[0] Basics Mathametics Snippet

[1] Eigenvalue Decomposition / Singular-Value Decomposition

① EVD

- $A \in M_{n\times n} \Longrightarrow A = PDP^T$

$$A = \underbrace{P}_{\text{orthogonal}}\,\underbrace{D}_{\text{diagonal } \in M_{n\times n}}\,P^T$$

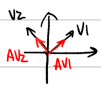
Geometric viewpoint of EVD : $AP = PD$이므로

$P = \begin{bmatrix} | & & | \\ v_1 & \cdots & v_n \\ | & & | \end{bmatrix}$, AP는 orthogonal basis $\{v_1, v_2, \cdots, v_n\}$의 선형변환($A$)

$PD = \begin{bmatrix} | & & | \\ d_1v_1 & \cdots & d_nv_n \\ | & & | \end{bmatrix}$이므로 A에 의한 Basis의 선형변환이 곧 d_i배 (길이 축소/확대)

대신 방향은 그대로!

e.g) in 2D, $P = \begin{bmatrix} | & | \\ v_1 & v_2 \\ | & | \end{bmatrix}$



Av_1, Av_2에 대한 v_1, v_2의 확대·축소 비율 $= d_1, d_2$ (eigenvalue)

② SVD

- $A \in M_{m\times n} \Longrightarrow A = U\Sigma V^T$

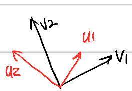
$U = n\times n$ orthogonal matrix, $V^T = m\times m$ orthogonal matrix

$\Sigma = n\times m$ matrix which every except the first r diagonal entries is zero where r is the rank of A

$$\begin{array}{cc} & \begin{array}{cc} r & m-r \end{array} \\ \begin{array}{c} r \\ n-r \end{array} & \left[\begin{array}{c|c} \sigma_r & 0 \\ \hline 0 & 0 \end{array}\right] \end{array}$$

Geometric viewpoint of SVD: $A = U\Sigma V^T$

$\Rightarrow AV = U\Sigma$, $B_V = \{v_1, v_2, \cdots, v_n\}$, $B_U = \{u_1, u_2, \cdots, u_m\}$



SVD의 기하학적 의미 = orthogonal basis를 다른 orthogonal basis로 변환하는 행렬 A, 이때 basis의 크기는 σ_i (singular value) 배되고 EVD와 다르게 방향도 달라진다.

③ Python Implementation

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

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

$AA^T = U\Sigma\Sigma^T U^T$

그러므로 singular value $= \sqrt{A^TA\,(AA^T)\text{ eigenvalue}}$

그러므로 EVD 구현 ⇒ SVD 구현 가능!

EVD를 위한 Algorithms are well-known

search for "eigenvalue algorithm"

let's implement SVD from EVD!

Remark)

- 최소자승법과 pseudoinverse 개념과도 직결된다!!
- Truncated SVD