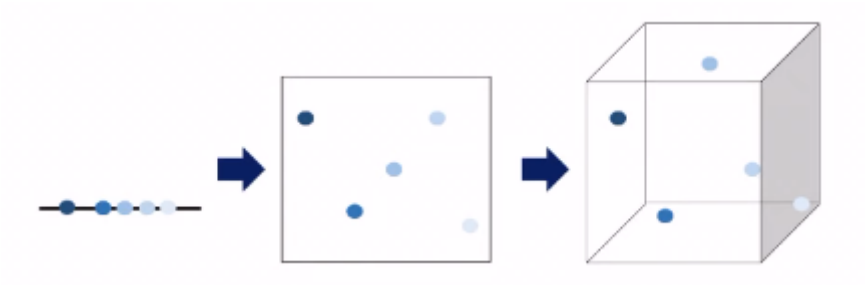
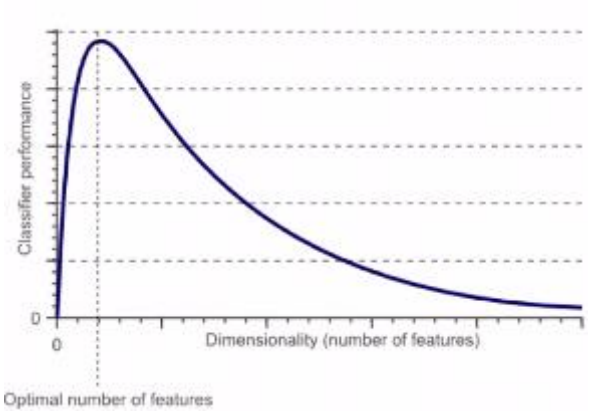
차원 축소 : 고차원의 데이터를 전처리 과정에서 차원을 축소하는 방법. 왜하냐?

차원의 저주 : 데이터의 차원이 증가할수록 동일 정보량을 표현하기 위해 필요한 데이터는 지수적으로 증가하게 되는데, 만약 모델을 학습시키려는 데이터의 수가 Feature(차원)의 수 보다 작아진다면 성능이 저하가 된다.



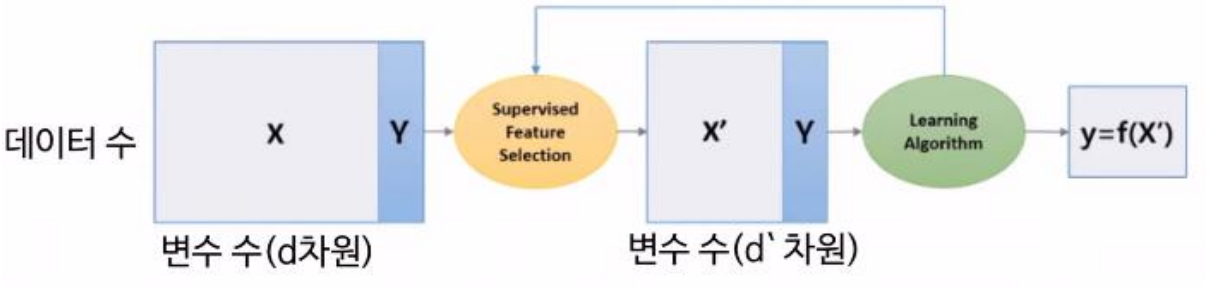
이론적으로, 차원의 증가는 모델의 성능을 향상시킨다고 하는데, 실제로는 모델의 성능이 저하가 되는 경우가 많다. 왜냐하면 변수 간의 상관관계가 있고 차원이 높을수록 노이즈도 증가하기 때문이다. 또한 모델의 학습과 추론의 계산 복잡도가 높아진다. 따라서 모델의 성능을 최대한 뽑아내는 변수의 일부 feature만 사용하는 것이다. 이때 전문 지식으로 중요한 특성을 뽑아내거나 목적함수에 Regularization term을 추가하거나 차원 축소 기술을 전처리로 사용한다.



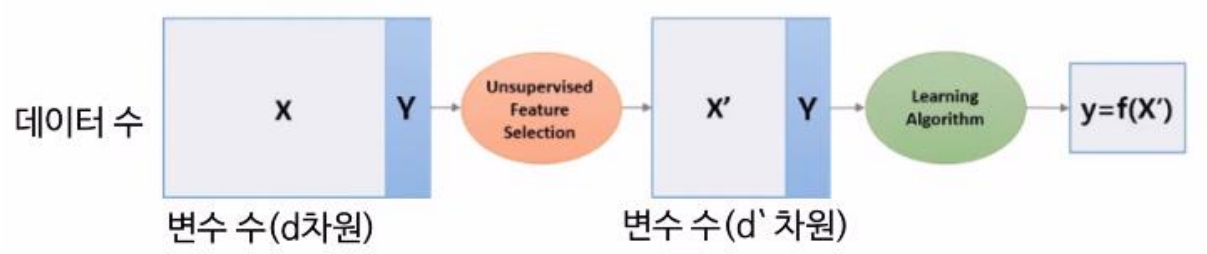
차원 축소 효과:

- 변수간 상관 관계 (correlations) 제거
- 단순한 후처리 (Post-processing)
- 적절한 정보를 유지하면서 중복되거나 불필요한 변수를 제거
- 시각화 가능 (3차원 이하로 축소했을 때 겠지?)

지도학습 기반 차원 축소: 학습결과가 피드백 되어 Feature Selection을 반복함



비지도학습 기반 차원 축소: 지도학습처럼 피드백을 통한 Feature Selection 반복 없음



변수/특성 선택 (Feature selection) : 유의미한 변수만 선택 (도메인 지식)

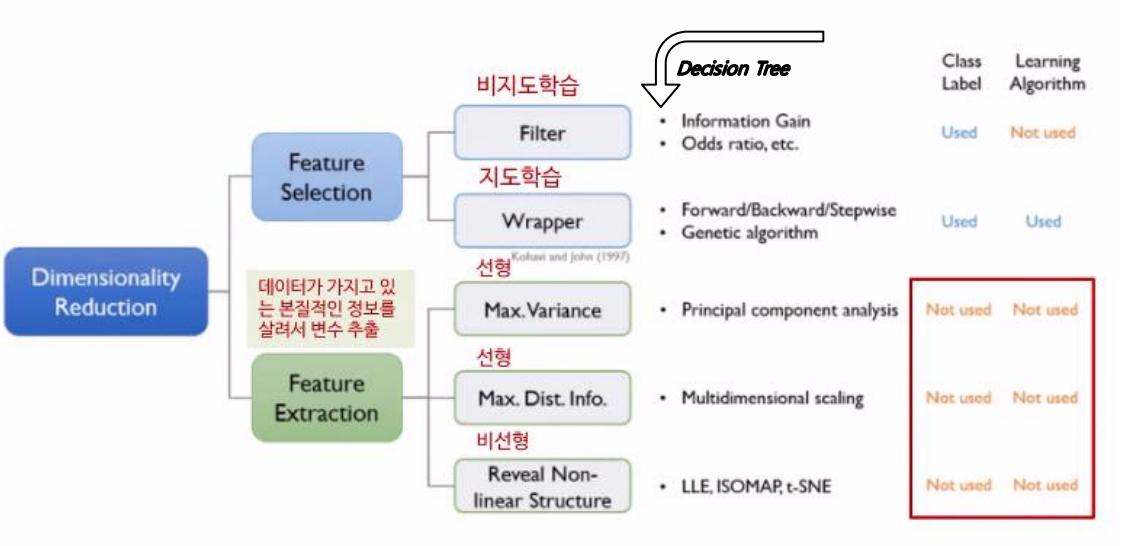
- 장점: 선택한 변수 해석 용이
- 단점: 변수간 상관관계 고려의 어려움

$$x_1, x_2, \ldots, x_{100} \rightarrow x_1, x_5$$

변수/특성 추출 (Feature extraction) : 특성의 선형 결합을 통해 새로운 변수 추출

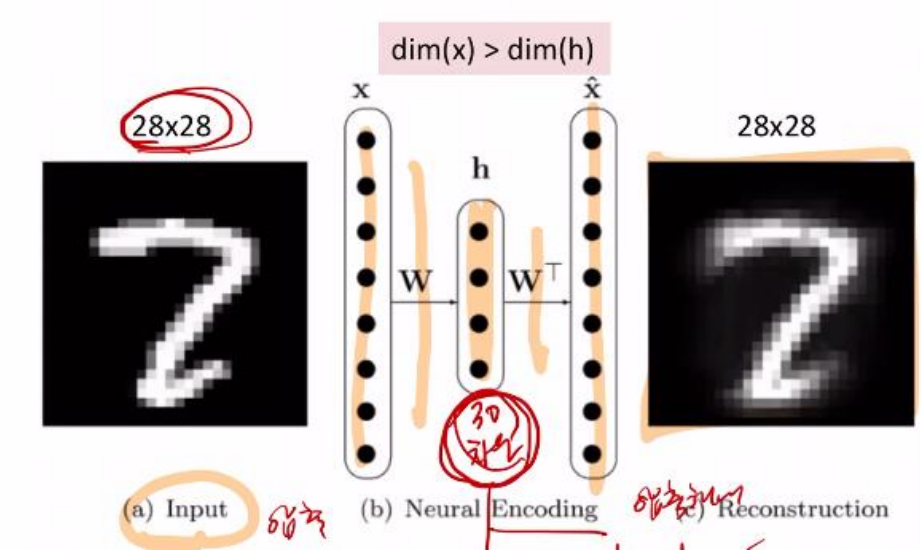
- 변수/특성 생성(Feature construction)이라고도 함
- 장점: 변수간 상관관계 고려, 변수의 개수를 "많이" 줄일 수 있음
- 단점: 추출된 변수의 해석이 어려움

$$z = f(x_1, x_2, \ldots, x_{100})$$

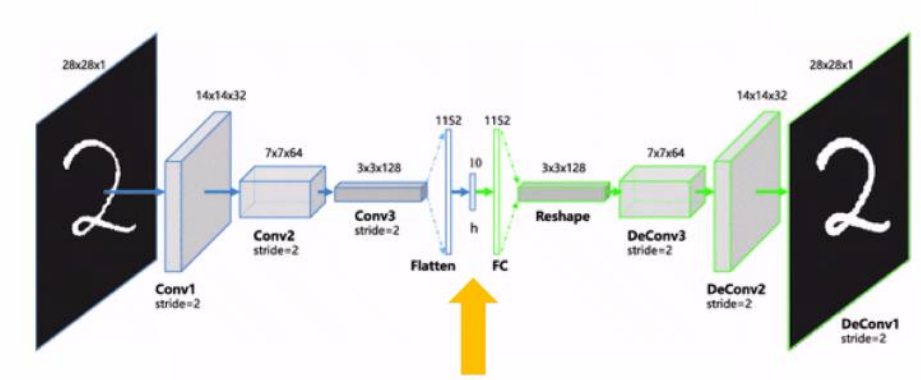


인공신경망 기반의 차원 축소 Representation learning (맛보기)

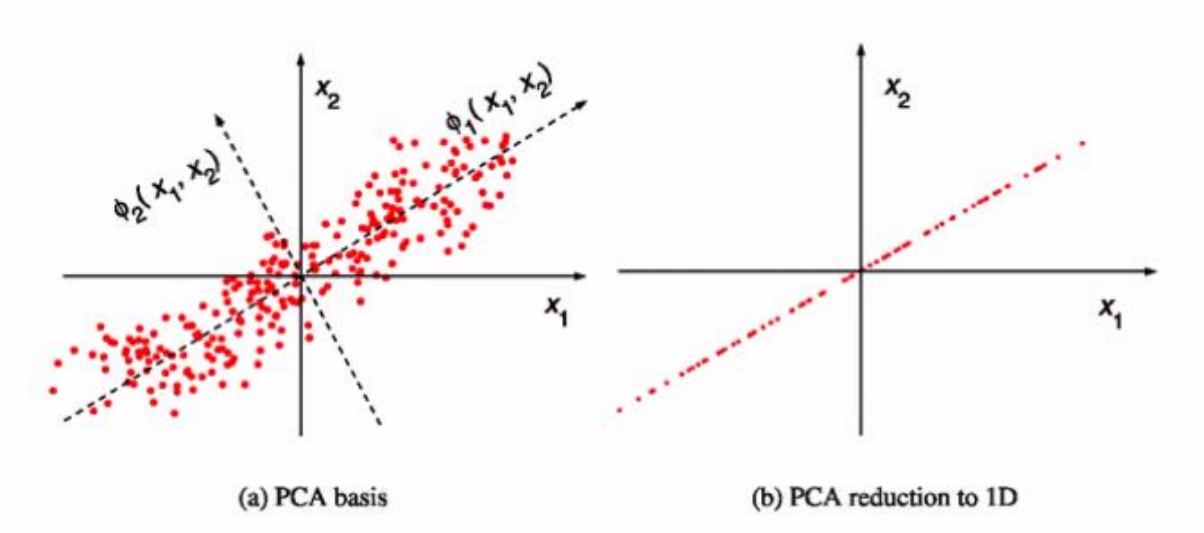
- Deep Auto-Encoder



- Convolution Neural Network



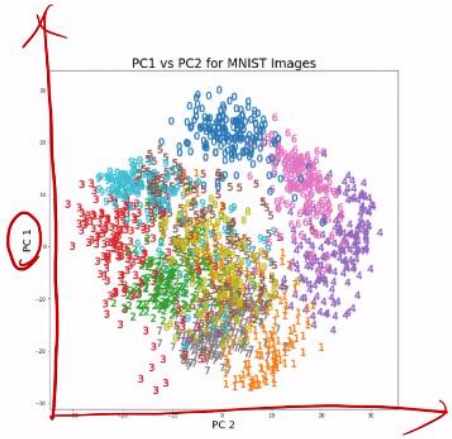
PCA(Principal Component Analysis)： Dimension Reduction의 unsupervised learning 방법 중 한가지. 사영(Projection) 후 원래 데이터의 분산(Variance)를 최대한 보존할 수 있는 기저(축, Principal Component)를 찾아 차원을 줄이는 방법.



(a) PCA basis (b) PCA reduction to 1D

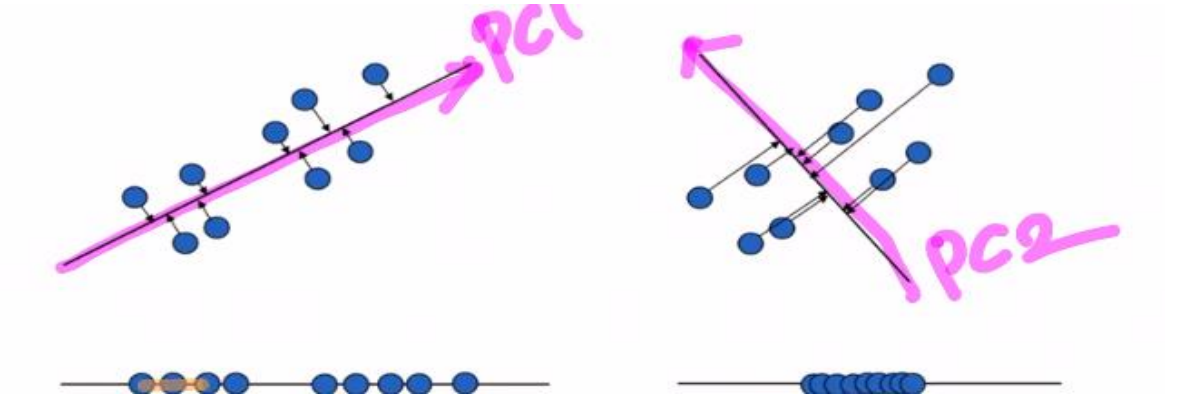
MNIST의 예시

16 * 16 = 256



MNIST PCA

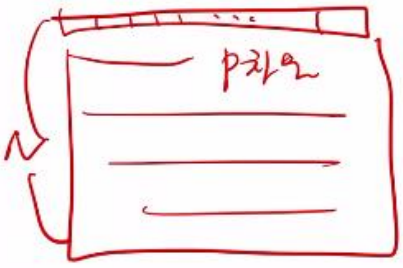
데이터를 사영(Projection)시킬 경우, 손실되는 정보의 양이 적은 쪽의 기저를 선택. 아래의 경우 왼쪽 기저가 오른쪽 기저보다 원 데이터의 분산을 최대로 유지하므로 왼쪽이 더 좋다.



PCA 수리적 배경 : **선형 결합**

데이터(X) projection 후(Z)에도 분산이 보존하는 기저(a)를 찾는 것

$$Z_1 = \alpha_1^T X = \alpha_{11}X_1 + \alpha_{12}X_2 + \cdots + \alpha_{1p}X_p$$
$$Z_2 = \alpha_2^T X = \alpha_{21}X_1 + \alpha_{22}X_2 + \cdots + \alpha_{2p}X_p$$
$$\vdots$$
$$Z_p = \alpha_p^T X = \alpha_{p1}X_1 + \alpha_{p2}X_2 + \cdots + \alpha_{pp}X_p$$



- $X_1, X_2, \ldots, X_p$: 원 데이터 P 개 변수
- $\alpha_i = [\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ip}]$: i 번째 기저(basis) 또는 계수(loading)
- $Z_1, Z_2, \ldots, Z_p$: 각 기저로 사영 변환 후 변수 (주성분)

공분산(Covatiance) : 변수의 상관 정도

X : 입력데이터(n개의 데이터, d개의 변수)

$$Cov(X) = \frac{1}{n}(X - \bar{X})(X - \bar{X})^T$$

[dxd] [dxn] [nxd]

데이터 셋의 전체 분산(Total variance)

➔ $tr[Conv(x)] = Conv(X)_{11} + Conv(X)_{22} + \cdots + Conv(X)_{dd}$

예시)

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 5 & 4 & 1 \\ 3 & 8 & 6 \end{bmatrix}$$

$$Conv(A) = \begin{bmatrix} 2.67 & 0.67 & -2.67 \\ 0.67 & 4.67 & 2.33 \\ -2.67 & 2.33 & 4.67 \end{bmatrix}$$

사영 (Projection)

$$(\vec{b} - p\vec{a})^{\mathrm{T}}\vec{a} = 0 \quad \Rightarrow \quad \vec{b}^{\mathrm{T}}\vec{a} - p\vec{a}^{\mathrm{T}}\vec{a} = 0 \quad \Rightarrow \quad p = \frac{\vec{b}^{\mathrm{T}}\vec{a}}{\vec{a}^{\mathrm{T}}\vec{a}}$$

$$\vec{x} = p\vec{a} = \frac{\vec{b}^{\mathrm{T}}\vec{a}}{\vec{a}^{\mathrm{T}}\vec{a}}\vec{a}$$

If $\vec{a}$ is unit vector

$$p = \vec{b}^{\mathrm{T}}\vec{a} \quad \Rightarrow \quad \vec{x} = p\vec{a} = (\vec{b}^{\mathrm{T}}\vec{a})\vec{a}$$

x: 사영후 벡터, p: 직교 사영을 위한 스케일러
b = 데이터, a = 기저축 (PC)

Mapping 에서 고유벡터는 방향 변화가 없다. (파란 색 벡터)

행렬 A 가 Non-Singular 하다면, d 개의 고유값과 고유벡터가 존재

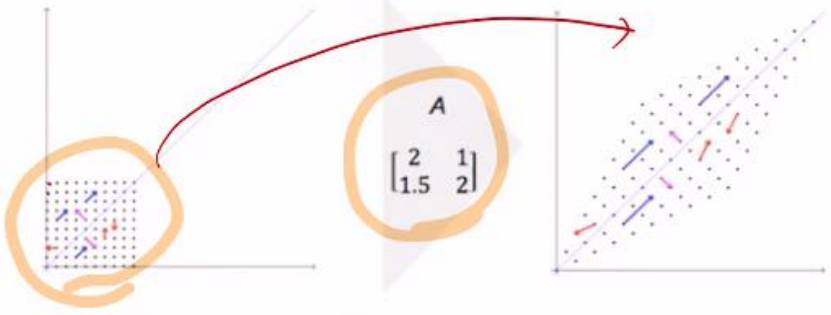
고유벡터는 서로 직교한다.

고유값(eigenvalue)과 고유벡터(eigenvector)

$$\mathbf{Ax} = \lambda\mathbf{x} \quad \rightarrow \quad (\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = 0$$

벡터에 행렬을 곱하는 것은 선형 변환의 의미를 가짐

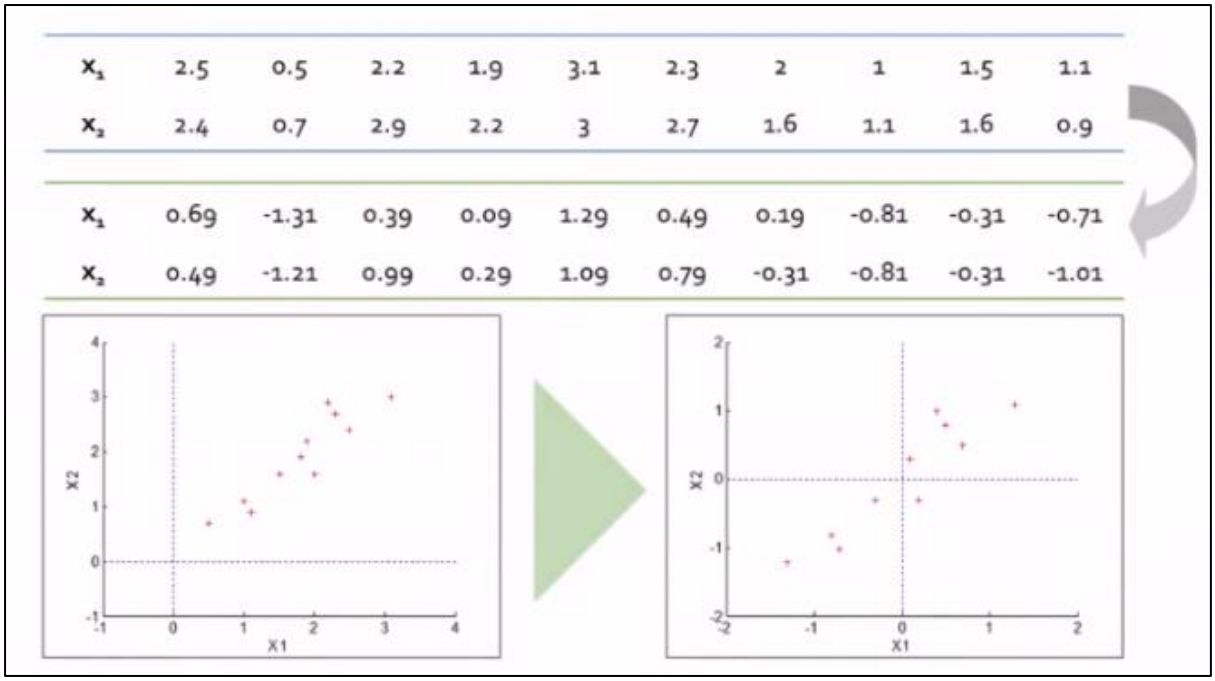
- 고유벡터는 변환에 의해 방향 변화가 발생하지 않음
- 고유벡터의 크기 변화는 λ 만큼



Blue vector, Pink vector ➔ Principal Axis

PCA 과정

Step 1 : 데이터 센터링 (데이터 평균을 0으로 변경)



Step 2 : 최적화 문제 정의

데이터 X를 기저 벡터 W에 사영하면, 사영 후 분산은 다음과 같음. S는 X의 Covariance matrix

$$V = \frac{1}{n}(\mathbf{w}^T\mathbf{X})(\mathbf{w}^T\mathbf{X})^T = \frac{1}{n}\mathbf{w}^T\mathbf{X}\mathbf{X}^T\mathbf{w} = \mathbf{w}^T\mathbf{S}\mathbf{w}$$

PCA의 목적은 사영 이후 분산 V를 최대화 하는 것

$$\max \mathbf{w}^T\mathbf{S}\mathbf{w}$$

$$\text{s.t.}\ \mathbf{w}^T\mathbf{w} = 1$$

$$\Rightarrow\ S = \begin{pmatrix} 0.6166 & 0.6154 \\ 0.6154 & 0.7166 \end{pmatrix}$$

wTw는 w가 단위 벡터 이므로 1

Step 3 : 최적화 문제 솔루션

- 라그랑지 멀티플라이어(Lagrangian multiplier) 적용

$$\max \quad \mathbf{w}^T\mathbf{S}\mathbf{w} \qquad s.t. \quad \mathbf{w}^T\mathbf{w} = 1$$

$$L = \mathbf{w}^T\mathbf{S}\mathbf{w} - \lambda(\mathbf{w}^T\mathbf{w} - 1)$$

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{S}\mathbf{w} - \lambda\mathbf{w} = 0 \Rightarrow (\mathbf{S} - \lambda\mathbf{I})\mathbf{w} = 0$$

W는 S의 eigen vector
λ는 S의 eigen value.

$$Eigenvectors = \begin{bmatrix} 0.6779 & -0.7352 \\ 0.7352 & 0.6779 \end{bmatrix} \qquad Eigenvalues = (1.2840 \quad 0.0491)$$

Step 4 : 주축 정렬

- Eigenvalue 에 해당되는 eigenvectors를 순서대로 정렬

$$Eigenvectors = \begin{bmatrix} 0.6779 & -0.7352 \\ 0.7352 & 0.6779 \end{bmatrix} \qquad Eigenvalues = (1.2840 \quad 0.0491)$$

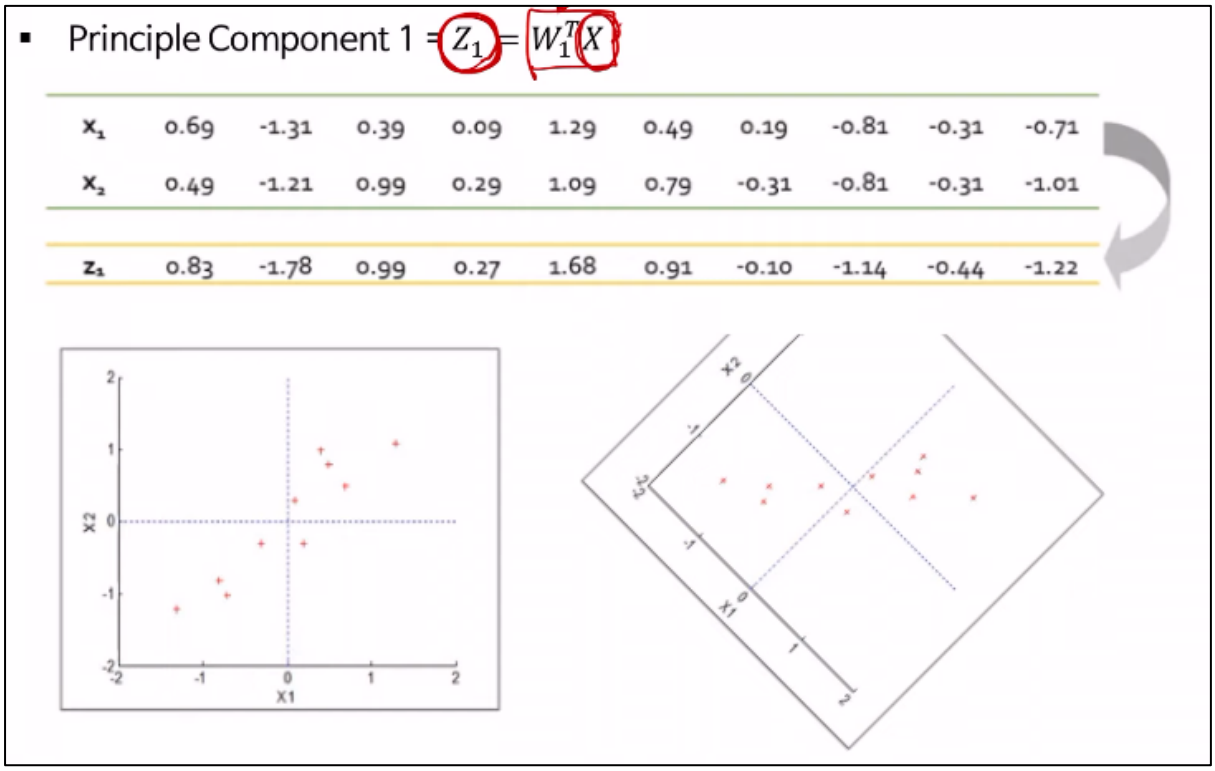
- W_1은 Eigen vector이고 λ_1은 대응되는 eigen value 이다. → basis축하나
- (아래 식의 유도에 의하면) W_1에 사영된 데이터의 분산은 λ_1 이다.

$$\mathbf{v} = (\mathbf{w}_1^T\mathbf{X})(\mathbf{w}_1^T\mathbf{X})^T = \mathbf{w}_1^T\mathbf{X}\mathbf{X}^T\mathbf{w}_1 = \mathbf{w}_1^T\mathbf{S}\mathbf{w}_1$$

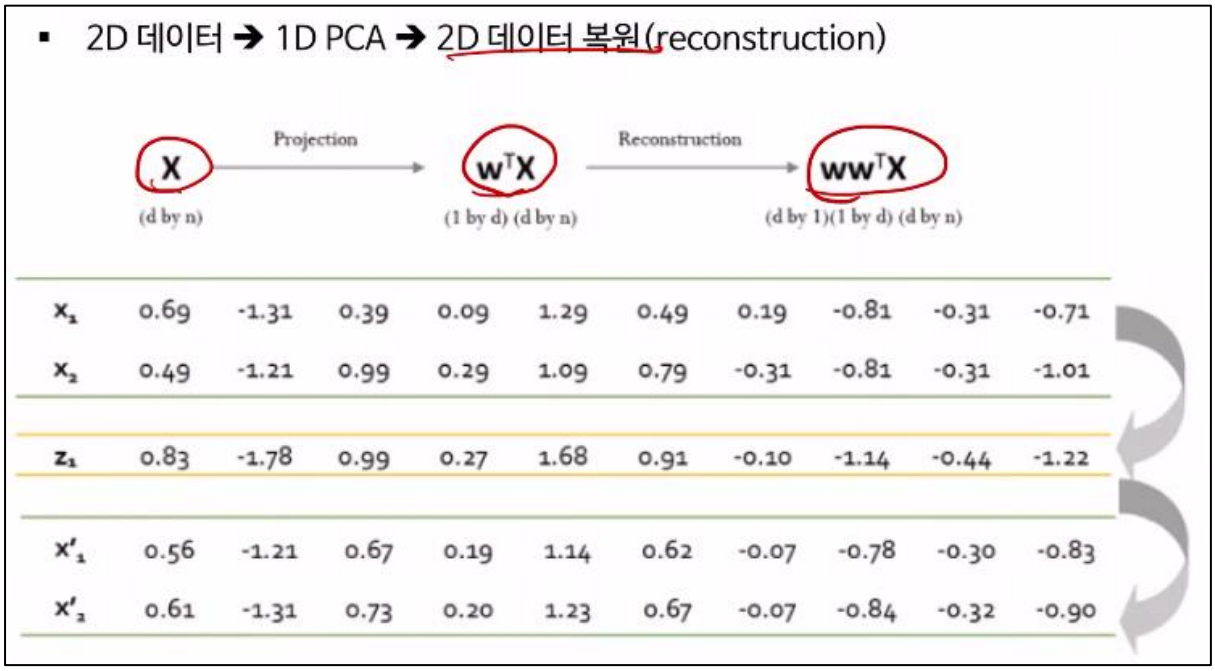
$$\text{Since} \quad \mathbf{S}\mathbf{w}_1 = \lambda_1\mathbf{w}_1, \quad \mathbf{w}_1^T\mathbf{S}\mathbf{w}_1 = \mathbf{w}_1^T\lambda_1\mathbf{w}_1 = \lambda_1\mathbf{w}_1^T\mathbf{w}_1 = \lambda_1$$

- 첫번째 주성분으로 설명가능한 데이터 비율 $= \frac{\lambda_1}{\lambda_1 + \lambda_2} = \frac{1.2840}{1.2840 + 0.0491} = 0.96$

Step 5 : PCA로 변환 된 데이터

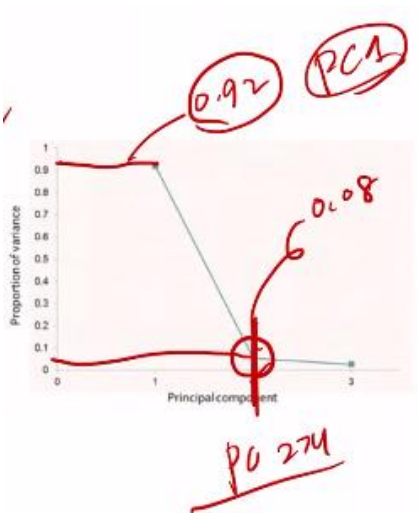


Step 6 : 원데이터로 복원 가능, 근데 원래대로 복원은 안된다.



주성분 개수 선정 법 : 몇 차원으로 줄일 것인가?

- 고유 값 감소율이 유의미하게 낮아지는 Elbow point에 해당하는 주성분을 선택
- 일정 수준 이상의 분산 비를 보존하는 최소의 주성분을 선택 (보통 70% 이상)



PC1 PC2 PC3

첫번째 주성분으로 설명가능한 데이터 비율 $= \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$

$= \frac{2.7596}{0.0786 + 0.1618 + 2.7596}$

$= 0.920$

예시

몇개의 주성분을 사용해야 할까?

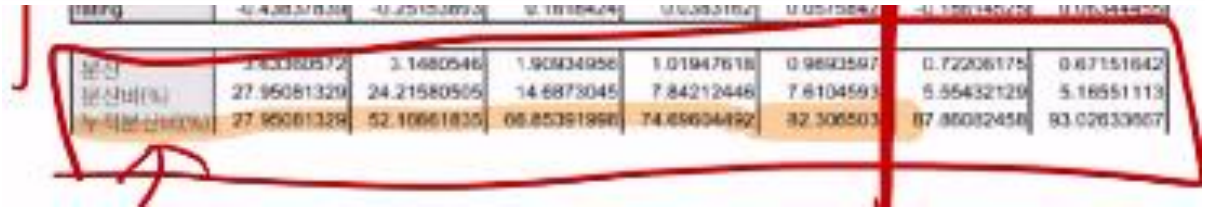
- 조건: 전체 분산의 80%를 포함해야 함

솔루션

- (위) 원 데이터, (아래) PCA 결과
- PCA결과의 누적분산비를 살펴보면 PC1~PC5까지 선택해야 전체 데이터 분산의 82%에 도달할 수 있음

시리얼 이름	제조업체명	유형	칼로리	단백질	지방	나트륨	식이섬유	복합탄수화물	설탕	칼륨	비타민
100% Bran	N	C	70	4	1	130	10	5	6	280	25
100% Natural Bran	Q	C	120	3	5	15	2	8	8	135	0
All-Bran	K	C	70	4	1	260	9	7	5	320	25
All-Bran with Extra Fiber	K	C	50	4	0	140	14	8	0	330	25
Almond Delight	R	C	110	2	2	200	1	14	8		25
Apple Cinnamon Cheerios	G	C	110	2	2	180	1.5	10.5	10	70	25
Apple Jacks	K	C	110	2	0	125	1	11	14	30	25
Basic 4	G	C	130	3	2	210	2	18	8	100	25
Bran Chex	R	C	90	2	1	200	4	15	6	125	25
Bran Flakes	P	C	90	3	0	210	5	13	5	190	25
Cap'n'Crunch	Q	C	120	1	2	220	0	12	12	35	25
Cheerios	G	C	110	6	2	290	2	17	1	105	25
Cinnamon Toast Crunch	G	C	120	1	3	210	0	13	9	45	25
Clusters	G	C	110	3	2	140	2	13	7	105	25
Cocoa Puffs	G	C	110	1	1	180	0	12	13	55	25
Corn Chex	R	C	110	2	0	280	0	22	3	25	25
Corn Flakes	K	C	100	2	0	290	1	21	2	35	25
Corn Pops	K	C	110	1	0	90	1	13	12	20	25
Count Chocula	G	C	110	1	1	180	0	12	13	65	25
Cracklin' Oat Bran	K	C	110	3	3	140	4	10	7	160	25

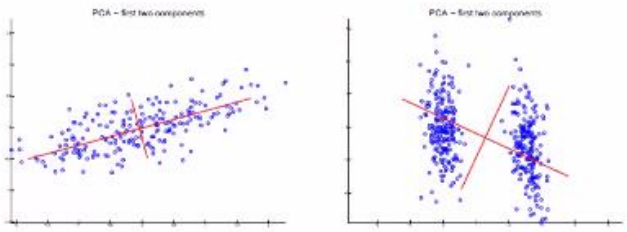
변수이름	1	2	3	4	5	6	7
calories	0.2995424	0.39314792	0.11485746	0.20435865	0.20389862	-0.25580635	-0.02569552
protein	-0.30739639	0.16532333	0.27728197	0.30674316	-0.319749	0.120752	0.28270504
fat	0.03991544	0.34572428	-0.20499509	0.19863317	0.59699332	0.34796733	-0.05115488
sodium	0.16339655	0.13722059	0.38943109	0.12633724	-0.33839424	0.66437215	-0.28370309
fiber	-0.45349041	0.17981192	0.09976604	0.03917367	-0.255119	0.0642436	0.11232537
carbo	0.19244903	-0.14944831	0.56245244	0.0878355	0.18274252	-0.32639283	-0.26046798
sugars	0.22808853	0.35143444	-0.35540519	-0.02270711	-0.31487244	-0.15208226	0.22788519
potass	-0.40196434	0.30054429	0.05762024	0.09087842	-0.14836049	0.02515389	0.14886823
vitamins	0.11598022	0.1729092	0.38785872	-0.6041106	-0.04928682	0.12948574	0.29427618
shelf	-0.17126338	0.26505029	-0.03153102	-0.63887852	0.32910112	-0.05204415	-0.17483434
weight	0.05029929	0.45030847	0.24713831	0.15342878	-0.22128329	-0.39877367	0.01392053
cups	0.29953556	-0.21224795	0.13999909	0.04748911	0.12081645	0.09945091	0.74856957
rating	-0.43837830	-0.25153853	0.1818424	0.0383162	0.05758421	-0.18614525	0.06344455
분산	3.63360572	3.1480546	1.90934956	1.01947618	0.98935974	0.72206175	0.67151642
분산비(%)	27.95081329	24.21580505	14.6873045	7.84212446	7.61045933	5.55432129	5.16551113
누적분산비(%)	27.95081329	52.16661835	66.85391966	74.69604492	82.3065033	87.86082458	93.02633567



PCA의 한계

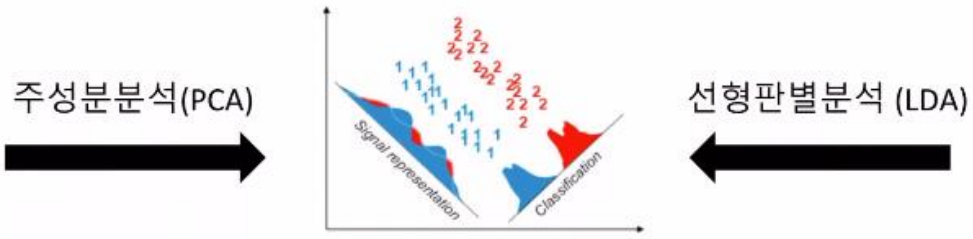
한계점 1

- 데이터 분포가 가우시안(non-gaussian)이 아니거나 다중 가우시안(multimodal-gaussian) 자료들에 대해서는 적용하기 어려움



한계점 2

- 분류 문제를 위해 디자인되지 않음, 즉 분류 성능 향상을 보장하지 못함



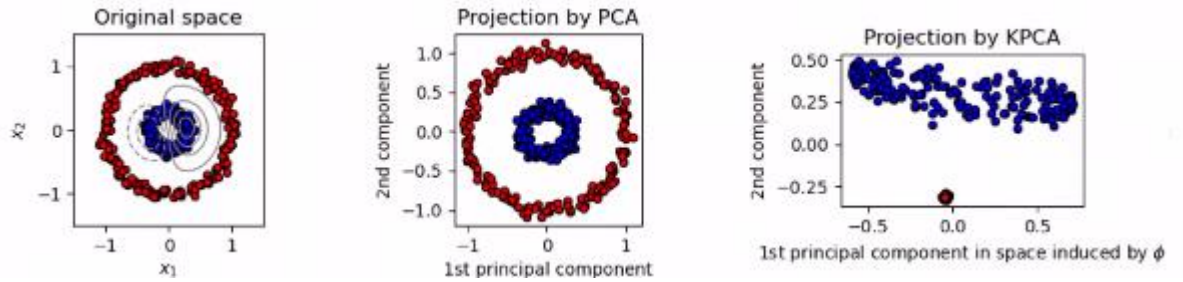
Randomized PCA, Kernelized PCA

Randomized PCA의 개념

- 자료의 크기 또는 특성변수의 크기가 매우 크면 주성분 W를 구하기 위한 SVD(특이값 분해)계산이 불가능하거나 시간이 많이 소요됨
- 이런 경우 유용하다
- QR 분해를 이용하여 행렬의 SVD를 수행한다.

Kernelized PCA의 개념

- PCA는 선형 변환이고 Kernelized PCA는 비선형 변환임
- SVM의 커널 트릭을 PCA에서도 사용
- 특성 변수 x를 비선형 h(x)로 변환 후 이에 대해 PCA를 이용하여 차원 축소


Original space
x_1
x_2
Projection by PCA
1st principal component
2nd component
Projection by KPCA
1st principal component in space induced by ϕ
2nd component