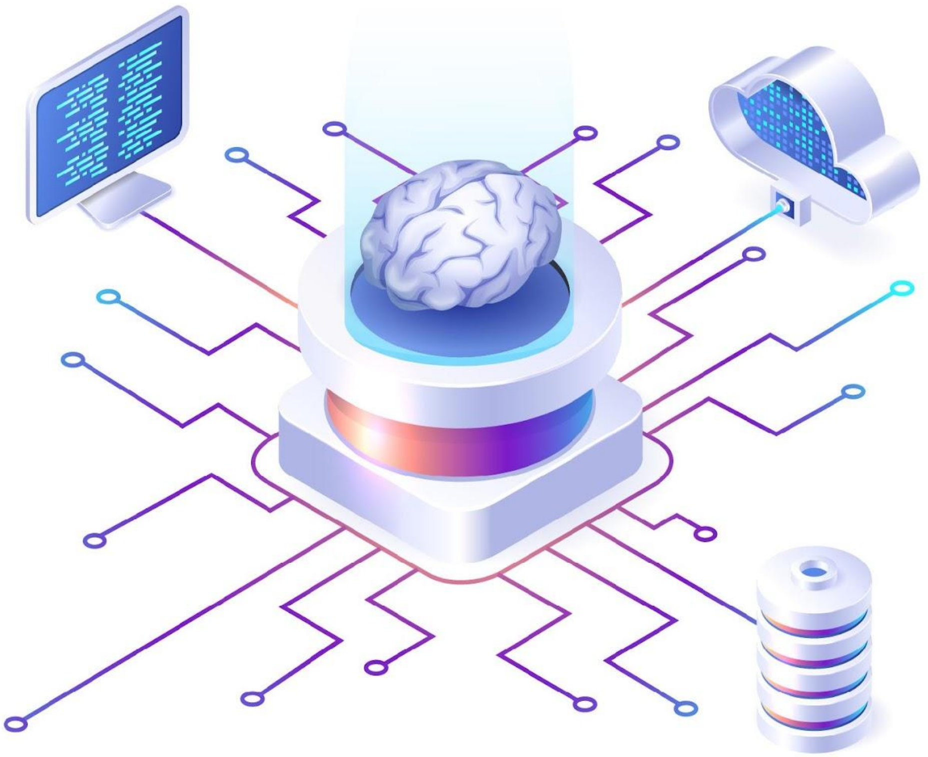
선형대수 기초

실무형 인공지능 자연어 처리



1 벡터 (Vector)

벡터

벡터는 크기와 방향을 가지고 있는 개념을 표현한 것이다. 자연어 처리에서 벡터를 많이 사용한다. 단어나 문장을 텍스트 그대로 사용할 수 없기 때문에 단어를 표현하거나 문장을 표현할 때 벡터를 사용한다.

$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, y = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

벡터의 덧셈

두 벡터간 덧셈은 각 벡터의 원소들간 값을 더하여 계산한다.

$$x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, y = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$x + y = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1+3 \\ 2+4 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

벡터의 뺄셈

마차가지로 두 벡터간 뺄셈은 각 벡터의 원소들간 값을 빼서 계산한다.

$$x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, y = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$x - y = \begin{bmatrix} 1 \\ 2 \end{bmatrix} - \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 - 3 \\ 2 - 4 \end{bmatrix} = \begin{bmatrix} -2 \\ -2 \end{bmatrix}$$

벡터의 스칼라 곱셈/나눗셈

벡테에 스칼라 곱셈과 나눗셈을 적용하면 스칼라 값을 각 원소에 곱하거나 나누어 계산한다.

$$x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$2x = 2 \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \cdot 1 \\ 2 \cdot 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

$$\frac{1}{2}x = \frac{1}{2} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \cdot 1 \\ \frac{1}{2} \cdot 2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

벡터의 norm

벡터의 노름(norm)은 벡터의 크기/길이를 의미한다.

$$\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$$

$$x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\|x\| = \sqrt{1^2 + 2^2} = \sqrt{5} \approx (2.236)$$

벡터의 내적(dot product)

두 벡터의 내적은 두 벡터의 개별 성분의 곱의 합이다. 두 개의 벡터 x 와 y가 있으면 내적은 다음과 같이 정의된다.

$$x \cdot y = x_1 y_1 + x_2 y_2 + \ldots x_n y_n$$

$$x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, y = \begin{bmatrix} -3 \\ 4 \end{bmatrix}$$

$$x \cdot y = x_1 y_1 + x_2 y_2 = (1 \cdot \ -3) + (2 \cdot 4) = 5$$

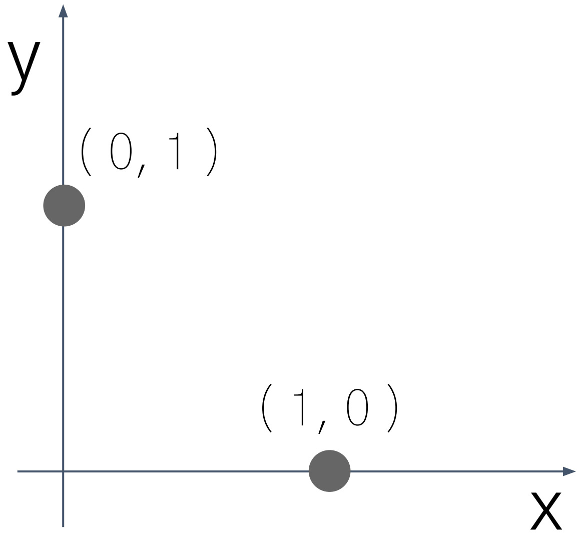
벡터의 직교

두 벡터 x 와 y가 내적이 0이면 서로 직교한다.

$$x \cdot y = 0$$

$$x = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$x \cdot y = x_1 y_1 + x_2 y_2 = (1 \cdot 0) + (0 \cdot 1) = 0$$



2 행렬 (Matrix)

행렬

행렬은 2차원 숫자 배열이다. 자연어 처리에서는 벡터의 묶음으로써 행렬을 사용한다.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

행렬의 곱셈

$$c_{ij} = \sum_k a_{ik} \cdot b_{kj} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + \ldots + a_{ik} \cdot b_{kj}$$

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}, B = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} \qquad C = A \cdot B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \cdot 4 + 2 \cdot 2 & 1 \cdot 3 + 2 \cdot 1 \\ 3 \cdot 4 + 4 \cdot 2 & 3 \cdot 3 + 4 \cdot 1 \\ 5 \cdot 4 + 6 \cdot 2 & 5 \cdot 3 + 6 \cdot 1 \end{bmatrix} = \begin{bmatrix} 8 & 5 \\ 20 & 13 \\ 32 & 21 \end{bmatrix}$$

Insight campus SeSAC

행렬의 곱셈

$$c_{ij} = \sum_k a_{ik} \cdot b_{kj} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + \ldots + a_{ik} \cdot b_{kj}$$

$$A = [a_1, a_2, \ldots, a_n], B = \begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{bmatrix}$$

$$A \cdot B = [a_1, a_2, \ldots, a_n] \cdot \begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{bmatrix}$$

$$= a_1 b_1 + a_2 b_2 + \ldots + a_n b_n = \sum_{i=1}^{n} a_i b_i$$

대각 행렬 (diagonal matrix)

행과 열의 크기가 같은 정방행렬 비대각 요소의 값이 모두 0이고 대각 요소만 0이 아닌 값을 가진 행렬을 대각 행렬이라 한다.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \end{bmatrix}$$

직교 행렬 (Orthogonal matrix)

$$X^T X = X X^T = I$$

$$XX^T = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I$$

단위 행렬 (diagonal matrix)

행과 열의 크기가 같은 정방행렬의 비대각 요소의 값이 모두 0이고 대각 요소만 1의 값을 가진 행렬을 단위 행렬이라 한다.

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

Insight campus SeSAC

전치 행렬

$$X = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, X^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

역행렬

$$XX^{-1} = I = X^{-1}X$$

$$X = \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix}, X^{-1} = \begin{bmatrix} 1 & -\frac{1}{2} \\ -2 & \frac{3}{2} \end{bmatrix}$$

$$XX^{-1} = \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} \\ -2 & \frac{3}{2} \end{bmatrix} = \begin{bmatrix} 3-2 & -\frac{3}{2}+\frac{3}{2} \\ 4-4 & -2+3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

3 공분산 행렬 (Covariance Matrix)

Insight campus SeSAC

공분산 행렬의 의미

- 공분산 행렬의 의미
  - 데이터의 의미 : 각 feature의 움직임이 얼마나 유사한가
  - 수학적 의미 : 선형변환

- PCA
  - 공분산 행렬의 eigenvector, eigenvalue
  - PCA = Projecting data onto eigenvector of 공분산 행렬

공분산 행렬의 의미 - 데이터 구조 (1)

- 데이터의 의미 : 각 feature의 움직임이 얼마나 유사한가
    - 행은 sample, 열은 feature를 의미
    - 열(feature)의 평균 값을 0으로 조정 (=각 feature의 값에 평균을 뺀 상태)

$$X = \begin{pmatrix} | & | & | & & | \\ X_1 & X_2 & X_3 & \cdots & X_d \\ | & | & | & & | \end{pmatrix} \in \mathbb{R}^{n \times d}$$

공분산 행렬의 의미 - 데이터 구조 (2)

- 데이터의 의미 : 각 feature의 움직임이 얼마나 유사한가
  - 행렬 X의 공분산 행렬 계산

$$Cov(x,y) = \frac{\sum (x-\bar{x})(y-\bar{y})}{n} \quad or$$

$$Cov_{xy} = \frac{\sum (x-\bar{x})(y-\bar{y})}{n-1}$$

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

공분산 행렬의 의미 - 데이터 구조 (3)

- 데이터의 의미 : 각 feature의 움직임이 얼마나 유사한가
  - 행렬 X의 공분산 행렬 계산

$$X^TX = \begin{pmatrix} — & X_1 & — \\ — & X_2 & — \\ & \cdots & \\ — & X_d & — \end{pmatrix} \begin{pmatrix} | & | & & | \\ X_1 & X_2 & \cdots & X_d \\ | & | & & | \end{pmatrix}$$

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

공분산 행렬의 의미 - 데이터 구조 (4)

- 데이터의 구조적 의미 : 각 feature의 변동이 얼마나 닮았나행렬 X의 공분산 행렬 계산
    - 공분산 행렬의 dot(X_1, X_1)은 X_1의 분산을 의미
    - 공분산 행렬의 dot(X_1, X_2)은 X_1과 X_2의 공분산을 의미

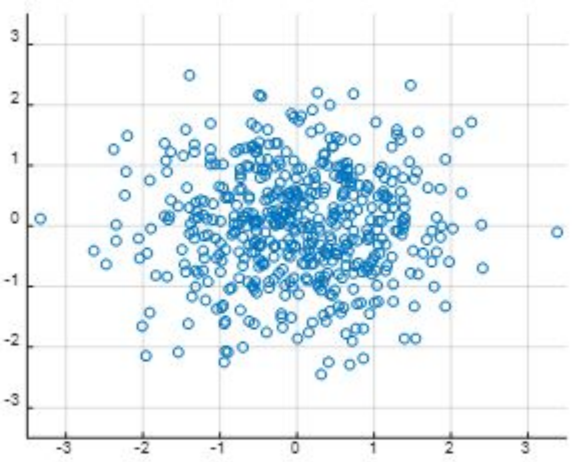
분산

$$\frac{X^T X}{n} = \frac{1}{n}\begin{pmatrix} dot(X_1, X_1) & dot(X_1, X_2) & \cdots & dot(X_1, X_d) \\ dot(X_2, X_1) & dot(X_2, X_2) & \cdots & dot(X_2, X_d) \\ \vdots & \vdots & \ddots & \vdots \\ dot(X_d, X_1) & dot(X_d, X_2) & \cdots & dot(X_d, X_d) \end{pmatrix}$$

공분산

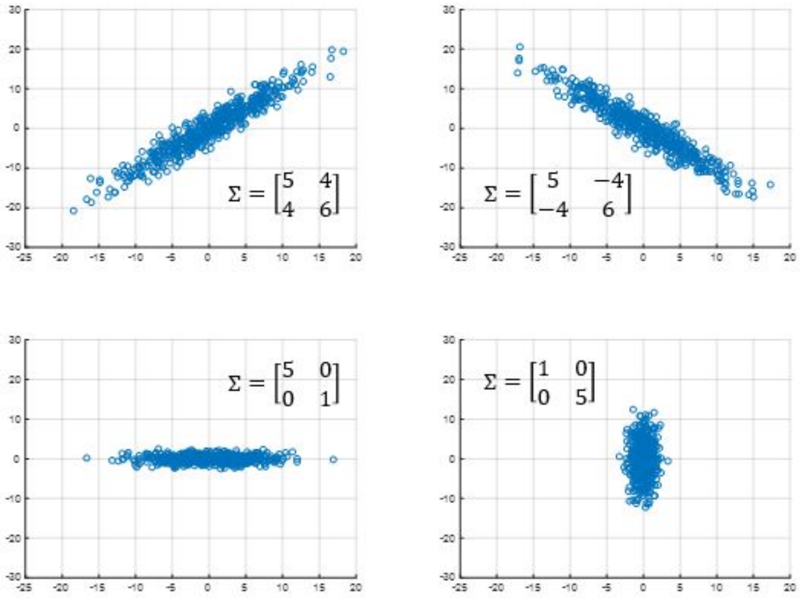
공분산 행렬의 의미 - 수학적 의미 (1)

- 수학적 의미 : 선형 변환 (shearing)



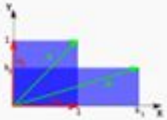
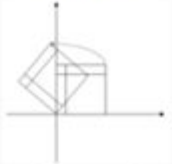
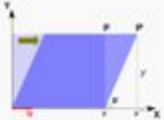
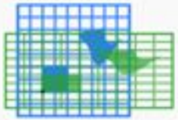
공분산 행렬의 의미 - 수학적 의미 (2)

- 수학적 의미 : 선형 변환 (shearing)



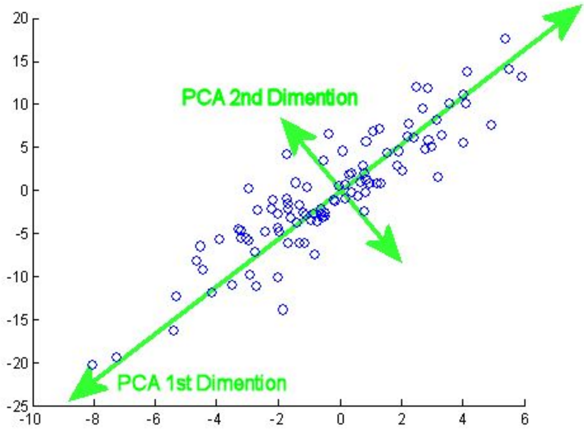
공분산 행렬의 의미 - 수학적 의미 (2)

- 수학적 의미 : 선형 변환 (shearing)

	scaling	unequal scaling	rotation	horizontal shear	hyperbolic rotation
illustration					
matrix	$\begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix}$	$\begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}$	$\begin{bmatrix} c & -s \\ s & c \end{bmatrix}$ $c = \cos\theta$ $s = \sin\theta$	$\begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} c & s \\ s & c \end{bmatrix}$ $c = \cosh\varphi$ $s = \sinh\varphi$

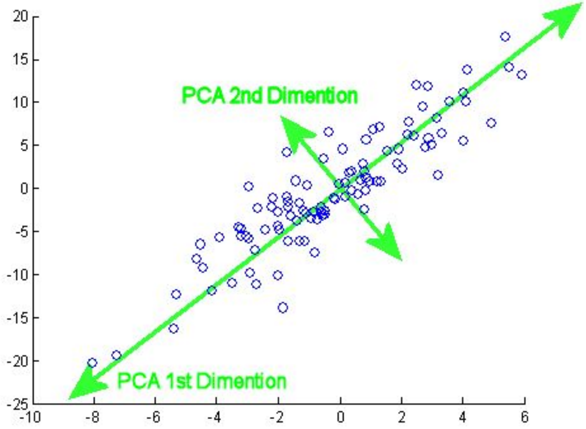
PCA (Principal Component Analysis)

- PCA 알고리즘은 데이터 구조를 잘 살리면서 차원을 감소할 수있게 하는 방법
  - 정사영 이후 데이터의 분포(=분산)이 제일 큰 것이 좋음
  - 공분산 형렬로 선형 변환할 때, 주축에 대해 정사영하는 것이 좋음



PCA (Principal Component Analysis)

- PCA 알고리즘은 데이터 구조를 잘 살리면서 차원을 감소할 수있게 하는 방법
  - 선형변환의 주축을 eigenvector라 부름
  - eigenvector를 찾는 다는 것은 “선형변환 이후 크기만 바뀌고 방향은 바뀌지 않는 벡터를 찾는것"
  - 이는 정사영 후 데이터 분포(분산)가 가장 큰 결과를 얻기위해 eigenvector에 정사영



PCA (Principal Component Analysis)

- PCA 알고리즘은 데이터 구조를 잘 살리면서 차원을 감소할 수있게 하는 방법
  - 3차원 데이터는?
    - 공분산 행렬을 고유값 분해하면 3개의 eigenvector가 나옴
    - 이 중 고유값이 큰순으로 2개의 eigenvector에 정사영 하면 2차원으로 축소됨
  - N차원 데이터는?
    - 공분산 행렬을 고유값 분해하면 N개의 eigenvector가 나옴
    - 이 중 고유값이 큰순으로 k개의 eigenvector에 정사영 하면 k차원으로 축소됨

PCA에서 왜 공분산 행렬을 사용하는가?

- 공분산 행렬은 feature간 분산 정도에 대한 정보를 가지고 있음
- PCA는 정보를 많이 보유하고 있는 순으로 주성분 축을 찾는 작업
- feature간 공분산 행렬을 사용하면 feature간의 분산정도(정보)를 반영하여 축을 찾을 수 있음
- 고유값 분해는 정방 행렬만 가능

분산

$$\frac{X^T X}{n} = \frac{1}{n}\begin{pmatrix} dot(X_1, X_1) & dot(X_1, X_2) & \cdots & dot(X_1, X_d) \\ dot(X_2, X_1) & dot(X_2, X_2) & \cdots & dot(X_2, X_d) \\ \vdots & \vdots & \ddots & \vdots \\ dot(X_d, X_1) & dot(X_d, X_2) & \cdots & dot(X_d, X_d) \end{pmatrix}$$

공분산

4 고유값 분해 (Eigenvalue Decomposition)

고유값, 고유벡터 의미

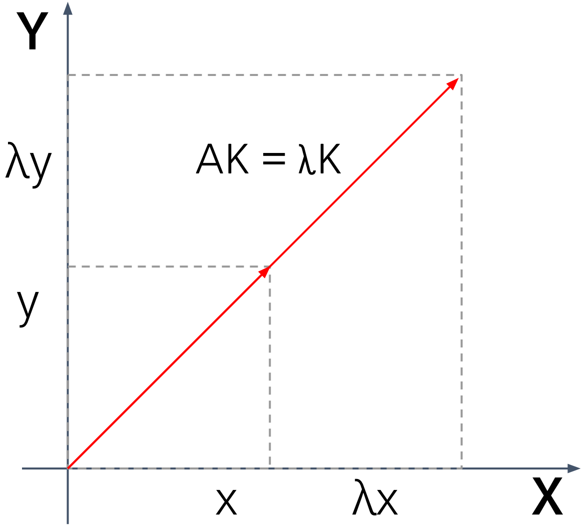
- 정방행렬 A : N x N
- AK = λK
  - K : 고유벡터(eigenvector)
  - λ : 고유값(eigenvalue)
- 임의의 정방행렬 A에 대해 고유값, 고유벡터를 찾아내는 것을 고유값 분해라고 함

$$AE = E\Lambda$$

$$A = E\Lambda E^{-1}$$

고유값, 고유벡터 의미

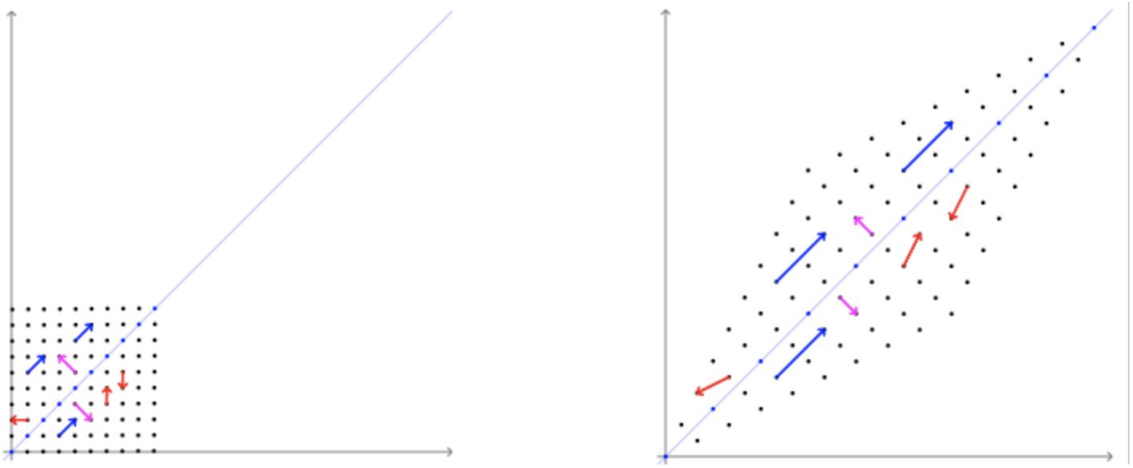
- 고유값, 고유벡터의 의미
  - A를 선형변환 행렬로 보았을 때 그 안에서 크기(scale)은 변하지만 방향을 유지되는 벡터(eigenvector)가 존재하는가



https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors

고유값, 고유벡터 의미

- 고유값, 고유벡터의 의미
  - A를 선형변환 행렬로 보았을 때 그 안에서 크기(scale)은 변하지만 방향을 유지되는 벡터 (eigenvector)가 존재하는가 => 주성분 축을 찾는다

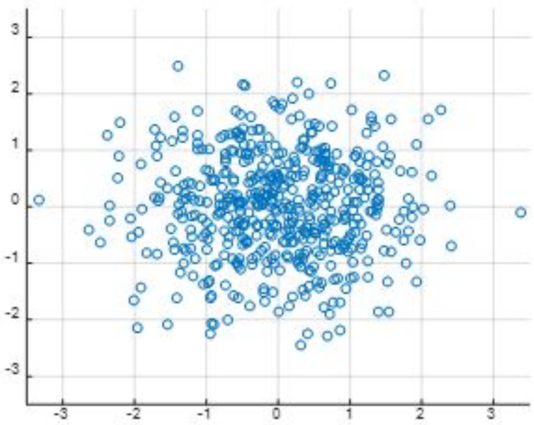


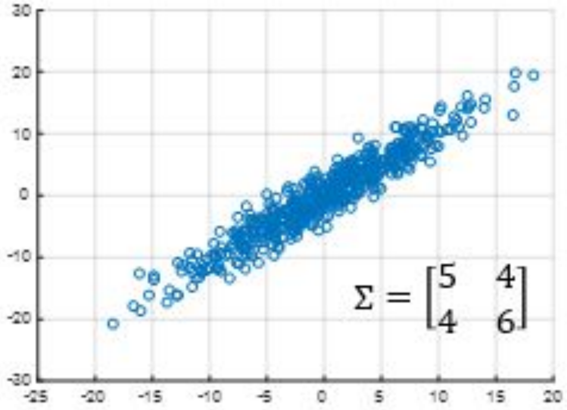
https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors

$$AK = \lambda K \cdots (1)$$

고유값, 고유벡터 의미

- 고유값, 고유벡터의 의미
  - A를 선형변환 행렬로 보았을 때 그 안에서 크기(scale)은 변하지만 방향을 유지되는 벡터 (eigenvector)가 존재하는가 => 주성분 축을 찾는다





https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors

고유벡터 활용

- 행렬 분해
    - 고유값 분해 (Eigendecomposition)
    - 특이값 분해 (SVD, Singular Value Decomposition)
- PCA (Principal Component Analysis)
    - 차원축소
- 그래프 분석
    - 그래프 영향도 측정 : Google PageRank
- 신호처리
    - 영상처리 : Eigenface

5

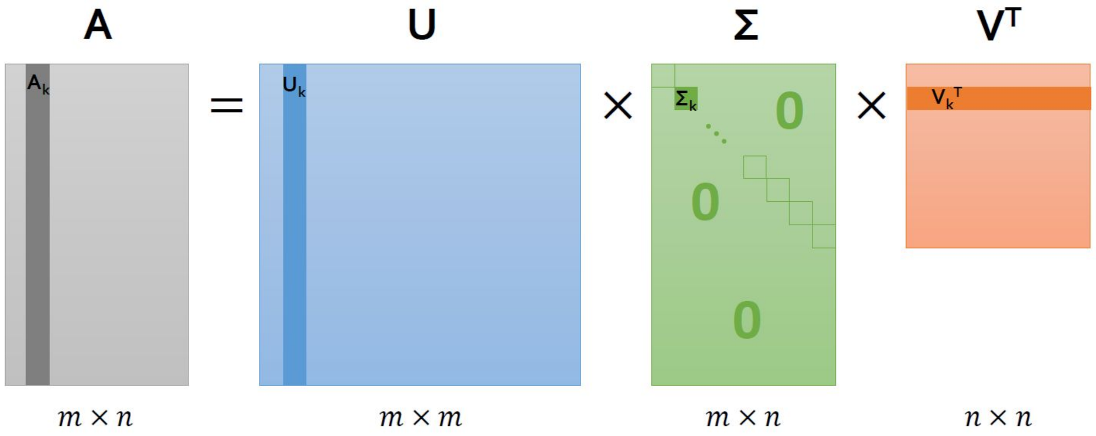
특이값 분해 (SVD)

Singular Value Decomposition

특이값 분해 (SVD) (1)

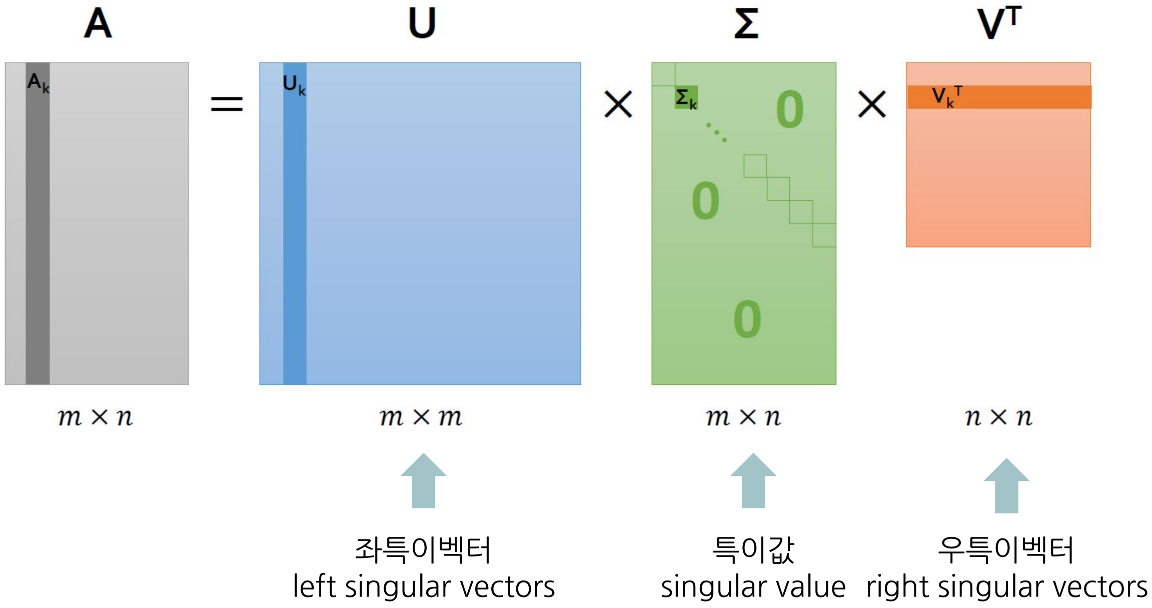
- 특이값 분해(Singular Value Decomposition, SVD)는 임의의 m×n차원의 행렬 A에 대하여 다음과 같이 행렬을 분해(decomposition)
- 고유값 분해가 정방 행렬에만 적용가능한데 비교하려 비정방형 행렬에도 적용이 가능

$$A_k = U_k \Sigma_k V_k^T$$



특이값 분해 (SVD) (2)

$$A_k = U_k \Sigma_k V_k^T$$



특이값 분해 (SVD) (2)

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

$$U : m \times m$$

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

$$V : n \times n$$

$$\Sigma : m \times n$$

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

특이값 분해 (SVD) (3)

$$A_k = U_k \Sigma_k V_k^T$$

$$A = U\Sigma V^T = \begin{pmatrix} | & | & & | \\ \vec{u}_1 & \vec{u}_2 & \cdots & \vec{u}_n \\ | & | & & | \end{pmatrix} \begin{pmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & 0 \\ & & \ddots & 0 \\ & & \sigma_m & 0 \end{pmatrix} \begin{pmatrix} - & \vec{v}_1^T & - \\ - & \vec{v}_2^T & - \\ & \vdots & \\ - & \vec{v}_n^T & - \end{pmatrix}$$

$$= \sigma_1 \vec{u}_1 \vec{v}_1^T + \sigma_2 \vec{u}_2 \vec{v}_2^T + \cdots + \sigma_m \vec{u}_m \vec{v}_m^T$$

좌특이벡터
left singular vectors

특이값
singular value

우특이벡터
right singular vectors

특이값 분해 (SVD) (4)

- 특이값 분해(Singular Value Decomposition, SVD)는 임의의 m×n차원의 행렬 A에 대하여 다음과 같이 행렬을 분해(decomposition)
- 고유값 분해가 정방 행렬에만 적용가능한데 비교하려 비정방형 행렬에도 적용이 가능

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

여기서 각 3개의 행렬은 다음과 같은 조건을 만족합니다.

$U : m \times m$ 직교행렬 ($AA^{\mathrm{T}} = U(\Sigma\Sigma^{\mathrm{T}})U^{\mathrm{T}}$)

$V : n \times n$ 직교행렬 ($A^{\mathrm{T}}A = V(\Sigma^{\mathrm{T}}\Sigma)V^{\mathrm{T}}$)

$\Sigma : m \times n$ 직사각 대각행렬

직교행렬(Orthogonal matrix)

선형대수학에서, **직교 행렬(直交行列, orthogonal matrix)은 행벡터와 열벡터가 유클리드 공간의 정규 직교 기저를 이루는 실수 행렬**이다.

$$Q^TQ = QQ^T = I$$

$$Q^{-1} = Q^T$$

$$Q^TQ=I \rightarrow \underbrace{\begin{bmatrix} - & \boldsymbol{q}_1{}^T & - \\ & \vdots & \\ - & \boldsymbol{q}_n{}^T & - \end{bmatrix}}_{Q^T} \underbrace{\begin{bmatrix} | & & | \\ \boldsymbol{q}_1 & \cdots & \boldsymbol{q}_n \\ | & & | \end{bmatrix}}_{Q} = \underbrace{\begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}}_{I}$$

대각행렬 (Diagonal matrix)

선형대수학에서, **대각행렬(對角行列, diagonal matrix)은 주대각선을 제외한 곳의 원소가 모두 0인 정사각행렬**이다. 대각행렬은 0일 수도, 아닐 수도 있는 대각원소들에 의해 결정된다. $n \times n$ 행렬 $D=[d_{i,j}]$가 대각행렬일 필요충분조건은

$$\text{임의의 } i, j \in \{1, 2, \cdots, n\} \text{ 에 대해, } i \neq j,\ d_{i,j} = 0$$

$$\begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{bmatrix} \quad \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & a & 0 & 0 \\ 0 & 0 & a & 0 \end{bmatrix} \quad \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \\ 0 & 0 & 0 \end{bmatrix}$$

특이값 분해 (SVD) (6)

$$
\begin{aligned}
&AA^T \\
&= \left(U\Sigma V^T\right)\left(U\Sigma V^T\right)^T \\
&= \left(U\Sigma V^T\right)\left(V\Sigma^T U^T\right) \quad V^T V = I \\
&= U\left(\Sigma\Sigma^T\right)U^T \\
&= U\Lambda U^T
\end{aligned}
$$

$$
\begin{aligned}
&A^T A \\
&= \left(U\Sigma V^T\right)^T\left(U\Sigma V^T\right) \\
&= \left(V\Sigma^T U^T\right)\left(U\Sigma V^T\right) \quad U^T U = I \\
&= V\left(\Sigma\Sigma^T\right)V^T \\
&= V\Lambda V^T
\end{aligned}
$$

$$\sigma_i = \sqrt{\lambda_i} \qquad \Sigma\Sigma^T = \Sigma^T\Sigma = \Lambda$$

특이값 분해 (SVD) (7)

$$A_k = U_k \Sigma_k V_k^T$$

$A = \begin{pmatrix} 3 & 6 \\ 2 & 3 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$ 일 때,

A의 특이값 분해 (Singular Value Decomposition)은

$$A = U\Sigma V^T = \begin{pmatrix} 0.881 & -0.471 & 0 & 0 \\ 0.471 & 0.881 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 7.605 & 0 \\ 0 & 0.394 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0.471 & -0.881 \\ 0.881 & 0.471 \end{pmatrix}^T$$

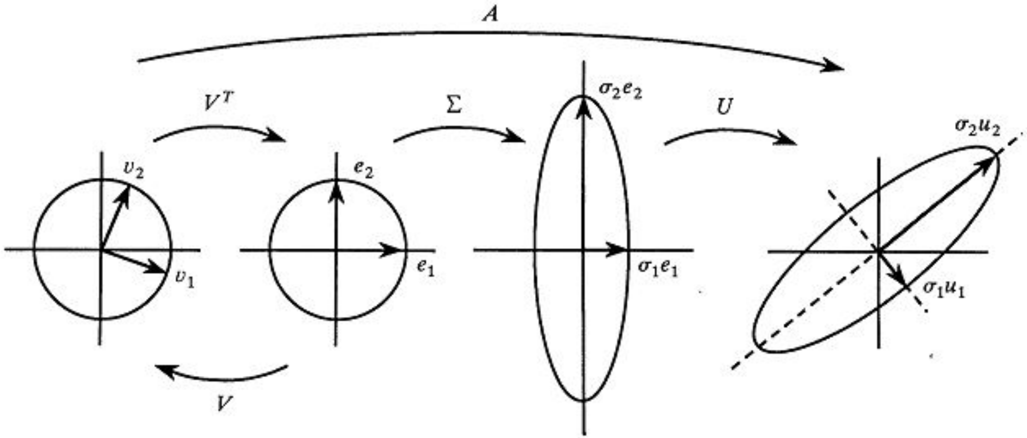
U ↑ AA^T 의 고유벡터 (eigenvectors of AA^T)

Σ ↑ $\sigma_i = \sqrt{\lambda_i}$ ($\lambda_1 \geq \lambda_2 ... \geq 0$이며, 대각원소이외 모두0)

V^T ↑ $A^T A$ 의 고유벡터 (eigenvectors of $A^T A$)

특이값 분해 기하학적 의미

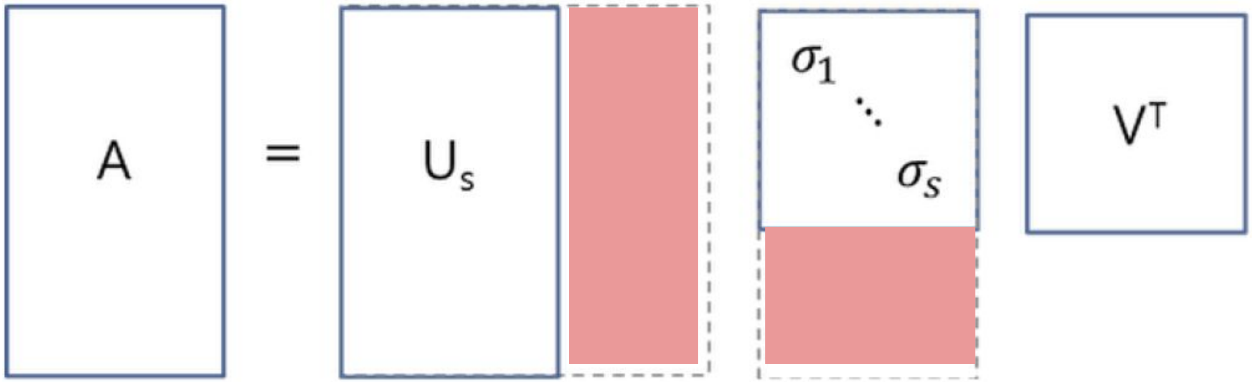
- $A = U\Sigma V^T$에서 U, V는 직교행렬, Σ는 대각행렬이므로 A_x는 x를 먼저 V_T에 의해 회전시킨 후 Σ로 스케일을 변화시키고 다시 U로 회전시키는 것



https://en.wikipedia.org/wiki/Singular_value_decomposition

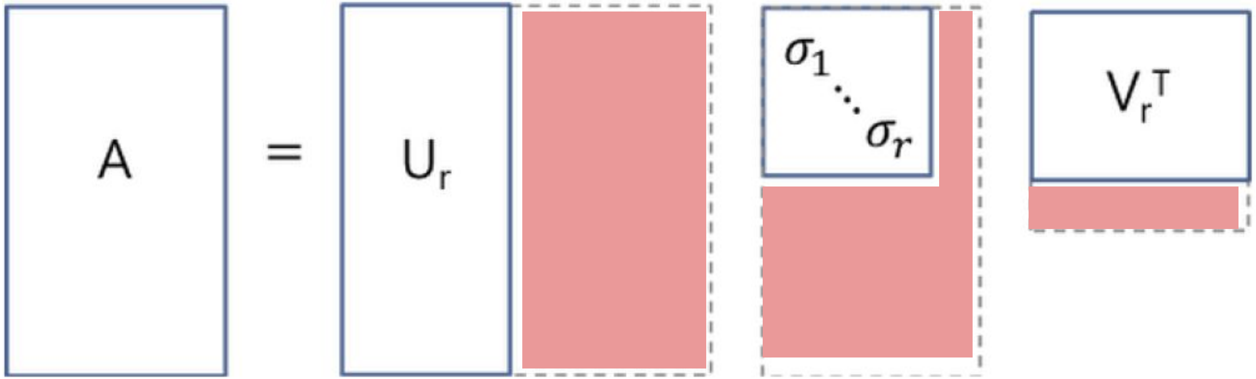
Thin SVD

- Σ 행렬 아랫부분 0으로 채워진 영역(비대각 파트)을 제거.
- <u>원복이 가능</u>



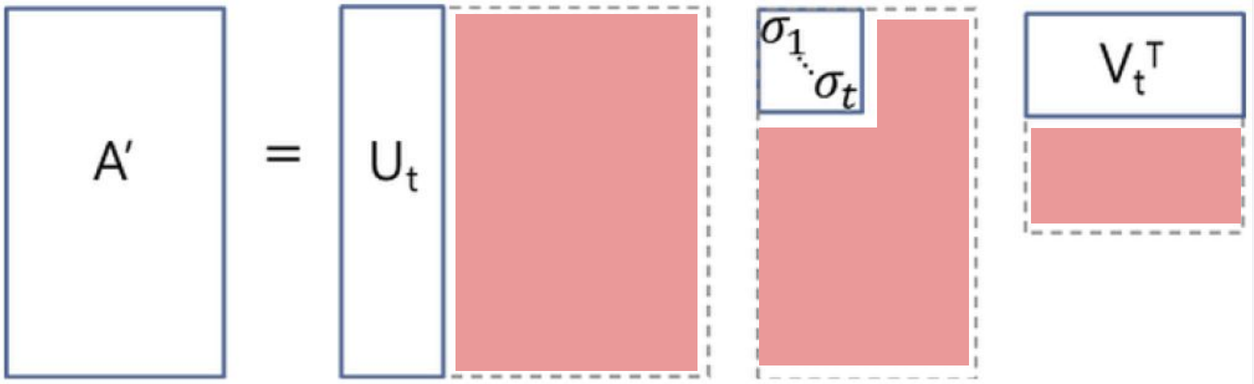
Compact SVD

- Σ 행렬에서 비대각파트 뿐만 아니라 특이값 중에서도 0인 부분을 제거한 형태
- 특이값(대각파트)이 0 초과인 값만 남겨 두고 제거
- <u>원복이 가능</u>



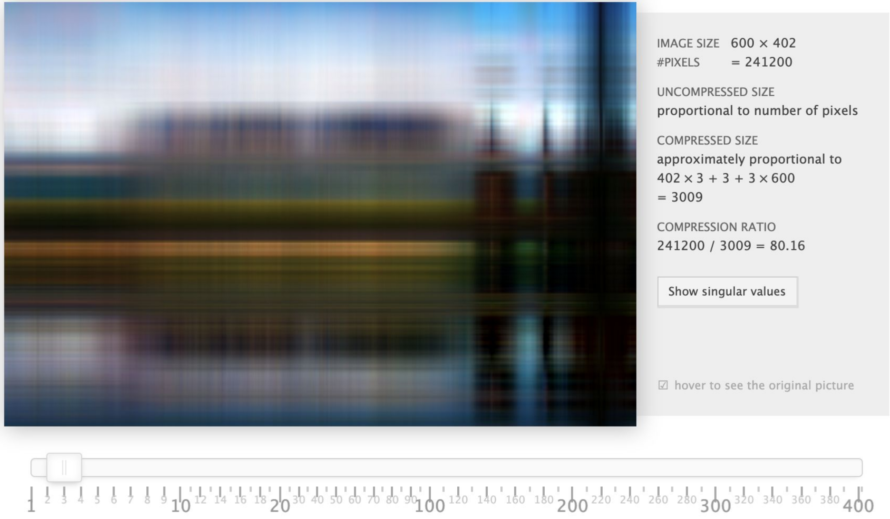
Truncated SVD

- Σ 행렬의 특이값 중 상위 n개를 선택하여 나머지를 제거한 형태
- 원복이 불가능 (= 정보 손실 발행 = 정보 압축)
- 정보 압축(=정보 손실)을 했음에도 원 행렬에 대한 근사가 가능. (LSA에서 사용)



특이값 분해 활용 예시

Image Compression with Singular Value Decomposition



http://timbaumann.info/svd-image-compression-demo/

감사합니다.

Insight campus

SeSAC

daniel@fins.ai