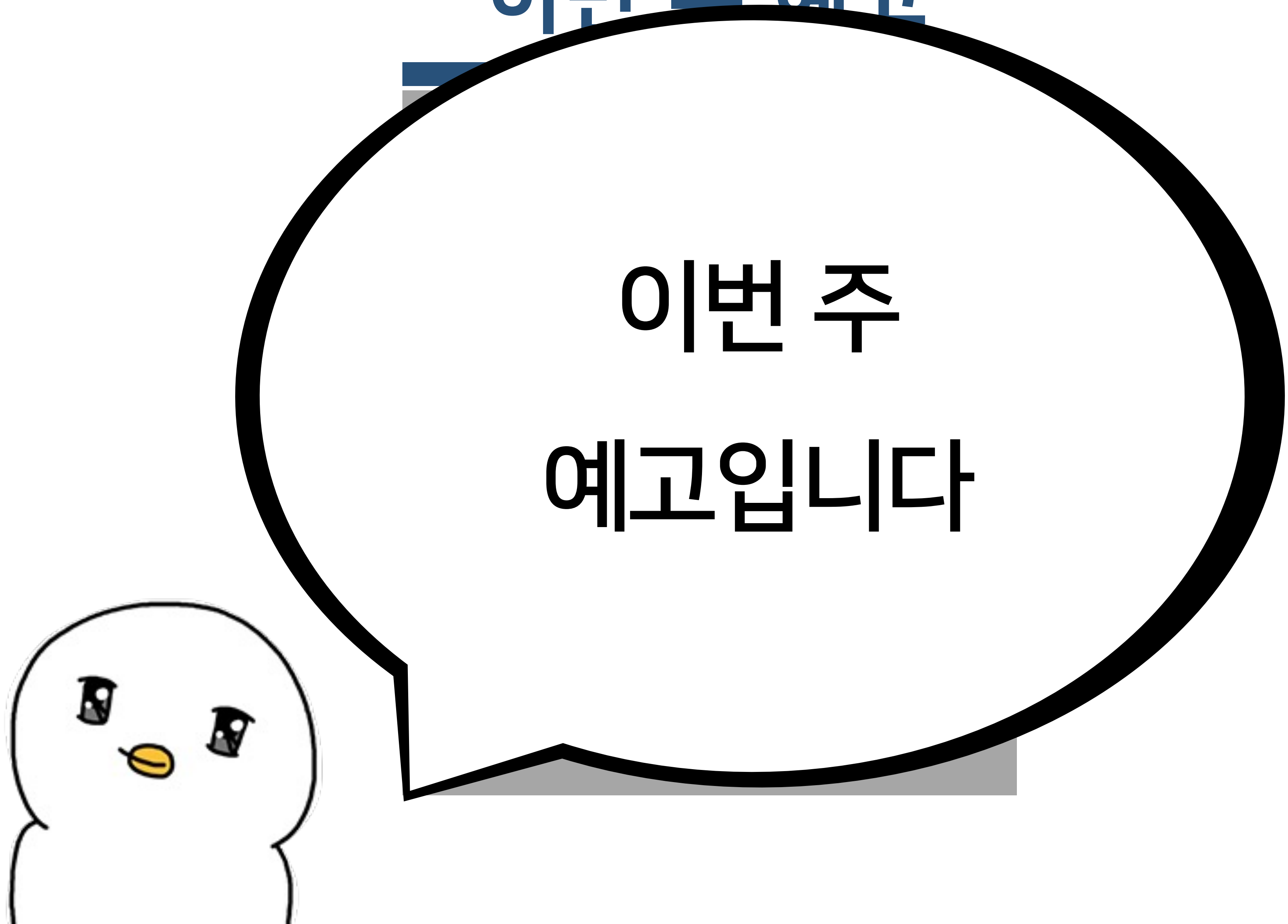
선형대수학팀

3팀

김다민
이지원
조성우
김수인
방건우

이번 주 예고



이번 주 예고

1. 행렬식,노름,내적,직교성

2. 정사영과 회귀

3.평균, 분산, 상관계수의 기하학적 해석

4.고유값과 고유 벡터

5.특이값 분해

1

행렬식, 노름, 내적, 직교성

행렬식 *Determinant*

행렬식 *Determinant*

$n \times n$ 정방행렬에 스칼라를 대응시키는 함수이며 $\det(A)$라고 표현

$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ 일 때, 행렬식은 $\det(A) = ad - cb$

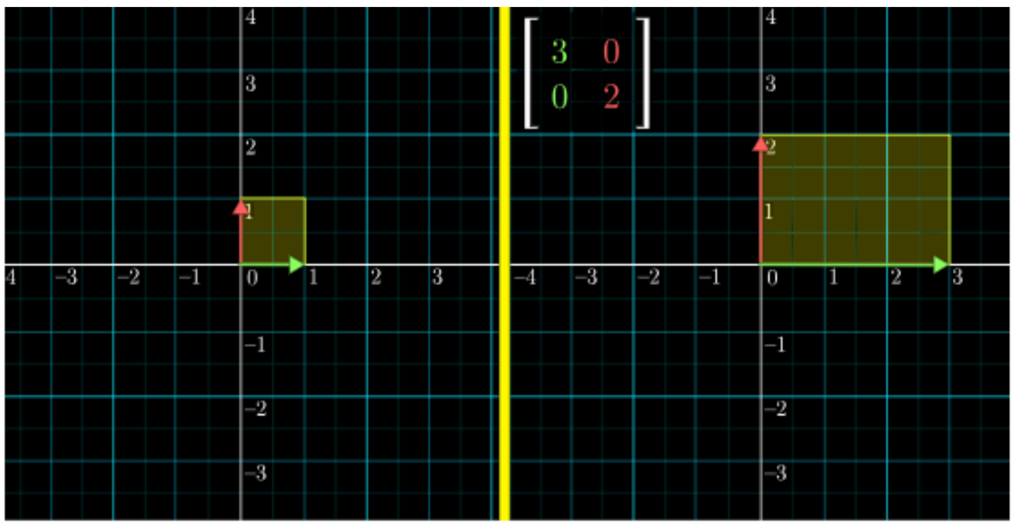
$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$ 일 때, 행렬식은 여인수분해를 통해 계산

행렬식 *Determinant*

행렬식의 기하학적 해석

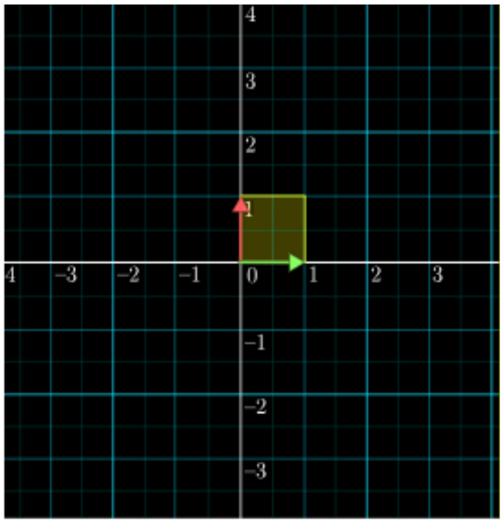
선형 변환을 할 때 선형 변환이 공간을 **얼마나 변환**시키는지

행렬식을 통해 알 수 있음



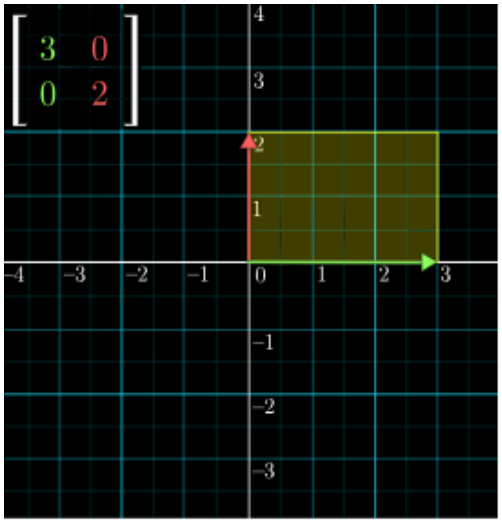
EX) 기저벡터 $\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}$ 에 행렬 $A = \begin{bmatrix}3 & 0\\0 & 2\end{bmatrix}$을 통한 선형변환

행렬식 *Determinant*



선형변환 전 넓이 = 1

기저벡터 $\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}$



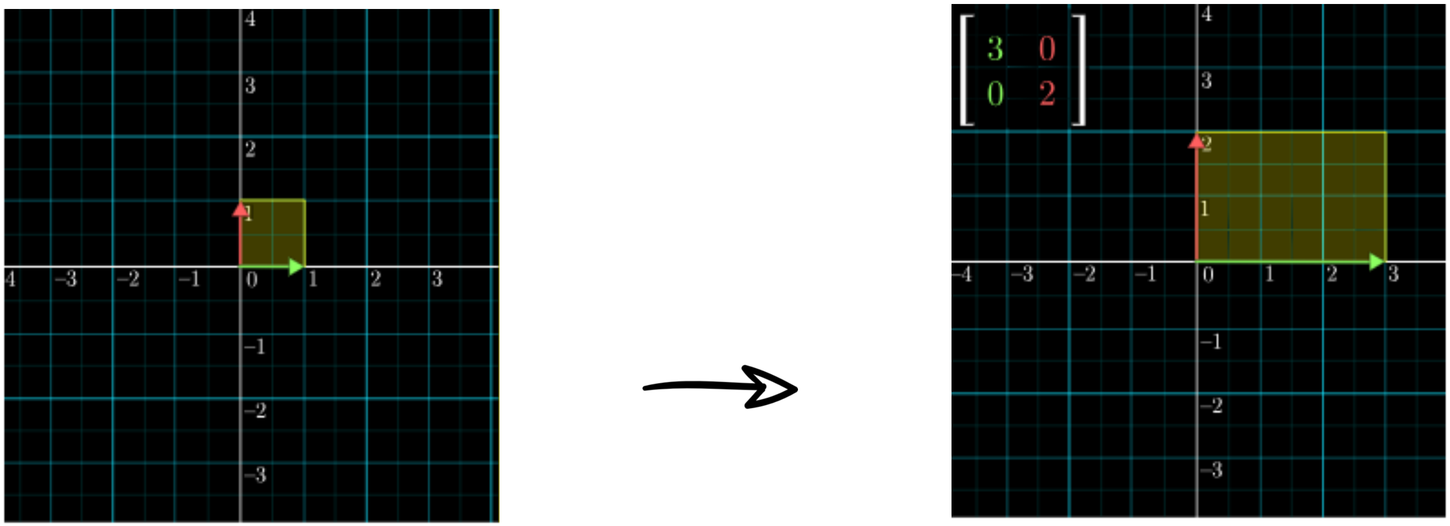
선형변환 후 넓이 = 6

기저벡터 $\begin{bmatrix}3\\0\end{bmatrix}, \begin{bmatrix}0\\2\end{bmatrix}$

$$\det(A) = \begin{vmatrix}3 & 0\\0 & 2\end{vmatrix} = 6$$

선형변환 후 평행사변형의 넓이

행렬식 *Determinant*



선형변환 전 넓이 = 1

기저벡터 $\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}$

선형변환 후 넓이 = 6

기저벡터 $\begin{bmatrix}3\\0\end{bmatrix}, \begin{bmatrix}0\\2\end{bmatrix}$

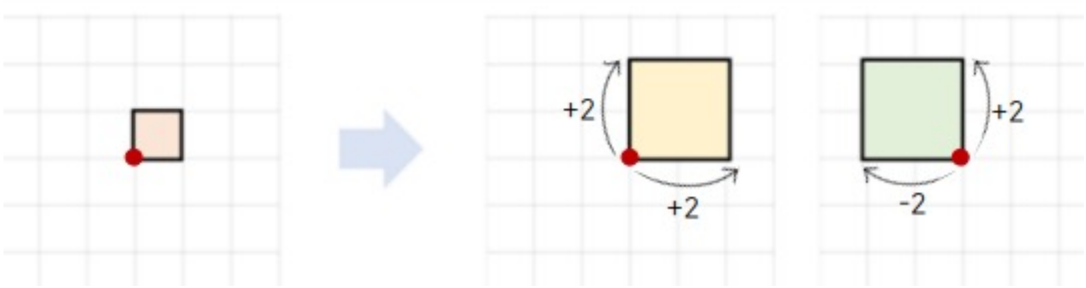
행렬식은 2차원 실수 공간 $\mathbb{R}^2$에서는 **넓이**

3차원 실수 공간 $\mathbb{R}^3$ 에서는 **부피**와 관련있음

행렬식 *Determinant*

행렬식의 부호

행렬식이 음수인 경우 **공간이 뒤집힌 경우**를 의미



좌측 선형변환

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \det(A) = 4$$

우측 선형변환

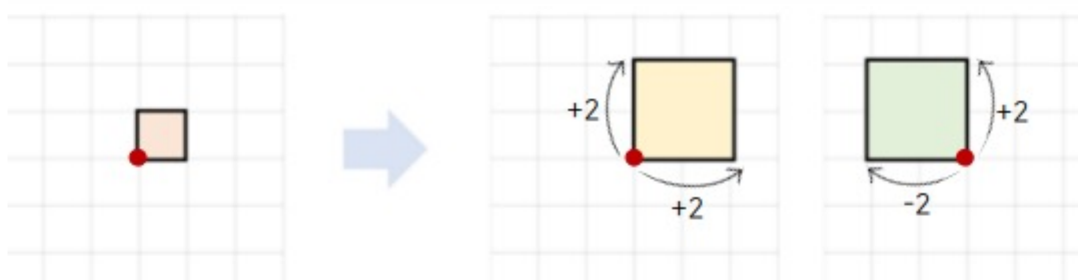
$$A = \begin{bmatrix} -2 & 0 \\ 0 & 2 \end{bmatrix}, \det(A) = -4$$

면적은 같지만 **좌우반전**된 상태

행렬식 *Determinant*

행렬식의 부호

행렬식이 음수인 경우 **공간이 뒤집힌 경우**를 의미



좌측 선형변환

$A = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \det(A) = 4$

우측 선형변환

$A = \begin{bmatrix} -2 & 0 \\ 0 & 2 \end{bmatrix}, \det(A) = -4$

행렬식의 부호 = 공간의 **반전** 상태

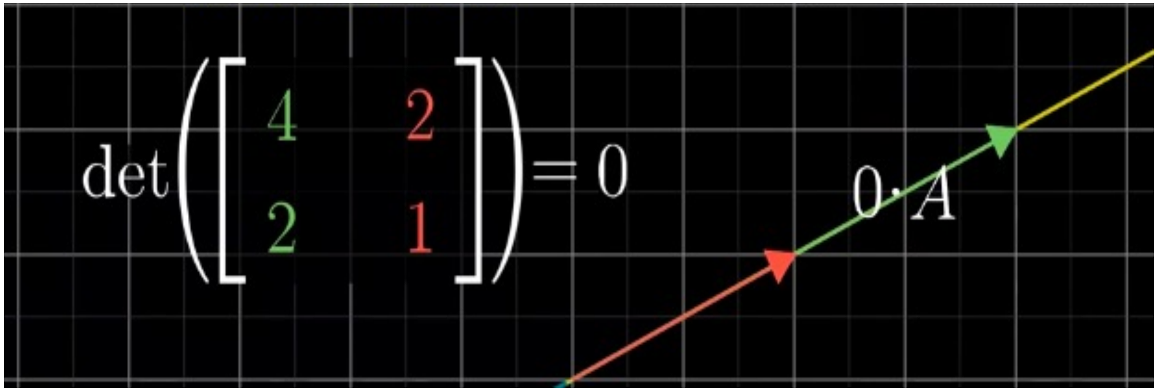
면적은 같지만 **좌우반전**된 상태

행렬식 *Determinant*

행렬식이 0인 것의 의미

$\det(A) = 0$이면 행렬의 넓이나 부피가 없음

행렬A가 **공간을 압축하는 선형변환**임을 의미



행렬식 *Determinant*

행렬식의 쓰임

행렬식을 통해 **역행렬**의 존재 유무 판단

역행렬이 존재

= $Ax = b$가 유일한 해를 갖는다.

= x 와 Ax가 서로 일대일 대응이다.

공간이 압축되는 선형변환의 경우, **역행렬이 존재하지 않음**

선형대수학팀 1주차 클린업 복습

행렬식 *Determinant*

행렬식의 쓰임

행렬식을 통해 **역행렬**의 존재 유무 판단

역행렬이 존재

$\leftrightarrow A\boldsymbol{x} = \boldsymbol{b}$가 유일한 해를 갖는다.

$\leftrightarrow \boldsymbol{x}$ 와 $A\boldsymbol{x}$가 서로 일대일 대응이다.

공간이 압축되는 선형변환의 경우, **역행렬이 존재하지 않음**

행렬식 *Determinant*

행렬식의 쓰임

행렬식을 통해 **역행렬**의 존재 유무 판단

$\det(A) = 0$ 인 경우 행렬A는 **공간을 압축**시키는 선형변환

이때, 행렬A의 역행렬은 **존재하지 않음**

역행렬이 존재

=$Ax = b$가 유일한 해를 갖는다.

= x 와 Ax가 서로 일대일 대응이다.

공간이 압축되는 선형변환의 경우, **역행렬이 존재하지 않음**

노름 *Norm*

노름 *Norm*

노름은 **벡터의 크기**를 의미

거리와 유사한 개념으로 이해

$$\|v\|_p = Lp = \sqrt[p]{|v_1|^p + |v_2|^p + \cdots + |v_n|^p}$$

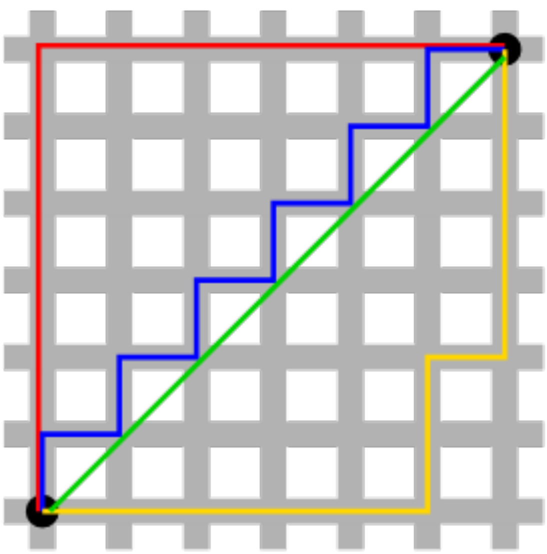
통계에서는 P=1인 L_1 Norm과 P=2인 L_2 Norm을 주로 사용

노름 *Norm*

맨해튼 노름 *L1 norm*

각 벡터들의 **절댓값**을 합한 결과

직선이 아닌 **좌표축**을 따라 움직이는 거리



$$L_1 = |v_1| + |v_2| + \cdots + |v_n|$$

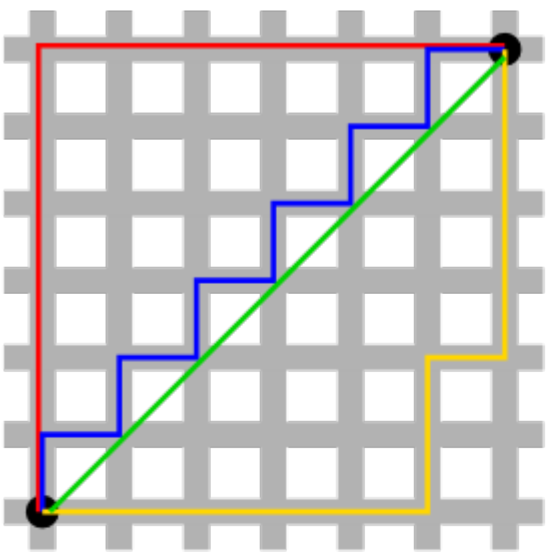
그림에서
빨간색, 파란색, 노란색 선으로 표현가능

노름 *Norm*

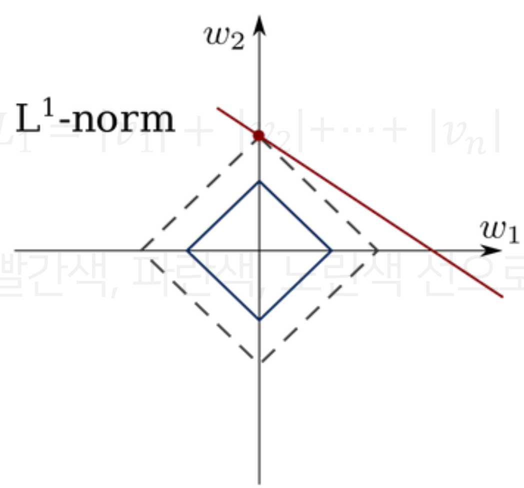
맨해튼 노름 *L1 norm*

각 벡터들의 **절댓값**을 합한 결과

직선이 아닌 **좌표축**을 따라 움직이는 거리



Lasso regression에서 β의 제약조건으로 쓰임

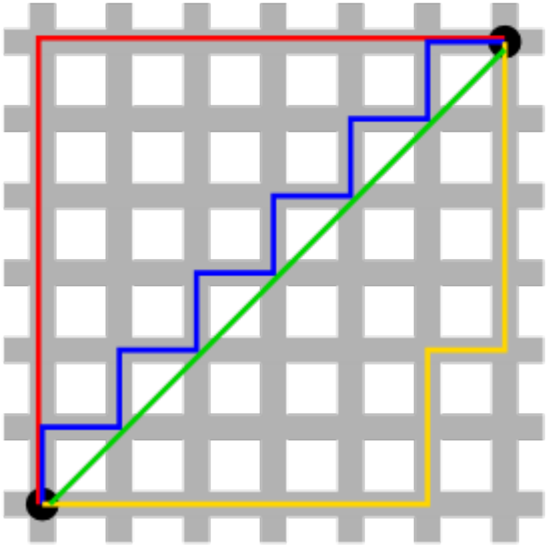


노름 *Norm*

유클리드 노름 *L2 norm*

원점에서 벡터에 연결된 **직선거리**

좌표평면에서의 **점과 점 사이의 거리**



$$L_2 = \sqrt{|v_1|^2 + |v_2|^2 + \cdots + |v_n|^2}$$

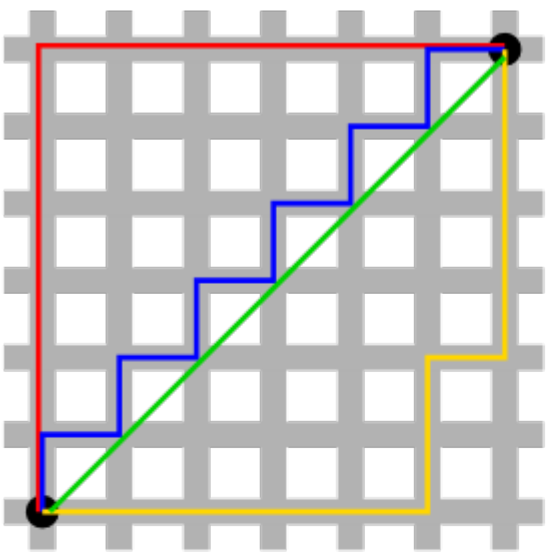
그림에서 초록색 선으로 표현가능

노름 *Norm*

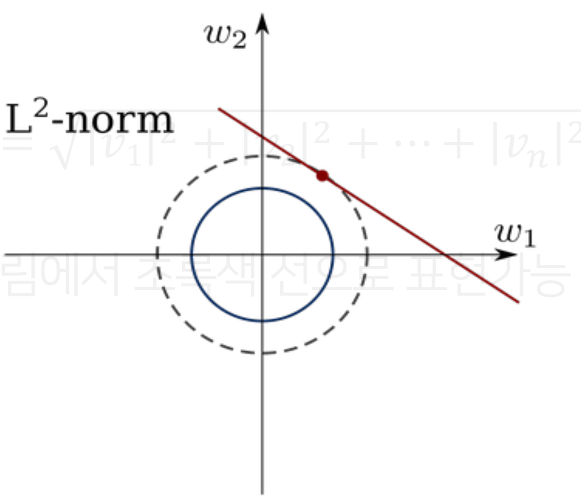
유클리드 노름 *L2 norm*

원점에서 벡터에 연결된 **직선거리**

좌표평면에서의 **점과 점 사이의 거리**



Ridge regression에서 β의 제약조건으로 쓰임



$L_2 = \sqrt{|v_1|^2 + |v_2|^2 + \cdots + |v_n|^2}$

그림에서 초록색 선으로 표현가능

노름 *Norm*

유클리드 노름 *L2 norm*

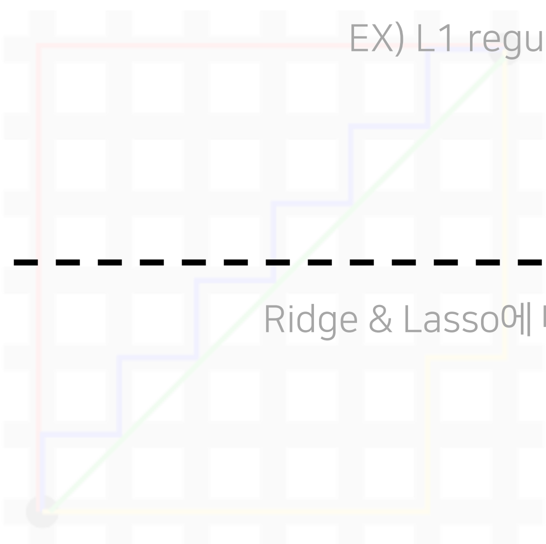
원점에서 벡터에 연결된 **직선거리**

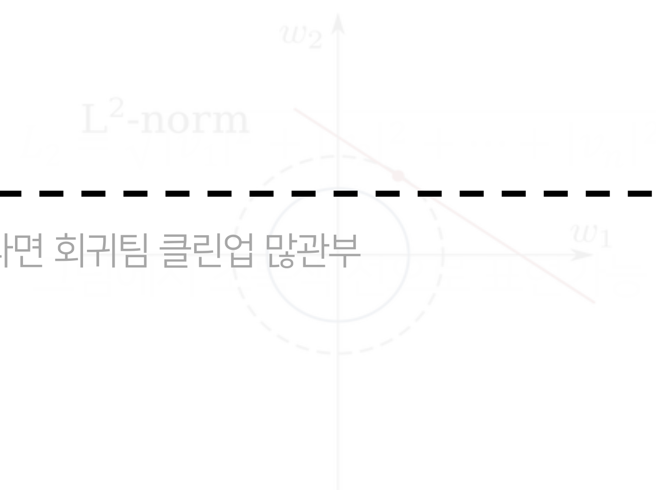
좌표평면에서의 점과 점 사이의 거리

L1, L2 Norm은 주로 머신러닝에서 제약조건을 설정할 때 많이 사용

EX) L1 regularization, L2 regularization

Ridge regression에서 β의 제약조건으로 쓰임





Ridge & Lasso에 대해 궁금하다면 회귀팀 클린업 많관부

내적 *Inner product*

내적 *Inner product*

벡터를 곱하는 방법

벡터의 내적 결과값은 **스칼라**

내적의 특징과 식

$$when\ \boldsymbol{x} = 0\ or\ \boldsymbol{y} = 0, \boldsymbol{x} \cdot \boldsymbol{y} = 0$$

$$\boldsymbol{x} \cdot \boldsymbol{y} = \|\boldsymbol{x}\| \times \|\boldsymbol{y}\| \times cos\theta$$

$$\boldsymbol{x} \cdot \boldsymbol{x} = \|\boldsymbol{x}\| \times \|\boldsymbol{x}\| \times \cos 0 = \|\boldsymbol{x}\|^2$$

$$when\ \boldsymbol{x} = (x_1, x_2, \dots, x_n), \|\boldsymbol{x}\|^2 = \sum_{i=1}^{n} x_i^2$$

내적 *Inner product*

내적 *Inner product*

벡터를 곱하는 방법

벡터의 내적 결과값은 **스칼라**

내적의 특징과 식

① $when\ \boldsymbol{x} = 0\ or\ \boldsymbol{y} = 0, \boldsymbol{x} \cdot \boldsymbol{y} = 0$

② $\boldsymbol{x} \cdot \boldsymbol{y} = \|\boldsymbol{x}\| \times \|\boldsymbol{y}\| \times cos\theta$

③ $\boldsymbol{x} \cdot \boldsymbol{x} = \|\boldsymbol{x}\| \times \|\boldsymbol{x}\| \times \cos 0 = \|\boldsymbol{x}\|^2$

④ $when\ \boldsymbol{x} = (x_1, x_2, \ldots, x_n), \|\boldsymbol{x}\|^2 = \sum_{i=1}^{n} x_i^2$

내적 *Inner product*

내적의 기하학적 의미

길이 × 길이 × 두 벡터가 이루는 코사인 사잇각

$$\boldsymbol{x} \cdot \boldsymbol{y} = \|\boldsymbol{x}\| \times \|\boldsymbol{y}\| \times cos\theta$$

$|\boldsymbol{x}|cos\theta$는 벡터$\boldsymbol{y}$와 평행하고 벡터$\boldsymbol{x}$에서 벡터$\boldsymbol{y}$로 수직을 이루는 삼각형의 밑변

↓

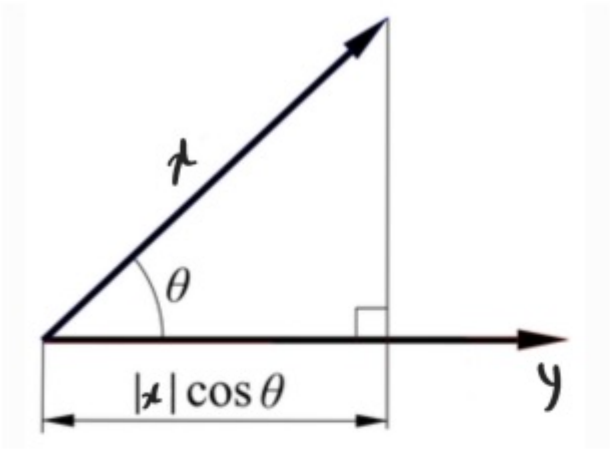
해당 밑변의 길이에 $|\boldsymbol{y}|$를 곱하면 **내적**

내적 *Inner product*

내적의 기하학적 의미

길이 × 길이 × 두 벡터가 이루는 코사인 사잇각

$$\boldsymbol{x} \cdot \boldsymbol{y} = \|\boldsymbol{x}\| \times \|\boldsymbol{y}\| \times cos\theta$$



$|\boldsymbol{x}|cos\theta$는 벡터 $\boldsymbol{x}$에서

벡터 $\boldsymbol{y}$로 수직을 이루는 삼각형의 밑변의 길이

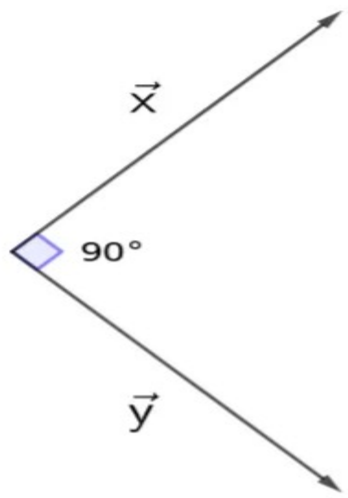
↓

해당 밑변의 길이에 $|\boldsymbol{y}|$를 곱하면 **내적**

직교성 *Orthogonality*

직교성 *Orthogonality*

두 벡터가 **수직**일때, 두 벡터가 **직교**한다고 표현



두 벡터가 수직이면 사잇값 θ는 $\pi/2$

$$\boldsymbol{x} \cdot \boldsymbol{y} = \|\boldsymbol{x}\| \times \|\boldsymbol{y}\| \times \cos\left(\frac{\pi}{2}\right) = 0$$

두 벡터가 직교하면 $\boldsymbol{x} \cdot \boldsymbol{y}=0$

직교성 *Orthogonality*

직교성 *Orthogonality*

두 벡터가 **수직**일때, 두 벡터가 **직교**한다고 표현

임의의 벡터 $\boldsymbol{u}$와 영벡터는 직교할까?

두 벡터를 내적했을 때 내적값이 0이면 두 벡터는 **직교**

어떤 벡터든 영벡터와 내적하면 0이 됨

⋮

임의의 벡터 $\boldsymbol{u}$와 영벡터는 항상 **직교**

직교성 *Orthogonality*

정규직교성 *Orthonormality*

두 벡터의 **노름**이 모두 1이면서 서로 **직교**하는 상황을 의미

노름이 1이기 때문에 오로지 **방향 성분**만을 나타냄

정규직교 조건

두 벡터 $\boldsymbol{x}, \boldsymbol{y}$가 orthonormal

$\vdots$

① $\boldsymbol{x} \cdot \boldsymbol{y} = 0$

② $\|\boldsymbol{x}\| = \|\boldsymbol{y}\| = 1$

직교성 *Orthogonality*

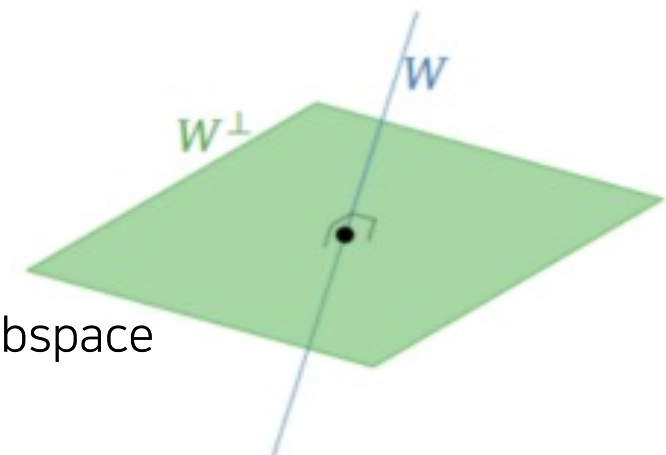
공간의 직교성 *Orthogonality of space*

어떤 공간 S와 부분공간 T가 **직교**

$\leftrightarrow$

공간 S 안에 있는 모든 벡터가 부분공간 T 안에 있는 **모든** 벡터와 **직교**

W는 1차원 Subspace



W^{T}는 2차원 Subspace

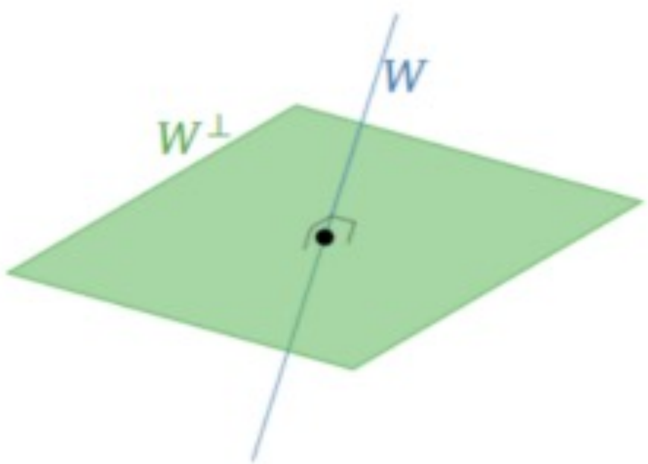
직교성 *Orthogonality*

공간의 직교성 *Orthogonality of space*

어떤 공간 S와 부분공간 T가 **직교**

↔

공간 S 안에 있는 모든 벡터가 부분공간 T 안에 있는 **모든** 벡터와 **직교**



두 subspace가 **직교**

⋮

초록 평면에 존재하는 모든 벡터가

파란색 선 내의 벡터와 **직교**

직교성 *Orthogonality*

직교행렬 *Orthogonal Matrix*

행벡터와 열벡터가 유클리드 공간의 정규 직교 기저를 이루는 실수 행렬

$$Q^{-1} = Q^T \; or \; QQ^T = Q^T Q = I$$

직교행렬의 성질

① 직교행렬의 전치행렬은 직교행렬

② 직교행렬의 곱은 직교행렬

③ 만약 Q가 직교행렬이라면 Q의 행렬식은 1 또는 -1

(공간의 크기가 유지)

④ 직교행렬 Q는 회전을 하는 선형변환

직교성 *Orthogonality*

직교행렬 *Orthogonal Matrix*

행벡터와 열벡터가 유클리드 공간의 정규 직교 기저를 이루는 실수 행렬

$$Q^{-1} = Q^T \text{ or } QQ^T = Q^TQ = I$$

직교행렬의 성질

① 직교행렬의 전치행렬은 직교행렬

② 직교행렬의 곱은 직교행렬

③ 만약 Q가 직교행렬이라면 Q의 행렬식은 1 또는 -1

(공간의 크기가 유지)

④ 직교행렬 Q는 **회전**을 하는 **선형변환**

2

정사영과 회귀

정사영과 회귀

$$Let\ X = [\mathbf{x_1} \quad \mathbf{x_2} \quad \cdots \quad \mathbf{x_p}]$$

$$then\ X\beta = [\mathbf{x_1} \quad \mathbf{x_2} \quad \cdots \quad \mathbf{x_p}] \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} = \boldsymbol{y}$$

$Ax = b$ 문제는 **회귀문제** $X\beta = y$ 로 생각 가능

행렬 A : 독립변수들의 집합 X

벡터 b : 종속변수 y

해 x : 회귀계수 β

정사영과 회귀

X의 각 열은 각각의 변수를 의미하므로

$X\beta$는 열벡터의 선형결합으로 y**를 표현 가능케 하는** β를 찾는 문제

1주차 Remark 행렬에 벡터를 곱한 것은 열벡터의 선형결합

X의 열공간 내에 $\boldsymbol{y}$가 존재한다

$\Leftrightarrow$ X의 열벡터의 선형결합으로 **y를 완벽하게 재현**할 수 있다

정사영과 회귀

1주차 Remark

행렬에 벡터를 곱한 것은 열벡터의 선형결합

그러나, 현실에서는 X로 y를 예측하고자 할 때 **오차** 발생

이때 **오차를 최소화**하는 방향으로 분석 진행

X의 각 열은 각각의 변수를 의미하므로

$X\beta$는 열벡터의 선형결합으로 **y를 표현 가능케 하는** β를 찾는 문제

정사영을 이용!

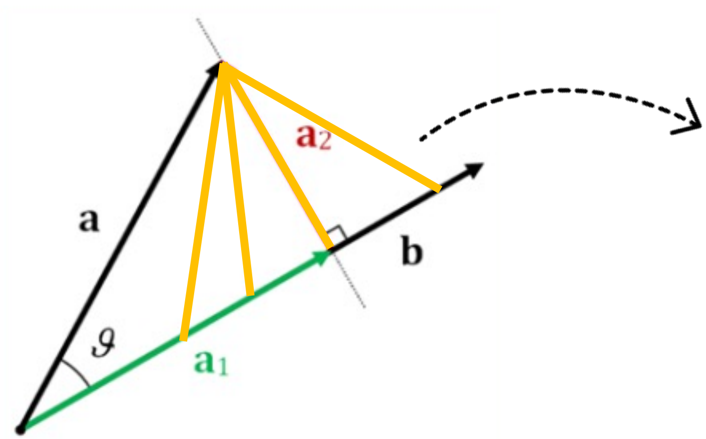
↔ X의 열벡터의 선형결합으로 **y를 완벽하게 재현**할 수 있다

정사영 *Orthogonal Projection*

투영 *Projection*

어떤 벡터를 다른 벡터로 옮겨서 표현하는 것이며,

벡터 b의 공간으로 여분의 공간을 압축시키는 **선형변환**의 일종



노란색 방향들로 모두

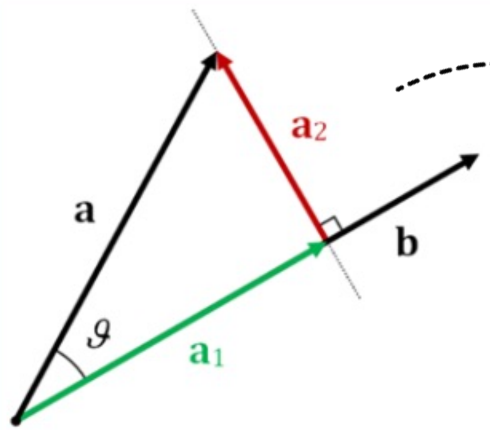
Projection 가능

정사영 *Orthogonal Projection*

정사영 *Orthogonal Projection*

벡터를 다른 벡터로 투영시킬 때는 여러 각도로 투영 가능

그 중 **정사영**은 **직각**을 이루어 투영시키는 것

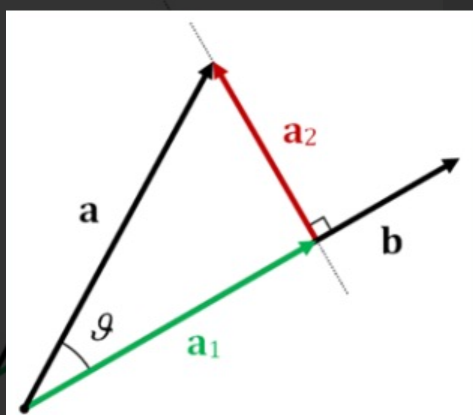


벡터 a_1은 벡터 b에 대해

Orthogonal Projection

정사영 *Orthogonal Projection*

왜 직각인가?



벡터 a를 투영시키면 원래 벡터와의 차이 a_2 발생

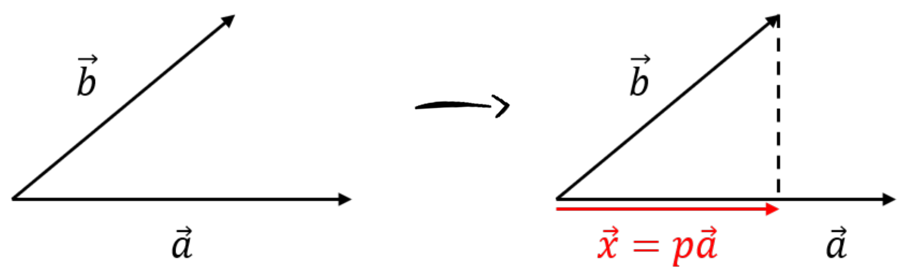
실제 값과 투영 이후의 차이를 **오차**라고 하면

수직으로 내리는 **정사영 벡터**가 **오차를 최소**가 되게 함

정사영 *Orthogonal Projection*

Vector Projection 구하기

b를 a에 정사영시키는 문제를 통해 Vector Projection 계산 방법을 알아보자



정사영된 벡터 $\mathbf{x} = p\mathbf{a}$에 대해, 편차벡터 $\mathbf{b} - p\mathbf{a}$는 벡터 $\mathbf{a}$와 수직이므로 **내적이 0**

이를 이용하여 벡터 $\mathbf{x}$를 구할 수 있음

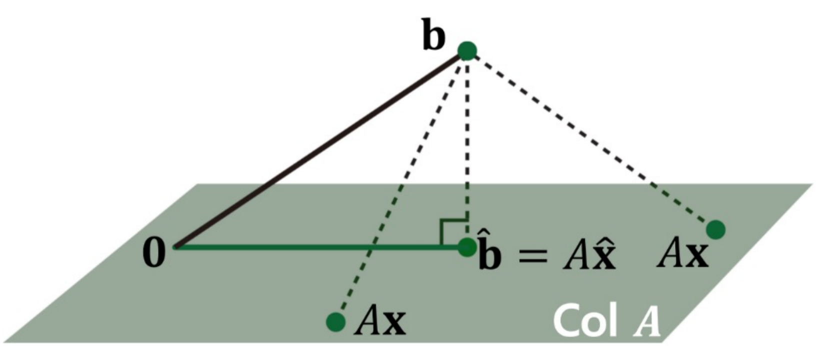
$$\mathbf{b}^T\mathbf{a} - p\mathbf{a}^T\mathbf{a} = 0$$

$$p = \frac{\mathbf{b}^T\mathbf{a}}{\mathbf{a}^T\mathbf{a}} \qquad \therefore \mathbf{x} = p\mathbf{a} = \frac{\mathbf{b}^T\mathbf{a}}{\mathbf{a}^T\mathbf{a}}\mathbf{a}$$

정사영 *Orthogonal Projection*

정사영: 벡터 → 공간

정사영의 목적은 **높은 차원의 부분공간에 속한 벡터**를 저차원으로 낮추는 것으로
3차원 공간에 있는 벡터를 2차원 벡터로 바꾸어 표현하는 방식



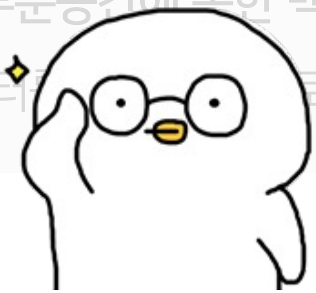
$col(A)$는 A의 열 공간으로,

오차 $\mathbf{b} - \hat{\mathbf{b}} = \mathbf{b} - X\hat{\beta}$ 가 최소화되도록

근사한 점이 $\hat{\mathbf{b}}$

정사영 *Orthogonal Projection*

정사영: 벡터 → 공간



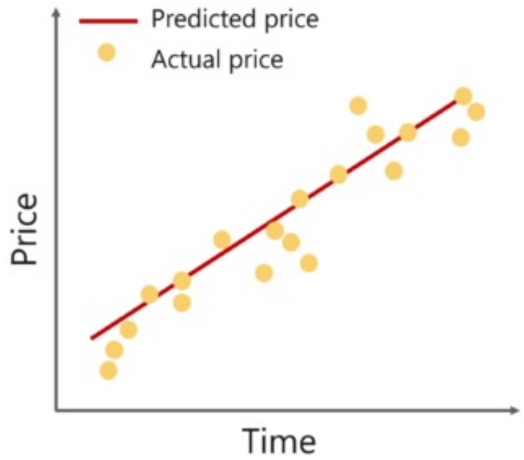
이는 회귀분석의 **최소제곱법** *(Least Square Method)* 에서 사용

회귀분석의 관점에서 정사영을 이해해보자!

최소제곱법과 정사영

모든 데이터를 통과하는 직선을 구하는 것이 이상적인 목표

= **연립방정식의 해**를 구하는 것이 **선형대수**의 목표



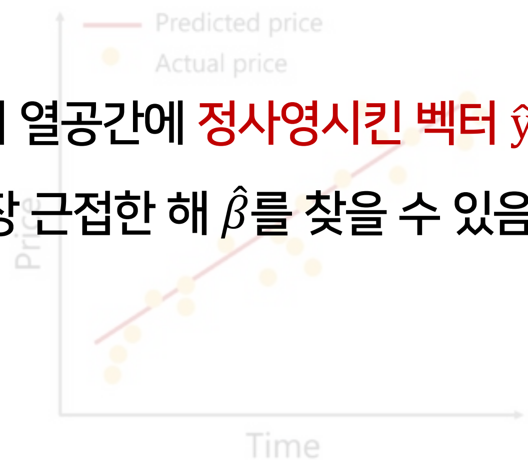
현실에서 **모든 점을 통과하는 직선은 불가능**

이는 $X\beta = y$에서 **벡터 y 가 행렬 X의 열공간에 존재하지 않음**을 의미

최소제곱법과 정사영

모든 데이터를 통과하는 직선을 구하는 것이 이상적인 목표

= 연립방정식의 해를 구하는 것이 선형대수의 목표



y벡터를 X의 열공간에 정사영시킨 벡터 $\hat{y}$를 구하면

가장 근접한 해 $\hat{\beta}$를 찾을 수 있음!

현실에서 모든 점을 통과하는 직선은 불가능

이는 $X\beta = y$에서 벡터 y 가 행렬 X의 열공간에 존재하지 않음을 의미

최소제곱법과 정사영

모든 데이터를 통과하는 직선을 구하는 것이 이상적인 목표

= 연립방정식의 해를 구하는 것이 선형대수의 목표

정리하자면

$\mathbf{y}$벡터를 X의 열공간에 정사영시킨 벡터 $\hat{\mathbf{y}}$를 구하면

가장 근접한 해 $\hat{\beta}$를 찾을 수 있음!

☝ 설명 불가능한 벡터 $\mathbf{y}$를 X의 열공간에 **정사영**시키기

✌ 실제 해 β가 아닌 **추정치 $\hat{\beta}$**를 구하기

현실에서 모든 점을 통과하는 직선은 불가능

이는 $X\beta = y$에서 벡터 y 가 행렬 X의 열공간에 존재하지 않음을 의미

최소제곱법과 정사영

편차벡터 $e = y - \hat{y} = y - X\hat{\beta}$

편차벡터와 X의 열공간은 수직이므로 다음 식이 성립

$$X^T\left(\mathbf{y} - X\hat{\beta}\right) = 0$$

$$X^T\mathbf{y} = X^T X\hat{\beta}$$

$$(X^T X)^{-1}\, X^T\mathbf{y} = \hat{\beta}$$

$$\therefore X(X^T X)^{-1}\, X^T\mathbf{y} = X\hat{\beta} = \hat{\mathbf{y}}$$

최소제곱법과 정사영

편차벡터 $e = y - \hat{y} = y - X\hat{\beta}$

편차벡터와 X의 열공간은 수직이므로 다음 식이 성립

$$X^T\left(\mathbf{y} - X\hat{\beta}\right) = 0$$

$$X^T\mathbf{y} = X^T X\hat{\beta}$$

$$(X^T X)^{-1} X^T\mathbf{y} = \hat{\beta}$$

$$\therefore X(X^T X)^{-1} X^T\mathbf{y} = X\hat{\beta} = \hat{\mathbf{y}}$$

정사영행렬 *projection matrix* 또는 모자행렬 *hat matrix*

최소제곱법과 정사영

미분을 통한 최소제곱법 풀이

1주차에 배웠던 **행렬 미분**을 통해서도 최소제곱법 해결 가능!

오차를 최소화하는 β를 찾기 위해 식을 다음과 같이 전개

$$\hat{\beta} = \arg\min_{\beta} \|\mathbf{y} - X\beta\| = \arg\min_{\beta} \|\mathbf{y} - X\beta\|^2 = \arg\min_{\beta} (\mathbf{y} - X\beta)^T(\mathbf{y} - X\beta)$$

$$= \arg\min_{\beta} \mathbf{y}^{\mathbf{T}}\mathbf{y} - \mathbf{y}^T X\beta - \beta^T X^T \mathbf{y} + \beta^T X^T \mathrm{X}\beta$$

β로 미분한 값이 0이 되는 β를 찾아보자!

최소제곱법과 정사영

미분을 통한 최소제곱법 풀이

미분대상 $y^T y - y^T X\beta - \beta^T X^T y + \beta^T X^T X\beta$이 스칼라이므로

스칼라를 벡터로 미분하는 경우에 해당

$$-X^T y - X^T y + X^T X\beta + X^T X\beta = 2X^T X\beta - 2X^T y = 0$$

$$\Rightarrow X^T X\beta = X^T y$$

$$\Leftrightarrow \beta = X(X^T X)^{-1} X^T \mathbf{y}$$

따라서 오차의 길이가 최소가 되는 $\hat{\beta}$ 역시 $X(X^T X)^{-1} X^T \mathbf{y}$가 됨

3

평균, 분산, 상관계수의 기하학적 해석

표본평균의 기하학적 해석

표본평균 유도 과정

① $n\times1$인 벡터 $\mathbf{1}_n^T = \mathbf{1}^T = [1,\cdots,1]$을 정의하고

크기를 1로 만들어 단위벡터 $(\frac{1}{\sqrt{n}})\mathbf{1}$ 정의

② i번째 변수인 열벡터 $\mathbf{x}_i^T = [x_{1i}, x_{2i}\cdots, x_{ni}]$ 생각

③ 단위벡터 $(\frac{1}{\sqrt{n}})\mathbf{1}$에 $\mathbf{x}_i$를 투영시키기

$$\mathbf{x}_i^{\mathrm{T}}\left(\frac{1}{\sqrt{n}}\mathbf{1}\right)\frac{1}{\sqrt{n}}\mathbf{1} = \frac{x_{1i}+x_{2i}+\cdots+x_{ni}}{n}\mathbf{1} = \overline{x}_i\mathbf{1}$$

표본평균벡터

결론 $\mathbf{x}_i$변수의 표본평균은 해당 열벡터를 단위벡터에 투영시킨 결과

표본분산의 기하학적 해석

편차 벡터 *Deviation Vector, Mean centered vector*

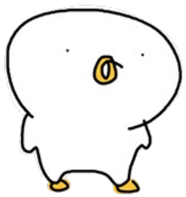
편차 벡터 $\mathbf{d}_i$는 다음과 같으며,

$\mathbf{d}_i$의 각 원소는 각 관찰값의 i번째 변수들의 편차와 같음

$$\mathbf{d}_i = \mathbf{x}_i - \overline{x}_i\mathbf{1} = \begin{bmatrix} x_{1i} - \overline{x}_i \\ x_{2i} - \overline{x}_i \\ \vdots \\ x_{ni} - \overline{x}_i \end{bmatrix}$$

표본분산의 기하학적 해석

표본분산과 편차 벡터의 관계



편차 벡터의 길이를 제곱(내적)한 값을 살펴보면

$$L^2_{d_i} = d_i^T d_i = \sum_{j=1}^{n} \left(x_{ji} - \overline{x_i}\right)^2 = S_{x_i x_i}$$

편차 벡터의 길이 제곱은 **편차 제곱의 합**과 동일하며,

이를 $n-1$로 나누어 주면 **i번째 변수의 표본 분산**이 됨

따라서, 표본분산은 편차 벡터 길이 제곱에 **비례**

상관계수의 기하학적 해석

두 벡터 $\mathbf{x} = [x_1, x_2 \cdots, x_n]^T, \mathbf{y} = [y_1, y_2 \cdots, y_n]^T$를 생각해보자!

두 편차 벡터 d_x와 d_y에 대해 다음 식 성립

$$d_x^T d_y = \sum_{j=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) = S_{xy}$$

내적의 정의에 의해 $d_x^T d_y = L_{d_x} L_{d_y} cos\theta$

θ는 $\mathbf{x}$와 $\mathbf{y}$의 편차 벡터가 이루는 각도

$$S_{xy} = d_x^T d_y = \sqrt{S_{xx}}\sqrt{S_{yy}} cos\theta$$

$$\therefore cos\theta = \frac{S_{xy}}{\sqrt{S_{xx}}\sqrt{S_{yy}}} = r_{xy}$$

결론적으로 두 편차 벡터 사이의 각도가 상관계수를 결정

상관계수의 기하학적 해석

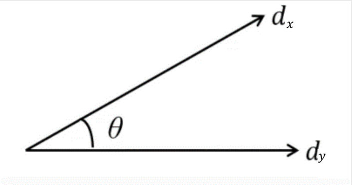
두 벡터 $\mathbf{x} = [x_1, x_2 \cdots, x_n]^T, \mathbf{y} = [y_1, y_2 \cdots, y_n]^T$를 생각해보자!

두 편차 벡터 d_x와 d_x에 대해 다음 식 성립

$$d_x^T d_y = \sum_{j=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) = S_{xy}$$

내적의 정의에 의해 $d_x^T d_y = L_{d_x} L_{d_y} cos\theta$

θ는 $\mathbf{x}$와 $\mathbf{y}$의 편차 벡터가 이루는 각도



$$S_{xy} = d_x^T d_y = \sqrt{S_{xx}}\sqrt{S_{yy}} cos\theta$$

$$\therefore cos\theta = \frac{S_{xy}}{\sqrt{S_{xx}}\sqrt{S_{yy}}} = r_{xy}$$

결론적으로 **두 편차 벡터 사이의 각도**가 상관계수를 결정

상관계수의 기하학적 해석

두 벡터 $\mathbf{x} = [x_1, x_2 \cdots, x_n]^T, \mathbf{y} = [y_1, y_2 \cdots, y_n]^T$를 생각해보자!

두 편차 벡터 d_x와 d_x에 대해 다음 식 성립

Remark

$$L_{d_i}^2 = d_i^T d_i = \sum_{j=1}^{n} (x_{ji} - \overline{x_i})^2 = S_{x_i x_i}$$

$$\Rightarrow L_{d_x} = \sqrt{S_{x_i x_i}}$$

내적의

θ는 $\mathbf{x}$와 $\mathbf{y}$의 편차 벡터가 이루는 각도

$$S_{xy} = d_x^T d_y = \sqrt{S_{xx}}\,\sqrt{S_{yy}} cos\theta$$

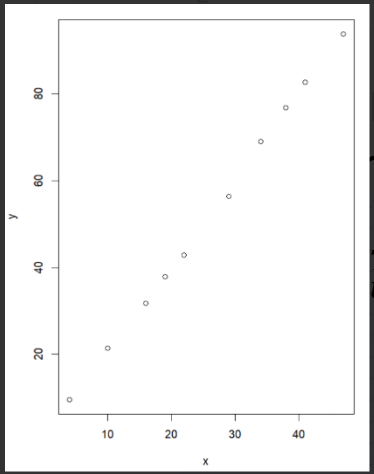
$$\therefore cos\theta = \frac{S_{xy}}{\sqrt{S_{xx}}\,\sqrt{S_{yy}}} = r_{xy}$$

결론적으로 **두 편차 벡터 사이의 각도**가 상관계수를 결정

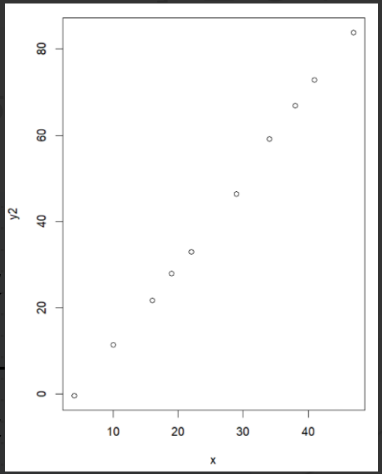
편차 벡터 사이의 각도는 어떤 의미인가?

상관계수의 기하학적 해석

두 벡터 $\mathbf{x} = [x_1, x_2 \cdots, x_n]^T, \mathbf{y} = [y_1, y_2 \cdots, y_n]^T$를 생각해보자!



상관계수가 높은 x, y의 plot



y 전체에 -10

편차에는 영향을 주지 않고 평균에만 영향을 주며 선형관계는 동일

결국 편차 벡터의 방향이 유사하다는 것은 함께 움직이는 정도가 크다는 의미

"상관계수에 영향을 주는 것은 오직 편차가 함께 움직이는 정도"

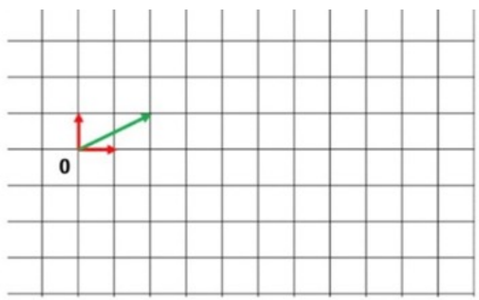
4

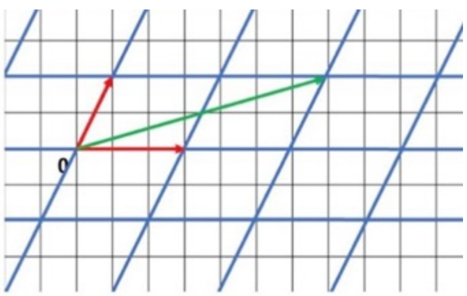
고윳값과 고유벡터

고윳값과 고유벡터 *Eigen value & Eigen vector*

고유벡터

선형변환에 관계없이 공간을 유지하는 벡터





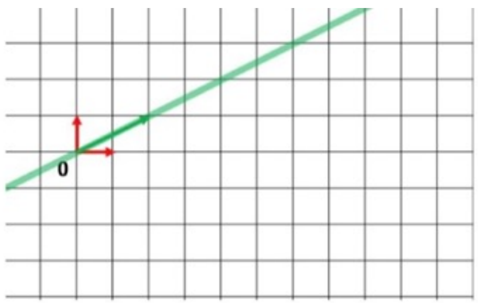
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$의 열공간에서 $\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}$의 열공간으로 기저가 변환

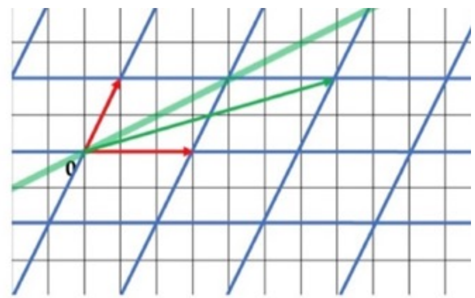
초록색 벡터는 $\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 7 \\ 2 \end{bmatrix}$ 로 변환

고윳값과 고유벡터 *Eigen value & Eigen vector*

고유벡터

선형변환에 관계없이 공간을 유지하는 벡터

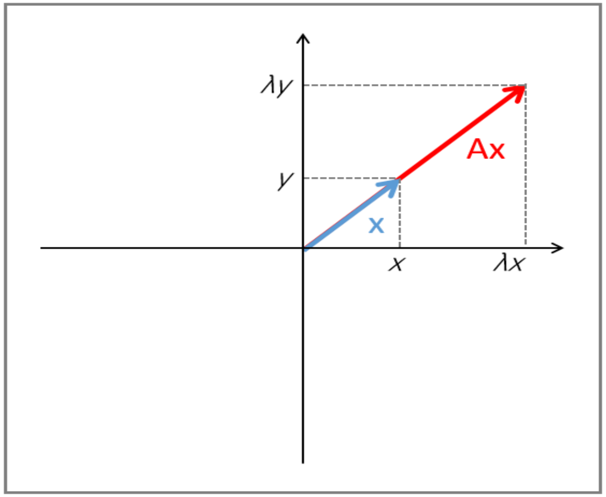




기존에 초록색 벡터가 존재하던 **열공간은 유지되지 않음**

하지만 고유벡터는 **변환 후에도 자신의 공간에 남아있음**

고윳값과 고유벡터 *Eigen value & Eigen vector*



고유벡터 $\boldsymbol{x}$는 A라는 선형변환을 가했음에도

벡터공간은 유지한 채 **길이만 바뀜**

이때 길이가 변하는 정도 : **고윳값** λ

고윳값과 고유벡터 *Eigen value & Eigen vector*

고유벡터는 선형변환 시 **방향은 유지**한 채로
길이만 변하는, 영벡터가 아닌 벡터

고윳값은 고유벡터의 **길이가 변하는 정도**

고유벡터 x는 A라는 선형변환을 가했음에도
벡터공간은 유지한 채 **길이만 바뀜**
이때 길이가 변하는 정도 : **고윳값** λ

고윳값과 고유벡터 *Eigen value & Eigen vector*

고윳값, 고유벡터 계산

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

$$\vdots$$

$\boldsymbol{v}$: 행렬 A의 **고유벡터**

λ : 행렬 A의 **고윳값**

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

$$\Leftrightarrow (A - \lambda I)\boldsymbol{v} = 0$$

행렬 $(A - \lambda I)$의 **역행렬 존재 여부**를 통해 고윳값과 고유벡터 계산!

고윳값과 고유벡터 *Eigen value & Eigen vector*

고윳값, 고유벡터 계산

$(A-\lambda I)$의 **역행렬이 존재**한다면?

$$\vdots$$

$$(A-\lambda I)^{-1}(A-\lambda I)\boldsymbol{v}=0$$

$$\Rightarrow \boldsymbol{v}=0$$

0은 어떤 변환을 가해도 0이기 때문에 **고유벡터가 될 수 없음**

$$\vdots$$

고유벡터가 존재한다면 $(A-\lambda I)$의 **역행렬은 존재하지 않아야 함**

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

고윳값과 고유벡터 *Eigen value & Eigen vector*

고윳값, 고유벡터 계산

$(A-\lambda I)$의 **역행렬이 존재**한다면?

$$\vdots$$

$$(A-\lambda I)^{-1}(A-\lambda I)\boldsymbol{v}=0$$

$$\Rightarrow \boldsymbol{v}=0$$

0은 어떤 변환을 가해도 0이기 때문에 **고유벡터가 될 수 없음**

고유벡터가 존재한다면 $(A-\lambda I)$의 **역행렬은 존재하지 않아야 함**

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

고윳값과 고유벡터 *Eigen value & Eigen vector*

고윳값 계산 예시

행렬식=0을 통해 **고윳값** 계산

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

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

$$= (2-\lambda)^2 - 1 = (\lambda - 1)(\lambda - 3) = 0$$

$$\lambda = 1 \; or \; 3$$

고윳값과 고유벡터 *Eigen value & Eigen vector*

고유벡터 계산 예시

구한 λ를 대입하여 **고유벡터** 계산

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

$$\text{if } \lambda = 1,$$

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 1 \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \longrightarrow \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\text{if } \lambda = 3,$$

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 3 \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \longrightarrow \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

고유공간 *Eigen space*

고유공간 *Eigen space*

각각의 고유벡터가 존재하는 공간

고유벡터 $\boldsymbol{v}$를 span하면 생김!

고유공간 내의 벡터는 모두 고유벡터

주로 사용하는 고유벡터 $\boldsymbol{v}$는 **정규화로 길이를 1로 맞춤**

고윳값 분해 *Eigen Decomposition*

대각화 *Diagonalization*

$$D = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix}$$

대각행렬은 행렬식, 거듭제곱, 역행렬 계산 등에서 굉장히 편리

행렬 $A \in \mathbb{R}^{n \times n}$이 대각행렬 $D \in \mathbb{R}^{n \times n}$로 다음과 같이 대각화 가능하다면

행렬 A의 행렬식, 거듭제곱 등을 **쉽게 계산 가능**

$$D = P^{-1}AP, P \in \mathbb{R}^{n \times n}$$

고윳값 분해 *Eigen Decomposition*

대각화 *Diagonalization*

$$D = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix}$$

대각행렬은 행렬식, 거듭제곱, 역행렬 계산 등에서 굉장히 **편리**

행렬 $A \in \mathbb{R}^{n \times n}$이 대각행렬 $D \in \mathbb{R}^{n \times n}$로 다음과 같이 대각화 가능하다면

행렬 A의 행렬식, 거듭제곱 등을 **쉽게 계산 가능**

$$D = P^{-1}AP, P \in \mathbb{R}^{n \times n}$$

고윳값 분해 *Eigen Decomposition*

행렬 A가 $n \times n$ 정방행렬이라고 한다면,

일반적으로 n**개의 고유벡터**와 n**개의 고윳값** 가짐

→ 각각 고유벡터 $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n$이 $A\boldsymbol{v} = \lambda\boldsymbol{v}$을 만족!

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

$$\Leftrightarrow A[\boldsymbol{v}_1 \quad \boldsymbol{v}_2 \quad \cdots \quad \boldsymbol{v}_n] = [\lambda_1\boldsymbol{v}_1 \quad \lambda_2\boldsymbol{v}_2 \quad \cdots \quad \lambda_n\boldsymbol{v}_n]$$

$$= [\boldsymbol{v}_1 \quad \boldsymbol{v}_2 \quad \cdots \quad \boldsymbol{v}_n]\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

고윳값 분해 *Eigen Decomposition*

양변에 고유벡터 행렬의 역행렬을 곱하면

$$A = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{bmatrix}^{-1}$$

$$\Leftrightarrow \mathrm{A} = \mathrm{V}\Lambda V^{-1}$$

행렬 A를 **세 행렬의 곱으로 분해**하는 것이 바로 **고윳값 분해**

V : 고유벡터들을 열벡터로 하는 행렬

Λ : 고윳값들을 대각원소로 하는 행렬

고윳값 분해 *Eigen Decomposition*

고윳값 분해를 통한 계산적 이점

행렬식

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

A의 거듭제곱

$$A^k = (V\Lambda V^{-1})^k$$
$$= (V\Lambda V^{-1})\ (V\Lambda V^{-1})\cdots(V\Lambda V^{-1})$$
$$= V\Lambda^k V^{-1} = Vdiag({\lambda_1}^k, \cdots {\lambda_n}^k)V^{-1}$$

역행렬

$$A^{-1} = (V\Lambda V^{-1})^{-1}$$
$$= (V\Lambda^{-1}V^{-1})$$
$$= Vdiag(1/\lambda_1, \cdots, 1/\lambda_n)V^{-1}$$

대각합

$$tr(A) = tr(V\Lambda V^{-1})$$
$$= tr(V^{-1}V\Lambda) = tr(\Lambda) = \sum_{i=1}^{n}\lambda_i$$

고윳값 분해 *Eigen Decomposition*

고윳값 분해를 통한 계산적 이점

행렬식

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

$$= \cancel{\det(V)}\,\det(\Lambda)\,\cancel{\det(V^{-1})}$$

$$= \det(\Lambda) = \lambda_1\lambda_2 \cdots \lambda_n$$

A의 거듭제곱

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

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

$$= V\Lambda^k V^{-1} = Vdiag({\lambda_1}^k, \cdots {\lambda_n}^k)V^{-1}$$

역행렬

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

$$= (V\Lambda^{-1}V^{-1})$$

$$= Vdiag(1/\lambda_1, \cdots, 1/\lambda_n)V^{-1}$$

대각합

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

$$= tr(V^{-1}V\Lambda) = tr(A) = \sum_{i=1}^{n} \lambda_i$$

고윳값 분해 *Eigen Decomposition*

고윳값 분해를 통한 계산적 이점

행렬식

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

A의 거듭제곱

$$A^k = (V\Lambda V^{-1})^k$$
$$= (V\Lambda V^{-1})\ (V\Lambda V^{-1})\cdots(V\Lambda V^{-1})$$
$$= V\Lambda^k V^{-1} = Vdiag({\lambda_1}^k, \cdots {\lambda_n}^k)V^{-1}$$

역행렬

$$A^{-1} = (V\Lambda V^{-1})^{-1}$$
$$= (V\Lambda^{-1}V^{-1})$$
$$= Vdiag(1/\lambda_1, \cdots, 1/\lambda_n)V^{-1}$$

대각합

$$tr(A) = tr(V\Lambda V^{-1})$$
$$= tr(V^{-1}V\Lambda) = tr(A) = \sum_{i=1}^{n}\lambda_i$$

고윳값 분해 *Eigen Decomposition*

고윳값 분해를 통한 계산적 이점

행렬식

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

A의 거듭제곱

$$A^k = (V\Lambda V^{-1})^k$$
$$= (V\Lambda V^{-1})\ (V\Lambda V^{-1})\cdots(V\Lambda V^{-1})$$
$$= V\Lambda^k V^{-1} = V diag({\lambda_1}^k, \cdots {\lambda_n}^k)V^{-1}$$

역행렬

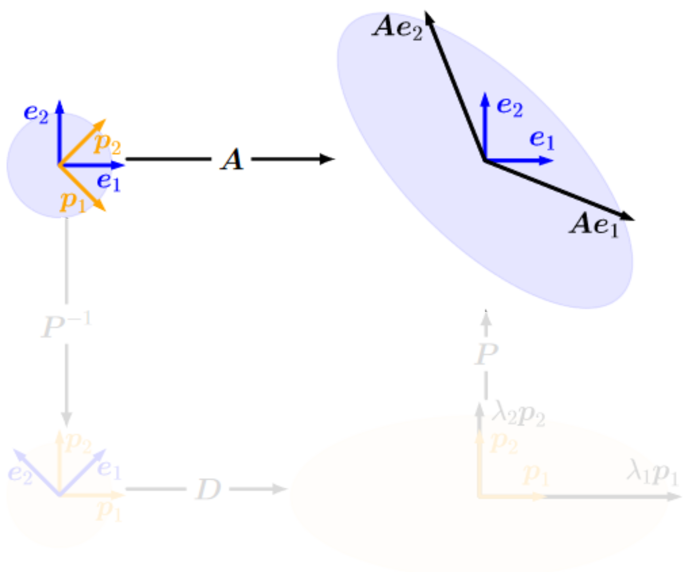
$$A^{-1} = (V\Lambda V^{-1})^{-1}$$
$$= (V\Lambda^{-1}V^{-1})$$
$$= V diag(1/\lambda_1, \cdots, 1/\lambda_n)V^{-1}$$

대각합

$$tr(A) = tr(V\Lambda V^{-1})$$
$$= tr(V^{-1}V\Lambda) = tr(A) = \sum_{i=1}^{n} \lambda_i$$

고윳값 분해 *Eigen Decomposition*

기하학적 의미



- A는 $n \times n$ 선형변환이므로

차원이 유지됨

- A의 **고유벡터**인 p_1, p_2는 변환 이후에도

공간이 유지됨

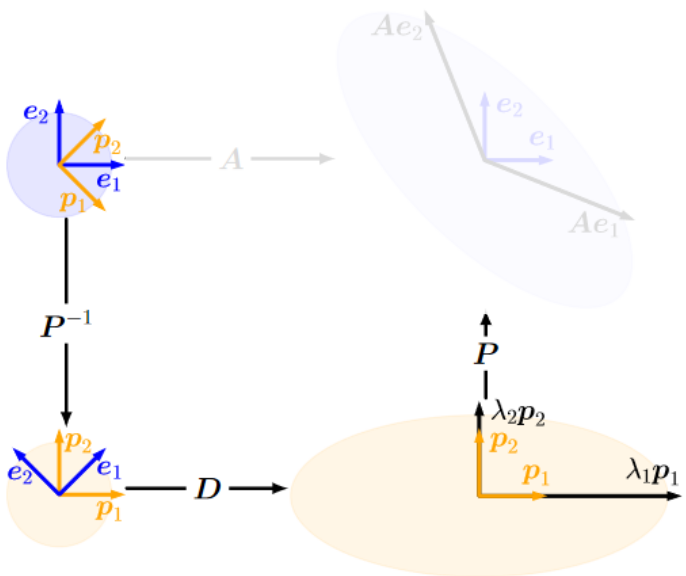
- P와 P^{-1}은 **직교행렬**이므로 공간을 **회전**시킴
- **고윳값** λ는 고유벡터의 방향으로 **스케일링**

- 고윳값 분해란? A 변환을

회전→스케일링→회전으로 분해한 것!

고윳값 분해 *Eigen Decomposition*

기하학적 의미



- A는 $n \times n$ 선형변환이므로

차원이 유지됨

- A의 **고유벡터**인 p_1, p_2는 변환 이후에도

공간이 유지됨

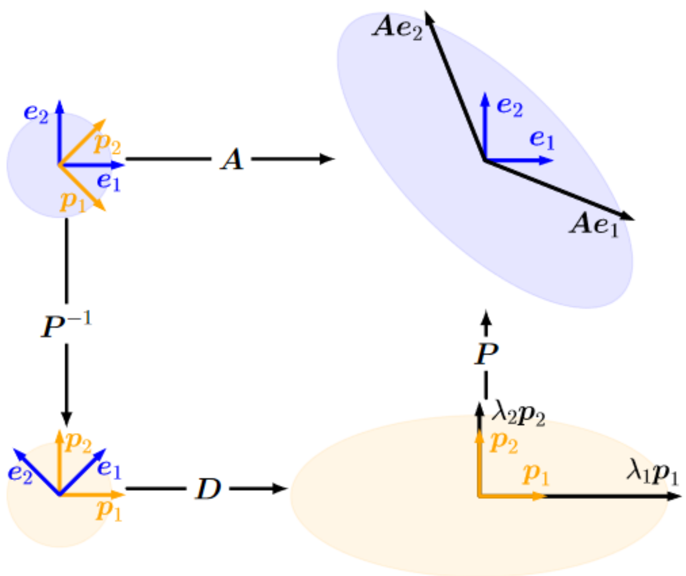
- P와 P^{-1}은 **직교행렬**이므로 공간을 **회전**시킴
- **고윳값** λ만큼 고유벡터의 방향으로 **스케일링**

- 고윳값 분해란? A 변환을

회전→스케일링→회전으로 분해한 것!

고윳값 분해 *Eigen Decomposition*

기하학적 의미



- A는 $n \times n$ 선형변환이므로

  차원이 유지됨

- A의 **고유벡터**인 p_1, p_2는 변환 이후에도

  공간이 유지됨

- P와 P^{-1}은 **직교행렬**이므로 공간을 **회전**시킴
- **고윳값** λ는 고유벡터의 방향으로 **스케일링**

- 고윳값 분해란? A 변환을

  회전→스케일링→회전으로 분해한 것!

고윳값 분해의 조건

A가 (고윳값 분해) 대각화 가능

$V^{-1}AV = \Lambda$인 V가 존재

V가 역행렬이 존재

A가 n개의 선형독립인 고유벡터를 가짐

→ 고유벡터행렬 **V가 선형독립**

대각화 가능한 행렬: Diagonalizable Matrix

대각화 불가능한 행렬: Defective Matrix

고윳값 분해의 조건

A가 (고윳값 분해) 대각화 가능

$V^{-1}AV = \Lambda$인 V가 존재

V가 역행렬이 존재

A가 n개의 선형독립인 고유벡터를 가짐

→ 고유벡터행렬 V가 선형독립

대각화 가능한 행렬: Diagonalizable Matrix

대각화 불가능한 행렬: Defective Matrix

대칭 행렬과 고윳값 분해

실수인 대칭행렬의 성질

① 모든 **고윳값이 실수**

② 항상 **대각화 (고윳값 분해) 가능**

③ 고유벡터들이 **서로 직교**

(Proof ③)

$$A\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i,\ A\boldsymbol{v}_j = \lambda_j \boldsymbol{v}_j\ (i \neq j)$$

$$\boldsymbol{\lambda}_i(\boldsymbol{v}_i \cdot \boldsymbol{v}_j) = \boldsymbol{v}_j^T \boldsymbol{\lambda}_i \boldsymbol{v}_i = \boldsymbol{v}_j^T A \boldsymbol{v}_i = \left(\boldsymbol{v}_j^T A \boldsymbol{v}_i\right)^T$$

$$= \boldsymbol{v}_i^T A \boldsymbol{v}_j = \boldsymbol{v}_i^T \lambda_j \boldsymbol{v}_j = \lambda_j \boldsymbol{v}_i^T \boldsymbol{v}_j = \lambda_j(\boldsymbol{v}_i \cdot \boldsymbol{v}_j)$$

내적이 0이어야 식이 성립!

대칭 행렬과 고윳값 분해

실수 대칭행렬에는 무엇이 있을까?

① 모든 고윳값이 실수

공분산 행렬 $Cov(X,X) = \begin{bmatrix} Var(X_1) & Cov(X_1,X_2) & \cdots & Cov(X_1,X_n) \\ Cov(X_2,X_1) & Var(X_2) & \cdots & Cov(X_2,X_n) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_n,X_1) & Cov(X_n,X_2) & \cdots & Var(X_n) \end{bmatrix}$

(Proof ③)

→ 고윳값 분해를 공분산 행렬에 적용한 것이 PCA

클린업 3주차에 등장 예정!

$$= v_i^T A v_j = v_i^T \lambda_j v_j = \lambda_j v_i^T v_j = \lambda_j (v_i \cdot v_j)$$

5

특잇값 분해(SVD)

특잇값 분해

특잇값 분해 *Singular Value Decomposition*

직교하는 벡터 집합에 대하여 선형변환 후에

크기는 변하지만 **여전히 직교**하는 직교집합을 찾는 것

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

특잇값 분해

특잇값 분해 *Singular Value Decomposition*

특잇값 분해의 아이디어

직교하는 벡터 집합에 대하여 선형변환 후에

크기는 변하지만 여전히 직교하는 직교집합을 찾는 것

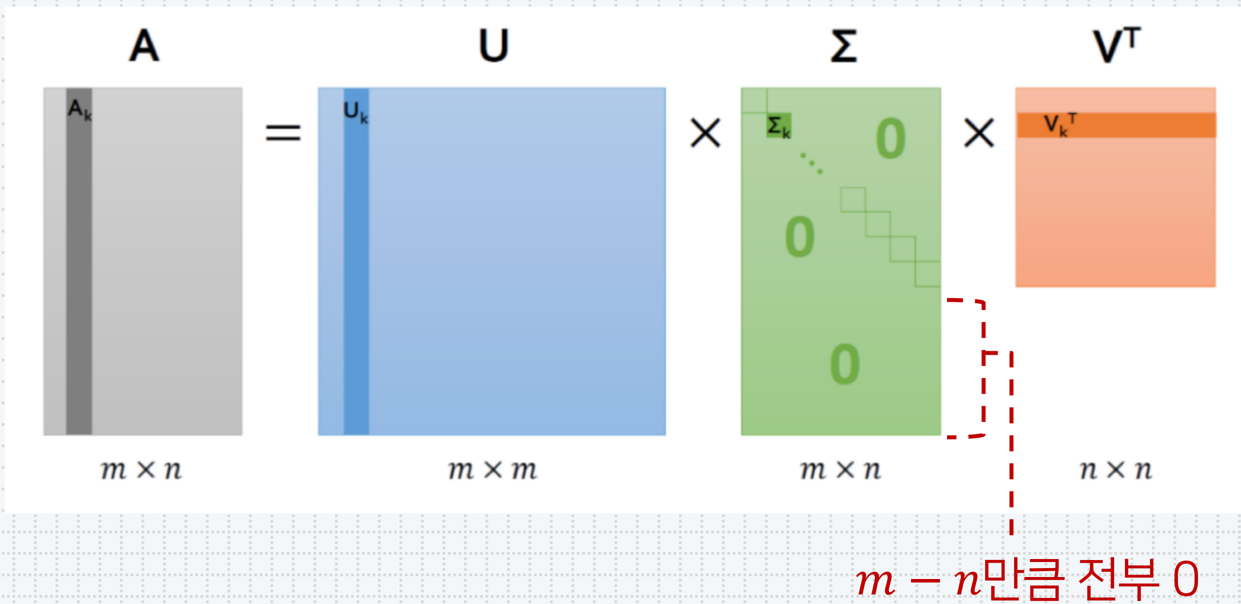
직사각 행렬은 고윳값 분해가 불가능

→ $\mathbf{A}^{\mathbf{T}}\mathbf{A}$, $\mathbf{A}\mathbf{A}^{\mathbf{T}}$는 대칭행렬이 되므로 이를 이용

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

SVD는 고윳값 분해의 일반화

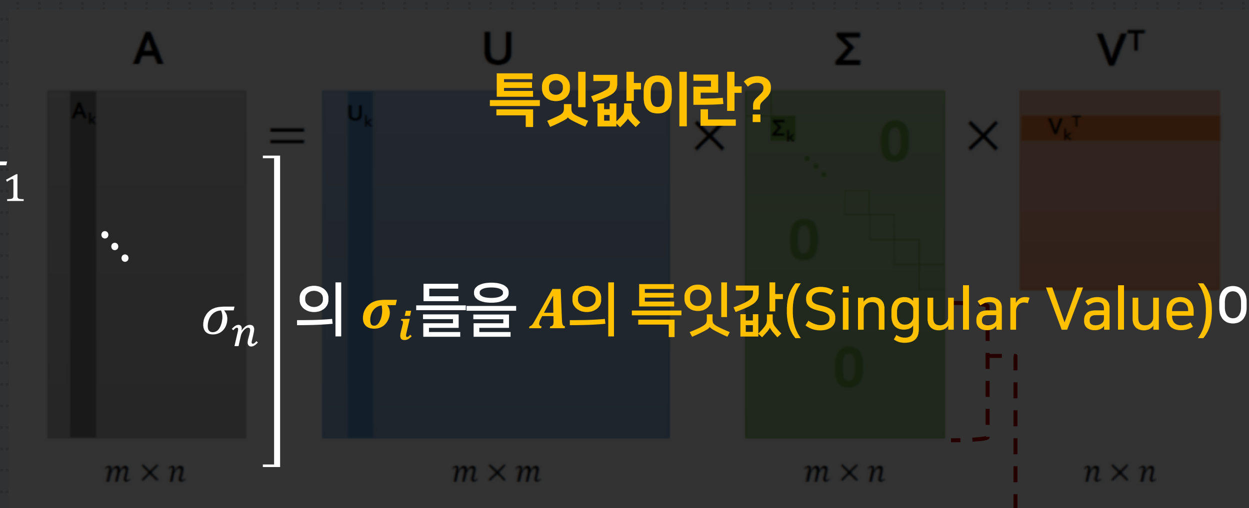
특잇값 분해



직사각 행렬 A

= 직교 행렬 U × 대각 직사각 행렬 Σ × 직교 행렬 V^T

특잇값 분해



직사각 행렬 A

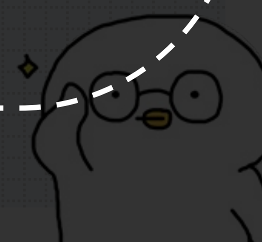
= 직교 행렬 U × 대각 직사각 행렬 Σ × 직교 행렬 V^T

특잇값이란?

$$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix}$$

의 σ_i들을 A의 특잇값(Singular Value)이라고 함

σ_i는 AA^T나 A^TA를 고윳값 분해해서 나오는 고윳값의 제곱근



특잇값 분해

U 도출하기

$$AA^T = (U\Sigma V^T)(U\Sigma V^T)^T = (U\Sigma V^T)(V\Sigma^T U^T)$$
$$= U\Sigma^T U^T = U\Lambda U^T$$
$$Then, U \in \mathbb{R}^{m\times m} \; (\because AA^T \in \mathbb{R}^{m\times m})$$

실수 대칭행렬 → U 직교행렬

V^T 도출하기

$$A^T A = (U\Sigma V^T)^T (U\Sigma V^T) = (V\Sigma^T U^T)(U\Sigma V^T)$$
$$= V\Sigma^2 V^T = V\Lambda V^T$$
$$Then, V^T \in \mathbb{R}^{m\times m} \; (\because A^T A \in \mathbb{R}^{m\times m})$$

특잇값 분해

U 도출하기

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

$$= U\Sigma^T U^T = U\Lambda U^T$$

$$Then, U \in \mathbb{R}^{m\times m} \ (\because AA^T \in \mathbb{R}^{m\times m})$$

실수 대칭행렬 → V^T 직교행렬

V^T 도출하기

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

$$= V\Sigma^2 V^T = V\Lambda V^T$$

$$Then, V^T \in \mathbb{R}^{m\times m} \ (\because A^T A \in \mathbb{R}^{m\times m})$$

특잇값 분해의 이해

특잇값 분해 *Singular Value Decomposition*

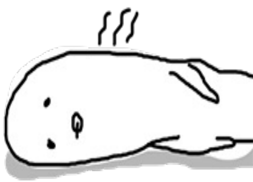
앞서, 직교하는 벡터 집합에 대하여 선형변환 후에

크기는 변하지만 **여전히 직교**하는 직교집합을 찾는 것이라고 했음

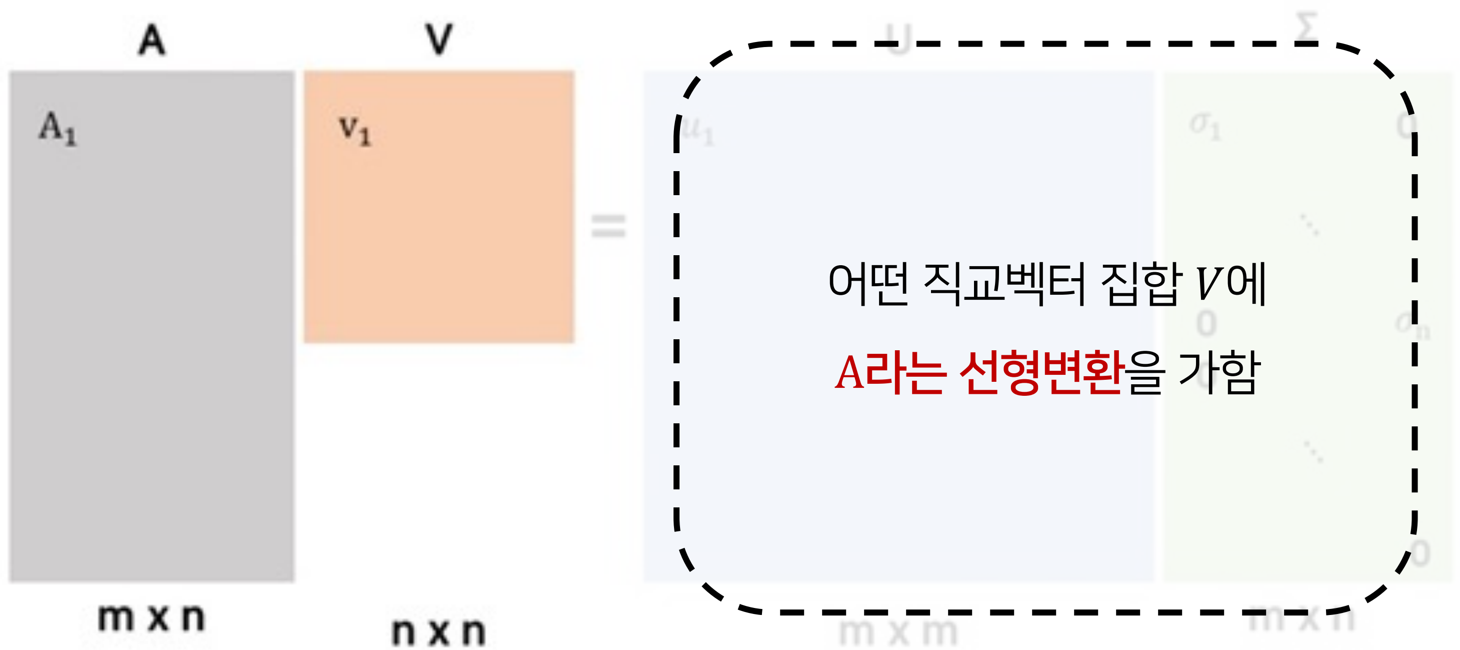
$$A = U\Sigma V^T$$

$$AV = U\Sigma$$

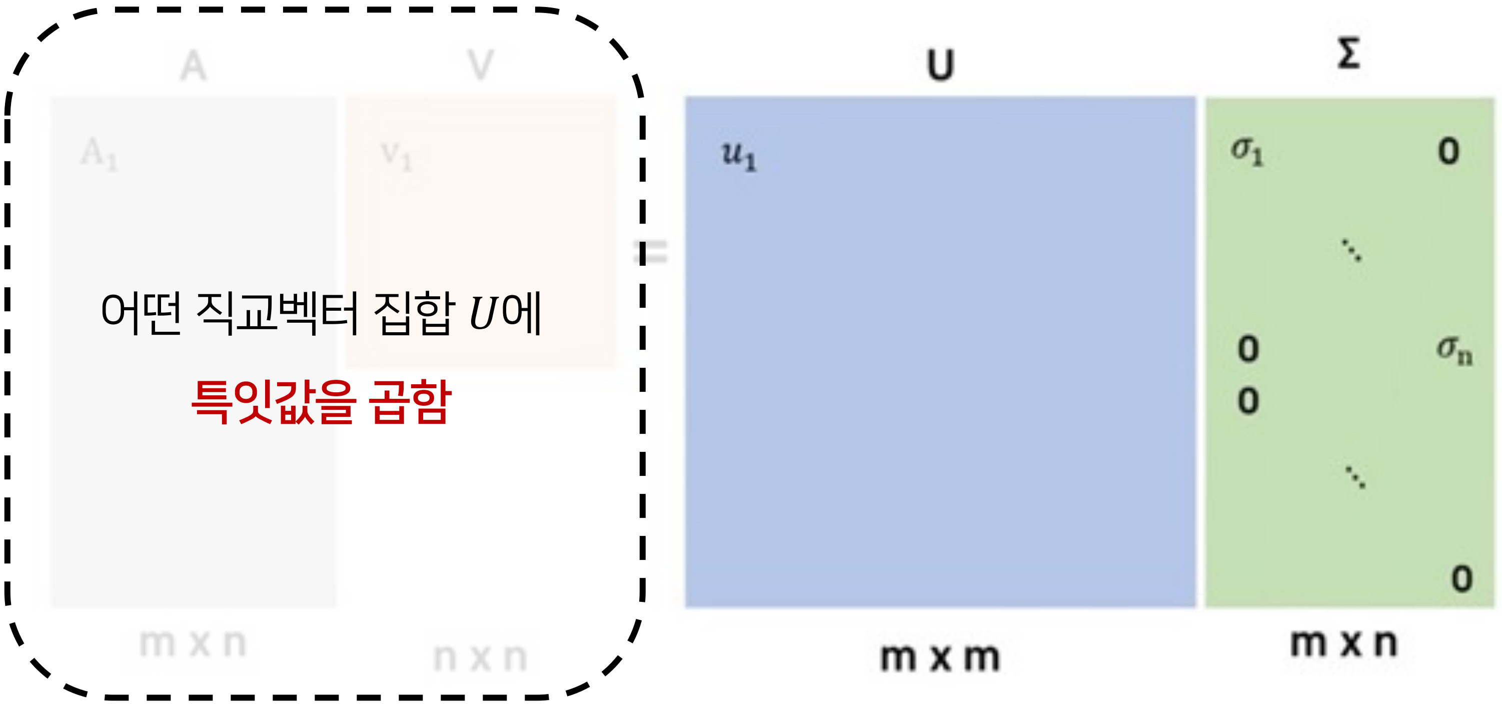
이해하기 쉽게 식을 변형하여 분석



특잇값 분해의 이해



특잇값 분해의 이해



이때, **스케일링**을 위해 Σ를 사용

특잇값 분해의 이해

직사각행렬 A에 대해서 선형변환을 했을 때,

벡터공간을 유지하는 공통의 벡터를 찾는 것(EVD)은 **불가능**

A로 선형변환 이후에도

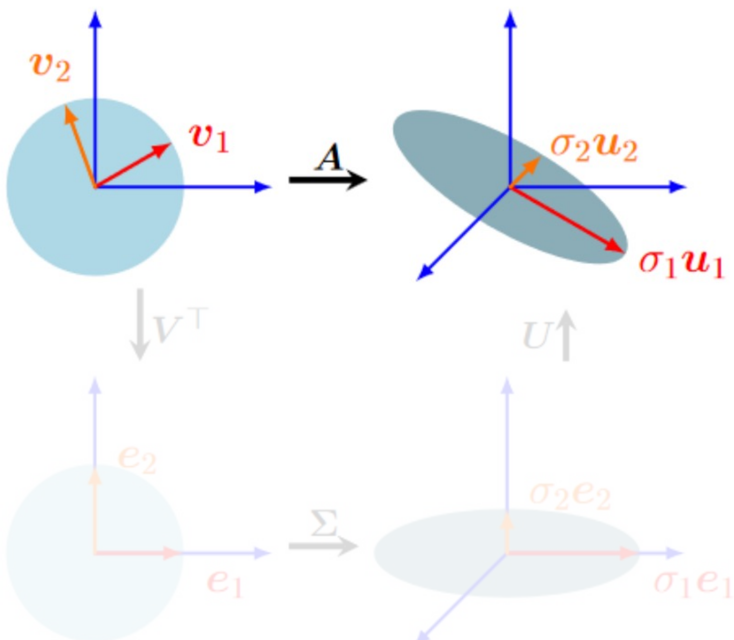
직교성만은 유지(SVD)하는 벡터 집합을 찾자!

어떤 직교벡터 집합 U 에

특잇값을 곱함

스케일링을 위해 Σ를 사용

특잇값 분해의 기하학적 의미



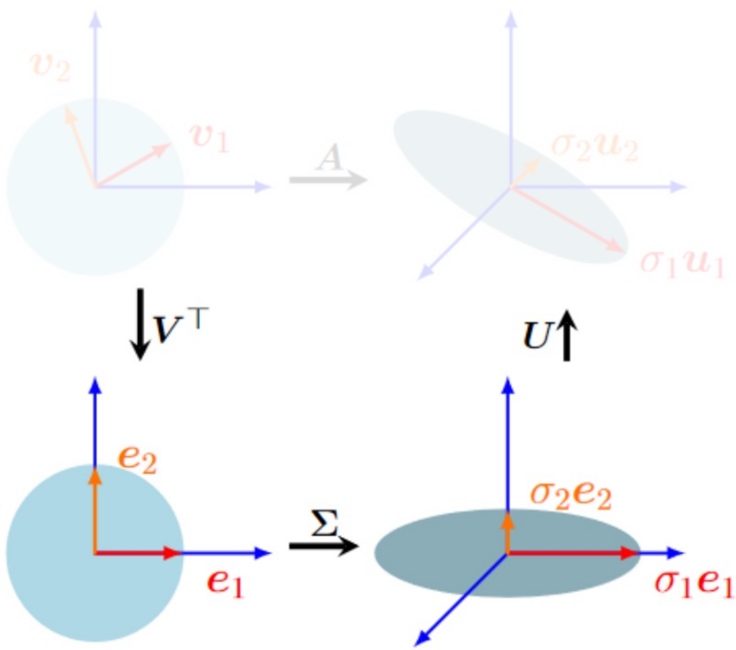
- 행렬 A는 2×3 변환이므로 차원이 변화함

- V^T와 U는 벡터공간을 회전시킴
- Σ는 벡터공간을 스케일링 + 차원변화

- 특잇값 분해란? A변환을

회전→스케일링 + 차원변화→회전으로 분해!

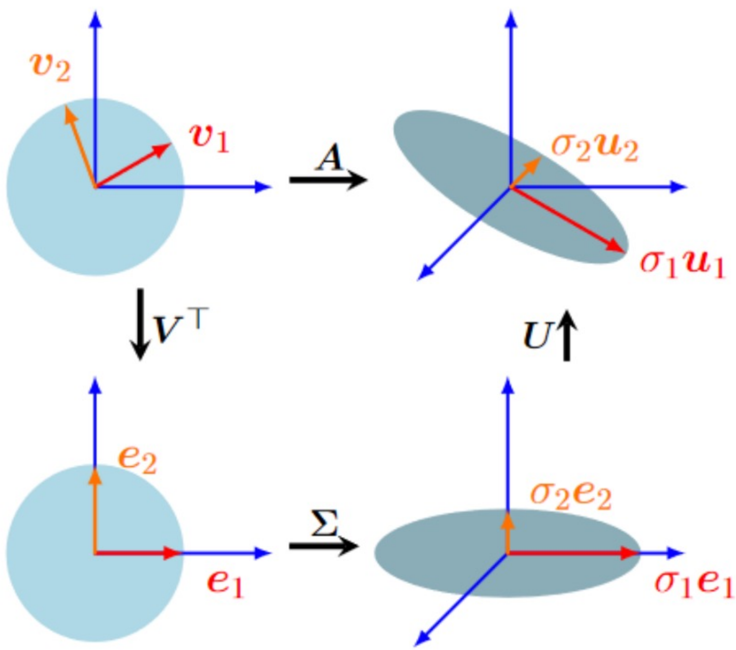
특잇값 분해의 기하학적 의미



- 행렬 A는 2×3 변환이므로 차원이 변화함
- V^T와 U는 벡터공간을 회전시킴
- Σ는 벡터공간을 스케일링 + 차원변화
- 특잇값 분해란? A변환을

회전⟶스케일링 + 차원변화⟶회전으로 분해!

특잇값 분해의 기하학적 의미



- 행렬 A는 2×3 변환이므로 차원이 변화함
- V^T와 U는 벡터공간을 회전시킴
- Σ는 벡터공간을 스케일링 + 차원변화
- 특잇값 분해란? A변환을

회전⟶스케일링 + 차원변화⟶회전으로 분해!

특잇값 분해의 이점

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

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

$$= \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T \cdots = \sum_{i=1}^{n} \sigma_i u_i v_i^T$$

$$dim(\sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T) = dim(A)$$

→ 행렬 A를 **정보량** σ_i에 따라 **여러 개의 layer**로 쪼갤 수 있음

특잇값 분해의 응용

Reduced SVD

기존 SVD는 비어 있는 열이 많아 비효율적

→ 여러 방법론으로 **빈 공간을 제거**하면 **효율적인 연산**이 가능

Reduced SVD
- Thin SVD — Σ 의 비어있는 행들을 제거
- Compact SVD — 특잇값이 0인 부분도 제거
- Truncated SVD — 상위 t개 특잇값만 사용

특잇값 분해의 응용

Reduced SVD

기존 SVD는 비어 있는 열이 많아 비효율적

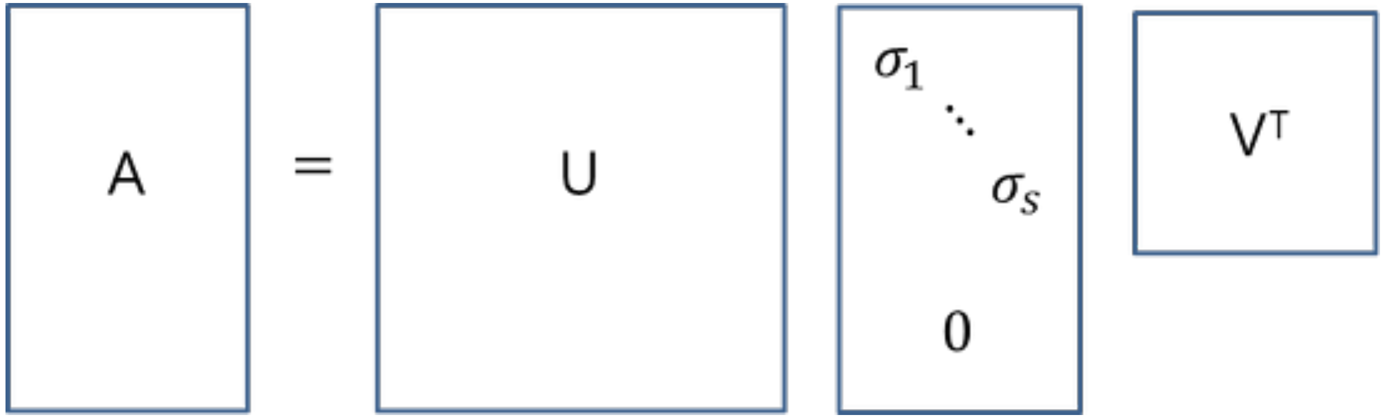
→ 여러 방법론으로 **빈 공간을 제거**하면 **효율적인 연산**이 가능

Reduced SVD
- Thin SVD — Σ 의 비어있는 행들을 제거
- Compact SVD — 특잇값이 0인 부분도 제거
- Truncated SVD — 상위 t개 특잇값만 사용

특잇값 분해의 응용

Full SVD

기존의 SVD로, **정보가 비어 있는 부분**이 다수 존재

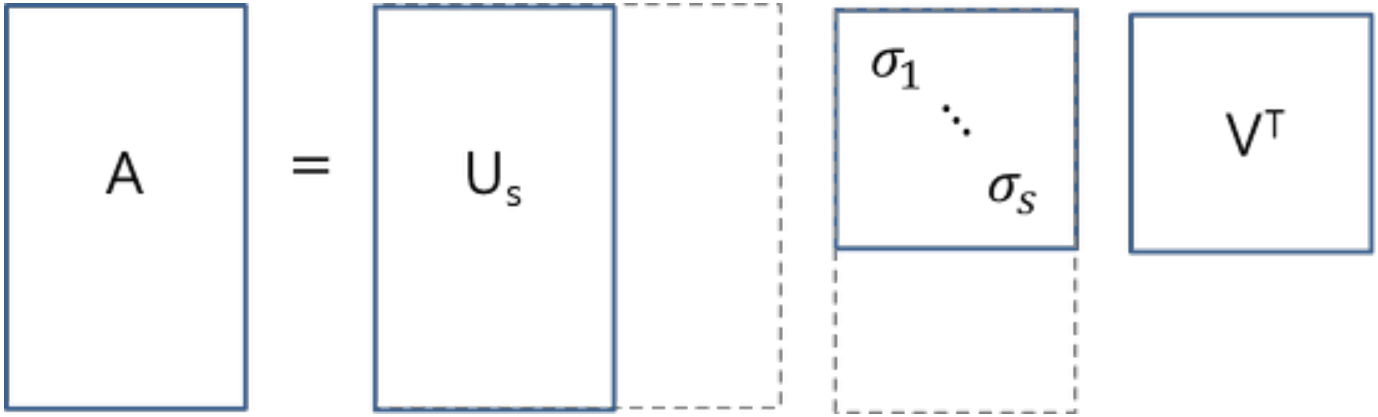


→ 0인 부분 모두 연산하므로 **연산량 낭비**

특잇값 분해의 응용

Thin SVD

Σ의 하단의 비어 있는 **비대각 부분**과 이에 **대응하는 U의 열**을 제거

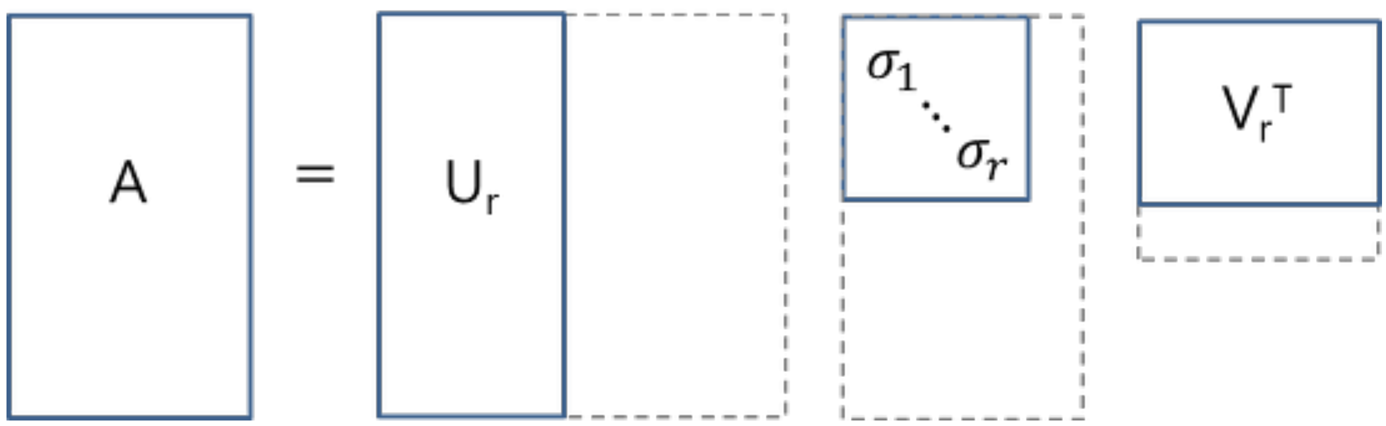


→ 0이었던 부분이 삭제되므로 A 복원 가능

특잇값 분해의 응용

Compact SVD

Thin SVD에 더해 Σ 중 σ_i**가 0인 열**과 이에 **대응하는 U의 열**을 제거

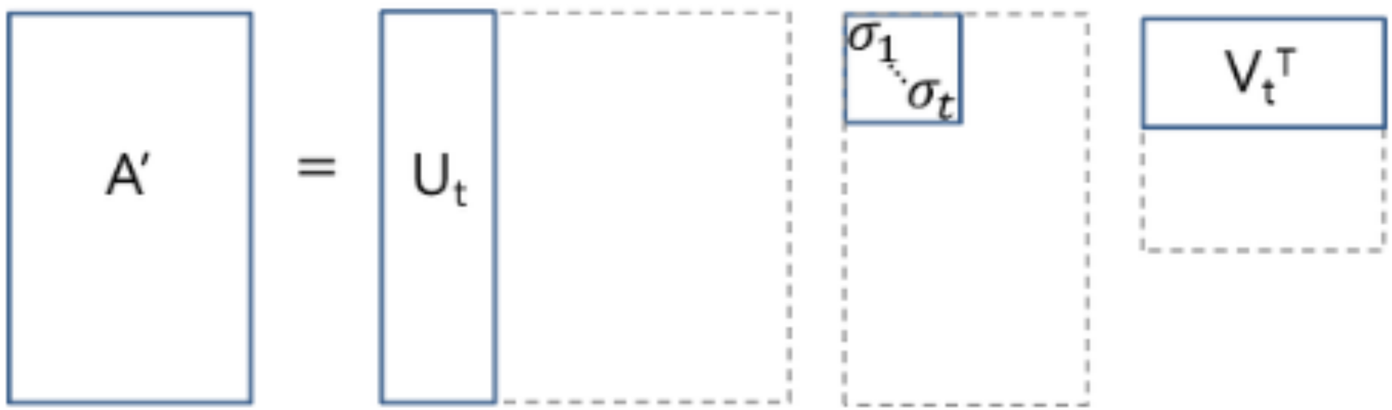


→ 정보량이 있는 (특잇값이 양수) 열만 남기고 제거

특잇값 분해의 응용

Truncated SVD

Σ 의 **상위 t개 σ_i**만 골라 이에 **대응하는 U의 열**을 제거



→ 온전한 복원이 아닌 데이터의 정보를 **압축**하여 A를 **근사**

5 특잇값 분해

Truncated SVD

선택한 **특잇값의 개수에 따라** 복원 행렬의 원본 행렬 **복원도**가 달라짐

$$\sum_{i=1}^{n=20} \sigma_i u_i v_i^T$$

$$\sum_{i=1}^{n=100} \sigma_i u_i v_i^T$$

Truncated SVD

$$\sum_{i=1}^{n=512} \sigma_i u_i v_i^T$$

원본

THANK YOU

세미나입니다. 자중해주세요

죄송합니다…잘못보냈습니디..