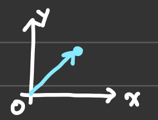
AI100

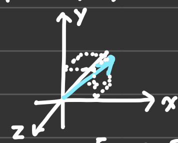
AI100

벡터

《벡터 : 공간의 한 점과 원점으로부터의 거리》

예





$v_1 = [1]$ $v_2 = [1, 2]$ $v_3 = [1, 3, 7]$

연산 : Scala곱, 덧·뺄셈, 곱 (같은 shape끼리만)

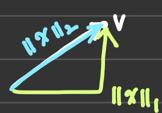
길이가 늘어나

벡터 이동

$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$

$x \odot y = \begin{bmatrix} x_1 y_1 \\ x_2 y_2 \end{bmatrix}$

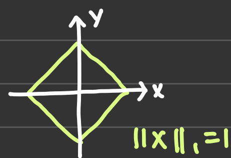
《norm : 벡터의 거리》

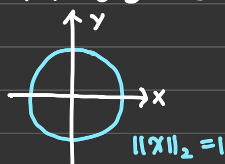


$\|x\|_1$ 직각 이동한 합

$\|x\|_2$ 대각선 거리

→ norm 종류에 따라 기하학적 성질이 달라짐





예. 원 (원점으로부터 거리가 같은 벡터의 합)

Robust 학습, Lasso 회귀

Laplace 근사 Ridge 회귀

《두 벡터 사이의 각도》

$$\|x-y\|_2 = \|y-x\|_1$$

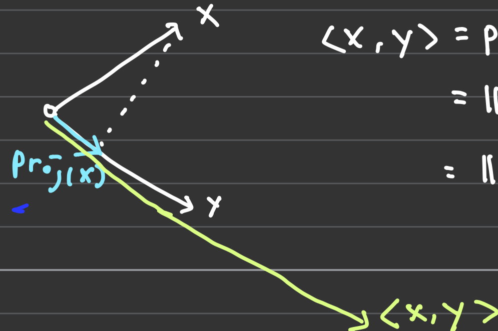
$$\|x-y\|_2^2 = \|x\|_2^2 + \|y\|_2^2 - 2\|x\|_2\|y\|_2\cos\theta$$

$$\cos\theta = \frac{\|x\|_2^2 + \|y\|_2^2 - \|x-y\|_2^2}{2\|x\|_2\|y\|_2}$$

$$= \frac{2\langle x, y\rangle}{2\|x\|_2\|y\|_2}$$

《내적》

$\underbrace{\text{정사영된 벡터의 길이}}_{proj(x)}$를 $\underbrace{y}_{\times\, y}$만큼 늘린 것



$$\langle x, y\rangle = proj(x)\cdot y$$

$$= \|x\|_2\cos\theta \cdot \|y\|_2$$

$$= \|x\|_2\|y\|_2\cos\theta$$

↳ 유사도 측정

행렬

공간 상 한 점   ⇒ 점들의 모임

《행렬》 벡터를 원소로 가지는 2차원 배열

$$X = \begin{bmatrix} 1 & -2 & 3 \\ 7 & 5 & 0 \\ -2 & -1 & 2 \end{bmatrix}$$

```
np.array([[1,-2,3],   ← 행이 기본단위
          [7,5,0],
          [-2,-1,2]])
```

표기법

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{array}{c} \\ \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \end{array} \begin{array}{l} \text{행벡터} \\ x_1 \\ x_2 \\ \\ x_n \end{array} = (x_{ij})$$

열 벡터

《transpose matrix》 행과 열의 인덱스가 바뀐 행렬

$$X^T = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & & x_{nm} \end{bmatrix} = (x_{ji})$$

《덧셈, 뺄셈, 성분곱, 스칼라곱》 벡터랑 똑같음

$$X \pm Y = (x_{ij} \pm y_{ij})$$

$$X \odot Y = (x_{ij} \times y_{ij})$$

$$aX = ax_{ij}$$

《 matrix multiplication 》

X의 i번째 행벡터   Y의 j번째 열 벡터 내적

갯수 같아야

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \quad Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1l} \\ y_{21} & y_{22} & \cdots & y_{2l} \\ \vdots & \vdots & & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{ml} \end{bmatrix} = \begin{array}{c} \\ n \end{array} \overset{l}{\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}}$$

$$XY = \left(\sum_k x_{ik} y_{kj} \right)$$

$$X = \begin{bmatrix} 1 & -2 & 3 \\ 7 & 5 & 0 \\ -2 & -1 & 2 \end{bmatrix} \quad Y = \begin{bmatrix} 0, & 1 \\ 1 & -1 \\ -2 & 1 \end{bmatrix}$$

$$XY = \begin{bmatrix} -8 & 6 \\ 5 & 2 \\ -5 & 1 \end{bmatrix} = X@Y$$

np.inner(X,Y) ≠ 행렬의 내적   X@Y

X가로·Y가로 ≠ X가로·Y세로

$$X = \begin{bmatrix} 1 & -2 & 3 \\ 7 & 5 & 0 \\ -2 & -1 & 2 \end{bmatrix} \quad Y^T = \begin{bmatrix} 0 & 1 & -2 \\ 1 & -1 & 1 \end{bmatrix}$$

$$XYT = \begin{bmatrix} -8 & 6 \\ 5 & 2 \\ -5 & 1 \end{bmatrix}$$

<< 행렬과 역행렬 >>

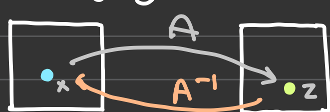
R^m : m차원공간 R^n : n차원 공간

x에 A라는 연산을 하면, z가 된다.

⇑

어떤 패턴 추출, 압축

< 역행렬 >>



R^m : m차원공간 R^n : n차원 공간

z에 A^{-1} 연산 하면, 다시 x가 된다.

$$AA^{-1} = A^{-1}A = I$$

역행렬 계산 가능? 조건

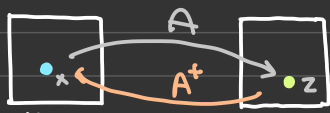
n=m 일 때만 가능.

A의 determinant가 0이 아니어야

부피 0. 즉 벡터들이 일직선 상에...

《역행렬 계산할 수 없다면? 》

pseudo-inverse



$\mathbb{R}^m$: m차원공간 $\quad$ $\mathbb{R}^n$: n차원 공간

$n \geq m \qquad A^+ = (A^TA)^{-1}A^T$

$n \leq m \qquad A^+ = A^T(AA^T)^{-1}$

예) 연립방정식

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1m}x_m &= b_1 \\
a_{12}x_1 + a_{22}x_2 + \cdots + a_{2m}x_m &= b_2 \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nm}x_m &= b_n
\end{aligned}
$$

$n \leq m$ 인 경우: 식이 변수 개수보다 작거나 같아야 함

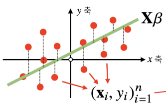
$n \leq m$ 이면 무어-펜로즈 역행렬을 이용하면 해를 하나 구할 수 있다

$$\rightarrow \mathbf{A}\mathbf{x} = \mathbf{b}$$

$$
\begin{aligned}
\Rightarrow \mathbf{x} &= \mathbf{A}^+\mathbf{b} \\
&= \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1}\mathbf{b}
\end{aligned}
$$

예) 선형회귀분석

- `np.linalg.pinv` 를 이용하면 데이터를 선형모델(linear model)로 해석하는 선형회귀식을 찾을 수 있다



$$\begin{bmatrix} — \mathbf{x}_1 — \\ — \mathbf{x}_2 — \\ \vdots \\ — \mathbf{x}_n — \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix} \neq \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$$\mathbf{X}\beta = \hat{\mathbf{y}} \approx \mathbf{y}$$

$$\min_{\beta} \|\mathbf{y} - \hat{\mathbf{y}}\|_2$$

L_2-노름을 최소화

$$
\begin{aligned}
\Rightarrow \beta &= \mathbf{X}^+\mathbf{y} \\
&= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}
\end{aligned}
$$

$n \geq m$ 인 경우: 데이터가 변수 개수보다 많거나 같아야 함

Moore-Penrose 역행렬을 이용하면 $\mathbf{y}$ 에 근접하는 $\hat{\mathbf{y}}$ 를 찾을 수 있다