Ch 10. 차원 축소와 척도 학습

10.1 k-최근접 이웃 기법

10.2 임베딩

10.3 주성분 분석

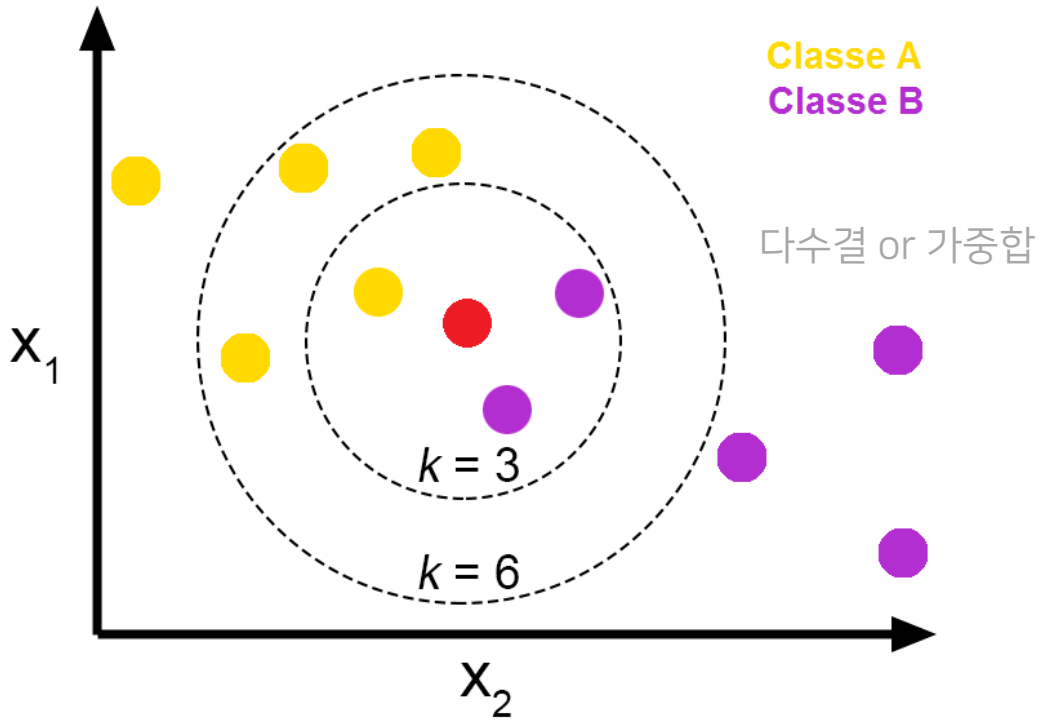
21. 05. 05 김혜수

10.1.1 k-최근접 이웃 기법 K-Nearest Neighbor, KNN

- **k-최근접 이웃 기법 (k-Nearest Neighbor, KNN)**

  새로운 데이터가 주어졌을 때, 기존 데이터 가운데 **가장 가까운 k개 이웃의 정보**를 바탕으로 예측
  거리 척도에 기반

  - 지도 학습 Supervised Learning
  - 게으른 학습 Lazy Learning



10.1.2 KNN의 하이퍼파라미터 Hyperparameter of KNN

▪ 거리 계산법

- 유클리디안 거리

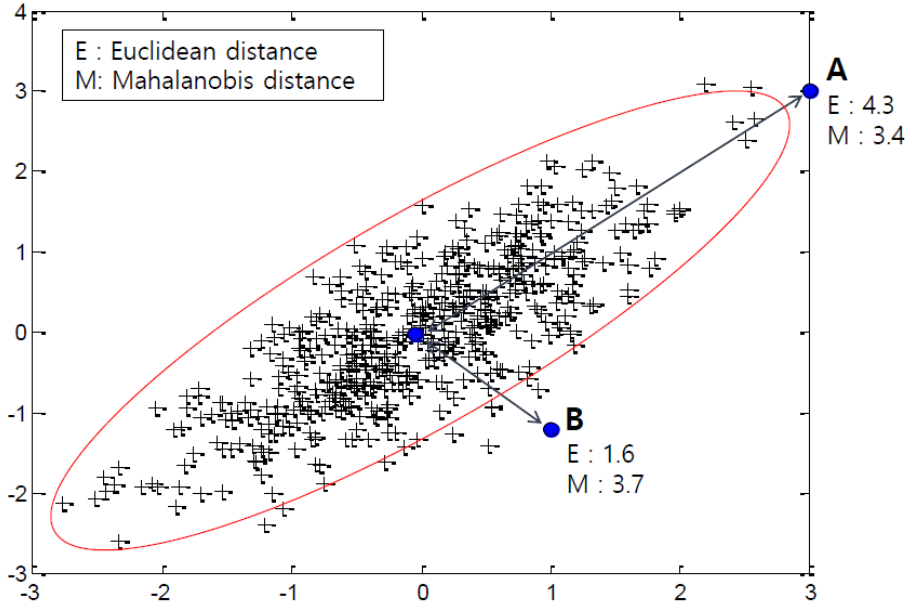
$$\|\mathbf{p}-\mathbf{q}\| = \sqrt{(\mathbf{p}-\mathbf{q})\cdot(\mathbf{p}-\mathbf{q})} = \sqrt{\|\mathbf{p}\|^2 + \|\mathbf{q}\|^2 - 2\mathbf{p}\cdot\mathbf{q}}.$$

- 맨하탄 거리

$$d_1(\mathbf{p},\mathbf{q}) = \|\mathbf{p}-\mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|$$

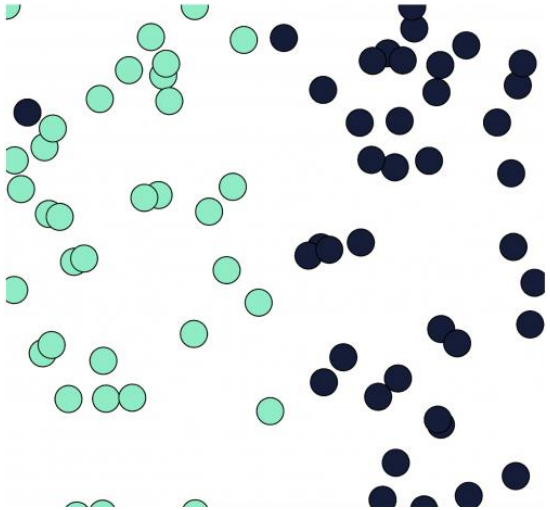
- 마할라노비스 거리

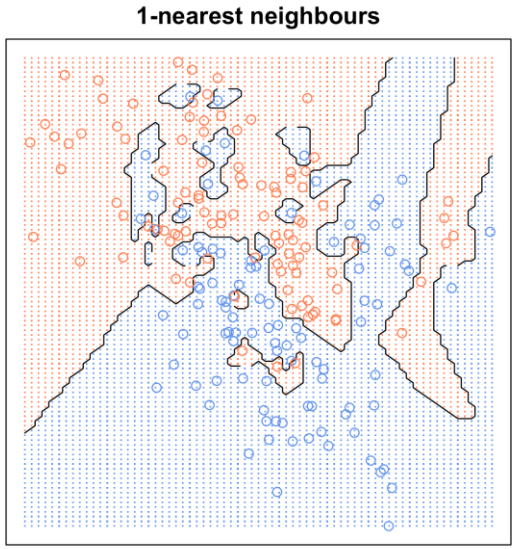
$$D_M(\vec{x}) = \sqrt{(\vec{x}-\vec{\mu})^T S^{-1} (\vec{x}-\vec{\mu})}.$$

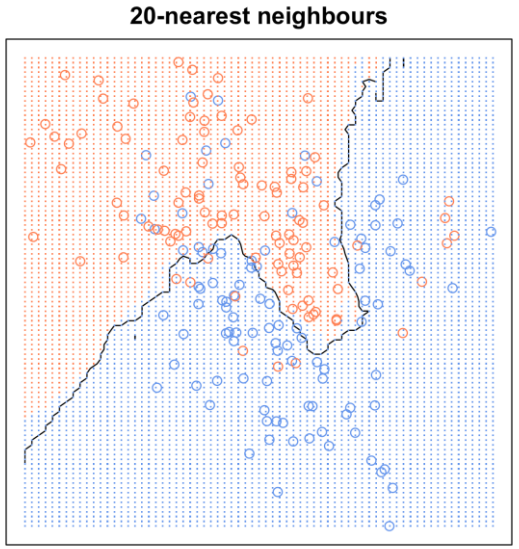


10.1.2 KNN의 하이퍼파라미터 Hyperparameter of KNN

- **탐색할 이웃 수, k**
  - k 가 너무 작을 경우 → **Overfitting**
  - k 가 너무 클 경우 → **Underfitting**

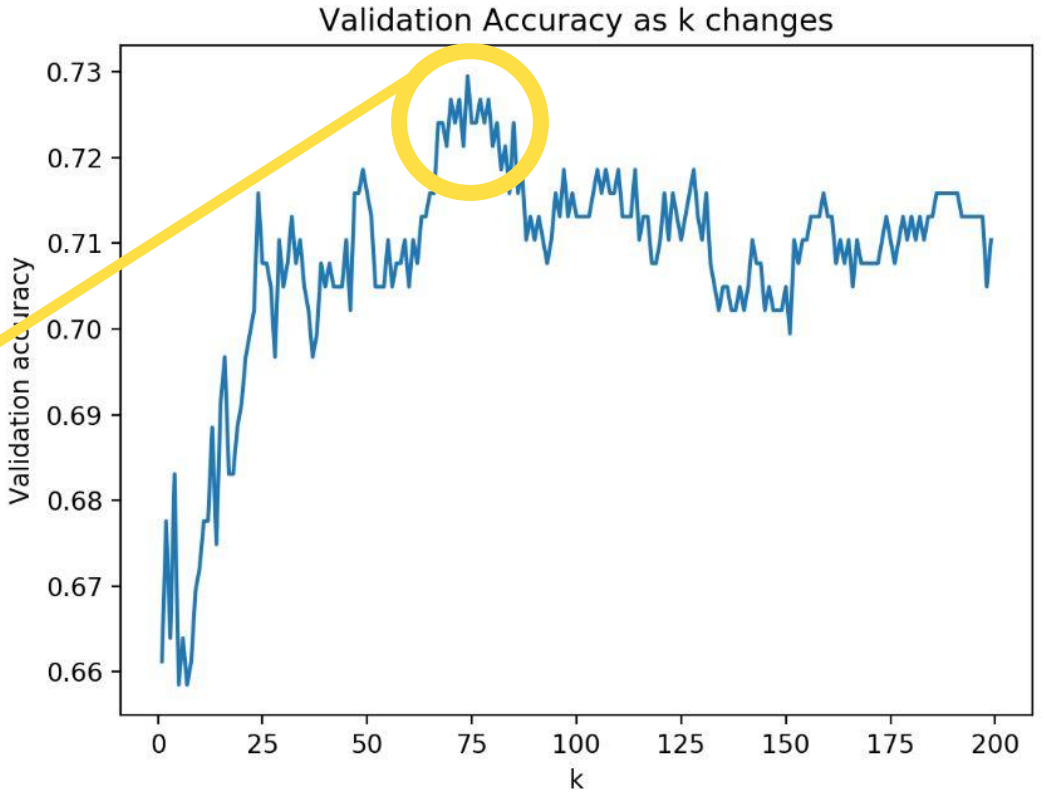






10.1.2 KNN의 하이퍼파라미터 Hyperparameter of KNN

- **탐색할 이웃 수, k**
  - k 가 너무 작을 경우 → **Overfitting**
  - k 가 너무 클 경우 → **Underfitting**
- **Best K? → Greedy Algorithm**



10.1.3 KNN 수행 시 고려해야 할 점

▪ Cut-off 기준 설정

학습데이터 범주의 **사전확률**을 고려해야 함 (특히, 범주 간 비율이 불균형한 데이터일 경우)

ex) 제조업 정상 / 불량 데이터 분류 문제 (범주 비율 정보 – 정상 : 0.8 / 불량 : 0.2)

▪ 변수 정규화

모든 특성들을 고르게 반영하기 위해 **정규화**가 필요

도시	인구(명)	미세먼지농도(㎍/㎥)
서울	1000만	200
시애틀	67만	40

10.1.4 KNN의 장단점

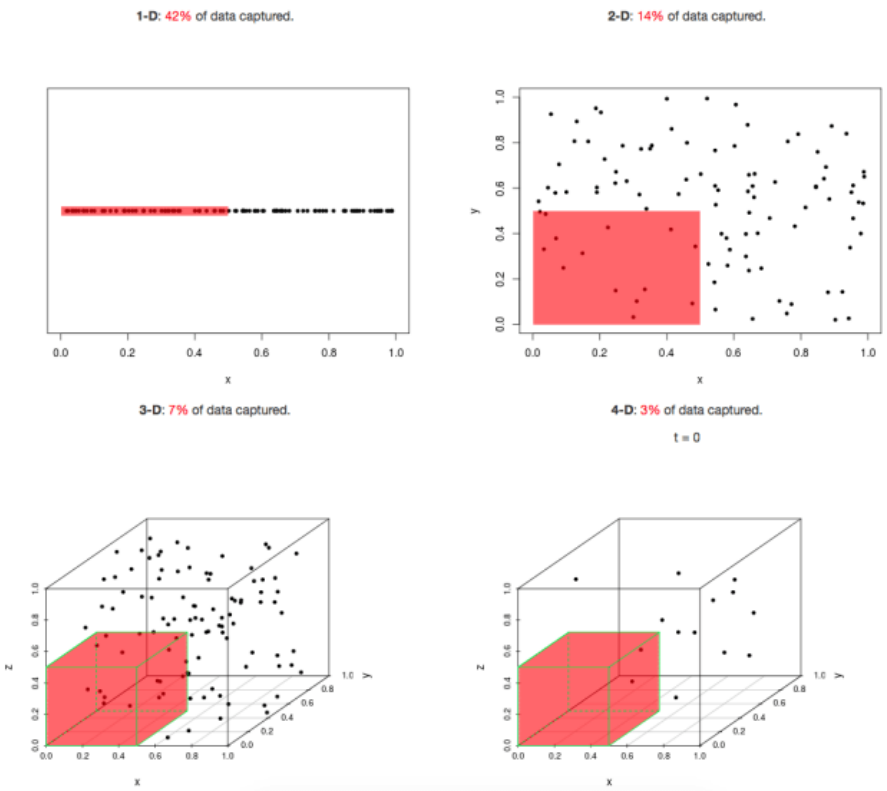
- **장점**
  - **Robust Model** : 학습데이터 내에 있는 노이즈, Outlier의 영향을 크게 받지 않음
  - 단순한 모델이지만, 적은 오차범위 : $Err(1-NN) \leq 2 * IdealErr$
- **단점**
  - 데이터 각각의 특성에 맞게 하이퍼파라미터 튜닝 해야 함
  - 높은 계산복잡도

10.2.0 차원의 저주 Curse of Dimensionality

▪ 차원의 저주 (Curse of Dimensionality)

When the dimensionality increases, the volume of the space increases so fast that the available data become sparse.

차원이 **커질수록**, 데이터 공간에서 데이터 샘플이 **희박해진다**

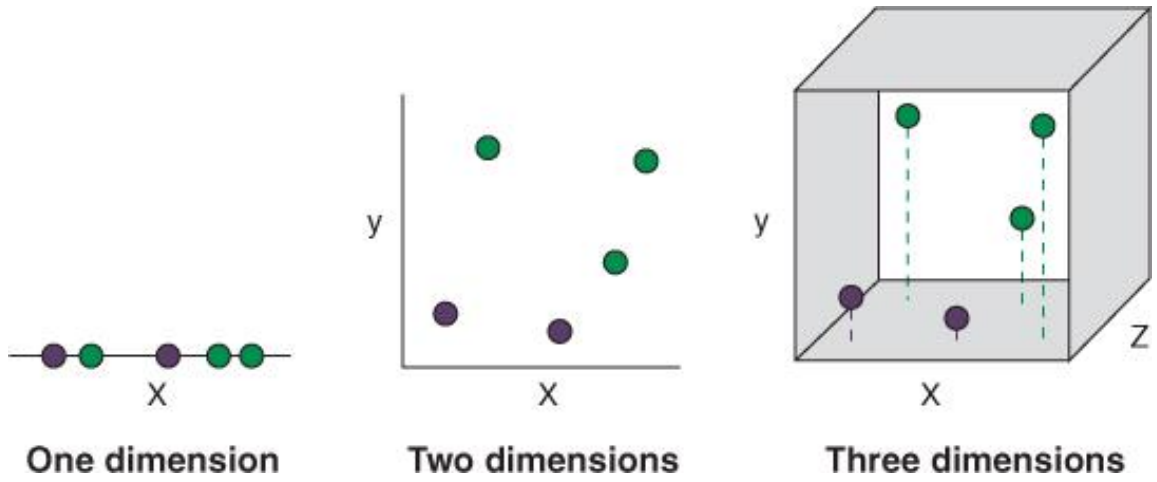


차원이 증가함에 따라 model complexity는 기하급수적으로 높아지고, 한정된 자료가 설명할 수 있는 공간은 줄어들게 됨.

10.2.0 차원의 저주 Curse of Dimensionality

▪ 고차원 데이터의 한계점

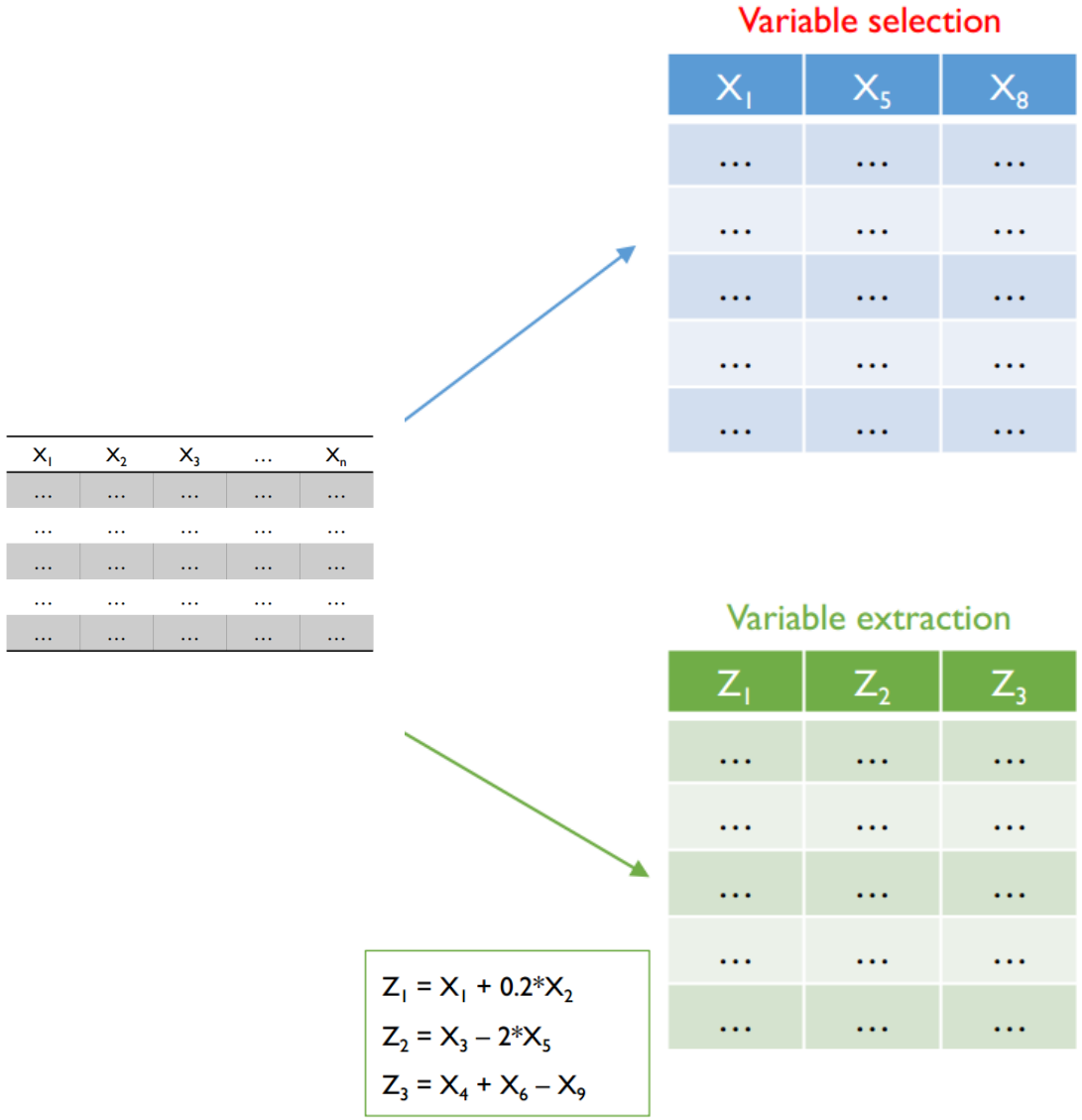
- 변수 多 : 불필요한 변수 존재
- 시각화 어려움
- 계산 복잡도 증가 : 모델링 비효율적



One dimension　Two dimensions　Three dimensions

이를 극복하는 방법 중 하나는, **차원을 축소**시키는 것!

10.2.1 차원 축소 Dimension Reduction



▪ 변수선택 (Feature Selection)

전체 특성 중 **유의미한 특성만을 선택**

⊕ 선택한 변수 해석 용이

⊖ 변수 간 상관관계 고려 어려움

▪ 변수추출 (Feature Extraction)

기존 특성을 조합하여 **새로운 변수 추출**

⊕ 변수 간 상관관계 고려 + 극적인 차원축소 가능

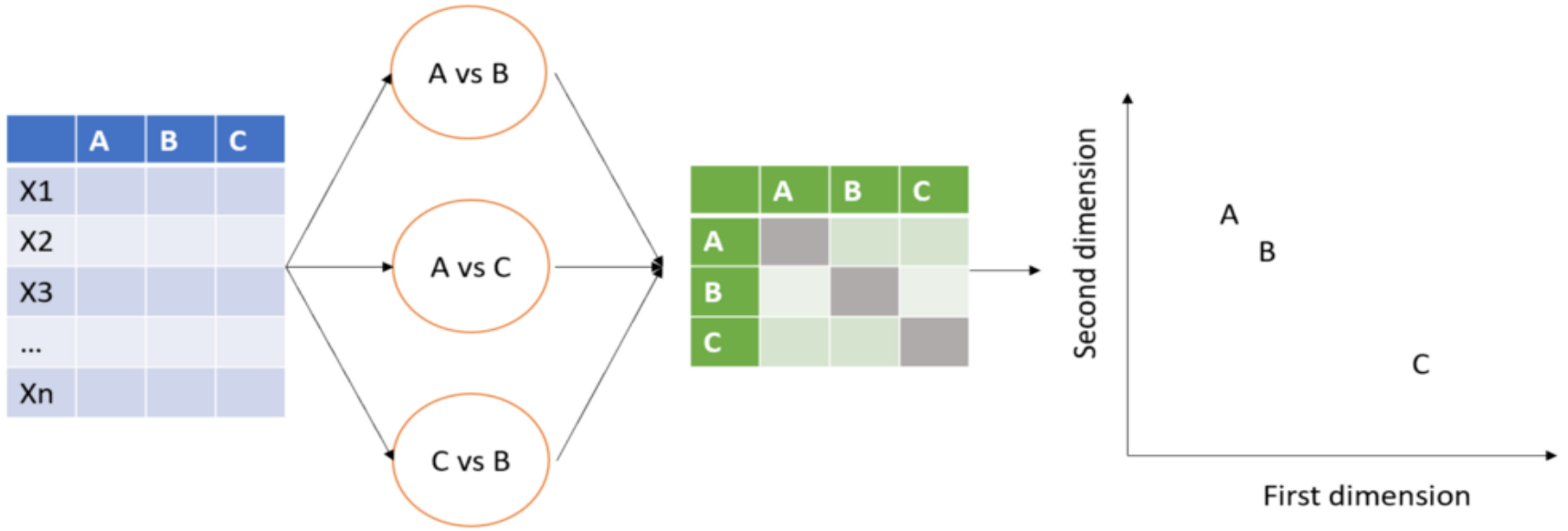
⊖ 추출된 변수의 해석이 어려움

10.2.2 다차원 스케일링 Multiple Dimensional Scaling, MDS

- **다차원 스케일링 (Multiple Dimensional Scaling)**

  d 차원 공간 상에 있는 객체 간 **거리**를 최대한 보존하는 저차원의 좌표계를 찾고자 함

  비음수성 / 동일성 / 대칭성 / 삼각부등식 성질 만족



10.2.2 다차원 스케일링 Multiple Dimensional Scaling, MDS

- **MDS 알고리즘**

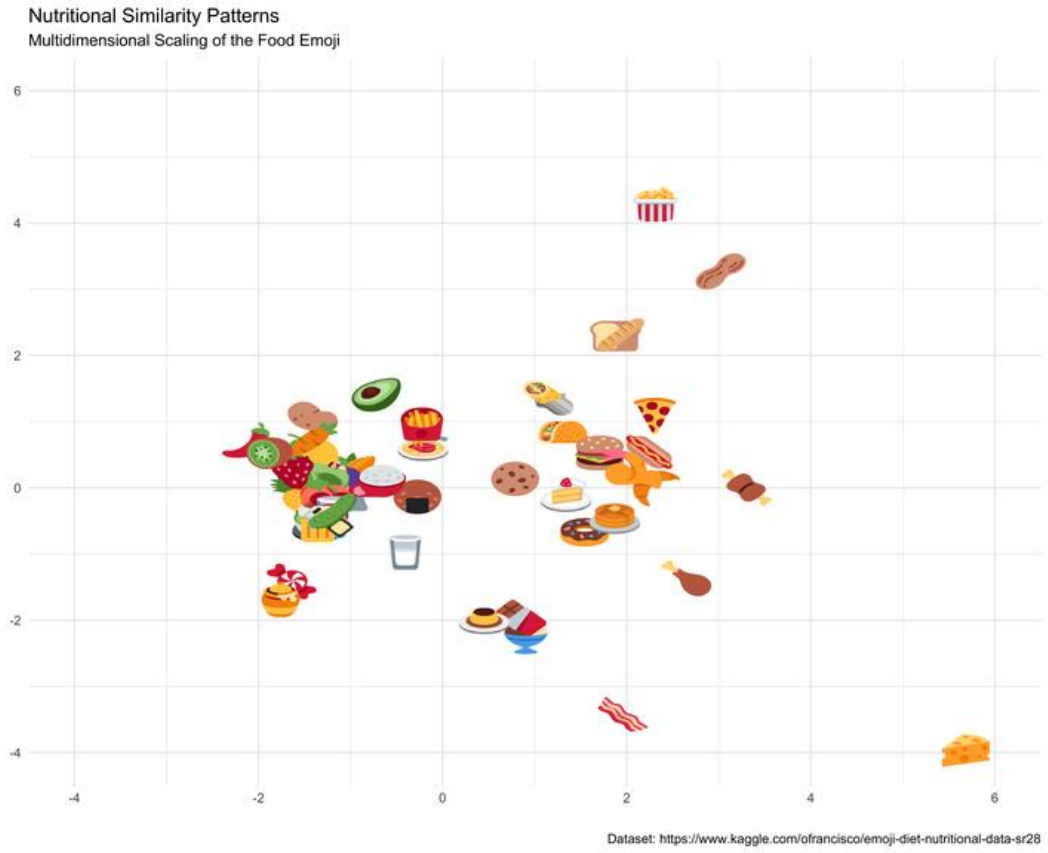
D *Distance matrix* ⇒ B *Inner product matrix* ⇒ Z *Coordinate matrix*

입력: 거리 행렬 $\mathbf{D} \in \mathbb{R}^{m \times m}$, 원소 $dist_{ij}$는 샘플 $\boldsymbol{x}_i$에서 $\boldsymbol{x}_j$까지의 거리
저차원 공간 차원수 d'

과정:

1: 식 10.7~10.9에 따라 $dist_{i\cdot}^2$, $dist_{\cdot j}^2$, $dist_{\cdot\cdot}^2$을 계산
2: 식 10.10에 따라 행렬 $\mathbf{B}$를 계산
3: 행렬 $\mathbf{B}$에 대해 고윳값 분해 실행
4: $\tilde{\mathbf{\Lambda}}$ d'개 최대 고윳값으로 구성된 대각 행렬로,
$\tilde{\mathbf{V}}$에 상응하는 고유 벡터 행렬로 하여 값을 구한다

출력: 행렬 $\tilde{\mathbf{\Lambda}}\tilde{\mathbf{V}}^{1/2} \in \mathbb{R}^{m \times d'}$, 각 행은 한 샘플의 저차원 좌표



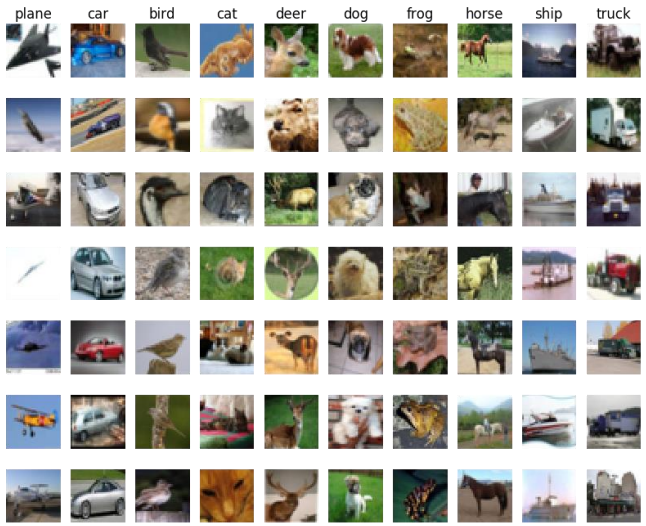
1. 객체 간 근접도 (유사도) 행렬 생성
2. 거리 행렬 D를 통해 내적 행렬 B 도출
3. B에 대해 고윳값 분해
4. 거리 정보를 최대한 보존하는 좌표 시스템 (Z) 찾기

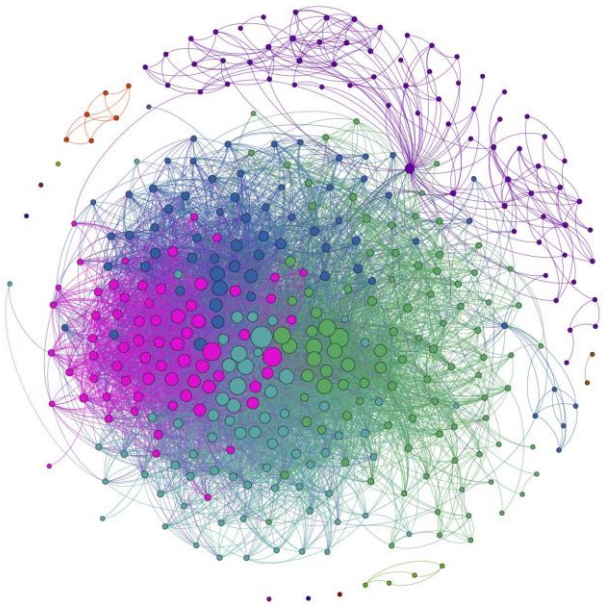
Dimensionality Reduction – MDS 참고

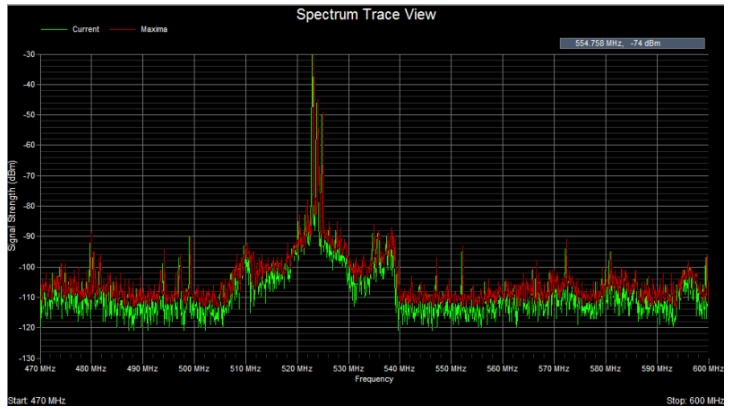
10.3.0 주성분분석개요 Principal Component Analysis, PCA

▪ 주성분 분석 (Principal Component Analysis, PCA)

고차원 데이터를 효과적으로 분석하기 위한 대표적인 분석 기법



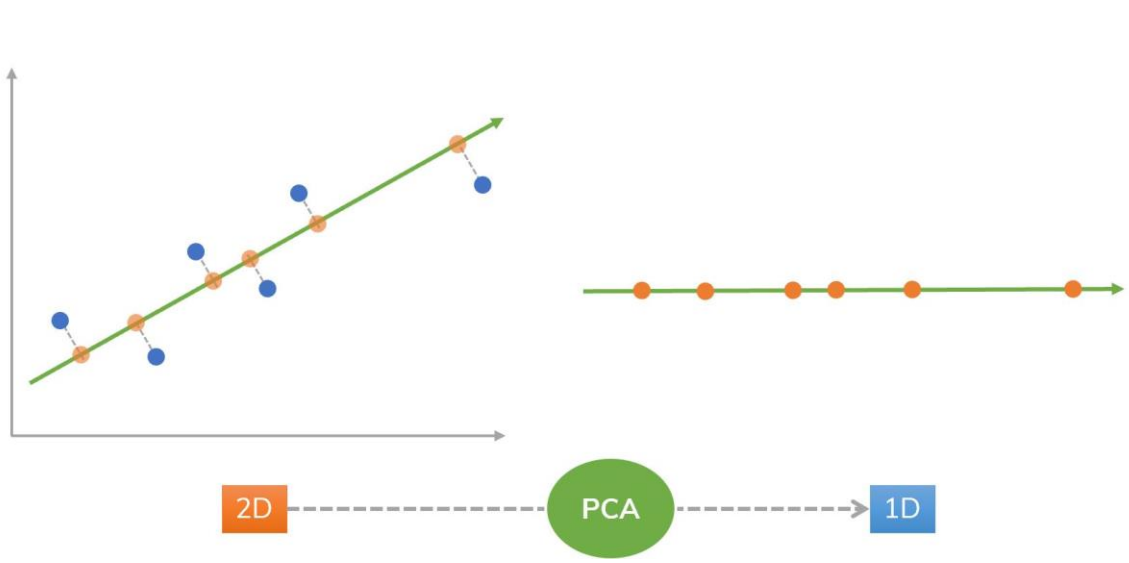


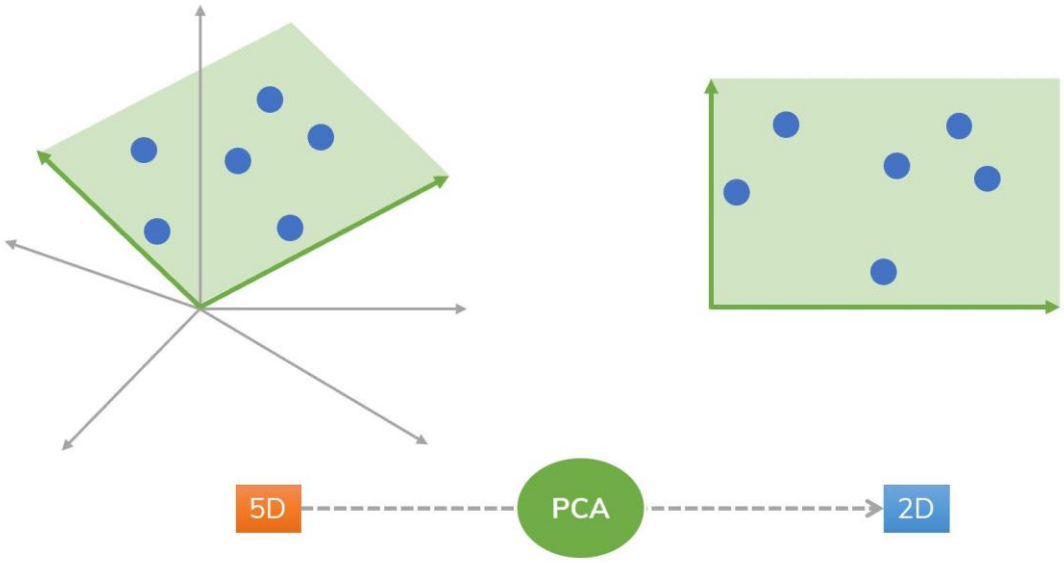


10.3.0 주성분분석개요 Principal Component Analysis, PCA

▪ 주성분 분석 (Principal Component Analysis, PCA)

차원을 줄이면서, 정보의 손실을 최소화하는 새로운 축을 찾아내는 기법

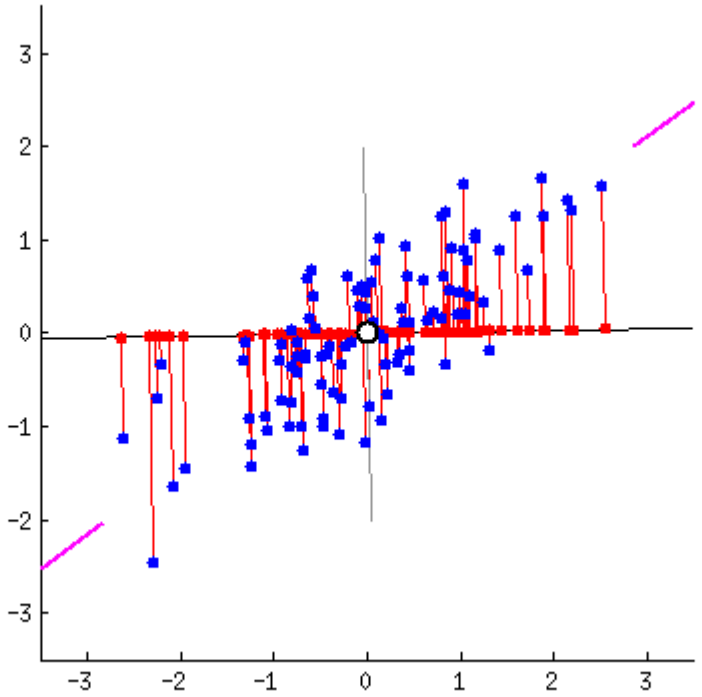


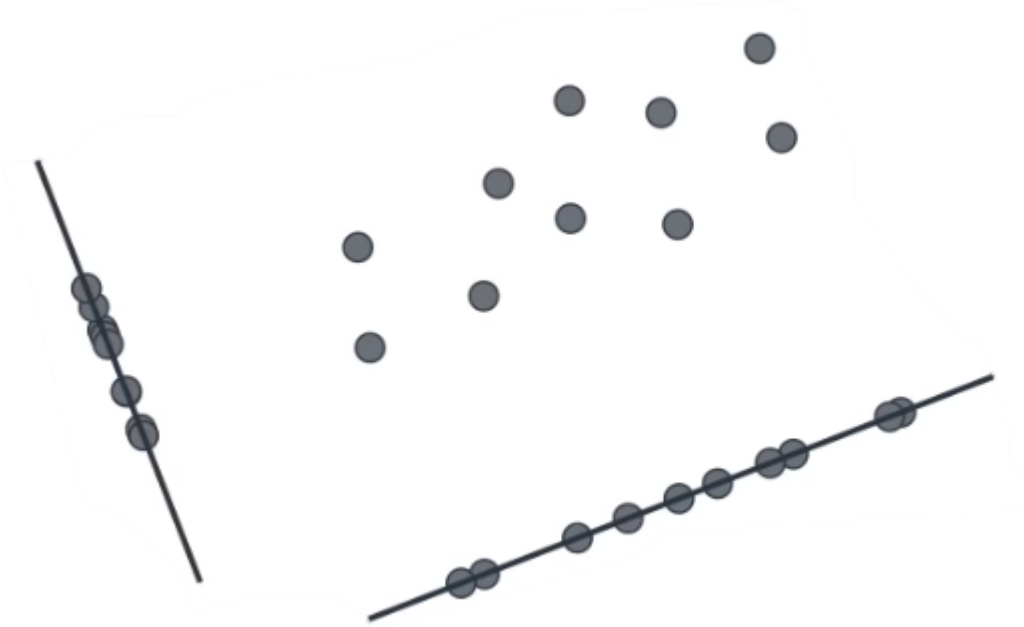


10.3.0 주성분분석개요 Principal Component Analysis, PCA

▪ 주성분 분석 (Principal Component Analysis, PCA)

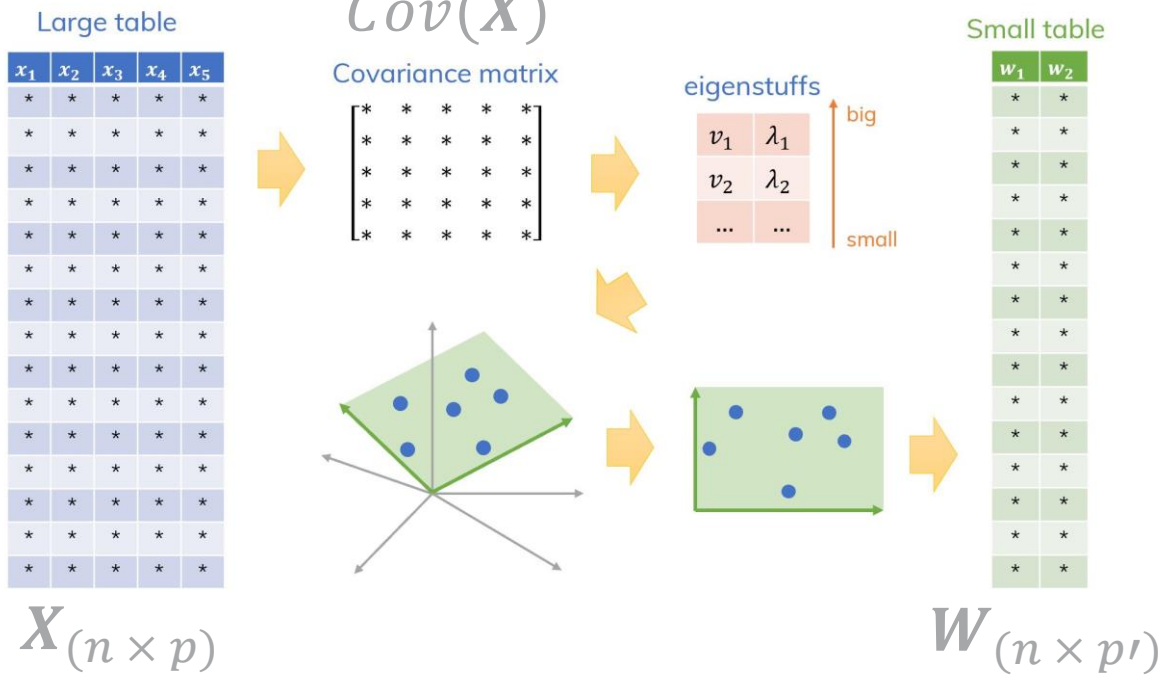
차원을 줄이면서, **정보의 손실을 최소화**하는 새로운 축을 찾아내는 기법





10.3.1 PCA 알고리즘

▪ PCA 알고리즘



1. Mean Centering

X_i의 평균이 0이라면

2. 샘플의 공분산 행렬 $\boldsymbol{XX^T}$ 구하기

3. 공분산 행렬 $\boldsymbol{XX^T}$에 대하여 고윳값 분해

4. eigenvalue와 eigenvector 구하여 정렬

$$\lambda(1) > \lambda(2) > \lambda(3) > \lambda(4) > \lambda(5)$$

$e(1) > e(2) > e(3) > e(4) > e(5)$ is a vector

5. 정렬된 eigenvector를 토대로 기존 변수 선형변환

$$\boldsymbol{W_1} = e(1)\boldsymbol{X} = e_{11} \cdot \boldsymbol{X_1} + e_{12} \cdot \boldsymbol{X_2} + \ldots + e_{15} \cdot \boldsymbol{X_5}$$

...

$$\boldsymbol{W_5} = e(5)\boldsymbol{X} = e_{51} \cdot \boldsymbol{X_1} + e_{52} \cdot \boldsymbol{X_2} + \ldots + e_{55} \cdot \boldsymbol{X_5}$$

10.3.2 PCA 예제

- Step 1. Mean Centering

$\mathbf{X}$ =

X_1	X_2	X_3
0.2	5.6	3.56
0.45	5.89	2.4
0.33	6.37	1.95
0.54	7.9	1.32
0.77	7.87	0.98

$\mathbf{X}$ =

X_1	X_2	X_3
-1.1930	-1.0300	1.5012
-0.0370	-0.7647	0.3540
-0.5919	-0.3257	-0.0910
0.3792	1.0739	-0.7140
1.4427	1.0464	-1.0502

(normalize X to $E(X_i)=0$, $Var(X_i)=1$)

- Step 2. 샘플의 공분산 행렬 구하기

$$\Sigma = \frac{1}{n} X^T X$$

Covariance($\mathbf{X}$) =

0.0468	0.1990	-0.1993
0.1990	1.1951	-1.0096
-0.1993	-1.0096	1.0225

10.3.2 PCA 예제

- **Step 3. 공분산 행렬 고윳값 분해 (Singular Value Decomposition, SVD)**

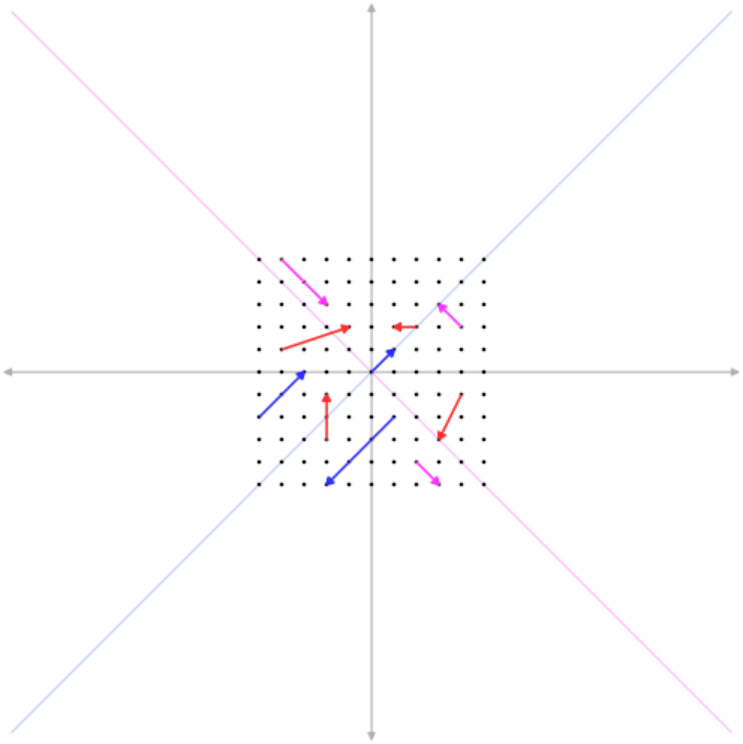
The eigenvalue-eigenvector pairs on the correlation matrix, Σ

$$[E \Lambda V] = \text{svd}(\Sigma)$$

$$\lambda_1 = 0.0786, \quad e_1^T = [0.2590 \quad 0.5502 \quad 0.7938]$$

$$\lambda_2 = 0.1618, \quad e_2^T = [0.7798 \quad -0.6041 \quad 0.1643]$$

$$\lambda_3 = 2.7596, \quad e_3^T = [0.5699 \quad 0.5765 \quad -0.5855]$$



10.3.2 PCA 예제

- **Step 4. eigenvalue와 eigenvector 구하여 정렬**

$$\lambda_1 = 0.0786, e_1^T = [0.2590 \quad 0.5502 \quad 0.7938]$$
$$\lambda_2 = 0.1618, e_2^T = [0.7798 \quad -0.6041 \quad 0.1643]$$
$$\lambda_3 = 2.7596, e_3^T = [0.5699 \quad 0.5765 \quad -0.5855]$$

$, \mathbf{X} =$

X_1	X_2	X_3
-1.1930	-1.0300	1.5012
-0.0370	-0.7647	0.3540
-0.5919	-0.3257	-0.0910
0.3792	1.0739	-0.7140
1.4427	1.0464	-1.0502

(normalize X to $E(X_i)=0$, $Var(X_i)=1$)

$$\lambda_3 > \lambda_2 > \lambda_1$$

- **Step 5. 기존 변수 선형 변환**

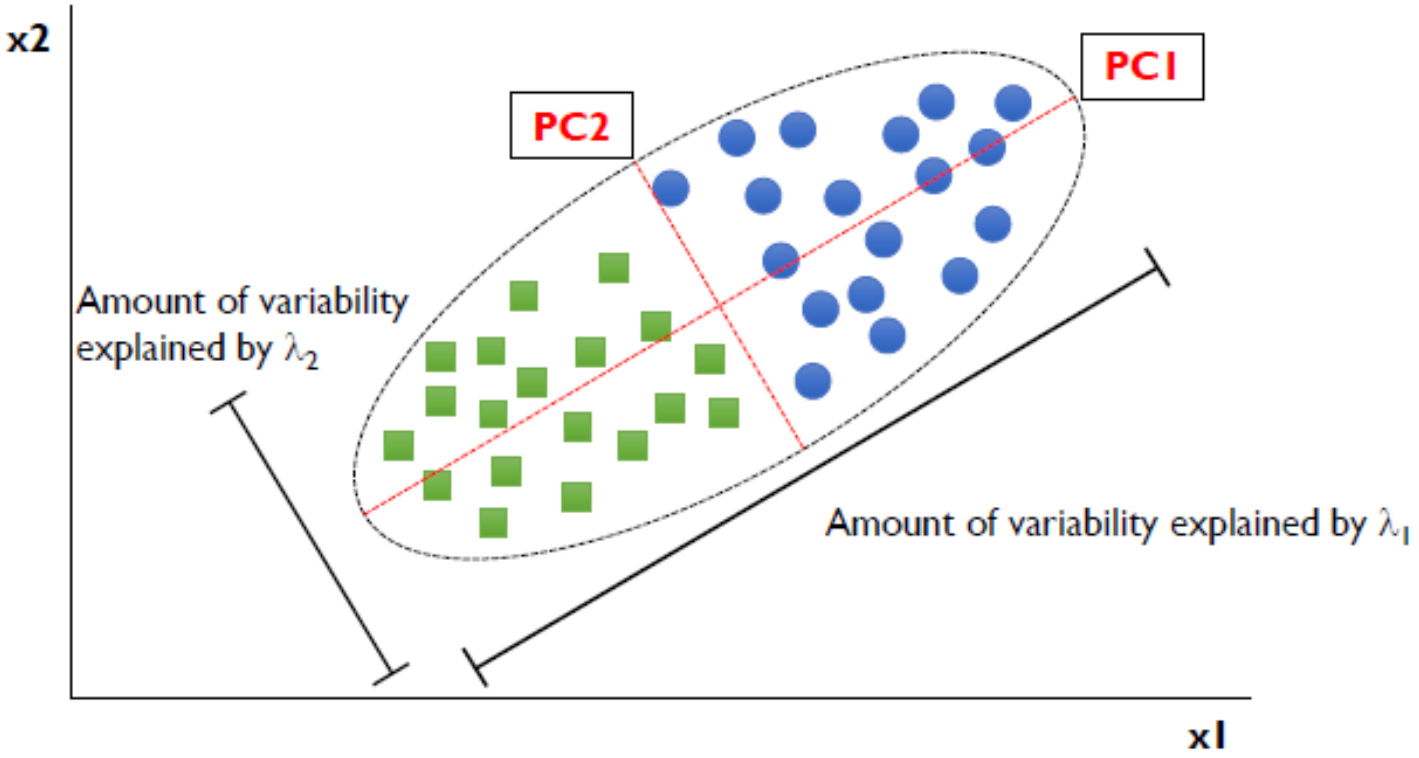
$$Z_1 = e_1^T X = 0.5699 \cdot X_1 + 0.5765 \cdot X_2 - 0.5855 \cdot X_3 = 0.5699 \cdot \begin{bmatrix} -1.1930 \\ -0.0370 \\ -0.5919 \\ 0.3792 \\ 1.4427 \end{bmatrix} + 0.5765 \cdot \begin{bmatrix} -1.0300 \\ -0.7647 \\ -0.3257 \\ 1.0739 \\ 1.0464 \end{bmatrix} - 0.5855 \cdot \begin{bmatrix} 1.5012 \\ 0.3540 \\ -0.0910 \\ -0.7140 \\ -1.0502 \end{bmatrix} = \begin{bmatrix} -2.1527 \\ -0.6692 \\ -0.4718 \\ 1.2533 \\ 2.0404 \end{bmatrix}$$

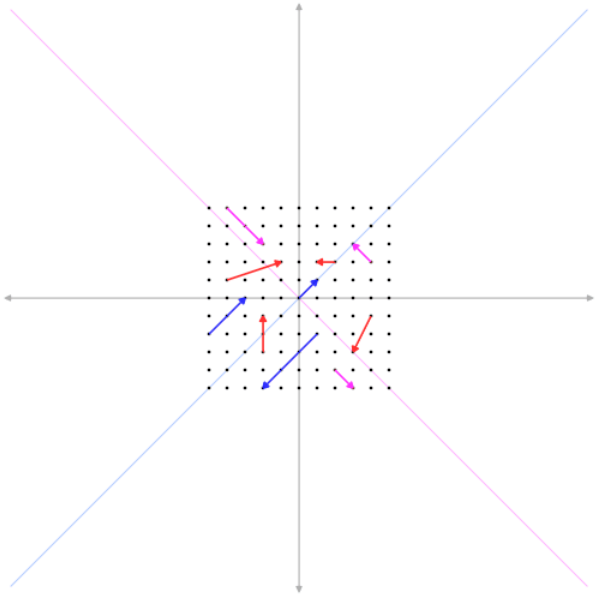
$$Z_2 = e_2^T X = \begin{bmatrix} -0.0615 \\ 0.4912 \\ -0.2798 \\ -0.4703 \\ 0.3204 \end{bmatrix} \qquad Z_3 = e_3^T X = \begin{bmatrix} 0.3160 \\ -0.1493 \\ -0.4047 \\ 0.1223 \\ 0.1157 \end{bmatrix}$$

$$\therefore Z = \begin{bmatrix} -2.1527 & -0.0615 & 0.3160 \\ -0.6692 & 0.4912 & -0.1493 \\ -0.4718 & -0.2798 & -0.4047 \\ 1.2533 & -0.4703 & 0.1223 \\ 2.0404 & 0.3204 & 0.1157 \end{bmatrix}$$

10.3.3 PCA의 이해

- Eigenvalues와 PC Scores





$\lambda(i), \quad i = 1, \ldots, p$

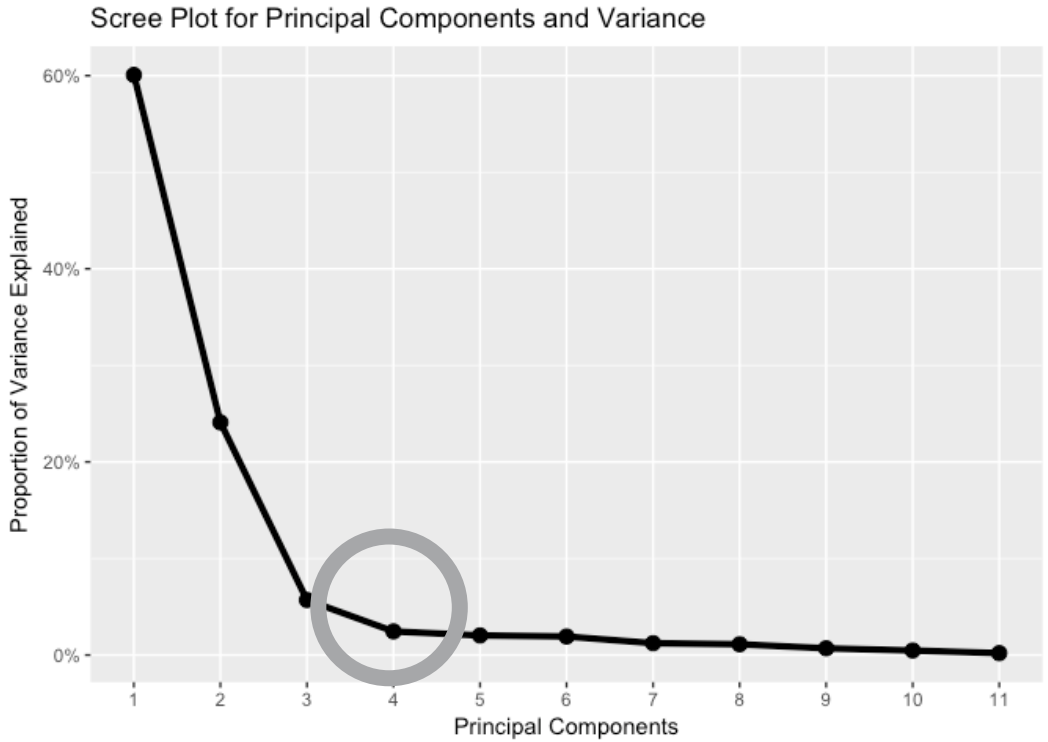
Eigenvalues of
the covariance matrix
=
Variances of
each principal component

10.3.4 몇 개의 주성분?

- **Scree Plot**

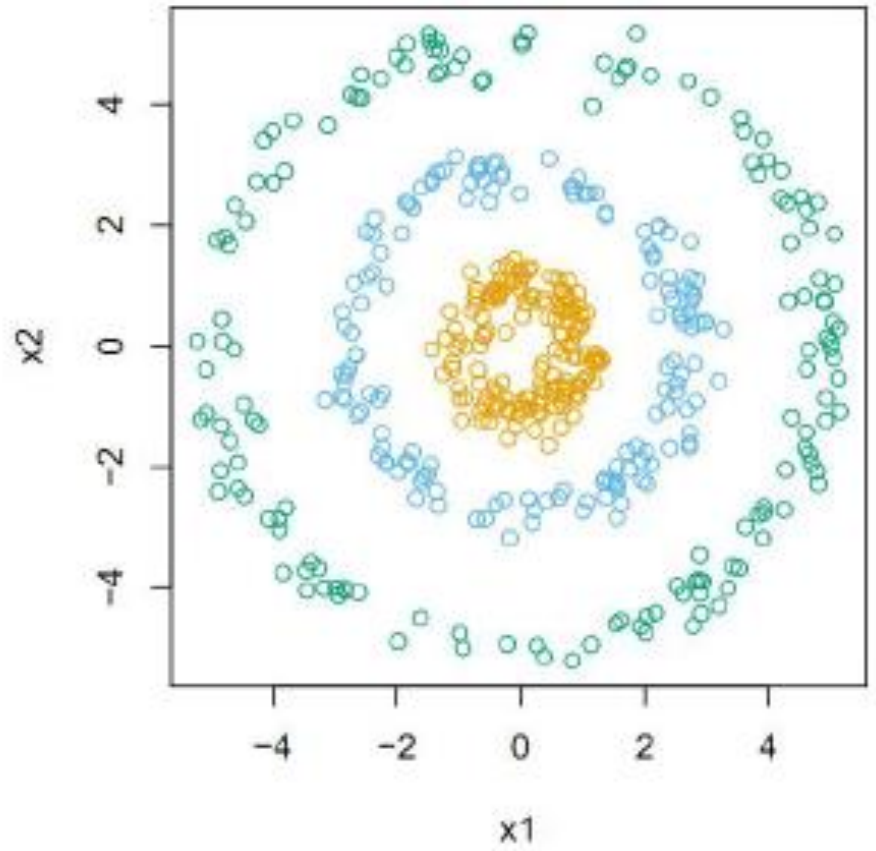
  주성분의 개수와, 총 분산 중 설명하는 비율을 보여주는 Plot

  - Elbow Point
  - Cut-off (보통 70% 이상)



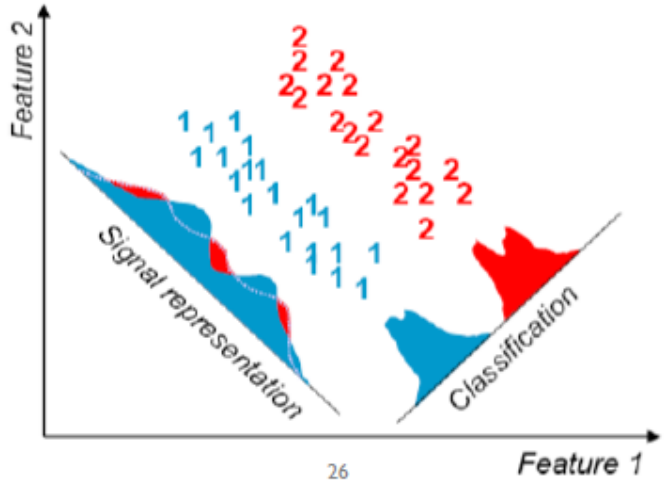
10.3.5 PCA의 한계점

- 데이터의 분포가 비선형적일 경우 적용하기 어려움



10.3.5 PCA의 한계점

- 분류 / 예측 자체에 주요한 변수 추출은 X



- 결과 해석이 어려움

Variable	PCA factor 1	2	3	4
Distance to main city	*0.83*	0.10	−0.13	0.01
Total area of the farm (ha)	0.10	0.01	*0.81*	0.13
Number of land division in the farms/total area	*−0.76*	−0.21	−0.39	−0.05
Hectares of pastures/total area	−0.52	0.42	−0.36	0.38
Hectares of grassland/total area	0.30	*−0.94*	0.11	0.05
Hectares of exotic woodland/total area	−0.11	−0.01	−0.04	−0.68
Hectares of native woodland/total area	0.29	*0.82*	0.21	0.04
Number of cattle on each farm	0.01	0.01	*−0.78*	0.23
Intensity of hunting	0.07	0.19	0.09	*0.75*
Number of dogs living on the farm	*−0.84*	0.28	0.07	0.01
Eigenvalue	2.4	1.9	1.6	1.3
Percentage of variance explained	23.9	19.3	16.3	12.6

Etc. 참고자료

- **고려대학교 김성범,강필성 교수님 강의**
  - Dimensionality Reduction – MDS
  - Principal Component Analysis (PCA, 주성분 분석)

- **참고 블로그**
  - ratsgo's blog
  - Data Science by Yngie