Unsupervised Learning

2022. 08. 11.

7기 김경한

목차

0. Intro

- 지도학습 vs 비지도학습
- 고윳값 분해
- 특잇값 분해
- MNIST 손글씨 data set
- Network Analysis

1. 차원 축소

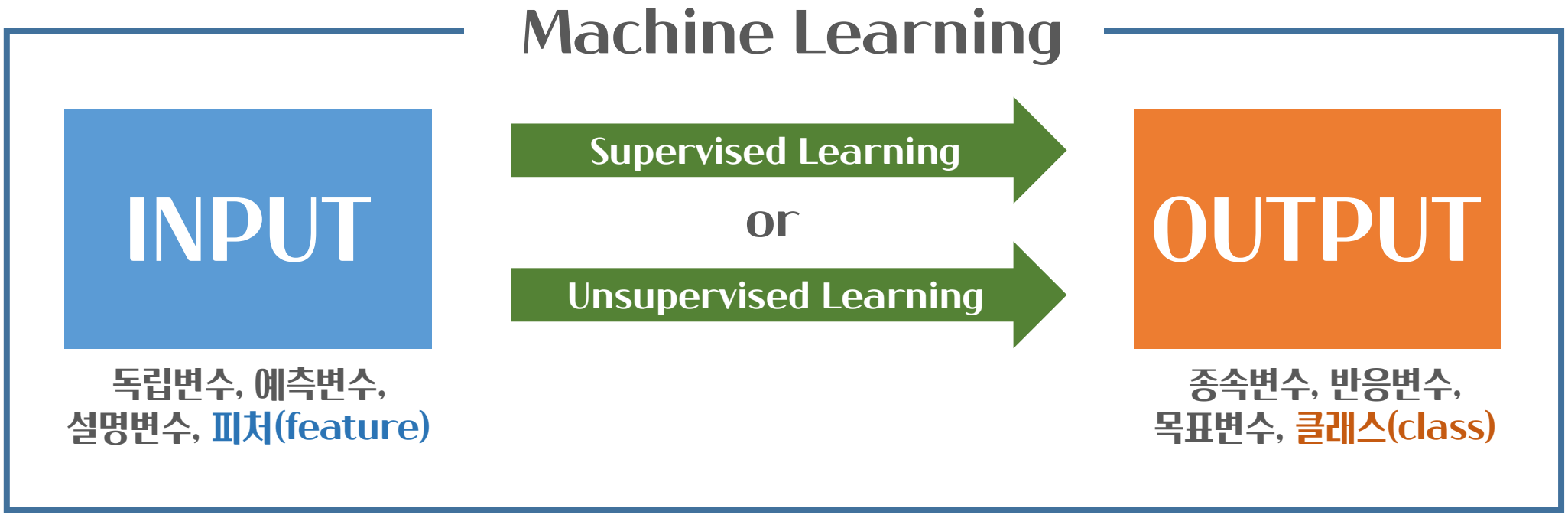
- 주성분 분석 (PCA)
- 특잇값 분해 (SVD)
- 랜덤 투영 (RP)
- IsoMap

2. 군집화

- K-means clustering
- Hierarchical clustering

→ 8/16 세션!

0-1. 지도학습 vs 비지도학습



기계학습 (Machine Learning)의 목표는,

Input (입력) 을 바탕으로 올바른 Output (출력) 을 만들어내는 것!

0-1. 지도학습 vs 비지도학습

지도학습과 비지도학습의 결정적인 차이: Label의 유무

Label: 학습이 정확할 경우 기대되는 Output (= 정답지/답안지!)

Ex]

지도학습 (채점 가능)

AI: 이거 개임 고양이임?
Human: 아 이거 고양이임! (Label)

비지도학습 (채점 불가능)

AI: 이거 무슨 사진이었음?
Human: 나도 모르니까 알아서 해.
(No Label)

0-1. 지도학습 vs 비지도학습

지도학습의 장점!

(1) Label된 Data가 많이 확보된 경우 성능이 탁월함

(2) Label이 명확하게 정의된 작업에서 성능이 탁월함

(3) Model의 성능을 정확히 측정하기 용이함

지도학습의 단점!

(1) 비용 소모가 심함 (Labeling 작업은 인력이 필요)

(2) 사람이 정해 주는 Label 이외의 항목을 새롭게 찾아 내거나, 문제를 확장 / 일반화하는 작업은 사실상 불가능함

지도학습의 예시: 회귀분석 (Linear Regression), k-최근접 이웃 (k-nearest Neighborhood, k-NN), 의사결정트리 (Decision Tree), 랜덤 포레스트 (Random Forest), 서포트벡터머신 (Support Vector Machine, SVM), 신경망 (Neural Network) 등

0-1. 지도학습 vs 비지도학습

비지도학습의 장점!

(1) 인간이 미처 몰랐던
분류/패턴을 찾아낼 수도 있음

(2) Label이 시간에 따라 변하는
task의 경우, 문제에 더 유연하게
접근할 수 있음

(3) Label이 충분히 확보되지 않은
data에도 활용 가능함 ~ 비용 절감!

비지도학습의 단점!

(1) Model의 성능을 정확히
측정하는 것이 까다로움

(2) 학습의 결과물이
"아주 직관적"이지는 못함

비지도학습의 예시: 주성분 분석 (Principal Component Analysis, PCA),
특잇값 분해 (Singular Value Decomposition, SVD), 오토인코더 (Autoencoder),
매니폴드 학습 (IsoMap, t-SNE 등), 군집화 (Clustering), GAN 등

0-1. 지도학습 ♥ 비지도학습

"비지도학습은 아직까지 지도학습만큼 많은 성공을 거두지는 못했지만 **그 잠재력은 엄청납니다.**
현실 세계 데이터 대부분은 레이블이 없습니다.
지도학습이 이미 해결한 과제보다 **더 큰 규모의 과제**에 머신러닝을 적용하기 위해서는
레이블이 있는 데이터와 레이블이 없는 데이터를 모두 사용해야 합니다.

비지도학습은 레이블이 없는 **데이터의 내재된 구조를 학습**해
숨겨진 패턴을 찾는 데 매우 유용합니다.
숨겨진 패턴이 발견되면 이 패턴을 유사한 패턴들과 함께 그룹화할 수 있습니다. (중략)

즉, **비지도학습은 지도학습 방법의 성공적인 적용을 가능케 합니다.**
비지도학습과 지도학습 사이의 시너지(=준지도학습)는
성공적인 머신러닝 응용 프로그램의 미래를 이끌 겁니다."

- 〈핸즈온 비지도 학습〉 p.428~429

0-2. 고윳값 분해 (Eigendecomposition)

$A = Q\Lambda Q^{-1}$ 형태로 행렬을 분해하는 방법

Λ : **고윳값**(Eigenvalue)을 대각성분으로 하는 대각 행렬

(고윳값: $\det(A - \lambda I) = 0$ 을 만족시키는 λ)

Q : 고윳값에 대응되는 **고유벡터**를 열벡터로 하여 합친 행렬

(고유벡터: $Ax - \lambda x = 0$ 을 만족시키는 벡터 x)

고윳값 분해를 알아야 특잇값 분해를 할 수 있다.
또한 고윳값 분해는 **정방 행렬 (정사각 행렬, Square Matrix) 에서만** 할 수 있다.

0-3. 특잇값 분해 (Singular Value Decomposition, SVD)

$A = U\Sigma V^T$ 형태로 행렬을 분해하는 방법

구체적으로는 $A^T A$ 를 고윳값 분해하여 구하게 된다

Σ: **고윳값**(Eigenvalue)의 양의 제곱근을 대각성분으로 하는 대각 행렬

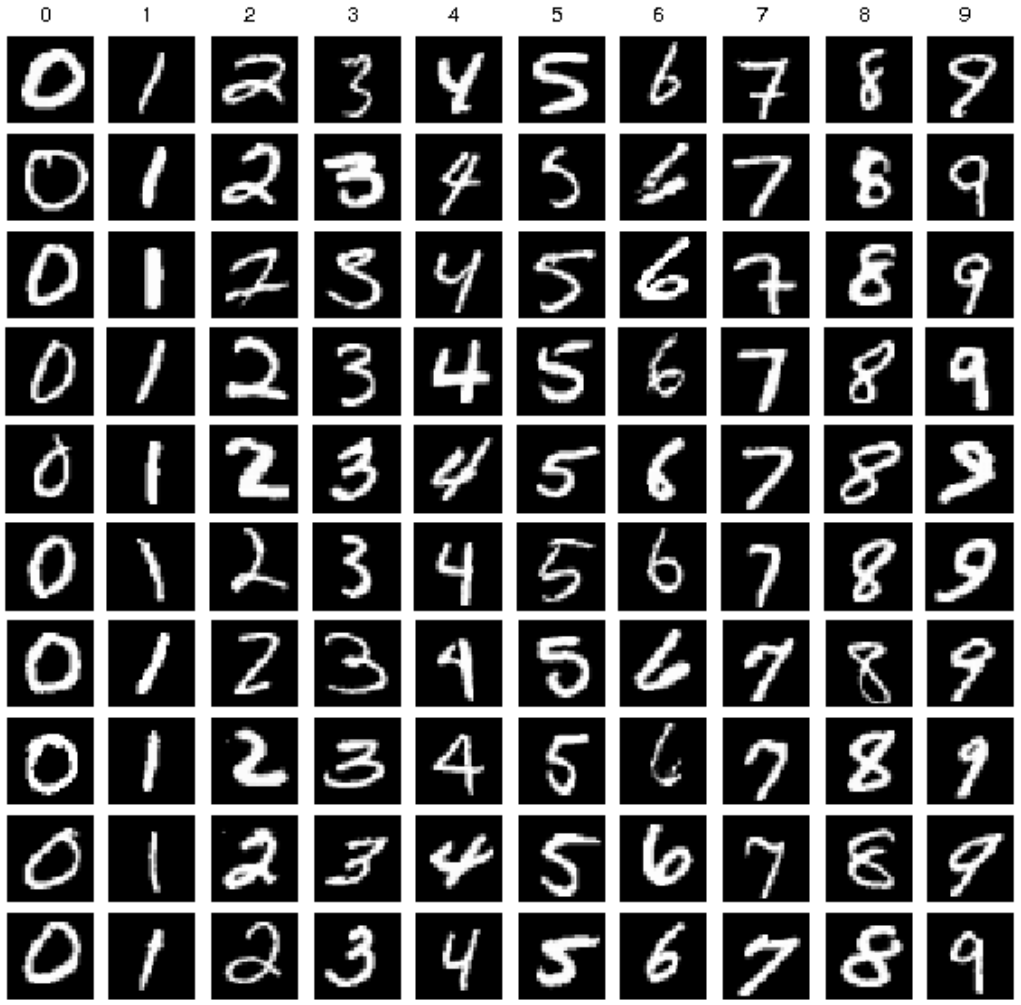
$(\Sigma_{ii}^2 = \Lambda_{ii})$

주의사항]

(1) Σ를 구성할 때, **큰 고윳값부터** 앞쪽 행/열에 써야 한다

(2) U와 V는 **Orthonormal**해야 한다 ~ 그래야 $\Sigma_{ii}^2 = \Lambda_{ii}$이 성립함

0-4. MNIST 손글씨 data set

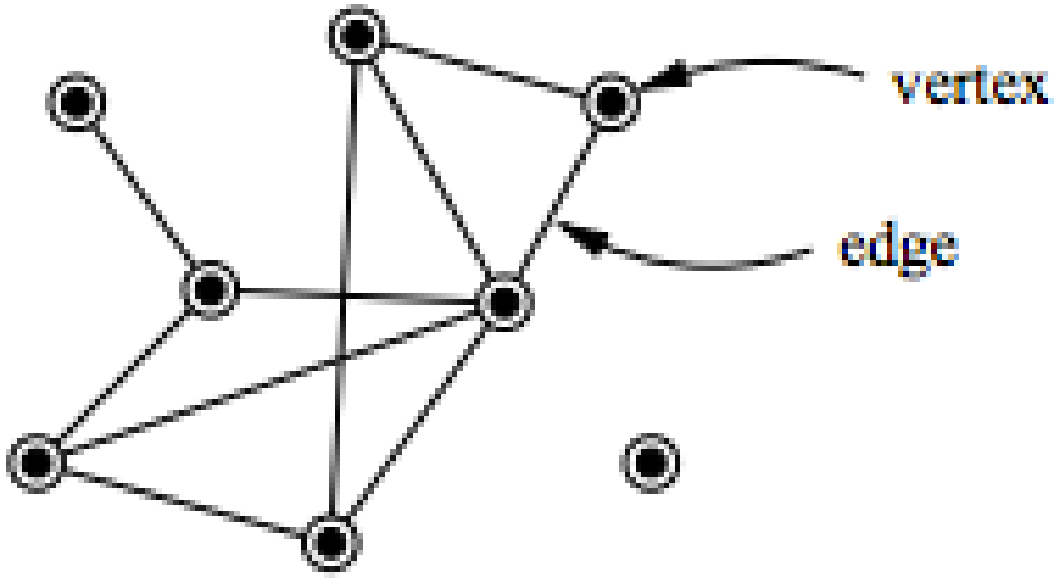


0부터 9까지의 숫자를
손으로 쓴 이미지 파일

가로 28px, 세로 28px 크기
총 70,000개의 파일로 구성
(Train: 60,000개, Test: 10,000개)

원래 데이터에는 사진 별로
Label이 대응돼 있지만,
오늘 세션에서는 학습 단계에서
이를 활용하지 않습니다!

0-5. Network Analysis



데이터를 그래프 상의 node와 edge로 나타내,
데이터 간의 연결 관계를 분석하는 학문

Node(Vertex): 각각의 data
Edge: data와 data를 연결하는 직선
Cycle: A에서 최소 두 점을 거쳐
다시 A로 이동하는 경로
Tree: cycle이 없는 그래프

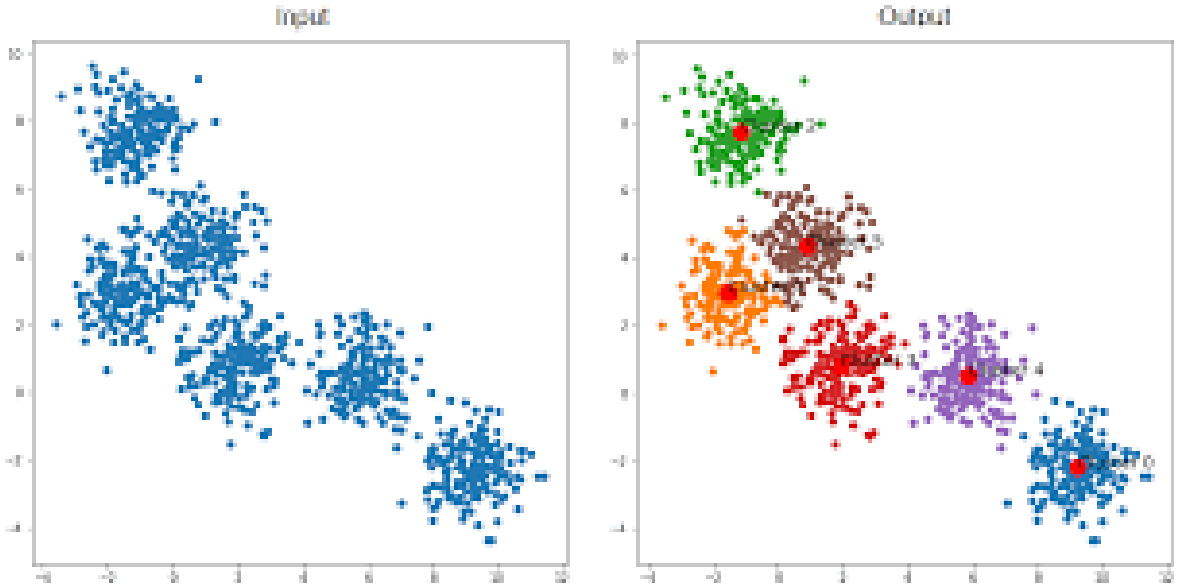
비선형 차원 축소 기법인 IsoMap과,
군집화 중 계층적 클러스터링에 활용됩니다!

비지도학습의 목적

비지도학습의 주된 목적은,

- **차원 축소 (Dimension Reduction)**
- **이상치 탐지** (차원 축소와 사용하는 기법이 유사)
- **군집화 (Clustering)**
- **데이터 보강** (너무 어려워용)

등입니다!



1. 차원 축소

차원 = 학습에 사용하는 데이터의 피처 수 (=독립변수의 개수, 좌표평면의 축의 개수)

Ex] MNIST 손글씨 데이터는 784차원!

차원의 저주를 해결하기 위해,
정보 손실은 최소화하면서 데이터의 차원을 낮추는 방법론

회귀분석의 변수 선택법과 유사합니다!
(변수 선택: 전체 N개의 독립변수(피처) 중 n개를 선택하고 나머지를 버리는 과정)

1. 차원 축소 – 차원의 저주

차원의 저주 (Dimension Curse)

: 데이터의 차원이 너무 높을 때 발생하는 **총체적 난국**을 의미함

데이터의 차원이 높을수록,
(1) 더 **많은 학습용 데이터**가 필요함
(2) 연산량이 늘어나 **학습이 오래 걸림**
(3) 시각화가 어려워 **직관적이지 못함** (4차원 이상의 경우)

그러나,
데이터의 차원이 **너무 낮아지면**
데이터를 **구분하기 힘들어짐**

1-1. 주성분 분석 (Principal Component Analysis, PCA)

"Frankenstein"식 차원 축소 / 변수 선택 기법

주성분: 원래 있던 성분(=feature)들의 선형결합으로 만들어 낸 새로운 성분

PCA를 수행하면 원래 feature 개수만큼의 주성분이 새롭게 만들어지고,
이 주성분 중 일부만을 사용해 차원을 축소합니다.
단, 주성분은 원래 feature들의 본질적 의미를 상실한다는 단점이 있습니다.

PCA 는 수학적으로,
Data Covariance Matrix($\frac{1}{n}\sum_{i=1}^{n} XX^T$)를 고윳값 분해해 수행합니다.

주성분 = 각각의 고윳값에 대응되는 고유 벡터

1-1-0. 고윳값, 정보, 엔트로피, 그리고 PCA

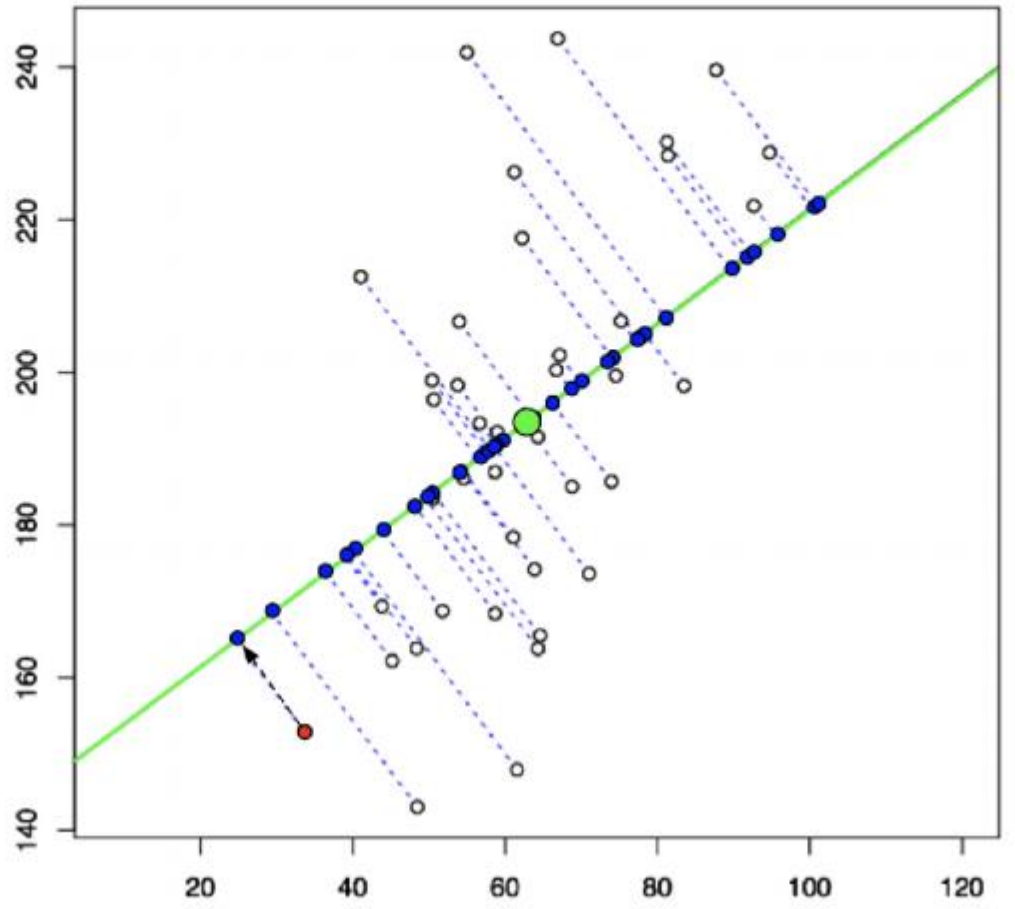
"PCA를 수행하면 원래 feature 개수만큼의 주성분이 새롭게 만들어지고,
이 주성분 중 일부만을 사용해 차원을 축소합니다."
= 고윳값이 큰 고유벡터(주성분)부터 순위를 매겨 사용!

Why?

: 차원 축소 시 정보 손실을 최소화하기 위해
= 초평면 상에서 적은 차원으로도 data의 변별력을 유지하기 위해
= 초평면 상에서 data들을 최대한 떨어트려 놓기 위해
= data의 분산을 최대한 크게 만들기 위해 (Maximum Variance Approach)
+ PCA에서 주성분 별 분산은 주성분(고유벡터)에 대응되는 고윳값이기 때문에!

1-1-0. 고윳값, 정보, 엔트로피, 그리고 PCA

- PCA is a dimensionality reduction algorithm that maximizes the variance in the low-dimensional representation of the data to retain as much information as possible.



1-1-0. 고윳값, 정보, 엔트로피, 그리고 PCA

정보량을 나타내는 함수 $I(X)$는 아래와 같이 정의됩니다.

$$I(X) = -logP(X)$$

정보량은,

1) 확률에 대한 함수여야 하며,
2) 항상 발생하는 사건은 정보량이 0이고,
   희귀한 사건일수록 정보량이 많다는 것을 반영할 수 있어야 하며,
3) $I(X+Y) = I(X) + I(Y)$ 여야 하기 때문에

로그함수를 활용해 정의됩니다.

1-1-0. 고윳값, 정보, 엔트로피, 그리고 PCA

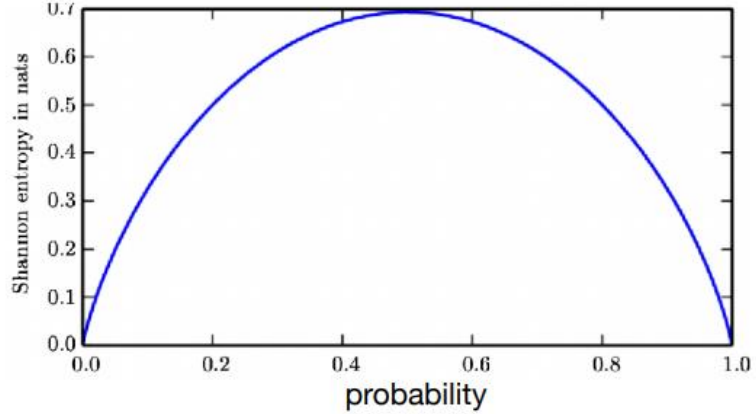
엔트로피는 정보량의 평균으로 정의되기 때문에,
그 함수인 $H(X)$는 아래와 같이 정의됩니다.
(Shannon Entropy)

$$H(X) = E[I(X)] = E[-logP(X)] = -\sum P(X)logP(X)$$

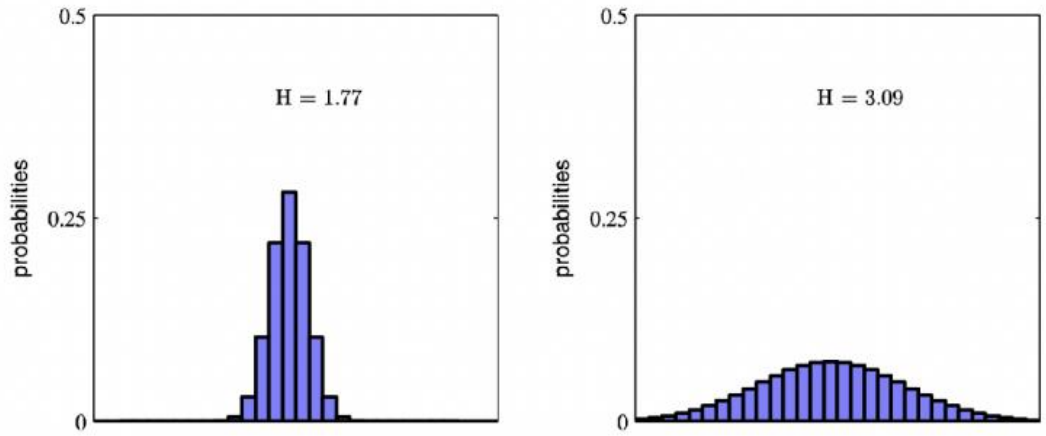
1-1-0. 고윳값, 정보, 엔트로피, 그리고 PCA

Entropy Examples

Example: Shannon entropy of a binary random variable

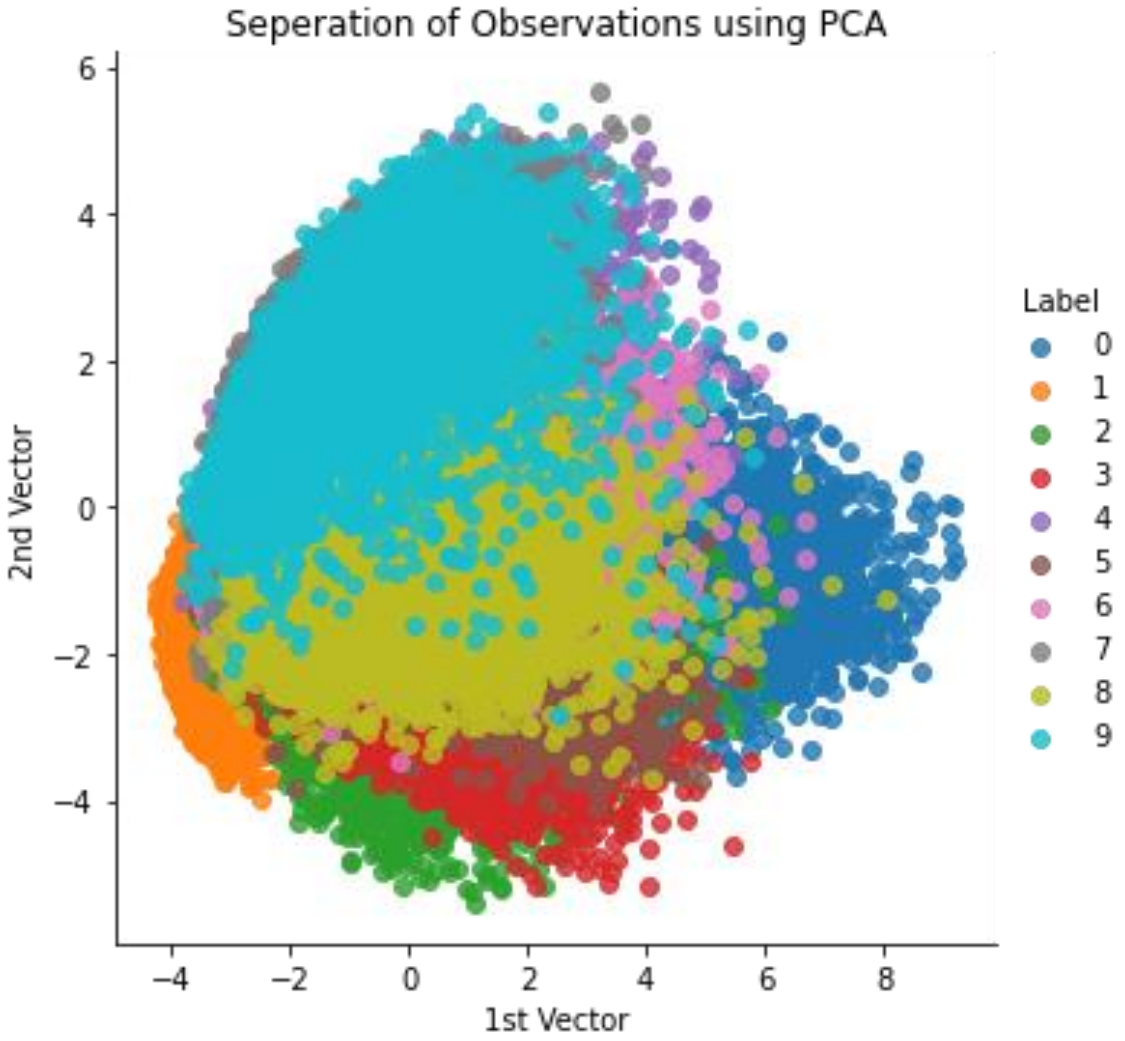


Example: Histograms of two probability distributions over 30 bins.



분산이 클수록, 정보량/엔트로피 또한 커집니다!

1-1-1. 일반 PCA

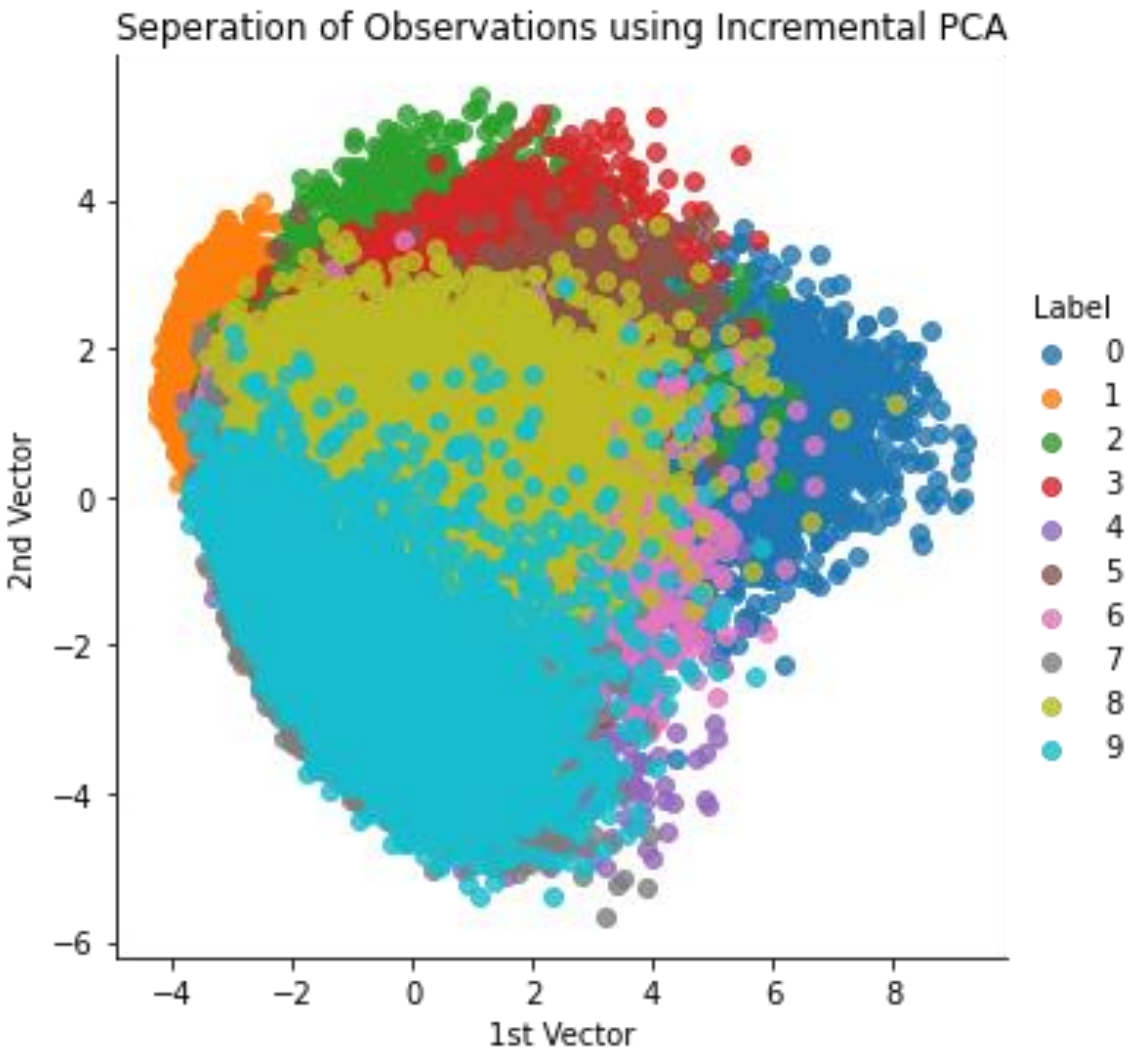


Scikit-learn 패키지를 이용해
MNIST data에 **일반 PCR**를 적용한 결과

0~9까지의 숫자들이
나름대로 잘 구분되어 있음을 볼 수 있다!

(**784차원 ➡ 2차원**으로의 시각화임을
감안해 주세요!)

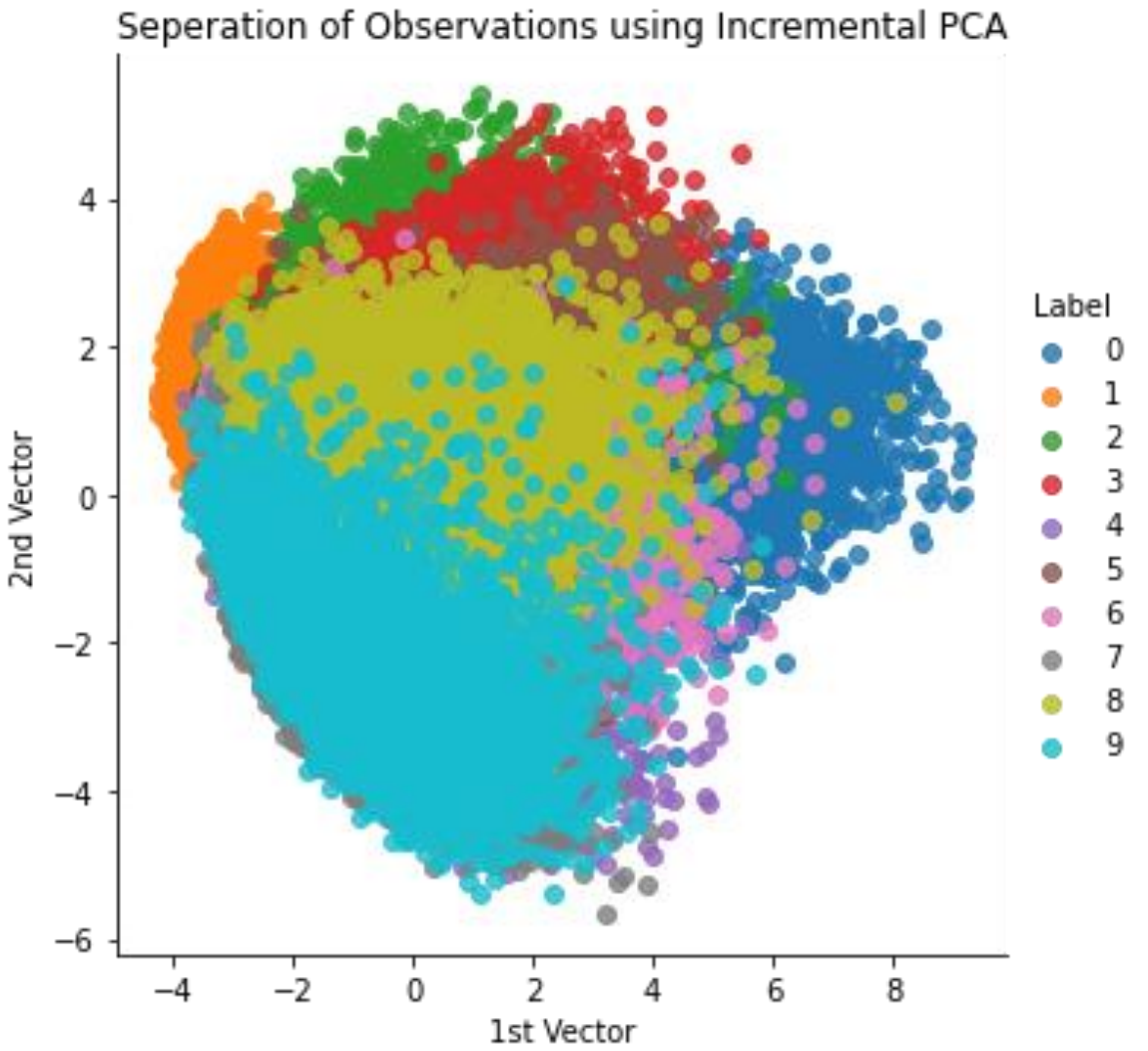
1-1-2. 점진적 PCA

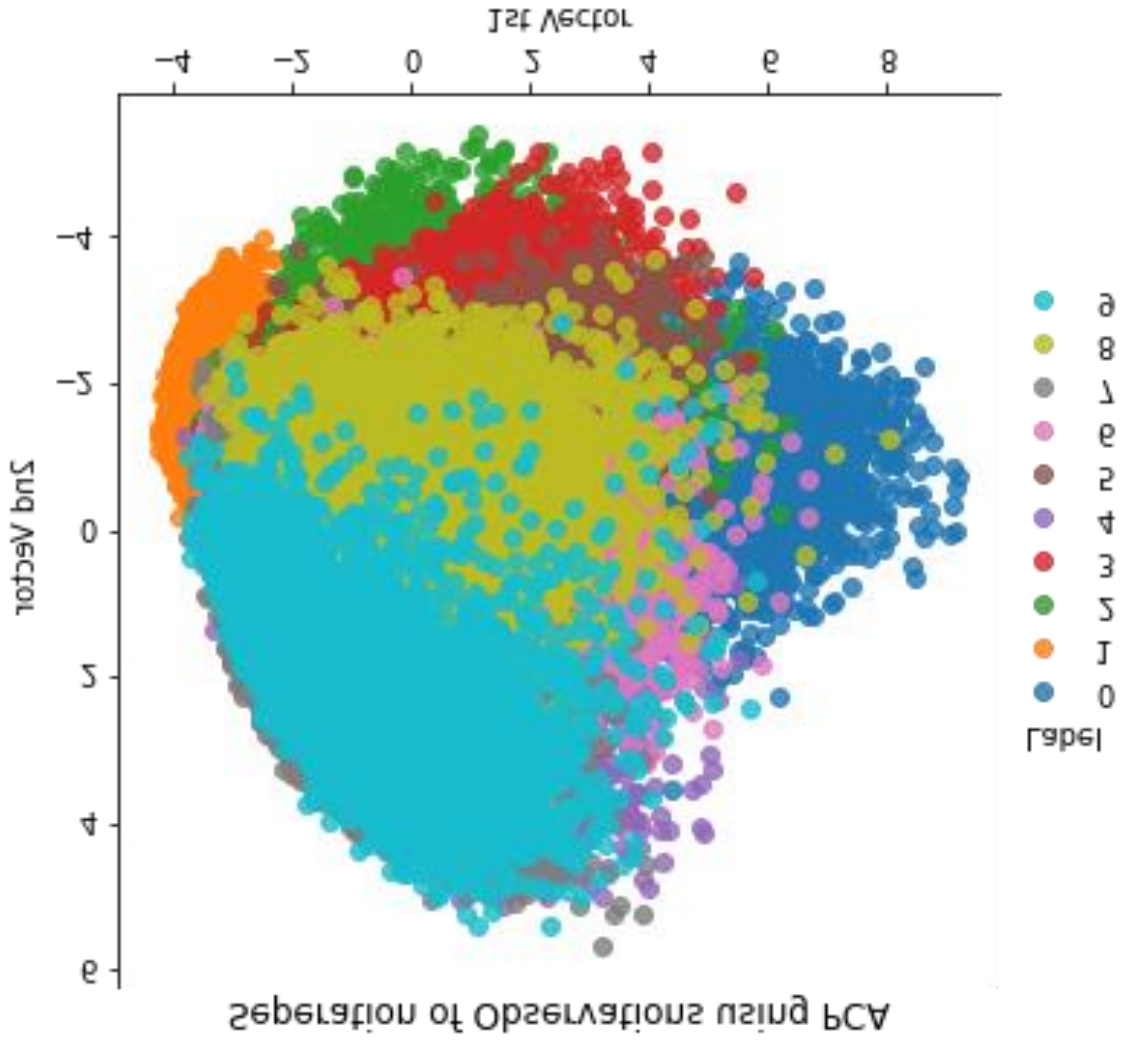


Dataset의 크기가 너무 클 경우,
메모리에 저장할 수 있을 정도의 크기로
배치(batch)를 설정해서
PCA를 점진적으로 수행하는 기법

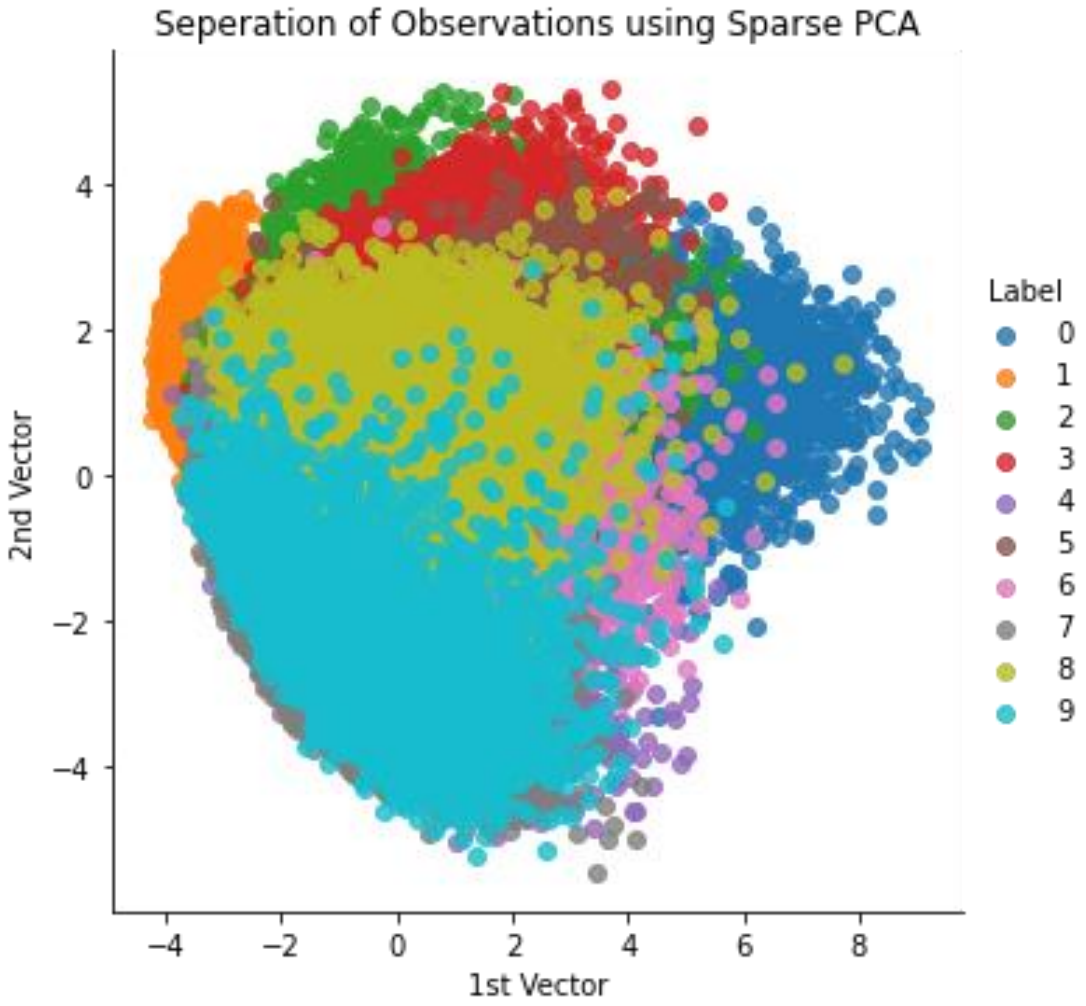
일반 PCA와 결과는 유사한 편.

1-1-2. 일반 vs 점진적 PCA





1-1-3. 희소 PCA

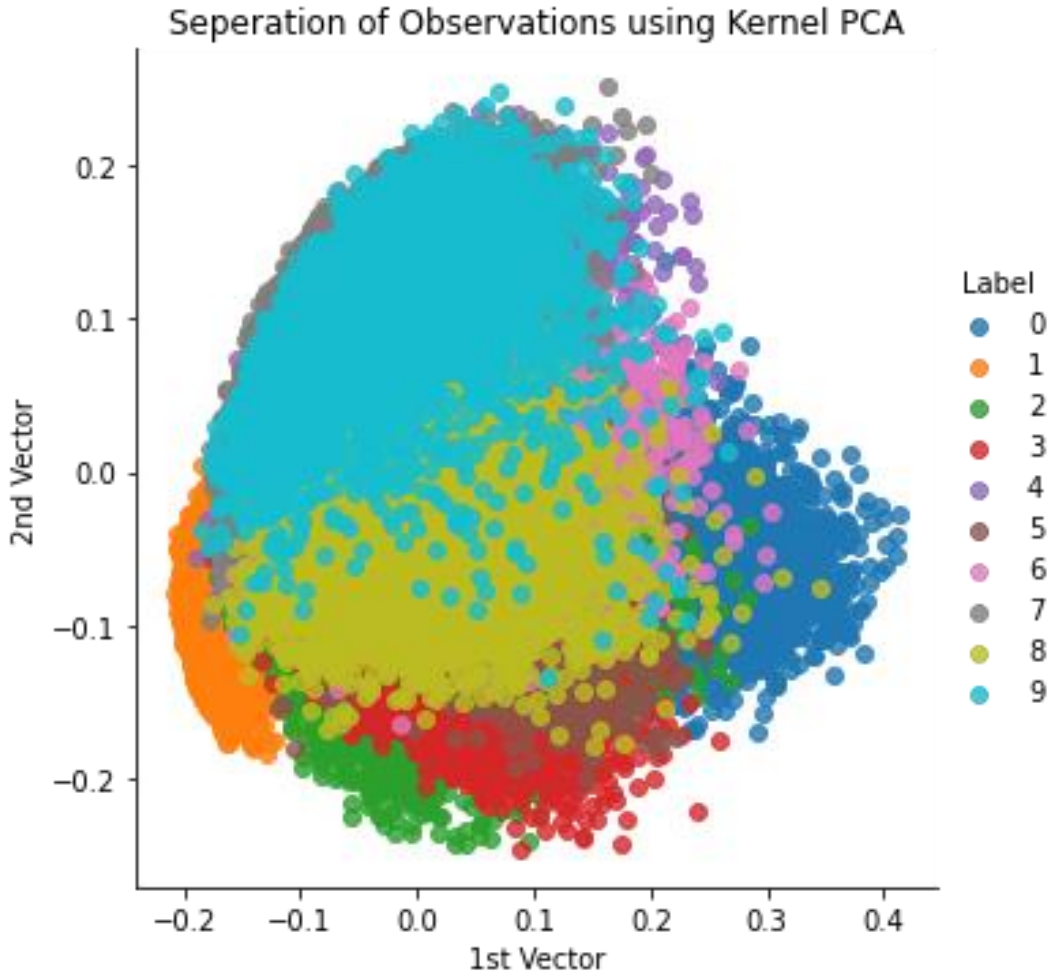


일반 PCA에서의 주성분은
모든 피처들의 선형결합으로 이루어집니다.

그러나 **희소 PCA**의 주성분은
일부 피처들만의 선형결합이 되어,
Sparsity가 비교적 잘 유지됩니다!

(Sparse: 값이 대부분 0인 행렬/데이터)

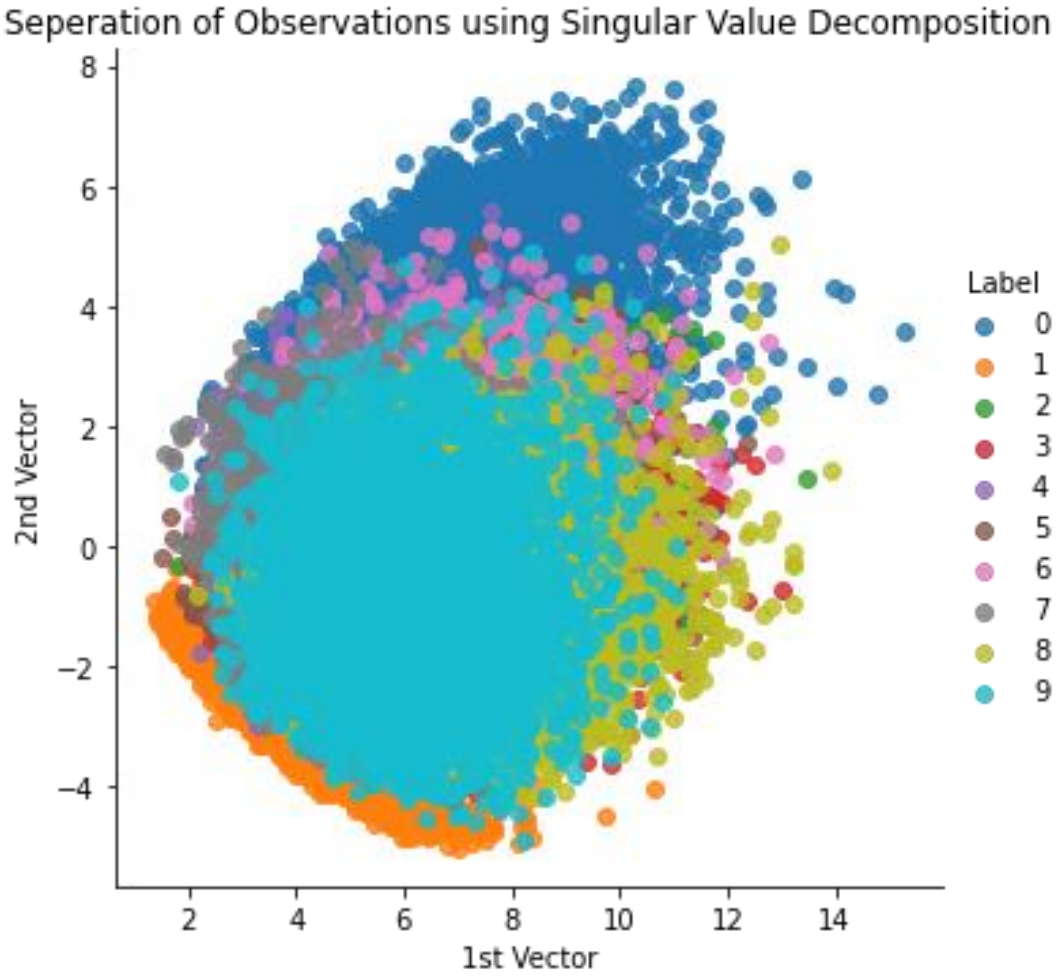
1-1-4. Kernel PCA



커널 PCA도 주성분 분석입니다
그런데 이제 **비선형성**을 곁들인..

주성분 분석은 원래 피처의 선형 결합을
사용하기 때문에 **선형적** 방법입니다!
Kernel Trick을 활용해
비선형성까지 잡고자 하는 기술입니다.

1-2. 특잇값 분해 (SVD)



앞서 봤던 $A = U\Sigma V^T$ 형태에서,
Σ 의 차원을 의도적으로 $k \times k$로 낮추는 기법!

(Rank-k approximation,
혹은 Low-rank approximation
이라고도 합니다)

1-3. 랜덤 투영 (Random Projection, RP)

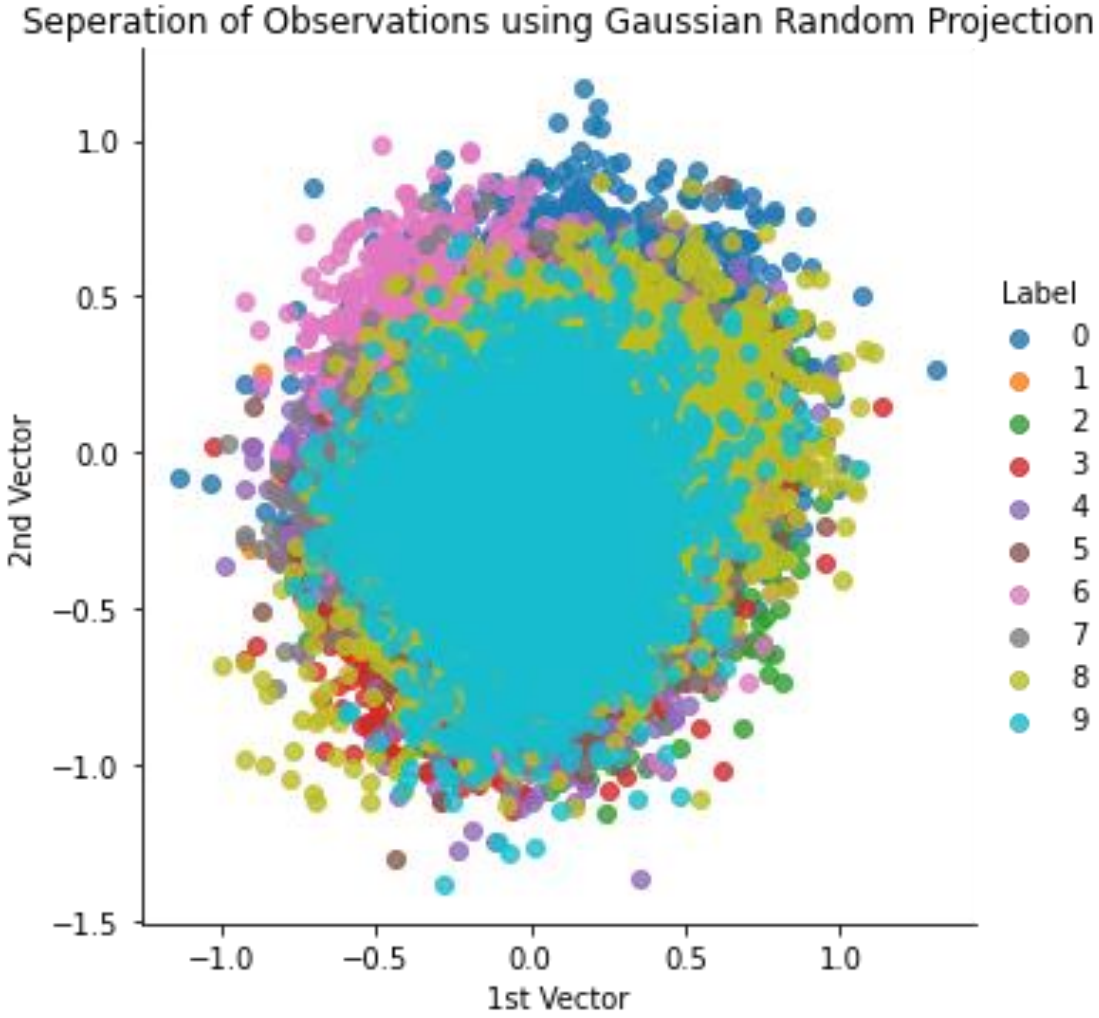
랜덤 투영의 이론적 기반은 Johnson-Lindenstrauss Lemma입니다.

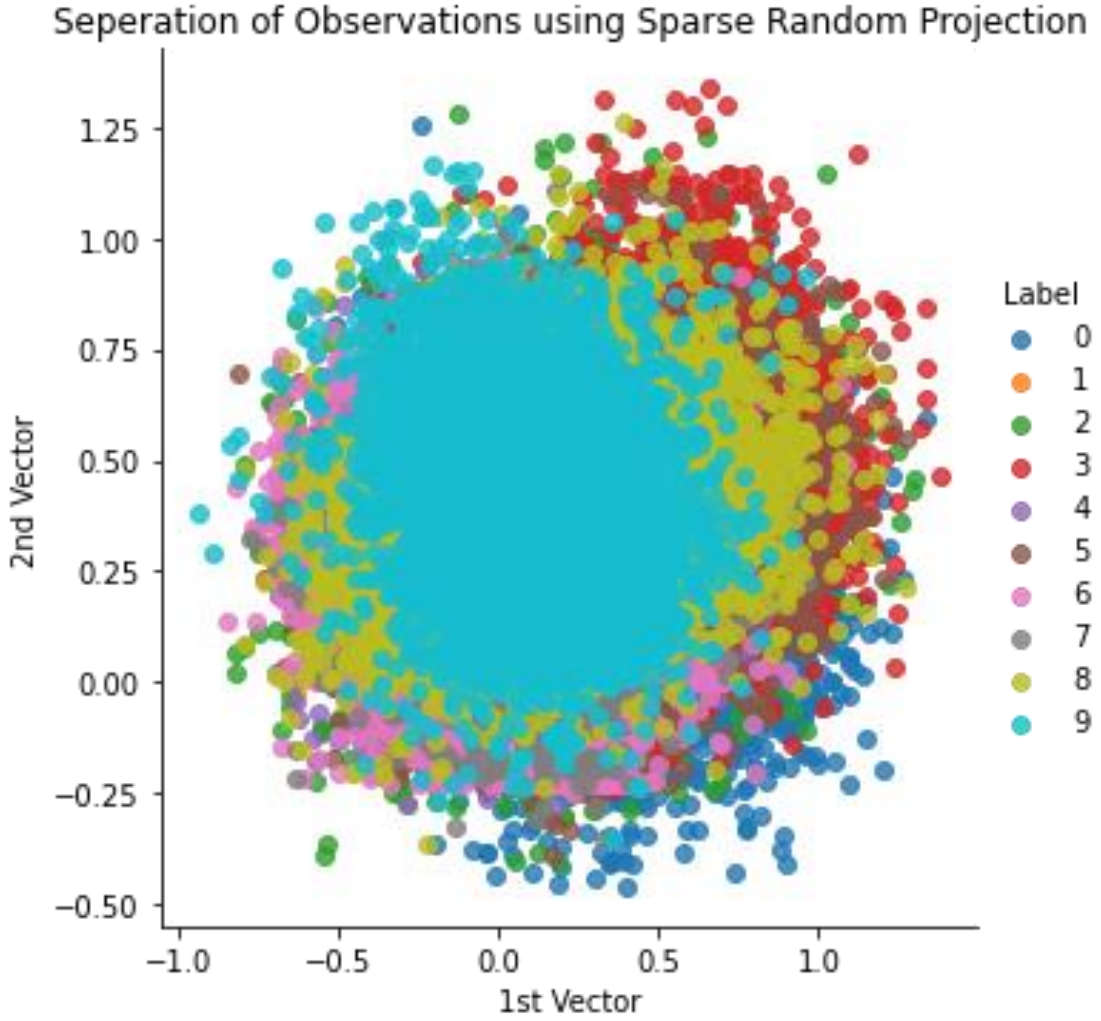
* Johnson-Lindenstrauss Lemma:
$0 < \varepsilon < 1$일 때 N차원으로 표현된 점 m개가 있으면,
다음 조건을 만족시키는 linear map $f : \mathbb{R}^N \to \mathbb{R}^n$ 이 반드시 존재한다.

$$(1-\epsilon)||u-v||^2 \leq ||f(u)-f(v)||^2 \leq (1+\epsilon)||u-v||^2$$

의미] 데이터 간의 거리를 큰 차이 없이 보존하면서,
데이터의 차원을 줄이는 것이 가능하다!

1-3-1~2. GRP & SRP


Seperation of Observations using Gaussian Random Projection
2nd Vector
1st Vector
Label
0
1
2
3
4
5
6
7
8
9


Seperation of Observations using Sparse Random Projection
2nd Vector
1st Vector
Label
0
1
2
3
4
5
6
7
8
9

1-4. IsoMap

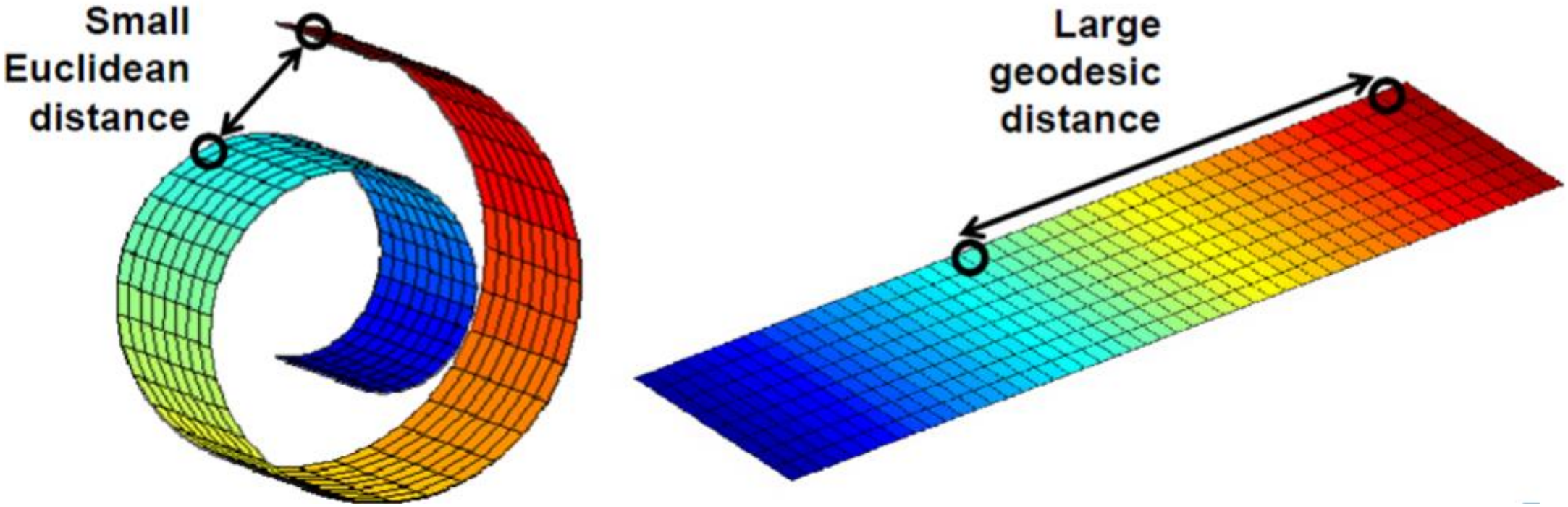
앞서 살펴본 PCA, SVD, RP는 모두 선형 차원 축소 방법론이지만,
IsoMap (Isometric Mapping)은 **비선형** 차원 축소 기법입니다!
이러한 기법을 *매니폴드 학습이라고 통칭하기도 합니다.

IsoMap에서는 데이터 간의 거리를 측정할 때
일반적인 유클리드 거리 (Euclidean Distance) 가 아니라,
**지오데식 거리 (Geodesic Distance) 를 사용합니다.

각 포인트들과 이웃하는 포인트들 간의 상대적인 위치 (네트워크에서의 연결 관계) 를
기반으로 원본 데이터의 고유한 기하학적 구조를 학습합니다!

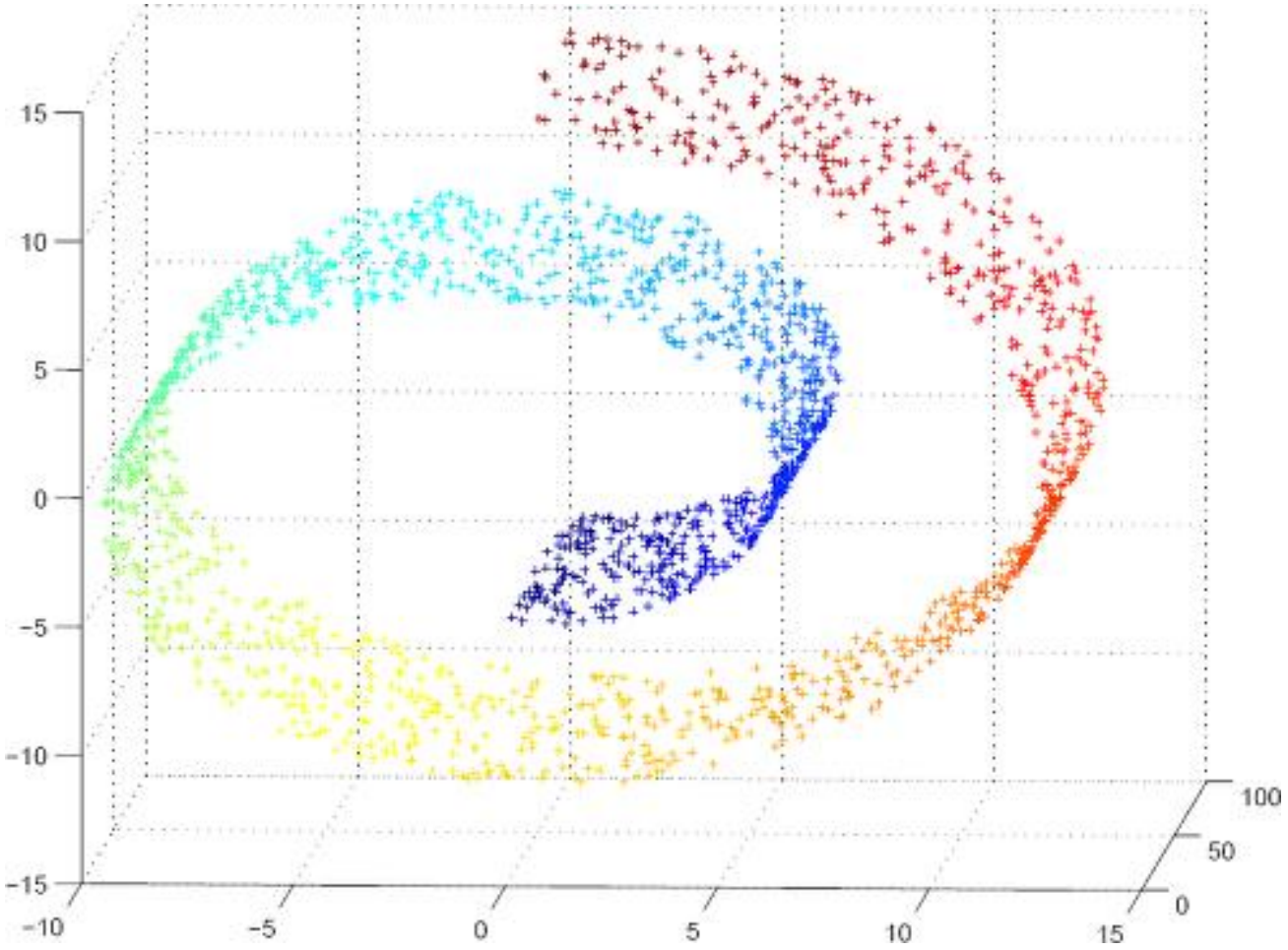
단, 선형 차원 축소 방법보다 학습 속도가 느립니다.

1-4. IsoMap



지오데식 거리: 곡선 거리라고도 하며, 각각의 데이터를 네트워크로 볼 때
Node와 Node 사이의 거리 (=Edge의 개수)
간편하게는, 어떤 공간의 표면을 타고 움직이는 식으로 구한 거리

1-4. IsoMap



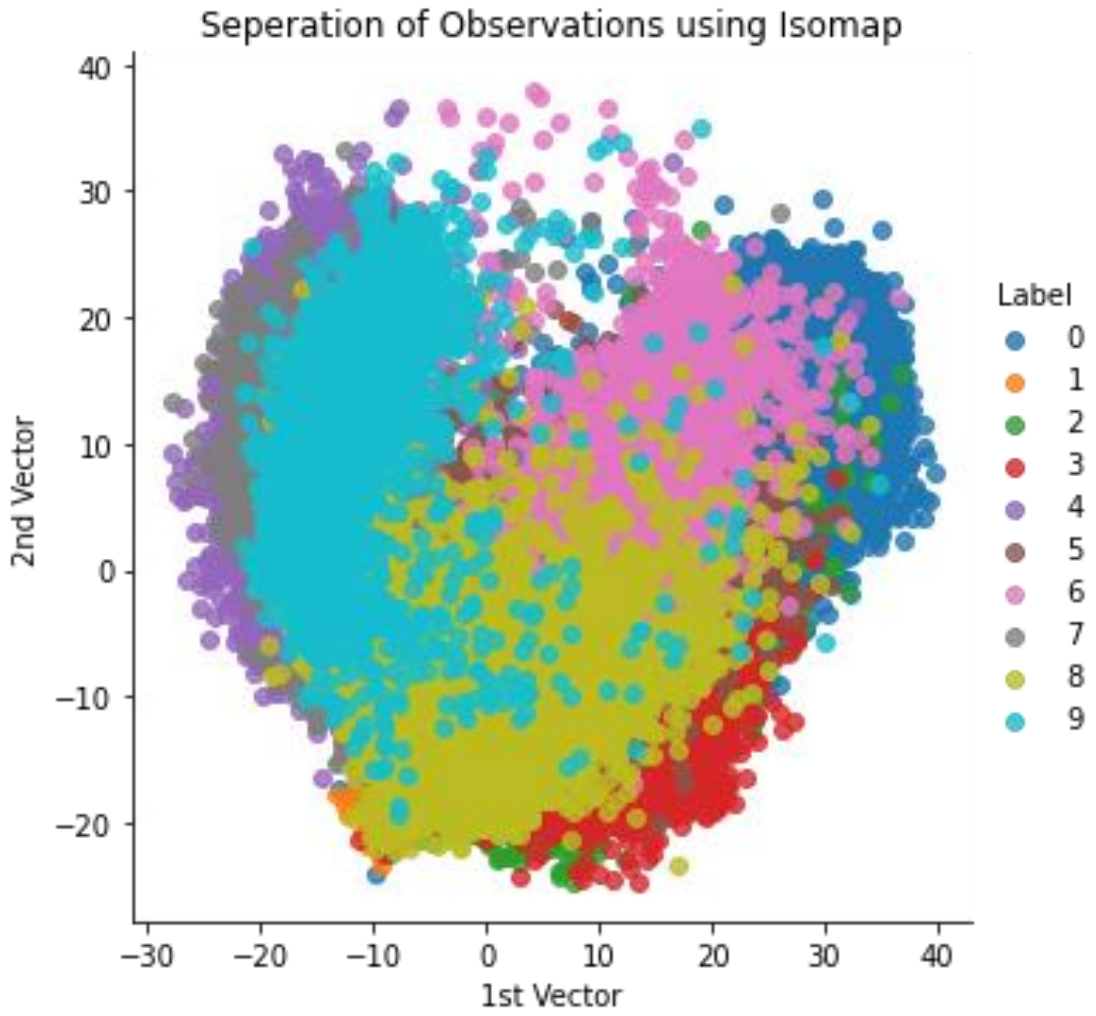
Ex] Swiss Roll

매니폴드 (Manifold):
국소적으로 유클리드 공간과 닮았으나,
전체적으로는 그렇지 않은 위상 공간

= 근처에서는 (유클리드)=(지오데식),
그 밖에서는 (유클리드)≠(지오데식)

= "꽤 복잡하게 생긴" 위상 공간!

1-4. IsoMap


Seperation of Observations using Isomap
2nd Vector
1st Vector
40
30
20
10
0
−10
−20
−30
−20
−10
0
10
20
30
40
Label
0
1
2
3
4
5
6
7
8
9

2. 군집화 (Clustering)

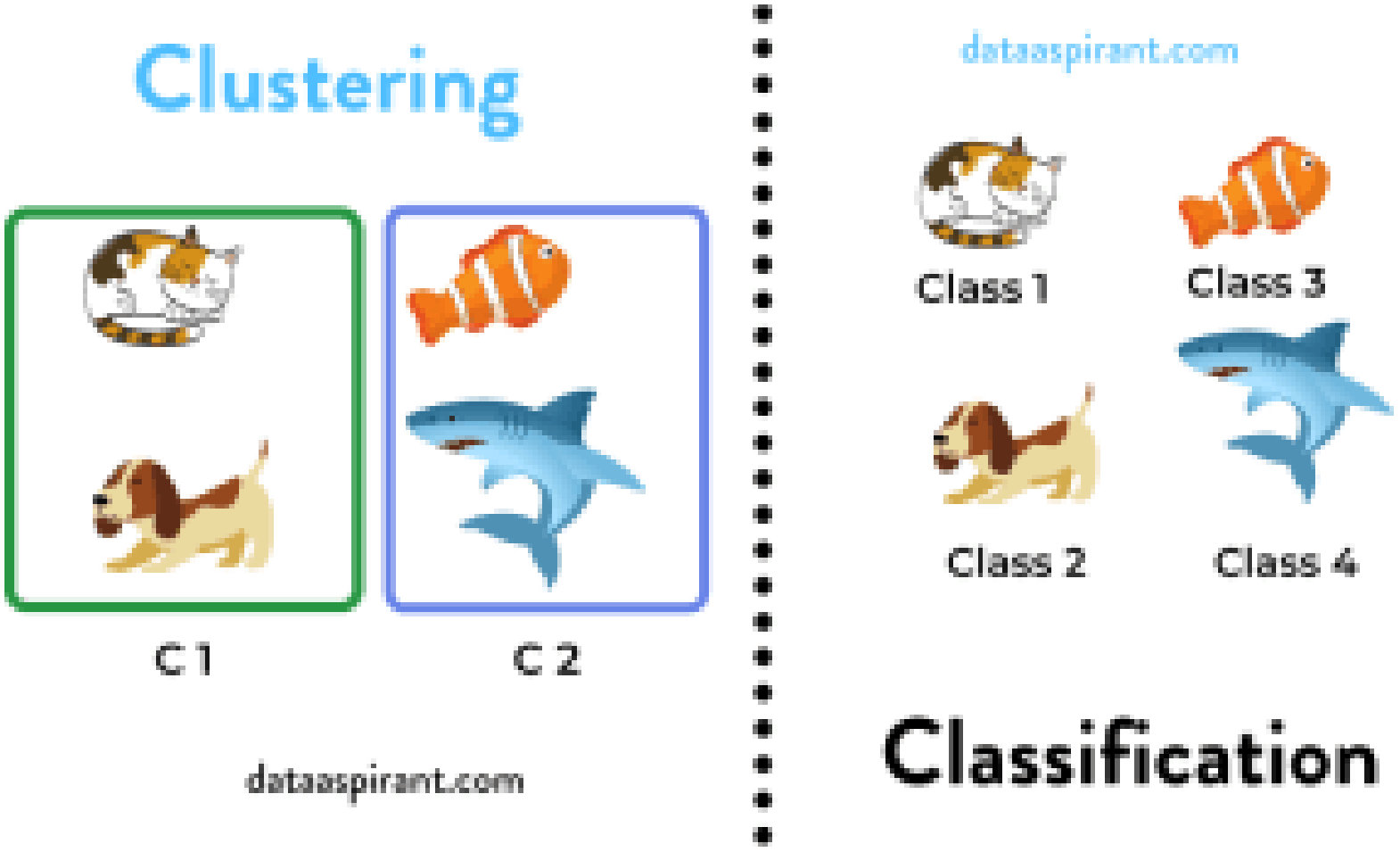
군집화(Clustering): 레이블을 사용하지 않고 하나의 관측치가 다른 관측치의 데이터와 얼마나 유사한지 비교해, 관측치를 그룹화하는 것

단순히 데이터의 유형을 구분 짓는 것부터,
마케팅/음악/책/SNS/쇼핑 등 추천 시스템에 응용될 수도 있고,
특정 서비스의 사용자 자체를 범주화하는 것도 가능합니다.

그 정보를 활용해 광고 타겟팅을 개선할 수도 있는 등,
활용 가능성이 무궁무진합니다!

2. 군집화 (Clustering)

주의] Clustering은 Classification과 다릅니다!



2-1. k-means Clustering

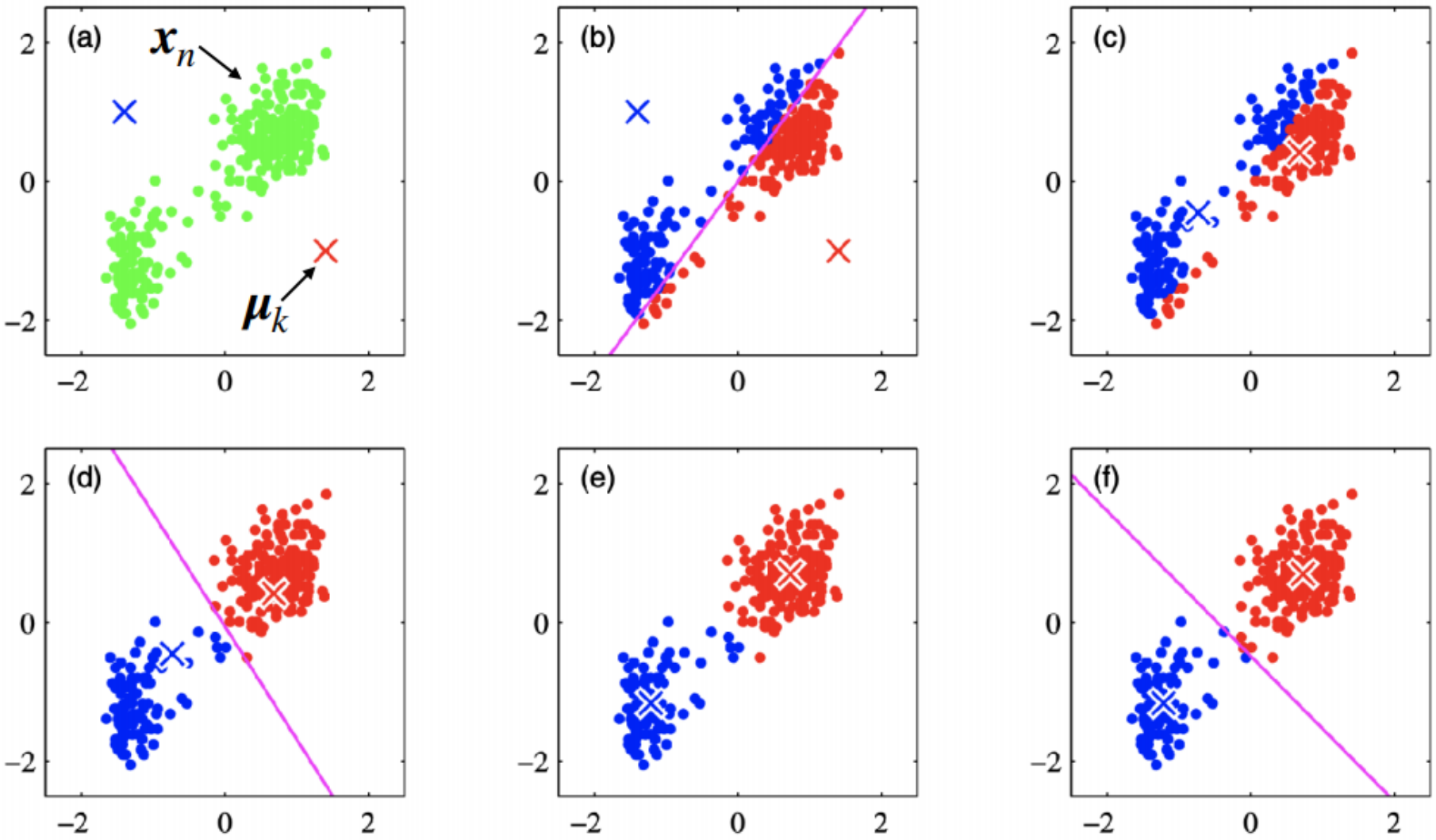
가장 대표적인 클러스터링 기법으로,
군집의 개수를 사용자가 설정할 수 있는 방식입니다.

구체적으로 k-means Clustering은 다음과 같은 방식으로 데이터를 학습합니다!

(1) 임의로 k개의 Centroid를 선정해 각각이 군집을 대표하게 하고,
(2) 그 값들과 가까이 있는 관측치를 군집으로 묶습니다.
(3) 새롭게 만들어진 군집에서 Centroid를 다시 계산해 수정합니다.
(4) 군집의 변동이 거의 없어질 때까지 (2)와 (3)을 반복합니다.

1단계에서 Centroid를 임의로 설정하기 때문에
K-means clustering은 시행할 때마다 결과가 조금씩 달라질 수 있습니다.
하지만 결과적으로 군집 내의 분산(=관성, Inertia)을 최소화하는 것이 목적입니다!

2-1. k-means Clustering



2-2. Hierarchical Clustering

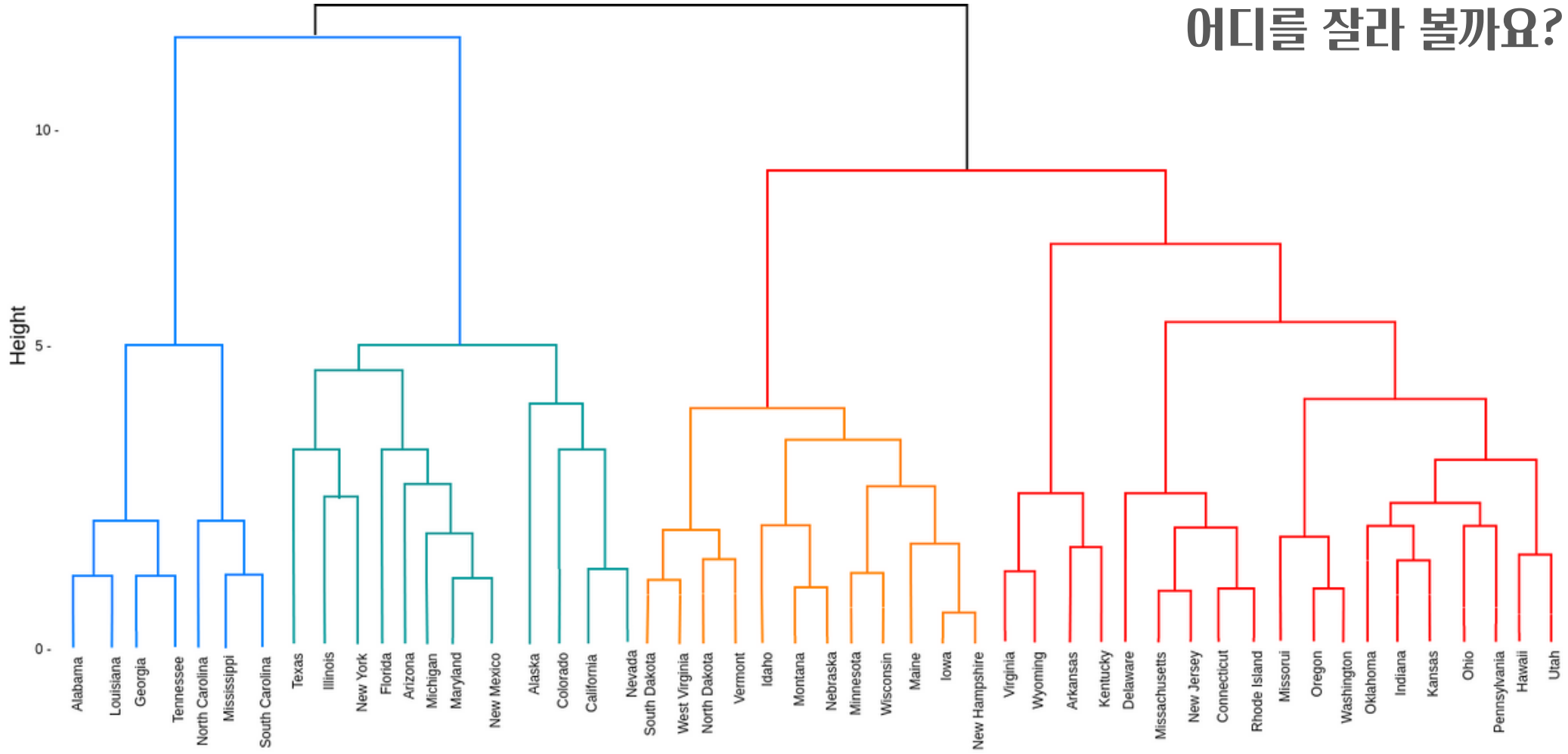
계층적 군집화는 사용자가 군집의 개수를 설정하지 않습니다.
대신 분석 결과를 확인한 후에, 사용자가 군집 수를 결정할 수 있습니다.

이는 계층적 군집화에서는
데이터 각각을 *덴드로그램(Dendrogram)으로 표현하기 때문에 가능합니다!

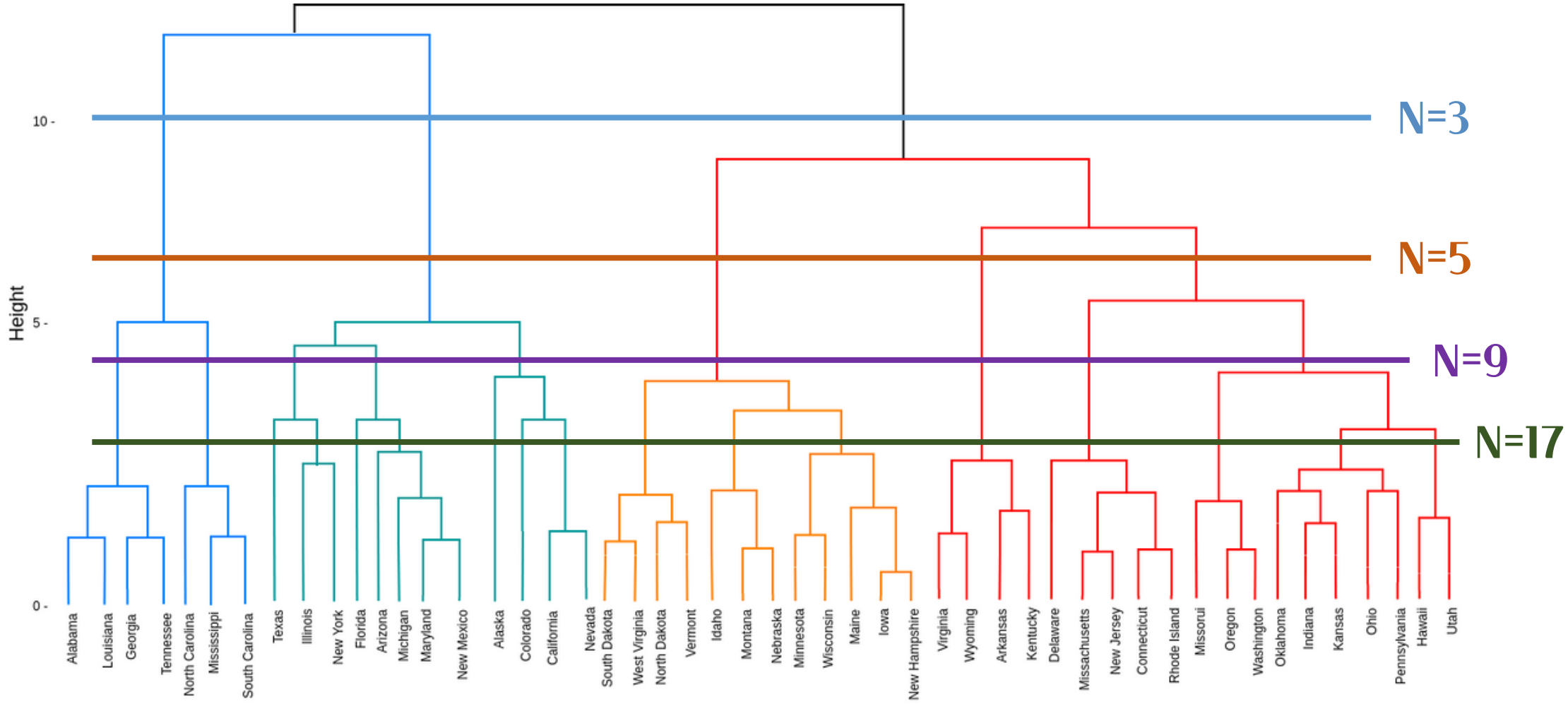
덴드로그램: 각각의 데이터의 유사도를 트리로 표현한 그림

2-2-1. 덴드로그램 (Dendrogram)

어디를 잘라 볼까요?



2-2-1. 덴드로그램 (Dendrogram)



3. Summary

0. Intro

- 지도학습 vs 비지도학습
- 고윳값 분해
- 특잇값 분해
- MNIST 손글씨 data set
- Network Analysis

1. 차원 축소

- 주성분 분석 (PCA)
- 특잇값 분해 (SVD)
- 랜덤 투영 (RP)
- IsoMap

2. 군집화

- K-means clustering
- Hierarchical clustering

References

https://www.thesprucepets.com/cat-talk-eyes-553942 (Cat)
https://www.notion.so/Unsupervised-Learning-a1813c4d381345df9ef4af16e5165db9 (22-2 DSL Unsupervised Learning Notion)
https://www.researchgate.net/figure/Example-images-from-the-MNIST-dataset_fig1_306056875 (MNIST image)
https://towardsai.net/p/l/centroid-neural-network-an-efficient-and-stable-clustering-algorithm (clustering)
https://lovit.github.io/machine%20learning/vector%20indexing/2018/03/28/lsh/ (JL lemma)
https://en.wikipedia.org/wiki/Low-rank_approximation (SVD - Low Rank Approximation)
https://woosikyang.github.io/first-post.html (IsoMap (1))
https://excelsior-cjh.tistory.com/168 (Manifold Learning)
https://scikit-learn.org/stable/modules/manifold.html#isomap (IsoMap (2))
https://bjlkeng.github.io/posts/manifolds/ (Manifold)
https://dataaspirant.com/5-clustering-vs-classification-example/ (Clustering vs Classification)
https://needjarvis.tistory.com/719 (k-means Clustering Example)
https://online.visual-paradigm.com/diagrams/templates/dendrogram/cluster-dendrogram/ (Dendrogram)
https://seongjuhong.com/2020-03-14pm-hierarchical-clustering/ (Hierarchical Clustering Example)
https://cran.r-project.org/web/packages/fastcluster/vignettes/fastcluster.pdf (fastcluster.linkage_vector)

22-1학기 머신러닝을위한수학 (박경덕 교수님) Lecture Note
22-1학기 계량경제학(1) (정진욱 교수님) Lecture Note
19-2학기 데이터사이언스(2):네트워크자료분석 (진익훈 교수님) Lecture Note
〈핸즈온 비지도 학습〉

Thank you!