선형대수학팀

3팀
윤지영
이지윤
이수린
임지훈
채희지

INDEX

1. 고유값과 고유벡터

2. 고유값 분해

3. 주성분분석

4. 특이값 분해

5. 특이값 분해 활용

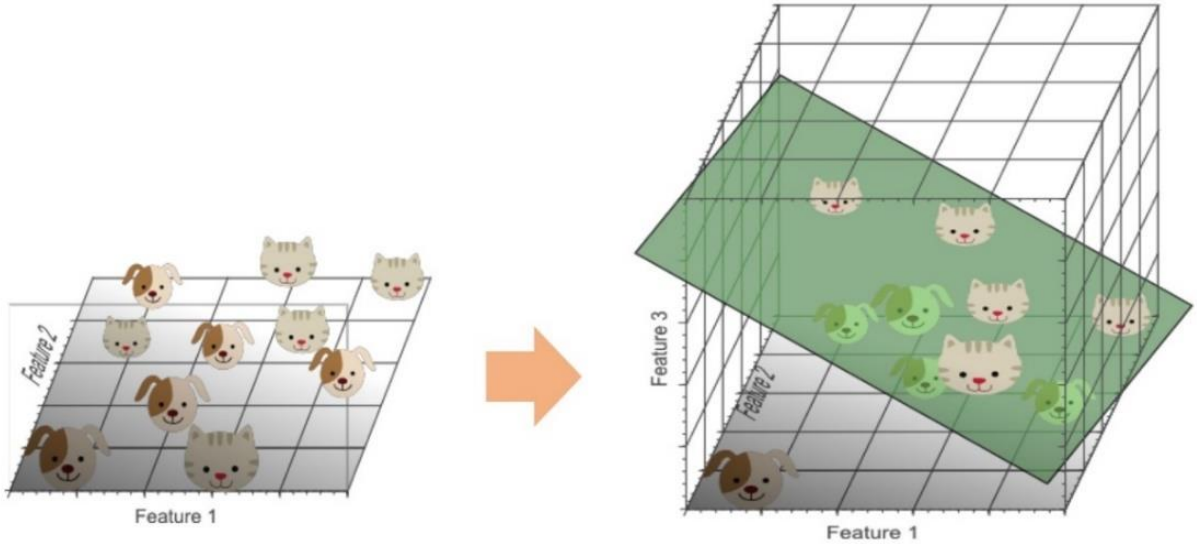
1

고유값과 고유벡터

차원의 저주

고차원 데이터는 많은 정보를 가지고 있다

차원의 증가가 고차원 데이터 분석에 긍정적으로 작용하는 경우



차원의 저주

고차원 데이터는 많은 정보를 가지고 있다

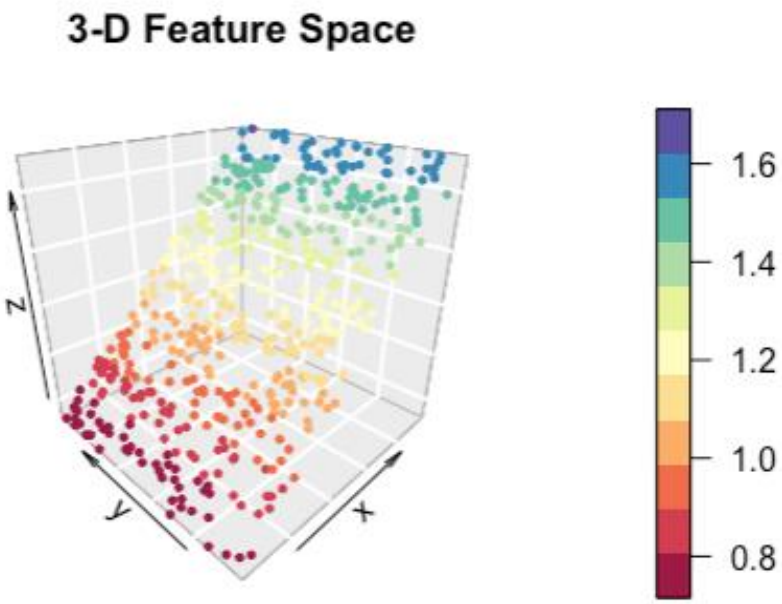
만약 차원의 증가가 고차원 데이터 분석에 긍정적으로 작용한다면?

그럼 변수의 수가 많을수록 무조건

모델이 좋아지는 걸까?

항상 그렇지는 않음

차원의 저주



데이터는 2차원 모델링은 3차원

더 적은 차원으로 표현 가능한 데이터를 **고차원의 공간**에 표현

점들 간의 거리가 더 늘어나고, 계산량 또한 증가

차원의 저주

차원 수 > 학습 데이터 수 → 과적합이 일어나기 쉬움

변수 간의 관련성이 높은 경우, **차원의 증가**가 공간의 낭비로 이어질 수 있음.

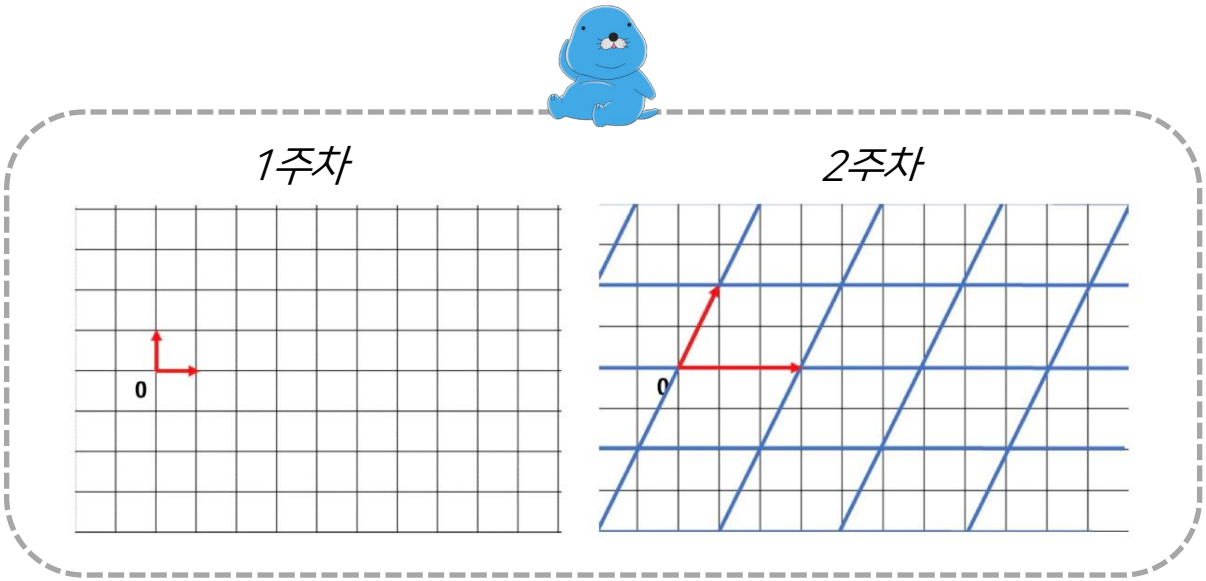
비효율적인 분석

⋮

차원의 저주

어떻게 다뤄야 할까…?

고유값과 고유벡터의 의미

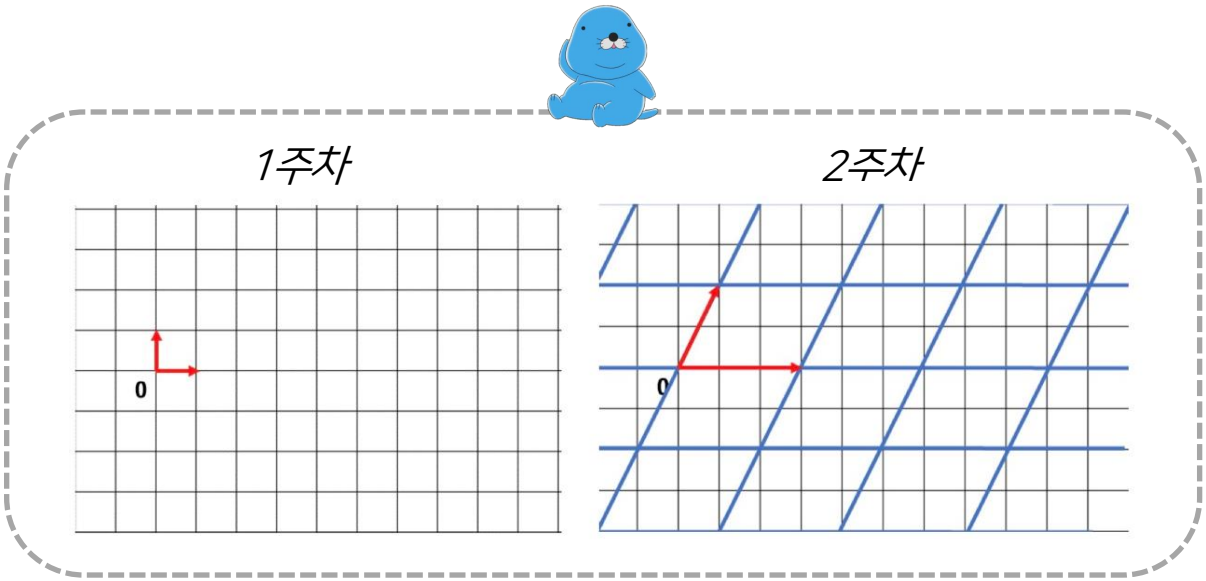


빨간색 기저 벡터 $\begin{matrix}1\\0\end{matrix}, \begin{matrix}0\\1\end{matrix}$ 를 변환시키는 방식으로 선형변환을 이해 …1주차

⋮

$\begin{matrix}1\\0\end{matrix}, \begin{matrix}0\\1\end{matrix}$ 가 R^2 공간의 유일한 기저 벡터가 아니라는 것을 배움 …2주차

고유값과 고유벡터의 의미

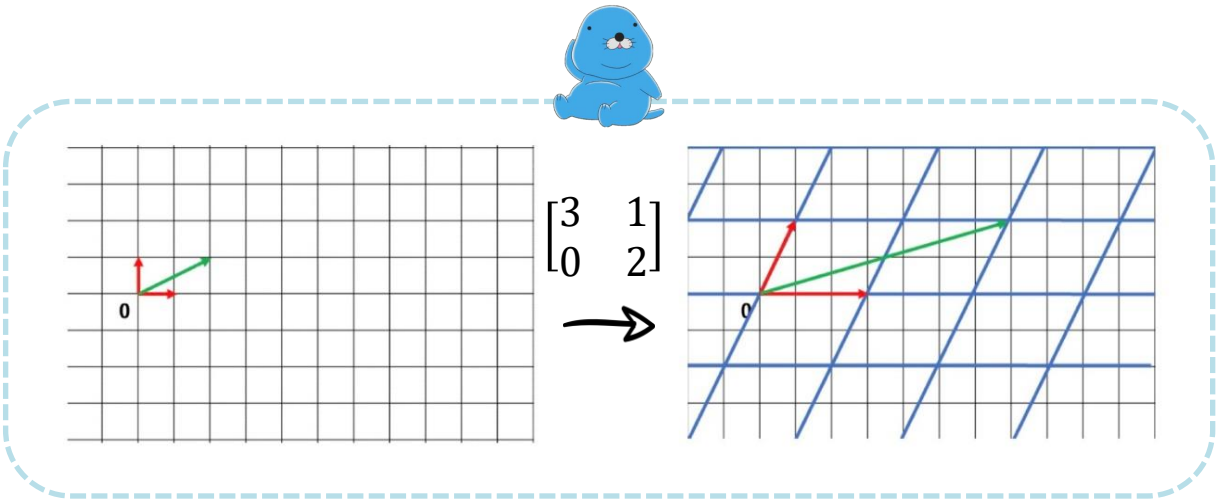


빨간색 기저 벡터 $\begin{matrix}1\\0\end{matrix}, \begin{matrix}0\\1\end{matrix}$ 를 변환시키는 방식으로 선형변환을 이해 …1주차

이 선형변환의 기준이 될 만한 축은 무엇일까?

$\begin{matrix}1\\0\end{matrix}, \begin{matrix}0\\1\end{matrix}$ 가 R^2 공간의 유일한 기저 벡터가 아니라는 것을 배움 …2주차

고유값과 고유벡터의 의미



기존 벡터는 $\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 7 \\ 2 \end{bmatrix}$로 매핑됨

⋮

기존의 초록색 벡터 $\begin{matrix} 2 \\ 1 \end{matrix}$ 과 변환 후의 초록색 벡터 $\begin{matrix} 7 \\ 2 \end{matrix}$ 는 다른 직선 위에 있음

$\begin{matrix} 2 \\ 1 \end{matrix}$ 벡터가 $span$하는 직선 공간에 $\begin{matrix} 7 \\ 2 \end{matrix}$ 가 존재하지 않음

고유값과 고유벡터의 의미

그렇다면 어떤 벡터가 **변환 후에도**

자신만의 고유한 $span$ 공간에 남아있을까?

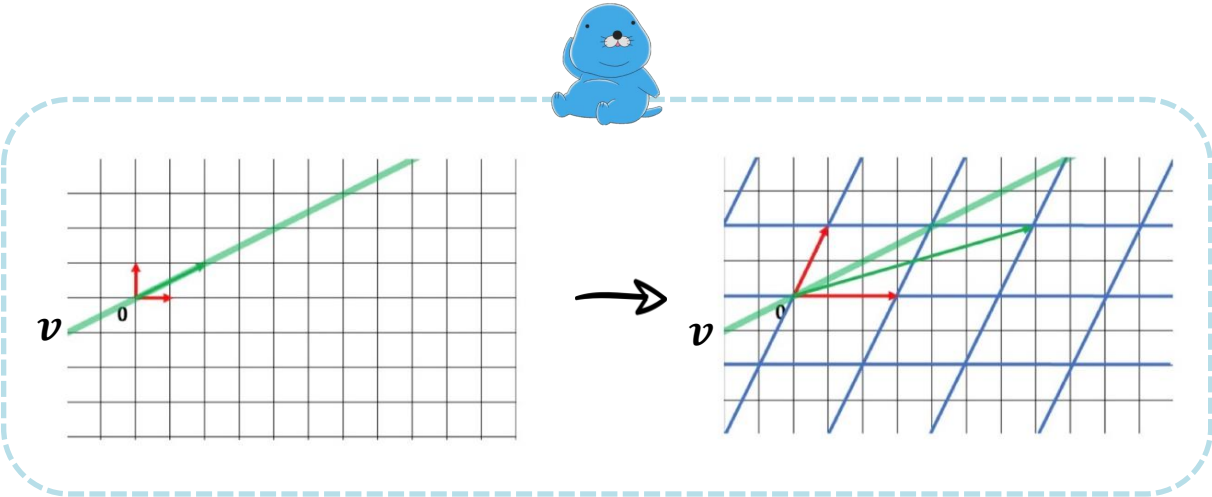
기존 벡터는 $\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 7 \\ 2 \end{bmatrix}$로 매핑됨

⋮

기존의 초록색 벡터 $\begin{matrix} 2 \\ 1 \end{matrix}$ 과 변환 후의 초록색 벡터 $\begin{matrix} 7 \\ 2 \end{matrix}$ 는 다른 직선 위에 있음

$\begin{matrix} 2 \\ 1 \end{matrix}$ 벡터가 $span$하는 직선 공간에 $\begin{matrix} 7 \\ 2 \end{matrix}$ 가 존재하지 않음

고유값과 고유벡터의 의미



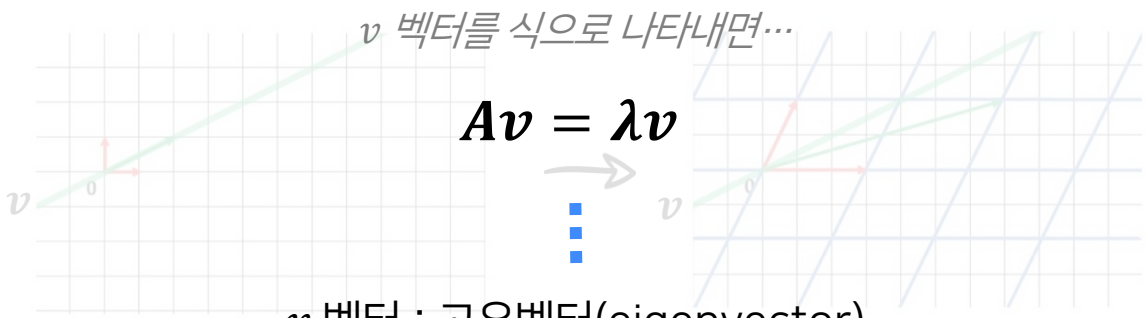
초록색 사선(v)은 변환 후에도 고유한 $span$ 공간에 남아 있음

⋮

행렬변환을 단순히 **벡터의 길이를 변화시키는 스칼라 값**으로 여기게 해줌

1 고유값과 고유벡터(Eigenvalue & Eigenvector)

고유값과 고유벡터의 의미



v 벡터를 식으로 나타내면…

$$\boldsymbol{Av = \lambda v}$$

v 벡터 : 고유벡터(eigenvector)

λ : **고유값**(eigenvalue)

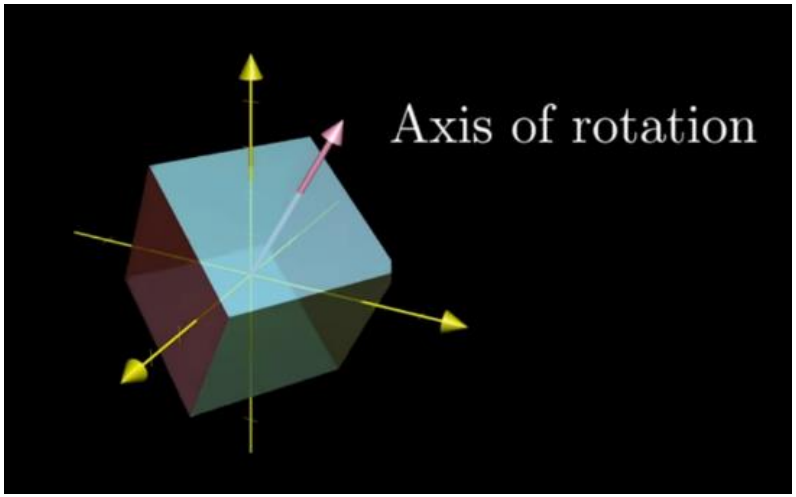
초록색 사선(v)은 변환 후에도 고유한 $span$ 공간에 남아 있음

v 벡터가 얼마나 늘어나는지 알려주는 값

행렬변환을 단순히 벡터의 길이를 변화시키는 스칼라 값으로 여기게 해줌

고유값, 고유벡터를 찾는 이유

3*3 행렬 A: 좌표축을 회전시키는 변환



핑크색 고유벡터는 그대로 유지될 것

⋮

변환의 회전축이라 생각할 수 있음

1 고유값과 고유벡터(Eigenvalue & Eigenvector)

고유값과 고유벡터의 의미

고유값과 고유벡터란?

고유값

고유벡터 방향으로 벡터공간이
얼마만큼의 크기로 늘려지는지를 의미

고유벡터

행렬이 벡터에 작용하는
주축(Principal Axis)의 방향을 의미

✓ 선형변환을 취해주었을 때, **크기만 바뀌고 방향은 바뀌지 않는** 경우

1 고유값과 고유벡터(Eigenvalue & Eigenvector)

고유값과 고유벡터의 의미

고유값과 고유벡터는 결국 회전변환?

단순히 정방향으로만 회전하는 것은 아님

- ✓ 변환 후의 기저 벡터 길이가 모두 같지 X
- ✓ 변환 시 뒤집어지면서 변환할 가능성 있음

완전히 일치하는 것은 아님

✓ 회전변환과 유사한 형태를 보인다!

선형변환을 취해주었을 때, **크기만 바뀌고 방향은 바뀌지 않는** 경우

고유값과 고유벡터의 의미

각 행렬에 대해 고유벡터가 존재하는지 어떻게 알 수 있을까?

$$(A - \lambda I)x = 0$$

$(A - \lambda I)$ 행렬의 역행렬이 존재하면 $x = 0$ (trivial solution)이 나옴

영벡터에 선형변환을 적용해도 영벡터니까 고유벡터?

고유값의 정의에 따라 영벡터는 제외됨

고유값과 고유벡터의 의미

각 행렬에 대해 고유벡터가 존재하는지 어떻게 알 수 있을까?

$$(A - \lambda I)x = 0$$

$(A - \lambda I)$ 행렬의 역행렬이 존재하면 $x = 0$ (trivial solution)이 나옴

영벡터에 선형변환을 적용해도 영벡터니까 고유벡터?

고유값의 정의에 따라 영벡터는 제외됨

고유값과 고유벡터의 의미

$(A - \lambda I)x$의 **역행렬이 존재 X**

⋮

Non-trivial한 해를 구할 수 있음

이때 구해지는 x가 고유벡터, λ가 고유값

고유값과 고유벡터의 의미

$(A - \lambda I)x$의 **역행렬이 존재 X**

Non-trivial한 해를 구할 수 있음

이때 구해지는 x가 고유벡터, λ가 고유값

1주차 클린업 내용!

공간을 압축시키는 선형변환의 경우

행렬식이 0

이 특징을 사용!

고유값과 고유벡터의 의미

$(A - \lambda I)x$의 **역행렬이 존재 X**

Non-trivial한 해를 구할 수 있음

이때 구해지는 x가 고유벡터, λ가 고유값

앞의 개념을 사용!

$$\det(A - \lambda I) = 0$$

$n * n$ 행렬의 고유값 n개 구할 수 있게 됨

$Ax = \lambda x$ 식에 고유값들을 대입해 **고유벡터들을 찾을 수 있음**

고유값과 고유벡터의 의미

2*2 행렬 A에 대한 고유값과 고유벡터

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\det(A - \lambda I) = \begin{vmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{vmatrix} = (2-\lambda)^2 - 1 = 0$$

$$\lambda^2 - 4\lambda + 3 = 0 \rightarrow (\lambda - 1)(\lambda - 3) = 0, \qquad \lambda = 1, 3$$

하나의 고유벡터는 A 선형변환으로 자신의 1배가,

나머지 하나는 자신의 3배가 된다는 뜻

고유값과 고유벡터의 의미

2*2 행렬 A에 대한 고유값과 고유벡터

$$Ax = \lambda_1 x$$

$$if\ \lambda_1 = 1,$$

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 1 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \boxed{\begin{bmatrix} 1 \\ -1 \end{bmatrix}}$$

$$if\ \lambda_2 = 3,$$

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 3 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \boxed{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}$$

고유공간(Eigenspace): 각 고유값 λ에 대해 고유벡터들을 모아 놓은 공간

⋮

λ_1는 선형변환 A를 적용하면 길이가 1배 되는 벡터들의 span으로,

λ_2는 3배 되는 벡터들의 span으로 고유공간 형성

고유값과 고유벡터의 의미

2*2 행렬 A에 대한 고유값과 고유벡터

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \qquad \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} 2 \\ -2 \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} 5 \\ -5 \end{bmatrix} = \begin{bmatrix} 5 \\ -5 \end{bmatrix}, \qquad \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} n \\ -n \end{bmatrix} = \begin{bmatrix} n \\ -n \end{bmatrix}$$

고유벡터 $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ 에 대해 상수배를 해준 벡터에 행렬 A를 적용시켜도 그대로 나옴

⋮

고유공간 : $\mathrm{span}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$

($\lambda_2 = 3$ 인 경우 선형변환 A를 적용시키면 똑같이 3배 증가하는 고유벡터들이 산출)

2

고유값 분해

고유값 분해(Eigen Decomposition)

A 행렬은 n개의 고유벡터와 그에 해당하는 n개의 고유값을 가짐

각 고유벡터는 아래의 식 만족

$A : n * n$ 정방행렬

$$Av = \lambda v$$

다음과 같이 표현 가능

$$A[v_1\ v_2 \cdots v_n] = [\lambda_1 v_1\ \lambda_2 v_2\ \cdots \lambda_n v_n]$$

$$= [v_1\ v_2\ \cdots v_n]\begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}$$

고유값 분해(Eigen Decomposition)

$$A[v_1\ v_2 \cdots v_n] = [\lambda_1 v_1\ \lambda_2 v_2\ \cdots \lambda_n v_n]$$

$$= [v_1\ v_2\ \cdots v_n]\begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}$$

양변에 $[v_1\ \cdots v_n]$ 고유벡터 행렬의 역행렬을 곱하면

$$A = [v_1\ \cdots v_n]\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix}[v_1\ \cdots v_n]^{-1}$$

고유값 분해(Eigen Decomposition)

고유값 분해 *(Eigen Decomposition)*

행렬을 자신의 고유벡터들을 열벡터로 하는 행렬과

고유값을 대각원소로 하는 행렬의 곱으로 대각화 분해

$$A = \underbrace{[v_1 \ \cdots v_n]}_{Q} \underbrace{\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix}}_{\Lambda} \underbrace{[v_1 \ \cdots v_n]^{-1}}_{Q^{-1}}$$

고유값 분해(Eigen Decomposition)

고유값 분해를 하면 뭐가 좋은데?

⋮

행렬식, A의 거듭제곱, 역행렬, 대각합 등

연산을 손쉽게 계산!

고유값 분해(Eigen Decomposition)

행렬식

$$\det(A) = \det(\,Q\Lambda Q^{-1})$$

$$= \det(Q)\det(\Lambda)\det(Q)^{-1}$$

$$= \det(\,\Lambda)$$

$$= \lambda_1\lambda_2\cdots\lambda_n$$

A의 거듭제곱

$$A^k = (\,Q\Lambda Q^{-1})^k$$

$$= (\,Q\Lambda Q^{-1})(\,Q\Lambda Q^{-1})\cdots(\,Q\Lambda Q^{-1})$$

$$= Q\Lambda^k Q^{-1}$$

$$= Qdiag(\lambda_1^k, \dots, \lambda_n^k)Q^{-1}$$

역행렬

$$A^{-1} = (\,Q\Lambda Q^{-1})^{-1}$$

$$= Q\Lambda^{-1}Q^{-1}$$

$$= Qdiag(1/\lambda_1, \dots, 1/\lambda_n)Q^{-1}$$

대각합

$$tr(A) = tr(Q\Lambda Q^{-1})$$

$$= tr(\Lambda) \quad (tr(AB) = tr(BA))$$

$$= \lambda_1 + \cdots + \lambda_n$$

고유값 분해(Eigen Decomposition)

행렬식

$$\det(A) = \det(Q\Lambda Q^{-1})$$
$$= \det(Q)\det(\Lambda)\det(Q)^{-1}$$
$$= \det(\Lambda)$$
$$= \lambda_1\lambda_2\cdots\lambda_n$$

A의 거듭제곱

$$A^k = (Q\Lambda Q^{-1})^k$$
$$= (Q\Lambda Q^{-1})(Q\Lambda Q^{-1})\cdots(Q\Lambda Q^{-1})$$
$$= Q\Lambda^k Q^{-1}$$

항상 고유값 분해를 할 수 있는 것은 아님

정방행렬 A의 column이 모두 linearly independent해야 함

즉…

n개의 선형독립인 고유벡터를 가져야 함

역행렬

$$A^{-1} = (Q\Lambda Q^{-1})^{-1}$$
$$= Q\Lambda^{-1}Q^{-1}$$
$$= Q\,diag(1/\lambda_1, \dots, 1/\lambda_n)Q^{-1}$$

대각합

$$tr(A) = tr(Q\Lambda Q^{-1})$$
$$= tr(\Lambda) \quad (tr(AB) = tr(BA))$$
$$= \lambda_1 + \cdots + \lambda_n$$

2 고유값 분해(Eigen Decomposition)

대칭행렬과 고유값분해

모든 정방행렬이 고유값 분해가 가능한 것은 아님…

대칭행렬

$A^T = A$인 행렬

① 항상 고유값 대각화가 가능

② **직교행렬**로 대각화가 가능

⋮

고유벡터끼리 서로 orthogonal

Q의 역행렬은 곧 Q^T와 같음

⋮

$$A = Q\Lambda Q^T$$

3

주성분분석

주성분분석의 개념

주성분분석 *(PCA)*

원 데이터의 분산을 최대한 보존할 수 있는 방향으로
변수를 변형시켜서 '주성분'들을 찾는 차원축소 기법

종류

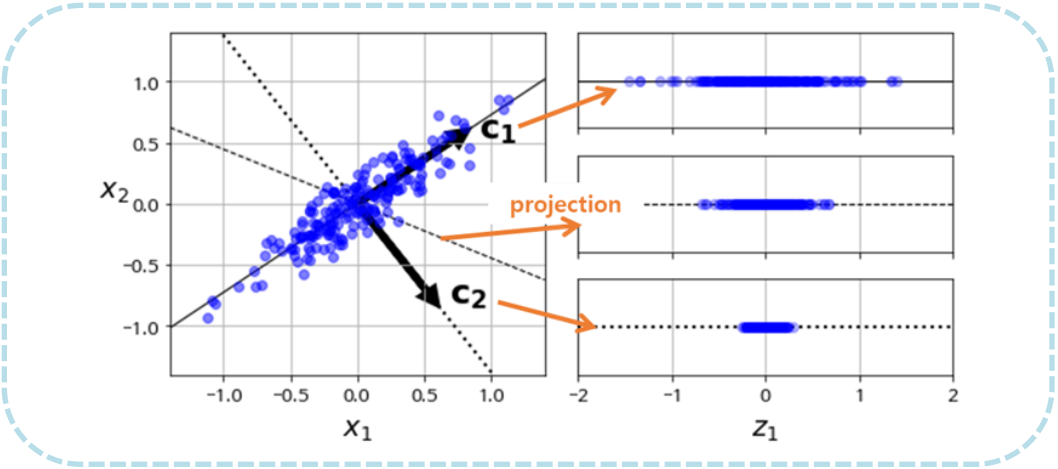
SVD를 이용한 방법

✓ Eigenvector를 이용한 방법

주성분분석의 개념

2차원 데이터에서 2개의 변수를 하나의 선으로 projection 시키고 싶다면,

어떤 선을 선택해야 가장 많은 분산을 보존할 수 있을까?



C1 직선으로의 projection이 데이터의 분산을 가장 잘 보존

⋮

C1 벡터를 구하기 위해 공분산 행렬의 Eigenvector를 찾아야 함

공분산 행렬의 수식적 의미

$$X = \begin{bmatrix} | & | & & | \\ X_1 & X_2 & \cdots & X_d \\ | & | & & | \end{bmatrix}$$

$$X^T X = \begin{bmatrix} - & X_1 & - \\ - & X_2 & - \\ & \cdots & \\ - & X_d & - \end{bmatrix} \begin{bmatrix} | & | & & | \\ X_1 & X_2 & \cdots & X_d \\ | & | & & | \end{bmatrix}$$

$$= \begin{pmatrix} dot(X_1, X_1) & dot(X_1, X_2) & \cdots & dot(X_1, X_d) \\ dot(X_2, X_1) & dot(X_2, X_2) & & dot(X_2, X_d) \\ \vdots & & \ddots & \vdots \\ dot(X_d, X_1) & dot(X_d, X_2) & \cdots & dot(X_d, X_d) \end{pmatrix}$$

i와 j번째 feature가 서로 얼마나 닮았는지

내적 연산을 통해 알 수 있음

숫자 ∝ 내적값

n으로 나눠주어 과도하게 커짐을 방지

공분산 행렬

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

공분산 행렬의 기하학적 의미

공분산 행렬

어떠한 데이터를 매핑한다면 분산과 공분산만큼 공간이 변화하게 되며

변수들의 분포를 나타냄

EXAMPLE

X축 방향으로 퍼진 정도

x, y축 방향으로 퍼진 정도

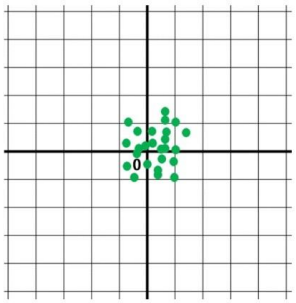
$$\begin{bmatrix} 3 & -2 \\ -2 & 4 \end{bmatrix}$$

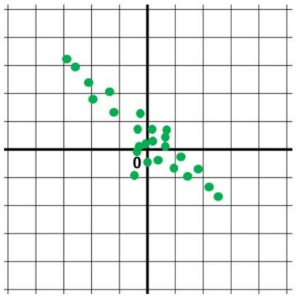
x, y축 방향으로 퍼진 정도

y축 방향으로 퍼진 정도

공분산 행렬의 기하학적 의미

EXAMPLE





$\begin{bmatrix} 3 & -2 \\ -2 & 4 \end{bmatrix}$ 행렬의 선형변환

우하향 대각선 방향으로 공간이 늘려짐

$\begin{bmatrix} 3 & 2 \\ 2 & 4 \end{bmatrix}$ 행렬의 선형변환

우상향 대각선 방향으로 공간이 늘여짐

공분산 행렬의 기하학적 의미

EXAMPLE

$\begin{bmatrix} 3 & -2 \\ -2 & 4 \end{bmatrix}$ 행렬의 선형변환

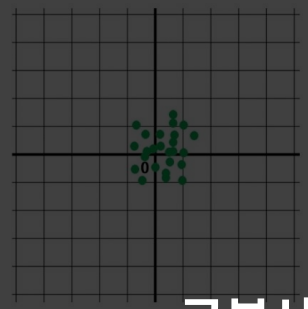
우하향 대각선 방향으로 공간이 늘여짐

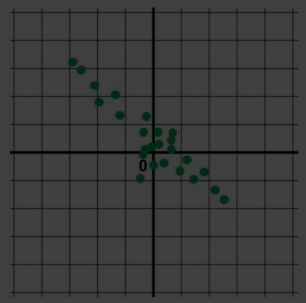
$\begin{bmatrix} 3 & 2 \\ 2 & 4 \end{bmatrix}$ 행렬의 선형변환

우상향 대각선 방향으로 공간이 늘려짐

공분산 행렬의 기하학적 의미

EXAMPLE





$\begin{bmatrix} 3 & -2 \\ -2 & 4 \end{bmatrix}$ 행렬의 선형변환

우하향 대각선 방향으로 공간이 늘여짐

공분산 행렬을 구한다면,

공간을 선형 변환시키는 정도로

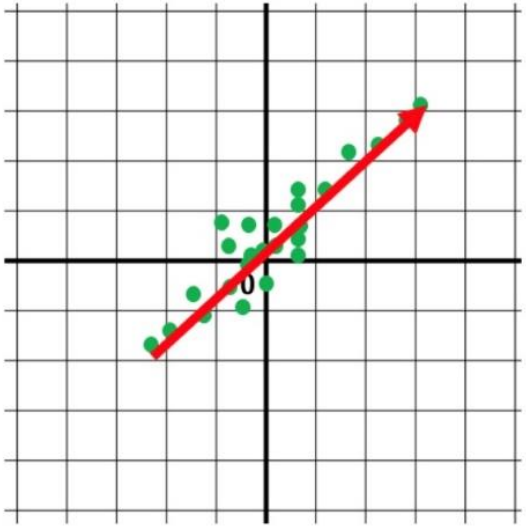
데이터가 어느 방향으로 어떻게 분포되어 있는지 확인 가능

$\begin{bmatrix} 3 & 2 \\ 2 & 4 \end{bmatrix}$ 행렬의 선형변환

우상향 대각선 방향으로 공간이 늘여짐

차원축소와 projection

이제 다시 PCA로…!



$$\begin{bmatrix} 3 & 2 \\ 2 & 4 \end{bmatrix}$$

빨간색 벡터 방향으로

projection 내리는 것이 데이터의 분산을 제일 잘 보존

대각선 방향으로

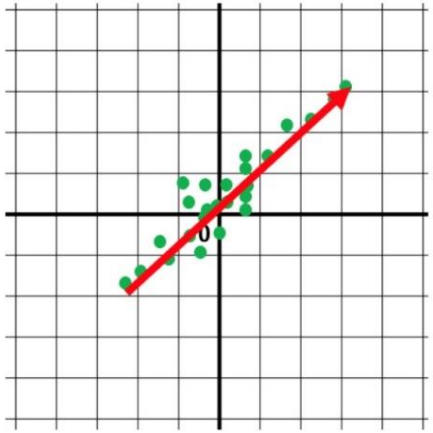
데이터를 잡고 늘리는 형태

선형 변환에서 빨간색 벡터는 방향이 변하지 않음

차원축소와 projection

고유 벡터는 그 행렬이 벡터에 작용하는 주축의 방향을 나타냄

공분산 행렬의 고유 벡터는 데이터의 분산 방향을 나타냄



이 방향으로 projection 시켰을 때,

데이터의 분산을 잘 보존할 수 있을 것이라는 추론 가능

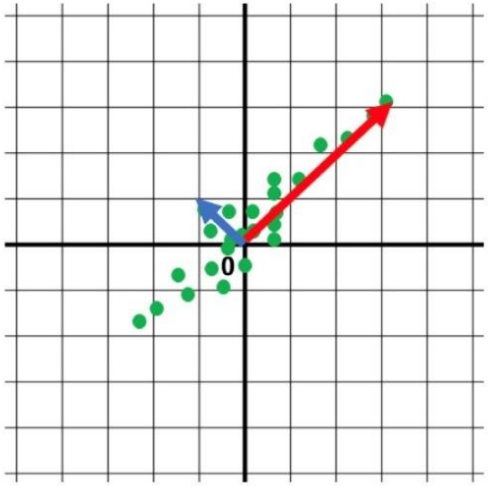
⋮

하지만, 2차원 공간에서 고유 벡터는 하나가 아님

차원축소와 projection

빨간색 고유벡터와 수직이 되는 방향의 벡터 역시 선형 변환 이후 방향이 변하지 않음

파란색 벡터도 A 행렬의 고유벡터

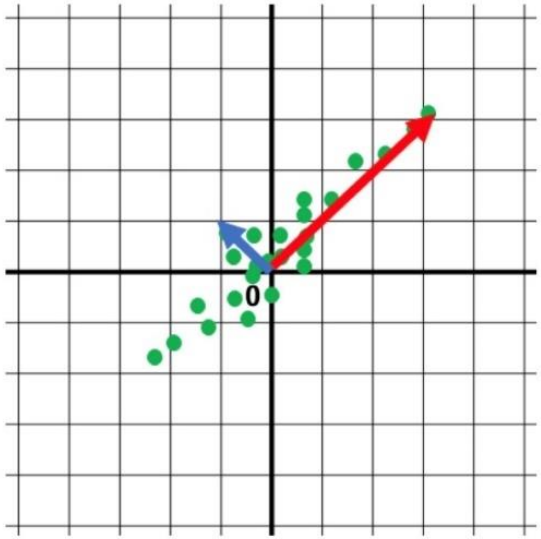


그렇다면 어디로 projection을 해야 하지?

⋮

고유값 사용하여 판단!

차원축소와 projection



빨간색 벡터 방향으로 데이터가 많이 분산되어 있으므로

빨간색 벡터의 고유값 > 파란색 벡터의 고유값

⋮

빨간색 벡터가 pc1, 파란색 벡터가 pc2가 됨

차원축소와 projection

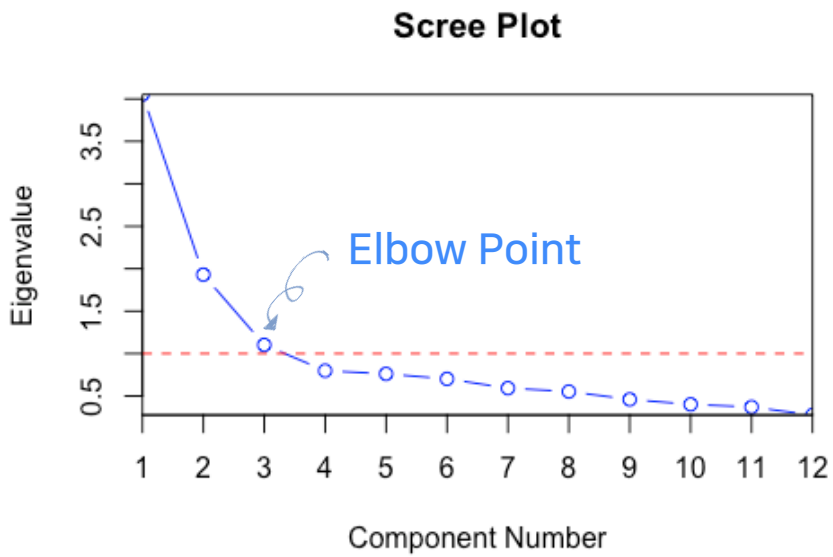
어디까지 차원을 감소시켜주는 것이 타당할까?

전체 데이터의 분산 중 90%만큼을 설명하는 차원까지 감소

$$\frac{\sum_{j=1}^{m} \lambda_j}{\sum_{i=1}^{d} \lambda_i} = 0.9$$

scree plot의 elbow point

차원축소와 projection



Elbow Point

eigenvalue를 기준으로

기울기가 갑자기 변화하는 지점

PC4 이후로는 정보량이 적음

유의한 변수가 되기 힘들다는 의미

⋮

PC3까지만 선택하는 것이 좋은 방안

PCA가 필요한 경우

PCA 과정 요약

데이터를 설명하는 **차원 축을 변경**하고

그 중에서 **많은 정보를 가진 축**만 남김

하지만…

원래 X data에 명시되었던 변수들의 축이 변경

해석력을 잃게 됨

언제 PCA를 고려해야 할지

변수 간의 많은 다중공선성

시각화

PCA가 필요한 경우

PCA 과정 요약

데이터를 설명하는 **차원 축을 변경**하고

그 중에서 **많은 정보를 가진 축**만 남김

하지만…

원래 X data에 명시되었던 변수들의 축이 변경

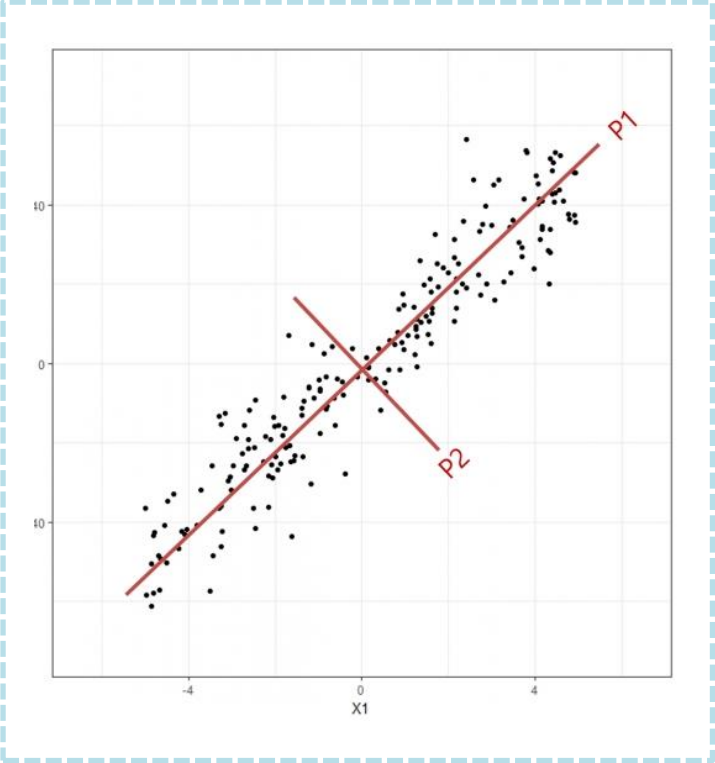
해석력을 잃게 됨

언제 PCA를 고려해야 할지

변수 간의 많은 다중공선성

시각화

다중공선성(Multicollinearity)



x변수들 간의 강한 상관관계가 나타남

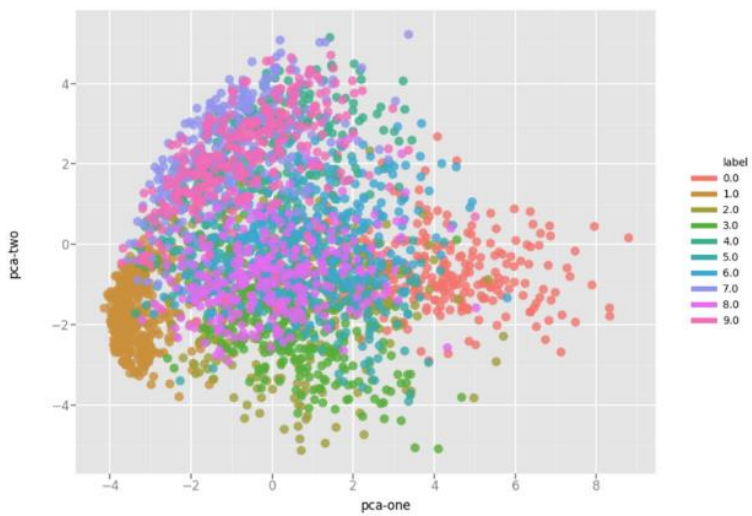
PCA 사용

주성분 PC1과 PC2는 직교

(두 변수 사이의 상관관계는 0)

PCA 클러스터링 시각화

PCA 클러스터링 시각화



고차원 데이터의 경우 시각화하기 어려움

⋮

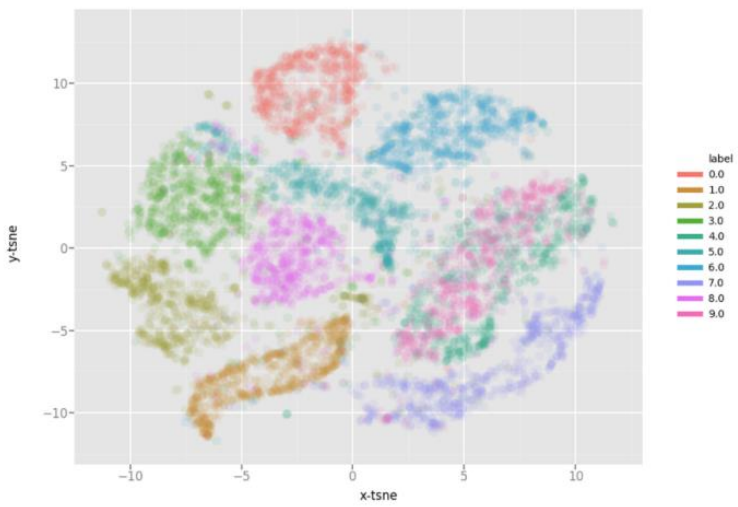
선형 연관성이 큰 변수들을 축소하면

변수 간 관계를 잘 표현할 수 있음

상대적으로 적은 축으로 시각화 가능!

t-SNE 클러스터링 시각화

t-SNE 클러스터링 시각화



참고…!

고차원 데이터 시각화 방법

t-SNE와 UMAP 같은 알고리즘이 대표적

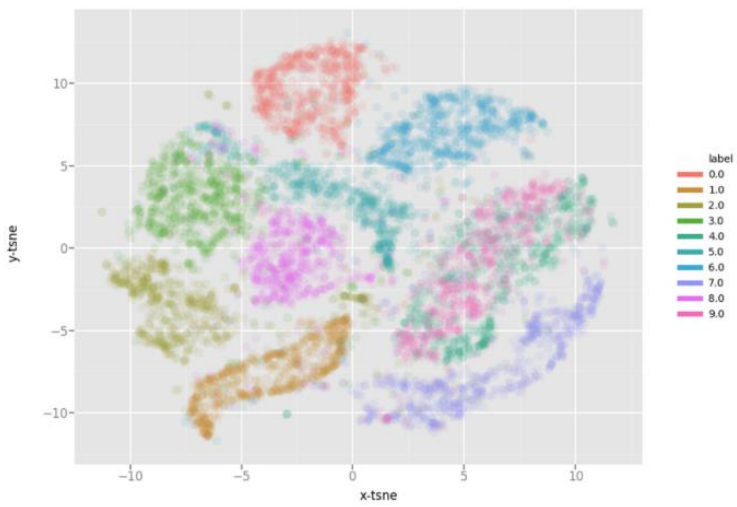
⋮

이웃 간의 거리를 잘 유지해주는 방식으로 데이터 축소

시각화 할 때 더 명확하게 데이터 구분

t-SNE 클러스터링 시각화

t-SNE 클러스터링 시각화



참고…!

고차원 데이터 시각화 방법

t-SNE와 UMAP 같은 알고리즘이 대표적

⋮

이웃 간의 거리를 잘 유지해주는 방식으로 데이터 축소

시각화 할 때 더 명확하게 데이터 구분

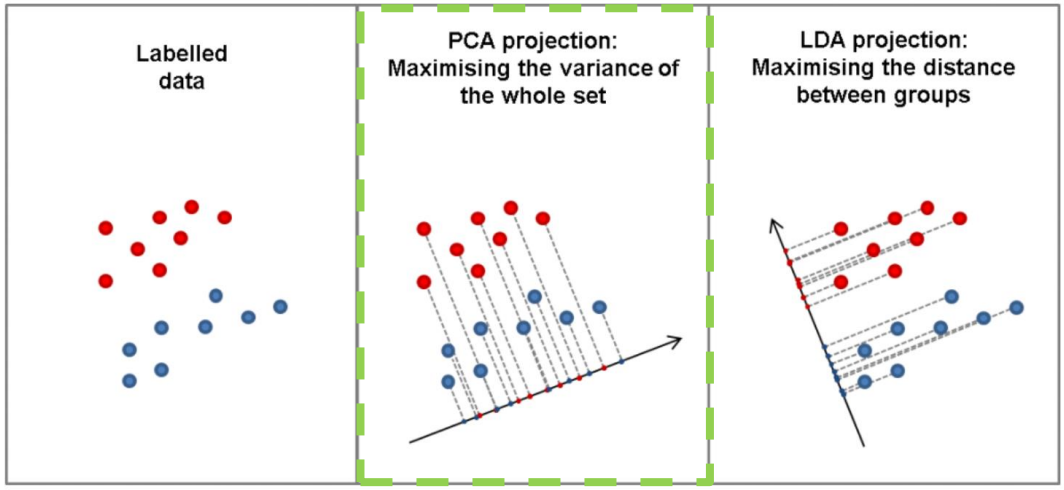
데이터의 특성이나 매핑되는 위치가 변해서 시각화에만 사용

새로운 x변수로 활용하지 X

그 외 차원축소 기법

LDA *(Linear Discriminant Analysis)*

차원축소 방법 중 하나로 클래스 분류까지 용이하게 해주는 장점이 있음



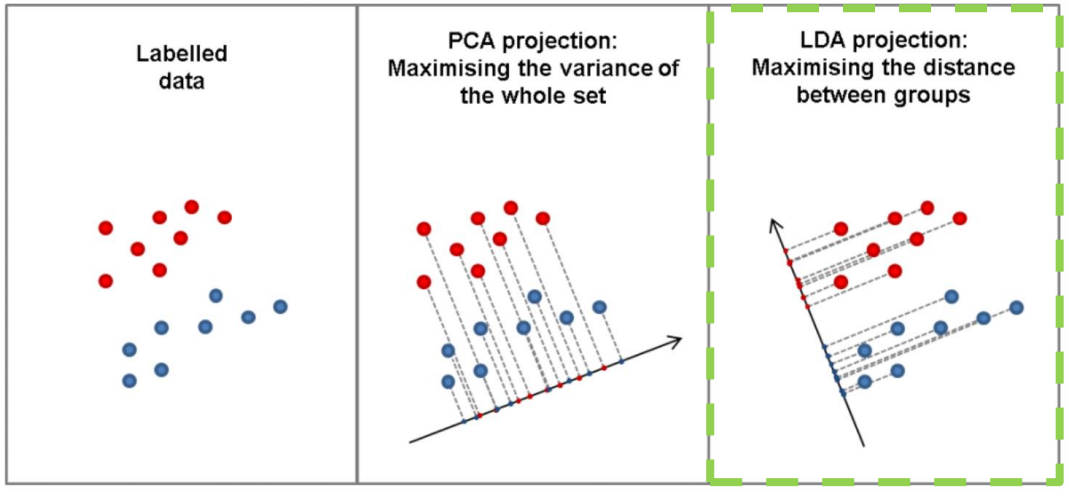
PCA로 축소

클래스와 상관없이 **마구잡이로** 축소됨

그 외 차원축소 기법

LDA *(Linear Discriminant Analysis)*

차원축소 방법 중 하나로 **클래스 분류**까지 용이하게 해주는 장점이 있음



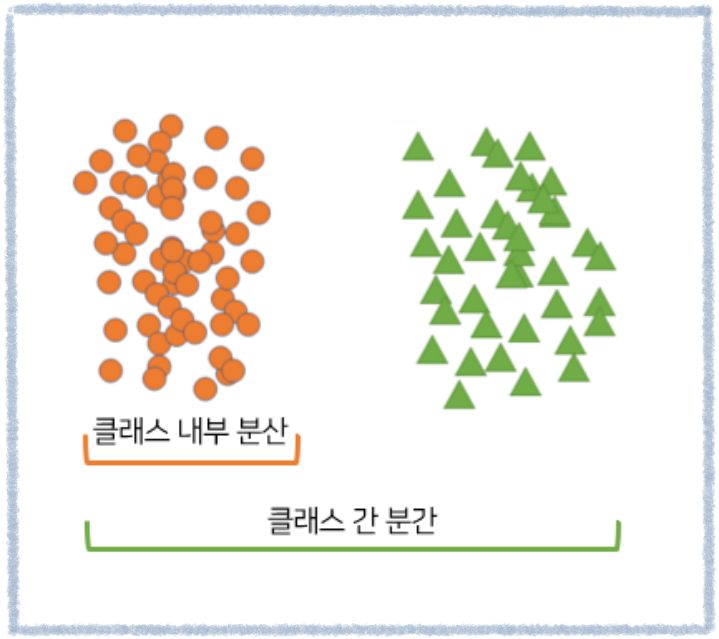
LDA로 축소

클래스가 **잘 구분**되어 있음

그 외 차원축소 기법

LDA *(Linear Discriminant Analysis)*

차원축소 방법 중 하나로 **클래스 분류**까지 용이하게 해주는 장점이 있음



LDA는 어떻게 클래스 분류를 용이하게 만들까?

클래스 **내부에서의 분산은 작게**,

클래스 간 분산은 크게 만들어 내는 축을 찾아 축소

그 외 차원축소 기법

정리하자면…

차원축소 기법

PCA

공분산 행렬을 구해서
고유벡터와 고유값을 구해 차원 축소

데이터의 **공분산 행렬**을 쓰는지

LDA

클래스 내부 분산과 클래스 간 분산 행렬을 생성
이들의 고유벡터와 고유값을 구해 차원 축소

클래스와 관련된 **분산**을 쓰는지

그 외 차원축소 기법

NMF *(Non-Negative Matrix Factorization)*

음수를 포함하지 않는 행렬의 행렬분해

Ex) 픽셀(양수)로 구성된 이미지 데이터

$$\begin{pmatrix} A & A & A & A & A & A \\ A & A & A & A & A & A \\ A & A & A & A & A & A \\ A & A & A & A & A & A \end{pmatrix} = \begin{pmatrix} W & W \\ W & W \\ W & W \\ W & W \end{pmatrix} \begin{pmatrix} H & H & H & H & H & H \\ H & H & H & H & H & H \end{pmatrix}$$

4 x 6

4 x 2 2 x 6

잠재요소 개수

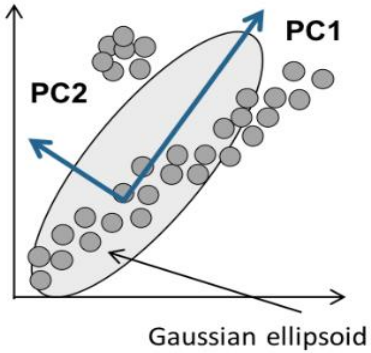
자세한 내용은 뒤에…!

그 외 차원축소 기법

데이터가 양수인 경우

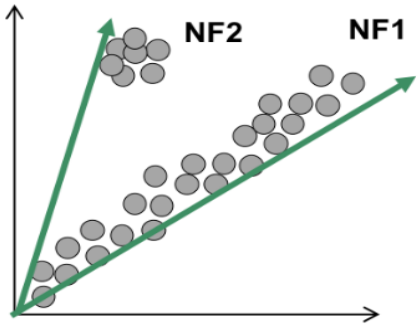
PCA

PCA does not 'sees' the data structure



NMF

NMF components are **non-negative**



더 높은 분류 성능!

언제나 그렇지는 않음

4

특이값 분해

WHY 특이값 분해?

PCA를 진행할 때 공분산 행렬 필요

고유값 분해는 정사각 행렬에서만 사용 가능

현실적 제약

① 대량 데이터는 공분산 행렬의 계산량 큼

② 직사각 행렬을 분해하는 경우가 더 많음

4 특이값 분해 (SVD)

WHY 특이값 분해?

① PCA를 진행할 때 공분산 행렬 필요

② 고유값 분해는 정사각 행렬에서만 사용 가능

고유값 분해를 직사각 행렬에 대해

일반화하는 과정 필요!

현실적 제약

① 대량 데이터는 공분산 행렬의 계산량 큼

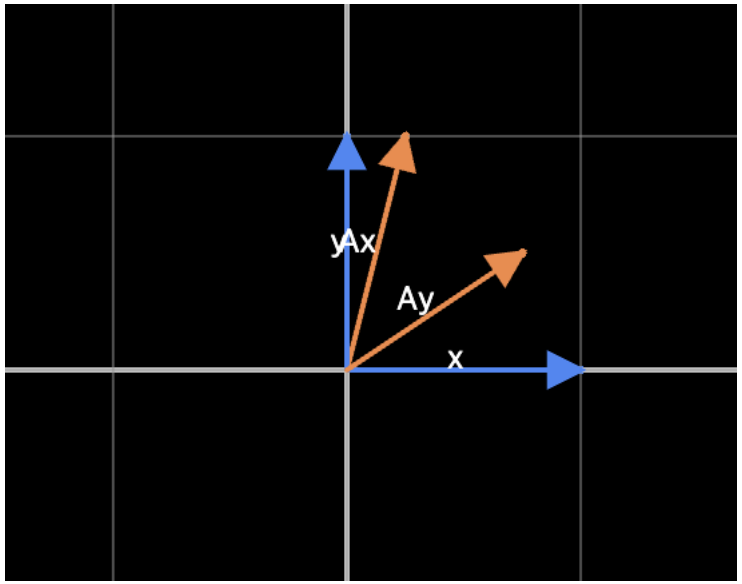
② 직사각 행렬을 분해하는 경우가 더 많음

특이값 분해 (SVD)

특이값 분해 (Singular Value Decomposition)

직교하는 벡터 집합에 대하여 선형변환 후에 그 크기는 변하지만

여전히 직교하는 직교집합을 찾는 것

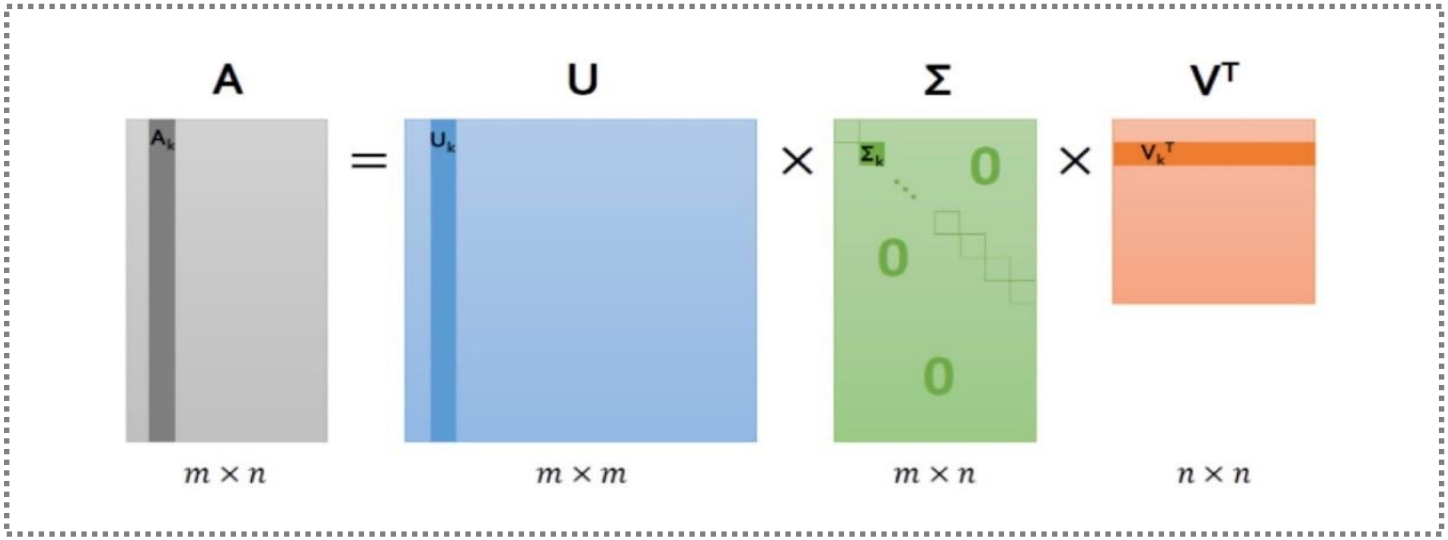


벡터 x,y가 직교할 때,

선형변환 시킨 Ax, Ay도 직교할까?

4 특이값 분해 (SVD)

특이값 분해의 구성요소



$$A = U\sum V^T$$

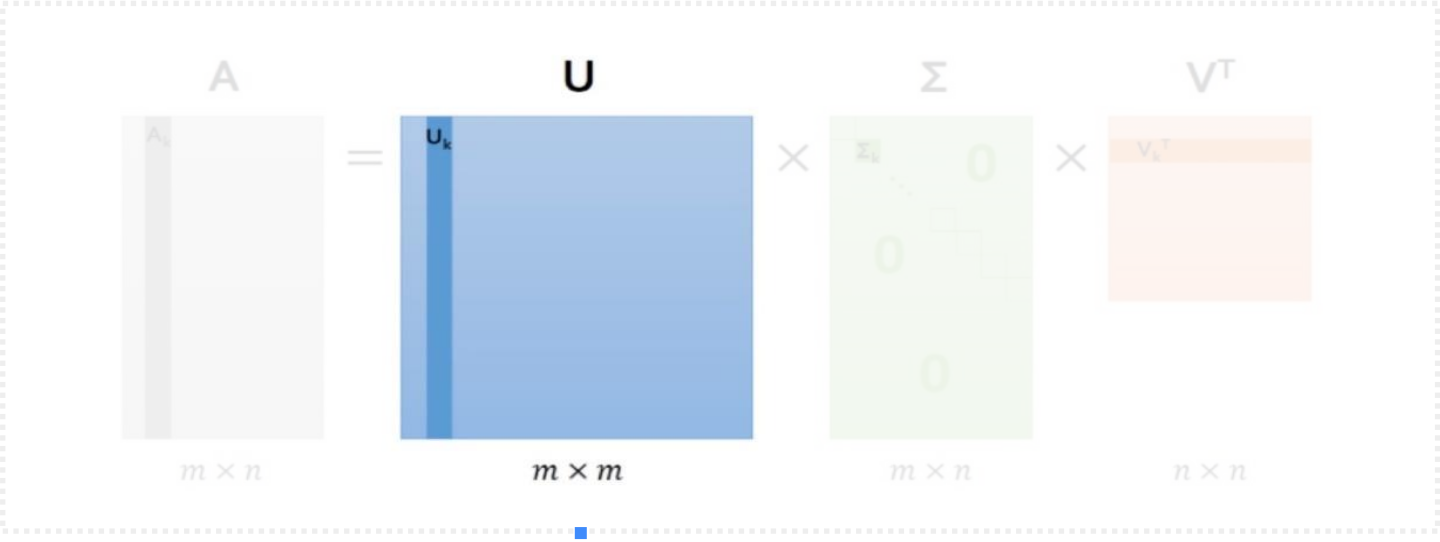
U : m x m 직교 행렬 (Orthogonal Matrix)

Σ : m x n 대각 행렬 (Diagonal Matrix)

V^T : n x n 직교 행렬 (Orthogonal Matrix)

4 특이값 분해 (SVD)

특이값 분해의 구성요소



AA^T를 고유값 분해 ($AA^T = U(\Sigma\Sigma^T)U^T$) 해서 얻어진 **직교행렬**

A의 Left Singular Vector

U : m x m 직교 행렬 (Orthogonal Matrix)

Σ : m x n 대각 행렬 (Diagonal Matrix)

V^T : n x n 직교 행렬 (Orthogonal Matrix)

특이값 분해의 구성요소

행렬 A를 자기 자신과 외적 후 고유값 분해

$$AA^T = U\Sigma V^T\left(U\Sigma V^T\right)^T$$

$$= U\Sigma V^T\left(V\Sigma^T U^T\right)$$

$$= U\Sigma V^T V\Sigma^T U^T$$

$$= U\Sigma\Sigma^T U^T \quad \left(\because V \text{ is orthogonal}, V^T V = I\right)$$

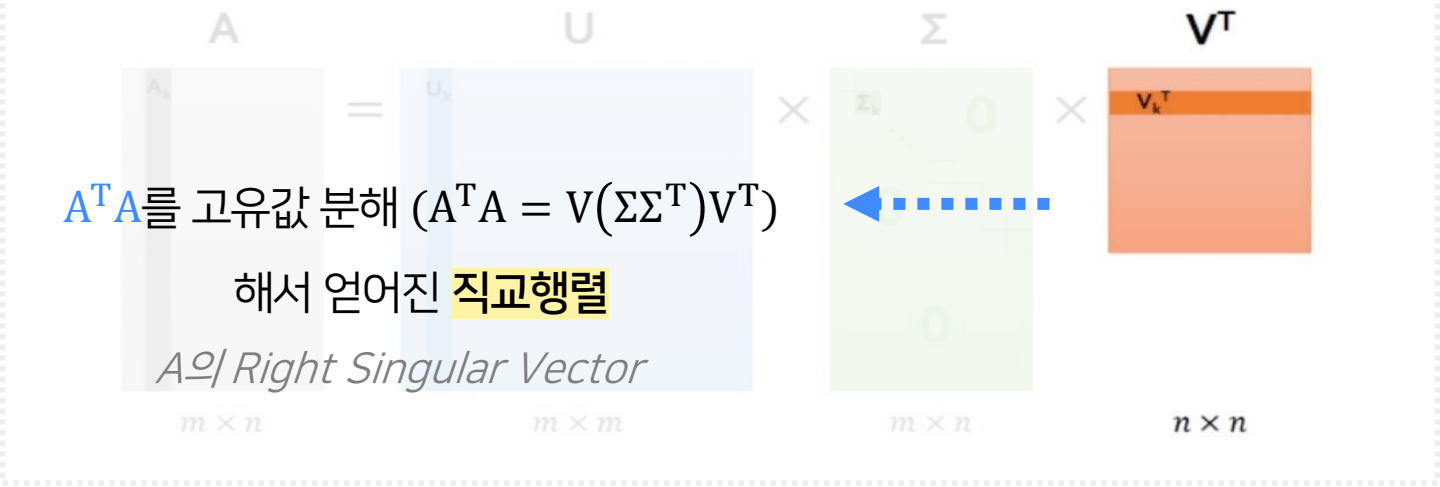
$$= U\Sigma^2 U^T\left(\because \Sigma\,\Sigma^T = \Sigma^2\right)$$

$$= \boxed{U\Lambda U^T} \longrightarrow$$ ***U*** 도출

$$U \in R^{m*m}\left(\because AA^T \in R^{m*m}\right)$$

4 특이값 분해 (SVD)

특이값 분해의 구성요소



$$A = U\Sigma V^T$$

U : m x m 직교 행렬 (Orthogonal Matrix)

Σ : m x n 대각 행렬 (Diagonal Matrix)

V^T : n x n 직교 행렬 (Orthogonal Matrix)

특이값 분해의 구성요소

행렬 A를 자기 자신과 외적 후 고유값 분해

$$A^TA = (U\Sigma V^T)^T U\Sigma V^T$$

$$= (V\Sigma^T U^T)U\Sigma V^T$$

$$= V\Sigma U^T U\Sigma^T V^T$$

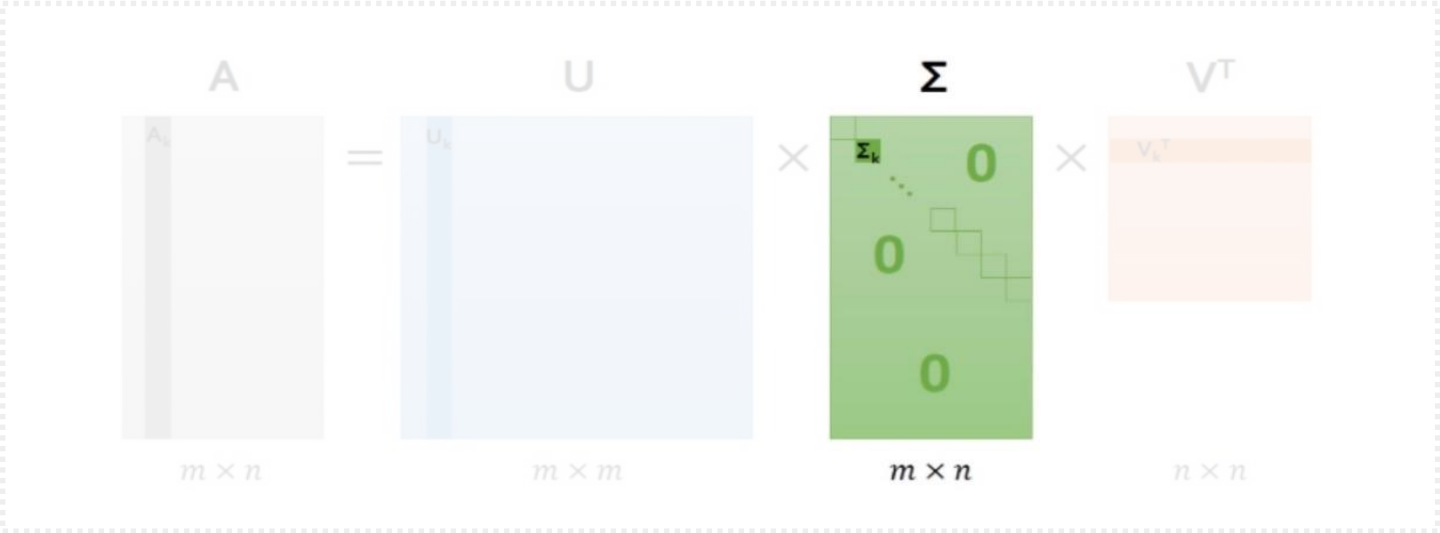
$$= V\Sigma\,\Sigma^T V^T (\because \text{U is orthogonal},\ U^TU = I)$$

$$= V\Sigma^2 V^T (\because \Sigma^T\Sigma = \Sigma^2)$$

$$= V\Lambda V^T \longrightarrow V^T\text{도출}$$

$$V^T \in R^{n*n} (\because A^TA \in R^{n*n})$$

특이값 분해의 구성요소



$$A = U\Sigma V^T$$

U : $m \times m$ 직교 행렬 (Orthogonal Matrix)

Σ : $m \times n$ 대각 행렬 (Diagonal Matrix)

V^T : $n \times n$ 직교 행렬 (Orthogonal Matrix)

AA^T, A^TA를 고유값 분해해서 나오는 고유값들의 Square Root를 **대각원소**로 하는 대각행렬

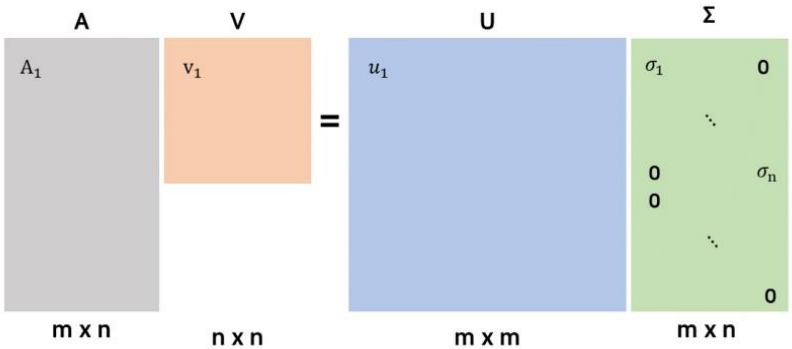
특이값

특이값 분해 (SVD)

$$A = U\Sigma V^T$$

양변에 V 곱함 ($VV^T = I$)

$$AV = U\Sigma$$



$AV = U\Sigma$

$AV = U\Sigma$

선형변환

U Column 벡터들을 스케일링

n차원의 V 행렬에서

더 고차원인 m차원의 U 행렬로 선형변환

특이값 분해 (SVD)

$$A = U\Sigma V^T$$

양변에 V 곱함 $(VV^T = I)$

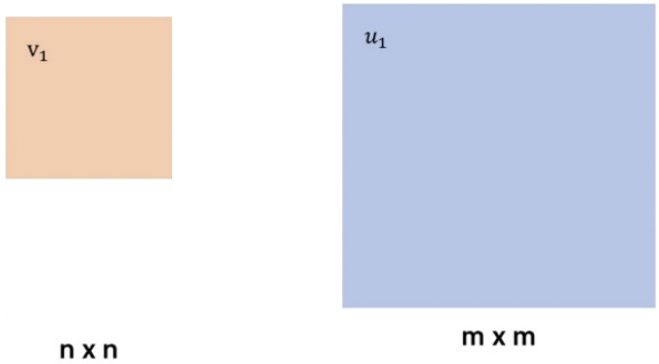
$$AV = U\Sigma$$

$AV = U\Sigma$

$$AV = U\Sigma$$

선형변환

U Column 벡터들을 스케일링



n차원의 V 행렬에서

더 고차원인 m차원의 U 행렬로 선형변환

특이값 분해 (SVD)

$AV =$

따라서…

선형변환

U Column 벡터들을 스케일링

벡터 x,y가 직교할 때,

선형변환 시킨 Ax, Ay도 직교할까?

달리 표현하면…

V를 선형변환할 때 그 크기는 σ_i 만큼 변하지만,

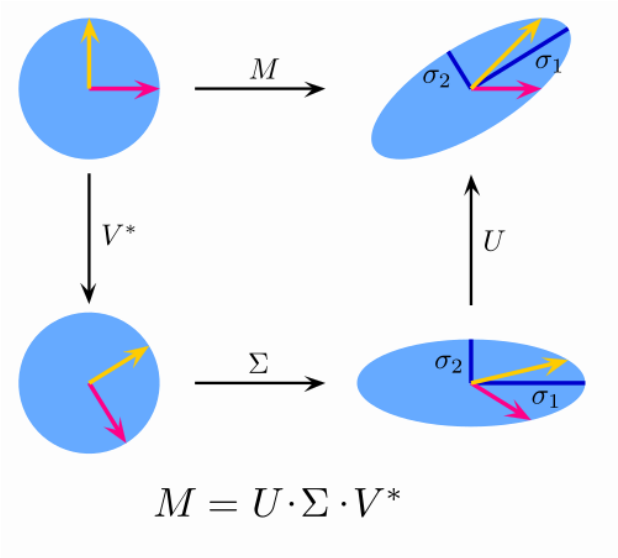
여전히 직교하는 U를 찾을 수 있는가?

n차원의 V 행렬에서

더 고차원인 m차원의 U 행렬로 선형변환

특이값 분해의 기하학적 의미

x → Mx의 연산과정



$$x \rightarrow Mx$$

$$M = U\Sigma V^T$$

$$x \rightarrow V^T x \rightarrow \Sigma V^T x \rightarrow U\Sigma V^T x\ (Mx)$$

⋮

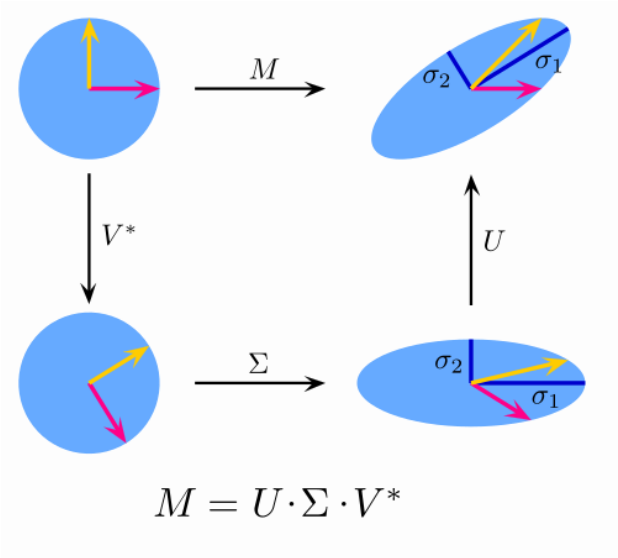
① x를 V^T를 거쳐 회전

② Σ를 통해 특이값 (σ_i) 만큼 **크기 변환**

③ U로 회전 변환

특이값 분해의 기하학적 의미

x → Mx의 연산과정



$x \rightarrow Mx$

$M = U\Sigma V^T$

$x \rightarrow V^T x \rightarrow \Sigma V^T x \rightarrow U\Sigma V^T x\ (Mx)$

⋮

① x를 V^T를 거쳐 회전

② Σ를 통해 특이값 (σ_i) 만큼 **크기 변환**

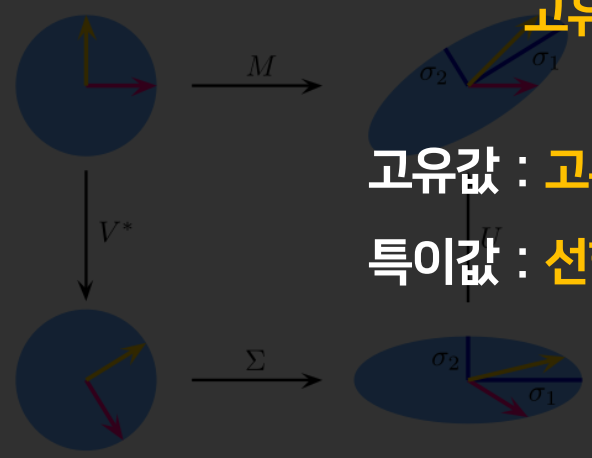
③ U로 회전 변환

⋮

A의 특이값들에 의해서만 형태변환!

특이값 분해의 기하학적 의미

$\mathbf{x} \rightarrow \mathbf{Mx}$의 연산과정



$M = U \cdot \Sigma \cdot V^*$

$x \rightarrow Mx$

$M = U\Sigma V^T$

$x \rightarrow V^T x \rightarrow \Sigma V^T x \rightarrow U\Sigma V^T x\ (Mx)$

① x를 V^T를 거쳐 회전

② Σ를 통해 특이값 (σ_i) 만큼 **크기 변환**

③ U로 회전 변환

고유값과 특이값의 차이?

고유값 : 고유벡터에 대한 스케일 factor

특이값 : 선형변환 자체의 스케일 factor

A의 특이값들에 의해서만 형태변환!

행렬분해의 이점

SVD로 분해한 행렬 ($A = U\Sigma V^T$)

$$= \begin{pmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_m \\ | & | & & | \end{pmatrix} \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_m \end{pmatrix} \begin{pmatrix} - & v_1^T & - \\ - & v_2^T & - \\ & \vdots & \\ - & v_n^T & - \end{pmatrix}$$

$$= \sigma_1 u_1 v_1^T + \cdots + \sigma_m u_m v_m^T$$

$u_k v_k^T$ 행렬들의 합

행렬 A와 같은 크기

4 특이값 분해 (SVD)

행렬분해의 이점

SVD로 분해한 행렬 ($A = U\Sigma V^T$)

$$= \begin{pmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_m \\ | & | & & | \end{pmatrix} \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_m \end{pmatrix} \begin{pmatrix} - & v_1^T & - \\ - & v_2^T & - \\ & \vdots & \\ - & v_n^T & - \end{pmatrix}$$

$$= \sigma_1 u_1 v_1^T + \cdots + \sigma_m u_m v_m^T$$

행렬 A와 같은 크기의 행렬들로 분해된 형태

원소의 값의 크기는 σ 에 따라 결정

행렬분해의 이점

SVD로 분해한 행렬 ($A = U\Sigma V^T$)

$$= \begin{pmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_m \\ | & | & & | \end{pmatrix} \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_m \end{pmatrix} \begin{pmatrix} - & v_1^T & - \\ - & v_2^T & - \\ & \vdots & \\ - & v_n^T & - \end{pmatrix}$$

$$= \sigma_1 u_1 v_1^T + \cdots + \sigma_m u_m v_m^T$$

SVD 통해 A 행렬을

여러 Layer로 쪼개는 것이 가능!

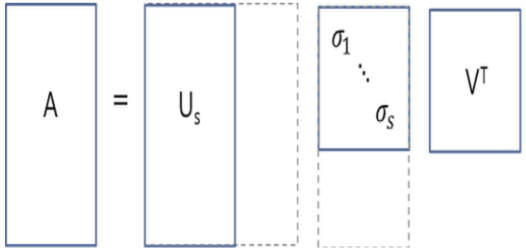
행렬 A와 같은 크기의 행렬들로 분해된 형태

원소의 값의 크기는 σ 에 따라 결정

다양한 SVD 형태

기존 Full SVD의 **연산량이 너무 많음** 형태 간략화!

Thin SVD



Σ 행렬의 비대각 파트 (아랫부분)와 해당하는 U 부분 제거

해당 부분은 연산값 항상 0, A 복원 가능

Compact SVD

Σ 행렬 내 비대각 파트, 특이값 0인 부분과 해당하는 U,V 부분 제거

특이값이 양수인 부분만 남김, A 복원 가능

Truncated SVD

Σ 행렬 내 특이값 중 상위 t개 선정

A 복원 불가능 하지만, A와 근사한 형태로 데이터 압축

4 특이값 분해 (SVD)

다양한 SVD 형태

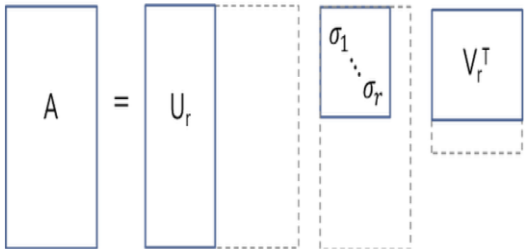
기존 Full SVD의 **연산량이 너무 많음** 형태 간략화!

Thin SVD

Σ 행렬의 비대각 파트 (아랫부분)과 해당하는 U 부분 제거

해당 부분은 연산값 항상 0, A 복원 가능

Compact SVD



Σ 행렬 내 비대각 파트, 특이값 0인 부분과 해당하는 U,V 부분 제거

특이값이 양수인 부분만 남김, A 복원 가능

Truncated SVD

Σ 행렬 내 특이값 중 상위 t개 선정

A 복원 불가능 하지만, A와 근사한 형태로 데이터 압축

다양한 SVD 형태

기존 Full SVD의 **연산량이 너무 많음** 형태 간략화!

Thin SVD

Σ 행렬의 비대각 파트 (아랫부분)과 해당하는 U 부분 제거

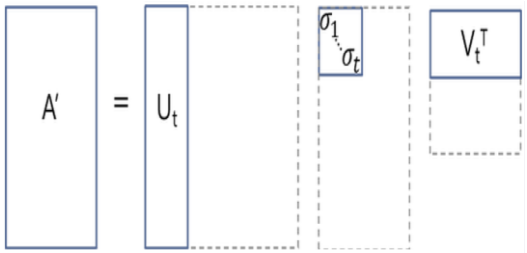
해당 부분은 연산값 항상 0,
A 복원 가능

Compact SVD

Σ 행렬 내 비대각 파트, 특이값 0인 부분과 해당하는 U,V 부분 제거

특이값이 양수인 부분만 남김,
A 복원 가능

Truncated SVD



Σ 행렬 내 특이값 중 상위 t개 선정

A 복원 불가능 하지만,
A와 근사한 형태로 데이터 압축

기존 행렬보다 연산 빠름,
데이터의 **핵심적 부분**만 사용

데이터 압축, 불필요한 노이즈 제거에 활용

5

특이값 분해의 응용

SVD to Regression

SVD를 반대로 돌리면 역행렬을 구할 수 없는 행렬의 의사역행렬을 구할 수 있음

분해된 행렬의 일부만 활용 → 데이터의 정보를 최대한 유지하는 부분 복원된 A를 구함

SVD를 이용하여 회귀식을 적합해보자...

$$X = U\Sigma V^{T}$$

X 데이터 분해

SVD to Regression

SVD를 반대로 돌리면 역행렬을 구할 수 없는 행렬의 의사역행렬을 구할 수 있음

분해된 행렬의 일부만 활용 → 데이터의 정보를 최대한 유지하는 부분 복원된 A를 구함

SVD를 이용하여 회귀식을 적합해보자...

$$X = U\Sigma V^{\mathrm{T}}$$

$$X^{*} = V\Sigma^{-1}U^{\mathrm{T}}$$

$\boldsymbol{V^T V} = \boldsymbol{I}, \boldsymbol{U^T U} = \boldsymbol{I}$ 이용

SVD to Regression

SVD를 반대로 돌리면 역행렬을 구할 수 없는 행렬의 의사역행렬을 구할 수 있음

분해된 행렬의 일부만 활용 → 데이터의 정보를 최대한 유지하는 부분 복원된 A를 구함

SVD를 이용하여 회귀식을 적합해보자...

$$X = U\Sigma V^{T}$$

$$X^{*} = V\Sigma^{-1}U^{T}$$

$$X^{*}X = V\Sigma U^{T}U\Sigma^{-1}V^{T} = V\Sigma\Sigma^{-1}V^{T} = VV^{T} = I_n$$

$\boldsymbol{V^T V} = \boldsymbol{I}, \boldsymbol{U^T U} = \boldsymbol{I}$ 이용

X^{*} 행렬은 x데이터의 **의사역행렬**

SVD to Regression

SVD를 반대로 돌리면 역행렬을 구할 수 없는 행렬의 의사역행렬을 구할 수 있음

분해된 행렬의 일부만 활용 → 데이터의 정보를 최대한 유지하는 부분 복원된 A를 구함

단, 데이터의 observation 수(n)가 variable 개수(p)보다 **많은 경우**만 적용

(n>p)

$$X = U\Sigma V^{\mathrm{T}}$$

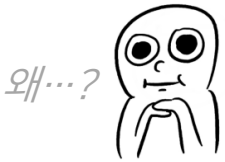
$\boldsymbol{V^T V} = \boldsymbol{I}, \boldsymbol{U^T U} = \boldsymbol{I}$ 이용

$$X^* = V\Sigma^{-1}U^{\mathrm{T}}$$

$$X^*X = V\Sigma U^{\mathrm{T}} U\Sigma^{-1}V^T = V\Sigma\Sigma^{-1}V^T = VV^T = I_n$$

X^* 행렬은 x데이터의 **의사역행렬**

SVD to Regression



SVD 자체는 n>p든, n<p든 다 상관없이 적용 가능

하지만 **SVD를 통해 회귀식을 적합하는 경우**는 n>p만 가능

n>p: $X^*X = V\Sigma U^T U\Sigma^{-1}V^T = V\Sigma\Sigma^{-1}V^T = VV^T = I_n$

n<p: $XX^* = U\Sigma V^T V\Sigma^{-1}U^T = U\Sigma\Sigma^{-1}U^T = UU^T = I_m$

각 경우에 단위행렬 성립!

SVD to Regression

∴ Ax=b의 양변 왼쪽에 A의 역행렬을 구하는 회귀식 상황

의사역행렬*원래행렬이 단위행렬이 되는 성질을 이용

→ n>p인 행렬이 가능

정리하면 SVD는 아무 행렬에 다 가능

하지만 **SVD를 통한 회귀식 적용은 n>p**일 때만 가능

SVD to Regression

회귀식에 적용하자면…

$$X^*X\beta \rightarrow X^*y$$

$$\hat{\beta} = X^*y$$

의사역행렬과 y벡터의 곱으로 회귀계수를 표현

5 특이값 분해 (SVD) 응용

Truncated SVD 이미지 압축

행렬 A (600 * 367)

이미지의 픽셀 값을 원소 값으로 함

Truncated SVD 이용

근사행렬 A'

특이값 367개 중 100개만 사용

사진이 흑백으로 변함

특이값 367개 중 20개만 사용

화질이 떨어짐

5 특이값 분해 (SVD) 응용

Truncated SVD 이미지 압축

행렬 A (600 * 367)

이미지의 픽셀 값을 원소 값으로 함

Truncated SVD 이용

근사행렬 A'

특이값 367개 중 100개만 사용

사진이 흑백으로 변함

특이값 367개 중 20개만 사용

화질이 떨어짐

5 특이값 분해 (SVD) 응용

잠재요인분석 (LSA)

토픽모델링

전체 문서의 주제를 연구자가 지정한 개수만큼 압축,

각 문서들이 어떤 주제를 가지는지 확인

LSA는 토픽모델링의 시초로 Truncated SVD를 사용

EXAMPLE

문서

문서 1: Pizza

문서 2: Pizza Hamburger Cookie

문서 3: Hamburger

문서 4: Ramen

문서 5: Sushi

문서 6: Ramen Sushi

단어-문서행렬 A

	문서1	문서2	문서3	문서4	문서5	문서6
Pizza	1	1	0	0	0	0
Hamburger	0	1	1	0	0	0
Cookie	0	1	0	0	0	0
Ramen	0	0	0	1	0	1
Sushi	0	0	0	0	1	1

변환 후 **특이값 분해** 진행

잠재요인분석 (LSA)

토픽모델링

전체 문서의 주제를 연구자가 지정한 개수만큼 압축,
각 문서들이 어떤 주제를 가지는지 확인

LSA는 토픽모델링의 시초로 Truncated SVD를 사용

EXAMPLE

문서

문서 1: Pizza
문서 2: Pizza Hamburger Cookie
문서 3: Hamburger
문서 4: Ramen
문서 5: Sushi
문서 6: Ramen Sushi

단어-문서행렬 A

	문서1	문서2	문서3	문서4	문서5	문서6
Pizza	1	1	0	0	0	0
Hamburger	0	1	1	0	0	0
Cookie	0	1	0	0	0	0
Ramen	0	0	0	1	0	1
Sushi	0	0	0	0	1	1

행렬로 변환

5 특이값 분해 (SVD) 응용

잠재요인분석 (LSA)

단어-문서행렬 A SVD진행

A Term x Document의 크기를 가진 행렬

	문서1	문서2	문서3	문서4	문서5	문서6
Pizza	1	1	0	0	0	0
Hamburger	0	1	1	0	0	0
Cookie	0	1	0	0	0	0
Ramen	0	0	0	1	0	1
Sushi	0	0	0	0	1	1

U

	T1	T2	T3	T4	T5
W1	0.6	0	0	0.7	-0.3
W2	0.6	0	0	-0.7	-0.3
W3	0.5	0	0	0	0.9
W4	0	0.7	-0.7	0	0
W5	0	0.7	0.7	0	0

$=$ $\times$

Σ

	T1	T2	T3	T4	T5	T6
T1	1.9	0	0	0	0	0
T2	0	1.7	0	0	0	0
T3	0	0	1	0	0	0
T4	0	0	0	1	0	0
T5	0	0	0	0	0.5	0

$\times$

V^T

	D1	D2	D3	D4	D5	D6
T1	0.3	0.9	0.3	0	0	0
T2	0	0	0	0.4	0.4	0.8
T3	0	0	0	-0.7	0.7	0
T4	0.7	0	-0.7	0	0	0
T5	-0.6	0.5	-0.6	0	0	0
T6	0	0	0	-0.6	-0.6	0.6

SVD 진행

= 문서와 단어 간의 관계에 어떤 topic이 **내재**되어 있다고 가정

⋮

이를 특이값 분해를 통해 찾고자 함

5 특이값 분해 (SVD) 응용

잠재요인분석 (LSA)

$A = U\Sigma V^T$ 연산은 Term x Document의 관계를 다음과 같이 표현

$$Term \times Document$$

$$= (Topic \times Term)(Topic \times Topic)(Document \times Topic)$$

U

	T1	T2	T3	T4	T5
W1	0.6	0	0	0.7	-0.3
W2	0.6	0	0	-0.7	-0.3
W3	0.5	0	0	0	0.9
W4	0	0.7	-0.7	0	0
W5	0	0.7	0.7	0	0

$\times$

Σ

	T1	T2	T3	T4	T5	T6
T1	1.9	0	0	0	0	0
T2	0	1.7	0	0	0	0
T3	0	0	1	0	0	0
T4	0	0	0	1	0	0
T5	0	0	0	0	0.5	0

$\times$

V^T

	D1	D2	D3	D4	D5	D6
T1	0.3	0.9	0.3	0	0	0
T2	0	0	0	0.4	0.4	0.8
T3	0	0	0	-0.7	0.7	0
T4	0.7	0	-0.7	0	0	0
T5	-0.6	0.5	-0.6	0	0	0
T6	0	0	0	-0.6	-0.6	0.6

⋮

각 행렬을 확인하는 것으로

단어와 토픽의 관계, 토픽의 영향력, 토픽과 문서의 관계 확인 가능

5 특이값 분해 (SVD) 응용

잠재요인분석 (LSA)

$A = U\Sigma V^T$ 연산은 Term x Document의 관계를 다음과 같이 표현

$$Term \times Document$$

문서당 하나씩 나타나고 있는 주제들을 압축

$$= (Topic \times Term)(Topic \times Topic)(Document \times Topic)$$

가장 큰 순으로 **2개의 토픽값**만을 사용하여 Truncated SVD 진행

$$\underbrace{U}_{} \quad \underbrace{\Sigma}_{} \quad \underbrace{V^T}_{}$$

U

	T1	T2	T3	T4	T5
W1	0.6	0	0	0.7	-0.3
W2	0.6	0	0	-0.7	-0.3
W3	0.5	0	0	0	0.9
W4	0	0.7	-0.7	0	0
W5	0	0.7	0.7	0	0

$\times$

Σ

	T1	T2	T3	T4	T5	T6
T1	1.9	0	0	0	0	0
T2	0	1.7	0	0	0	0
T3	0	0	1	0	0	0
T4	0	0	0	1	0	0
T5	0	0	0	0	0.5	0

$\times$

V^T

	D1	D2	D3	D4	D5	D6
T1	0.3	0.9	0.3	0	0	0
T2	0	0	0	0.4	0.4	0.8
T3	0	0	0	-0.7	0.7	0
T4	0.7	0	-0.7	0	0	0
T5	-0.6	0.5	-0.6	0	0	0
T6	0	0	0	-0.6	-0.6	0.6

각 행렬을 확인하는 것으로

단어와 토픽의 관계, 토픽의 영향력, 토픽과 문서의 관계 확인 가능

5 특이값 분해 (SVD) 응용

잠재요인분석 (LSA)

원래 행렬 A

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Truncated SVD로 만든 유사행렬 A'

$$\begin{bmatrix} 0.342 & 1.026 & 0.342 & 0 & 0 & 0 \\ 0.342 & 1.026 & 0.342 & 0 & 0 & 0 \\ 0.285 & 0.855 & 0.285 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.476 & 0.476 & 0.952 \\ 0 & 0 & 0 & 0.476 & 0.476 & 0.952 \end{bmatrix}$$

오차는 있지만 A의 **경향성을 유지**하고 있다는 것을 확인

U: 문서와 토픽의 관계를 알 수 있음

V^T :잠재되어 있는 주제가 무엇인지 판단

U

	T1	T2	T3	T4	T5
W1	0.6	0	0	0.7	-0.3
W2	0.6	0	0	-0.7	-0.3
W3	0.5	0	0	0	0.9
W4	0	0.7	-0.7	0	0
W5	0	0.7	0.7	0	0

×

Σ

	T1	T2	T3	T4	T5	T6
T1	1.9	0	0	0	0	0
T2	0	1.7	0	0	0	0
T3	0	0	1	0	0	0
T4	0	0	0	1	0	0
T5	0	0	0	0	0.5	0

×

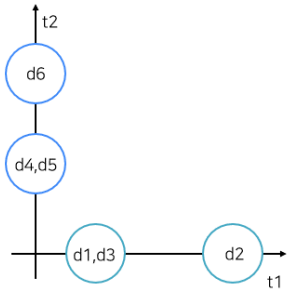
V^T

	D1	D2	D3	D4	D5	D6
T1	0.3	0.9	0.3	0	0	0
T2	0	0	0	0.4	0.4	0.8
T3	0	0	0	-0.7	0.7	0
T4	0.7	0	-0.7	0	0	0
T5	-0.6	0.5	-0.6	0	0	0
T6	0	0	0	-0.6	-0.6	0.6

잠재요인분석 (LSA)

토픽과 문서 간의 관계를 알아보기 위해 계산하면 (ΣV^T) ...

토픽과 문서 간의 관계



	D1	D2	D03	D4	D5	D6
T1	0.57	1.71	0.57	0	0	0
T2	0	0	0	0.68	0.68	1.36

해석하자면…

d1, d2, d3 → t1(양식)

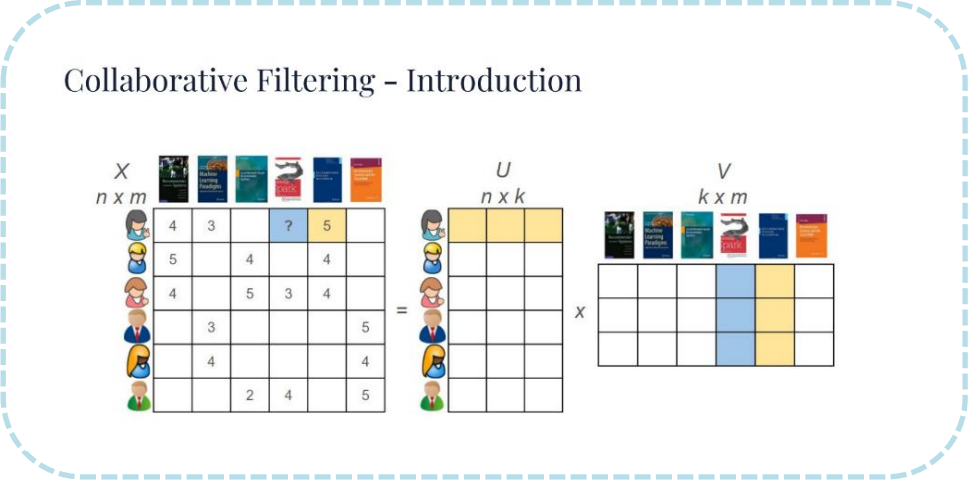
d4, d5, d6 → t2(일식)

5 특이값 분해 (SVD) 응용

잠재요인분석 (LSA)

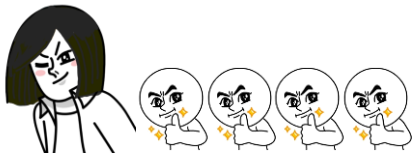
SVD는 **다양한 분야**에서 활용되고 있음

e.g) 추천시스템에서 사용자들의 평점 행렬을 분해, 사용자 – 아이템 간 상호작용을 표현



추천시스템에 대한 자세한 내용은

데마팀 3주차 클린업 참고



감사합니다