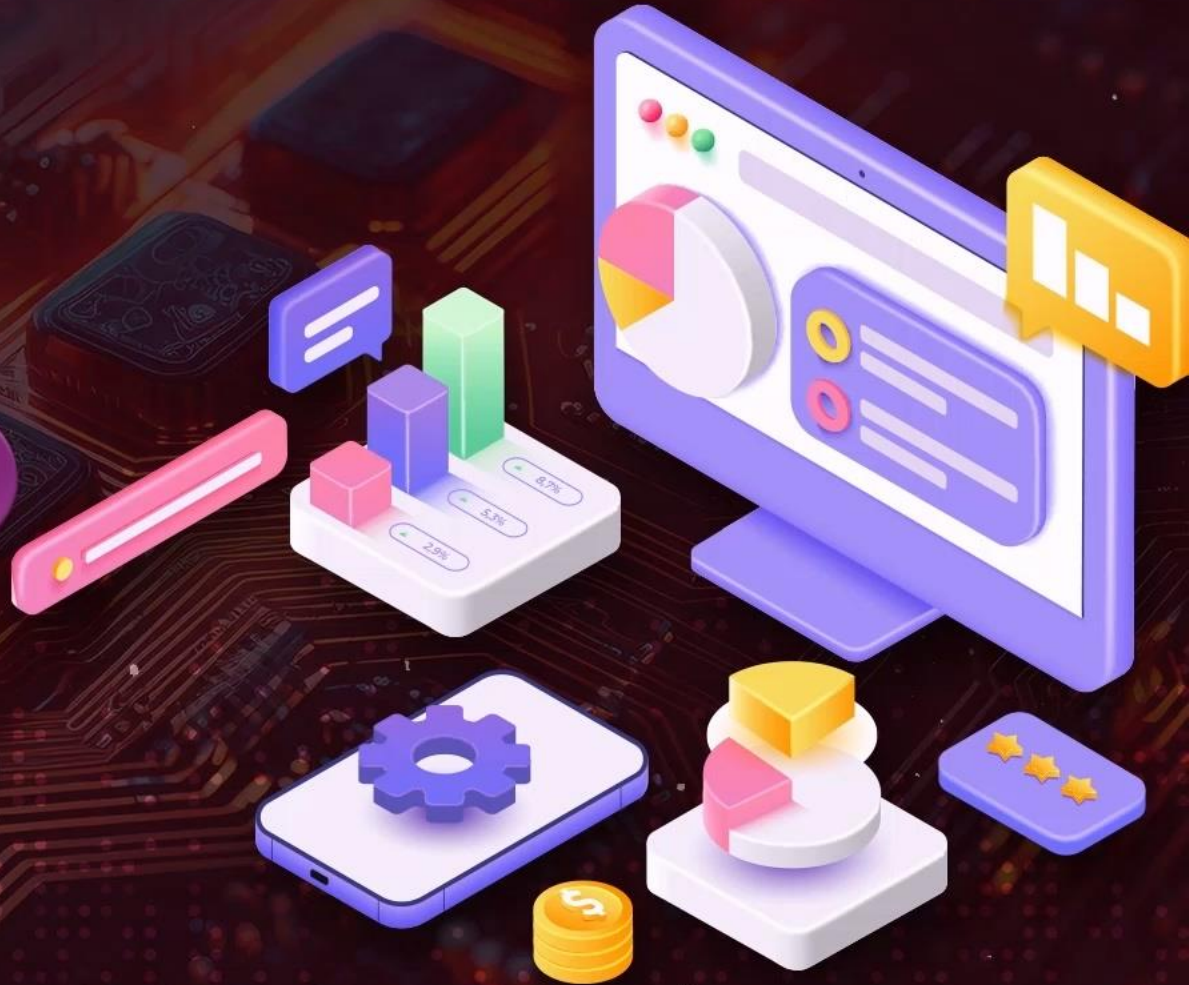
기초통계 및 머신러닝

군집분석 및 차원축소



특이값 분해(SVD)

- 특이값 분해(SVD)는 행렬을 직교행렬(U, V)과 대각행렬(Σ)로 분해하여 데이터를 효율적으로 분석하는 방법임을 설명할 수 있다.
- 저차원 근사를 활용해 데이터의 압축, 시각화, 이미지 처리 등 다양한 응용 사례에서 특이값 분해를 적용할 수 있다.

Matrix and (Linear) Transformation

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

$$\vec{y} = M\vec{x}$$

$$\begin{bmatrix} \ \\ \ \\ \ \end{bmatrix} = \begin{bmatrix} \quad & \quad & \quad \\ \ & & \\ \ & & \end{bmatrix} \begin{bmatrix} \ \\ \ \\ \ \end{bmatrix}$$

Given		Interpret
linear transformation	→	matrix
matrix	→	linear transformation

Linear Transformation

- **If $\vec{v}_1$ and $\vec{v}_2$ are basis, and we know $T(\vec{v}_1) = \vec{\omega}_1$ and $T(\vec{v}_2) = \vec{\omega}_2$**

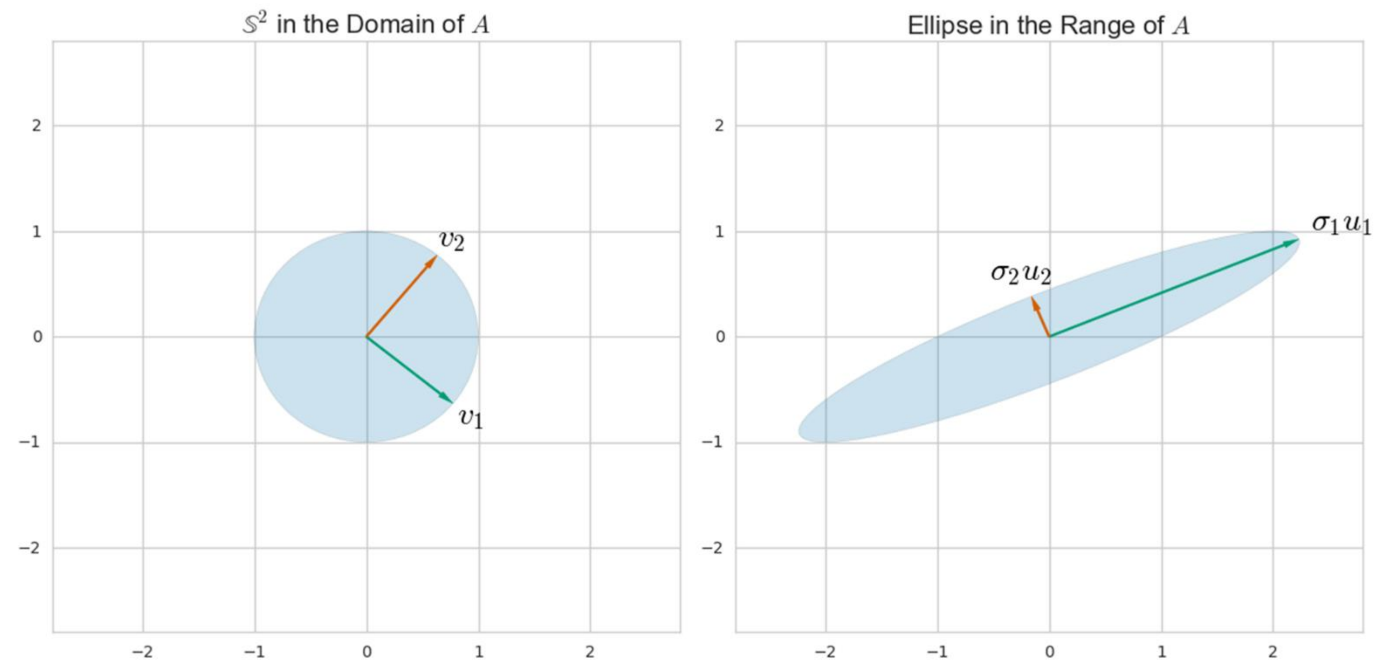
- **Then, for any $\vec{x}$**

$$
\begin{aligned}
\vec{x} &= a_1\vec{v}_1 + a_2\vec{v}_2 && (a_1 \text{ and } a_2 \text{ unique}) \\
T(\vec{x}) &= T(a_1\vec{v}_1 + a_2\vec{v}_2) \\
&= a_1T(\vec{v}_1) + a_2T(\vec{v}_2) \\
&= a_1\vec{\omega}_1 + a_2\vec{\omega}_2
\end{aligned}
$$

Only thing that we need is to observe how basis are linearly-transformed

Geometry of Linear Maps

Matrix A (or linear transformation)
= rotate + stretch/compress



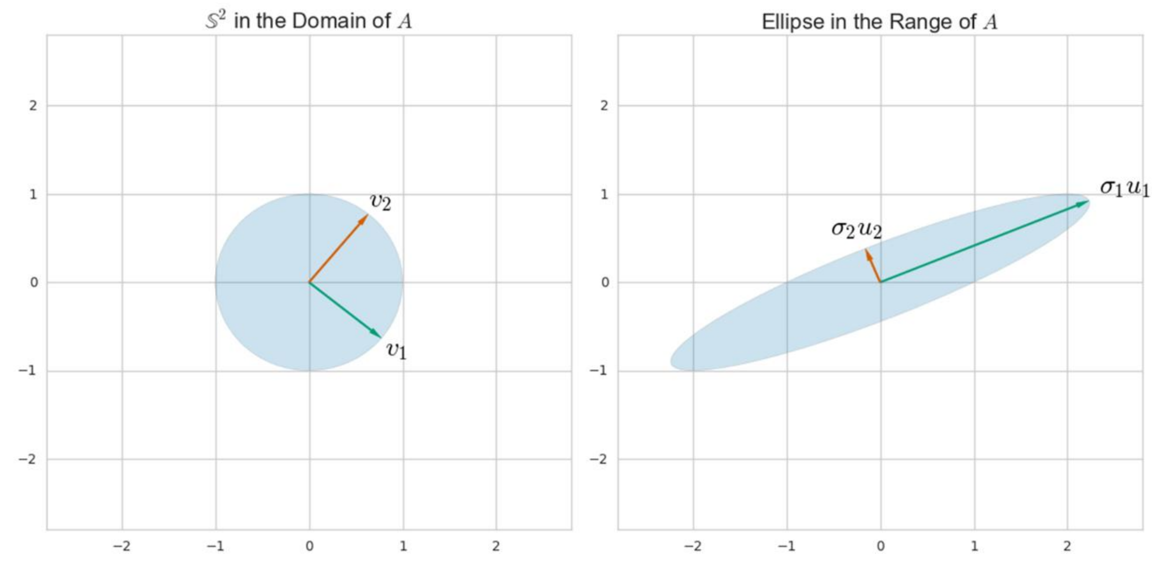
Singular Values and Singular Vectors

- The numbers $\sigma_1, \cdots, \sigma_n$ are called the singular values of A by convention, $\sigma_i > 0$
- The vectors $u_1, \cdots, u_n$ these are unit vectors along the principal semi-axes of AS
- The vectors $v_1, \cdots, v_n$ these are the preimages of the principal semi-axes, defined so that

특이값 분해

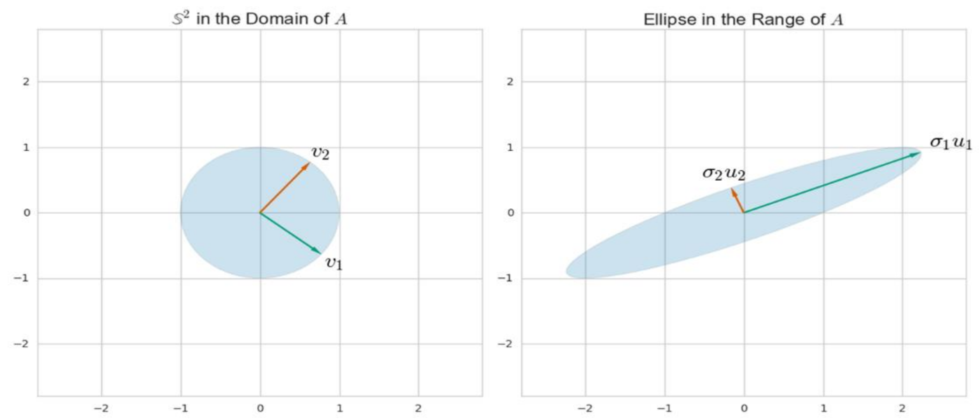
Singular Values and Singular Vectors

$$\boxed{Av_i = \sigma_i u_i}$$



특이값 분해

Singular Values and Singular Vectors



$$
\begin{aligned}
A\begin{bmatrix} v_1 & v_2 & \cdots & v_r \end{bmatrix} &= \begin{bmatrix} Av_1 & Av_2 & \cdots & Av_r \end{bmatrix} \\
&= \begin{bmatrix} \sigma_1 u_1 & \sigma_2 u_2 & \cdots & \sigma_r u_r \end{bmatrix} \\
&= \begin{bmatrix} u_1 & u_2 & \cdots & u_r \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{bmatrix}
\end{aligned}
$$

$$\therefore\ AV = U\Sigma \quad (r \le m, n)$$

Thin Singular Value Decomposition

$A \in \mathbb{R}^{m\times n}$, skinny and full rank (*i.e.*, $r = n$)

$$\hat{U} = \begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix}$$

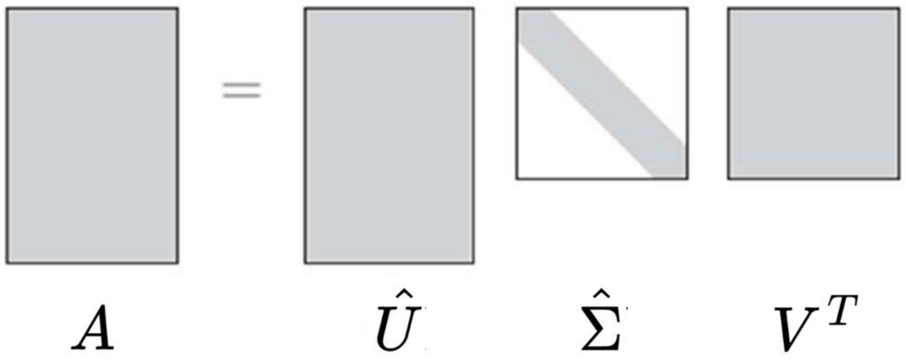
$$\hat{\Sigma} = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}$$

$$V = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$$

Thin Singular Value Decomposition

$$Av_i = \sigma_i u_i \quad \text{for } 1 \le i \le n$$

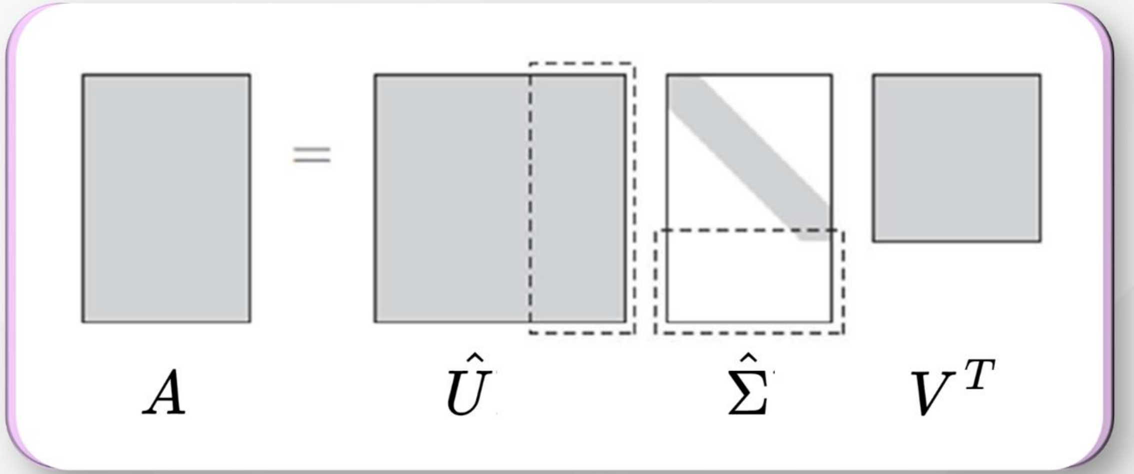
$$A = \hat{U}\hat{\Sigma}V^T$$



Full Singular Value Decomposition

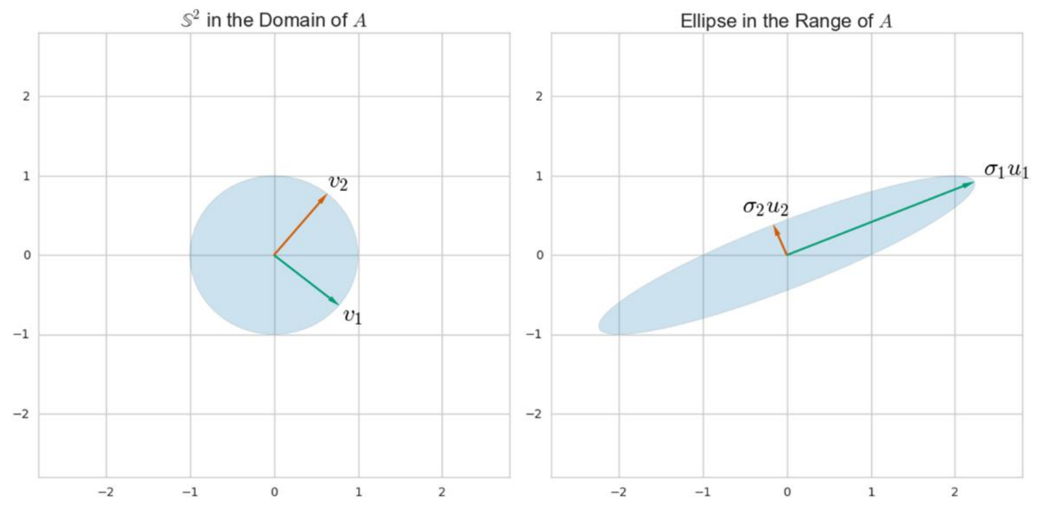
- We can add extra orthonormal columns to U
- We also add extra rows of zeros to Σ

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



특이값 분해

Low Rank Approximation: Dimension Reduction



$$A = U_r \Sigma_r V_r^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

$$\tilde{A} = U_k \Sigma_k V_k^T = \sum_{i=1}^{k} \sigma_i u_i v_i^T \qquad (k \leq r)$$

특이값 분해

Low Rank Approximation: Dimension Reduction

$$
\begin{aligned}
x &= c_1 v_1 + c_2 v_2 \\
&= \langle x, v_1 \rangle v_1 + \langle x, v_2 \rangle v_2 \quad \because \langle x, v_1 \rangle = c_1 \langle v_1, v_1 \rangle + c_2 \langle v_2, v_1 \rangle \\
&= (v_1^T x) v_1 + (v_2^T x) v_2
\end{aligned}
$$

$$
\begin{aligned}
Ax &= c_1 A v_1 + c_2 A v_2 \\
&= c_1 \sigma_1 u_1 + c_2 \sigma_2 u_2 \\
&= u_1 \sigma_1 v_1^T x + u_2 \sigma_2 v_2^T x \\
&= \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix} x \\
&= U \Sigma V^T x \\
&\approx u_1 \sigma_1 v_1^T x \quad (\text{if } \sigma_1 \gg \sigma_2)
\end{aligned}
$$

Example: Image Approximation

Approximation of A

$$\tilde{A} = U_k \Sigma_k V_k^T = \sum_{i=1}^{k} \sigma_i u_i v_i^T \qquad (k \leq r)$$

Original Image

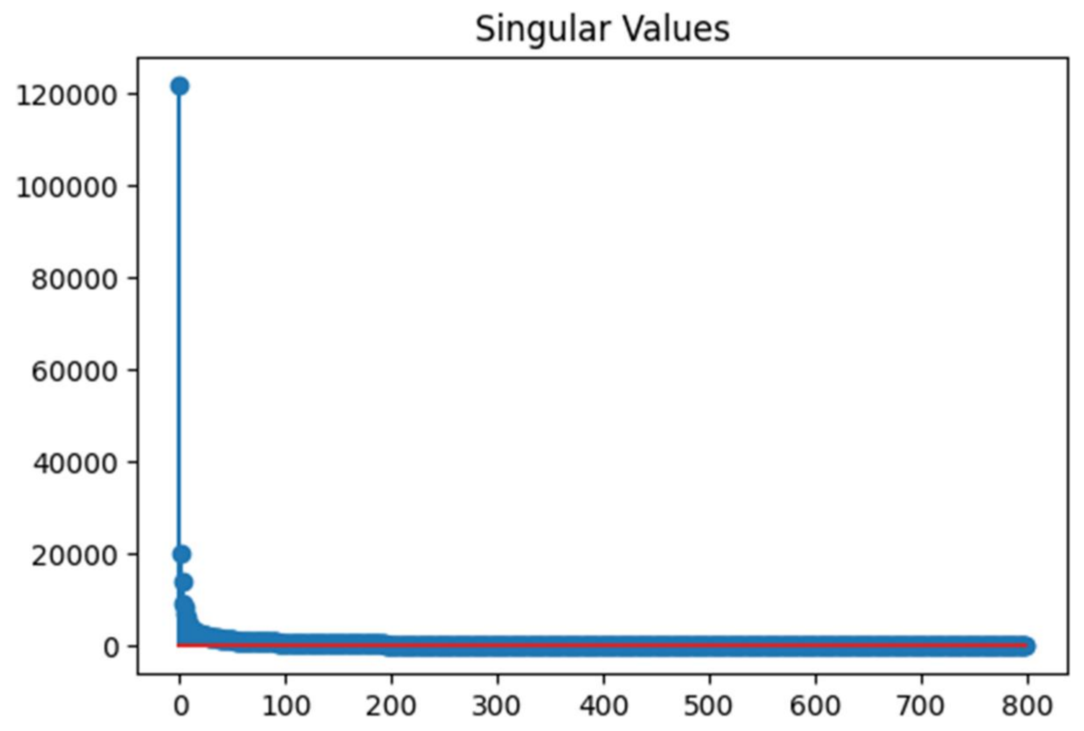
```
>> % A = img; % 이미지 데이터를 행렬 A로 설정 (사전에 정의된 img)

% SVD 계산
[U, S, VT] = svd(A);

% U, S, VT를 MATLAB의 기본 출력으로 처리
U = U; % MATLAB에서 기본적으로 행렬 형식
S = diag(S); % MATLAB의 S는 대각 행렬로 자동 생성됨
VT = VT; % MATLAB에서 V는 전치 형태로 반환
```

Example: Image Approximation

Singular values



Example: Image Approximation

Rank 20 (k=20)

Original Image

Approximated image w/ rank = 20

Example: Image Approximation

Approximated images with k varied