DATA
SCIENCE LAB

Dimensionality Reduction

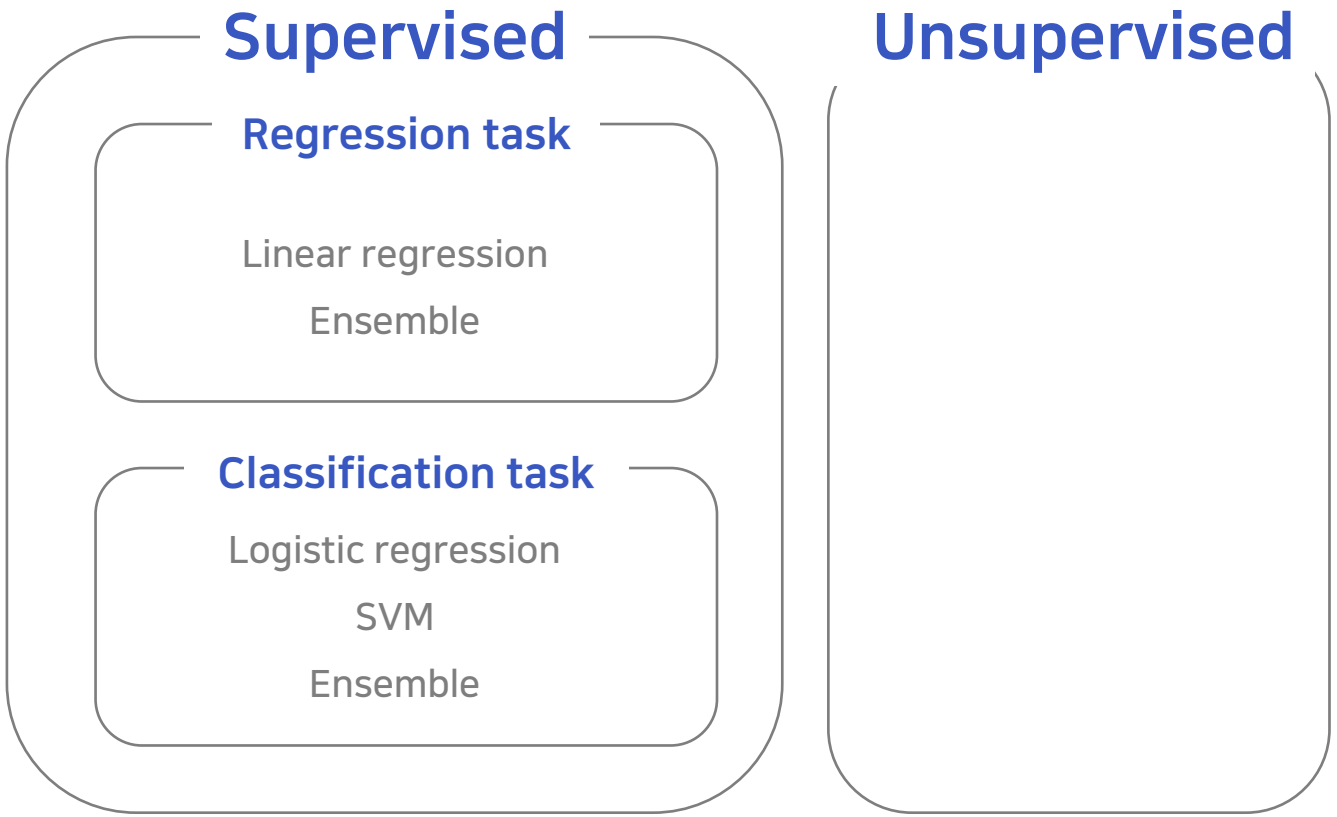
23.02.07 / 8기 최윤서

CONTENTS

0. Review

지금까지 배운 4가지 방법론

- Linear regression
- Logistic regression
- SVM
- Ensemble (Decision Tree)



0. Review

Supervised

Regression task

Linear regression

Ensemble

Classification task

Logistic regression

SVM

Ensemble

4가지 방법론의 공통점 및 차이점

- (공통점) y라벨이 존재하는 지도학습이다
- (공통점) 주어진 학습데이터를 바탕으로 적절한 y값을 잘 예측하는 것이 목표임

  Regression task라면 알맞은 값(숫자) 예측

  Classification task라면 알맞은 클래스 예측
- (차이점) 학습데이터의 입력변수들과 출력변수의 관계를 학습하는 방식이 다름
- 즉, 방법은 다르지만 y라벨값을 잘 예측하도록 학습한다는 목표 자체는 동일

1. INTRO

지금까지의 관점

- 입력변수(X1, X2, …)를 가지고 y를 잘 예측해보자
- using Linear regression / Logistic regression / SVM / Ensemble

차원축소란?

- 차원 = 학습에 사용하는 데이터 입력변수의 수 (feature의 수, 독립변수 X의 수)
- 차원축소 = 학습에 사용하는 데이터 입력변수의 수를 줄여보자

1. INTRO

Why 차원축소?

차원의 저주 때문에

차원의 저주란, 입력변수가 너무 많을 때 발생하는 총체적 난국이다

- 필요없거나 중복되는 입력변수들이 많으면 모형이 과적합되거나 다중공선성 문제가 발생할 수 있음.
  이로 인해 모형의 성능이 떨어질 수 있음
- 입력변수가 많으면 학습 속도가 더 오래 걸림
- 차원이 많으면(3차원 이상) 시각화를 할 수 없어 직관적이지 않음

그러나 입력변수 공간을 줄일 때(차원 축소할 때) 본래의 정보량을 최대한 보존하는 방향으로 해야함

1. INTRO

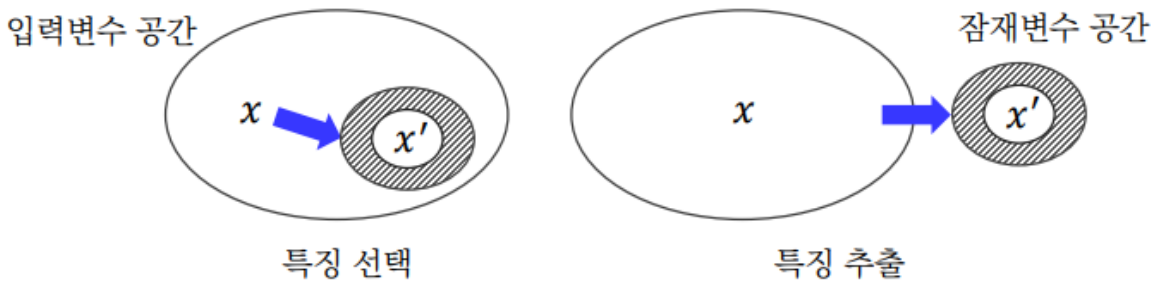
차원 축소의 종류

- 특징 선택

  중요한 입력변수를 찾는 과정. 입력변수간에 중복되고, 출력변수 예측에 상관없는 입력변수를 제거하여 출력변수와 관련이 깊은 중요 입력변수를 선택하는 과정

- 특징 추출

  기존 입력변수들의 조합으로 새로운 특징을 생성함.



2. Feature selection

출력변수와 관련이 깊은 중요 입력변수를 선택하는 과정

중요 입력변수들을 고를 때 그 기준이 되는 것이 출력변수. 라벨값과 관련성이 높은 입력변수들을 선택하고 있기 때문에 이는 supervised dimensionality reduction task이다

- Forward selection
  중요할 것으로 생각되는 변수들을 순차적으로 선택하면서 변수집합 늘려나가기
  until 현재 조합에 변수 추가했을 때 성능 향상이 유의미하게 일어나지 않을 때까지
- Backward elimination
  전체 입력변수 공간에서 입력변수 하나씩 제거하면서 최적의 변수 집합 찾기
  until 현재 집합에서 어떤 변수를 빼더라도 성능 저하가 급격히 일어날 때까지

2. Feature selection

mRMR

- minimum Redundancy Maximum Relevance의 약자로
  입력변수간의 관계는 최소화하고 입력변수와 출력변수 간의 관계는 최대화하는 특징 선택 방법
- 공집합에서부터 중요한 변수들을 순차적으로 늘려가면서 선택하는 forward selection 예시

minimum Redundancy Maximum Relevance Algorithm

1: p개의 입력변수 $X_1, X_2, \cdots, X_p$가 존재할 때 핵심 입력변수 후보 집합을 F로 핵심 입력변수로 선택된 집합을 S로 지정하고 선택할 변수의 개수인 k를 설정

$$F = \{X_1, X_2, \cdots, X_p\}, S = \emptyset$$

2: 입력변수 별로 출력변수 Y와의 상관계수 $r(X_i; Y)$를 계산

3: 첫 번째 핵심 입력변수를 아래 식을 이용해 탐색

$$X^*_{first} = argmax_{i=1,\cdots,p}\{|r(X_i; Y)|\}$$

탐색 후 $F \leftarrow F - \{X^*_{first}\}$ 그리고 $S \leftarrow \{X^*_{first}\}$로 설정

4: $|S| = k$가 될 때까지 다음을 반복

A: 집합 F와 S에 속한 입력변수들간의 상관계수 $\{r(X_i; X_j)|i \in F, j \in S\}$를 계산

B: F에 속한 변수 중에서 다음 핵심 입력변수는 아래 식을 이용해 탐색

$$X^*_{next} = argmax_{i \in F}\left\{|r(X_i; Y)| - \frac{1}{|S|}\sum_{j \in S}|r(X_i; X_j)|\right\}$$

탐색 후 $F \leftarrow F - \{X^*_{next}\}$ 그리고 $S \leftarrow S + \{X^*_{next}\}$로 설정

2. Feature selection

mRMR 예시

상관계수	X_1	X_2	X_3	X_4	X_5
X_1	-	0.38	0.10	0.3	0.21
X_2	-	-	0.89	0.96	0.01
X_3	-	-	-	0.71	0.96
X_4	-	-	-	-	0.14
X_5	-	-	-	-	-

상관계수	y
X_1	0.73
X_2	0.07
X_3	0.78
X_4	0.80
X_5	0.87

변수 선택을 통해 몇 개를 뽑을 것인가?

3개(하이퍼 파라미터)

1. Y와 상관성이 가장 높은 X_5 선택

$$X^*_{first} = argmax_{i=1,\cdots,p}\{|r(X_i;Y)|\}$$

$$F = \{X_1, X_2, X_3, X_4\}, S = \{X_5\}$$

2. 다음 중요변수 선택

$$X^*_{next} = argmax_{i\in F}\left\{|r(X_i;y| - \frac{1}{|S|}\sum_{j\in S}|r(X_i;X_j)|\right\}$$

$$= argmax_{i\in F}\{|r(X_i;y| - |r(X_i;X_5)|\}$$

$$X_1 = |r(X_1;y)| - |r(X_1;X_5)| = 0.73 - 0.21 = 0.52$$

$$X_2 = 0.06, X_3 = -0.18, X_4 = 0.66$$

X_4를 다음 변수로 선택

$$F = \{X_1, X_2, X_3\}, S = \{X_4, X_5\}$$

3. 다음 중요변수 선택

$$X_1 = |r(X_1;y)| - \frac{|r(X_1;X_4)| + |r(X_1;X_5)|}{2} = 0.73 - \frac{0.3 + 0.21}{2} = 0.475$$

$$X_2 = -0.415, X_3 = -0.055$$

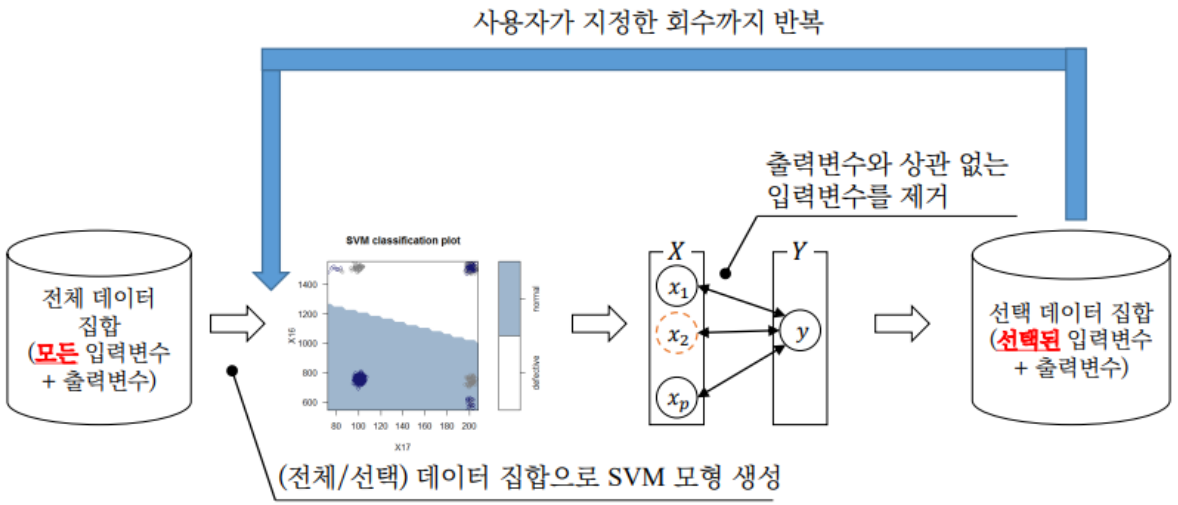
X_1을 다음 변수로 선택

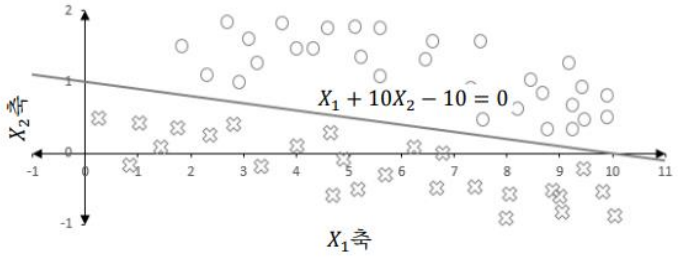
$$F = \{X_2, X_3\}, S = \{X_1, X_4, X_5\}$$

2. Feature selection

SVM-RFE

- Support Vector Machine – Recursive Feature Elimination의 약자
- SVM 모형을 활용하며 출력변수와 관계가 적은 입력변수를 하나씩 제거해주는 backward elimination 알고리즘.
- (장점) 비선형 연관성도 고려할 수 있다
- (단점) 입력변수들 간의 다중공선성을 반영할 수 없다





ex) X2가 출력변수를 더 잘 설명해주는 중요 변수이다

2. Feature selection

Ridge&Lasso

- 선형 회귀에서 과적합, 다중공선성을 제어하기 위하여 규제항을 추가
- 필요 없거나 겹치는 입력변수의 회귀계수를 알아서 0으로 만들어준다
- 차원 축소와 모델 학습을 동시에 시키는 것

Ridge 회귀에서는 **회귀계수의 제곱합을** $f(\hat{\beta})$에 대입 : L2 규제

$$Minimize \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \hat{\beta}_j^2$$

Lasso 회귀에서는 **회귀계수의 절대값 합을** $f(\hat{\beta})$에 대입 : L1 규제

$$Minimize \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\hat{\beta}_j|$$

λ는 하이퍼 파라미터로 크면 클수록 보다 많은 회귀계수를 0으로 (또는 0으로 가깝게) 만듦

3. Linear Feature extraction

기존 입력변수들의 조합으로 새로운 특징을 생성함.

		기존 입력변수 어떻게 조합할 것인가?	
		선형 결합	비선형 결합
Y라벨을 고려하여 특징을 생성할 것인가?	Unsupervised	PCA MDS	KPCA Isomap LLE t-SNE AutoEncoder
	Supervised	LDA	KFDA

3. Linear Feature extraction

		기존 입력변수 어떻게 조합할 것인가?	
		선형 결합	비선형 결합
Y라벨을 고려하여 특징을 생성할 것인가?	Unsupervised	PCA MDS	KPCA Isomap LLE t-SNE AutoEncoder
	Supervised	LDA	KFD

PCA(주성분 분석)

- **기존 입력변수들의 선형결합을** 통해 새로운 특징을 만들어냄
  이와 같이 만들어진 새로운 특징 = 주성분, 잠재변수
- Y라벨을 고려하지 않고 (unsupervised) 특징 추출. 그렇다면 어떤 것을 기준으로 특징 추출?
  분산을 크게 만드는 새로운 특징을 만들자

PCA에서의 기본 가정) 분산이 큰 변수가 중요한 변수이다

	국어	수학
평균	90	70
분산	20	600

변별력이 있으려면 점수의 분포가 넓게 퍼져 있어야 되므로
분산이 큰 수학이 가장 변별력이 높은 시험이다
주성분분석에서는 분산이 큰 변수 = 변별력 있는 변수 = 중요한 변수

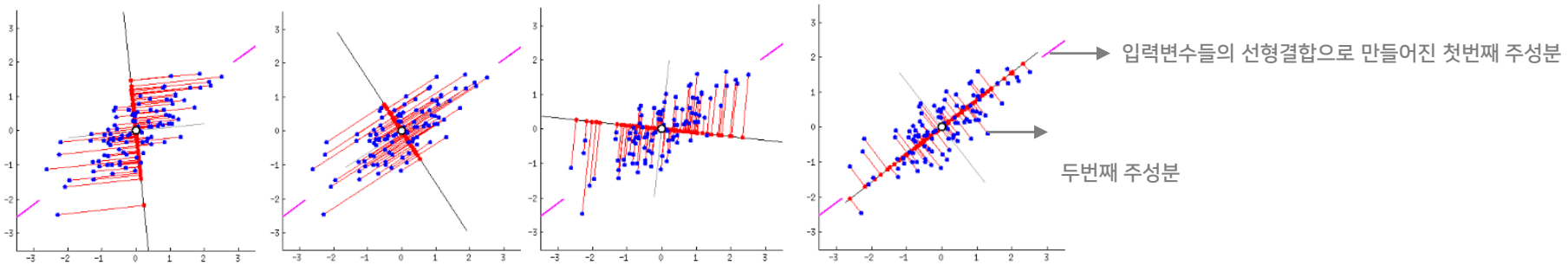
3. Linear Feature extraction

PCA(주성분 분석) 수행방법

1. 각 변수는 변수 스케일링(Z-score)를 수행한다

   PCA는 전체 분산을 가장 크게 하는 특징을 찾기 때문에 변수 스케일링을 통해 모든 변수들을 동일한 기준(동일한 분산)으로 맞추고 특징을 찾아야한다.

2. 학습데이터로부터 데이터들의 분산을 가장 크게 하는 축(T1)을 탐색



3. 첫번째 축(T1)과 수직인 축 중에서 분산이 가장 큰 축(T2)을 탐색

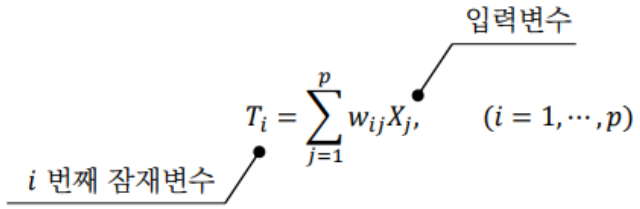
   2차원에서는 수직인 축은 하나만 존재하므로 T1이 결정되면 T2는 정해짐

4. 입력변수의 수(원 데이터의 차원)만큼 새로운 축(잠재변수)를 탐색

3. Linear Feature extraction

PCA(주성분 분석) 수행방법

잠재변수의 분산을 크게 만드는 선형결합 계수들(=로딩벡터)를 구하는 것이 목적



잠재변수는 기존 입력변수들의 선형결합으로 이루어짐

1. 각 변수는 변수 스케일링(Z-score)를 수행한다.
2. 학습데이터로부터 데이터들의 **분산을 가장 크게 하는 축(T1)**을 탐색

$$Maximize \quad Var(T_1) \quad Subject\ to \quad \sum_{j=1}^{p} w_{1j}^2 = 1$$

3. 첫번째 축(T1)과 수직인 축 중에서 분산이 가장 큰 축(T2)을 탐색
4. 입력변수의 수(원 데이터의 차원)만큼 새로운 축(잠재변수)를 탐색

$$Maximize \quad Var(T_k) \quad Subject\ to \quad \sum_{j=1}^{p} w_{kj}^2 = 1$$
$$Cov(T_v, T_k) = 0 \; for \; v = 1, 2, \cdots, k-1$$

3. Linear Feature extraction

PCA(주성분 분석) 수행방법

잠재변수의 분산을 크게 만드는 선형결합 계수들(=로딩벡터)를 구하는 것이 목적

$$\text{Maximize} \quad Var(T_1) \quad \text{Subject to} \quad \sum_{j=1}^{p} w_{1j}^2 = 1$$

$$\text{Maximize} \quad Var(T_k) \quad \text{Subject to} \quad \sum_{j=1}^{p} w_{kj}^2 = 1, \quad Cov(T_v, T_k) = 0 \; for \; v = 1, 2, \cdots, k-1$$

해당 최적화식은 Lagrangian multiplier을 적용해서 풀 수 있음(과정 생략)

$\Rightarrow Sw = \lambda w$ (S는 **공분산 행렬**)

특성방정식 풀면 p개의 eigenvector, eigenvalue

Eigenvector = **주성분을 구성하는 로딩벡터** $(w_1, w_2, \cdots, w_p)$

Eigenvalue = 대응하는 eigenvector로 만들어지는 **주성분의 분산** $(\lambda_1, \lambda_2, \cdots, \lambda_p)$

가장 큰 eigenvalue(λ_1)에 대응되는 eigenvector(w_1)

가장 큰 분산을 가지게 하는 첫번째 주성분(잠재변수) 만들어줌

$$T_1 = {w_1}^T x = w_{11}x_1 + w_{21}x2 + \cdots + w_{p1}x_p$$

두번째로 큰 eigenvalue(λ_2)에 대응되는 eigenvector(w_2)

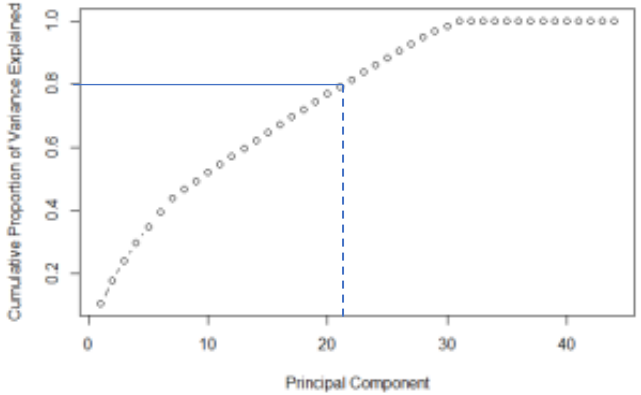
두번째 주성분 만들어줌

$$T_2 = {w_2}^T x = w_{12}x_1 + w_{22}x2 + \cdots + w_{p2}x_p$$

3. Linear Feature extraction

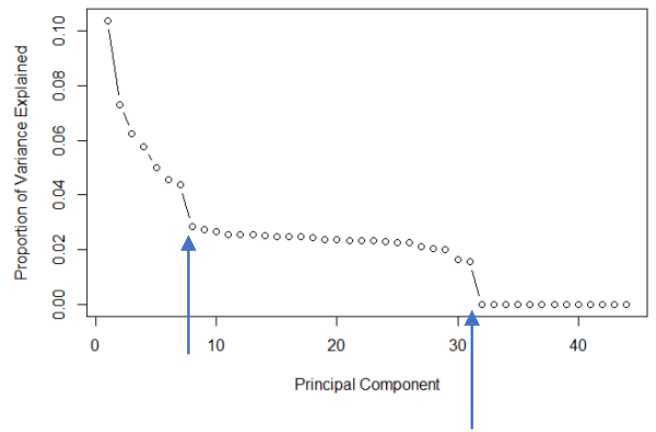
PCA(주성분 분석) 수행결과

- p개의 수직인 주성분을 얻게 됨.
- PCA 수행한 이후 분산이 작은 잠재변수는 제거(즉, 차원축소)



몇 개의 주성분(잠재변수)을 선택할 것인가?

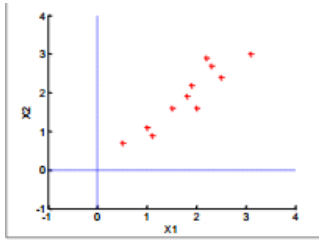
- 전체 데이터의 분산 내에서 선택한 p'개의 잠재변수가 차지하는 분산의 비율을 이용 80-90%를 설명하는 만큼의 주성분 선택
- Elbow point



3. Linear Feature extraction

PCA(주성분 분석) 수행 예시

Original space (2차원)



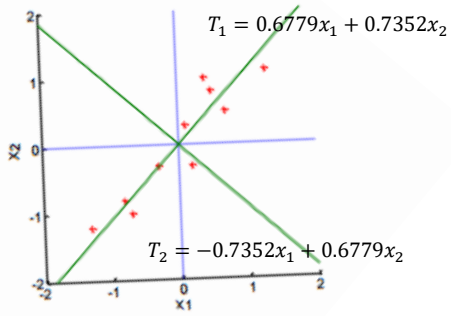
주성분(잠재변수)를 구성하기 위한 로딩벡터 구하기 ➡ 잠재변수 공간

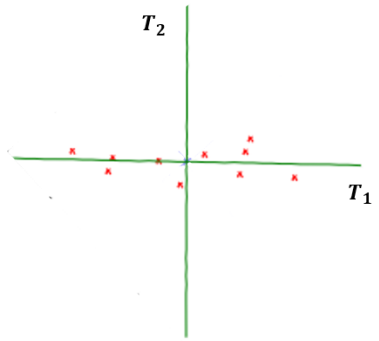
$$Sw = \lambda w \qquad S = \begin{pmatrix} 0.6166 & 0.6154 \\ 0.6154 & 0.7166 \end{pmatrix}$$

$$\Downarrow$$

$$Eigenvectors = \begin{matrix} w_1 & w_2 \\ \begin{bmatrix} 0.6779 & -0.7352 \\ 0.7352 & 0.6779 \end{bmatrix} \end{matrix}$$

$$Eigenvalues = \begin{matrix} \lambda_1 & \lambda_2 \\ (1.2840 & 0.0491) \end{matrix}$$





첫번째 주성분이 설명하는 분산

$$= \frac{\lambda_1}{\lambda_1+\lambda_2} = \frac{1.2840}{1.2840+0.0491} = 0.96$$

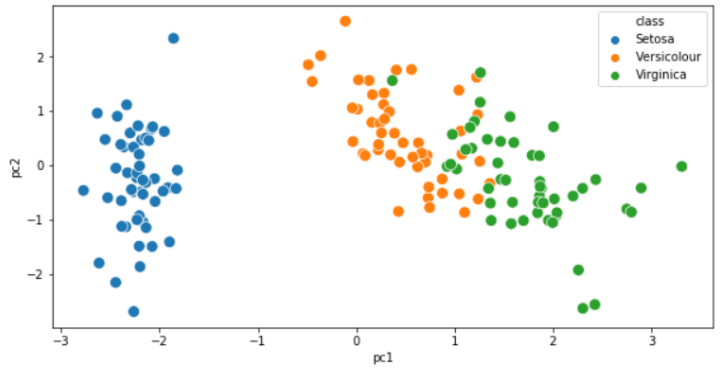
If T2 제거하고 T1만 사용 → 차원축소

3. Linear Feature extraction

PCA(주성분 분석) 활용

- 입력변수의 공간을 축소시킴으로써 다양한 예측/분류 모델의 **성능 향상에** 기여할 수 있다
  PCA가 다른 방법론들을 위한 일종의 **전처리 과정**으로 사용된다

- **시각화** (첫번째와 두번째 주성분 축 이용해서)



Iris datasets

3개의 labels (setosa, virginica, versicolour)

4개의 features(sepal length, sepal width, petal length, petal width)

차원 축소를 통해 2차원으로 시각화 가능

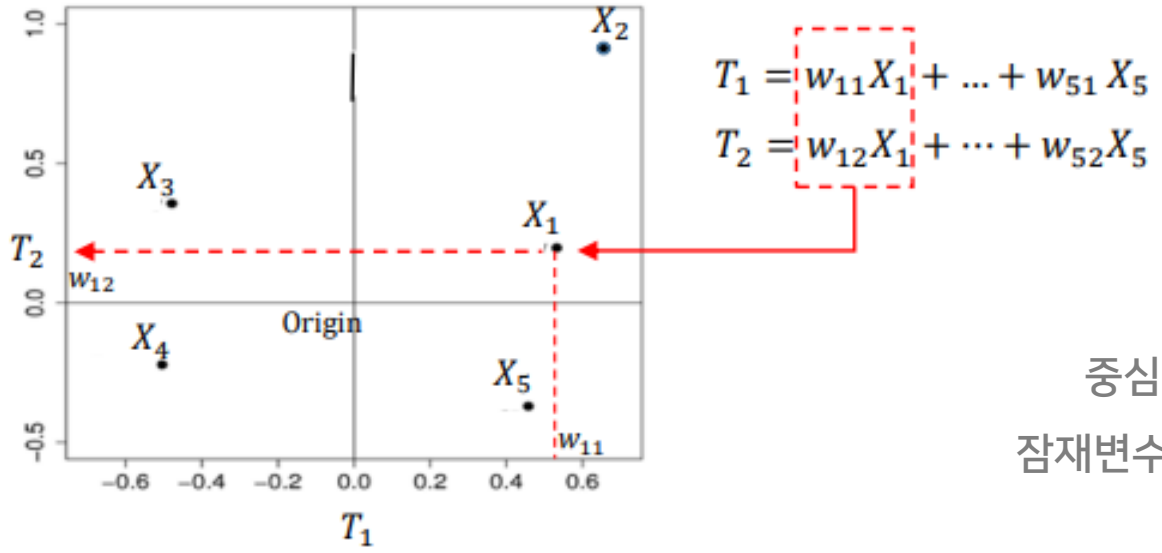
- 저차원에서 클러스터 분석: 어떤 관측치들이 서로 밀집해 있는지 파악

3. Linear Feature extraction

PCA(주성분 분석) 활용

- 중요 변수 확인(with 로딩 플롯)

  잠재변수 공간에 각 원 입력변수의 계수(로딩)을 plotting



Loading Plot

$$T_1 = w_{11}X_1 + \ldots + w_{51}X_5$$
$$T_2 = w_{12}X_1 + \cdots + w_{52}X_5$$

중심으로부터 멀리 떨어져 있는 원변수가

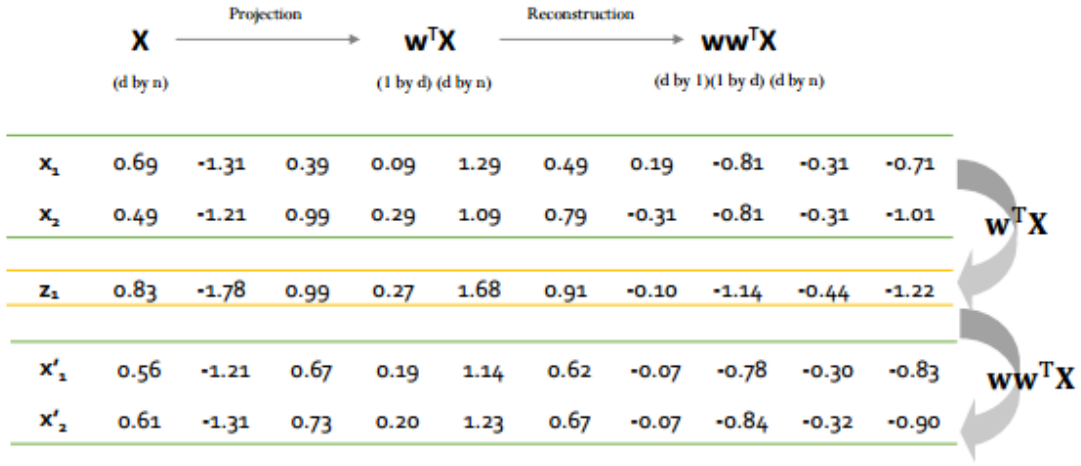
잠재변수 구성에 기여를 많이 하는 중요 변수이다

3. Linear Feature extraction

PCA(주성분 분석) 활용

- PCA를 활용한 이상치 탐지

  Reconstruction error가 클수록 해당 데이터가 이상치일 가능성이 높다



3. Linear Feature extraction

		기존 입력변수 어떻게 조합할 것인가?	
		선형 결합	비선형 결합
Y라벨을 고려하여 특징을 생성할 것인가?	Unsupervised	PCA MDS	KPCA Isomap LLE t-SNE AutoEncoder
	Supervised	LDA	KFD

MDS(다차원 척도법)

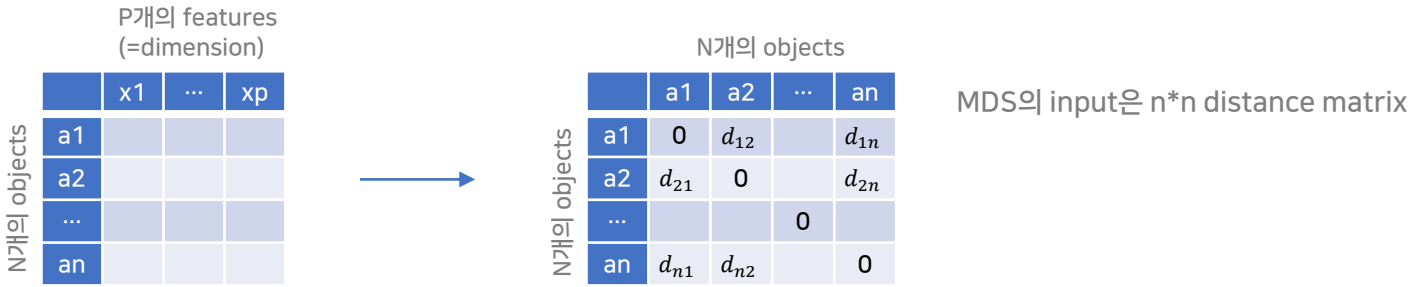
- Multi Dimensional Scaling
- **기존 입력변수들의 선형결합을** 통해 새로운 특징을 만들어냄
- Y라벨을 고려하지 않고 (unsupervised) 특징 추출. 그렇다면 어떤 것을 기준으로 특징 추출?

  원래 공간에서 **데이터 포인트간 pairwise 거리가 저차원에서도 잘 보존되도록 하는** 좌표체계를 찾기

3. Linear Feature extraction

MDS(다차원 척도법) 수행방법

1. Raw data로부터 Distance matrix(대칭행렬)을 만들어줌.



MDS의 input은 n*n distance matrix

2. Distance matrix(D)를 잘 반영하는 좌표 체계를 추출

   (Distance matrix(D) → Inner product matrix(B) → 좌표체계)

3. Linear Feature extraction

** MDS의 coverage가 더 높다
raw data(n*p)일 경우 PCA, MDS 모두 사용 가능
Distance matrix(n*n)일 경우 MDS만 가능

PCA vs MDS

	PCA	MDS
	새로운 축(특징)만들어내면서 차원 축소. 데이터 속에 잠재된 패턴, 구조를 찾아낸다는 측면에서는 동일	
Data input	n*p (n objects in a p-dimensional space)	n*n (distance matrix between n objects)
Purpose	데이터의 분산을 최대한 많이 설명하도록 저차원에 mapping	데이터들 간의 거리를 최대한 그대로 보존하도록 저차원에 mapping

데이터의 특성에 따라 알맞은 차원축소 방법을 사용하자.

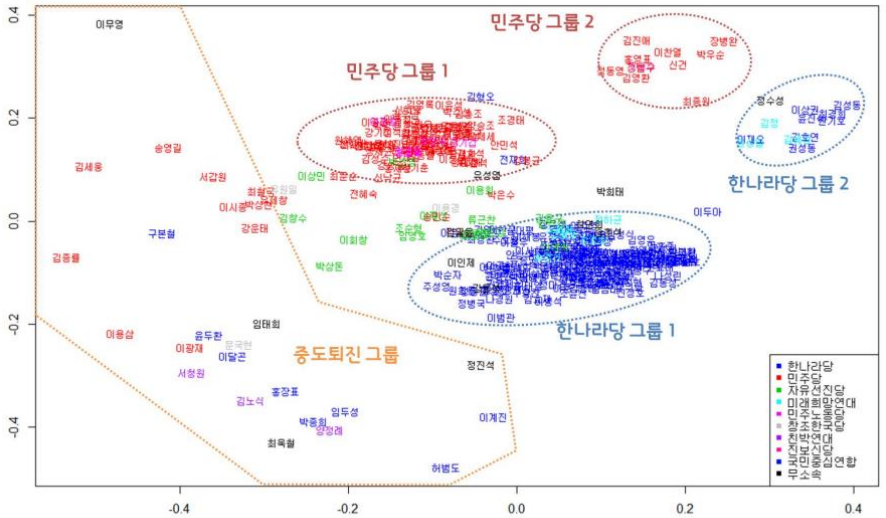
흩어져 있는 정도가 중요하다면 PCA, 거리를 보존하는 것이 중요하다면 MDS

3. Linear Feature extraction

MDS 활용

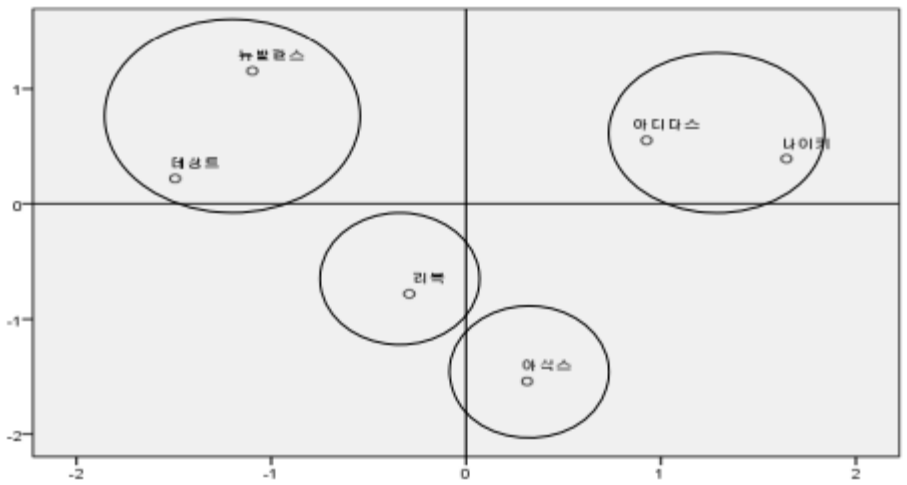
- MDS는 개체들간의 거리를 저차원에서 그대로 보존하도록 mapping.
- 거리 대신에 **다양한 유사도 지표를 이용**한다면(ex. 코사인 유사도) **개체들간의 유사성을 저차원에서 시각적으로 볼 수 있음**.

유사한 단어 시각화



강필성, 박영준, 조수곤, 김성범. (2013). 대한민국 18대 국회의원 의정활동 분석, 한국경영과학회 추계학술대회

기업 유사도 분석 → 마케팅에 활용



3. Linear Feature extraction

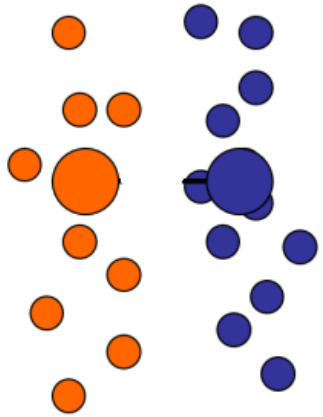
		기존 입력변수 어떻게 조합할 것인가?	
		선형 결합	비선형 결합
Y라벨을 고려하여 특징을 생성할 것인가?	Unsupervised	PCA MDS	KPCA Isomap LLE t-SNE AutoEncoder
	Supervised	LDA	KFD

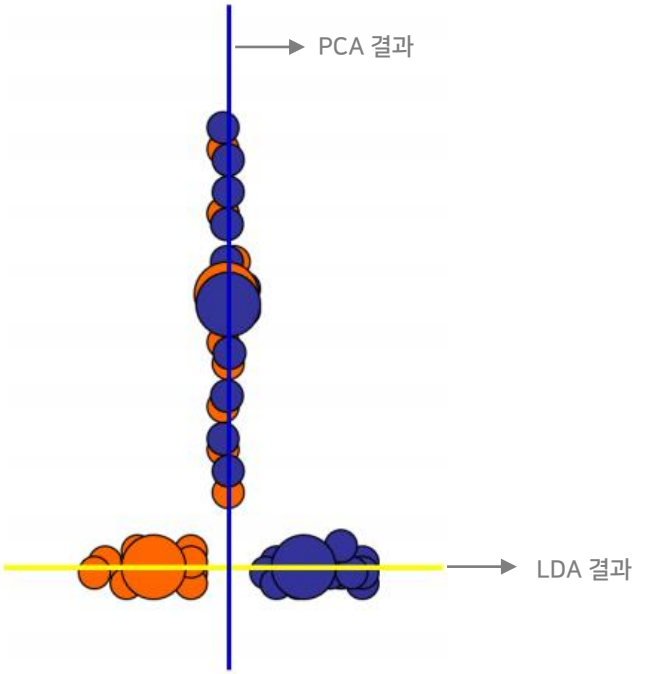
LDA(Linear Discriminant Analysis)

- **기존 입력변수들의 선형결합을** 통해 새로운 특징을 만들어냄
- Y라벨을 고려한 (supervised) 특징 추출. 즉, **데이터의 클래스를 잘 구분하도록 하는 특징을** 뽑아냄.

PCA vs LDA

- PCA는 unsupervised 차원 축소 방법.
  분산을 최대화 하는 특징 구하기
- LDA는 supervised 차원 축소 방법.
  클래스 구분이 잘 되도록 하는 특징 구하기





3. Linear Feature extraction

LDA 수행방법

데이터의 class를 잘 구분해주는 축을 찾아 projection하자

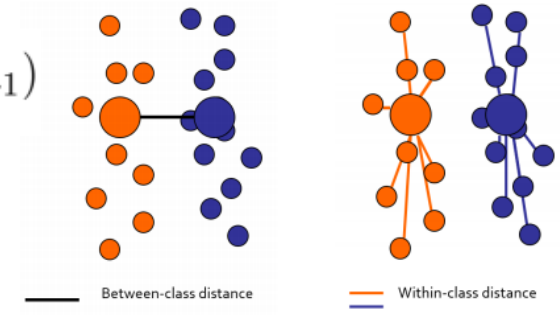
= 저차원으로 projection된 이후의 Between-class distance 최대화, Within-class distance 최소화 되도록

Between-class distance = 두 class의 centroid 간의 거리

Within-class distance = 각 개체들로부터 centroid까지의 평균 거리

$$m_2 - m_1 = \mathbf{w}^T(\mathbf{m}_2 - \mathbf{m}_1)$$

$$s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2$$



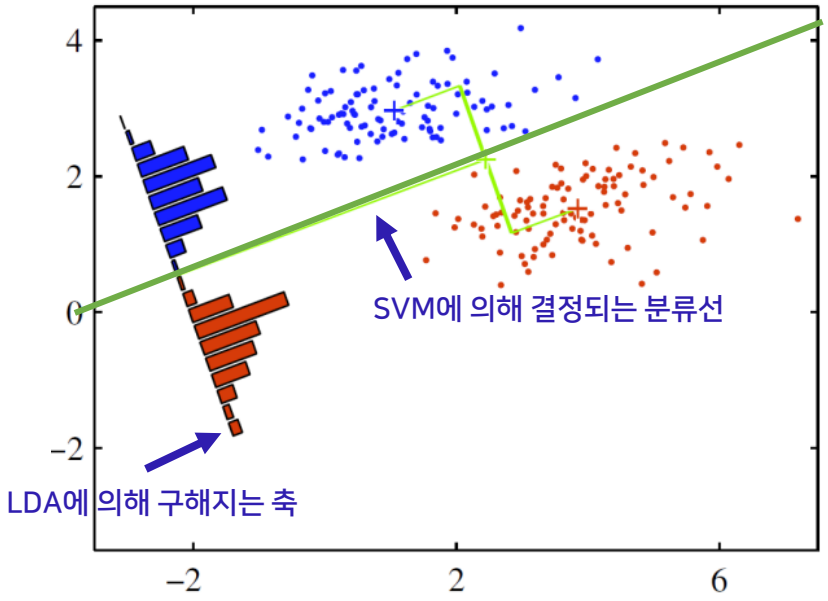
$$\max_{w} J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$$

$$\mathbf{S}_W = \sum_{n \in C_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in C_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T$$

3. Linear Feature extraction

LDA vs SVM

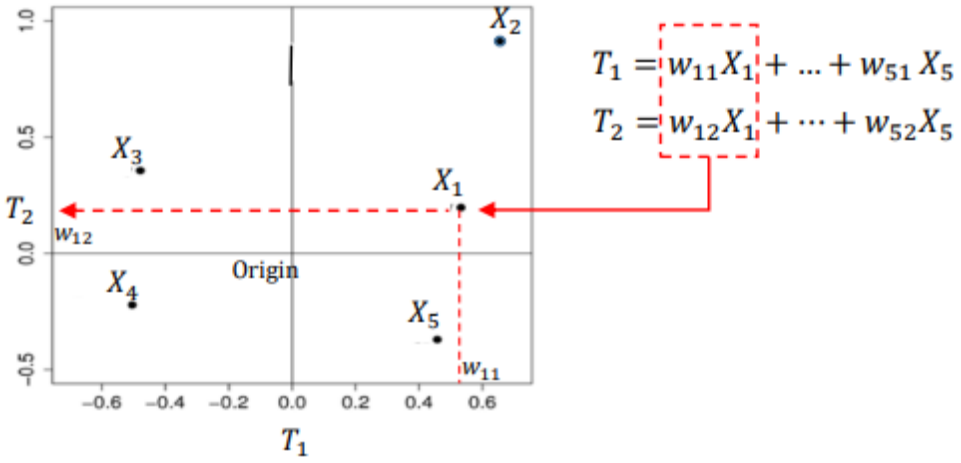


LDA	SVM	
y라벨이 있는 데이터를 학습하는 supervised learning		
Projection 했을 때 class끼리 최대한 분리되도록 하는 축	Original space에서 class를 구분짓는 선	수직인가?
모든 관측치들의 정보를 반영	Boundary 근처 관측치들만 고려하여 분류선 결정	

3. Linear Feature extraction

LDA 활용

- 입력변수 전처리 과정으로 이용하여 예측/분류 모델의 **성능 향상에** 기여
- 저차원에서 시각화
- 클래스 구분 짓는 중요 변수 확인(like PCA)



Loading Plot

3. Linear Feature extraction

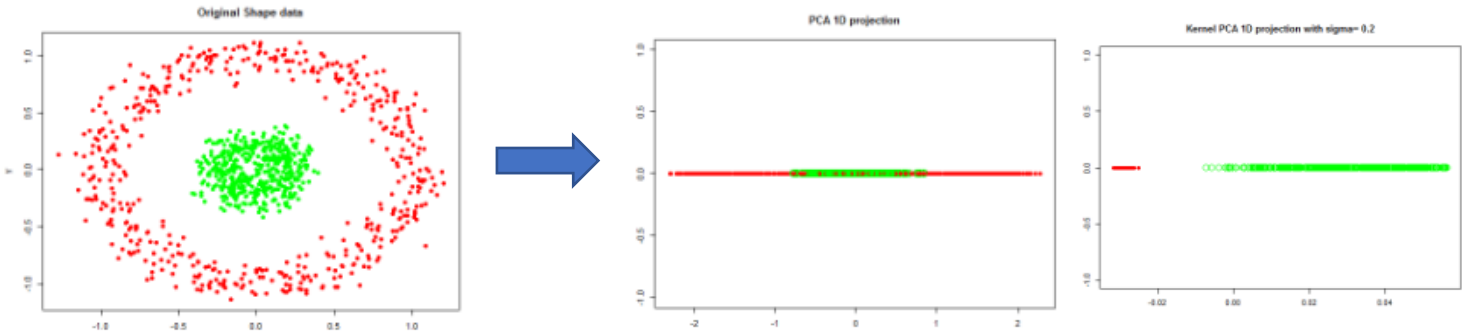
기존 입력변수들의 조합으로 새로운 특징을 생성함.

		기존 입력변수 어떻게 조합할 것인가?	
		선형 결합	비선형 결합
Y라벨을 고려하여 특징을 생성할 것인가?	Unsupervised	PCA MDS	KPCA Isomap LLE t-SNE AutoEncoder
	Supervised	LDA	KFDA

4. Nonlinear Feature extraction

Why 비선형 결합을 통한 특징 추출?

기하학적으로 데이터가 선형이 아닐 경우에는 구조를 명확하게 파악하지 못하는 문제점



선형차원축소(PCA) 비선형 차원축소

Manifold learning

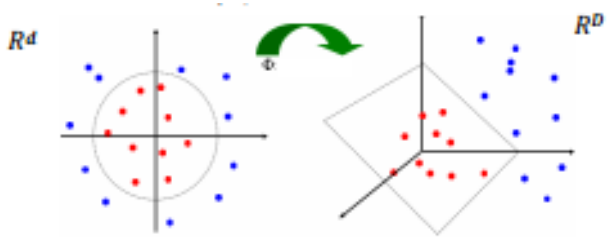
- 비선형 차원 축소 방법을 의미
- 고차원 데이터가 있을 때 sample들을 잘 아우르는 subspace가 있을 것이라는 가정하에 학습을 진행

4. Nonlinear Feature extraction

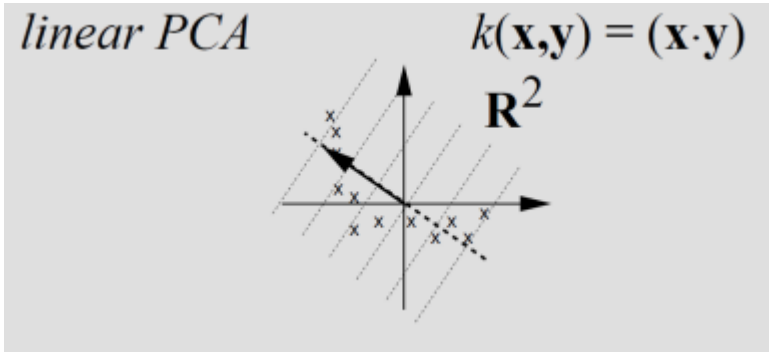
KPCA 커널 함수를 이용해 고차원에서 PCA를 진행

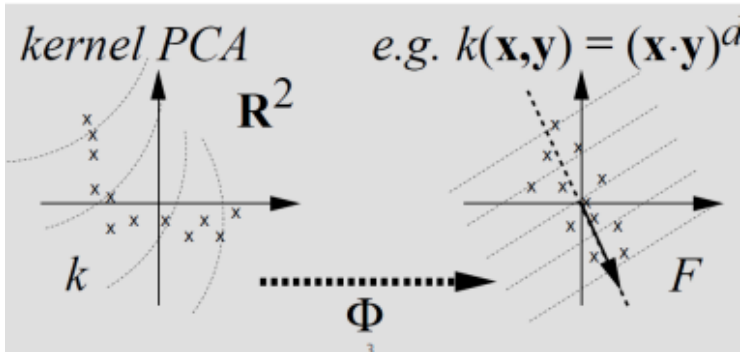
Remind 비선형 SVM

- 저차원에서 하나의 hyperplane으로 데이터를 분류할 수 없는 경우
- 데이터 포인트를 고차원으로 보내면 고차원 공간에서는 하나의 hyperplane으로 구분할 수 있다
- 고차원에서의 hyperplane을 저차원에서 그려본다면 마치 비선형 분류선이 그어진 것 처럼 보인다



데이터들의 임베딩 공간이 선형적이지 않은 경우





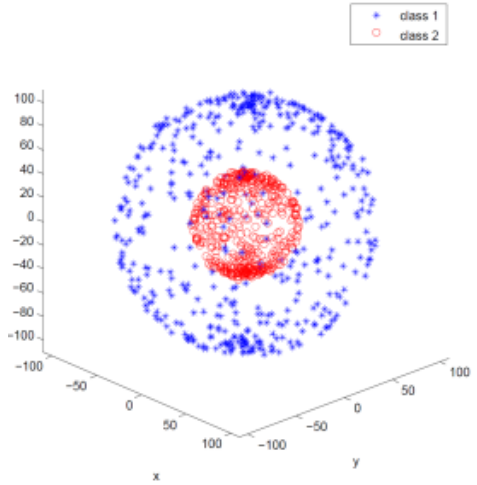
데이터의 분산을 가장 크게 하는 직선(선형결합)

고차원에서 분산을 가장 크게 하는 직선 (고차원에서 선형결합)
= 저차원에서 보면 곡선

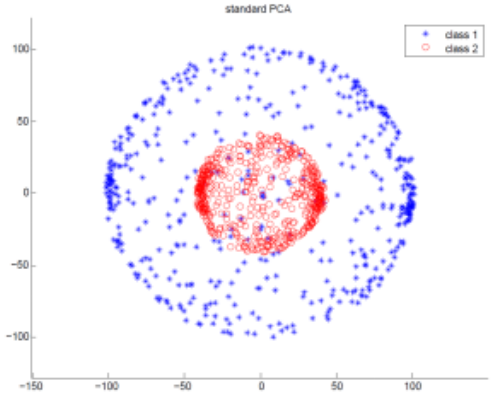
$$\mathbf{v}_k = \sum_{i=1}^{N} \alpha_{ki} \Phi(\mathbf{x}_i)$$

4. Nonlinear Feature extraction

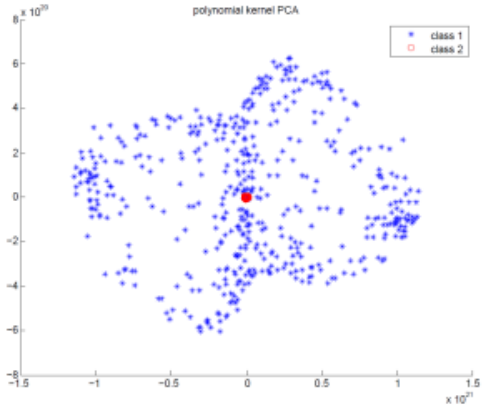
KPCA 예시



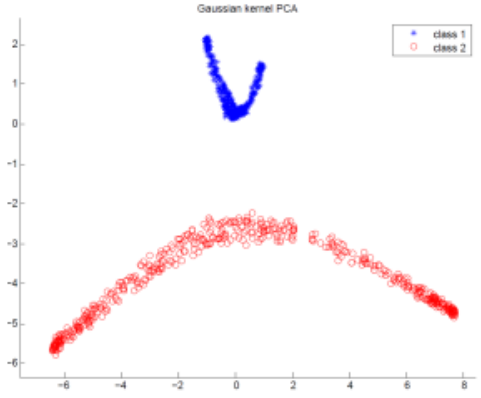
3차원 original 공간
2차원으로 축소할 것!



PCA



K-PCA
(다항함수 커널 사용)



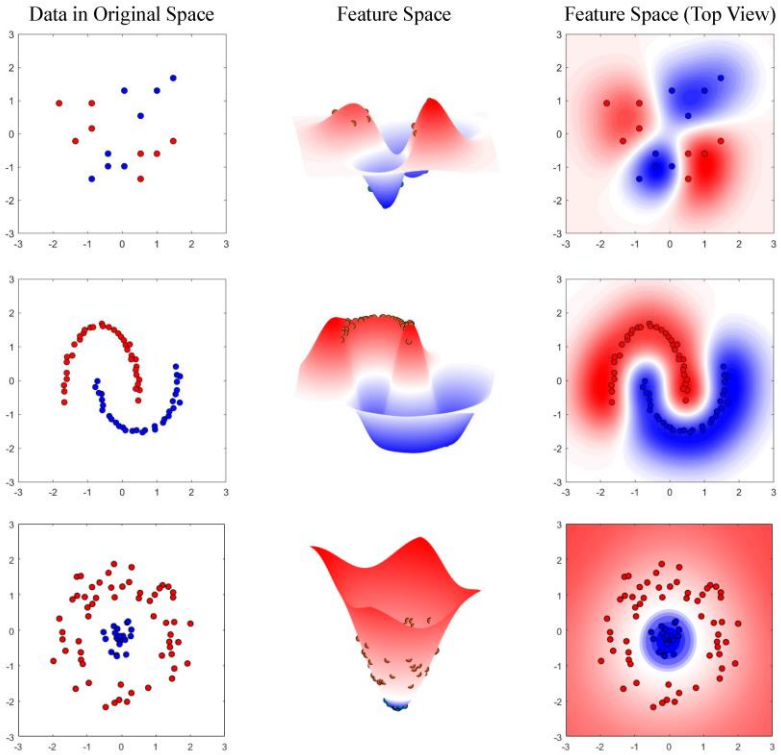
K-PCA
(가우시안 커널 사용)

4. Nonlinear Feature extraction

		기존 입력변수 어떻게 조합할 것인가?	
		선형 결합	비선형 결합
Y라벨을 고려하여 특징을 생성할 것인가?	Unsupervised	PCA MDS	KPCA Isomap LLE t-SNE AutoEncoder
	Supervised	LDA	KFD

KFDA 커널 함수를 이용해 고차원에서 LDA를 진행

$$x_i \rightarrow \Phi(x_i)$$



$$\mathbf{v}_k = \sum_{i=1}^{N} \alpha_{ki} \Phi(\mathbf{x}_i)$$

고차원으로 보내서 class를 잘 구분 짓도록 선형결합하자

= 고차원에서 class를 잘 구분짓는 특징을 만들자

= 저차원에서 class를 잘 구분짓는 비선형적 특징으로 작동한다

4. Nonlinear Feature extraction

		기존 입력변수 어떻게 조합할 것인가?	
		선형 결합	비선형 결합
Y라벨을 고려하여 특징을 생성할 것인가?	Unsupervised	PCA MDS	KPCA Isomap LLE t-SNE AutoEncoder
	Supervised	LDA	KFD

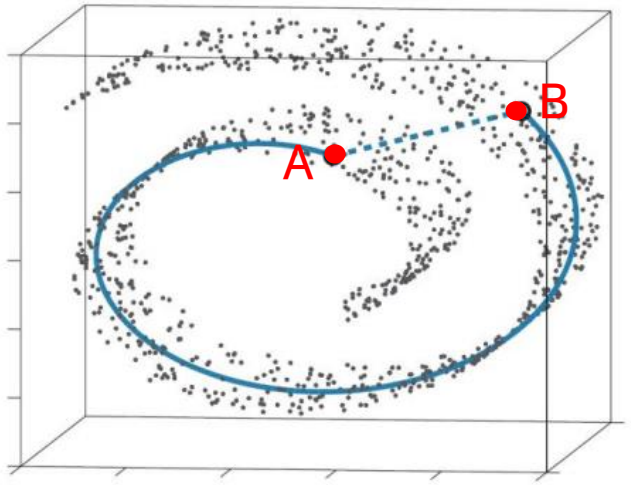
Isomap

MDS(다차원 축도법)의 확장판

1) 데이터 포인터들 사이의 distance matrix 구축

   MDS와 다른 방식으로 distance matrix를 구축한다는 점에서 MDS의 확장판

2) Distance matrix로부터 모든 pairwise 거리를 잘 보존하도록 하는 저차원의 공간 구하기



어떤 방식으로 distance matrix를 구축하는가?

- 유클리디안 거리(점선표시)가 데이터를 잘 반영하는 '실질적인' 거리인가? X
- 데이터 모양대로 넓게 펼쳐보았을 때의 거리(실선 표시)가 '실질적인' 거리
- **데이터의 모양을 반영해서 측정한 곡선 거리(=지오데식 거리)를 기준으로 distance matrix 구축**

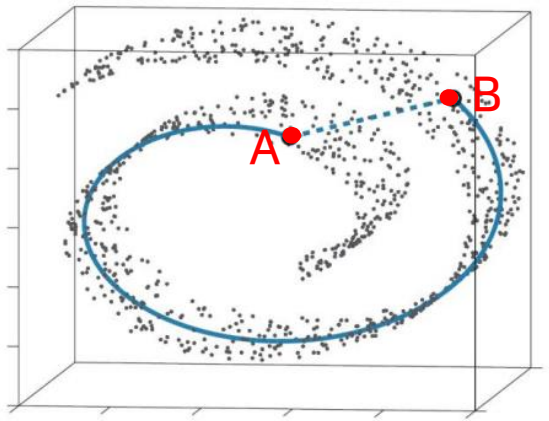
4. Nonlinear Feature extraction

Isomap

어떤 방식으로 distance matrix를 구축하는가?

- MDS는 유클리디안 거리를 이용한 distance matrix
- Isomap은 지오데식 거리를 이용한 distance matrix

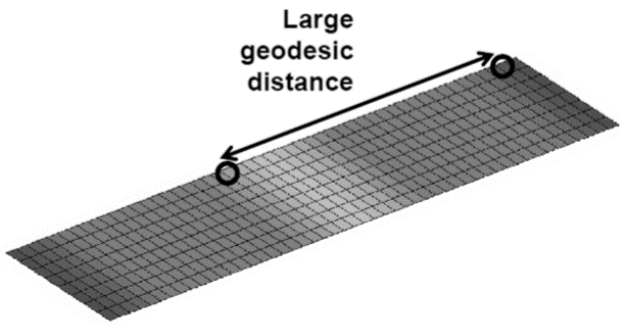
원본 데이터의 고유한 기하학적 구조를 학습할 필요가 있다면
지오데식 거리를 활용한 Isomap을 이용하는 것이 바람직



Original data space

차원축소 시켜보자

MDS



ISOMAP

4. Nonlinear Feature extraction

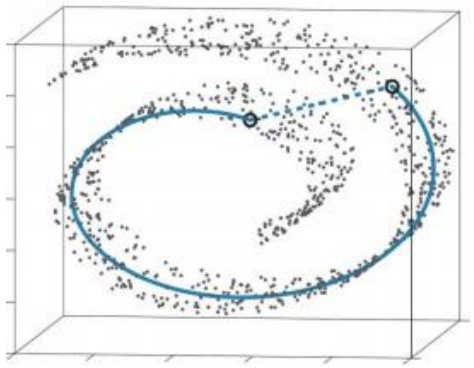
Isomap

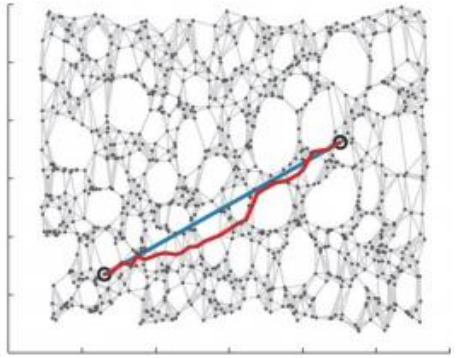
어떻게 지오데식 거리를 구하는가?

*지오데식 거리 = 데이터 공간의 표면을 타고 움직이는 곡선 거리 = 실질적인 거리

1. 가까이 존재하는 이웃들에 edge를 그어서 graph를 만들어준다
   즉, **데이터의 기하학적 구조를 graph로 나타내는 과정**.
   (epsilon-isomap) 반경 epsilon 안에 있는 모든 이웃들과 edge로 연결
   (k-isomap) 가장 가까운 k개의 이웃을 연결

2. **Graph에서 두 노드(데이터 포인트)사이의 최단거리를 구하는 알고리즘** 이용해서 distance matrix 채우기





3. 이후에 distance matrix의 정보를 최대한 유지하도록
저차원 공간으로 mapping... (MDS와 똑같음)

4. Nonlinear Feature extraction

		기존 입력변수 어떻게 조합할 것인가?	
		선형 결합	비선형 결합
Y라벨을 고려하여 특징을 생성할 것인가?	Unsupervised	PCA MDS	KPCA Isomap LLE t-SNE AutoEncoder
	Supervised	LDA	KFD

LLE(Locally Linear Embedding)

- 데이터들의 비선형 특징을 학습해야 한다
- **비선형성도 local하게 보면 선형성**이 나타난다
- 즉, **local한 정보들**을 이용해서 차원 축소한다면 비선형 특징이 잘 반영될 것이다

4. Nonlinear Feature extraction

LLE 수행방법

1. 각각 데이터 포인트의 이웃들을 찾자
2. 해당 데이터 포인트를 이웃들의 선형 결합으로 잘 구축할 수 있는 **가중치 벡터(w)를 구하자**

$$\min_W E(\mathbf{W}) = \sum_i \left| \mathbf{x}_i - \sum_j \mathbf{W}_{ij}\mathbf{x}_j \right|^2$$

$$s.t.\ \mathbf{W}_{ij} = 0 \text{ if } \mathbf{x}_j \text{ does not belong to the neighbor of } \mathbf{x}_i$$

$$\sum_j \mathbf{W}_{ij} = 1 \text{ for all } i$$

3. 가중치가 잘 보존되도록 하는 **저차원의 좌표계(y)를 구하자**

   낮은 차원에서도 최대한 유사한 가중치로 나와 이웃들의 관계를 표현할 수 있도록

$$\min_{\mathbf{y}} \Phi(\mathbf{W}) = \sum_i \left| \mathbf{y}_i - \sum_j \mathbf{W}_{ij}\mathbf{y}_j \right|^2 \quad s.t.\ \sum_i \mathbf{y}_i = 0,\ \frac{1}{n}\sum_i \mathbf{y}\mathbf{y}^T = \mathbf{I}$$

4. Nonlinear Feature extraction

		기존 입력변수 어떻게 조합할 것인가?	
		선형 결합	비선형 결합
Y라벨을 고려하여 특징을 생성할 것인가?	Unsupervised	PCA MDS	KPCA Isomap LLE t-SNE AutoEncoder
	Supervised	LDA	KFD

t-SNE(t-stochastic neighbor embedding)

- 데이터들의 비선형 특징을 학습해야 한다
- 고차원 공간에서 **비슷한 데이터 구조는 저차원 공간에서 가깝게** 대응하며, **비슷하지 않은 데이터 구조는 멀리 떨어져** 대응되도록 만든다
- 저차원에서도 이웃 데이터 포인트에 대한 정보를 보전한다

4. Nonlinear Feature extraction

t-SNE(t-stochastic neighbor embedding)

$$p_{j|i} = \frac{e^{-\frac{||x_i - x_j||^2}{2\sigma_i^2}}}{\sum_{k \neq i} e^{-\frac{||x_i - x_k||^2}{2\sigma_i^2}}}$$

$p_{j|i}$= 원래 차원에서 i의 이웃으로 j를 뽑을 확률

가우시안 분포를 사용해 이웃을 뽑을 확률을 정의

데이터 포인트 i와 가까이 위치할 수록 이웃으로 뽑힐 확률이 커진다

고차원에서 **비슷한 데이터 구조는 저차원에서 가깝게** 대응하며, **비슷하지 않은 데이터 구조는 멀리 떨어지도록**

= 고차원(original)에서 i의 이웃으로 j 뽑을 확률과 저차원에서 i의 이웃으로 j 뽑을 확률을 비슷하도록 만들자

$p_{j|i}$ $q_{j|i}$

4. Nonlinear Feature extraction

t-SNE(t-stochastic neighbor embedding)

SNE

고차원에서 **비슷한 데이터 구조는 저차원에서 가깝게** 대응하며, **비슷하지 않은 데이터 구조는 멀리 떨어지도록**

= 고차원(original)에서 i의 이웃으로 j 뽑을 확률과 저차원에서 i의 이웃으로 j 뽑을 확률을 비슷하도록 만들자

$$p_{j|i} \qquad q_{j|i}$$

Symmetric SNE

= 고차원(original)에서 i와 j 사이의 pairwise 확률과 저차원에서 i와 j 사이의 pairwise 확률이 비슷하도록 만들자

$$\frac{p_{j|i} + p_{i|j}}{2n} \qquad \frac{q_{j|i} + q_{i|j}}{2n}$$

4. Nonlinear Feature extraction

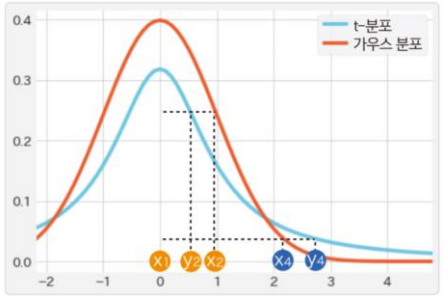
t-SNE(t-stochastic neighbor embedding)

고차원에서 **비슷한 데이터 구조는 저차원에서 가깝게** 대응하며, **비슷하지 않은 데이터 구조는 멀리 떨어지도록**

= 고차원(original)에서 i와 j 사이의 pairwise 확률과 저차원에서 i와 j 사이의 pairwise 확률이 비슷하도록 만들자

앞에서 정의한 그대로
가우시안 분포 이용한 확률

가우시안 분포 대신에
자유도가 1인 t분포를 이용



Why? 가우시안 분포를 사용했을때 일정 부분을 벗어나면 너무 적은 확률을 가지게 되는 문제점을 해결하기 위하여 꼬리부분이 두터운 t분포를 사용한다

4. Nonlinear Feature extraction

t-SNE(t-stochastic neighbor embedding)

고차원에서 **비슷한 데이터 구조는 저차원에서 가깝게** 대응하며, **비슷하지 않은 데이터 구조는 멀리 떨어지도록**

= 고차원(original)에서 i와 j 사이의 pairwise 확률과 저차원에서 i와 j 사이의 pairwise 확률이 비슷하도록 만들자

고차원에서의 확률분포와 저차원에서의 확률분포가 비슷하도록

= 고차원과 저차원에서의 **차이를 최소화**하도록

= 두 확률분포 간의 KL distance가 최소화되도록

= 고차원과 저차원의 유사도 분포가 비슷해지도록

**KL distance는 한 확률분포가 두번째 예상 확률분포와 어떻게 다른지 측정하는 척도

$$C \equiv \sum_i KL(P_i \mid Q_i) = \sum_i \sum_j p_{ij} \log p_{ij} - p_{ij} \log q_{ij}$$

Gradient descent 방식 사용

4. Nonlinear Feature extraction

Manifold learning 총정리

고차원으로 보내서 비선형성을 반영한다

- PCA → KPCA
- LDA → KFDA

Local한 정보들을 활용해서 비선형성을 반영한다

- ISOMAP

  local한 정보를 통해 실질적인 거리 정보를 반영한다

  → 실질적인 거리 정보가 잘 보존되도록 한다
- LLE

  저차원에서도 (k개의) 이웃들과의 관계가 잘 보존되도록 한다
- T-SNE

  저차원에서 (확률적인) 이웃들과의 관계가 잘 보존되도록 한다

4. Nonlinear Feature extraction

데이터의 특성에 맞게 알맞은 차원 축소 방법론을 사용해보자

		기존 입력변수 어떻게 조합할 것인가?	
		선형 결합	비선형 결합
Y라벨을 고려하여 특징을 생성할 것인가?	Unsupervised	PCA MDS	KPCA Isomap LLE t-SNE AutoEncoder
	Supervised	LDA	KFD

5. 차원 축소 활용

- 입력변수의 공간을 축소시킴으로써 다양한 예측/분류 모델의 **성능 향상에** 기여할 수 있다
  다른 방법론들을 위한 일종의 **전처리 과정**으로 사용된다

- 데이터 포인트의 정보들을 최대한 보존하도록 저차원에 표현할 수 있다
  → 관측치들 간의 관계를 보기 쉽게 **시각화**
  → **클러스터 분석**

5. 차원 축소 활용

프리윌린 프로젝트: **성적 향상**을 위해 어떻게 해야할까?

우리가 가진 데이터

궁금해하는 라벨값인 성적에 대한 지표가 존재하지 X

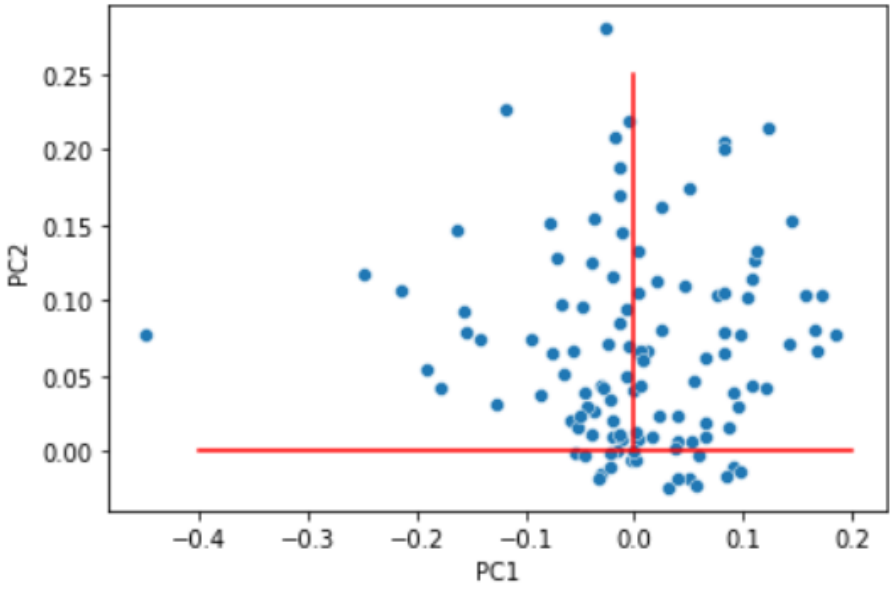
1. 한 사람이 같은 개념, 수준에서 어떤 위치에 있는지 z-score로 표현
2. 비슷한 아이들끼리 clustering
3. **해당 그룹에 있는 학생들을 잘 변별하는 변수(개념과 난도)에 집중**하면 **비슷한 아이들에 비해 더 성적 향상**할 수 있다

5. 차원 축소 활용

프리윌린 프로젝트: **성적 향상**을 위해 어떻게 해야할까?

해당 그룹에 있는 학생들을 잘 변별하는 변수(개념과 난도)에 집중하면
비슷한 아이들에 비해 더 성적 향상할 수 있다

← PCA 이용



성적 높은 그룹의 로딩플롯

원점에서 멀리 떨어져서 분포하는 개념, 난이도들이
해당 군집 학생들의 성적을 변별하는 문제.

해당 군집에서 성적 향상하기 위해서는 어디를 **집중공략!**

6. Summary

- Why 차원축소?

  차원의 저주 해결하기 위해서(과적합, 시각화 어려움, …)

- 어떠한 방법이 존재하는가?
  - Feature selection: mRMR, SVM-RFE, Ridge(Lasso)
  - Feature extraction:

	선형 결합	비선형 결합
Unsupervised	PCA MDS	KPCA Isomap LLE t-SNE AutoEncoder
Supervised	LDA	KFD

- 어떻게 활용할 수 있는가?

  성능향상, 저차원에 시각화(사후에 군집분석 등등)

6. Reference

- 김창욱 교수님 <머신러닝과 산업응용> 강의안
- 7기 김경한님 <Unsupervised Learning> 세션 자료
- 고려대 산업경영공학부 DSBA 연구실 <Dimensionality reduction>, <kernel-based learning> 강의안
- 핸즈온 머신러닝
- 박현욱, 강원석 <다차원척도법을 활용한 스포츠신발브랜드 포지셔닝에 관한 연구>
- https://velog.io/@swan9405/%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D-T-SNE-T-distributed-Stochastic-Neighbor-Embedding
- https://gaussian37.github.io/ml-concept-t_sne/
- https://nate9389.tistory.com/1730

DATA SCIENCE LAB

발표자 최윤서 010-9138-9452
E-mail: rangchoi337@naver.com