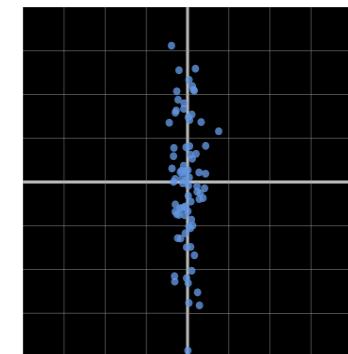
$$\text{Covariance Matrix 1} : \begin{bmatrix} 3 & 2 \\ 2 & 4 \end{bmatrix}$$



$$\text{Covariance Matrix 2} : \begin{bmatrix} 3 & -2 \\ -2 & 4 \end{bmatrix}$$



$$\text{Covariance Matrix 3} : \begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\text{Covariance Matrix 4} : \begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix}$$

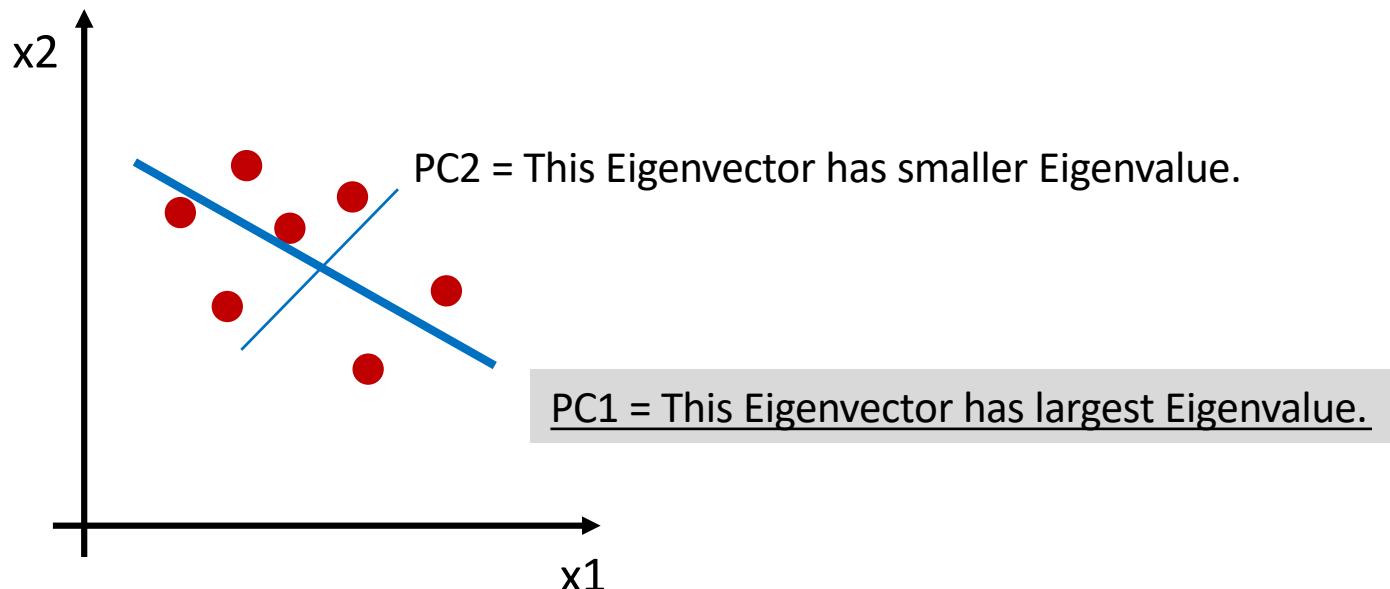
- 데이터 벡터를 어떤 벡터에 내적하는 것이 최적의 결과를 주는가?
 - 공분산 행렬의 고유값 (λ)과 고유 벡터 (X)를 구함으로써 가능



Dimension Reduction

■PCA (Principal Component Analysis) 동작원리

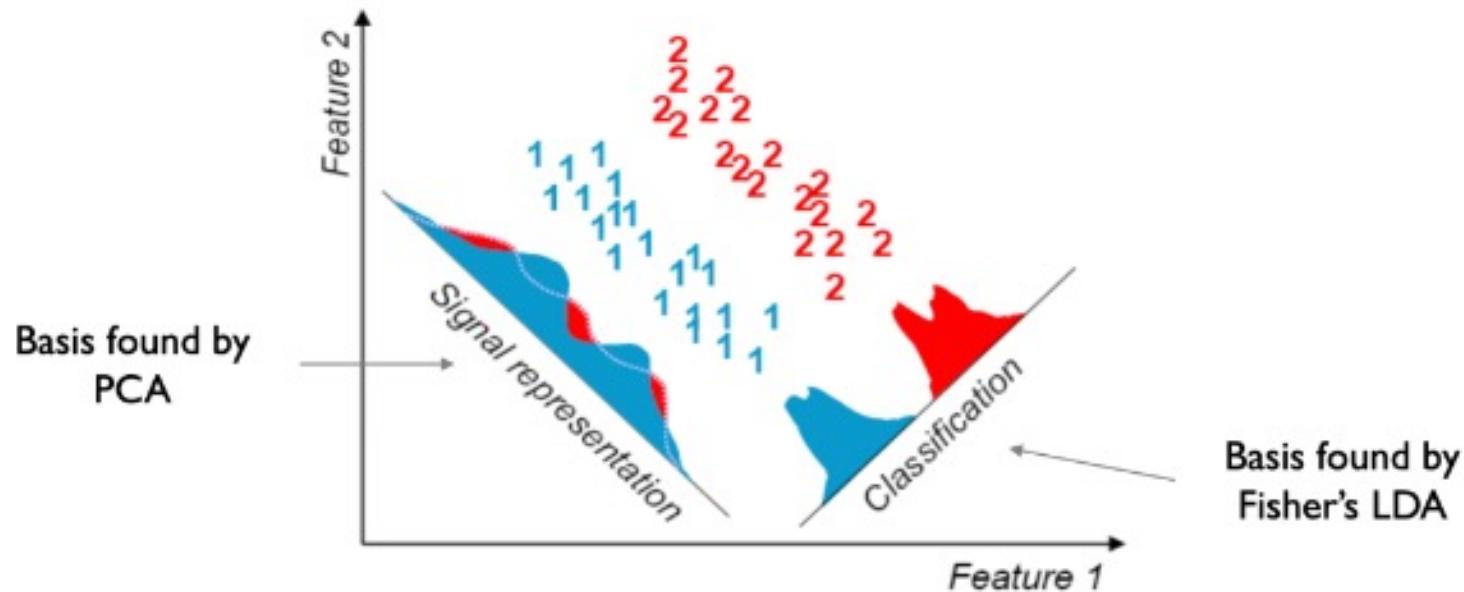
- PC (Principal Component) 가 2개 존재할 경우
- 분산이 큰 Eigen Vector를 선택 -> 기준으로 데이터의 차원 축소



Dimension Reduction

■PCA (Principal Component Analysis) 동작원리

- PCA는 최대 분산량을 보존하는 알고리즘일 뿐, 분류 목적은 없음



LDA는 학습 단계에서 클래스를
가장 잘 구분하는 축을 학습



Dimension Reduction

■파이썬 이용한 PCA 실습

- PCA example (scikit-learn).ipynb 실습코드
- 다운로드 후 참고



Thank you





Introduction to Artificial Intelligence [AICS223]

Dimension Reduction & Association Rule (W11)

Prof. Mee Lan Han (aeternus1203@gmail.com)

고려대학교

인공지능사이버보안학과



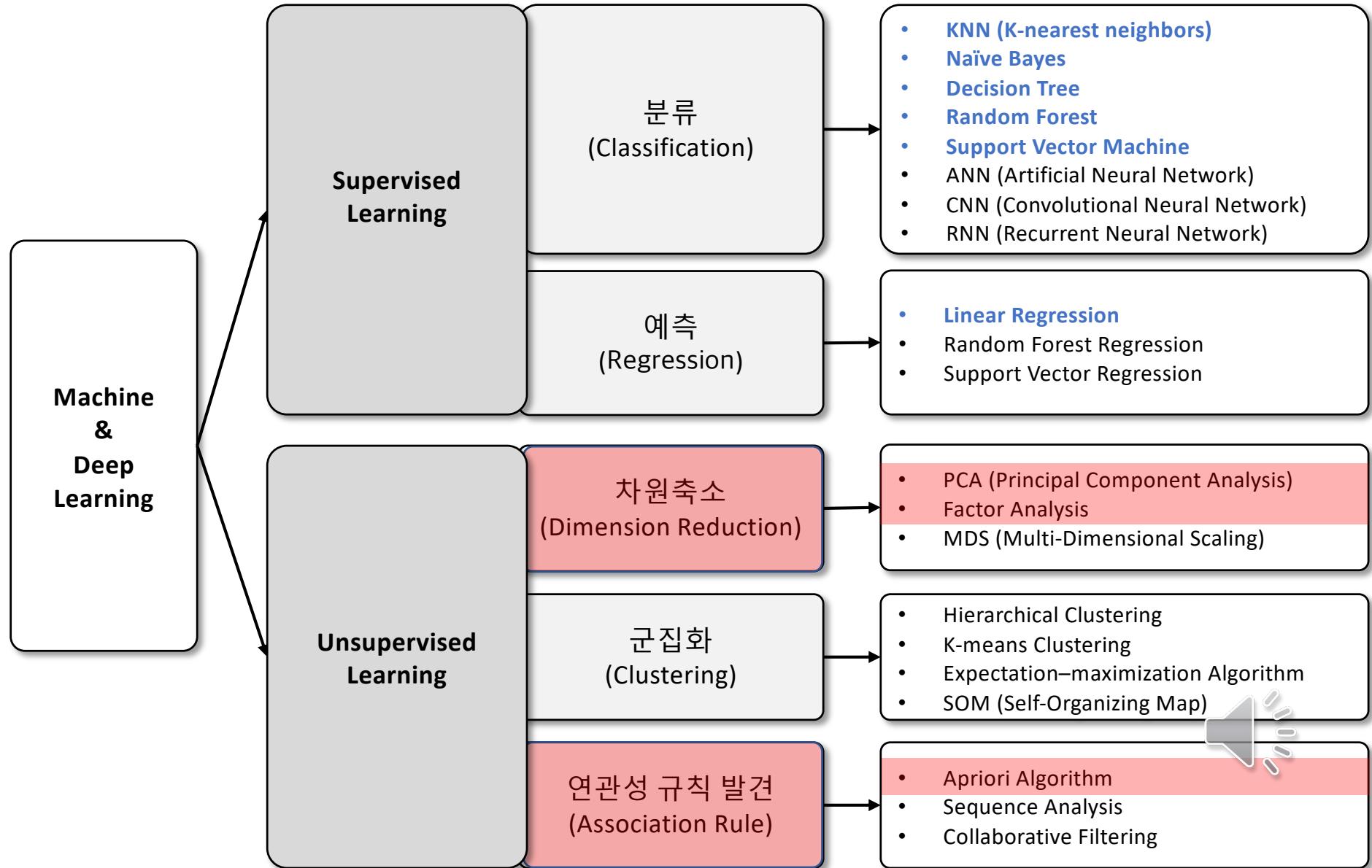
KOREA
UNIVERSITY

CONTENTS

- 차원의 저주 (Curse of dimensionality)
- 차원 축소 (Dimension Reduction)
 - PCA (Principal Component Analysis)
 - Factor Analysis
- 연관성 규칙 (Association Rule)
 - A Priori Algorithm



Machine Learning/Deep Learning



Dimension Reduction

■ Factor Analysis (인자분석, 요인분석)

- 여러 개의 서로 관련이 있는 변수들로 측정된 자료에서 해당 변수들을 설명할 수 있는 새로운 공통 변수를 파악하는 통계적 분석방법
 - 변수들 간의 상관관계를 고려하여 서로 유사한 변수들끼리 묶어주는 방법
 - 예) 7개의 과목 (수학, 물리, 영어, 중국어, 독어, 작곡, 연주)에 대한 평가
 - 영어, 중국어, 독어 -> 외국어 능력
 - 수학, 물리 -> 수리 능력
 - 작곡, 연주 -> 음악적 능력
- > 7개의 변수 (과목)에서 3개의 잠재 변수로 재구성



Dimension Reduction

■ Factor Analysis (인자분석, 요인분석)

□ 목적

1) 입력변수들의 특성 파악

- 많은 입력 변수들 간의 상관관계 파악

2) 새 변수 생성 (전처리 가능)

- 본래의 많은 변수보다 더 적절한 변수를 생성하여 회귀분석/분류분석에 사용
- 새로운 (잠재)변수를 생성하여 분석을 수행하게 되면 더 좋은 분석 결과(모델)를 얻을 수 있음

3) 데이터 축소

- 데이터의 축소를 통하여 분석결과에 대한 해석이 용이
- 몇 개의 축소된 (잠재)변수를 사용할 경우 결과 모델이 단순
- 모델을 이해하고 설명하기가 쉬움



Dimension Reduction

■ Factor Analysis (인자분석, 요인분석)

- 주성분 분석(PCA)과 요인 분석(FA)의 공통점과 차이점
- 공통점
 - 관측된 여러 개의 변수들로부터 소수의 새로운 변수들을 생성
 - 차원 축소의 방법으로 활용
- 차이점

PCA

- 변수간의 중요성 있음
- 주로 제1주성분,
제2주성분 등으로 구분
- 변수간의 순서가 주어짐

FA

- 변수들은 기본적으로 **대등한 관계**를 갖음
- 변수의 **중요도의 차이가 없음**
- 변수간의 **순서 없음**



Dimension Reduction

■ Factor Analysis (인자분석, 요인분석)

□ 주성분 분석(PCA)과 요인 분석(FA)의 차이점

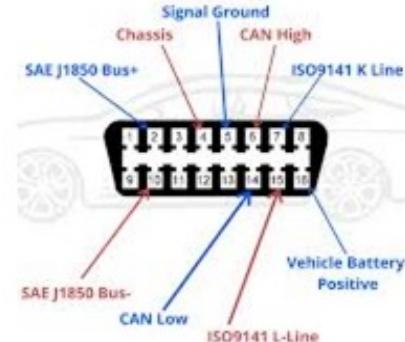
	PCA	FA
생성되는 변수의 수	<ul style="list-style-type: none">• 주성분• 보통 2개 선택 (제1주성분, 제2주성분)	<ul style="list-style-type: none">• 지정할 수 없음• 데이터 상관성에 따라 변수들이 군집
생성되는 변수의 의미	<ul style="list-style-type: none">• 보통 2개 선택 (제1주성분, 제2주성분)	<ul style="list-style-type: none">• 생성된 변수에 적절한 이름 부여
생성된 변수들의 관계	<ul style="list-style-type: none">• 제1주성분이 가장 중요하고, 그 다음 제2주성분이 중요하게 취급• 변수들 간의 중요성의 순위가 존재	<ul style="list-style-type: none">• 새롭게 생성된 (잠재)변수들은 기본적으로 대등한 관계
분석방법의 의미	<ul style="list-style-type: none">• 목표 변수를 잘 예측/분류하기 위하여 원래 변수들의 선형 결합으로 이루어진 몇 개의 주성분(변수)들을 찾아 냈	<ul style="list-style-type: none">• 데이터가 주어지면 변수들을 비슷한 성격들로 묶어서 새로운 [잠재]변수들을 생성

Dimension Reduction

■ Factor Analysis (인자분석, 요인분석)

□ 예시) 차량의 주행 데이터

- 차량 OBD II port를 통해 추출된 51개의 변수



연료소모량(mcc)	액셀포지션(%)	스로틀포지션(%)	단기연료보정 뱅크1(%)	흡입공기압(kPa)	액셀포지션-필터링값(%)
연료압력(kPa)	장기연료보정 뱅크1(%)	엔진속도(rpm)	엔진토크 - 보정후(%)	마찰토크(%)	플라이휠토크-조정후(Nm)
계산 부하값(%)	엔진토크-최소지시(%)	엔진토크-최대지시(%)	플라이휠토크(Nm)	토크환산계수(Nm)	표준토크비율
에멀플러그제어요청	컴퓨레서 동작	토크컨버터 속도(rpm)	현재기어	미션오일온도(°C)	휠속도 - 전방좌측(km/h)
클러치 동작확인	컨버터클러치	기어선택	차량속도(km/h)	종방향 가속도(m/s2)	브레이크스위치
스로틀포지션-절대값(%)	엔진압력유지시간(Min)	연료차단 억제	엔진 연료차단	브레이크실린더압력(bar)	스티어링휠 각도(°)
현재점화시기(°)	엔진냉각수온도(°C)	엔진공회전목표속도(rpm)	엔진 토크(%)	도로경사(%)	
점화각도 지연요청-TCU(°)	엔진 토크 제한 요청-TCU(Nm)	엔진속도증가요청-TCU(%)	목표엔진속도-락업모듈(rpm)	횡방향 가속도(m/s2)	
휠속도 - 후방우측(km/h)	휠속도 - 전방우측(km/h)	휠속도 - 후방좌측(km/h)	토크컨버터 터빈속도-필터링전	스티어링휠 회전속도(° /s)	

생성된 변수에
적절한 이름을
부여할 수 있음

New Variables	Variable extracted from CarbigsP (OBD II scanner)
Fuel-related	Intake air pressure (kPa), Calculated load value (%), Engine torque, Accelerator position (%), Flywheel torque (Nm), Fuel consumption (mcc)
Engine-related	Wheel speed, Engine idle target speed (rpm), Vehicle speed (km/h), Torque convertor speed (rpm), Torque convertor turbine speed (before filtering)
Gear-related	Current gear position
Wheel-related	Steering wheel angle (%), Accelerator speed-lateral

데이터가 주어
지면 변수들을
비슷한 성격들
로 묶어서 새로
운 [집재]변수들
을 생성



Dimension Reduction

■ R 이용한 FA (Factor Analysis) 실습

데이터 가져오기

```
secu_com_finance <- read.table(file = "secu_com_finance.csv", header = T, sep = ",",
stringsAsFactors = FALSE)
```

	company	V1	V2	V3	V4	V5
1	SK증권	2.43	11.10	18.46	441.67	0.90
2	교보증권	3.09	9.95	29.46	239.43	0.90
3	대신증권	2.22	6.86	28.62	249.36	0.69
4	대우증권	5.76	23.19	23.47	326.09	1.43
5	동부증권	1.60	5.64	25.64	289.98	1.42
6	메리츠증권	3.53	10.64	32.25	210.10	1.17
7	미래에셋증권	4.26	15.56	24.40	309.78	0.81
8	부국증권	3.86	5.50	70.74	41.36	0.81
9	브릿지증권	4.09	6.44	64.38	55.32	0.32
10	삼성증권	2.73	10.68	24.41	309.59	0.64
11	서울증권	2.03	4.50	42.53	135.12	0.59
12	신영증권	1.96	8.92	18.48	441.19	1.07
13	신흥증권	3.25	7.96	40.42	147.41	1.19
14	우리투자증권	2.01	10.28	17.46	472.78	1.25
15	유화증권	2.28	3.65	63.71	56.96	0.12
16	한양증권	4.51	7.50	63.52	57.44	0.80
17	한화증권	3.29	12.37	24.47	308.63	0.57
18	현대증권	1.73	7.57	19.59	410.45	1.19

>> 변수 V1~V5

- V1 : 총자본 순이익율
- V2 : 자기자본 순이익율
- V3 : 자기자본 비율
- V4 : 부채 비율
- V5 : 자기자본 회전율



Dimension Reduction

■ R 이용한 FA (Factor Analysis) 실습

각 변수들의 표준화 변환 (standardization) - 표준편차 측정

```
secu_com_finance <- transform(secu_com_finance, V1_s = scale(V1),  
V2_s = scale(V2), V3_s = scale(V3), V4_s = scale(V4), V5_s = scale(V5))
```

	company	V1	V2	V3	V4	V5	V1_s	V2_s	V3_s	V4_s	V5_s
1	SK증권	2.43	11.10	18.46	441.67	0.90	-0.5327135	0.38287402	-0.9182381	1.338962052	0.05067071
2	교보증권	3.09	9.95	29.46	239.43	0.90	0.0484285	0.13119119	-0.3116550	-0.074929674	0.05067071
3	대신증권	2.22	6.86	28.62	249.36	0.69	-0.7176223	-0.54506962	-0.3579759	-0.005507479	-0.52973922
4	대우증권	5.76	23.19	23.47	326.09	1.43	2.3994120	3.02882650	-0.6419671	0.530924049	1.51551482
5	동부증권	1.60	5.64	25.64	289.98	1.42	-1.2635436	-0.81207227	-0.5223048	0.278473346	1.48787625
6	메리츠증권	3.53	10.64	32.25	210.10	1.17	0.4358565	0.28220089	-0.1578035	-0.279980329	0.79691205
7	미래에셋증권	4.26	15.56	24.40	309.78	0.81	1.0786347	1.35896567	-0.5906833	0.416898267	-0.19807641
8	부국증권	3.86	5.50	70.74	41.36	0.81	0.7264275	-0.84271191	1.9646859	-1.459668273	-0.19807641
9	브릿지증권	4.09	6.44	64.38	55.32	0.32	0.9289467	-0.63698856	1.6139706	-1.362071712	-1.55236624
10	삼성증권	2.73	10.68	24.41	309.59	0.64	-0.2685580	0.29095507	-0.5901318	0.415569947	-0.66793206
11	서울증권	2.03	4.50	42.53	135.12	0.59	-0.8849208	-1.06156655	0.4090760	-0.804177330	-0.80612490
12	신영증권	1.96	8.92	18.48	441.19	1.07	-0.9465570	-0.09422908	-0.9171353	1.335606296	0.52052637
13	신흥증권	3.25	7.96	40.42	147.41	1.19	0.1893114	-0.30432952	0.2927223	-0.718256002	0.85218918
14	우리투자증권	2.01	10.28	17.46	472.78	1.25	-0.9025311	0.20341322	-0.9733821	1.556456967	1.01802059
15	유화증권	2.28	3.65	63.71	56.96	0.12	-0.6647912	-1.24759298	1.5770241	-1.350606213	-2.10513761
16	한양증권	4.51	7.50	63.52	57.44	0.80	1.2987643	-0.40500265	1.5665468	-1.347250457	-0.22571497
17	한화증권	3.29	12.37	24.47	308.63	0.57	0.2245321	0.66081940	-0.5868232	0.408858436	-0.86140204
18	현대증권	1.73	7.57	19.59	410.45	1.19	-1.1490762	-0.38968283	-0.8559255	1.120698109	0.85218918



Dimension Reduction

■ R 이용한 FA (Factor Analysis) 실습

Correlation analysis (상관관계 분석)

```
cor(secu_com_finance_2[,-1])  
round(cor(secu_com_finance_2[,-1]), digits=3) # 반올림
```

```
> cor(secu_com_finance_2[,-1])  
      V1_s       V2_s       V3_s       V4_s2       V5_s  
V1_s  1.00000000  0.6165153  0.3239780  0.3553930  0.01387883  
V2_s  0.61651527  1.0000000 -0.5124351 -0.4659444  0.42263462  
V3_s  0.32397800 -0.5124351  1.0000000  0.9366296 -0.56340782  
V4_s2 0.35539305 -0.4659444  0.9366296  1.0000000 -0.53954570  
V5_s  0.01387883  0.4226346 -0.5634078 -0.5395457  1.00000000
```

```
> round(cor(secu_com_finance_2[,-1]), digits=3) # 반올림  
      V1_s     V2_s     V3_s     V4_s2     V5_s  
V1_s  1.000   0.617   0.324   0.355   0.014  
V2_s  0.617   1.000  -0.512  -0.466   0.423  
V3_s  0.324  -0.512   1.000   0.937  -0.563  
V4_s2 0.355  -0.466   0.937   1.000  -0.540  
V5_s  0.014   0.423  -0.563  -0.540   1.000
```

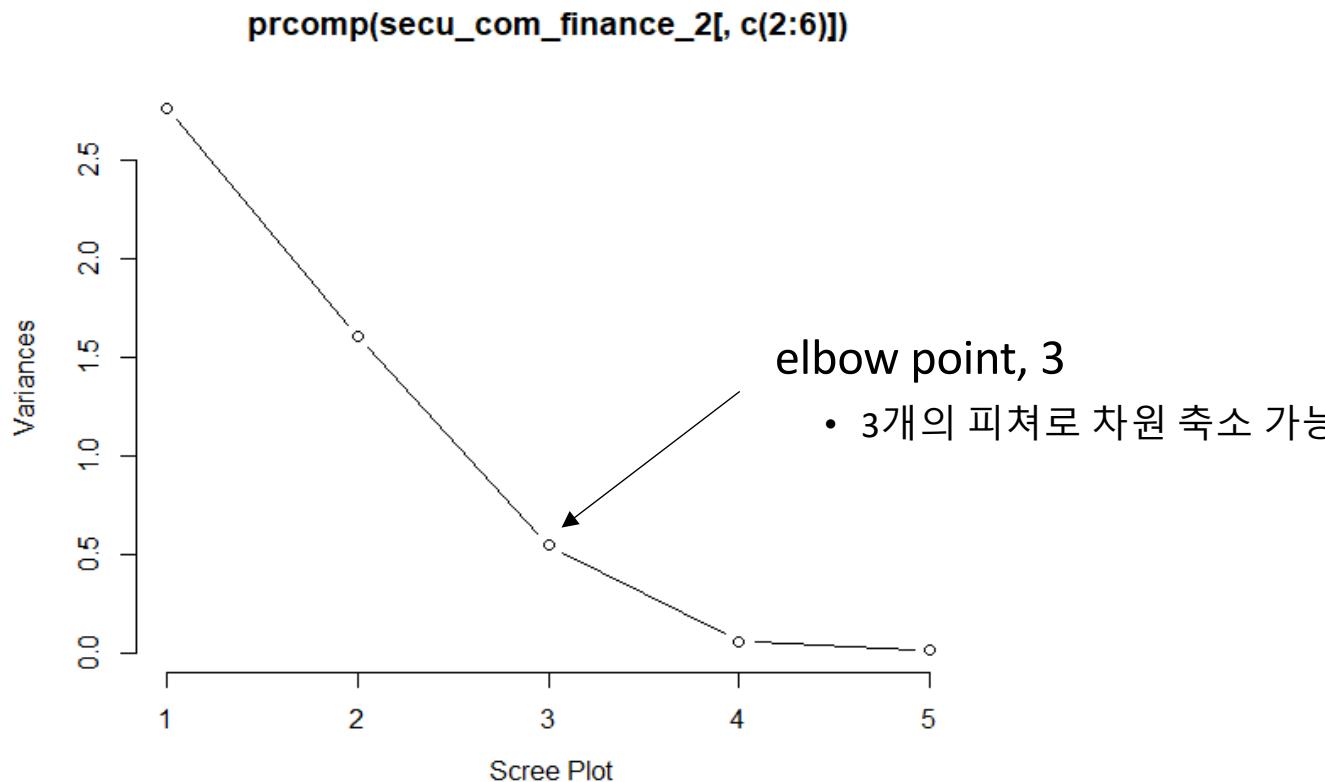


Dimension Reduction

■ R 이용한 FA (Factor Analysis) 실습

Scree Plot

```
plot(prcomp(secu_com_finance_2[,c(2:6)]), type="l", sub = "Scree Plot")
```



Dimension Reduction

■ R 이용한 FA (Factor Analysis) 실습

요인분석(maximum likelihood factor analysis)

```
secu_factanal <- factanal(secu_com_finance_2[,2:6], factors = 3,  
                           rotation = "varimax", scores="regression")
```

```
> # Scree Plot  
> plot(prcomp(secu_com_finance_2[,c(2:6)]), type="l", sub = "Scree Plot")  
> secu_factanal <- factanal(secu_com_finance_2[,2:6], factors = 3,  
+                                rotation = "varimax", # "varimax", "promax", "none"  
+                                scores="regression") # "regression", "Bartlett"  
Error in factanal(secu_com_finance_2[, 2:6], factors = 3, rotation = "varimax", :  
  3 factors are too many for 5 variables
```

요인 회전 (Rotation)

- 최초 주어진 데이터 (행렬)를 수학적으로 회전시키면서 해석이 쉬운 새로운 데이터 셋 (행렬)을 산출하려는 목적



Dimension Reduction

■ R 이용한 FA (Factor Analysis) 실습

요인분석(maximum likelihood factor analysis)

```
secu_factanal <- factanal(secu_com_finance_2[,2:6], factors = 2,  
                           rotation = "varimax", scores="regression")
```

- Factor1은 V3_s, V4_s2, V5_s를 대표함

Loadings:		
	Factor1	Factor2
V1_s	0.252	0.965
V2_s	-0.588	0.792
V3_s	0.979	0.080
V4_s2	0.950	0.120
V5_s	-0.562	0.155

- Factor2는 V1_s, V2_s를 대표함

	Factor1	Factor2
ss Loadings	2.586	1.604
Proportion Var	0.517	0.321
Cumulative Var	0.517	0.838

- Factor1과 Factor 2가 데이터의 대략 84%를 설명하고 있음

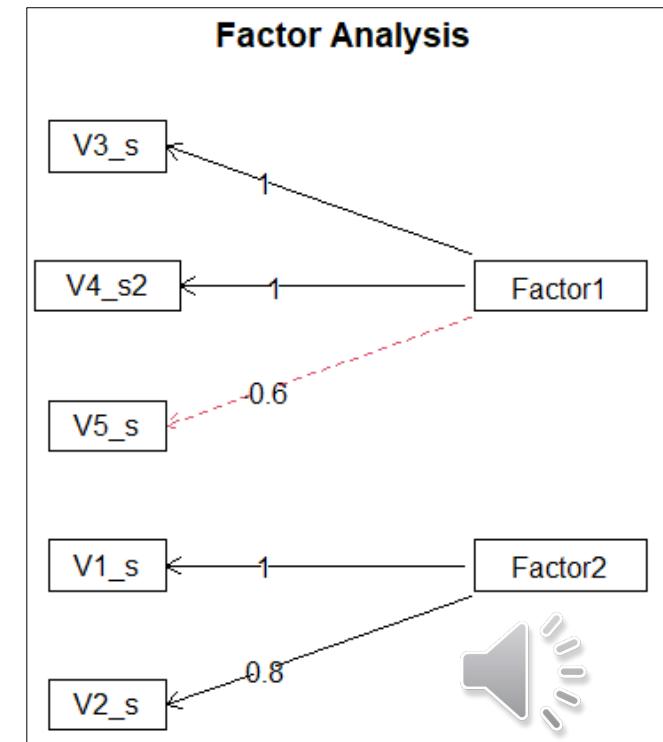
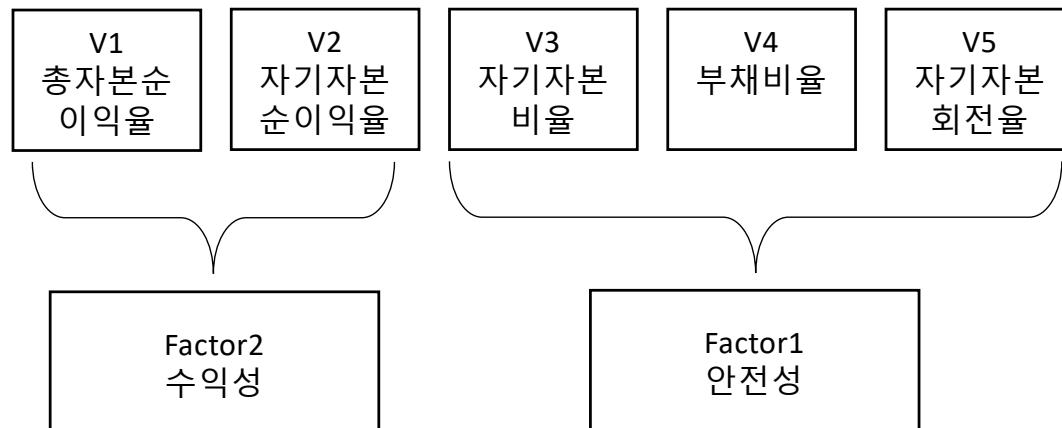


Dimension Reduction

■ R 이용한 FA (Factor Analysis) 실습

요인모형 시각화

```
colnames(secu_factanal$loadings) <- c("Factor1", "Factor2")
fa.diagram(secu_factanal$loadings)
```



Association Rule

■ A Priori Algorithm

□ Y값 (종속 변수, 라벨링)이 없는 상태에서 데이터 속에 숨겨져 있는 패턴이나 규칙을 찾아내는 비지도 학습

- ✓ 암 데이터에서 빈번히 발생하는 DNA 패턴과 단백질 서열을 검색
- ✓ 사기성 신용카드 및 보험의 이용과 결합되어 발생하는 구매/의료비 청구의 패턴 분석
- ✓ 유통업에서는 장바구니 분석을 통해 상품 추천

□ 특징

- "If (조건) then (결과)" 형태 ($\text{if } A \rightarrow B$)
- 이해하기가 쉽고 명료하며 실질적으로 적용하는데 용이함
- 데이터로부터 계산된 연관 규칙들은 확률에 근거하고 있음
- 너무 많은 조합이 발생하기 때문에 수시로 변경되는 많은 항목 집합 (item set)을 고려해야 함

* 항목 집합 (Item set) : 전체 Item (I) 중에서 가능한 부분 집합



Association Rule

■ A Priori Algorithm

□ 연관 규칙 평가를 위한 주요 지표

- **지지도(Support)**

- 특정 아이템들이 데이터에서 발생하는 빈도

- **신뢰도(Confidence)**

- 두 아이템의 연관 규칙이 유용한 규칙일 가능성의 척도
 - 규칙의 신뢰성에 대한 척도

- **향상도(Lift)**

- 두 아이템의 연관 규칙이 우연인지 아닌지를 나타내는 척도



Association Rule

■ A Priori Algorithm

- 연관 규칙 평가를 위한 주요 지표

- 지지도(Support)

- 특정 아이템들이 전체 데이터셋에서 발생하는 빈도
- 전체 거래 항목 중 상품 A와 상품 B를 포함하는 거래하는 비율
- 지지도값이 클 수록 자주 발생하는 거래

$$S(A \rightarrow B) = P(A \cap B) = A \text{와 } B \text{를 포함하는 거래 수} / \text{전체 거래 수}$$

User ID	Transaction ID	Items
980	1	계란, 우유
1132	2	계란, 기저귀, 맥주, 사과
1987	3	우유, 기저귀, 맥주, 콜라
2980	4	계란, 우유, 맥주, 기저귀
4102	5	계란, 우유, 맥주, 콜라

연관규칙 {계란, 맥주} -> {기저귀}

A

B

$$S(A \rightarrow B) = \frac{2}{5} = 0.4$$


Association Rule

■ A Priori Algorithm

- 연관 규칙 평가를 위한 주요 지표

- 신뢰도(Confidence)

- 상품 A를 포함하는 거래 중 A와 B가 동시에 거래되는 비중 (조건부 확률)
- 신뢰도값이 클 수록 A상품 구매 시 B상품 구매율이 높음

$$C(A \rightarrow B) = \frac{P(B|A)}{P(B)} = \frac{P(A \cap B)}{P(A)} = A \text{와 } B \text{가 동시에 포함된 거래 수} / A \text{가 포함된 거래 수}$$

User ID	Transaction ID	Items
980	1	계란, 우유
1132	2	계란, 기저귀, 맥주, 사과
1987	3	우유, 기저귀, 맥주, 콜라
2980	4	계란, 우유, 맥주, 기저귀
4102	5	계란, 우유, 맥주, 콜라

연관규칙 {계란, 맥주} -> {기저귀}

A

B

$$C(A \rightarrow B) = \frac{\frac{2}{5}}{\frac{3}{5}} = \frac{2}{3} = 0.667$$


Association Rule

■ A Priori Algorithm

□ 연관 규칙 평가를 위한 주요 지표

- 향상도(Lift)

- 상품 A를 구매할 시 그 거래가 상품 B를 포함하는 경우와 상품 B가 임의로 구매되는 경우의 비율
- **상품A와 상품B의 구매 패턴이 독립적인지, 서로 상관이 있는지를 의미**

$$L(A \rightarrow B) = \frac{P(A \cap B)}{P(A)} * \frac{1}{P(B)} = \frac{P(A \cap B)}{P(A)P(B)} = \text{신뢰도}^*$$

User ID	Transaction ID	Items
980	1	계란, 우유
1132	2	계란, 기저귀, 맥주, 사과
1987	3	우유, 기저귀, 맥주, 콜라
2980	4	계란, 우유, 맥주, 기저귀
4102	5	계란, 우유, 맥주, 콜라

- 신뢰도: 상품 A를 포함하는 거래 중 A와 B가 동시에 거래되는 비중

연관규칙 {계란, 맥주} -> {기저귀}

A B

$$L(A \rightarrow B) = \frac{2}{3} * \frac{1}{2} = \frac{2}{3} * \frac{5}{5}$$

$$= \frac{0.667}{0.6} \quad \text{Speaker icon}$$

$$= 1.111666$$

상호 연관성이 높음, 양의 상관관계

Association Rule

■ A Priori Algorithm

- 연관 규칙 평가를 위한 주요 지표
 - 향상도(Lift)
 - 향상도(lift)는 $Lift(X \rightarrow Y)$ 값과 $Lift(Y \rightarrow X)$ 의 값이 서로 같음
 - 이런 특성을 가지는 척도를 대칭적 척도(symmetric measure)라고 함

Lift > 1	Lift = 1	0 < Lift < 1
상호 연관성이 높음, 양의 상관관계	아무런 상호 관계가 없음	품목간의 음의 상관관계
예) 맥주와 치킨	예) 맥주와 후추	예) 지사제와 변비약

연관규칙 $\{\underline{\text{계란}}, \underline{\text{맥주}}\} \rightarrow \{\underline{\text{기저귀}}\}$

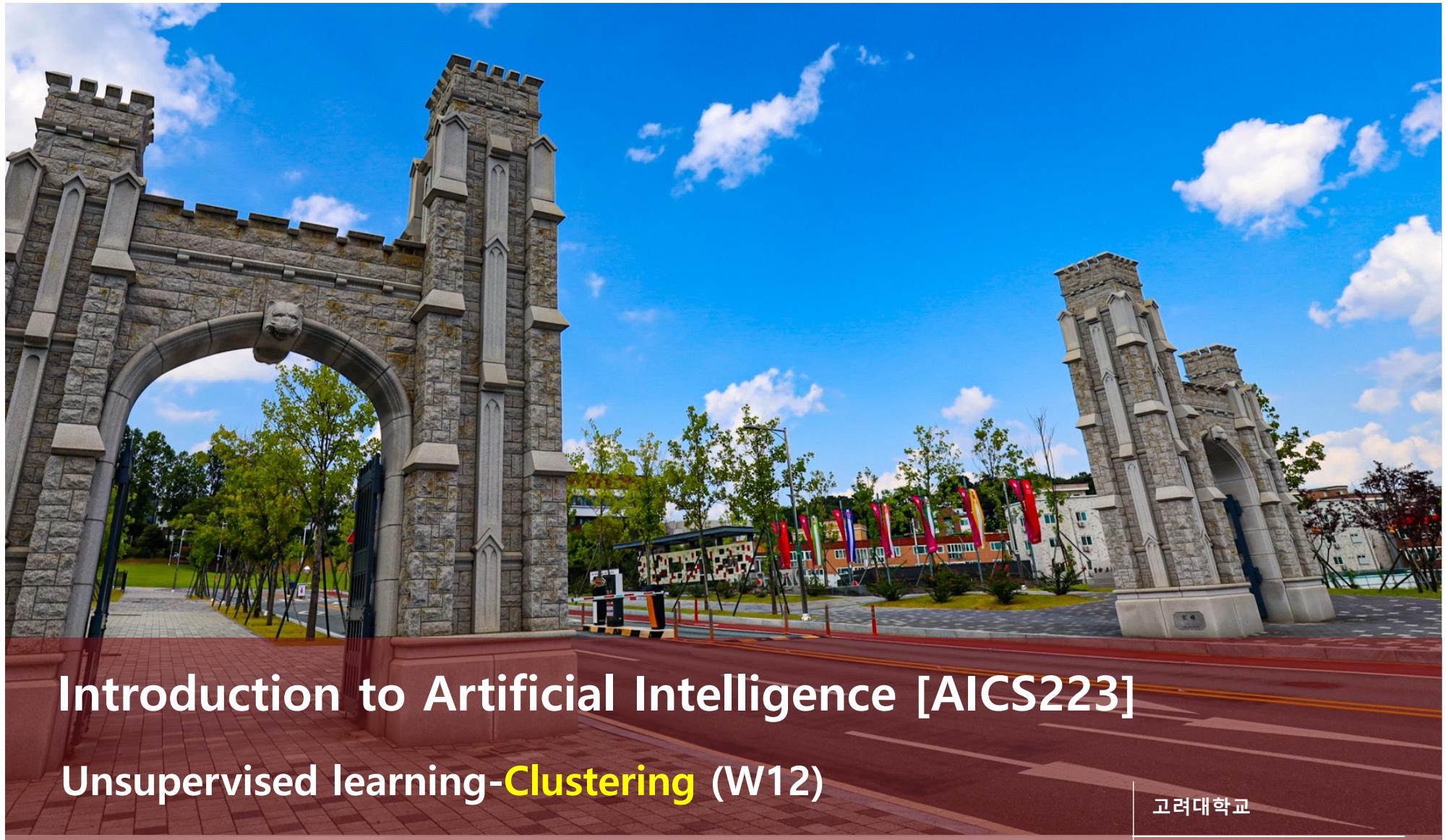
A B

$$L(A \rightarrow B) = \frac{\frac{2}{3}}{\frac{3}{5}} = \frac{0.667}{0.6} = 1.111666$$



Thank you





Introduction to Artificial Intelligence [AICS223]

Unsupervised learning-**Clustering** (W12)

Prof. Mee Lan Han (aeternus1203@gmail.com)

고려대학교

인공지능사이버보안학과



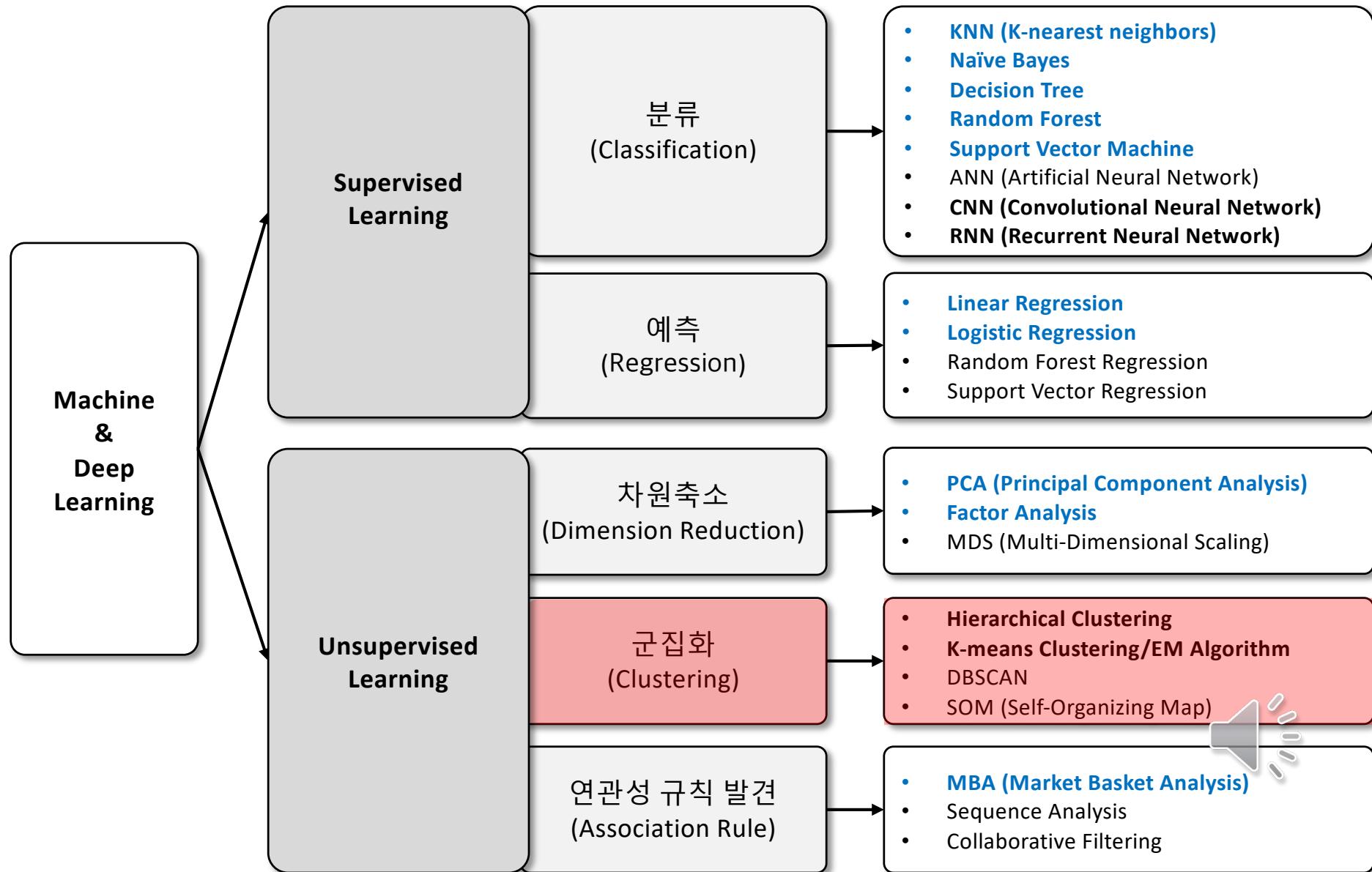
KOREA
UNIVERSITY

CONTENTS

- 군집화 (Clustering)
 - Hierarchical Clustering
 - K-means Clustering/EM algorithm
 - DBSCAN
 - SOM (Self-Organizing Map)



Machine Learning/Deep Learning



Clustering

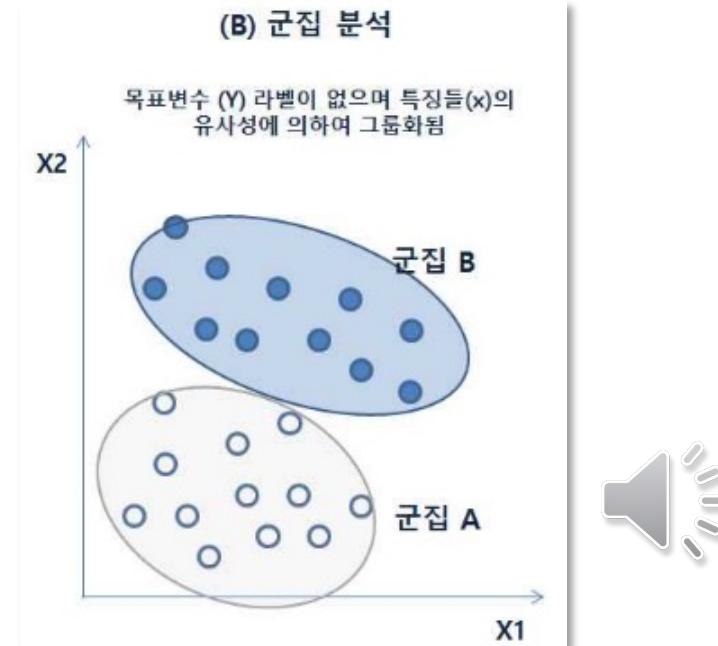
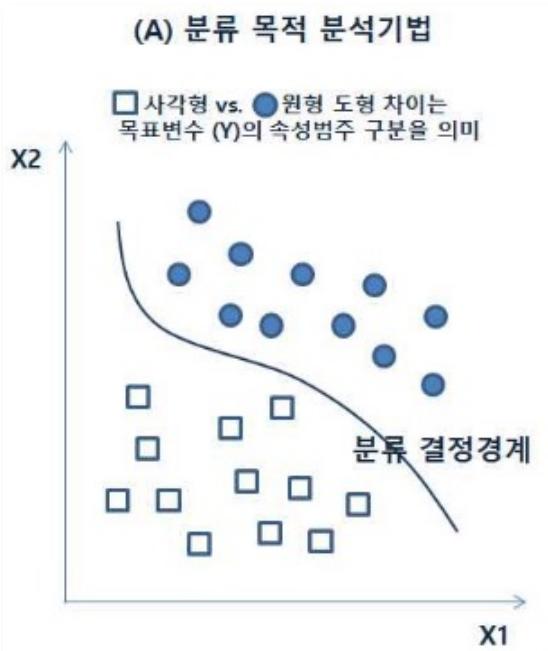
■ 군집화 (Clustering) 개념 및 특징

□ (A) 분류

- 목적변수의 라벨이 주어진 상태에서, 새로운 데이터가 해당 목적변수의 범주 속성 중 어떤 속성으로 분류되는지 판별

□ (B) 군집

- 목적변수가 없는 상태에서 유사한 특성을 가진 개체끼리 그룹화



Clustering

■ 군집화 (Clustering) 개념 및 특징

- 목적변수가 없는 상태에서 유사한 특성을 가진 개체끼리 그룹화
- 각 개체에 대해 관측된 여러 개의 변수(x_1, x_2, \dots, x_p) 값들로부터 n 개의 개체를 유사한 성격을 가지는 몇 개의 군집으로 집단화
- 형성된 군집들의 특성을 파악하여 군집들 사이의 관계를 분석하는 기법
- 오로지 개체들 간의 유사성(similarity)에만 기초하여 군집을 형성

- 군집분석은 이상값 탐지, 심리학, 사회학, 경영학, 생물학 등 다양한 분야에 이용
- 클러스터링은 정답이 없는 **비지도학습 (unsupervised learning)**

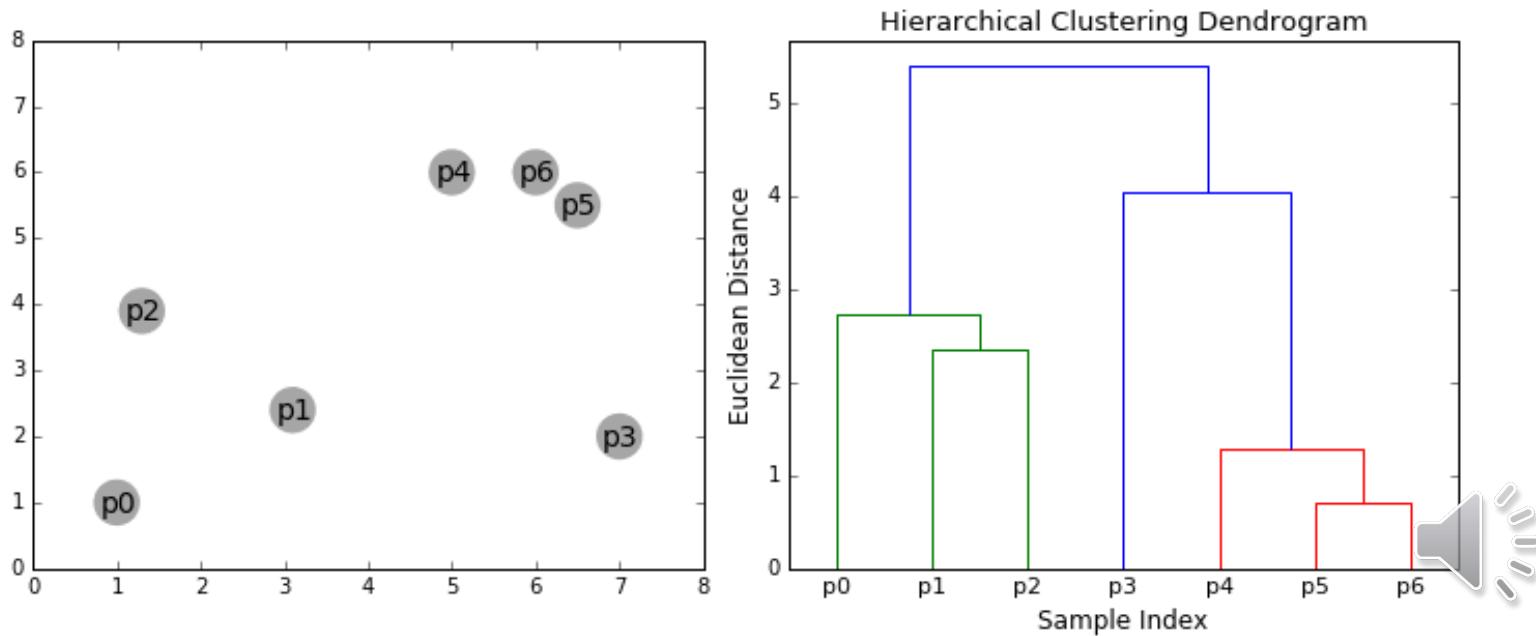
- 비슷한 개체끼리 한 그룹으로, 다른 개체는 다른 그룹으로 묶어 표현
- 다른 머신러닝 알고리즘처럼 **정확도 (Accuracy)** 지표로 평가할 수 없음



Clustering

■ 계층적 군집화 (Hierarchical Clustering)

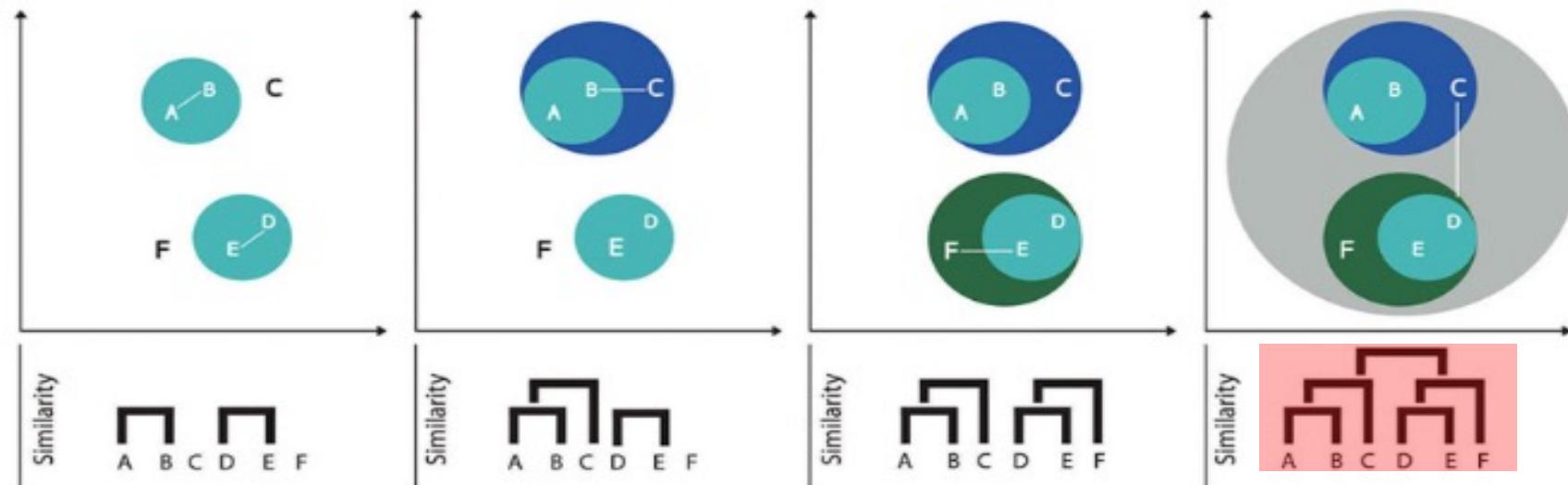
- 계층적 트리 모형을 이용해 유사한 개체끼리 계층적으로 통합하여 군집화를 수행하는 알고리즘
- 주어진 데이터에서 군집간의 거리 계산 후 → 데이터 이진 트리 구성
- H-Cluster 에서는 군집들 간 서로 포함하는 관계를 맺고 있음
- 군집 수를 사전에 정하지 않고 학습 수행



[참고자료] <https://www.digitalvidya.com/blog/the-top-5-clustering-algorithms-data-scientists-should-know/>

Clustering

■ 계층적 군집화 (Hierarchical Clustering)



- Combine A&B based on similarity
- Combine D&E based on similarity
- Combination of A & B is combined with C
- Combination of D & E is combined with F
- Final tree contains all clusters combined into a single cluster

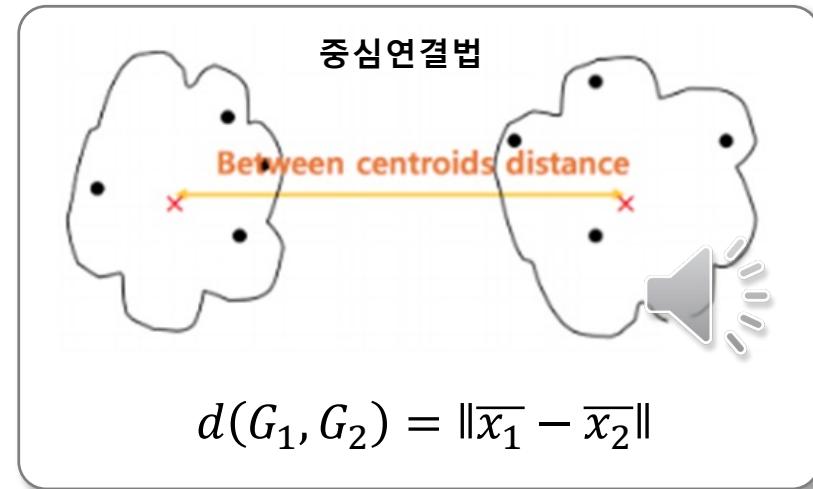
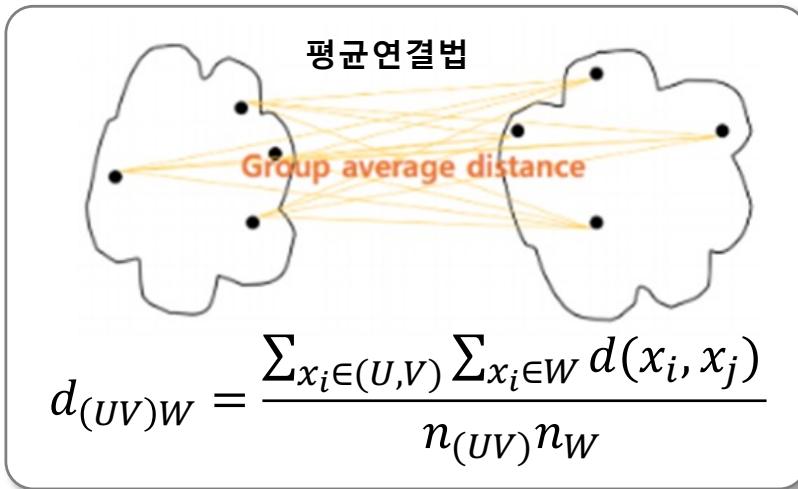
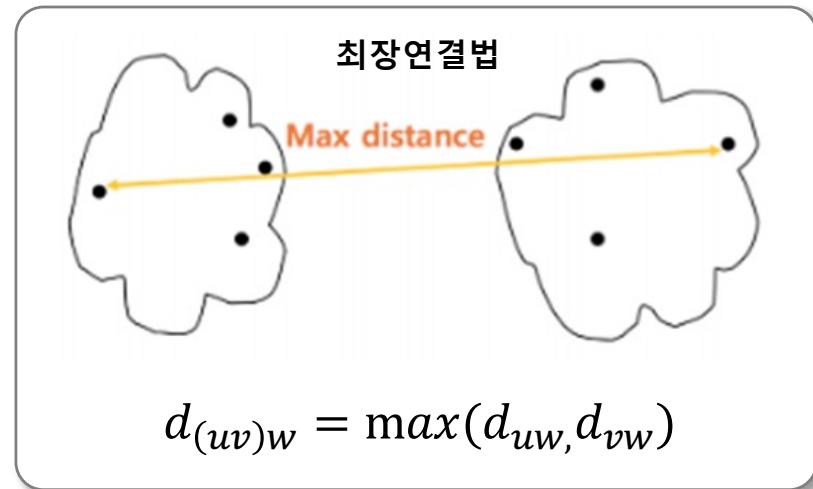
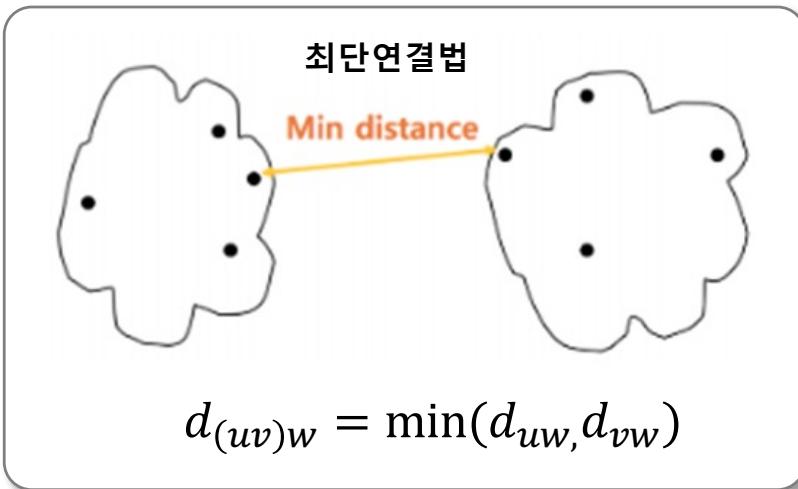
- 가장 가까운 거리에 있는 데이터를 서로 묶음 (반복 수행)
- 최종적으로 하나의 클러스터로 합쳐질 때까지 진행
- Dendrogram 형태의 계층 구조로 정렬됨
- 계층적 군집화 수행 시 모든 개체들 간의 거리 (distance) 측정이나 유사도 (similarity) 측정을 진행한 후 군집화 처리



Clustering

■ 계층적 군집화 (Hierarchical Clustering)

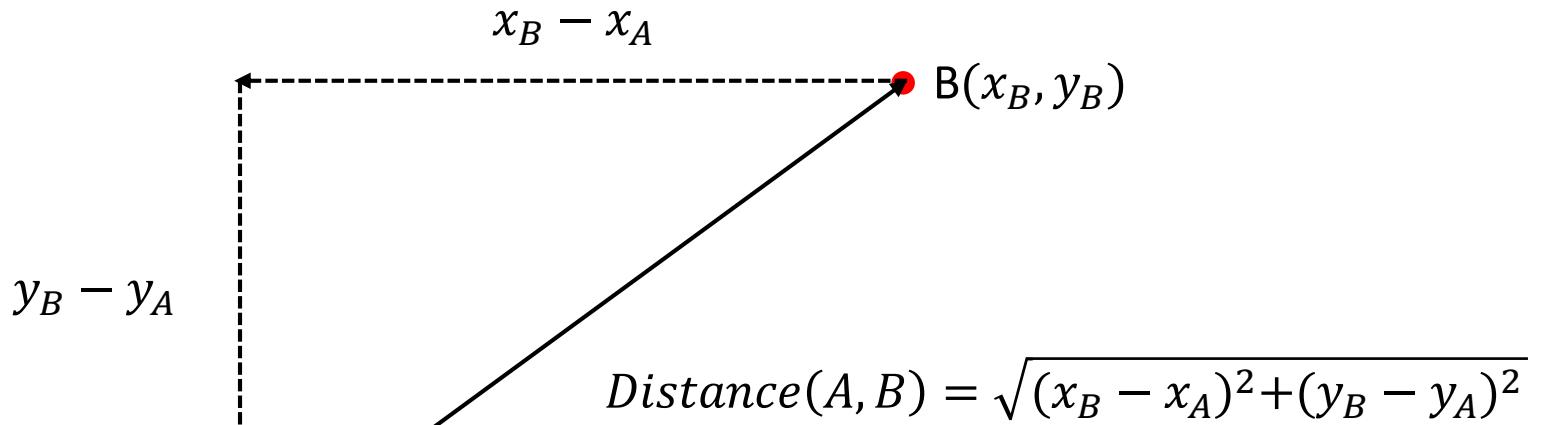
- 두 클러스터가 하나의 클러스터로 합쳐질 때의 거리 (Distance) 측정 방식



Clustering

■ 계층적 군집화 (Hierarchical Clustering)

- Euclidean distance (유클리드 제곱 거리 사용)



Clustering

■ 계층적 군집화 (Hierarchical Clustering)

□ 최단연결법 (Single Linkage)

- 두 개의 군집 간의 가장 가까운 데이터들의 거리로 군집간의 거리를 정의

Data	(x, y)
A	(1, 5)
B	(2, 4)
C	(4, 6)
D	(4, 3)
E	(5, 3)

거리(d)	A	B	C	D	E
A	0				
B	2	0			
C	10	8	0		
D	13	5	9	0	
E	-20	10	10	1	0

가장 가까운 데이터인 D와 E는
하나의 군집으로 처리

$$d = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

최소거리
(closest distance)

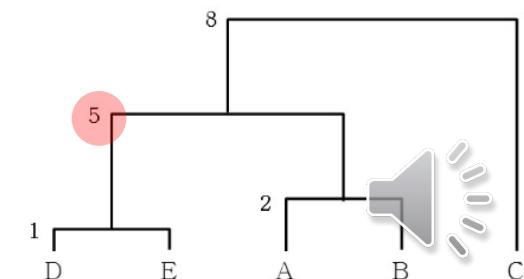
거리(d)	A	B	C	(D, E)
A	0			
B	2	0		
C	10	8	0	
(D, E)	13	5	9	0

가장 가까운 데이터인 A와 B는
하나의 군집으로 처리

최소거리
(closest distance)

거리(d)	(A, B)	C	(D, E)
(A, B)	0		
C	8	0	
(D, E)	5	9	0

(A, B)와 (D, E)의 거리



Clustering

■ 계층적 군집화 (Hierarchical Clustering)

□ 최장연결법 (Complete Linkage)

- 두 개의 군집 간의 가장 먼 데이터들의 거리로 군집간의 거리를 정의

Data	(x, y)
A	(1, 5)
B	(2, 4)
C	(4, 6)
D	(4, 3)
E	(5, 3)

거리	A	B	C	D	E
A	0				
B	2	0			
C	10	8	0		
D	13	5	9	0	
E	20	10	10	1	0

가장 가까운 데이터인 D와 E는
하나의 군집으로 처리

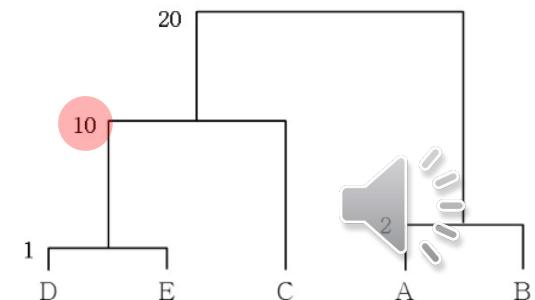
$$d = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

거리	A	B	C	(D, E)
A	0			
B	2	0		
C	10	8	0	
(D, E)	20	10	10	0

가장 가까운 데이터인 A와 B는
하나의 군집으로 처리,
최장거리 값으로 군집의 값을 결정

거리	(A, B)	C	(D, E)
(A, B)	0		
C	10	0	
(D, E)	20	10	0

(A,B)와 (D,E)의 거리,
두 군집에서 가장 먼 데이터인 A와 E의 거리



Clustering

■ 계층적 군집화 (Hierarchical Clustering)

□ 평균연결법 (Average Linkage)

- 두 개의 군집에 속하는 모든 데이터들의 거리의 평균을 군집간의 거리로 정의

Data	(x, y)
A	(1, 5)
B	(2, 4)
C	(4, 6)
D	(4, 3)
E	(5, 3)

거리	A	B	C	D	E
A	0				
B	2	0			
C	10	8	0		
D	13	5	9	0	
E	20	10	10	1	0

가장 가까운 데이터인 D와 E는
하나의 군집으로 처리

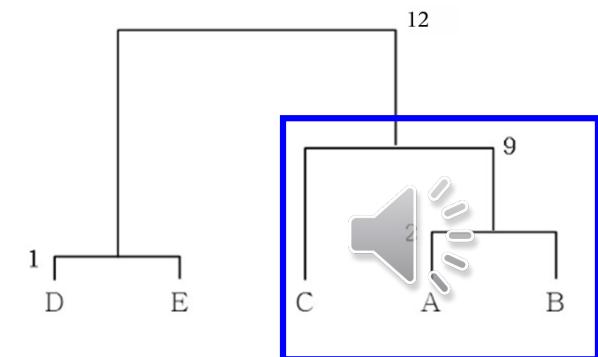
$$d = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

거리	A	B	C	(D, E)
A	0			
B	2	0		
C	10	8	0	
(D, E)	16.5	7.5	9.5	0

가장 가까운 데이터인 A와 B는 하나의
군집으로 처리,
평균값으로 군집의 값을 결정

거리	(A, B)	C	(D, E)
(A, B)	0		
C	9	0	
(D, E)	12	9.5	0

두 군집에서 속하는 데이터의 모든
거리(A-D, A-E, B-D, B-E)의 평균



Clustering

■ 계층적 군집화 (Hierarchical Clustering)

□ 중심연결법 (Centroid Linkage)

- 두 개의 군집에 속하는 모든 데이터들의 중심값을 거리로 정의

Data	(x, y)
A	(1, 5)
B	(2, 4)
C	(4, 6)
D	(4, 3)
E	(5, 3)

가중평균 값으로 계산
D와 E 거리(4.5, 3)

거리	A	B	C	D	E
A	0				
B	2	0			
C	10	8	0		
D	13	5	9	0	
E	20	10	10	1	0

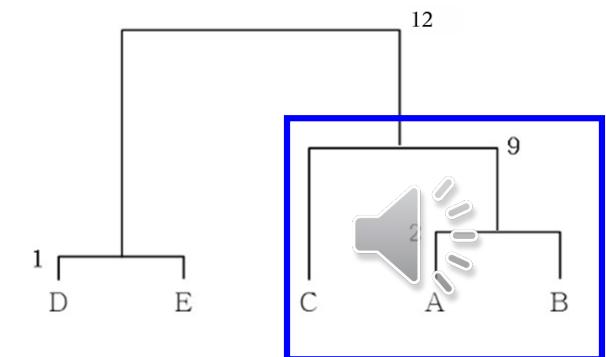
가장 가까운 데이터인 D와 E는
하나의 군집으로 처리

거리	A	B	C	(D, E)
A	0			
B	2	0		
C	10	8	0	
(D, E)	16.5	7.5	9.5	0

가장 가까운 데이터인 A와 B는 하나의
군집으로 처리,
중심값으로 군집의 값을 결정

거리	(A, B)	C	(D, E)
(A, B)	0		
C	9	0	
(D, E)	12	9.5	0

두 군집에서 속하는 데이터의 모든 거리를
중심값으로 표현

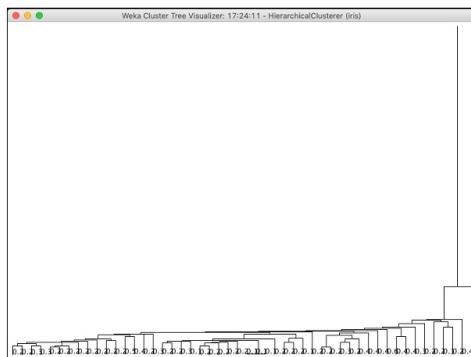
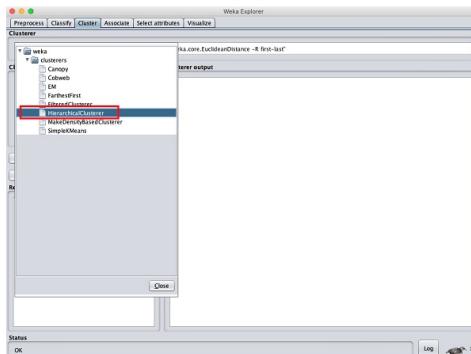


Clustering

■ 계층적 군집화 (Hierarchical Clustering)

- 사용 가능한 도구

Weka



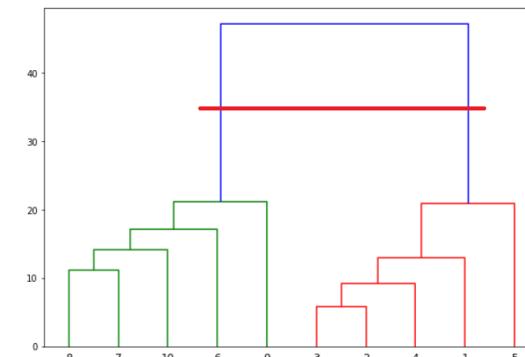
Python

```
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt

linked = linkage(X, 'single')

labelList = range(1, 11)

plt.figure(figsize=(10, 7))
dendrogram(linked,
            orientation='top',
            labels=labelList,
            distance_sort='descending',
            show_leaf_counts=True)
plt.show()
```



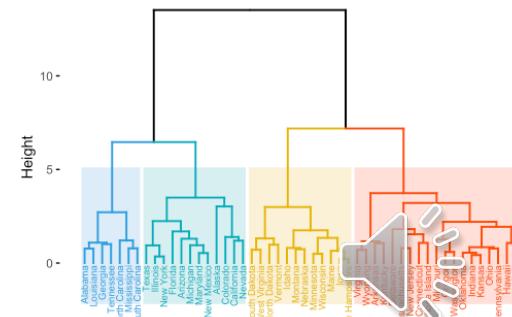
R

```
library(factoextra)
fviz_dend(hc, cex = 0.5)
```

```
fviz_dend(hc, cex = 0.5,
          main = "Dendrogram - ward.D2",
          xlab = "Objects", ylab = "Distance", sub = "")
```

```
fviz_dend(hc, cex = 0.5, horiz = TRUE)
```

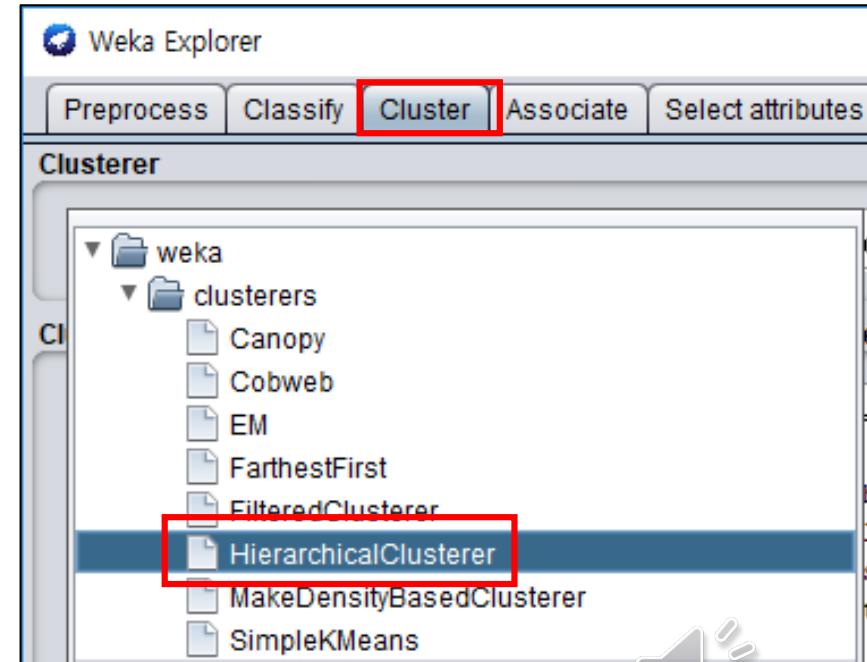
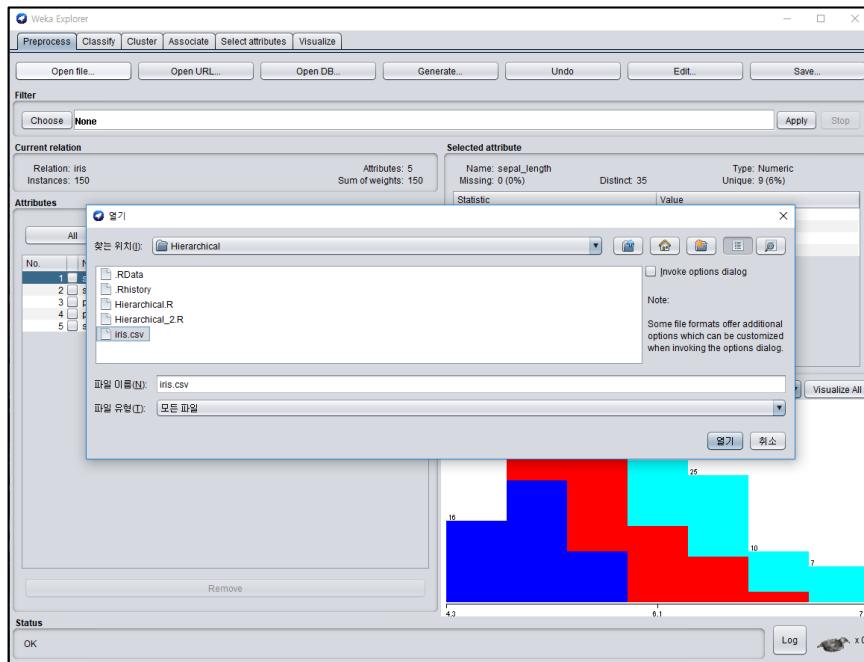
Cluster Dendrogram



Clustering

■ 계층적 군집화 (Hierarchical Clustering)

- Weka 이용한 Hierarchical Clustering 실습
 - Weka에서 iris.csv 파일 열기



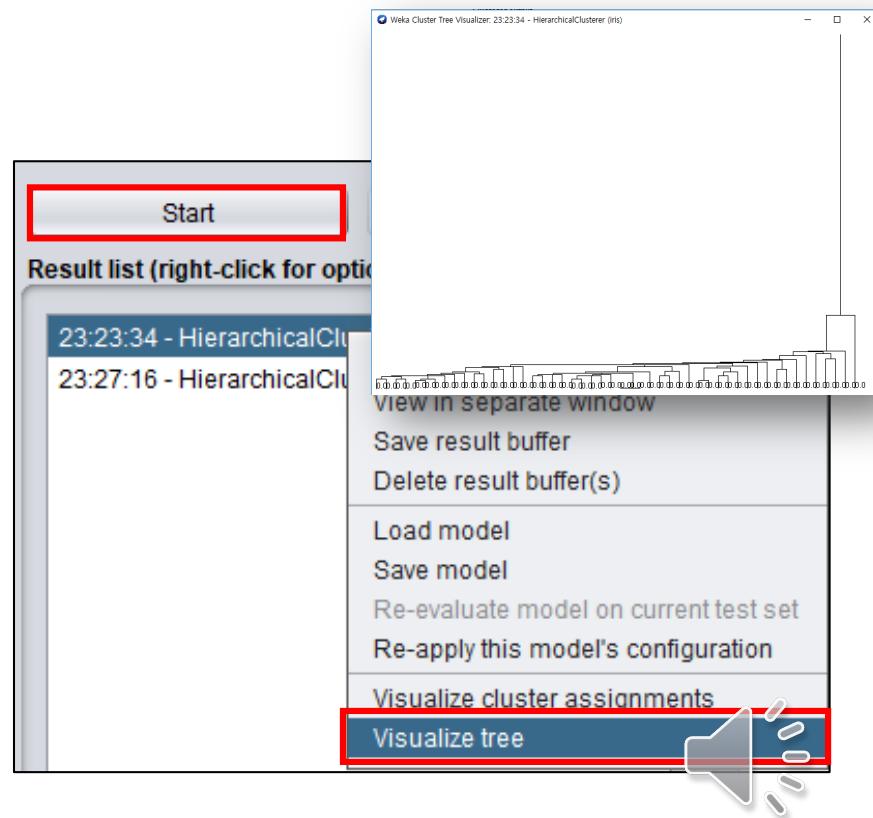
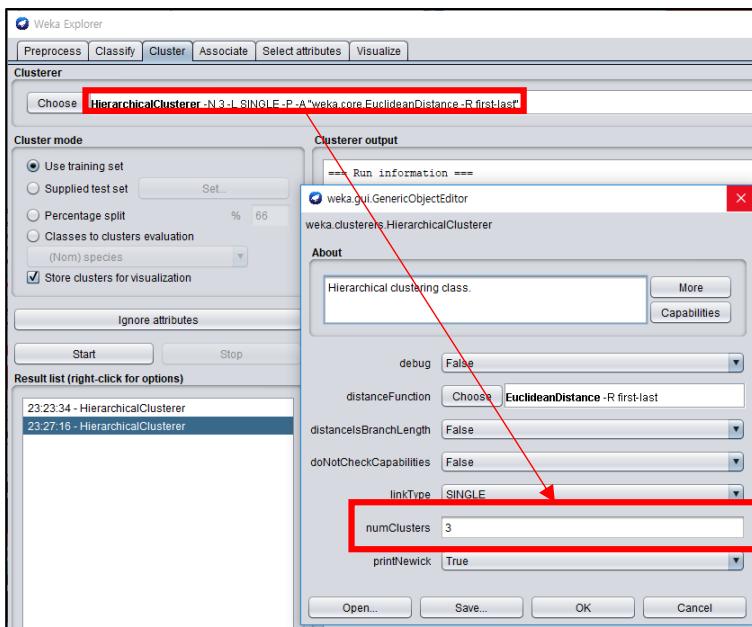
Cluster -> HierarchicalClusterer 선택

Clustering

■ 계층적 군집화 (Hierarchical Clustering)

□ Weka 이용한 Hierarchical Clustering 실습

- numClusters = 3 // 클러스터 개수 (클러스터 개수 설정 탭이 존재하나 실제 적용되지 않음)



Clustering

■ 계층적 군집화 (Hierarchical Clustering)

□ R를 이용한 Hierarchical Clustering 실습

"Wines" Data 불러오기

```
data("wines")
```

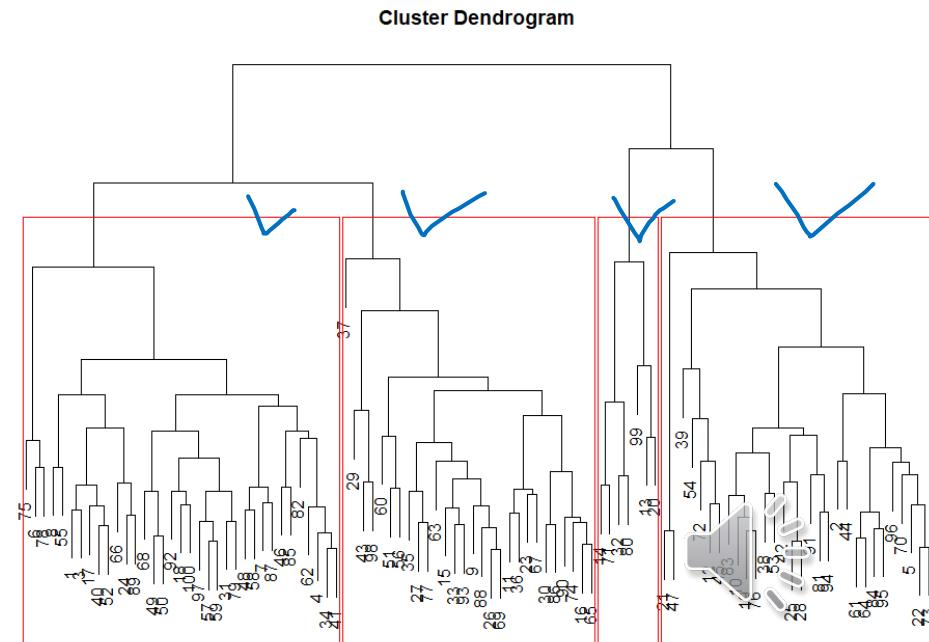
데이터 정규화 (표준편차)

```
wines.sc <- scale(wines)
```

Hierarchical 설정 (Dendrogram)

```
sd <- wines.sc[sample(1:nrow(wines.sc),100),-1]  
d <- dist(sd, method = "euclidean")  
fit <- hclust(d, method="complete")  
plot(fit)  
rect.hclust(fit, k=4, border = "red")
```

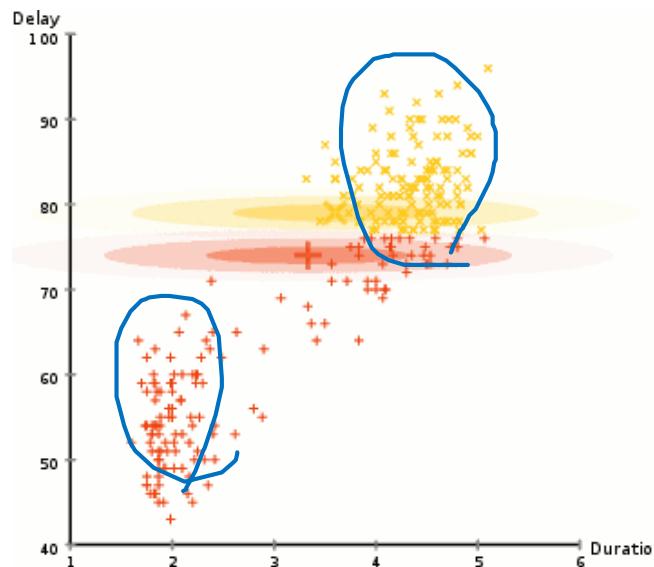
경계의 개수
(학습 시 클러스터의 개수 아님)



Clustering

■ K-평균 군집화 (K-means Clustering)

- 미리 정해 놓은 각 군집의 프로토타입에 개체들이 얼마나 유사한가 (혹은 가까운가)를 측정하여 군집을 형성하는 기법
- K는 군집의 수 (number of clusters)
 - 클래스 (레이블, 정답지) 가 없는 데이터에서는 몇 개의 클러스터가 존재하는지 모름
 - K-평균 군집화 알고리즘에서는 분류할 클러스터의 수를 미리 정함
- K-means Clustering 는 EM 알고리즘을 기반으로 작동



[참고자료] <https://www.digitalvidya.com/blog/the-top-5-clustering-algorithms-data-scientists-should-know/>

Clustering

■ K-평균 군집화 (K-means Clustering)

□ EM알고리즘

- **E**xpectation (기대값)과 **M**aximization (최대화) 으로 나뉘어 수렴할 때까지 반복하는 방식

□ **E**-step 과 **M**-step 을 반복 적용해도 결과가 바뀌지 않거나 (=해가 수렴), 사용자가 정한 반복 수를 채우게 되면 학습이 완성됨

예시) 군집 수 k=2설정

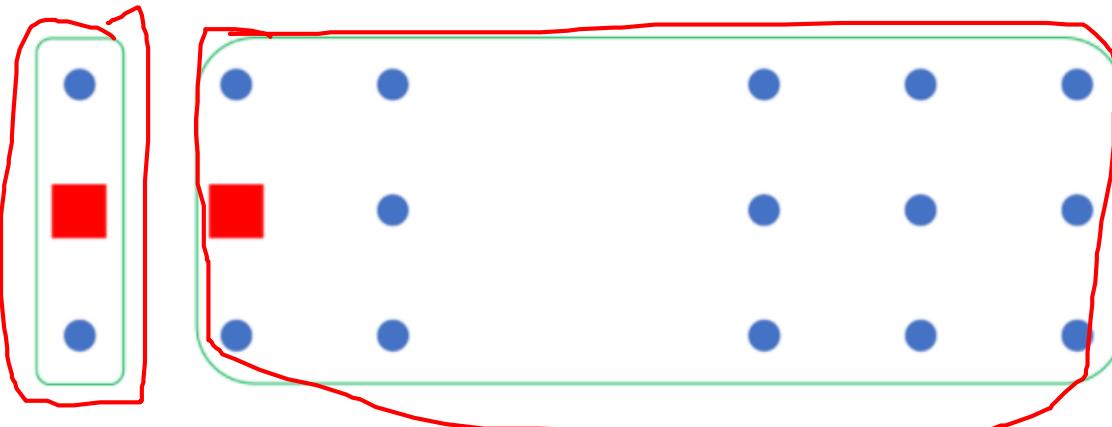
- 군집의 중심 (빨간색 점)을 랜덤 초기화



Clustering

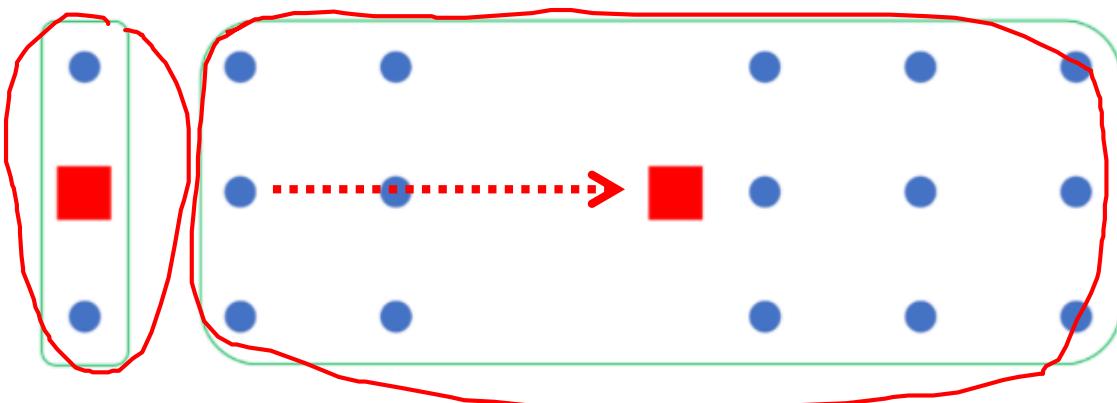
■ K-평균 군집화 (K-means Clustering)

- 모든 개체들을 군집의 중심을 기준으로 가장 가까운 중심에 군집 (녹색 박스)으로 할당



1st Expectation 스텝
(E-step)

- 중심을 군집 경계에 맞게 업데이트

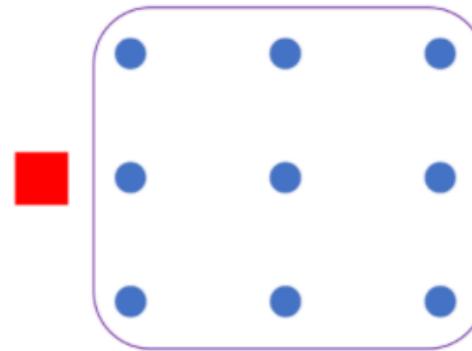
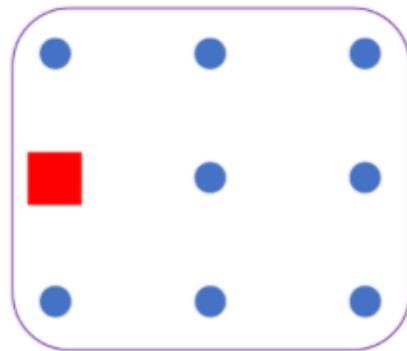


1st Maximization 스텝
(M-step)

Clustering

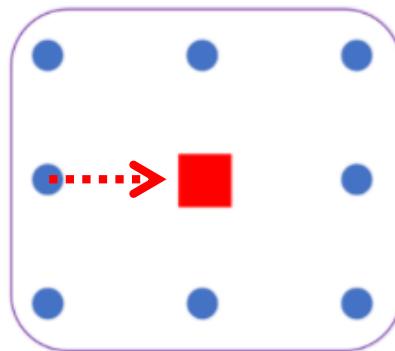
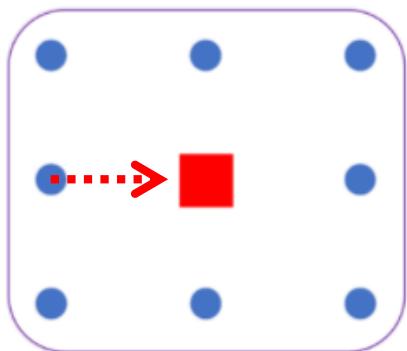
■ K-평균 군집화 (K-means Clustering)

- 모든 개체들을 가장 가까운 중심에 군집 (보라색 박스)으로 할당



2nd Expectation 스텝
(E-step)

- M-step 을 적용해 중심을 군집 경계에 맞게 다시 업데이트



2nd Maximization 스텝
(M-step)



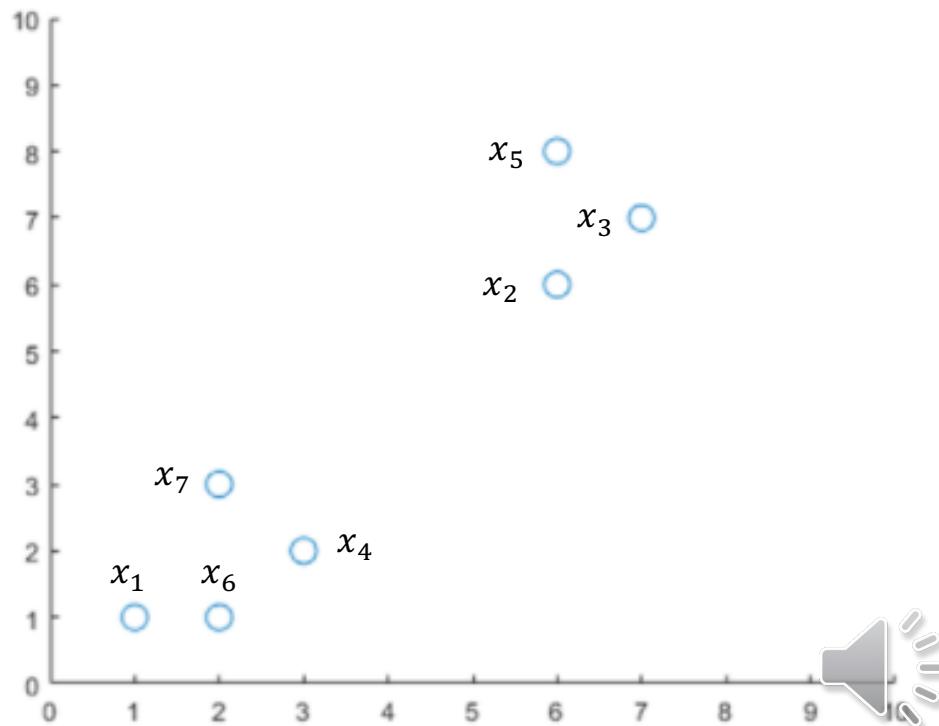
Clustering

■ K-평균 군집화 (K-means Clustering)

예) 2차원 데이터에 대해 k-means clustering, K=2

- 초기 cluster의 중심을 랜덤하게 초기화

Data	x	y
x_1	1	1
x_2	6	6
x_3	7	7
x_4	3	2
x_5	6	8
x_6	2	1
x_7	2	3



Clustering

■ K-평균 군집화 (K-means Clustering)

- 각 데이터 벡터 x_n 과 cluster의 중심 c_k 간의 거리를 계산

$$c_k = \frac{1}{\sum_{n=1}^N r_{nk}} \sum_{n=1}^N r_{nk} x_n$$

$$c1=(4, 3), c2=(7, 9)$$

Data	x	y
x_1	1	1
x_2	6	6
x_3	7	7
x_4	3	2
x_5	6	8
x_6	2	1
x_7	2	3

$$\begin{aligned} S1 \text{ Distance} &= \sqrt{(4 - 3)^2 + (3 - 2)^2} \\ &= 1.4142... \end{aligned}$$

$$\begin{aligned} S2 \text{ Distance} &= \sqrt{(7 - 3)^2 + (9 - 2)^2} \\ &= 8.0623... \end{aligned}$$

Data	s_1	s_2
x_1	3.6056	10
x_2	3.6056	3.1623
x_3	5	2
x_4	1.4142	8.0623
x_5	5.3852	1.4142
x_6	2.8284	9.4340
x_7	2	7.8102

- 데이터를 가장 가까운 cluster에 할당

- S1 군집: x_1, x_4, x_6, x_7
- S2 군집: x_2, x_3, x_5



Clustering

■ K-평균 군집화 (K-means Clustering)

- cluster의 중심을 $c_1=(2, 1.75)$, $c_2=(6.3333, 7)$ 로 간신

k 번째 cluster에 할당된 모든 데이터 벡터의 평균으로 간신

$$c_k = \frac{1}{\sum_{n=1}^N r_{nk}} \sum_{n=1}^N r_{nk} x_n$$

Data	x	y
x_1	1	1
x_2	6	6
x_3	7	7
x_4	3	2
x_5	6	8
x_6	2	1
x_7	2	3

$$\begin{aligned} S1 \text{ Distance} &= \sqrt{(2 - 3)^2 + (1.75 - 2)^2} \\ &= 1.03077... \end{aligned}$$

$$\begin{aligned} S2 \text{ Distance} &= \sqrt{(6.3333 - 3)^2 + (7 - 2)^2} \\ &= 6.00923... \end{aligned}$$

Data	s_1	s_2
x_1	1.25	8.0277
x_2	5.8363	1.0541
x_3	7.25	0.6667
x_4	1.0308	6.0092
x_5	7.4204	1.0541
x_6	0.75	7.4012
x_7	1.25	5.8972

- 데이터를 가장 가까운 cluster에 할당

- S1 군집: x_1, x_4, x_6, x_7
- S2 군집: x_2, x_3, x_5



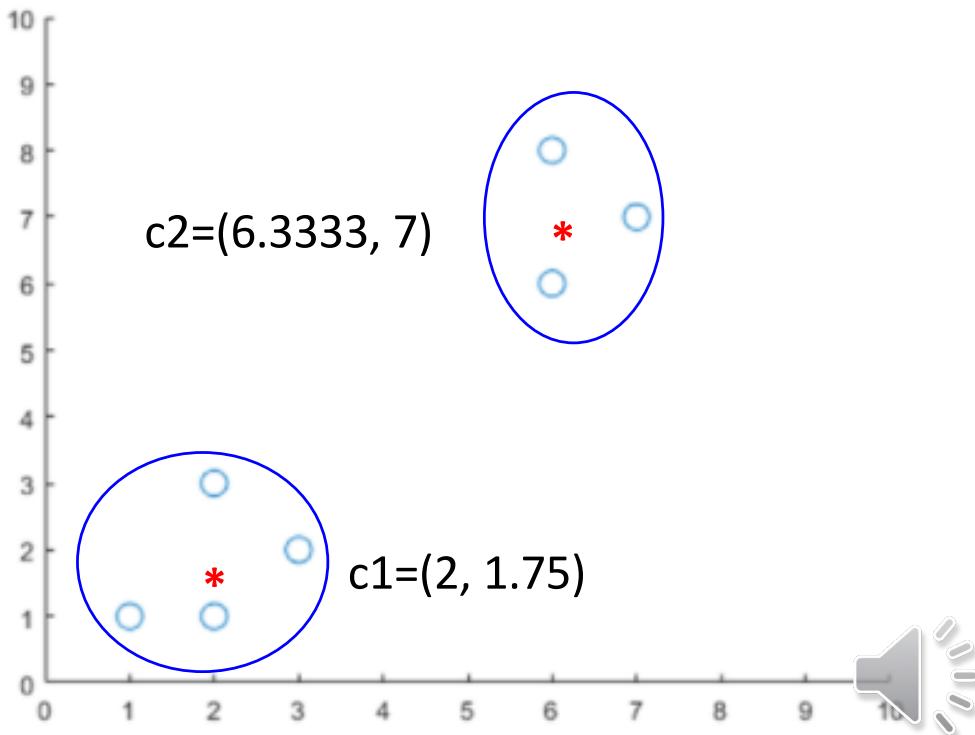
Clustering

■ K-평균 군집화 (K-means Clustering)

- cluster의 중심을 다시 계산할 시 $c1=(2, 1.75)$, $c2=(6.3333, 7)$ 로 변화 없음
-> 알고리즘 종료

k 번째 cluster에 할당된 모든
데이터 벡터의 평균으로 갱신

Data	x	y
x_1	1	1
x_2	6	6
x_3	7	7
x_4	3	2
x_5	6	8
x_6	2	1
x_7	2	3

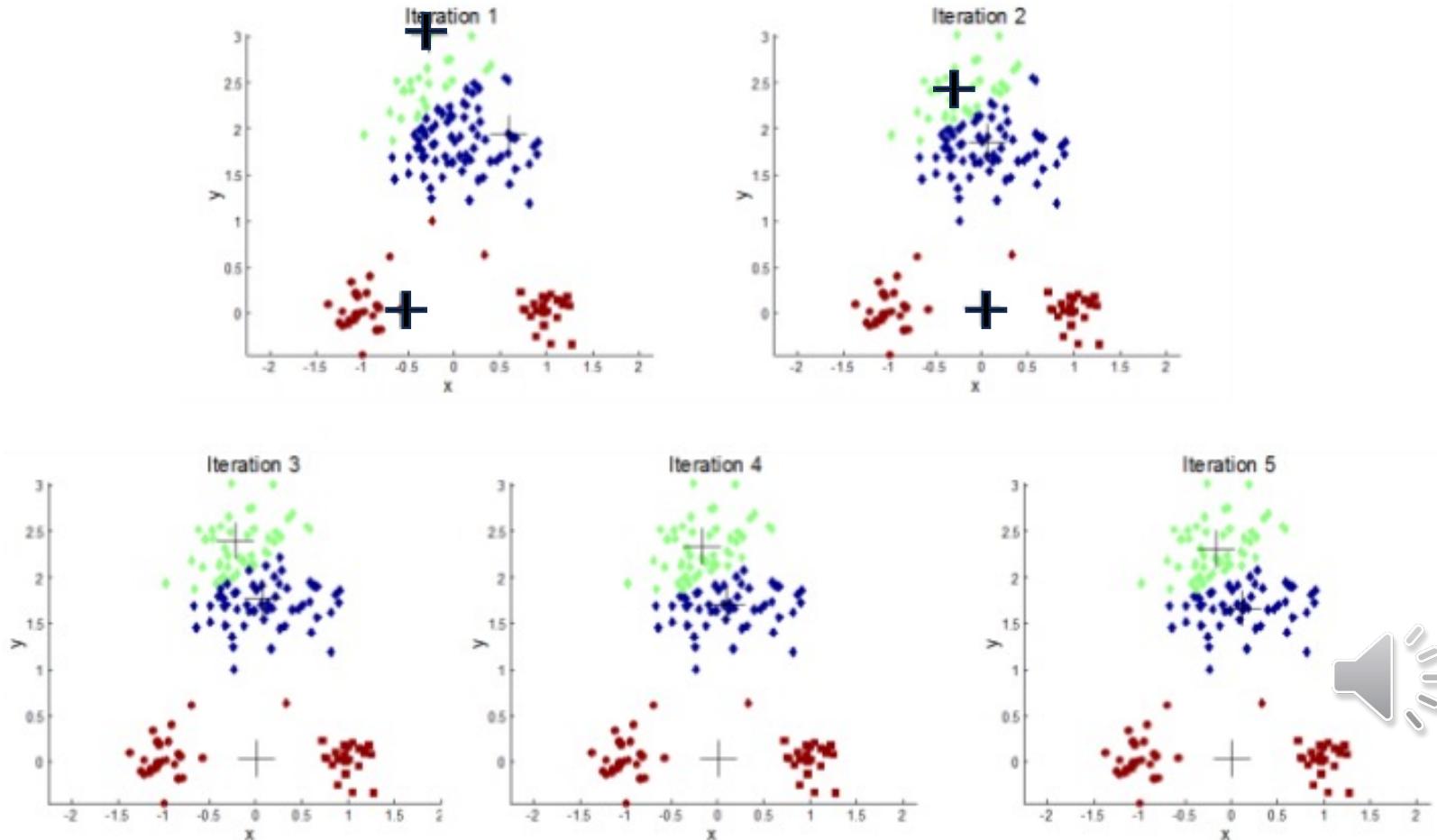


K-means clustering의 결과

Clustering

■ K-평균 군집화 (K-means Clustering)

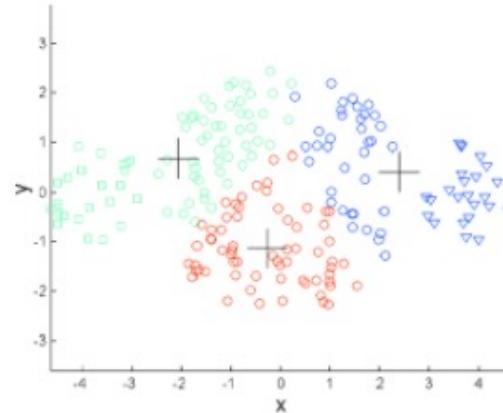
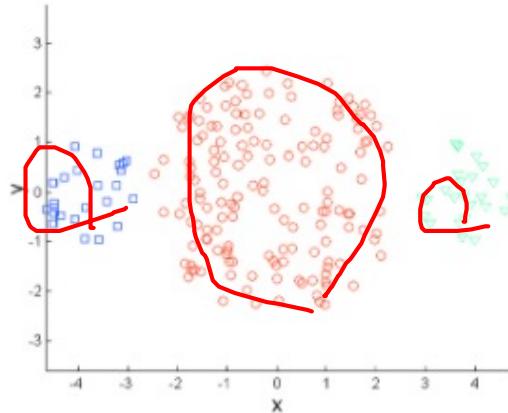
- K-means Clustering는 각 군집 중심의 초기값을 랜덤하게 정하기 때문에, 초기값 위치에 따라 원하는 결과가 나오지 않을 수도 있음



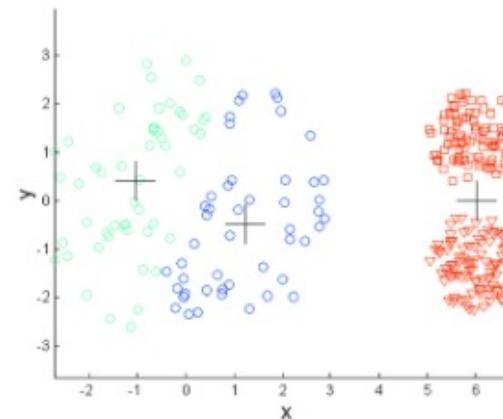
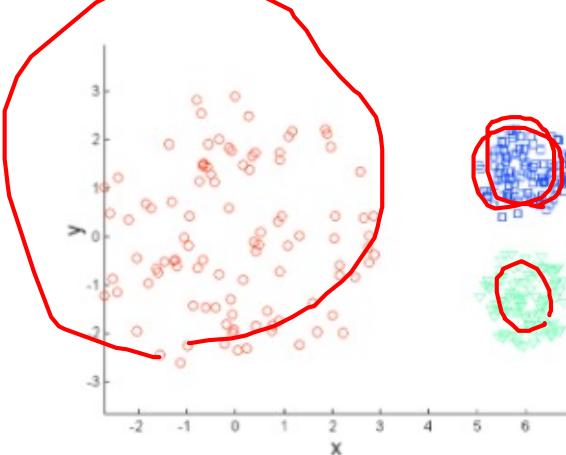
Clustering

■ K-평균 군집화 (K-means Clustering)

- 군집의 크기가 다를 경우, 제대로 작동하지 않을 수 있음



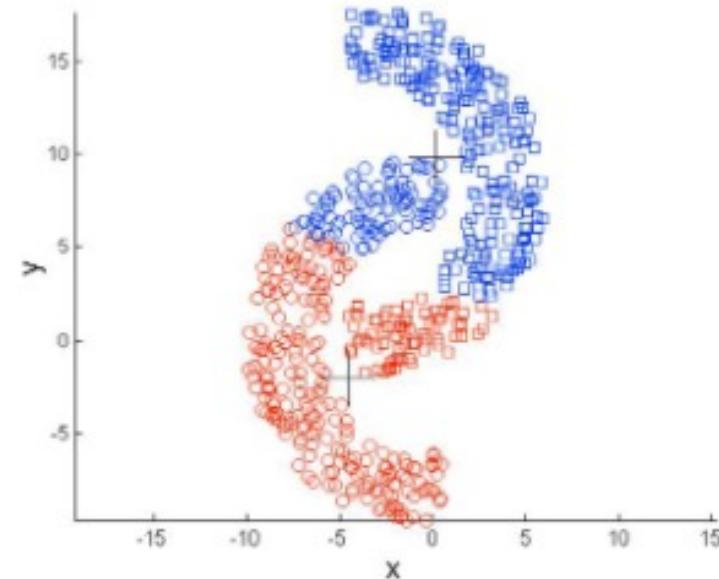
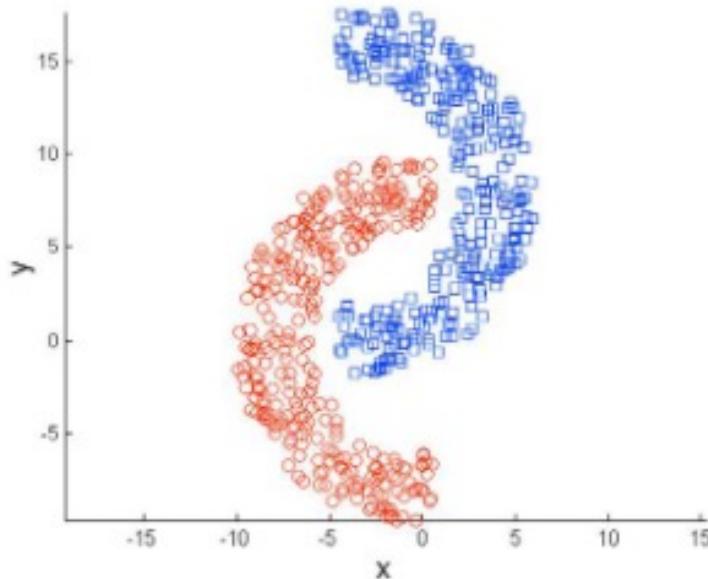
- 군집의 밀도가 다를 경우, 제대로 작동하지 않을 수 있음



Clustering

■ K-평균 군집화 (K-means Clustering)

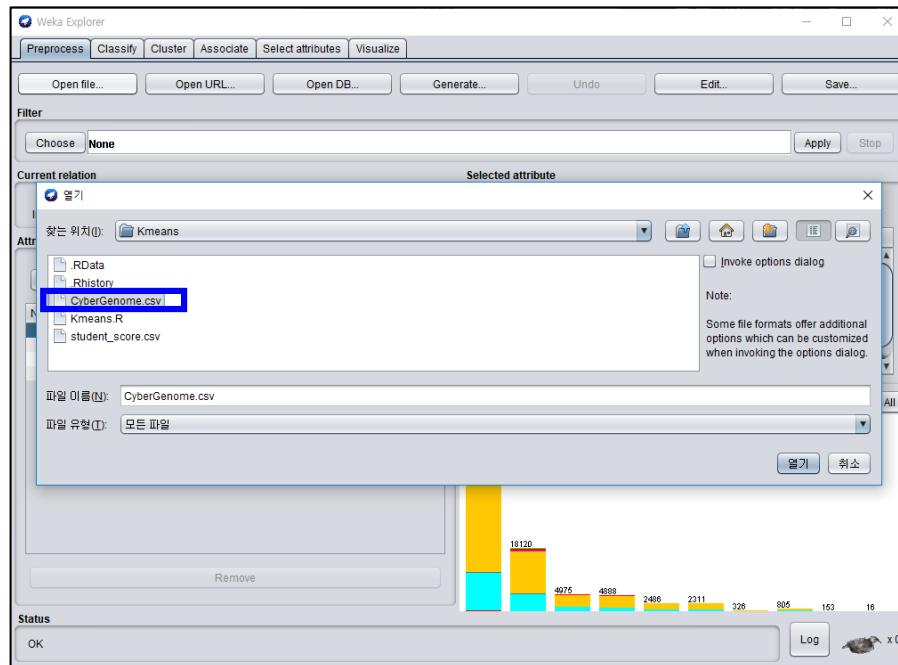
- 데이터 분포가 특이한 케이스일 경우, 제대로 작동하지 않을 수 있음



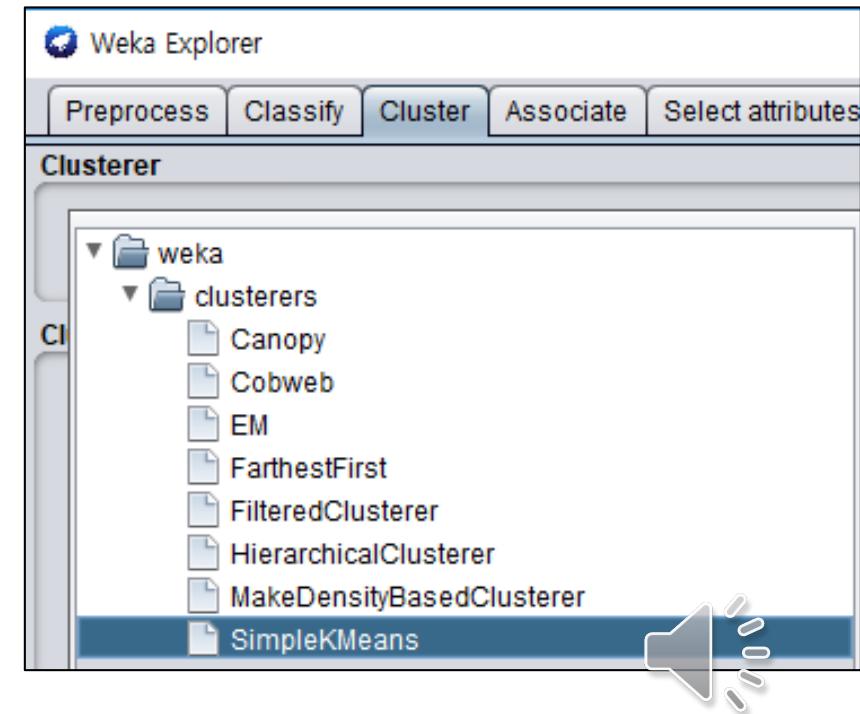
Clustering

■ Weka 이용한 K-means Clustering 실습

Weka 에서 CyberGenome.csv 열기



Cluster -> SimpleKMeans 선택

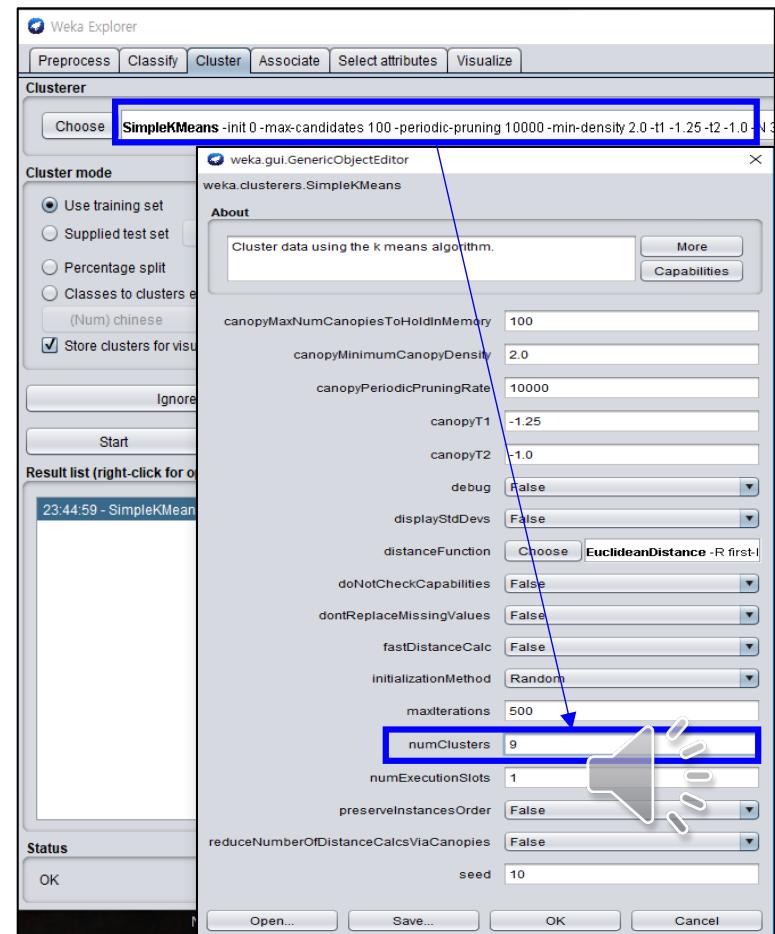


Clustering

■ Weka 이용한 K-means Clustering 실습

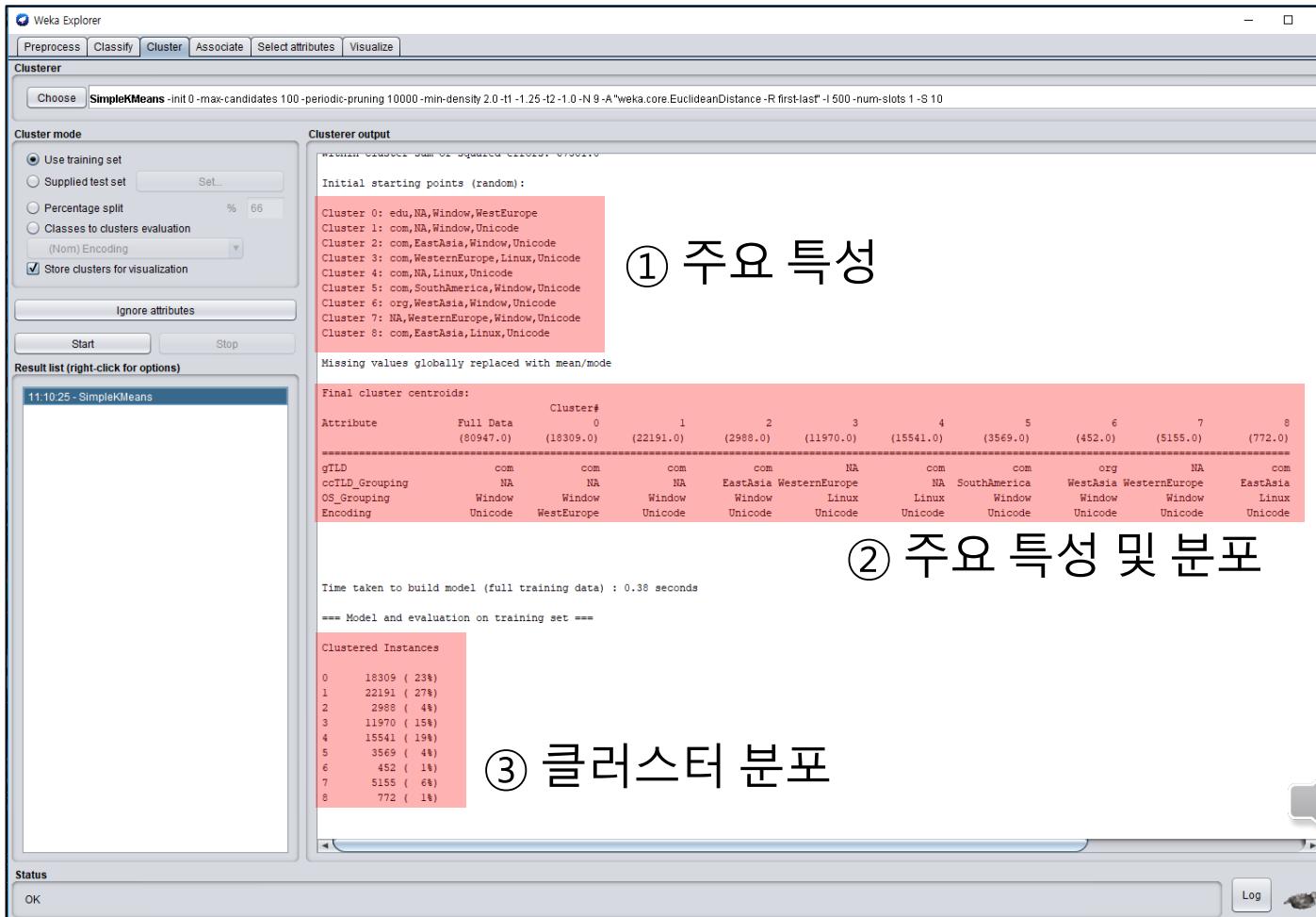
numClusters = ?

- 클러스터 개수 임의로 선택 가능
- 클러스터 개수가 자동으로 설정되도록 하기 위해 EM 알고리즘을 사용



Clustering

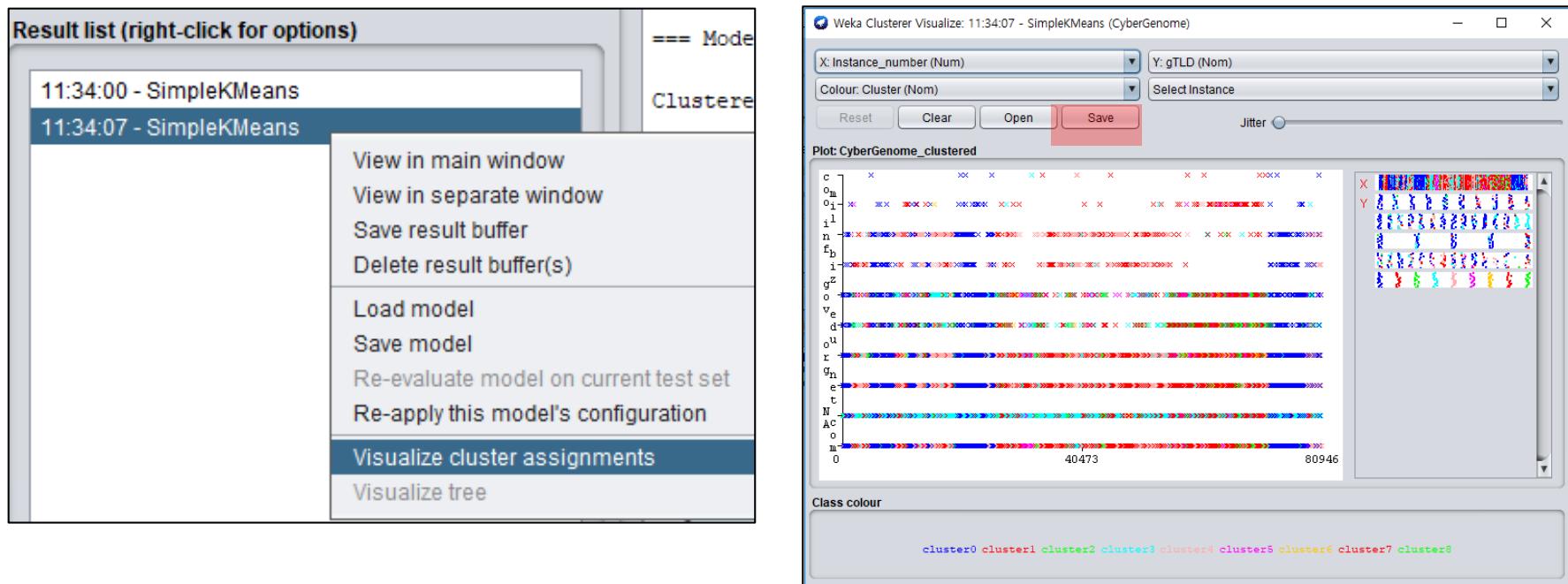
■ Weka 이용한 K-means Clustering 실습



Clustering

■ Weka 이용한 K-means Clustering 실습

SimpleKMeans 실행 후 오른쪽 클릭 -> Visualize cluster assignments 선택



Visualize cluster assignments에서 save -> 파일명.arff 저장



Clustering

■ Weka 이용한 K-means Clustering 실습

CyberGenome.arff 를 메모장/노트패드에서 오픈

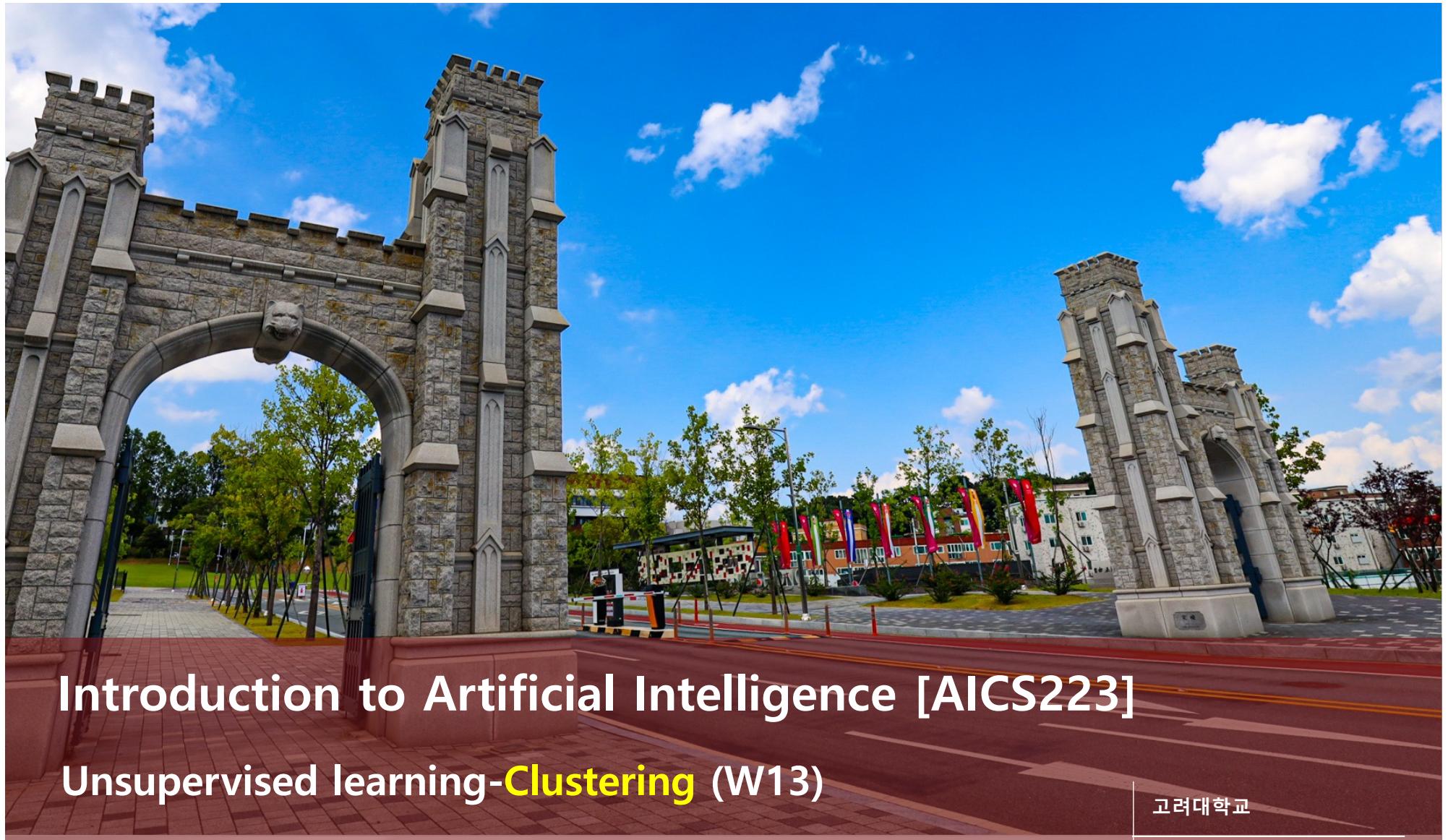
	A	B	C	D
1	gTLD	ccTLD_Grouping	OS_Grouping	Encoding
2	com	NA	Window	Taiwanese
3	com	NA	Linux	Taiwanese
4	NA	EastAsia	Window	Taiwanese
5	NA	SouthAsia	Linux	Chinese
6	com	NA	Window	Chinese
7	net	NA	NA	WestEurope
8	org	NA	Linux	WestEurope
9	com	NA	Linux	WestEurope
10	com	NA	Window	WestEurope
11	net	NA	NA	WestEurope
12	com	NA	Linux	WestEurope
13	com	NA	Linux	WestEurope
14	com	NA	Window	WestEurope
15	NA	Australia	Window	WestEurope
16	com	NA	Unix	WestEurope
17	NA	WesternEurope	Unix	WestEurope
18	edu	NA	Window	WestEurope
19	com	NA	Window	WestEurope
20	NA	Africa	NA	WestEurope
21	gov	NA	Unix	WestEurope
22	edu	Australia	NA	WestEurope
23	edu	WesternEurope	Linux	WestEurope
24	edu	WesternEurope	Linux	WestEurope
25	edu	WesternEurope	Linux	WestEurope
26	edu	WesternEurope	Linux	WestEurope
27	NA	NorthAmerica	Unix	WestEurope
28	edu	NA	Unix	WestEurope
29	NA	NorthAmerica	Window	WestEurope
30	NA	SouthAmerica	Linux	WestEurope



```
@data
0,com,NA,Window,Taiwanese,cluster0
1,com,NA,Linux,Taiwanese,cluster4
2,NA,EastAsia,Window,Taiwanese,cluster2
3,NA,SouthAsia,Linux,Chinese,cluster3
4,com,NA,Window,Chinese,cluster0
5,net,NA,NA,WestEurope,cluster0
6,org,NA,Linux,WestEurope,cluster0
7,com,NA,Linux,WestEurope,cluster0
8,com,NA,Window,WestEurope,cluster0
9,net,NA,NA,WestEurope,cluster0
10,com,NA,Linux,WestEurope,cluster0
11,com,NA,Linux,WestEurope,cluster0
12,com,NA,Window,WestEurope,cluster0
13,NA,Australia,Window,WestEurope,cluster0
14,com,NA,Unix,WestEurope,cluster0
15,NA,WesternEurope,Unix,WestEurope,cluster3
16,edu,NA,Window,WestEurope,cluster0
17,com,NA,Window,WestEurope,cluster0
18,NA,Africa,NA,WestEurope,cluster0
19,gov,NA,Unix,WestEurope,cluster0
20,edu,Australia,NA,WestEurope,cluster0
21,edu,WesternEurope,Linux,WestEurope,cluster3
22,edu,WesternEurope,Linux,WestEurope,cluster3
23,edu,WesternEurope,Linux,WestEurope,cluster3
24,edu,WesternEurope,Linux,WestEurope,cluster3
25,NA,NorthAmerica,Unix,WestEurope,cluster0
26,edu,NA,Unix,WestEurope,cluster0
27,NA,NorthAmerica,Window,WestEurope,cluster0
28,NA,SouthAmerica,Linux,WestEurope,cluster3
29,NA,SouthernEurope,Linux,WestEurope,cluster3
30,NA,SouthernEurope,Window,WestEurope,cluster0
```

Thank you





Introduction to Artificial Intelligence [AICS223]

Unsupervised learning-**Clustering** (W13)

Prof. Mee Lan Han (aeternus1203@gmail.com)

고려대학교

인공지능사이버보안학과



KOREA
UNIVERSITY

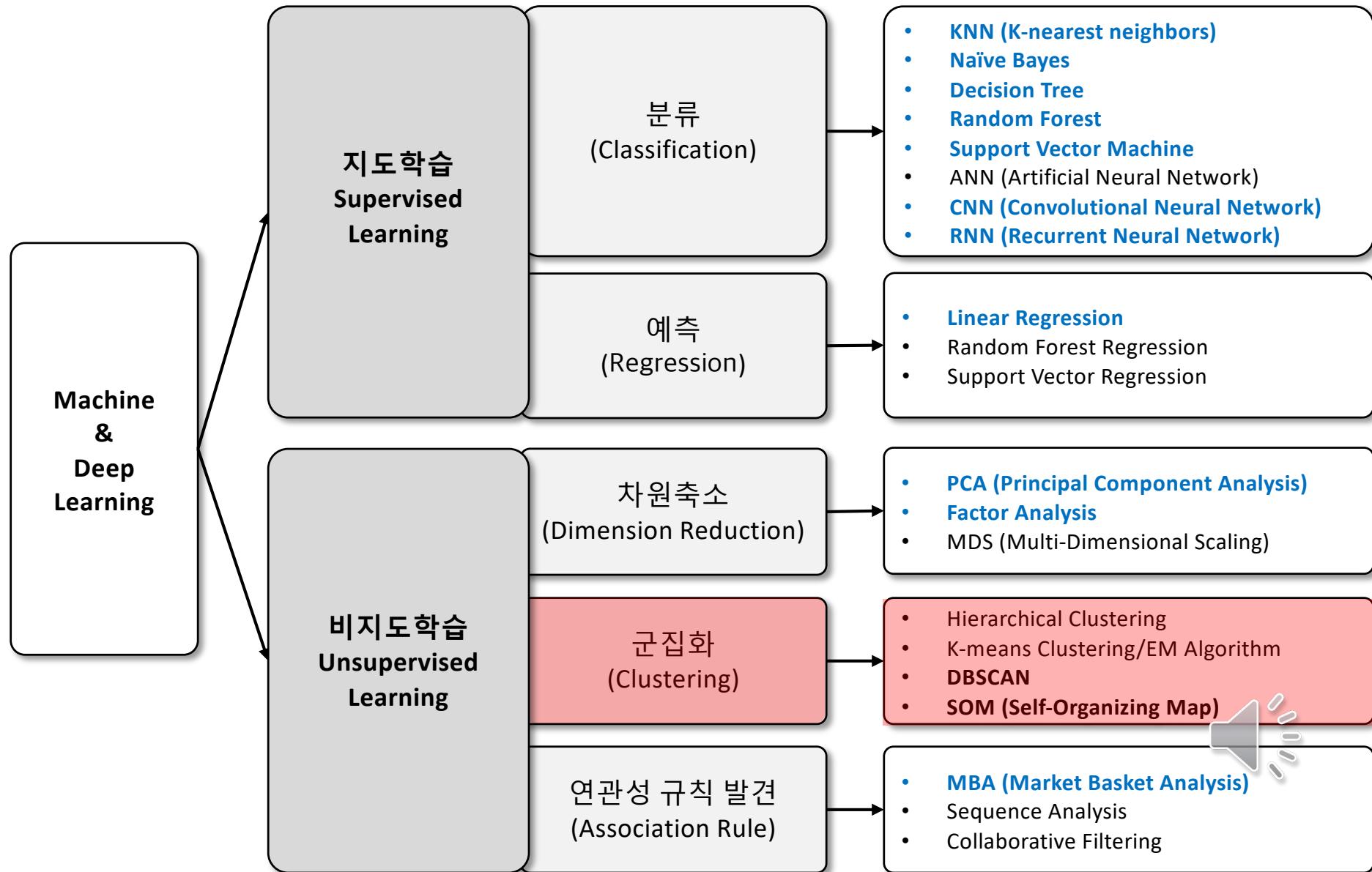
CONTENTS

■ 군집화 (Clustering)

- Hierarchical Clustering
- K-means Clustering/EM algorithm
- DBSCAN
- SOM (Self-Organizing Map)



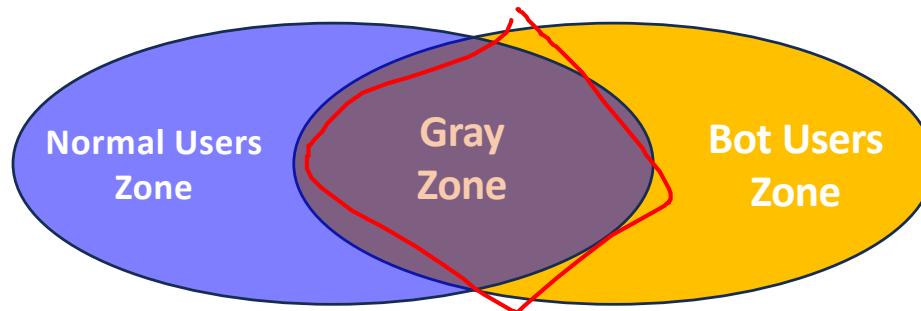
Machine Learning/Deep Learning



Clustering

■ 군집화 (Clustering) 개념 및 특징

- 클러스터링은 일반적으로 정답이 없는 비지도학습 (unsupervised learning)
 - Gray-zone 데이터 셋에 대한 클러스터링이 필요한 경우, 정답이 있는 데이터셋 활용



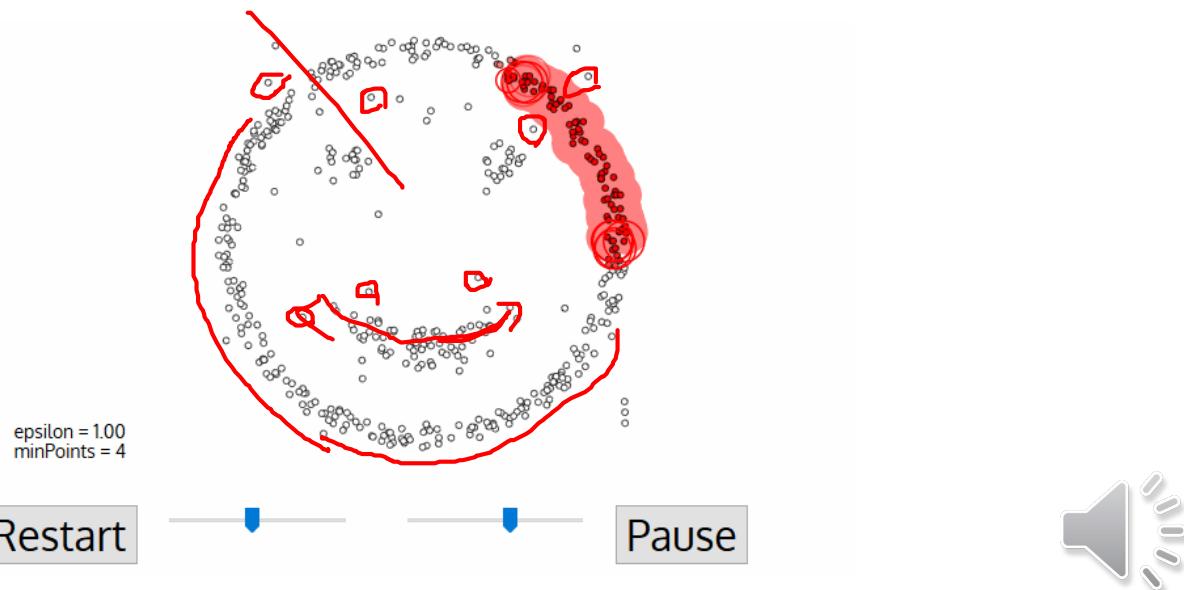
- 비슷한 개체끼리 한 그룹으로, 다른 개체는 다른 그룹으로 묶어 표현
- 클러스터링 수행해야하는 데이터는 정답지가 없음
 - 따라서, 다른 머신러닝 알고리즘처럼 정확도 지표 Accuracy, Recall, Precision, F-measure로 평가할 수 없음
(평가지표는 Evaluation metric에서 설명)



Clustering

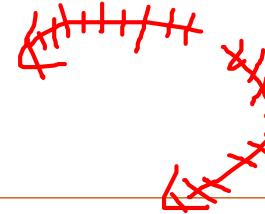
■ DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

- ▣ 기존 클러스터링은 거리 기반의 알고리즘
- ▣ DBSCAN은 밀도 기반의 데이터 클러스터링 알고리즘
- ▣ 밀집되어 있는 지역을 하나의 군집으로 정의
- ▣ 그 지역이 다른 밀집되어 있는 지역과 연결되어 있다면 군집을 확장하면서 군집분석을 진행



[참고자료] <https://www.digitalvidya.com/blog/the-top-5-clustering-algorithms-data-scientists-should-know/>

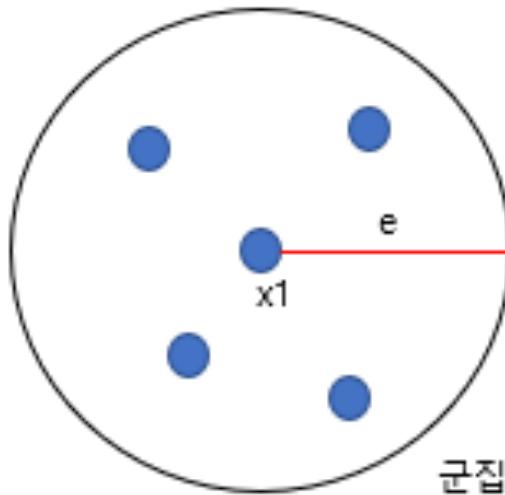
Clustering



■ DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

□ DBSCAN의 핵심: 밀도가 높은 지역에 대한 정의 (군집 확장을 어떻게...?)

- 밀도가 높은 지역을 정의하기 위해서 두 가지의 개념 (Parameter) 이 사용
 - 1 지정거리 (e , eps, epsilon) : 지정된 데이터 포인트를 기준으로 하여 군집을 탐색할 거리를 의미함
 - 2 데이터 개수 (n, MinPts, minimum points) : 지정거리 이내 필요한 최소 데이터 개수를 의미



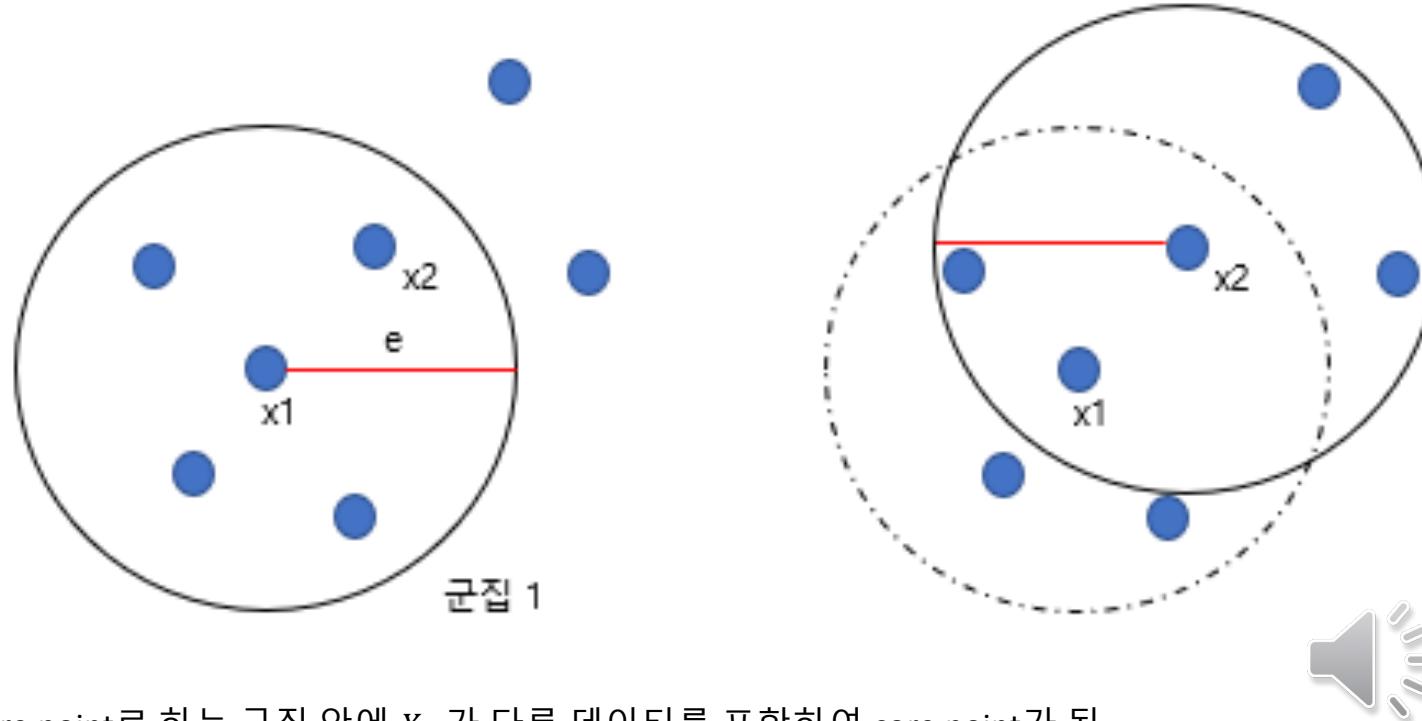
- x_1 데이터 기준으로 $e = 2, n = 5$
- 밀도가 높은 지역으로 선정되며 하나의 군집으로 할당
- x_1 은 core point로 정의됨



Clustering

■ DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

- X_1 데이터 기준으로 $e = 2, n = 5$
- X_1 은 이미 core point로 정의됨
- X_1 을 core point로 갖는 밀도 높은 지역(군집 1) 안에 x_2 데이터는 현재 core point의 요구를 만족함
→ x_2 를 기점으로 e 거리 안에 5개의 샘플 이상이 존재함

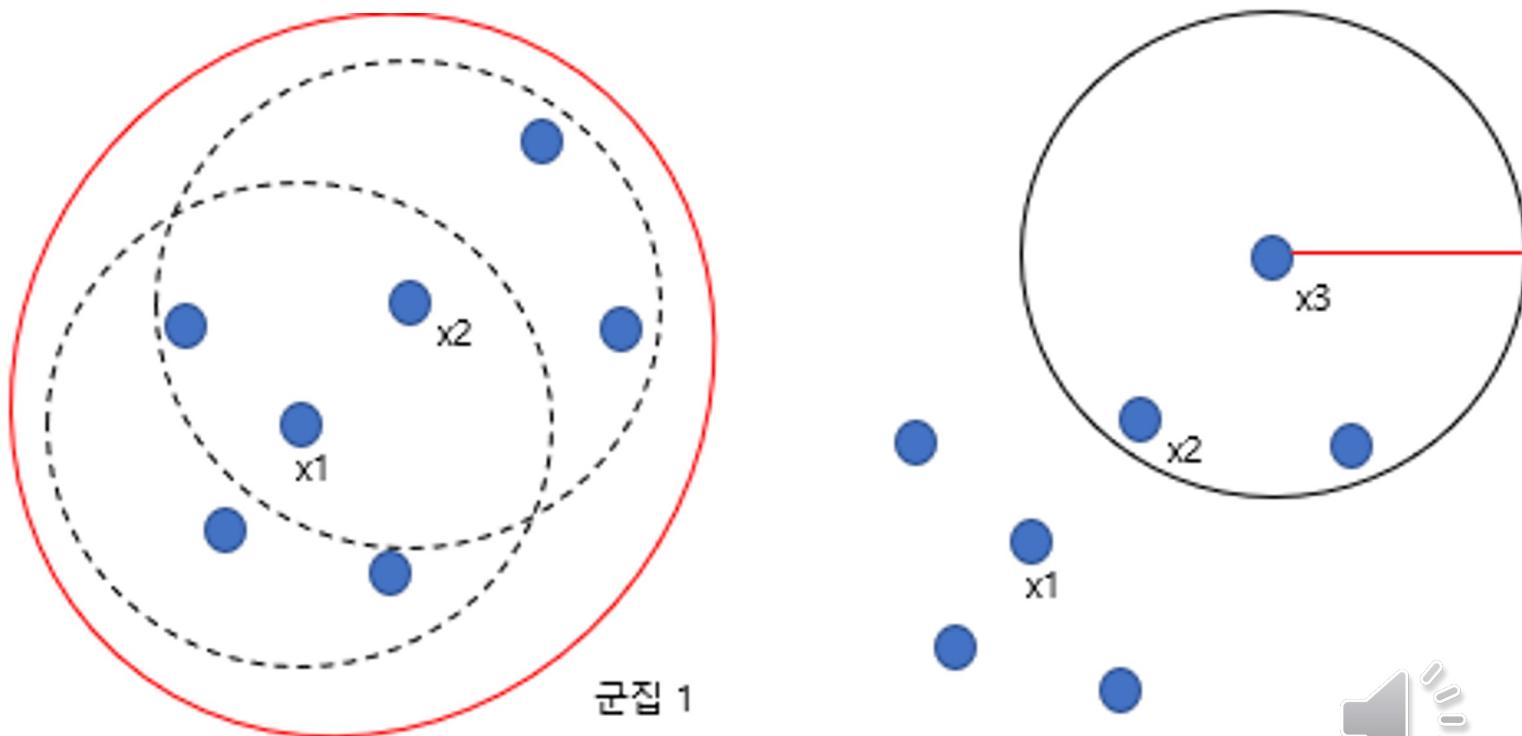


- X_1 core point로 하는 군집 안에 X_2 가 다른 데이터를 포함하여 core point가 됨
- 밀도가 높은 지역이 두 개가 형성되며, 이 경우 밀도가 높은 두 개의 지역이 서로 연결됨
- 처음에 할당했던 군집 1을 확장하여 밀도 높은 지역 두개를 포함하는 방식

Clustering

■ DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

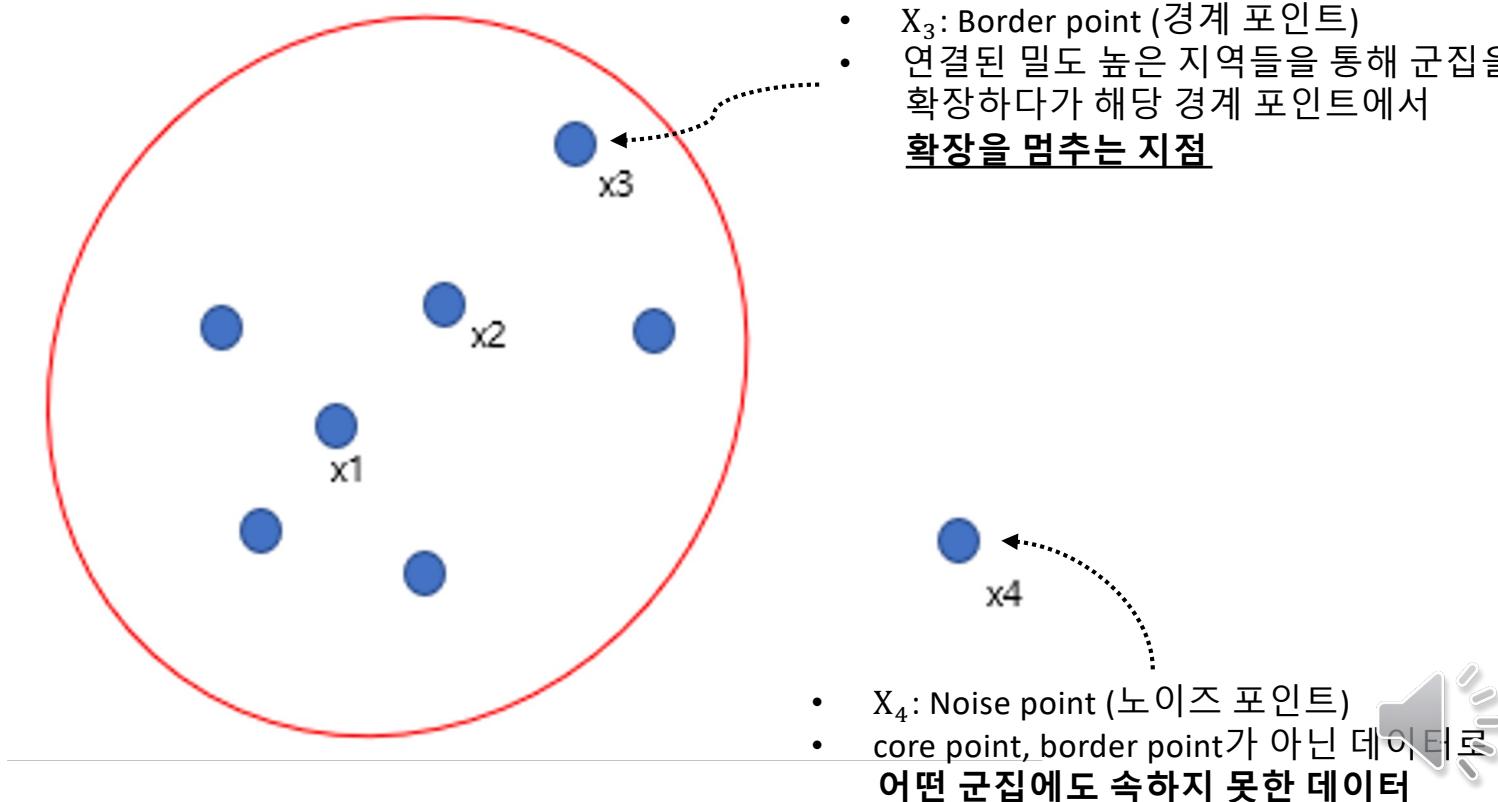
- 밀도가 높은 지역이 연결된 경우 군집을 확장하면서 군집화를 하는 기법
- 군집을 확장하면 어느 순간 확장을 멈추게 되는 지점이 존재함!!!
- 더 이상 연결된 밀도 높은 지역이 없는 상황 발생 → **core point가 없는 상황**



- x_3 를 기점으로 ϵ 거리 안에 5개의 샘플 이상이 존재하지 않음
(밀도가 높은 지역을 형성할 수 없음)
- $\epsilon = 2$, $n = 5$ 만족하지 않음

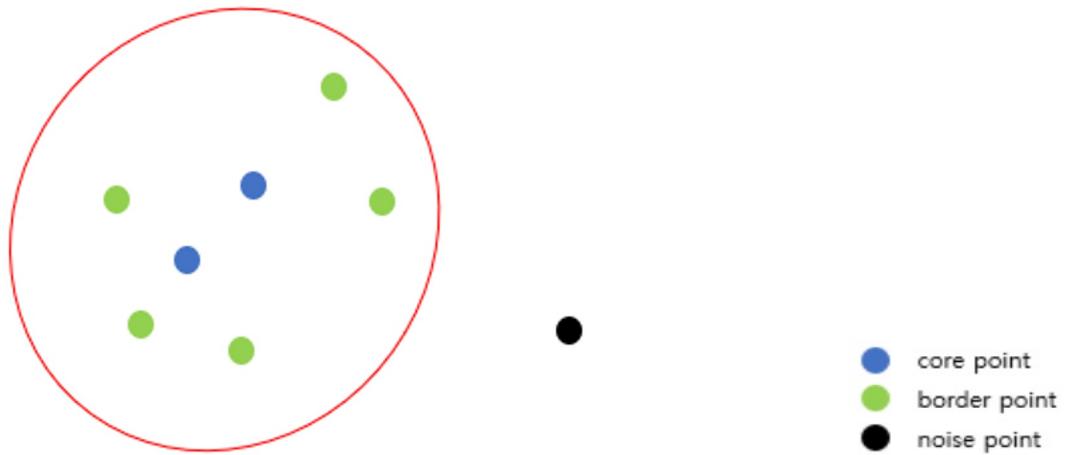
Clustering

■DBSCAN (Density-Based Spatial Clustering of Applications with Noise)



Clustering

■DBSCAN (Density-Based Spatial Clustering of Applications with Noise)



- ① 거리 척도, e (지정 거리), n (필요 최소 샘플 수) 설정을 통해 밀도 높은 지역 정의
- ② 밀도 높은 지역을 만족하는 core point를 찾고 그 지역을 군집으로 할당
- ③ 해당 밀도 높은 지역 안에 core point를 만족하는 데이터가 있다면 그 지역을 포함하여 군집 확장
- ④ 해당 밀도 높은 지역 안에 더 이상 core point를 정의할 수 없을 때까지 2~3 단계 반복
- ⑤ 어떤 군집에도 해당되지 않은 데이터 noise point로 정의



Clustering

■ DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

□ 장점

- 클러스터의 개수를 미리 지정할 필요가 없음
- **복잡한 형상의 데이터셋**에서도 무리 없이 적용 가능
- 잡음 (Noise) 지점도 걸러내기 때문에 **이상치를 구별하기에 유용함**
- 저밀도 클러스터에서 고밀도 클러스터를 분리하는데 유용

□ 단점

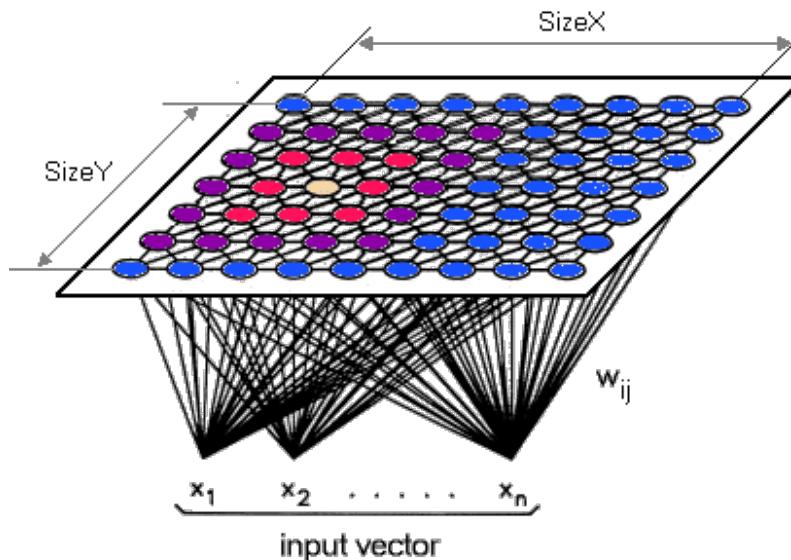
- 부분적으로 비슷한 밀도를 가진 데이터셋의 경우 좋은 성능을 발휘하지 못함
- 데이터 포인트 처리 순서가 매번 다르기 때문에 해당 알고리즘을 시행할 때마다 다른 결과가 도출됨
- 데이터의 차원이 높아질수록 **eps매개변수**의 값을 지정하기 어려움
 - 다른 유클리디언 거리를 사용하는 군집분석처럼 **차원의 저주 문제** 발생 가능



Clustering

■ 자기조직화지도 (Self-Organizing Map, SOM)

- 차원축소 (Dimensionality reduction)와 군집화 (clustering)를 동시에 수행하는 기법
- Map의 노드(뉴런)에 있는 가중치를 변화(경쟁) 시켜가며 클러스터링 (학습) 하는 방법
 - 주어진 데이터를 사용자가 설정한 크기의 Map(1D, 2D 또는 3D)에 할당
 - 고차원의 raw 데이터를 감싸는 저차원 map 구현
- 저차원 (2차원 내지 3차원) 격자에 고차원 데이터의 각 개체들이 대응함
- 인공신경망과 유사한 방식의 학습을 통해 군집을 도출해내는 기법

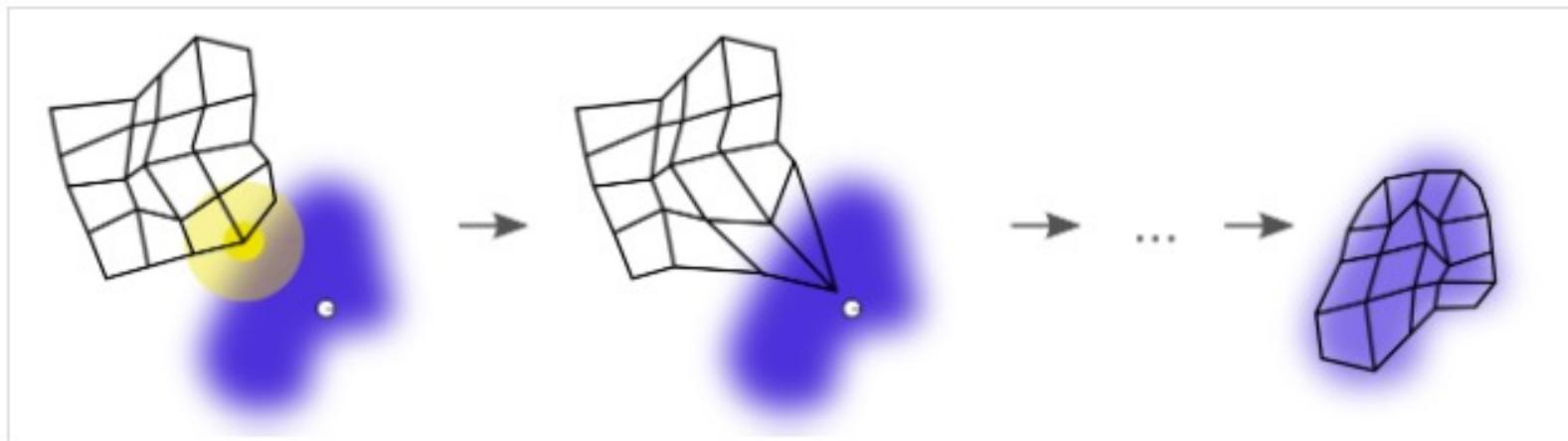


Clustering

■ 자기조직화지도 (Self-Organizing Map, SOM)

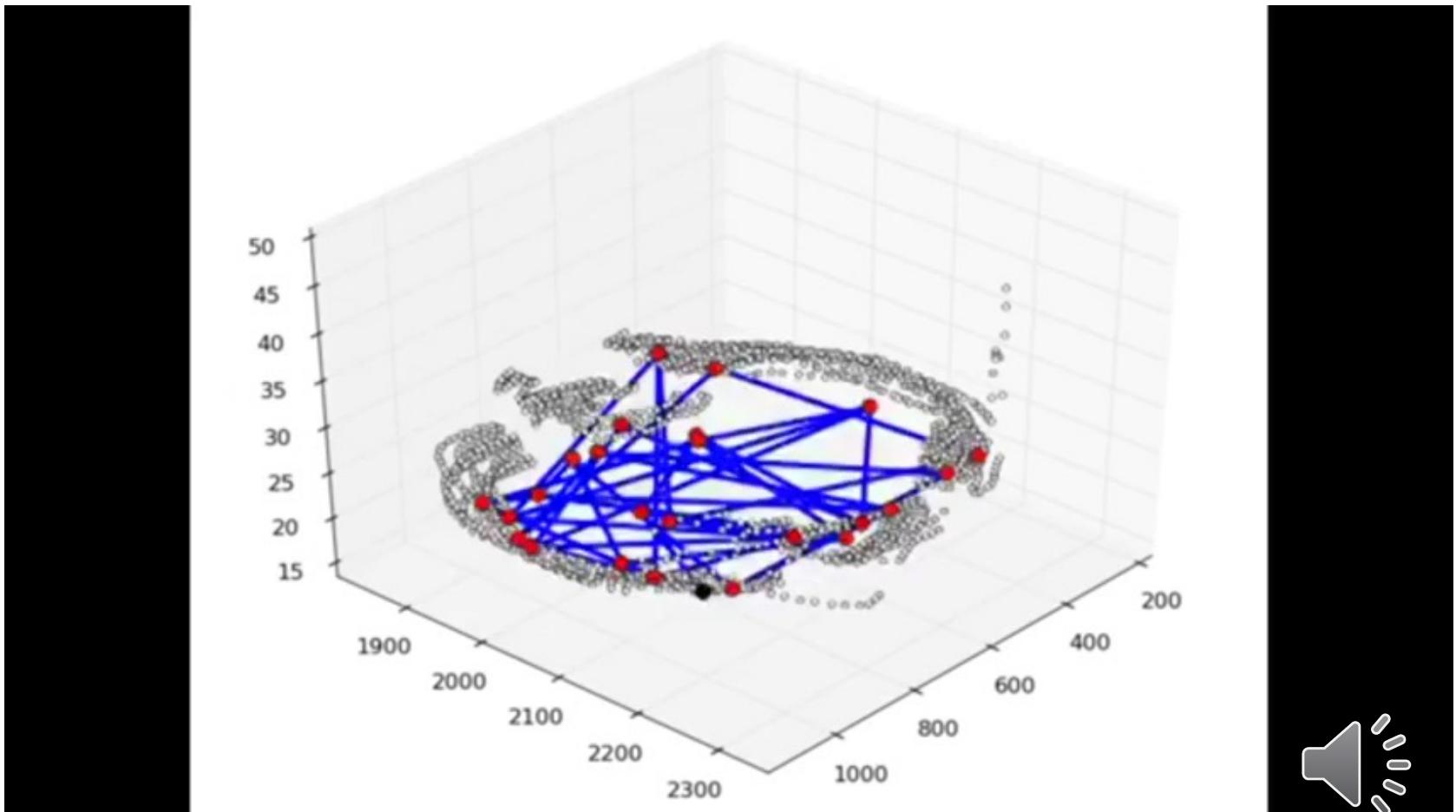
- 2차원 공간은 주로 격자 (grid)로 표현
- 파란색은 고차원 데이터 공간에서의 밀도
- 하얀색 점은 현재의 학습데이터의 한 포인트

격자가 데이터 공간을 학습



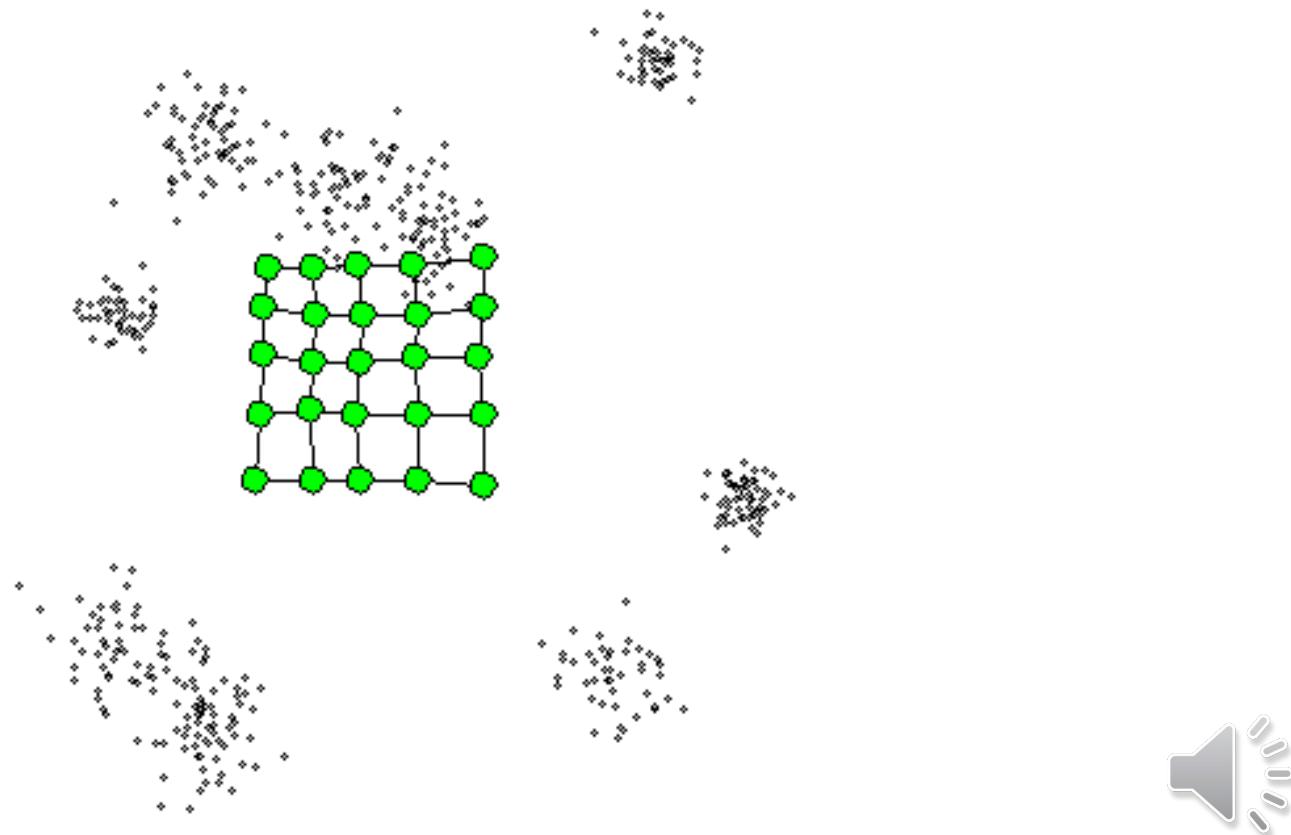
Clustering

■ 자기조직화지도 (Self-Organizing Map, SOM)



Clustering

■ 자기조직화지도 (Self-Organizing Map, SOM)

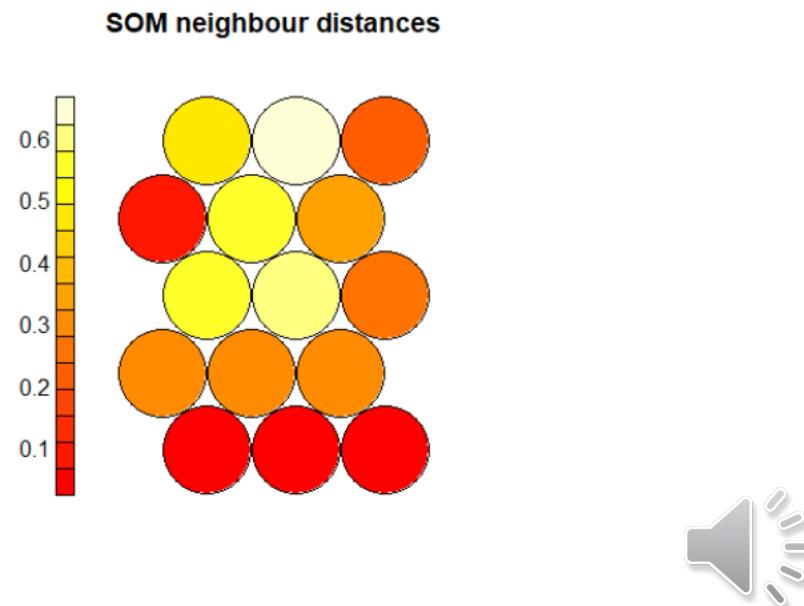


Clustering

■ 자기조직화지도 (Self-Organizing Map, SOM)

□ 장점 & 단점

- SOM에는 오직 숫자형 데이터만 입력할 수 있음
- 거리 함수를 이용하여 손쉽게 2차원 데이터의 클러스터를 생성하여 데이터의 패턴을 이해할 수 있음
- 어떤 유사성/거리 함수를 선택하느냐에 따라 클러스터의 내용이 매우 크게 달라질 수 있음
 - type="dist.neighbours" : 설정 후 뉴런의 이웃간 거리 값 확인 (값이 높을 수록 비유사성이 높음)
 - type="counts"
 - type="quality"
 - type="mapping"



- SOM 모델의 수학 연산상의 복잡성으로 인해 수천 개 이상의 데이터세트는 분석하는 것이 불가능함

Clustering

R

■ 자기조직화지도 (Self-Organizing Map, SOM)

데이터 불러오기

```
#####
# SOM Clustering with R
#####
library(kohonen)
```



"Wines" Data 불러오기
data("wines")

	alcohol	malic acid	ash	ash alkalinity	magnesium	tot. phenols	flavonoids	non-flav. phenols	proanth
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28
2	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81
3	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18
4	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82
5	14.20	1.76	2.45	15.2	112	3.27	3.39	0.34	1.97
6	14.39	1.87	2.45	14.6	96	2.50	2.52	0.30	1.98

데이터 정규화

데이터 정규화 (표준편차)
wines.sc <- scale(wines)

	alcohol	malic acid	ash	ash alkalinity	magnesium	tot. phenols	flavonoids	non-flav. phenols	proanth
1	0.255100804	-0.50020530	-0.82215294	-2.493037162	0.02909756	0.57104557	0.737543740	-0.82081008	-0.53705186
2	0.205645336	0.01796903	1.10455622	-0.274858998	0.09964918	0.81048430	1.218188951	-0.49991911	2.13990403
3	1.701673241	-0.34832662	0.48655516	-0.814415849	0.94626865	2.48655536	1.468524998	-0.98125566	1.03762808
4	0.304556272	0.22345196	1.83161628	0.444550136	1.29902677	0.81048430	0.667449647	0.22208555	0.40775610
5	1.491487502	-0.51807338	0.30479015	-1.294021938	0.87571703	1.56072563	1.368390579	-0.17902835	0.67020276
6	1.726400975	-0.41979894	0.30479015	-1.473874222	-0.25310893	0.33160685	0.497221134	-0.49991911	0.68769920

Clustering

■ 자기조직화지도 (Self-Organizing Map, SOM)

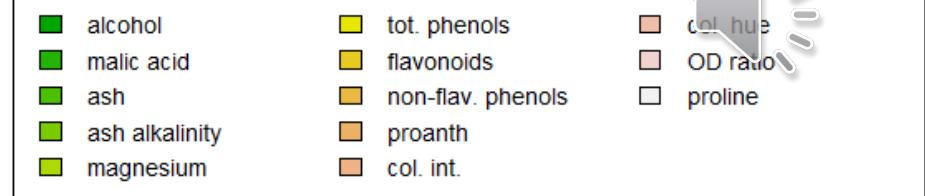
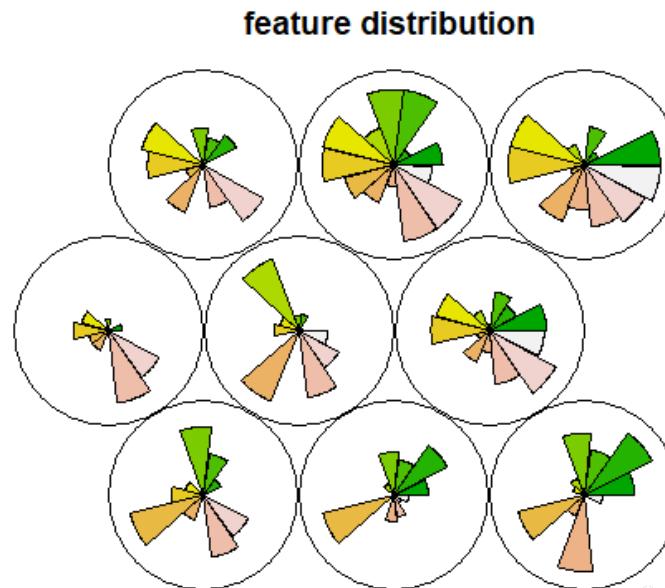
SOM 설정

```
set.seed(7)  
wine.som <- som(wines.sc, grid = somgrid(3, 3, "hexagonal"))  
str(wine.som)
```

SOM 시각화

```
plot(wine.som, main = "feature distribution")
```

속성들 가중치 기여율

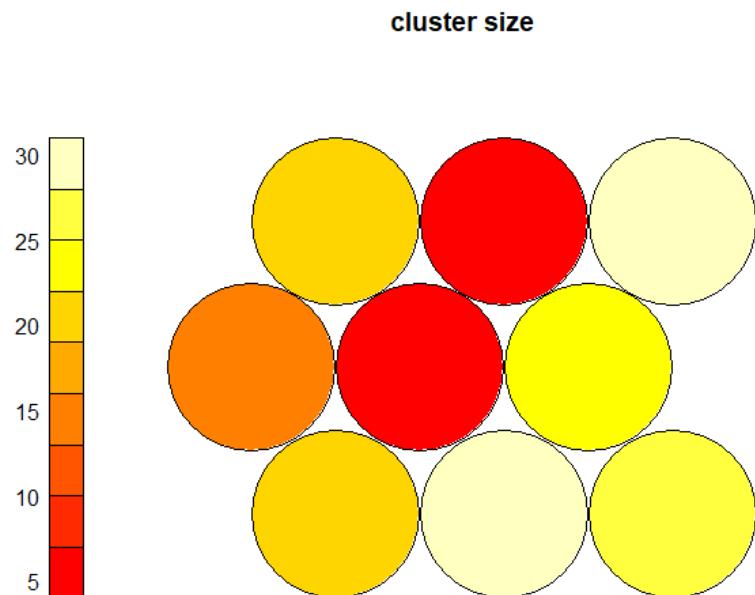


Clustering

■ 자기조직화지도 (Self-Organizing Map, SOM)

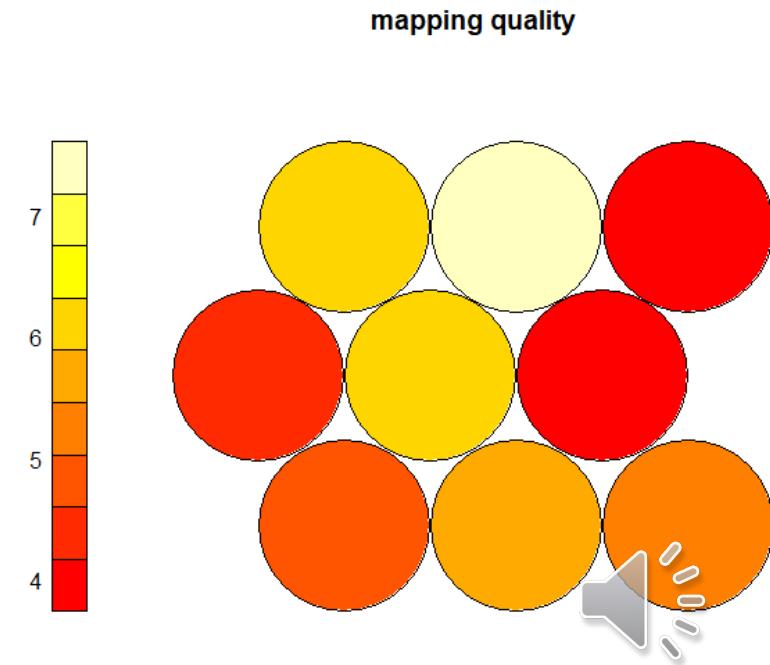
SOM 시각화

```
plot(wine.som, type="counts",  
main = "cluster size")
```



할당된 element들의 유사도

```
plot(wine.som, type="quality",  
main = "mapping quality")
```



Thank you





Introduction to Artificial Intelligence [AICS223]

Evaluation Metric - 성능평가 (W14)

Prof. Mee Lan Han (aeternus1203@gmail.com)

고려대학교

인공지능사이버보안학과



**KOREA
UNIVERSITY**

CONTENTS

- 분류모델 성능지표
- 회귀모델 성능지표
- 군집모델 성능지표
 - 유사도측정
- 복잡도 계산
 - 시간 복잡도



Evaluation Metric (Classification)

■ Confusion Matrix (혼동행렬)

- 모델의 성능을 평가할 때 사용되는 지표
- Training (실제 관측값)을 통한 Prediction 성능 (예측값, 검증값) 측정

		True condition, Actual class	
		1 True condition	Condition negative
Predicted condition, Predicted class	2	Total population	Condition positive
	Predicted Condition	Predicted condition positive	True positive
		Predicted condition negative	False negative
			True negative

- 라벨링 된 데이터셋 (Training): True condition, Actual class
- 예측을 위한 데이터셋 (Testing): Predicted condition, Predicted class



Evaluation Metric (Classification)

■ Confusion Matrix (혼동행렬)

- 모델의 성능을 평가할 때 사용되는 지표
- Training (실제 관측값)을 통한 Prediction 성능 (예측값, 검증값) 측정

		Predicted Class		Predicted condition/ Predicted class	
		Positive	Negative		
True condition/ Actual class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$	민감도
	Negative	False Positive (FP) Type I Error	True Negative (TN)		
		Positive	Negative Predictive	Accuracy $\frac{TN}{(TN + FP)}$	특이도
• 라벨링 된 데이터셋 (Training): True condition, Actual class • 예측을 위한 데이터셋 (Testing): Predicted condition, Predicted class			(TN + FN)		

Evaluation Metric (Classification)

■ Confusion Matrix (혼동행렬)

- 4가지 정보

		Predicted Class	
		P	N
Actual Class	P	True Positive (TP)	False Negative (FN)
	N	False Positive (FP)	True Negative (TN)

P (관심 범주에서 대상)
N (관심 범주에서 대상이 아닌 것)

- TP: 공격을 공격으로 분류
- FN: 공격을 정상으로 분류
- FP: 정상을 공격으로 분류
- TN: 정상을 정상으로 분류
- TP: 암환자를 암환자로 분류
- FN: 암환자를 일반환자로 분류
- FP: 일반환자를 암환자로 분류
- TN: 일반환자를 일반환자로 분류



Evaluation Metric (Classification)

■ Confusion Matrix (혼동행렬)

- TP, FN, FP, TN를 통한 주요 성능지표 산출식

P (관심 범주에서 대상)
N (관심 범주에서 대상이 아닌 것)

		Predicted Class	
		P	N
Actual Class	P	True Positive (TP)	False Negative (FN)
	N	False Positive (FP)	True Negative (TN)

용어	산출식	설명
Accuracy (정확도)	$(TP+TN) / (TP+TN+FN+FP)$	모델이 입력된 데이터에 대해 얼마나 정확하게 예측하는지 나타내는 지표
Precision (정밀도)	$TP / (TP+FP)$	모델의 예측값이 얼마나 정확하게 예측됐는가를 나타내는 지표
Recall (재현율)	$TP / (TP+FN)$	P라벨인 실제값 항목 중에서 P라벨로 예측된 항목의 비율
F1-score (F-measure)	$2 * \text{재현율} * \text{정밀도} / (\text{재현율} + \text{정밀도})$	두 값 조화평균내서 하나의 수치로 나타낸 지표



Evaluation Metric (Classification)

■ Confusion Matrix (혼동행렬)

- TP, FN, FP, TN를 통한 주요 성능지표 산출식

P (관심 범주에서 대상)
N (관심 범주에서 대상이 아닌 것)

		Predicted Class	
		P	N
Actual Class	P	True Positive (TP)	False Negative (FN)
	N	False Positive (FP)	True Negative (TN)

용어	산출식	설명
Accuracy (정확도)	$(TP+TN) / (TP+TN+FN+FP)$	모델이 입력된 데이터에 대해 얼마나 정확하게 예측하는지 나타내는 지표
Precision (정밀도)	$TP / (TP+FP)$	모델의 예측값이 얼마나 정확하게 예측됐는가를 나타내는 지표
Recall (재현율)	$TP / (TP+FN)$	P라벨인 실제값 항목 중에서 P라벨로 예측된 항목의 비율
F1-score (F-measure)	$2 * \text{재현율} * \text{정밀도} / (\text{재현율} + \text{정밀도})$	두 값 조화평균내서 하나의 수치로 나타낸 지표



Evaluation Metric (Classification)

■ Confusion Matrix (혼동행렬)

- TP, FN, FP, TN를 통한 주요 성능지표 산출식

P (관심 범주에서 대상)

N (관심 범주에서 대상이 아닌 것)

		Predicted Class	
		P	N
Actual Class	P	True Positive (TP)	False Negative (FN)
	N	False Positive (FP)	True Negative (TN)

용어	산출식	설명
Accuracy (정확도)	$(TP+TN) / (TP+TN+FN+FP)$	모델이 입력된 데이터에 대해 얼마나 정확하게 예측하는지 나타내는 지표
Precision (정밀도)	$TP / (TP+FP)$	모델의 예측값이 얼마나 정확하게 예측됐는가를 나타내는 지표
Recall (재현율)	$TP / (TP+FN)$	P라벨인 실제값 항목 중에서 P라벨로 예측된 항목의 비율
F1-score (F-measure)	$2 * \text{재현율} * \text{정밀도} / (\text{재현율} + \text{정밀도})$	두 값 조화평균내서 하나의 수치로 나타낸 지표



Evaluation Metric (Classification)

■ Confusion Matrix (혼동행렬)

- TP, FN, FP, TN를 통한 주요 성능지표 산출식

P (관심 범주에서 대상)
N (관심 범주에서 대상이 아닌 것)

		Predicted Class	
		P	N
Actual Class	P	True Positive (TP)	False Negative (FN)
	N	False Positive (FP)	True Negative (TN)

용어	산출식	설명
Accuracy (정확도)	$(TP+TN) / (TP+TN+FN+FP)$	모델이 입력된 데이터에 대해 얼마나 정확하게 예측하는지 나타내는 지표
Precision (정밀도)	$TP / (TP+FP)$	모델의 예측값이 얼마나 정확하게 예측됐는가를 나타내는 지표
Recall (재현율)	$TP / (TP+FN)$	P라벨인 실제값 항목 중에서 P라벨로 예측된 항목의 비율
F1-score (F-measure)	$2 * \text{재현율} * \text{정밀도} / (\text{재현율} + \text{정밀도})$	두 값 조화평균내서 하나의 수치로 나타낸 지표



Evaluation Metric (Classification)

■ Confusion Matrix (혼동행렬)

- 성능지표 산출식

		Predicted Class	
		P	N
Actual Class	P	True Positive (TP)	False Negative (FN)
	N	False Positive (FP)	True Negative (TN)

- 클래스 불균형 문제가 있는 경우에는 정확도와 정밀도가 높고, 재현율이 크게 떨어지는 경향이 발생할 수 있음

Actual Class	Predicted Class		
		DoS 공격	정상
DoS 공격	27	3	
정상	2	68	

Binary classification

용어	산출식	결과
Accuracy (정확도)	$(TP+TN) / (TP+TN+FN+FP)$	$95/100=0.95$
Precision (정밀도)	$TP / (TP+FP)$	$27/29=0.931$
Recall (재현율)	$TP / (TP+FN)$	$27/30=0.9$
F1-score (F-measure)	$2 * \text{재현율} * \text{정밀도} / (\text{재현율} + \text{정밀도})$	$1.6758/1.831=0.915$



Evaluation Metric (Classification)

■ Confusion Matrix (혼동행렬)

□ 성능지표 산출식

		Predicted Class	
		P	N
Actual Class	P	True Positive (TP)	False Negative (FN)
	N	False Positive (FP)	True Negative (TN)

- 클래스 불균형 문제가 있는 경우에는 정확도와 정밀도가 높고, 재현율이 크게 떨어지는 경향이 발생할 수 있음

Actual Class	Predicted Class			
		DoS	Fuzzy	Mal
DoS	94	16	10	
Fuzzy	21	113	16	
Mal	4	4	92	

Multi-class classification

용어	산출식	결과
Accuracy (정확도)	$(TP+TN) / (TP+TN+FN+FP)$	<ul style="list-style-type: none">• DoS 클래스로 식별된 원소는 DoS 클래스 (정답)이 됨
Precision (정밀도)	$TP / (TP+FP)$	<ul style="list-style-type: none">• 나머지 Fuzzy, Mal이라고 식별된 것들은 A' (오답) 클래스로 분류됨
Recall (재현율)	$TP / (TP+FN)$	
F1-score (F-measure)	$2 * \text{재현율} * \text{정밀도} / (\text{재현율} + \text{정밀도})$	



Evaluation Metric (Classification)

■ Confusion Matrix (혼동행렬)

□ 클래스 불균형 문제 (Class imbalance)

- 학습 데이터 내 클래스 변수가 균일하게 분포하지 않아, 하나의 성능지표 산출식 값에 치우치는 편향된 모델이 생성되도록 데이터를 학습하는 문제
 - 편향된 모델: 대부분의 샘플을 치우친 클래스로 분류하는 모델을 의미

		Predicted Class	
		공격	정상
Actual Class	공격	0	8
	정상	2	9990

TN으로 치우침

클래스 변수의 분포가 한 쪽에 치우친 데이터로 학습한 분류 모델



입력 데이터에 대해 다수 클래스(TN)라고 분류, 그 외 소수 클래스에 대한 예측 성능이 저하



$$\begin{aligned} \text{Accuracy} &= (\text{TP}+\text{TN}) / (\text{TP}+\text{TN}+\text{FN}+\text{FP}) \\ &= (0+9990) / (0+9990 + 8 + 2) \\ &= 0.999 \text{ (공격 데이터는 제대로 탐지가 되지 않음)} \end{aligned}$$

?

Evaluation Metric (Classification)

■ Confusion Matrix (혼동행렬)

□ 클래스 불균형 문제 (Class imbalance)

		Predicted Class	
		암환자	일반환자
Actual Class	암환자	0	8
	일반환자	2	9990

		Predicted Class	
		불량품	정상품
Actual Class	불량품	0	8
	정상품	2	9990



Evaluation Metric (Classification)

■ Confusion Matrix (혼동행렬)

□ 혼동행렬의 개념과 수식을 통해 Accuracy 와 Recall 값 측정

TABLE IV: CONFUSION MATRIX

		Predicted class	
		Intrusion (class = positive)	Normal (class = negative)
Actual class	Intrusion (class = positive)	TP	FN
	Normal (class = negative)	FP	TN

Anomaly detection

- TP: an attack packet classified as **attack**
- FP: a normal packet classified as **attack**
- FN: an attack packet classified as **normal**
- TN: a normal packet classified as **normal**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

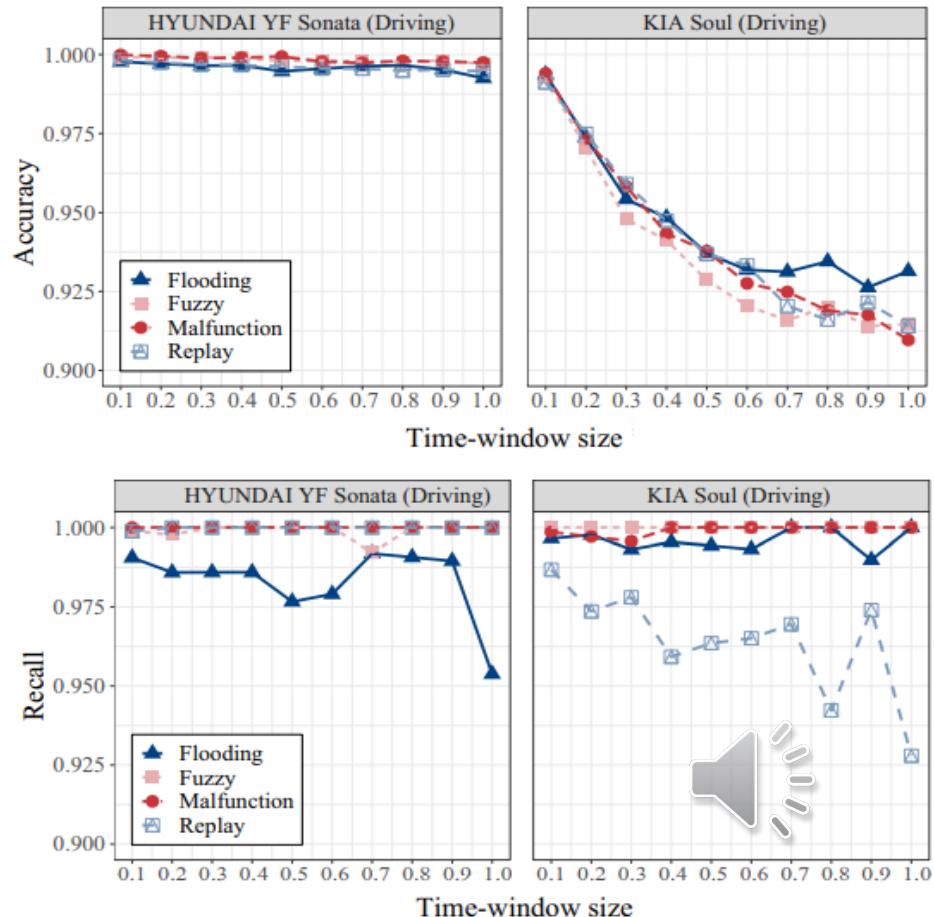
$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$F1 = \frac{2(\text{precision} * \text{recall})}{\text{precision} + \text{recall}}$$

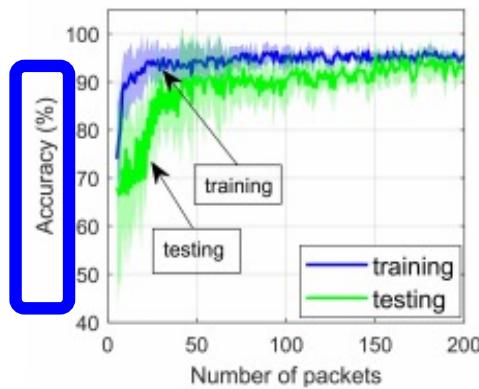
$$AUC = \int_0^1 \Pr [TP] (v) dv,$$

where $\Pr [TP]$ is a function of $v = \Pr [FP]$

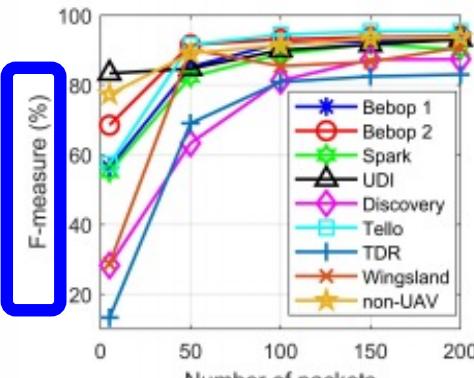


Evaluation Metric (Classification)

■Confusion Matrix (혼동행렬)



(a) Training and testing accuracy on each subset j .



(b) F-measure metric.

Fig. 6. Classification performance evaluation.

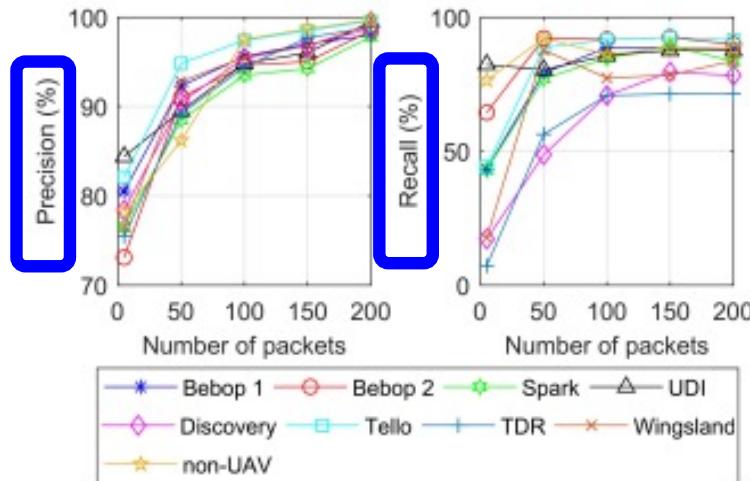


Fig. 7. Precision and recall metrics.

TABLE VII EXCERPT OF THE OUTCOMES (PERCENTAGES) FOR THE WORST (MOST RESTRICTIVE) AND BEST (MOST RELAXED) STRATEGIES						
File size	γ	TP	FN	FP	TN	Accuracy (TP+TN)
1 (Others)	0.1					
	2	46.70	3.29	26.09	23.90	70.60
1	0.1	26.73	23.27	14.31	35.69	62.42
	2	46.69	3.31	28.01	21.99	68.68
2	0.1	26.58	23.42	10.41	39.59	66.17
	2	46.68	3.32	21.18	28.82	75.50
4	0.1	26.74	23.26	7.13	42.87	69.61
	2	46.77	3.23	15.1	34.90	81.67
8	0.1	26.68	23.32	4.84	45.16	71.84
	2	46.69	3.31	10.49	39.51	86.20
16	0.1	26.74	23.26	3.11	46.89	73.63
	2	46.86	3.14	7.01	42.99	89.85
32	0.1	26.85	23.15	1.81	48.19	75.04
	2	46.66	3.34	3.89	46.11	92.77
64	0.1	27.01	22.90	0.98	49.02	76.03
	2	46.82	3.18	2.09	47.91	94.72

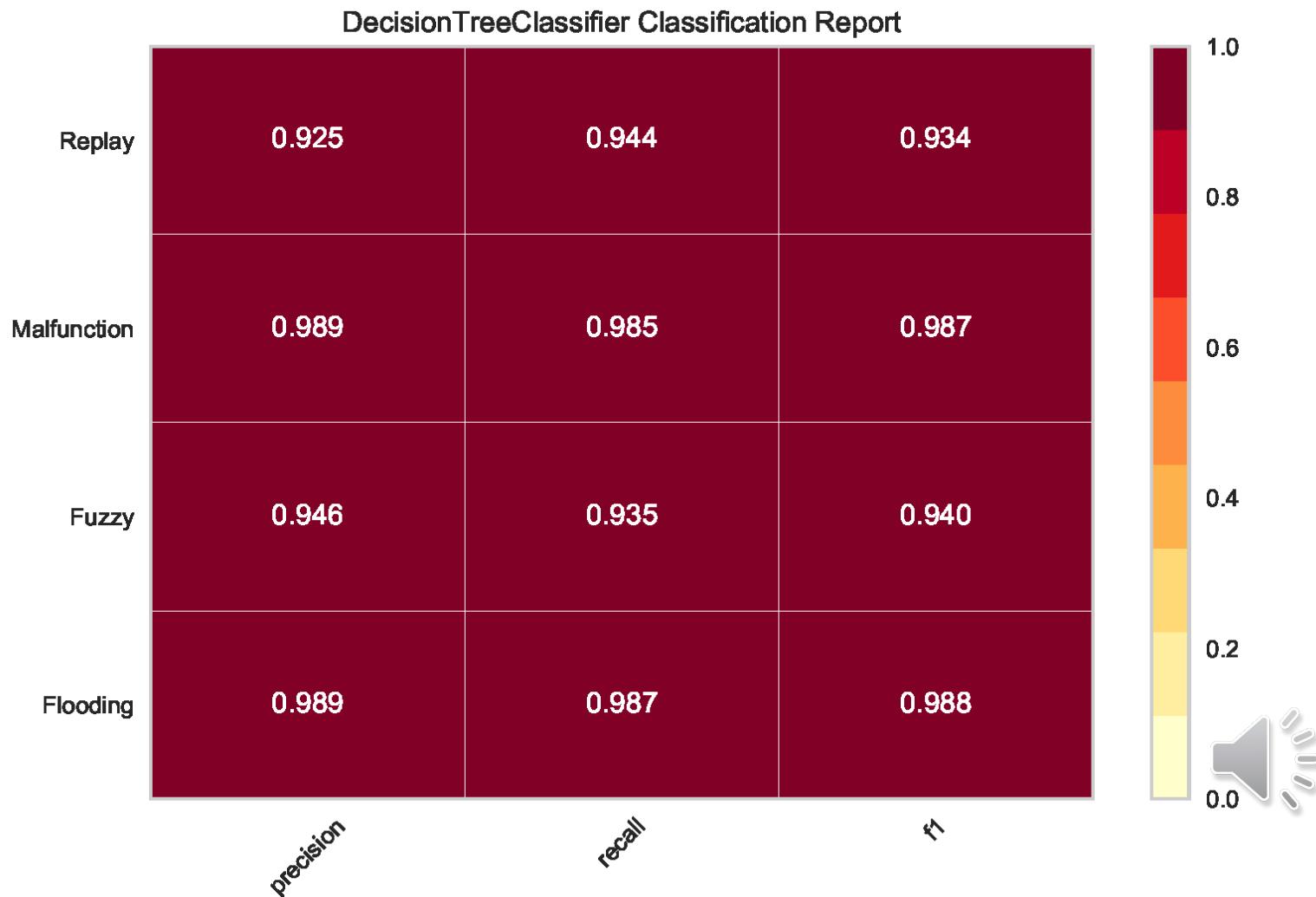
TABLE VI: PRECISION, RECALL, AND F1-SCORE FOR DECISION TREE CLASSIFIER (0.2 s TIME-WINDOW)

Dataset	Condition	HYUNDAI YF Sonata			KIA Soul		
		Types	P	R	F	P	R
Driving	Flooding	0.989	0.987	0.988	0.944	0.958	0.951
	Fuzzy	0.946	0.935	0.940	0.939	0.950	0.945
	Malfunction	0.989	0.985	0.987	0.982	0.953	0.967
	Replay	0.925	0.944	0.934	0.953	0.941	0.941
Stationary	Flooding	0.978	0.990	0.984	0.970	0.976	0.973
	Fuzzy	0.952	0.940	0.946	0.947	0.945	0.946
	Malfunction	0.991	0.989	0.990	0.968	0.969	0.969
	Replay	0.929	0.935	0.932	0.933	0.928	0.931

P: Precision, R: Recall, F: F1-score

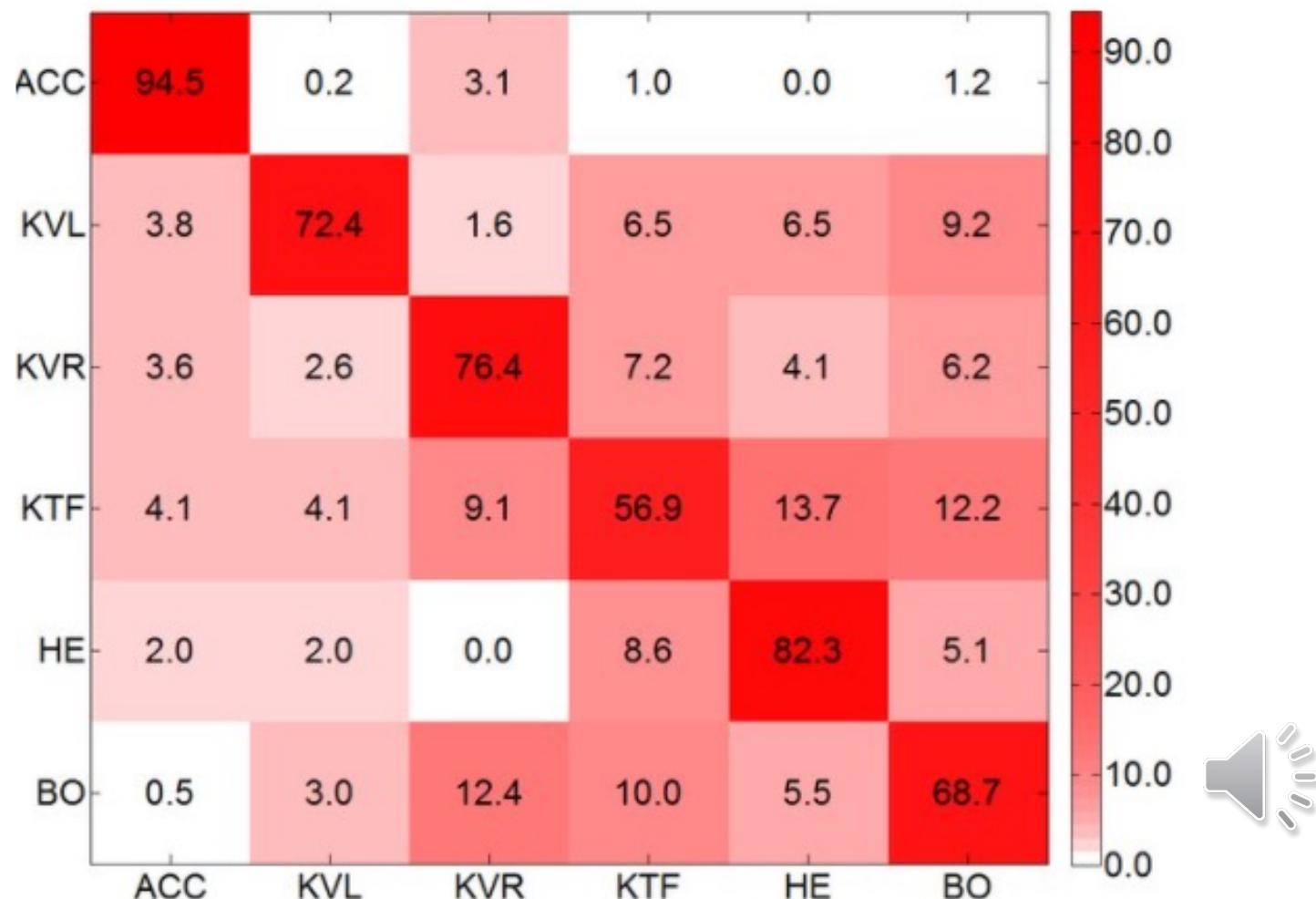
Evaluation Metric (Classification)

■ Confusion Matrix 시각화 표현 (Heatmap visualization)



Evaluation Metric (Classification)

■Confusion Matrix 시각화 표현 (Heatmap visualization)



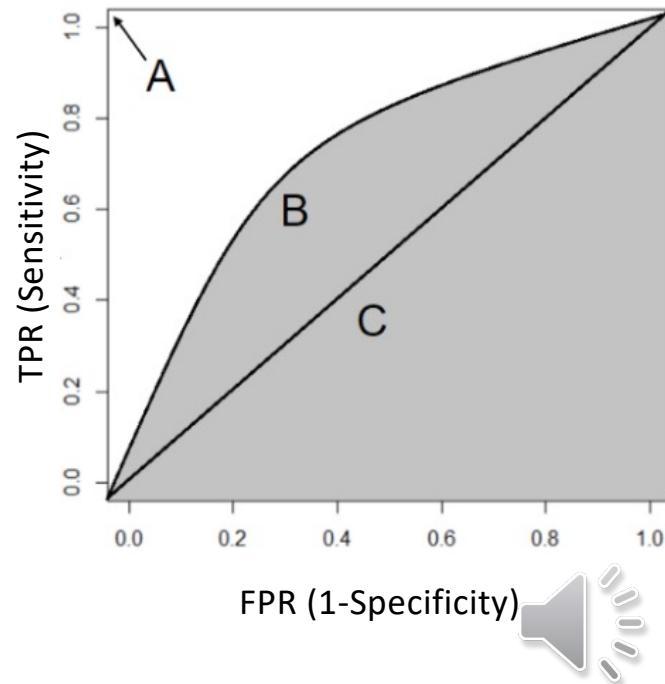
Evaluation Metric (Classification)

■ ROC curve & AUC (Area Under Curve)

- 모델의 성능이 기준선을 넘었는지 확인하기 위해 시각적으로 표현한 그래프
- FPR (1-Specificity, 1-특이도) 을 x축으로 정의
- TPR (Sensitivity, 민감도) 을 Y축으로 정의

- A-완벽한 모델: TPR은 1, FPR은 0
- B-좋은 모델
- C-평균/기본 모델

※ 적용하고자 하는 모델의 ROC 곡선이
기본 대각선보다 위에 있다면,
이 모델의 성능은 기본선보다 좋다고 평가



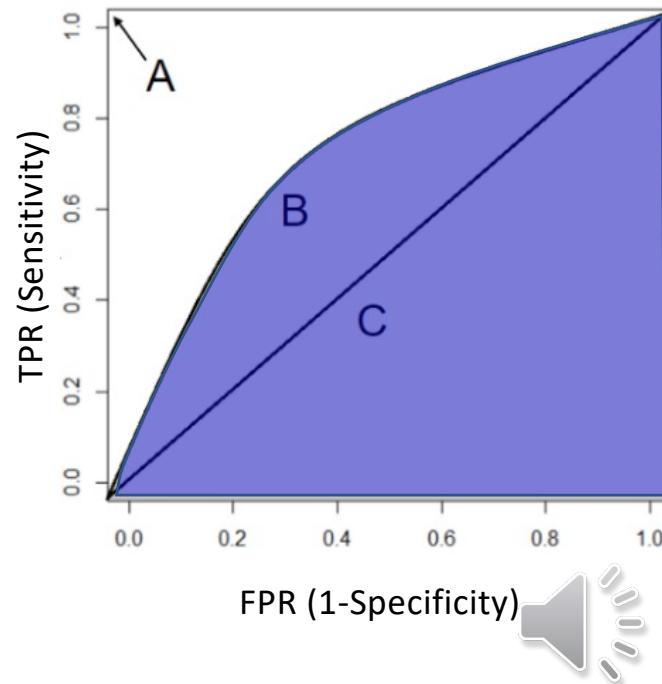
Evaluation Metric (Classification)

■ ROC curve & AUC (Area Under Curve)

- 평가 모델의 ROC 곡선 아래 영역의 넓이를 의미
- AUC 값은 모델에서 항목을 임의로 추출 시 긍정 항목이 부정 항목보다 더 선택될 확률을 나타냄
- 높은 AUC가 더 좋은 모델

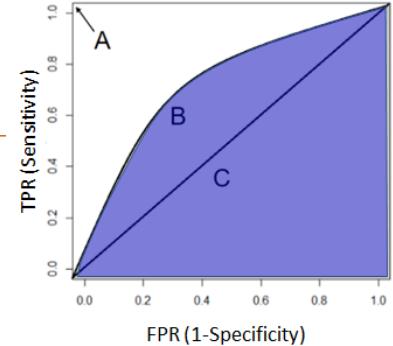
- A-완벽한 모델: TPR은 1, FPR은 0
- B-좋은 모델
- C-평균/기본 모델

※ 적용하고자 하는 모델의 AUC 면적이 넓을수록 이 모델의 성능은 좋다고 평가



Evaluation Metric (Classification)

■ ROC curve & AUC

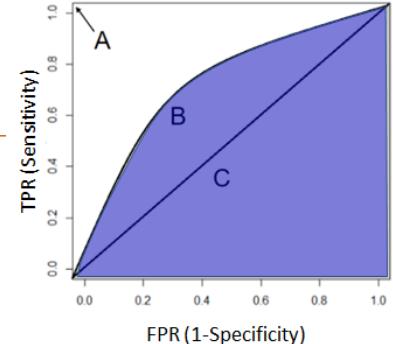


		Predicted Class	
		P	N
Actual Class	P	True Positive (TP)	False Negative (FN)
	N	False Positive (FP)	True Negative (TN)

용어	산출식	설명
Specificity (특이도)	$TN / (FP+TN)$	N 라벨인 실제값 항목 중에서 N 라벨로 예측된 항목의 비율
FPR (긍정오류율)	$FP / (FP+TN)$	N 라벨인 실제값 항목 중에서 P라벨로 잘못 예측된 항목의 비율
Sensitivity (민감도)	$TP / (TP+FN)$	P 라벨인 실제값 항목 중에서 P 라벨로 예측된 항목의 비율
FNR (부정오류율)	$FN / (TP+FN)$	P 라벨인 실제값 항목 중에서 N 라벨로 잘못 예측된 항목의 비율

Evaluation Metric (Classification)

■ ROC curve & AUC

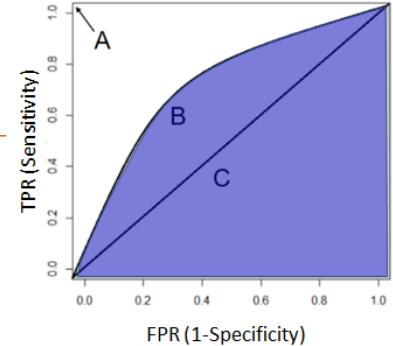


		Predicted Class	
		P	N
Actual Class	P	True Positive (TP)	False Negative (FN)
	N	False Positive (FP)	True Negative (TN)

용어	산출식	설명
Specificity (특이도)	$TN / (FP+TN)$	N 라벨인 실제값 항목 중에서 N 라벨로 예측된 항목의 비율
FPR (긍정오류율)	$FP / (FP+TN)$	N 라벨인 실제값 항목 중에서 P라벨로 잘못 예측된 항목의 비율
Sensitivity (민감도)	$TP / (TP+FN)$	P 라벨인 실제값 항목 중에서 P 라벨로 예측된 항목의 비율
FNR (부정오류율)	$FN / (TP+FN)$	P 라벨인 실제값 항목 중에서 N 라벨로 잘못 예측된 항목의 비율

Evaluation Metric (Classification)

■ ROC curve & AUC



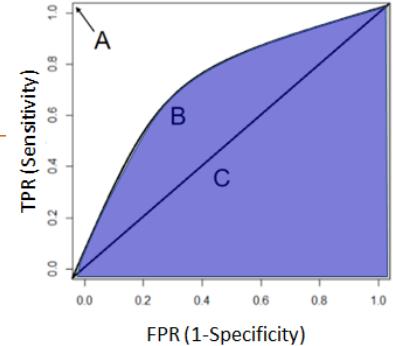
		Predicted Class	
		P	N
Actual Class	P	True Positive (TP)	False Negative (FN)
	N	False Positive (FP)	True Negative (TN)

용어	산출식	설명
Specificity (특이도)	$TN / (FP+TN)$	N 라벨인 실제값 항목 중에서 N 라벨로 예측된 항목의 비율
FPR (긍정오류율)	$FP / (FP+TN)$	N 라벨인 실제값 항목 중에서 P라벨로 잘못 예측된 항목의 비율
Sensitivity (민감도)	$TP / (TP+FN)$	P 라벨인 실제값 항목 중에서 P 라벨로 예측된 항목의 비율
FNR (부정오류율)	$FN / (TP+FN)$	P 라벨인 실제값 항목 중에서 N 라벨로 잘못 예측된 항목의 비율



Evaluation Metric (Classification)

■ ROC curve & AUC



		Predicted Class	
		P	N
Actual Class	P	True Positive (TP)	False Negative (FN)
	N	False Positive (FP)	True Negative (TN)

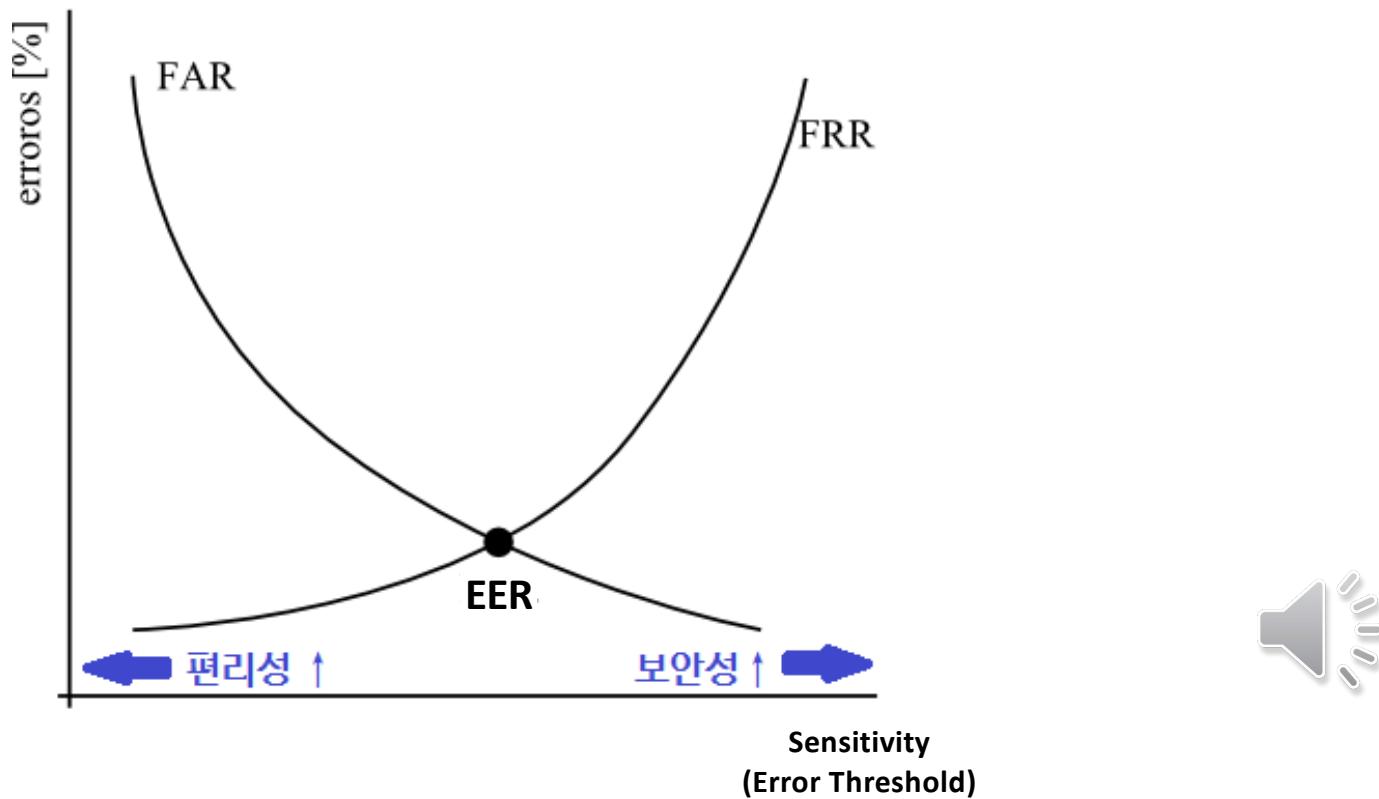
용어	산출식	설명
Specificity (특이도)	$TN / (FP+TN)$	N 라벨인 실제값 항목 중에서 N 라벨로 예측된 항목의 비율
FPR (긍정오류율)	$FP / (FP+TN)$	N 라벨인 실제값 항목 중에서 P라벨로 잘못 예측된 항목의 비율
Sensitivity (민감도)	$TP / (TP+FN)$	P 라벨인 실제값 항목 중에서 P 라벨로 예측된 항목의 비율
FNR (부정오류율)	$FN / (TP+FN)$	P 라벨인 실제값 항목 중에서 N 라벨로 잘못 예측된 항목의 비율



Evaluation Metric (Classification)

■ 인증시스템의 모델 성능 측정 시 사용되는 중요 지표

- **FRR** (False Rejection Rate): 부정거부률, 본인거부률
- **FAR** (False Acceptance Rate): 부정허용률, 타인수락률
- **EER** (Equal Error Rate) == **CER**(Crossover Error Rate): 교차오류률



Evaluation Metric (Classification)

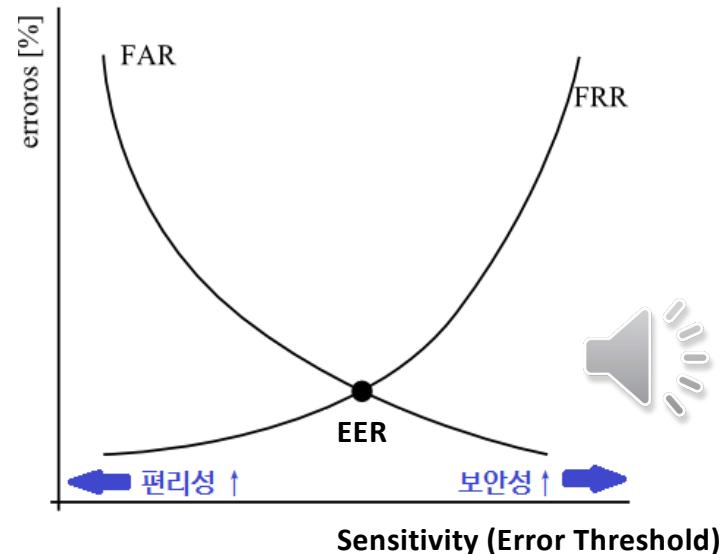
■ FRR (False Rejection Rate): 본인거부률

- 허가된 사용자가 시스템 오류로 인해 거부되는 비율
- 시스템에 등록된 사용자가 사용 시 본인임을 정상적으로 확인하지 못하고 인증이 거부되는 오류
- FRR 0.1%는 1000회의 인증 시 1회 오류가 발생할 가능성

■ FAR (False Acceptance Rate): 타인수락률

- 허가되지 않은 사용자가 시스템의 오류로 인한 접근 허용되는 비율
- 시스템에 등록된 사용자 외 다른 사람을 등록자로 오인하고 인증을 수행하는 오류
- FAR 0.001%는 10만회 인증 시 1회 오류 가능성
- FAR이 낮을수록 인증 시스템은 보안이 더욱 강화되어 있다는 의미

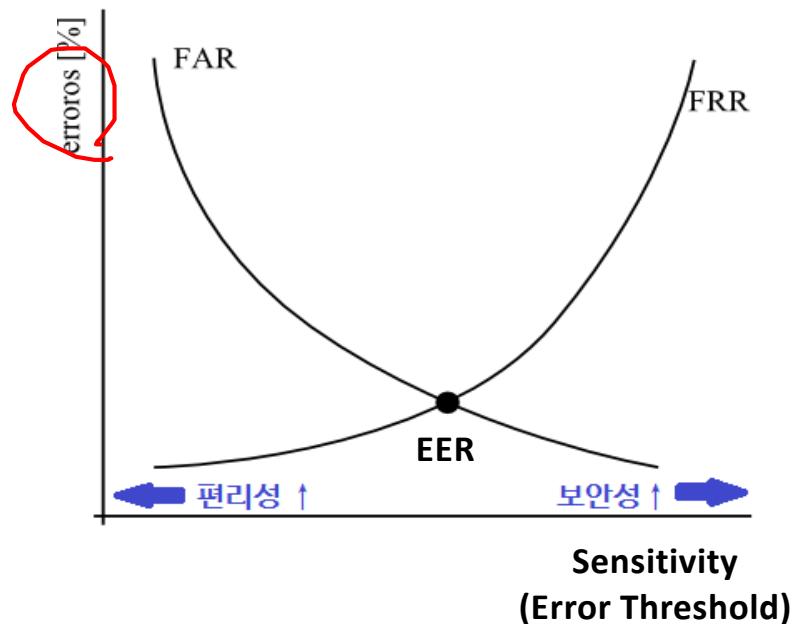
높은 보안 성능을 위한 중요도:
FRR 보다는 FAR 이 더 중요!!



Evaluation Metric (Classification)

■ EER (Equal Error Rate) == CER(Crossover Error Rate)

- 교차오류률
- FRR과 FAR의 교차점 (FAR과 FRR의 수치가 같아질 때의 오류률)
- 생체인증장치 성능을 측정하는 표준평가지점, 낮을수록 정확함



Thank you





Introduction to Artificial Intelligence [AICS223]

Evaluation Metric - 성능평가 (W15)

Prof. Mee Lan Han (aeternus1203@gmail.com)

고려대학교

인공지능사이버보안학과



**KOREA
UNIVERSITY**

CONTENTS

- 분류모델 성능지표
- 회귀모델 성능지표
- 군집모델 성능지표
 - 유사도측정
- 복잡도 계산
 - 시간 복잡도



Evaluation Metric (Regression)

- MSE (Mean Squared Error): 평균 제곱 오차
- MAE (Mean Absolute Error): 평균 절대 오차



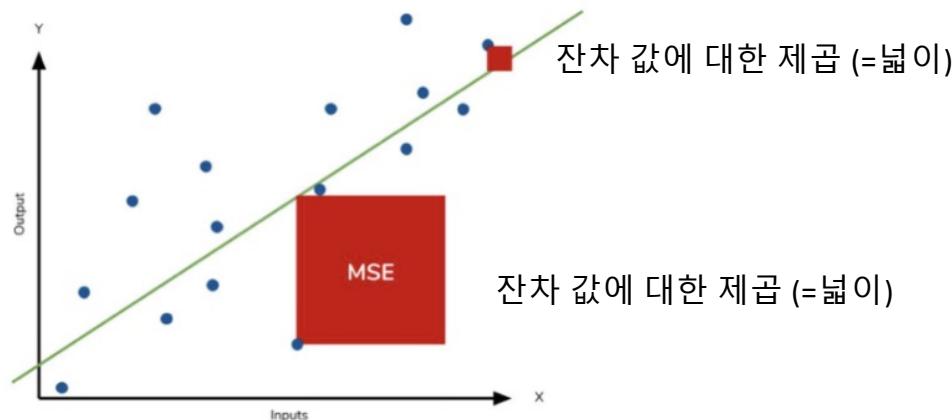
Evaluation Metric (Regression)

■ MSE (Mean Squared Error)

- 가장 일반적이고 직관적인 에러 지표
- 잔차 (에러)의 제곱에 대한 평균으로 계산 -> 값은 낮을수록 좋음
- 회귀 모델은 $Y_i - \hat{Y}_i$ 즉, 오차/잔차/에러가 낮을수록 좋은 모델임

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Y_i : 실제 관측값
 \hat{Y}_i : 예측값 (회귀식), $\hat{Y}_i = \beta_0 + \beta_1 X_i$
 \bar{Y} : 평균, $\bar{Y} = \frac{1}{n} \sum (Y_i)$



Y_i : 실제 관측값
 \hat{Y}_i : 예측값 (회귀식), $\hat{Y}_i = \beta_0 + \beta_1 X_i$
 \bar{Y} : 평균, $\bar{Y} = \frac{1}{n} \sum (Y_i)$

Evaluation Metric (Regression)

■ MSE (Mean Squared Error) 특징

- 지표 자체는 **직관적**이나 제곱을 하기 때문에 **실제값과 예측변수의 단위가 다름**
- 즉, **결과값에 대한 해석의 차이가 발생함**
 - ex) 기온을 예측하는 모델의 MSE가 4라면 이 모델은 평균적으로 2도 정도를 잘못 예측하는 것임
- **잔차에 제곱값을 씌우기 때문에 실제 값에 대해 UnderEstimates/OverEstimates 인지 파악하기 어려움**
 - ex) 테슬라 주가를 예측하는 모델의 MSE가 800이라면 이 모델이 평균적으로 주가를 800원을 높게 예측하는지 800원을 낮게 예측하는지 파악하기 어려움
- 잔차를 제곱하기 때문에 **이상치에 민감**
- 잔차를 제곱하기 때문에 **1미만의 에러는 더 작아지고, 그 이상의 에러는 더 커짐**



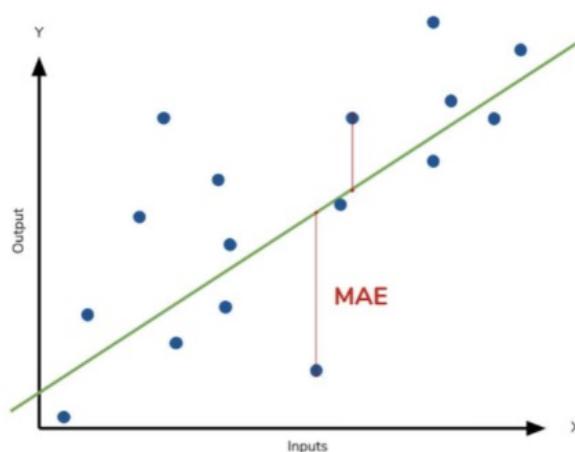
Evaluation Metric (Regression)

■ MAE (Mean Absolute Error)

- 가장 일반적이고 직관적인 에러 지표
- 잔차 (에러)의 절대값에 대한 평균으로 계산 -> 값은 낮을수록 좋음
- 회귀 모델은 $Y_i - \hat{Y}_i$ 즉, 오차/잔차/에러가 낮을수록 좋은 모델임

$$MSE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i|$$

Y_i : 실제 관측값
 \hat{Y}_i : 예측값 (회귀식), $\hat{Y}_i = \beta_0 + \beta_1 X_i$
 \bar{Y} : 평균, $\bar{Y} = \frac{1}{n} \sum (Y_i)$



Evaluation Metric (Regression)

■ MAE (Mean Absolute Error)

- 지표 자체가 **직관적**이며 **예측변수와 단위가 같음**
- **잔차에 절댓값을 써우기 때문에 실제 값에 대해 UnderEstimates/OverEstimates 인지 파악하기 어려움**

ex) 테슬라 주가를 예측하는 모델의 MAE가 800이라면,
→ 이 모델이 평균적으로 주가를 800원을 높게 예측하는지,
800원을 낮게 예측하는지 파악하기 어려움



Evaluation Metric (Clustering)

■ 군집타당성지표

- 목적 변수가 있을 경우: 라벨링이 되어 있는 경우
 - Homogeneity score (동질성, 균질성 점수)
 - Completeness score (완성도 점수)
 - V measure
- 목적 변수가 없을 경우: 라벨링이 없는 경우
 - Dunn Index (DI)
 - Silhouette Index

■ 유사도 측정 (거리 측정)을 위한 지표

- 유클리디언 거리(Euclidean distance)
- 맨하튼 거리 (Manhattan Distance)
- 마할라노비스 거리(Mahalanobis distance)
- 코사인 유사도(Cosine similarity)
- 자카드 유사도(Jaccard similarity)



Evaluation Metric (Clustering)

■ 군집화 (Clustering) 타당성 평가

□ 군집타당성지표

- 군집을 만든 결과가 얼마나 유용한지 검증
- 목적 변수가 있을 경우와 목적 변수가 없을 경우로 나누어 타당성지표 적용
- 목적변수가 없을 경우
 - 군집 간 거리(Distance), 군집의 지름(Diameter), 군집의 분산(Distribution)을 고려
- 목적 변수가 있을 경우
 - ① Homogeneity score (동질성, 균질성 점수)
 - ② Completeness score (완성도 점수)
 - ③ V measure
- 목적 변수가 없을 경우
 - ① Dunn Index (DI)
 - ② Silhouette Index



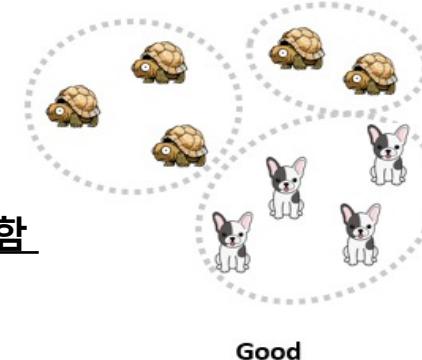
Evaluation Metric (Clustering)

■ 군집화 (Clustering) 타당성 평가

1) 목적변수가 존재할 경우

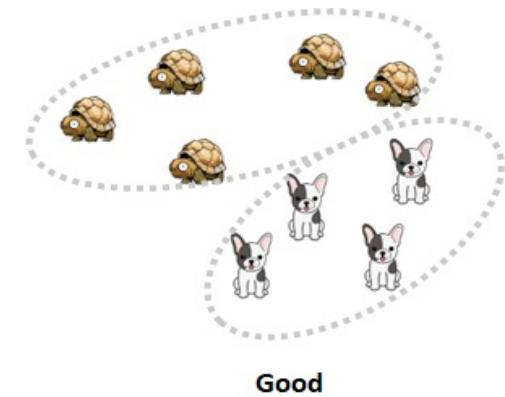
- **Homogeneity score (동질성 점수)**

- 목적변수에 대한 클러스터 라벨링의 동질성 측정항목
- 클러스터링 수행 후 도출된 모든 클러스터가 단일 클래스의 멤버만 포함
- 척도 값이 클수록 좋은 군집 알고리즘으로 평가



- **Completeness score (완성도 점수)**

- 각 클래스의 모든 객체들이 동일한 클러스터의 멤버가 될 때, 클러스터링 결과는 완전성을 만족시킴
- 주어진 범주의 모든 데이터가 같은 군집 내에 있는 것을 의미
- 척도 값이 클수록 좋은 군집 알고리즘으로 평가



- **measure (= Normalised Mutual Information (NMI))**

- Homogeneity score (동질성)과 Completeness score (완전성)의 조화평균



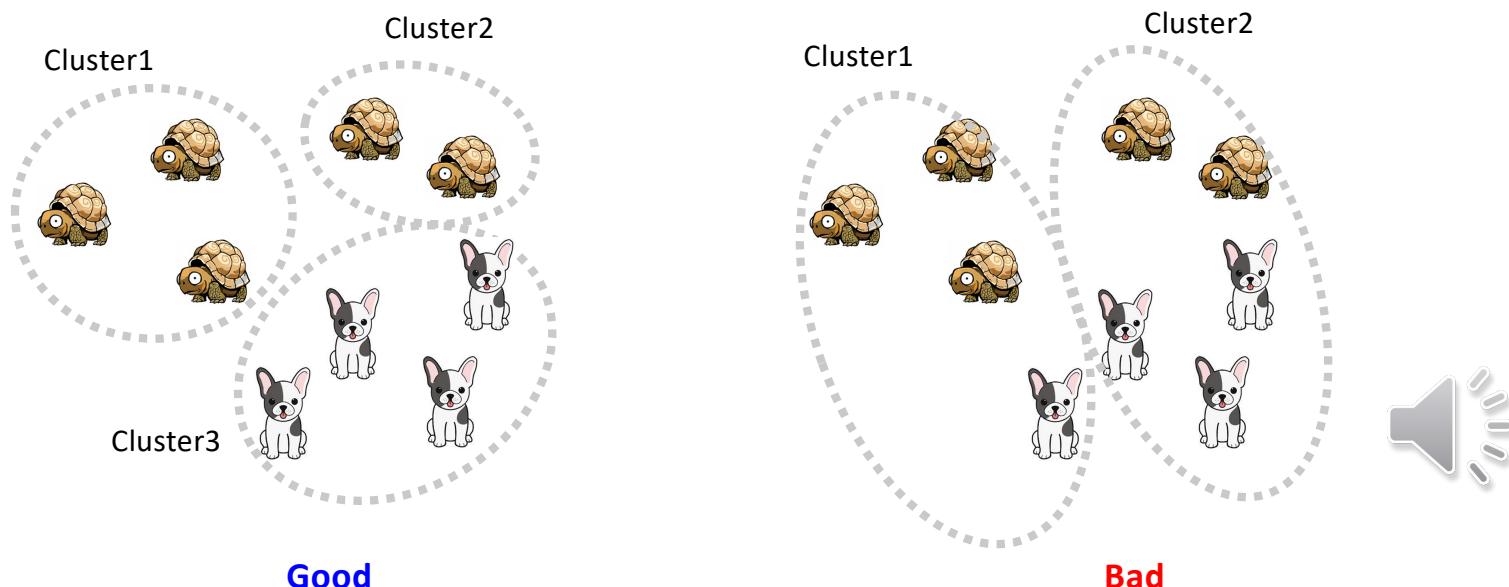
Evaluation Metric (Clustering)

■ 군집화 (Clustering) 타당성 평가

1) 목적변수가 존재할 경우

- **Homogeneity score (동질성 점수)**

- 목적변수에 대한 클러스터 라벨링의 동질성 측정항목
- 클러스터링 수행 후 도출된 모든 클러스터가 단일 클래스의 멤버만 포함
- 각 클래스의 모든 객체들이 동일한 클래스로부터 온 객체일 때, 클러스터링 결과는 동질성을 만족시킴
- Score: 0.0에서 1.0까지, (큰 값이 좋음)



Evaluation Metric (Clustering)

■ 군집화 (Clustering) 타당성 평가

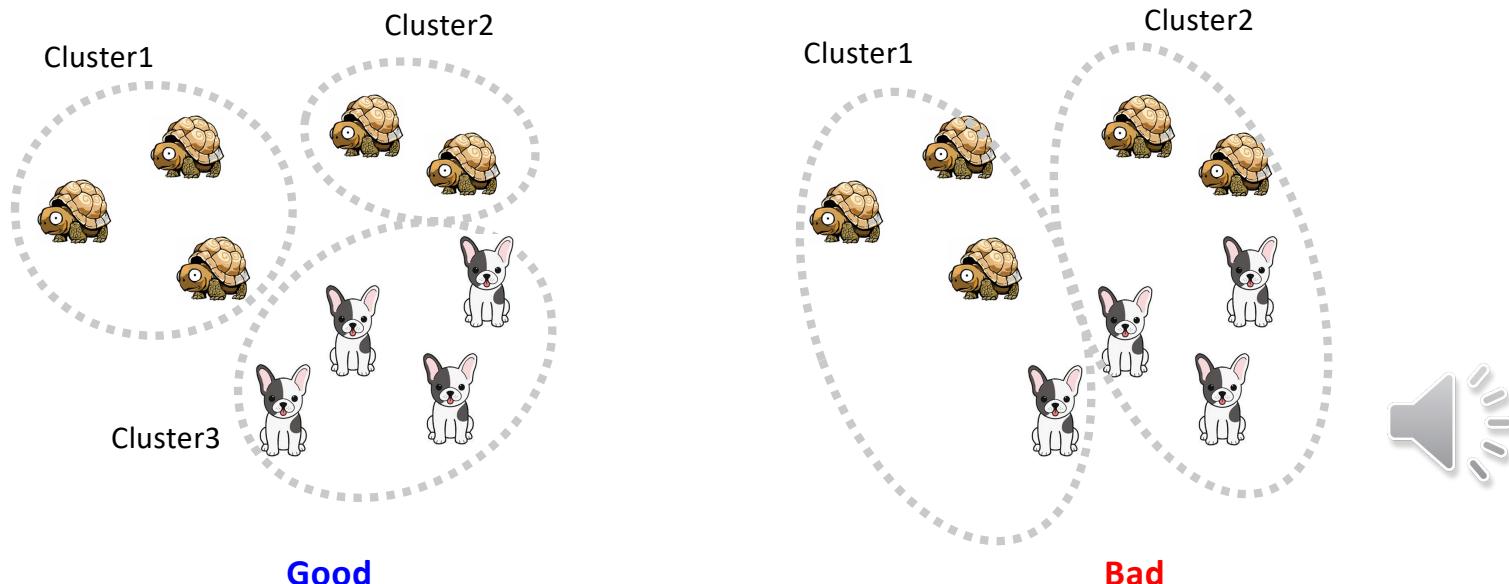
1) 목적변수가 존재할 경우

- Homogeneity score (동질성 점수)

$$H(C|K) : n_{C_k} / n_k$$

n_{C_k} : 클러스터 k 내 객체가 c 로 라벨링된 샘플 수
 n_k : 클러스터 k 내 객체 수

$$h = 1 - \frac{H(C|K)}{H(C)}$$



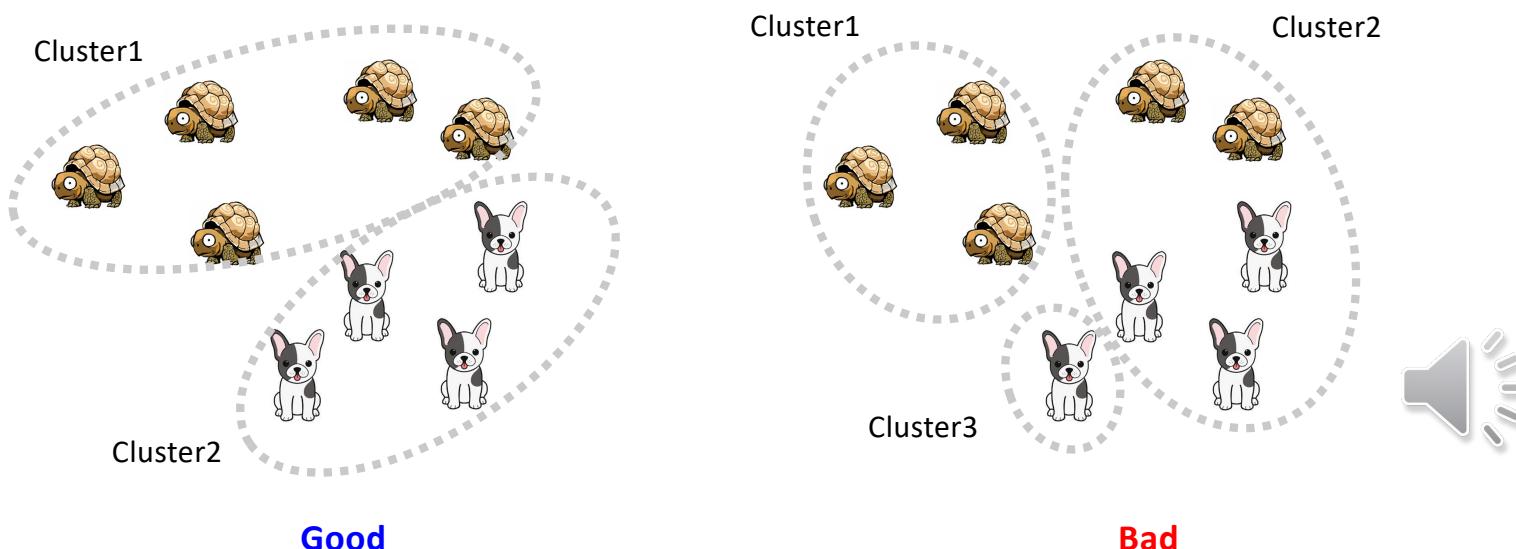
Evaluation Metric (Clustering)

■ 군집화 (Clustering) 타당성 평가

1) 목적변수가 존재할 경우

- **Completeness score (완성도 점수)**

- 각 클래스의 모든 객체들이 동일한 클러스터의 멤버가 될 때, 클러스터링 결과는 완전성을 만족시킴
- 주어진 범주의 모든 데이터 점이 같은 군집 내에 있는 것을 의미함
- 척도 값이 클수록 좋은 군집 알고리즘으로 평가
- 동일한 실제 레이블을 가진 모든 샘플을 동일한 클러스터에 할당해야 함
- Score: 0.0에서 1.0까지, (큰 값이 좋음)



Evaluation Metric (Clustering)

■ 군집화 (Clustering) 타당성 평가

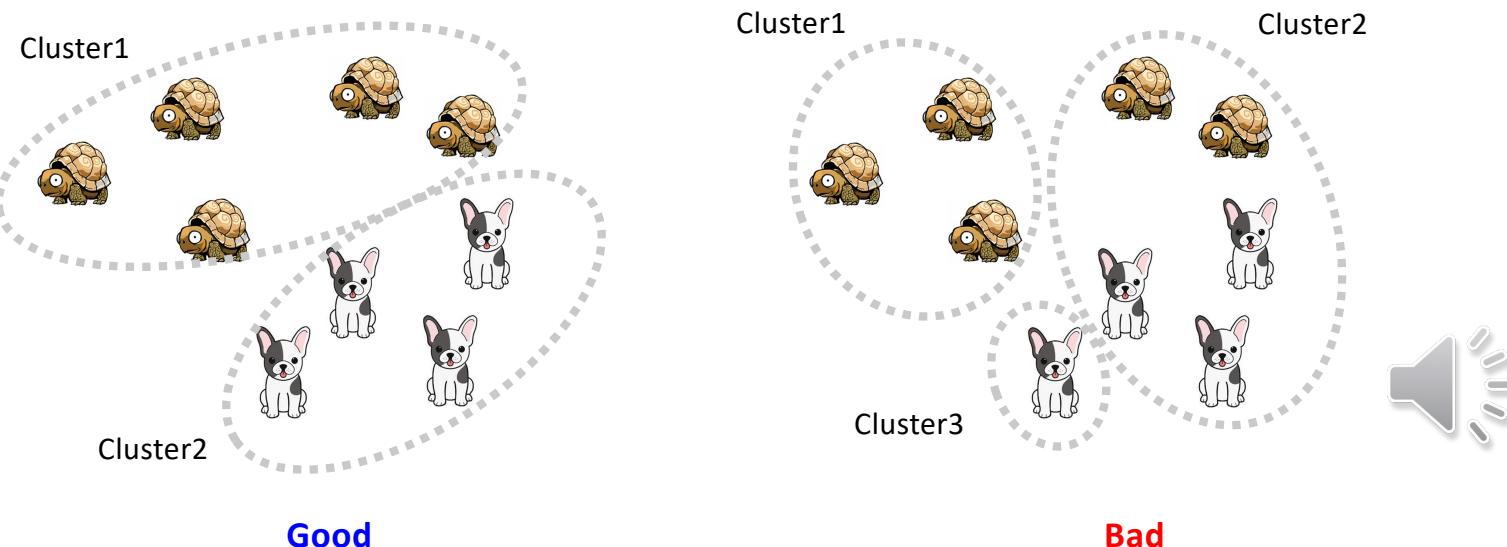
1) 목적변수가 존재할 경우

- Completeness score (완성도 점수)

$$H(K|C) : nc_k / nc$$

nc_k : 클러스터 k 내 객체가 c 로 라벨링된 샘플 수
 nc : c 로 라벨링된 총 샘플 수

$$c = 1 - \frac{H(K|C)}{H(K)}$$



Evaluation Metric (Clustering)

■ 군집화 (Clustering) 타당성 평가

1) 목적변수가 존재할 경우

- **V measure (= Normalised Mutual Information (NMI))**
 - Homogeneity score (동질성)과 Completeness score (완전성)의 조화평균
 - 실제로 동질성 h 와 완전성 c 가 모두 최대화되어야 함(h 와 c 가 모두 1일 때 NMI는 1)
 - 또한 클러스터링이 두 조건 중 어느 하나도 만족하지 않으면 NMI는 0이됨
 - Score: 0.0에서 1.0까지, (큰 값이 좋음)

$$NMI = 2 * \frac{h * c}{h + c}$$



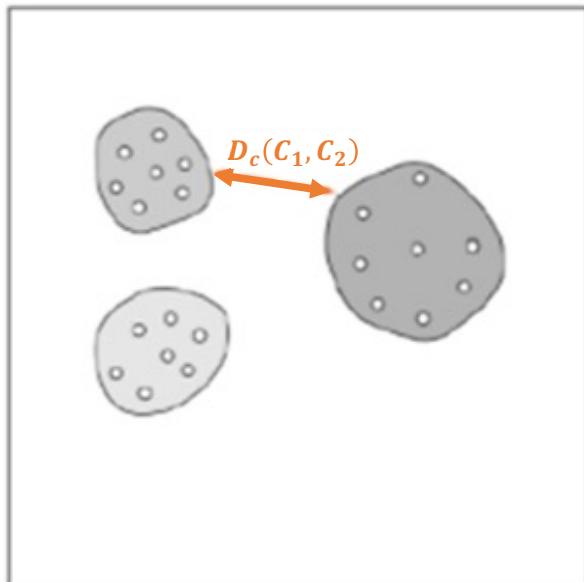
Evaluation Metric (Clustering)

■ 군집화 (Clustering) 타당성 평가

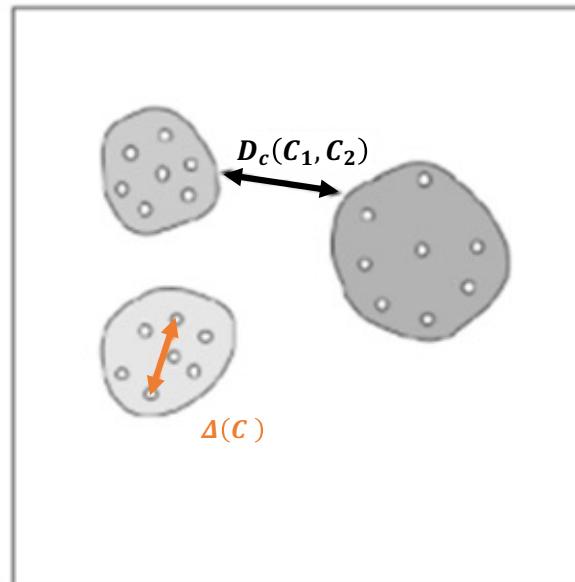
2) 목적변수가 존재하지 않을 경우

- 군집을 만든 결과가 얼마나 유용한지 검증
- **군집 간 거리, 군집의 지름, 군집의 분산 등을 고려**

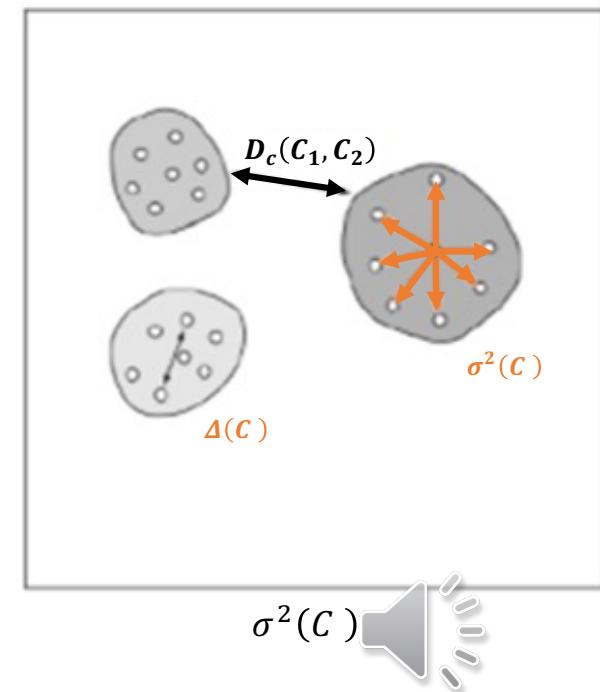
Distance between two clusters



Diameter of a cluster



Distribution within a cluster



- ✓ 평균 (μ): 주어진 수의 합을 측정개수로 나눈 값
- ✓ 분산 (σ^2): 변량들이 퍼져있는 정도, 분산이 크면 들죽날죽 불안정하다는 의미
- ✓ 표준편차: 분산은 수치가 너무 커서, 제곱근으로 적당하게 줄인 값

Evaluation Metric (Clustering)

■ 군집화 (Clustering) 타당성 평가

2) 목적변수가 존재하지 않을 경우

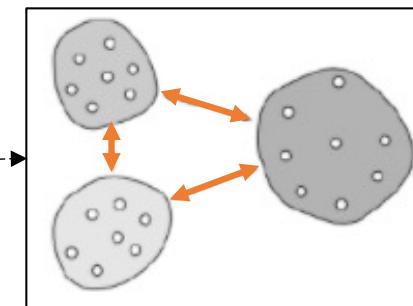
- **Dunn Index (DI)**

- 분자: 군집 간 거리의 **최소값**
- 분모: 군집 내 요소 간 거리의 **최대값**

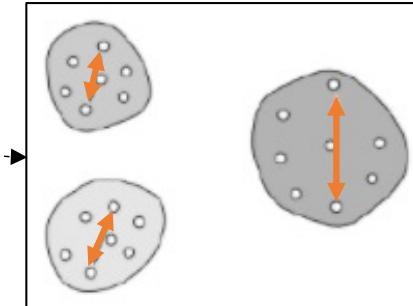
$$I(C) = \frac{\min_{i \neq j} \{d_c(C_i, C_j)\}}{\max_{1 \leq l \leq k} \{\Delta(C_l)\}}$$

- 분자의 min 값 >>... >> 분모의 max 값
- $I(C)$ 가 높은 값: 군집이 잘 되었음

$$D_c(C_i, C_j)$$



$$\Delta(C_l)$$



- 군집 간 거리는 멀수록, 군집 내 분산은 작을 수록 좋은 군집화

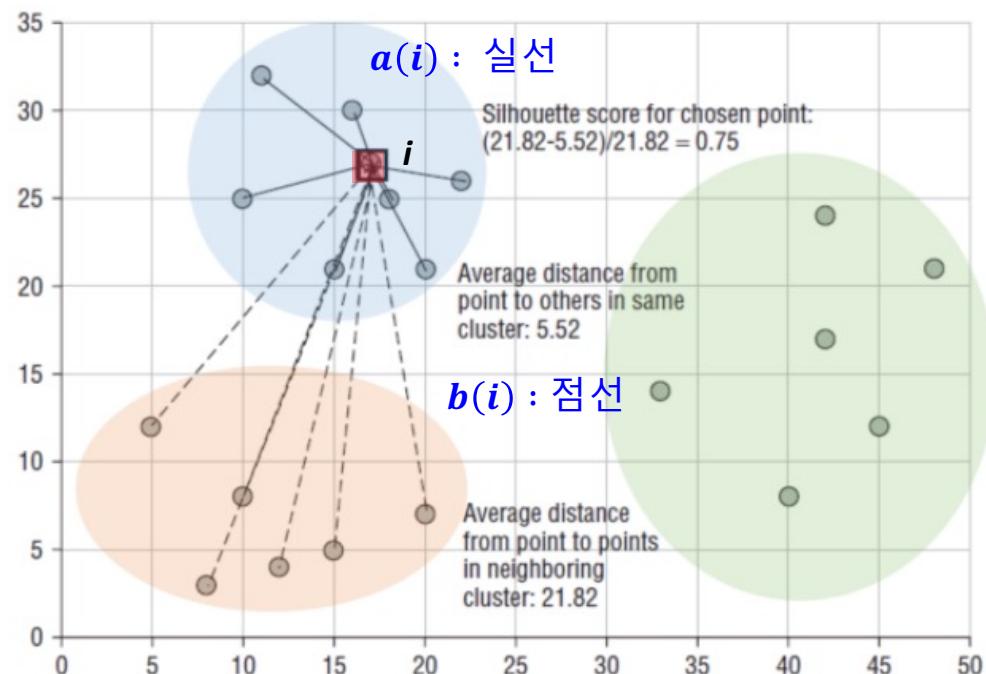
Evaluation Metric (Clustering)

■ 군집화 (Clustering) 타당성 평가

2) 목적변수가 존재하지 않을 경우

- **Silhouette (실루엣 지표)**

- $a(i)$: i번째 개체와 같은 파란색 군집에 속한 요소들 간 거리들의 평균
- $b(i)$: i번째 개체와 다른 오렌지색 군집에 속한 요소들 간 거리들의 평균



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



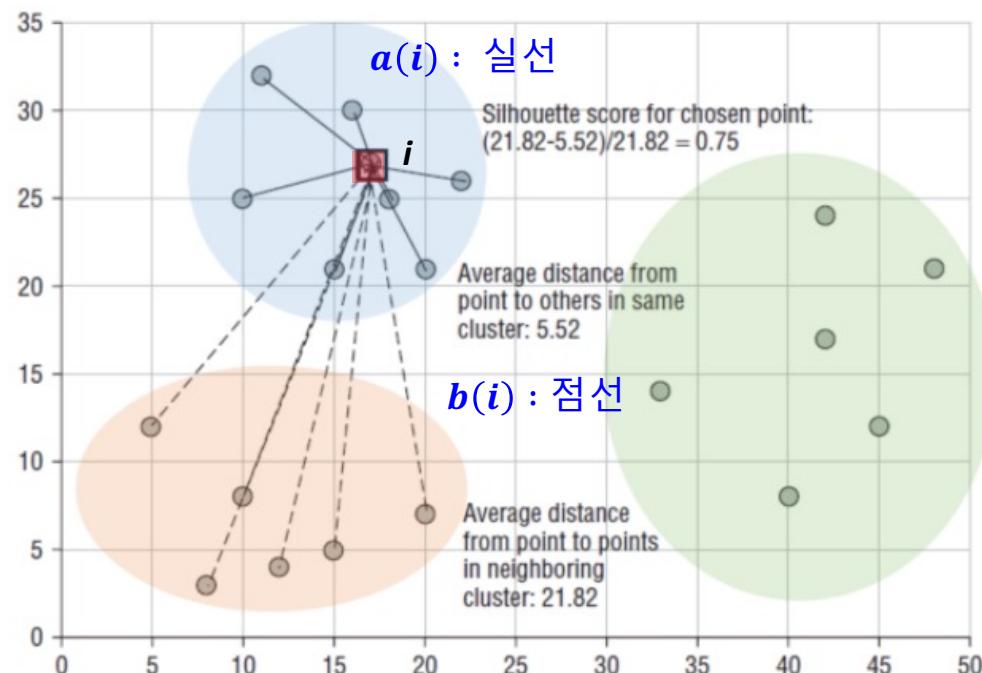
Evaluation Metric (Clustering)

■ 군집화 (Clustering) 타당성 평가

2) 목적변수가 존재하지 않을 경우

- **Silhouette (실루엣 지표)**

- $a(i)$: i번째 개체와 같은 파란색 군집에 속한 요소들 간 거리들의 평균
- $b(i)$: i번째 개체와 다른 오렌지색 군집에 속한 요소들 간 거리들의 평균



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

>> 가장 이상적인 경우

- $a(i) = 0$, 실루엣 지표 = 1

>> 최악의 경우

- $b(i)$ 가 0, 실루엣 지표 = -1
- 실루엣 지표가 0.5 보다 크면 군집 결과가 타당하다고 평가

Evaluation Metric (Clustering)

■ 군집타당성지표

□ 목적 변수가 있을 경우

- Homogeneity score (동질성, 균질성 점수)
- Completeness score (완성도 점수)
- V measure

□ 목적 변수가 없을 경우

- Dunn Index (DI)
- Silhouette Index

■ 유사도 측정 (거리 측정)을 위한 지표

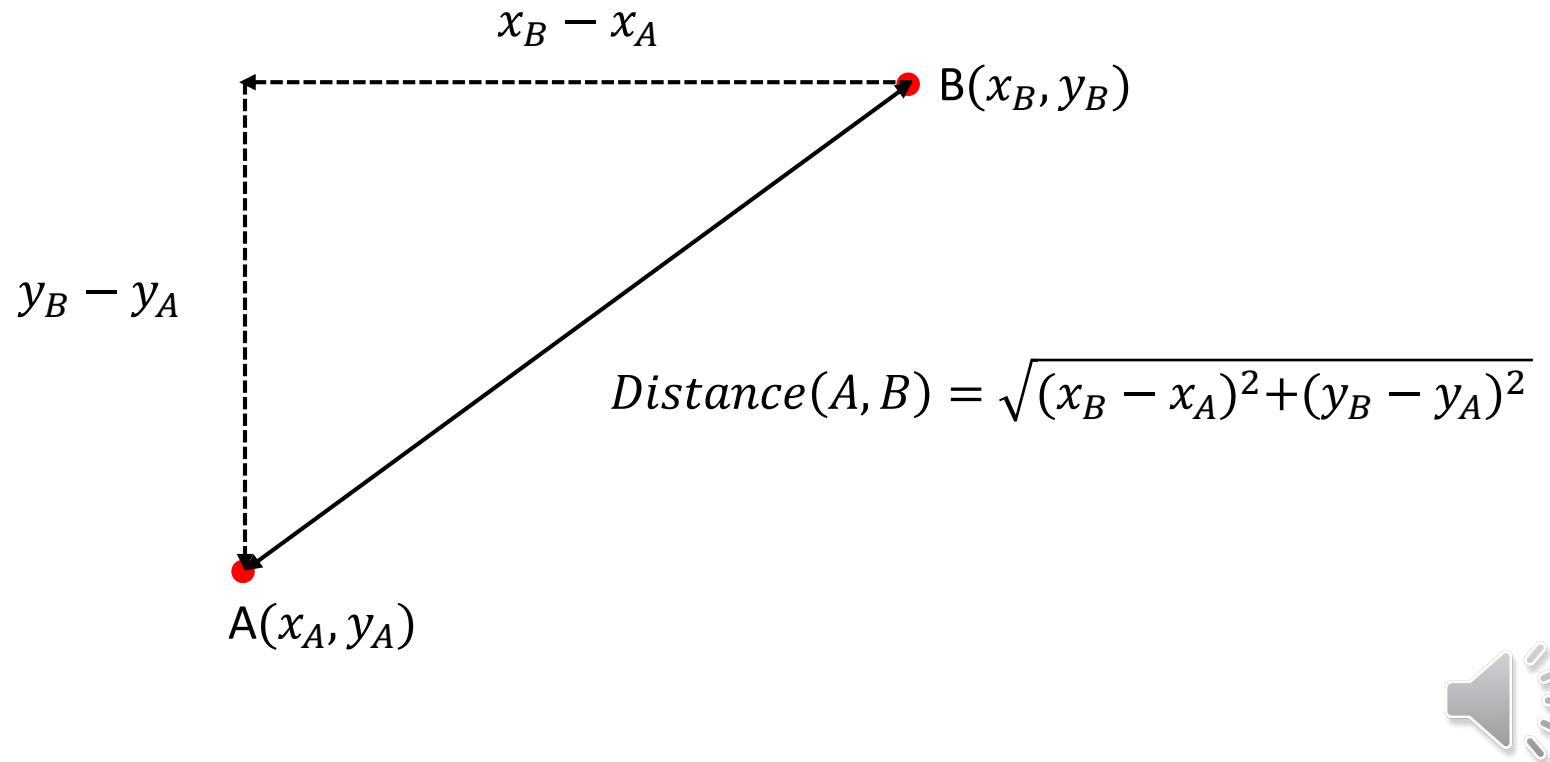
- 유클리디언 거리(Euclidean distance)
- 맨하튼 거리 (Manhattan Distance)
- 마할라노비스 거리(Mahalanobis distance)
- 코사인 유사도(Cosine similarity)
- 자카드 유사도(Jaccard similarity)



Distance Metrics

■ 유클리디언 거리 (Euclidean distance)

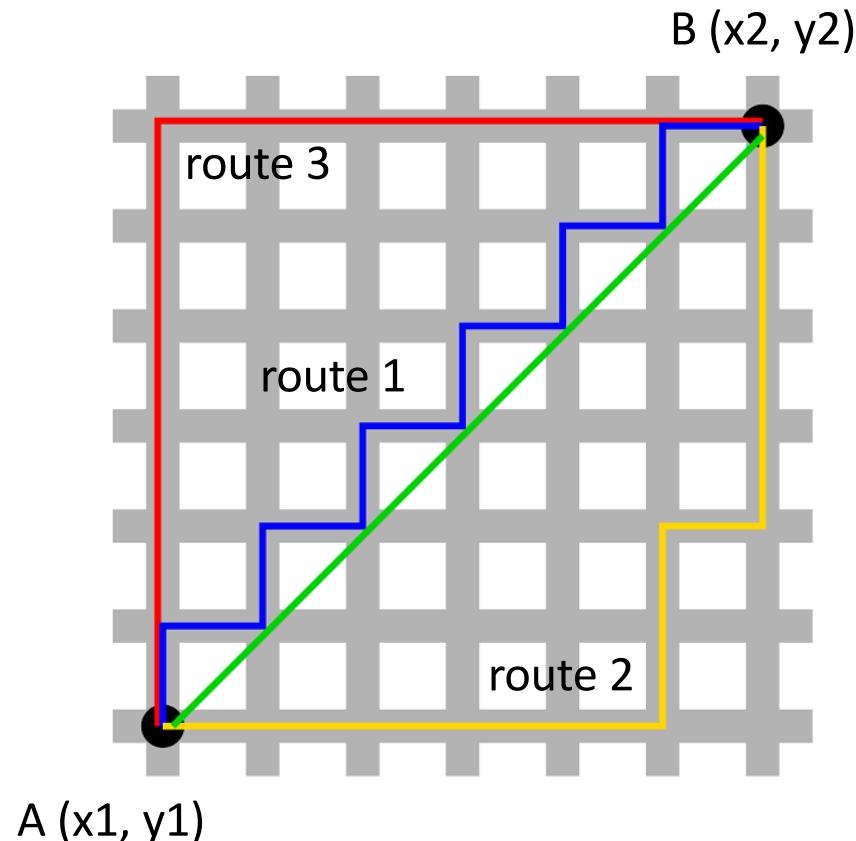
- 유클리드 공간에서 두 점 사이의 유클리드 거리는 두 점 사이의 선분의 길이
- 측정된 거리 값이 0에 가까울수록 클러스터의 유사성이 높음



Distance Metrics

■ 맨하튼 거리 (Manhattan Distance)

- 대각선으로 거리를 측정할 수 없는 상황의 데이터일 때, 좌표 간의 거리를 구하는 방식



$$\text{Manhattan Distance} = |x_1 - x_2| + |y_1 - y_2|$$

$$D(A, B) = \sum_{i=1}^n |a_i - b_i|$$

- 최단거리 (유클리디언 거리)
- route 1
 - 느낌상으로는 초록색 직선거리 다음으로 파란색이 가장 가까운 거리?
- route 2
 - route 1 = route 2 = route 3



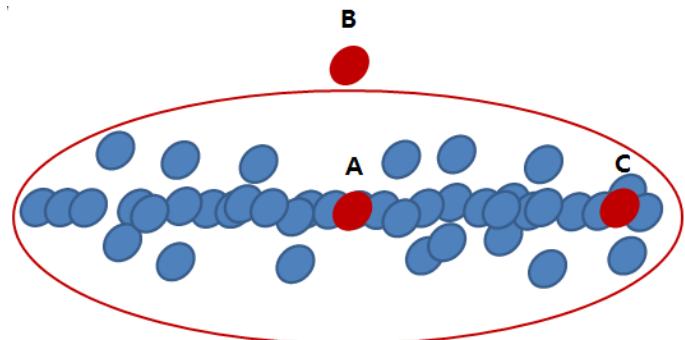
Distance Metrics

■ 마할라노비스 거리 (Mahalanobis distance)

- 유clidean 거리로 측정하기 힘든 데이터, 공분산(상관관계)이 존재하는 데이터 간의 거리 측정 방법
- 유clidean 거리상으로는 A는 B와 가깝지만, 확률분포/분산을 고려했을 경우, B보다는 C에 더 가까울 수 있음

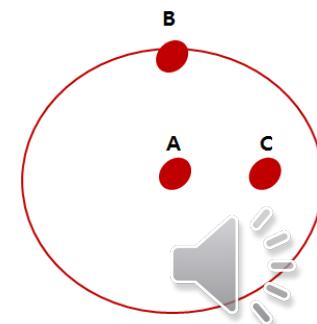
$$Distance(A, B) = \sqrt{(A - B) \sum^{-1} (A - B)^T}$$

공분산: 두 변수 사이의 관계가 존재
 $Cov(X, Y) > 0$ X 가 증가 할 때 Y 도 증가
 $Cov(X, Y) < 0$ X 가 증가 할 때 Y 는 감소
 $Cov(X, Y) = 0$ 두 변수간에는 관계가 없음



화이트닝 변환
(Whitening transform)

공분산 행렬이 단위 행렬 I가 되도록
원래 벡터 x 를 y 로 변환

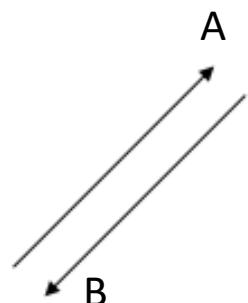


유clidean 거리로
거리 측정 가능

Distance Metrics

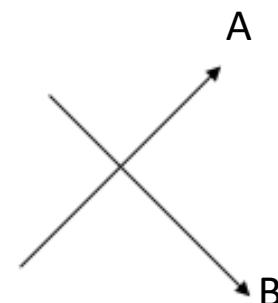
■ 코사인 유사도 (Cosine similarity)

- 두 벡터 간의 코사인 각도를 측정하여 두 벡터의 유사도 파악
- 코사인 유사도 값: -1 이상 1 이하의 값
- 값이 1에 가까울수록 유사도가 높음
- 텍스트マイ닝에서도 사용 (검색어 기본 랭킹, 문서 유사도)



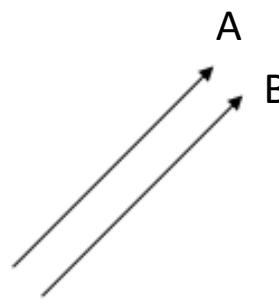
코사인 유사도 : -1

각도가 180°로써
화살의 방향이 반대일
경우



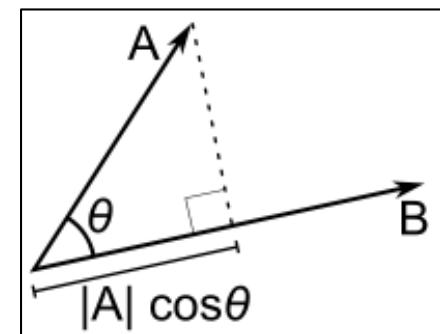
코사인 유사도 : 0

각도가 90°로써
화살의 방향이
서로 각을 이루는 경우



코사인 유사도 : 1

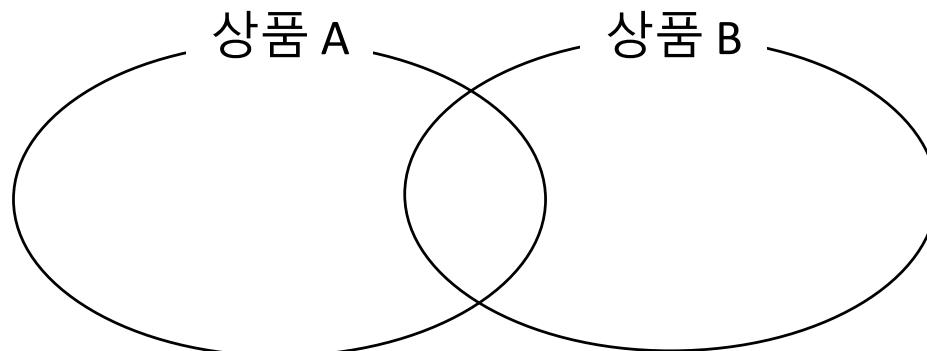
화살 각도의 방향이
완전히 동일한 경우



Distance Metrics

■ 자카드 유사도 (Jaccard similarity)

- 두 집합 간의 교집합 크기를 이용하여 유사도를 측정하는 방법
- 두 집합 사이의 교집합의 크기가 클수록, 소비자의 구매 선호도 측면에서 유사도가 높음



$$Jaccard\ Similarity = \frac{A \cap B}{A \cup B}$$

A = {A 상품을 구매한 소비자}
B = {B 상품을 구매한 소비자}

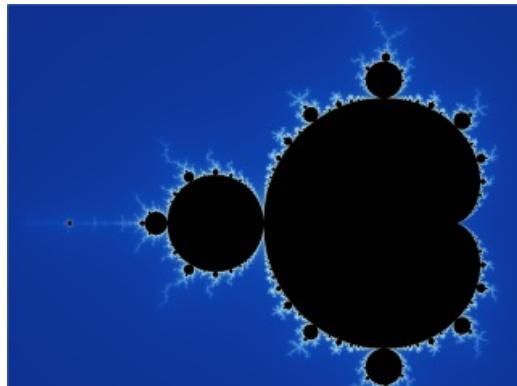


Appendix

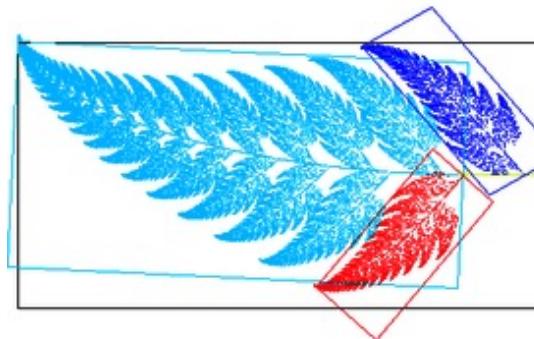
■ Self-similarity

□ 자기유사성

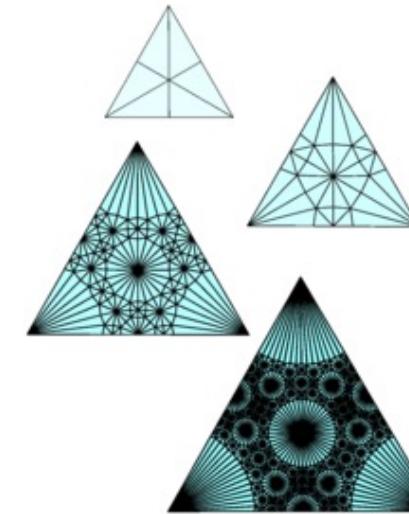
- 부분을 확대할 때 자신을 포함한 전체와 닮은 모습을 보여주는 성질
- 물체의 일부가 여러 측면에서 같은 통계적 특성을 나타냄
- 자기 유사성은 인공적인 프랙탈의 전형적인 특성



망델브로 집합



아핀 자기유사성을
나타내는 양치류



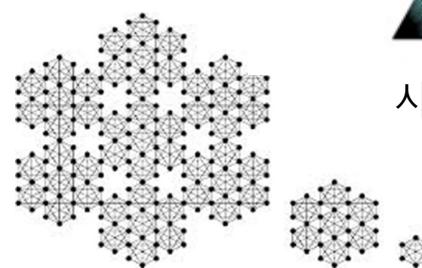
시에르핀스키 카펫



로마네스코 브로콜리



코흐곡선



Evaluation Metric

■ 시간 복잡도 (Time Complexity)

- 명령어 실행 횟수나 명령어의 실제 실행시간을 계산
- 시간복잡도를 측정하여 알고리즘이 효율적인지 분석

1) 알고리즘을 구성하는 명령어의 실행 횟수

- Big-O 표기법

2) 알고리즘을 구성하는 명령어의 실제 실행시간

- 실제 실행시간 측정
- 각 명령어의 실행시간은 특정 하드웨어 혹은 프로그래밍 언어에 따라 값이 달라질 수 있음



Evaluation Metric

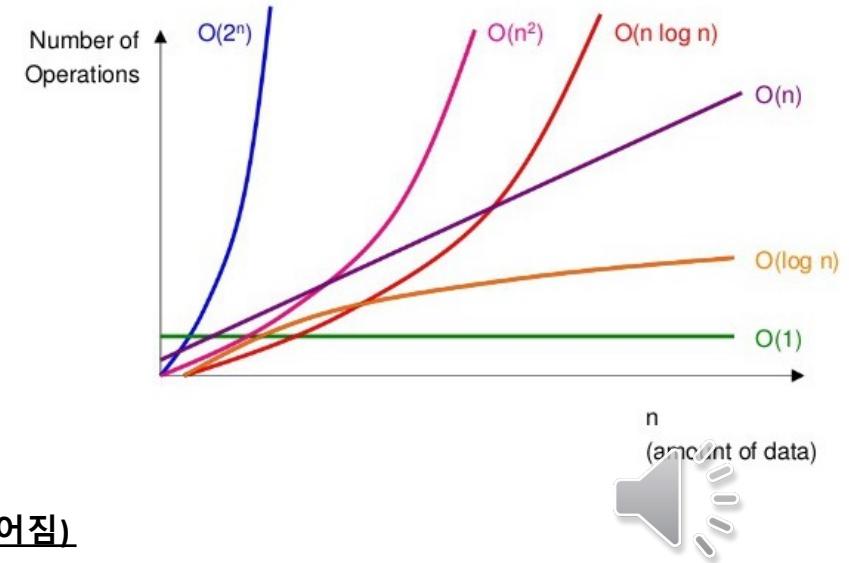
■ 시간 복잡도 (Time Complexity)

□ Big-O 표기법

- 상대적으로 불필요한 연산을 제거하여 알고리즘의 분석을 조금 더 간편하게 할 목적으로 시간 복잡도를 표기하는 방법

$$f(n) = 5 \text{ 인 함수} \rightarrow O(1) \quad f(n) = 2n + 1 \text{ 인 함수} \rightarrow O(n) \quad f(n) = 3n^2 + 100 \text{ 인 함수} \rightarrow O(n^2)$$

빅오 표기법	1	4	8	32
$O(1)$	1	1	1	1
$O(\log n)$	0	2	3	5
$O(n)$	1	4	8	32
$O(n \log n)$	2	8	24	160
$O(n^2)$	1	16	64	1,024
$O(n^3)$	1	64	512	32,768
$O(2^n)$	2	16	256	4,294,967,296
$O(n!)$	1	24	$40,326$	$26,313 \times 10^{33}$



상수함수 < 로그함수 < 선형함수 < 다항함수 < 지수함수 (효율성이 떨어짐)

Evaluation Metric

■ 시간 복잡도 (Time Complexity)

□ Big-O 표기법

- $T_1(n)$: a polynomial for the anomaly detection module
- $T_2(n)$: a polynomial for the survival rate operation module

Table 3

Time complexity of the detection algorithm for the HYUNDAI YF Sonata vehicle.

	$T_1(n)$	$T_2(n)$	Big O	Total steps
An individual CAN ID	$2(n^2) + 3$	$3(n^3) + 2n + 8$	$O(n^3)$	1
All CAN IDs	$2(n^2) + 3$	$3(n^3) + 2n + 8$	$O(n^3)$	19,683
CAN IDs with a short cycle	$2(n^2) + 3$	$3(n^3) + 2n + 8$	$O(n^3)$	9261



Evaluation Metric

■ 시간 복잡도 (Time Complexity)

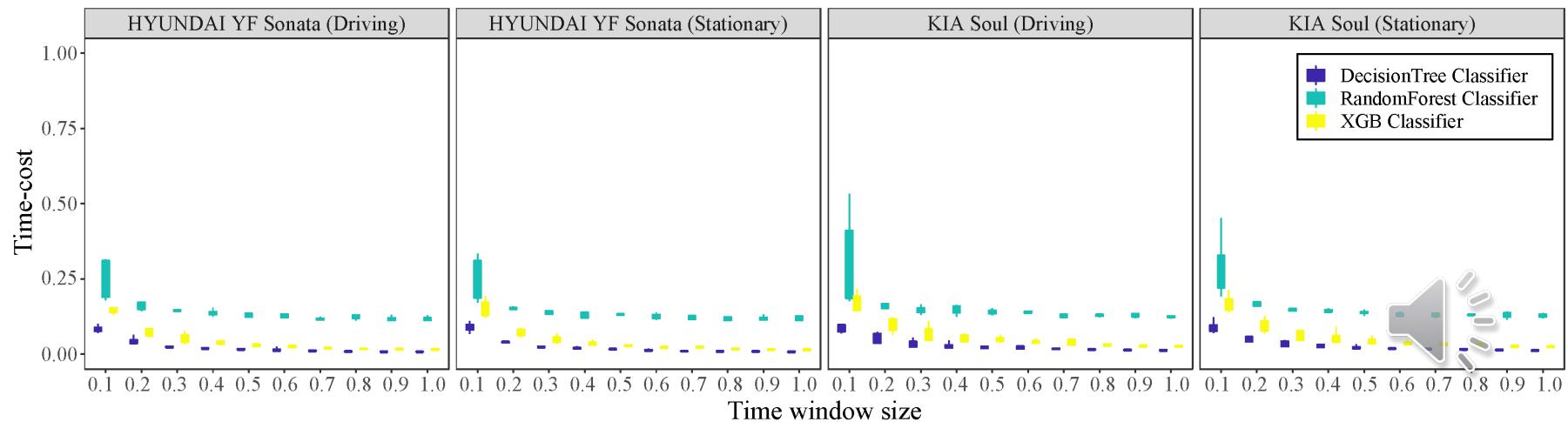
- 실제 실행시간 (RunTime)

```
begin_Time = time.time()
```

••• ## 분류 알고리즘의 학습 or 테스트 시 수행 시간 측정

```
finish_Time = time.time()
```

```
RunTime = abs(float(finish_Time) - float(begin_Time))
```



Thank you

