

기초 선형대수

(Spectral Property – 고유값/고유벡터/행렬의 인수분해)

윤정훈

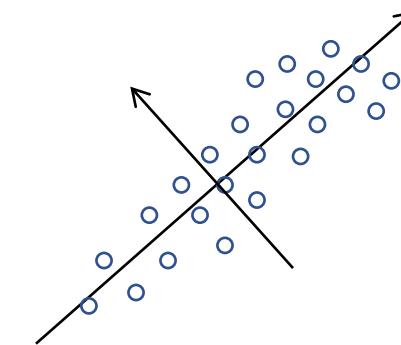
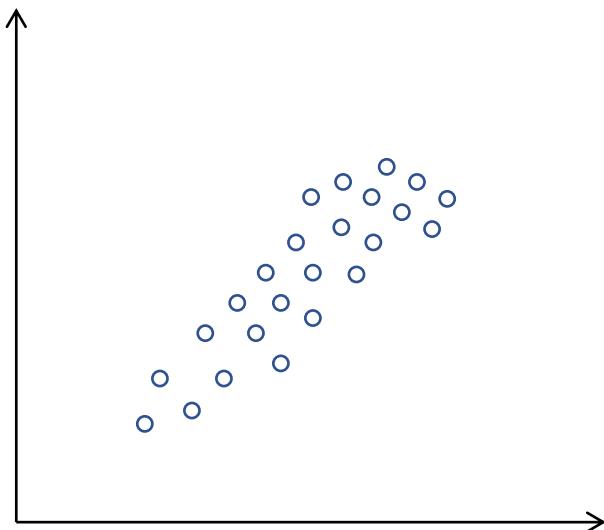
들어가며

Question 1)

데이터 집합이 다음과 같다.

어떤 것이 더 나은 표현인가?

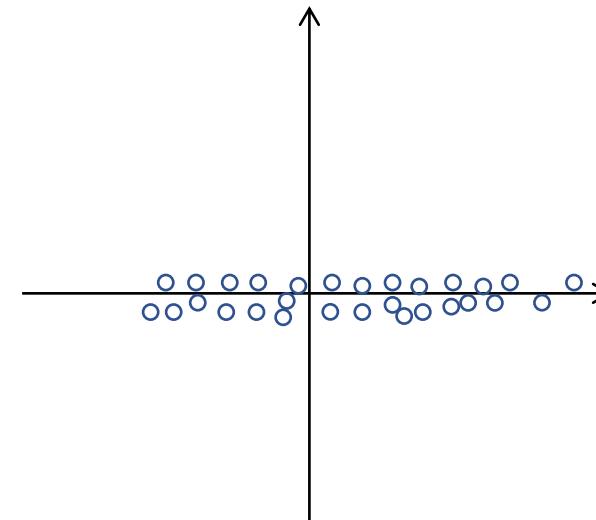
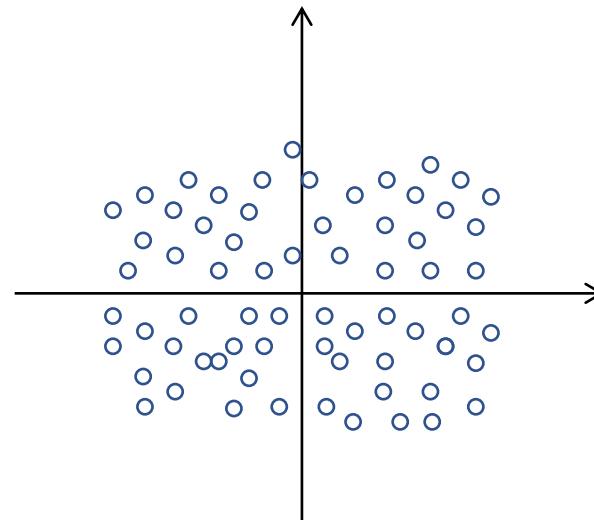
더 나은 표현이라는 것은 무슨 의미인가?



들어가며

Question 2)

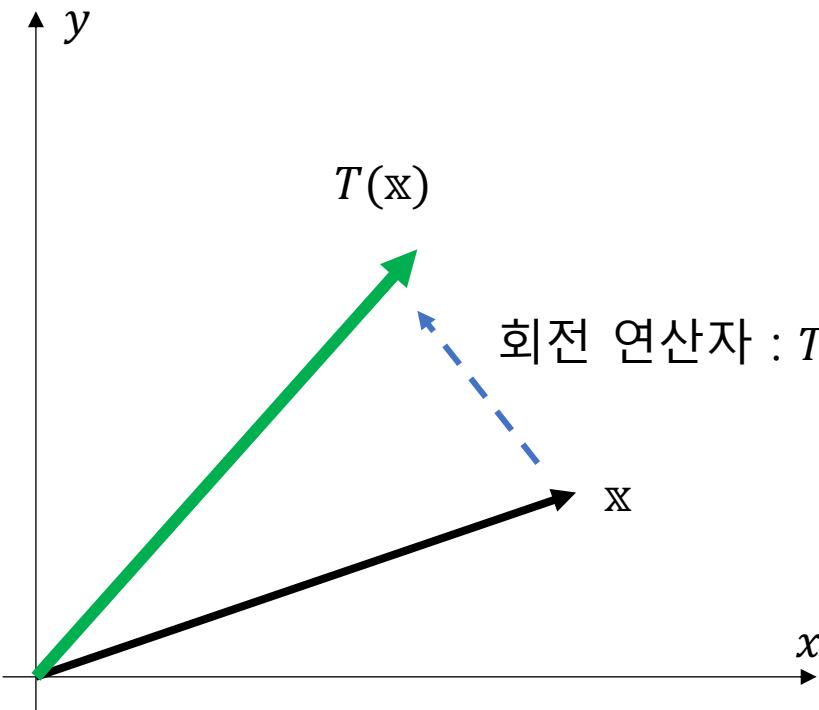
데이터 집합을 표현할 때, x 축 y 축이 모두 필요한가?



다양한 선형변환

- $R^n \rightarrow R^m(R^n)$ 으로의 대표적인 선형변환
 - 회전변환
 - 반사변환
 - 사영변환

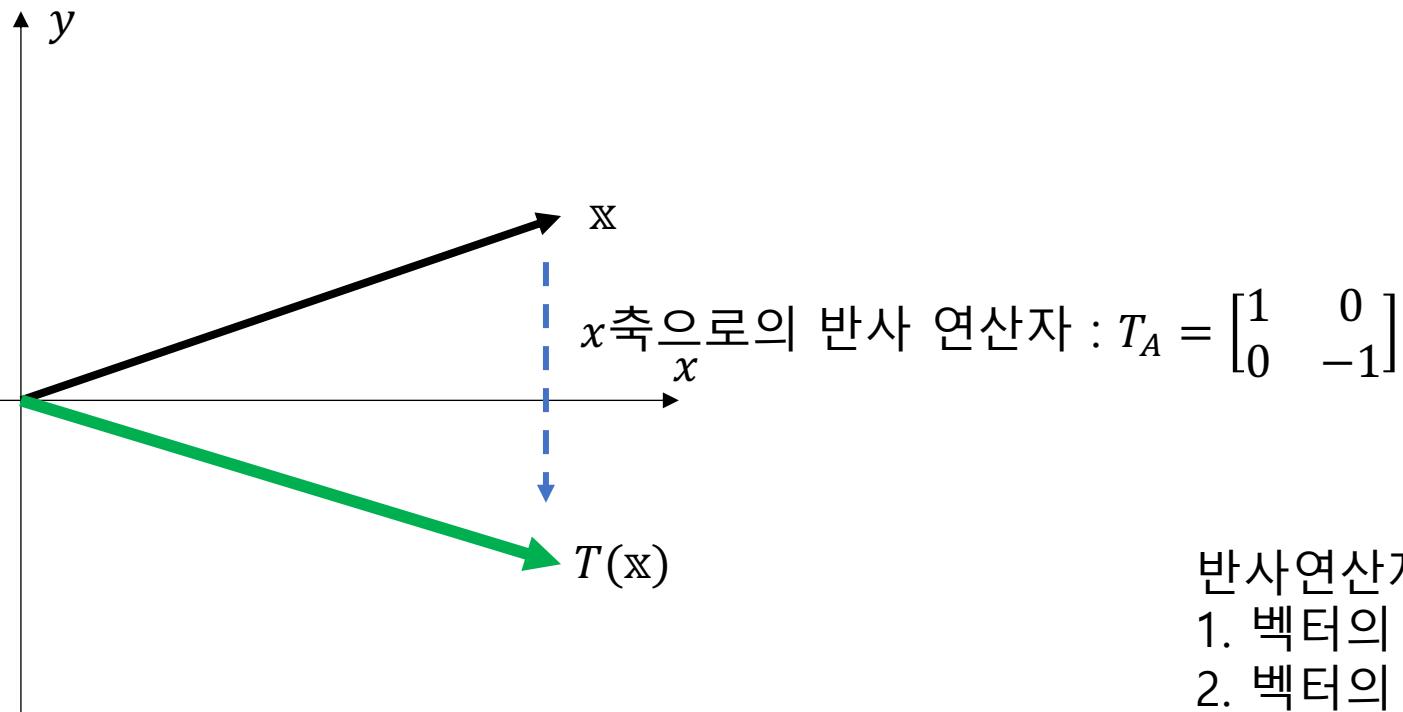
$R^2 \rightarrow R^2$ 의 회전변환



회전 연산자 : $T_A = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$

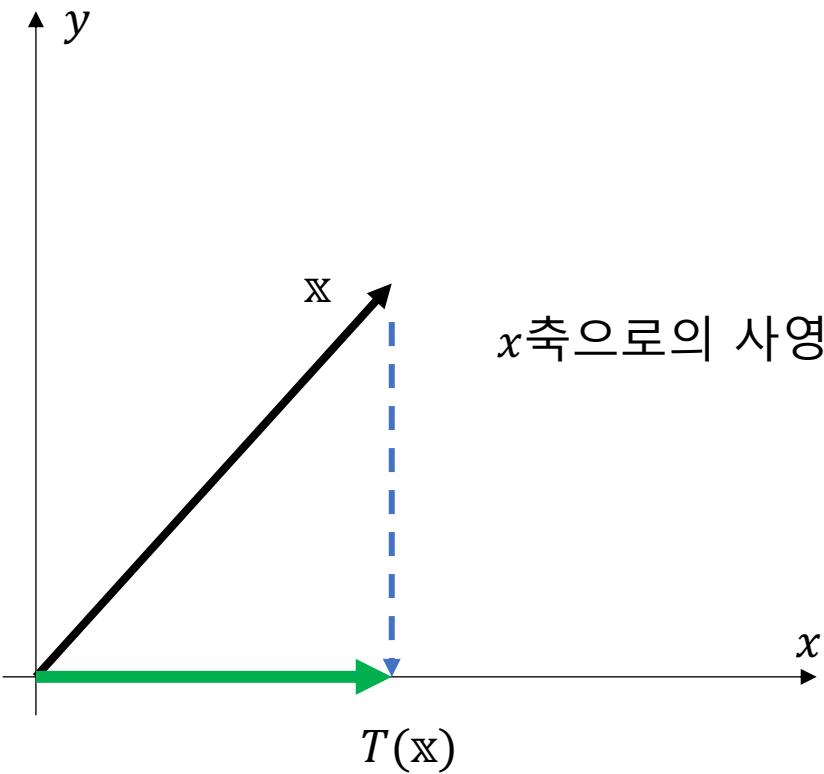
- 회전연산자의 특징
1. 벡터의 방향이 변한다.
 2. 벡터의 길이는 변하지 않는다.

$R^2 \rightarrow R^2$ 의 반사변환



- 반사연산자의 특징
1. 벡터의 방향이 변한다.
 2. 벡터의 길이는 변하지 않는다.

$R^2 \rightarrow R^2$ 의 사영변환



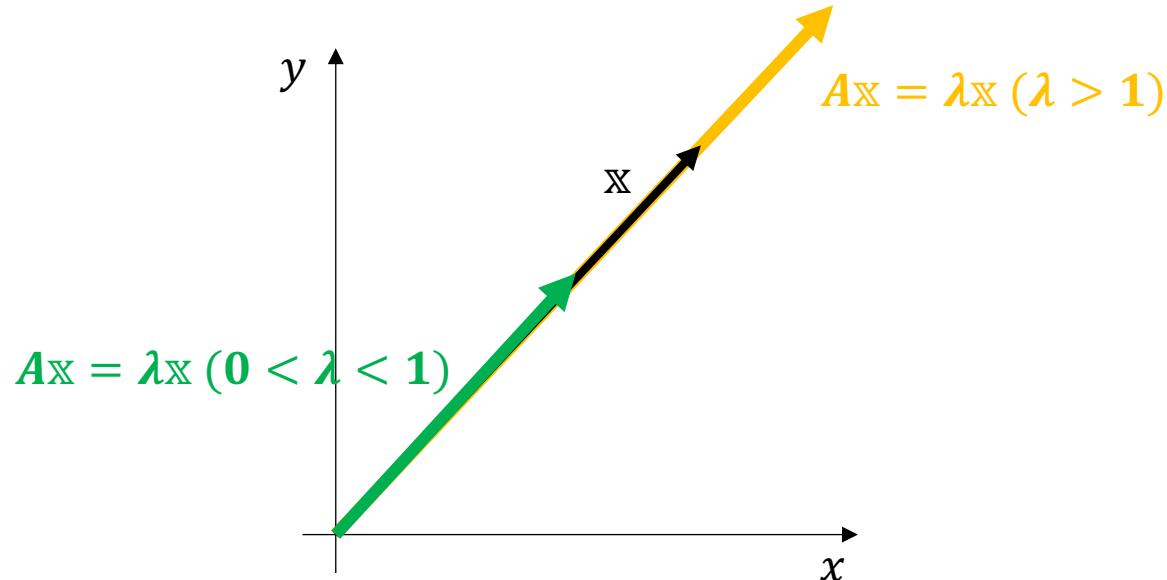
x 축으로의 사영연산자 : $T_A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$

- 회전연산자의 특징
1. 벡터의 방향이 변한다.
 2. 벡터의 길이도 변하다.

그러면 선형변환의 결과로 벡터의 방향은 바뀌지 않고, 벡터의 길이가 바뀌는 선형변환은 무엇이 있을까?

고유값과 고유벡터

- 정방행렬 A 에 대하여, 스칼라(scala)인 λ 와 영이 아닌 벡터 v 에 대해 $Av = \lambda v$ 를 만족하는 경우, λ 는 A 의 고유값(eigenvalue), v 는 대응하는 고유벡터(eigenvector)라고 한다.
- λ 가 행렬 A 의 고유값이면, 대응하는 고유벡터는 무수히 많다.
- 집합 $\{v : Av = \lambda v\}$ 는 벡터공간이며 고유값 λ 에 대응하는 고유공간(eigenspace)이라 한다.



고유값과 고유벡터 예제

- $A = \begin{bmatrix} 4 & 0 & 1 \\ 2 & 3 & 2 \\ 1 & 0 & 4 \end{bmatrix}$ 의 고유값 중 하나는 $\lambda = 5$ 이고, 이에 대응하는 고유벡터 $\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$ 이다.
- $A = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & 1 & 2 \end{bmatrix}$ 의 고유값 중 하나는 $\lambda = 0$ 이고, 이에 대응하는 고유벡터 $\mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ 이다.
- A 를 대각행렬이라고 하자. $A = diag(\lambda_1, \dots, \lambda_n)$ 의 고유벡터와 고유값은? 표준 기저 벡터 $\mathbf{e}_1, \dots, \mathbf{e}_n$ 는 고유벡터이고, 대각원소인 $\lambda_1, \dots, \lambda_n$ 은 고유값이다.

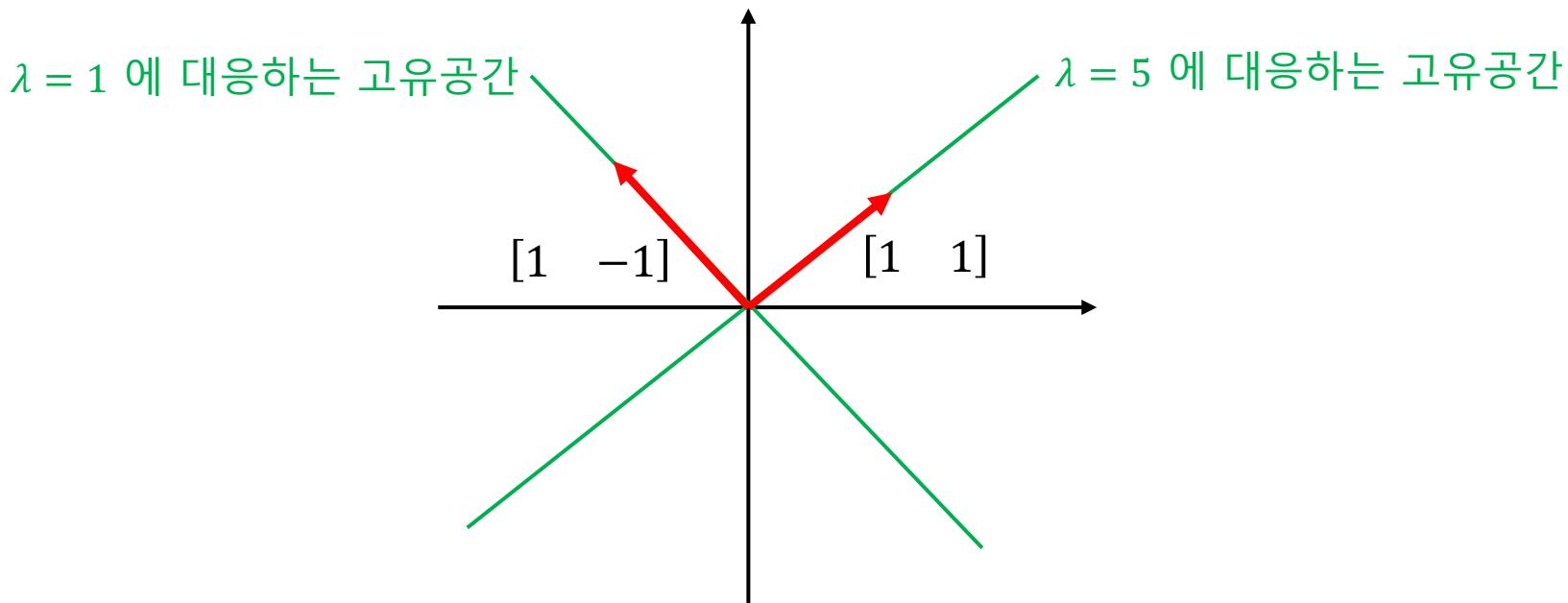
고유값과 고유벡터의 계산

행렬 A 의 고유값을 λ , 대응하는 고유벡터를 v 라고 하자.

- $Av = \lambda v \rightarrow (A - \lambda I_n)v = \emptyset$ 즉, 특성방정식 $\det(A - \lambda I_n) = 0$ 을 만족시키는 λ 를 구하면 된다.
 - 2×2 의 경우는 $\det(A - \lambda I_2) = \lambda^2 - \text{tr}(A)\lambda + \det(A) = 0$ 을 만족시켜주는 λ 를 구하면 된다.
- 행렬 $A = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}$ 의 고유값과 그에 대응하는 고유벡터?

고유벡터 예제

- 행렬 $A = \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix}$ 의 고유값과, 그에 대응하는 고유벡터를 구하여라. 그리고 고유공간을 그래프로 그려라.
 - $tr(A) = 6$, $\det(A) = 5$ 이므로 $\lambda^2 - 6\lambda + 5 = (\lambda - 1)(\lambda - 5) = 0$ 을 만족시키는 λ 의 값은 1, 5이다. (대칭행렬이기 때문에 고유벡터는 orthogonal)



유사성

- 역행렬이 존재하는 행렬을 가역행렬이라고 한다.
- 가역행렬 S 에 대해 $S^{-1}AS = B$ 가 만족되면 두 정방행렬 A 와 B 는 "유사 혹은 닮은(similar) 행렬"이라고 한다.
- 유사행렬(similar matrix)들은 동일한 고유값을 가진다.

대각화 가능성

- 만약 어떤 정방행렬 A 가 대각행렬 D 와 유사행렬이면, 즉 대각행렬 D 에 대해 $S^{-1}AS = D$ 를 만족하는 가역행렬 S 가 있으면, A 는 "대각화 가능하다(diagonalizable)" 라고 한다.
- $n \times n$ 행렬 A 가 대각화 가능 $\Leftrightarrow A$ 가 n 개의 일차독립인 고유벡터를 갖는 것
- n 개의 일차독립인 고유벡터를 갖는 $n \times n$ 행렬 A 를 대각화하는 법
 - A 의 n 개의 일차독립인 고유벡터 s_1, \dots, s_n 을 구한다.
 - 행렬 $S = [s_1 \quad \cdots \quad s_n]$ 을 구성한다.
 - 행렬 $S^{-1}AS$ 는 s_1, \dots, s_n 에 각각 대응하는 고유값을 대각성분으로 하는 대각행렬이 될 것이다.

대각화 가능성 예제

행렬 $A = \begin{bmatrix} 0 & 0 & -2 \\ 1 & 2 & 1 \\ 1 & 0 & 3 \end{bmatrix}$ 은 고유값 $\lambda = 1, \lambda = 2$ 를 갖는다.

$\lambda = 1$ 에 대응하는 고유공간의 기저벡터는 $s_1 = \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix}$

$\lambda = 2$ 에 대응하는 고유공간의 기저벡터는 $s_2 = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}, s_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ 이다.

3개의 기저벡터는 일차독립이므로 A 는 대각화 가능하고 $S = \begin{bmatrix} -2 & -1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$

$S^{-1}AS = \begin{bmatrix} -1 & 0 & -1 \\ 1 & 0 & 2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -2 \\ 1 & 2 & 1 \\ 1 & 0 & 3 \end{bmatrix} \begin{bmatrix} -2 & -1 & 0 \\ 1 & 0 & 2 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} = D$ 로 대각화 된다.

대각화 가능과 고유값 사이의 관계

- 먼저, 대각행렬 $D = diag(\lambda_1, \dots, \lambda_n)$ 의 행렬의 고유값은 $\lambda_1, \dots, \lambda_n$ 이다. 행렬 A 와 D 가 유사행렬이이라고 하자. 즉, 가역행렬 S 에 대하여, $S^{-1}AS = D$ 라고 하자. 그리고 D 가 대각행렬이므로, A 는 대각화 가능하다. A 의 고유값은 D 의 고유값, 즉 D 의 대각원소들이다. 그리고 S 의 열들은 일차독립이며 A 의 고유벡터들이 된다.

직교대각화 가능

- 정사각행렬 S 에 대하여, $S^T = S^{-1}$ 인 경우, A 를 **직교행렬**이라고 한다.
- 정사각행렬 A 에 대해서, $D = S^T AS$ 를 만족하는 대각행렬 D 와, 직교행렬 S 가 존재하면 A 는 **직교대각화 가능**(orthogonally diagonalizable) 하다고 한다.
- 정사각행렬 A 가 직교대각화 가능 \Leftrightarrow 정사각행렬 A 가 대칭행렬

행렬의 인수분해

정수는 인수분해된다. 예를들어 $12 = 3 \times 4$ 처럼, 12는 3과 4로 인수분해된다.

마찬가지로 행렬도 $A = BC$ 처럼 인수분해 될 수 있다.

예를들어, $\begin{bmatrix} 2 & 6 & 2 \\ -3 & -8 & 0 \\ 4 & 9 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ -3 & 1 & 0 \\ 4 & -3 & 7 \end{bmatrix} \begin{bmatrix} 1 & 3 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}$ 이렇게 인수분해 될 수 있다.

행렬 A 를 인수분해하여, 행렬 A 와 가장 비슷하고(similar), 크기가 작은 행렬을 A 대신 사용하려고 한다.

여러가지 행렬의 인수분해

- LU 분해 (LU Decomposition)
- 스펙트럼 분해 (Spectrum Decomposition) or 고유값 분해
- 특이값 분해 (Singular Value Decomposition)

LU 분해

- LU분해는 행렬을 아래의 두 행렬로 인수분해한다.
 - 하부삼각행렬(행렬의 대각선의 윗부분이 모두 0)
 - 상부삼각행렬(행렬의 대각선의 아랫부분이 모두 0)

$$A = \begin{matrix} & \text{하부삼각행렬} \\ \begin{pmatrix} & & & \\ & 0 & & \\ & & 0 & \\ & & & 0 \end{pmatrix} & \times \\ & (\text{Lower Triangular Matrix}) & \end{matrix} \quad \begin{matrix} & \text{상부삼각행렬} \\ \begin{pmatrix} & & & 0 \\ & & & & \\ & & & & \\ & & & & \end{pmatrix} & \times \\ & (\text{Upper Triangular Matrix}) & \end{matrix}$$

- LU분해를 통하여 Linear Systems 문제를 간단하게 해결할 수 있다.

LU 분해 예제

- $\begin{bmatrix} 2 & 6 & 2 \\ -3 & -8 & 0 \\ 4 & 9 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix}$ 를 쉽게 풀어보자!

하부삼각행렬 상부삼각행렬

$$\begin{bmatrix} 2 & 6 & 2 \\ -3 & -8 & 0 \\ 4 & 9 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ -3 & 1 & 0 \\ 4 & -3 & 7 \end{bmatrix} \begin{bmatrix} 1 & 3 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$
 처럼 삼각행렬의 곱으로 이루어진다.

$$\begin{bmatrix} 2 & 0 & 0 \\ -3 & 1 & 0 \\ 4 & -3 & 7 \end{bmatrix} \begin{bmatrix} 1 & 3 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix}$$
 처럼 풀고자하는 식을 다시 쓸 수 있다.

먼저, $\begin{bmatrix} 1 & 3 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$ 으로 정의하자.

LU 분해 예제

그러면 우리가 풀고자 하는 식은, $\begin{bmatrix} 2 & 0 & 0 \\ -3 & 1 & 0 \\ 4 & -3 & 7 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix}$ 으로 다시 쓸 수 있다.

즉, $2y_1 = 2$, $-3y_1 + y_2 = 2$, $4y_1 - 3y_2 + 7y_3 = 3$ 을 풀면 된다. 즉 $y_1 = 1$, $y_2 = 5$, $y_3 = 2$ 가 된다.

식을 다시 써 보면, $\begin{bmatrix} 1 & 3 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \\ 2 \end{bmatrix}$ 가 된다.

즉, $x_1 + 3x_2 + x_3 = 1$, $x_2 + 3x_3 = 5$, $x_3 = 2$ 을 풀면 된다. 즉 $x_1 = 2$, $x_2 = -1$, $x_3 = 2$ 가 된다.

이 예제는 행렬 A 를 하부삼각행렬, 상부삼각행렬로 인수분해하면, 선형계 $A\mathbf{x} = \mathbf{b}$ 는 풀이과정이 쉬워진다는 것을 명확히 보여주고 있다.

LU 분해

- 정사각행렬 A 가 하부삼각행렬 L 과 상부삼각행렬 U 의 곱 $A = LU$ 로 되는 인수분해를 LU 분해 또는 LU 인수분해(LU factorization)라고 부른다.
- 일반적으로 모든 정사각행렬 A 가 LU 분해를 갖지 않으며, LU 분해가 존재하여도 LU 분해가 유일하지도 않다.

스펙트럼 분해

- 행렬 A 가 $S = [\mathbf{s}_1 \ \cdots \ \mathbf{s}_n]$ 로 직교대각화되는 대칭행렬이고, $\lambda_1, \dots, \lambda_n$ 은 각각 $\mathbf{s}_1, \dots, \mathbf{s}_n$ 에 대응하는 A 의 고유값이라고 하면, 행렬 $D = S^T A S$ 는 대각성분이 A 의 고유값인 대각행렬이 된다.

$$\bullet A = SDS^T = [\mathbf{s}_1 \ \cdots \ \mathbf{s}_n] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix} \begin{bmatrix} \mathbf{s}_1^T \\ \vdots \\ \mathbf{s}_n^T \end{bmatrix} = \lambda_1 \mathbf{s}_1 \mathbf{s}_1^T + \cdots + \lambda_n \mathbf{s}_n \mathbf{s}_n^T \text{ 와 같이 표현 가능하다.}$$

- 이 공식을 A 의 스펙트럼 분해(spectral decomposition) 또는 A 의 고유값 분해(eigenvalue decomposition)라고 한다.

특이값 분해

- SVD는 모든(any) 행렬에 대하여 적용가능한 분해(정수와 비교하면 인수분해) 기법이기 때문에, 행렬의 분해에 가장 널리 사용되는 기법이다.
- $n \times n$ 행렬 A 가 대칭행렬이 아니면, 고유값 분해는 존재하지 않는다. 일반적인 경우에 대한 행렬의 인수분해를 위해서 특이값 분해(Singular Value Decomposition)를 사용한다.
- 디지털화된 정보의 압축, 저장, 전송에 활용

특이값 분해

- 행렬 A 가 계수 k 인 $m \times n$ 행렬일 때, $\textcolor{red}{A} = U\Sigma V^T$ 로 인수분해된다.

$$A = U\Sigma V^T = [\mathbb{U}_1 \quad \cdots \quad \mathbb{U}_k \quad | \quad \mathbb{U}_{k+1} \quad \cdots \quad \mathbb{U}_m] \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_k \\ \hline O_{(m-k) \times k} & & O_{(m-k) \times (n-k)} \end{bmatrix} \begin{bmatrix} \mathbb{V}_1^T \\ \vdots \\ \mathbb{V}_k^T \\ - \\ \mathbb{V}_{k+1}^T \\ \vdots \\ \mathbb{V}_n^T \end{bmatrix}$$

- $U : m \times m$, $\Sigma : m \times n$, $V : n \times n$
- $V = [\mathbb{V}_1 \quad \cdots \quad \mathbb{V}_n]$ 는 $A^T A$ 를 직교대각화한다. 즉, $\textcolor{red}{V}$ 의 컬럼은 $A^T A$ 의 고유벡터들이다. \mathbb{V}_i 는 orthonormal.
- $\lambda_1, \dots, \lambda_k$ 가 V 의 열벡터에 대응하는 $A^T A$ 의 영이 아닌 고유값일 때, Σ 의 영이 아닌 대각성분은 $\sigma_1 = \sqrt{\lambda_1}, \dots, \sigma_k = \sqrt{\lambda_k}$ 이다. 이 대각성분들을 **특이값(singular value)**이라고 한다. k 는 A 의 계수이다.
- V 의 열벡터는 $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0$ 을 만족시키도록 배열되어 있다.
- $\{\mathbb{U}_1, \dots, \mathbb{U}_k\}$ 는 $\text{col}(A)$ 의 정규직교기저이다. \mathbb{U}_i 는 orthonormal.

$A^T A$, AA^T

- $A^T A$ (임의의 행렬 A)
 - $A^T A = V\Sigma^T U^T U\Sigma V^T = V(\Sigma^T \Sigma)V^T = VDV^T$
 - $D = \Sigma^T \Sigma$ 이기 때문에 D 의 대각성분은 singular value의 제곱
 - $(A^T A)V = VDV^T V = VD$
 - 위의 식에 의하여 $A^T A$ 의 고유값은 V 의 대각성분들이 되고, V 의 columns은 $A^T A$ 의 고유벡터가 된다.
- AA^T (임의의 행렬 A)
 - $AA^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma^T \Sigma U^T = UDU^T$
 - $D = \Sigma^T \Sigma$ 이기 때문에 D 의 대각성분은 singular value의 제곱
 - $(AA^T)U = UDU^T U = UD$
 - 위의 식에 의하여 AA^T 의 고유값은 U 의 대각성분들이 되고, U 의 columns은 AA^T 의 고유벡터가 된다.
- $A^T A$ 의 eigenvector는 PCA에서의, A 의 공분산 행렬의 주성분이다.

특이값 분해 예제

- 행렬 $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$ 의 특이값분해를 구하라.

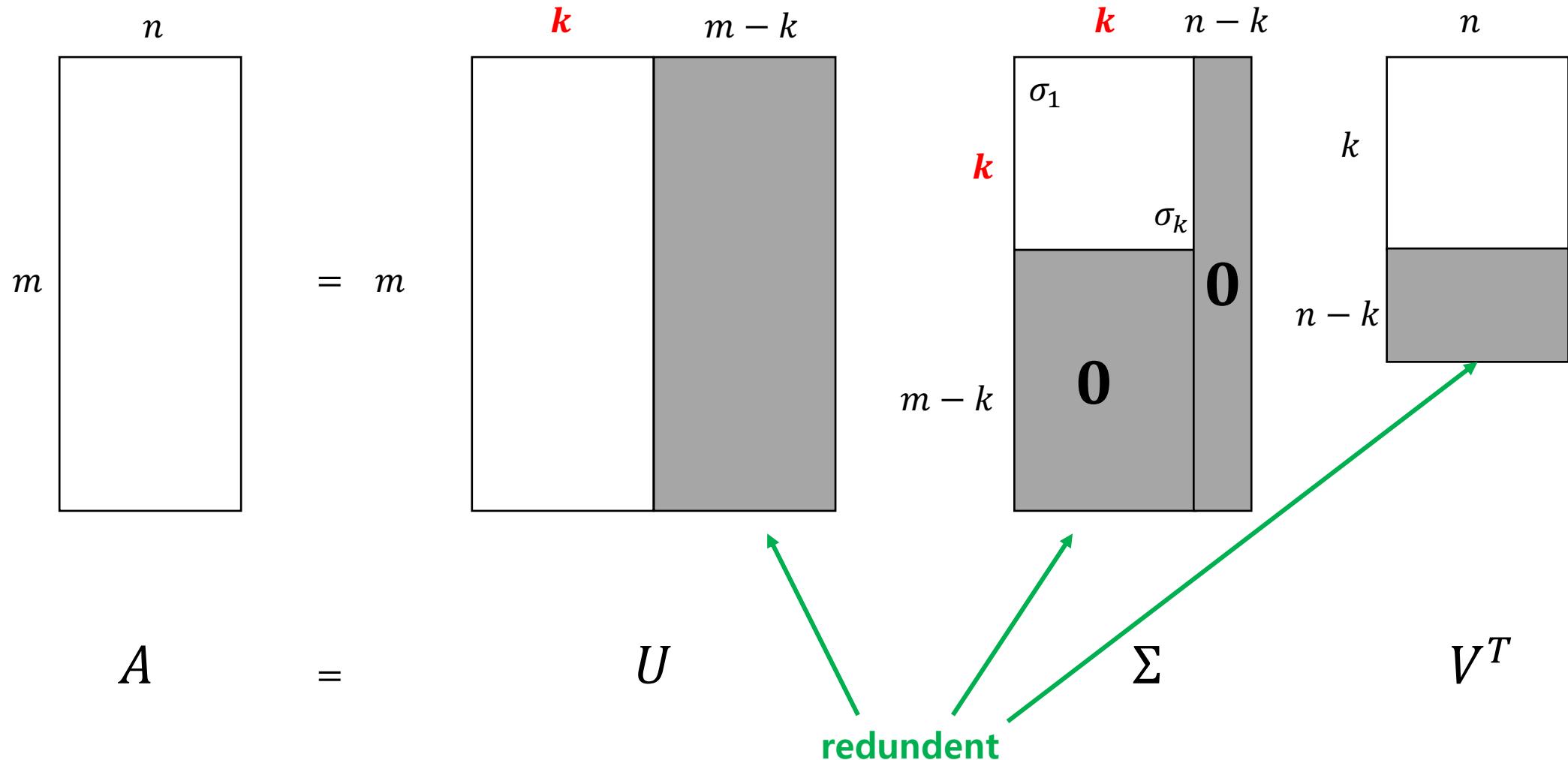
- $A^T A$ 의 고유값을 구한다. $\lambda_1 = 3, \lambda_2 = 1$
- 특이값을 구한다. $\sigma_1 = \sqrt{\lambda_1} = \sqrt{3}, \sigma_2 = \sqrt{\lambda_2} = \sqrt{1}$
- $A^T A$ 의 고유벡터를 구한다. (v_1, \dots, v_n)
- $\frac{1}{\sigma_i} A v_i = u_i$ 를 구한다. (u_1, \dots, u_m)

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

$$= U \Sigma V^T$$

$$= \begin{bmatrix} \frac{\sqrt{6}}{3} & 0 & -\frac{1}{\sqrt{3}} \\ \frac{\sqrt{6}}{6} & -\frac{\sqrt{2}}{2} & \frac{1}{\sqrt{3}} \\ \frac{\sqrt{6}}{6} & \frac{\sqrt{2}}{2} & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}$$

축소된 SVD



축소된 SVD

$$A = U_1 \Sigma_1 V_1^T$$

The diagram illustrates the decomposition of matrix A into three components. Matrix A is shown as a rectangle with dimensions m (height) by n (width). It is equated to the product of three matrices: U_1 , Σ_1 , and V_1^T . Matrix U_1 is m by k , where k is indicated by a red k above it. Matrix Σ_1 is a k by k diagonal matrix, represented by a rectangle containing the diagonal elements σ_1 and σ_k , with a red k to its left. Matrix V_1^T is n by k , where n is indicated by a red n above it.

축소된 SVD

- 영블록을 인수로 하는 곱을 없애보자.

$$A = U\Sigma V^T = [\mathbb{U}_1 \quad \cdots \quad \mathbb{U}_k \quad | \quad \mathbb{U}_{k+1} \quad \cdots \quad \mathbb{U}_m] \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_k \\ \hline O_{(m-k) \times k} & & O_{(m-k) \times (n-k)} \end{bmatrix} \begin{bmatrix} \mathbb{V}_1^T \\ \vdots \\ \mathbb{V}_k^T \\ - \\ \mathbb{V}_{k+1}^T \\ \vdots \\ \mathbb{V}_n^T \end{bmatrix}$$



$$A = U_1 \Sigma_1 V_1^T = [\mathbb{U}_1 \quad \cdots \quad \mathbb{U}_k] \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_k \end{bmatrix} \begin{bmatrix} \mathbb{V}_1^T \\ \vdots \\ \mathbb{V}_k^T \end{bmatrix}$$

- $U_1 : m \times k$, $\Sigma_1 : k \times k$, $V_1^T : k \times n$

축소된 SVD

- A 의 축소된 특이값 확장(reduced singular expansion)

$$\bullet A = U_1 \Sigma_1 V_1^T = [\mathbb{U}_1 \quad \cdots \quad \mathbb{U}_k] \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_k \end{bmatrix} \begin{bmatrix} \mathbb{V}_1^T \\ \vdots \\ \mathbb{V}_k^T \end{bmatrix} = \sigma_1 \mathbb{U}_1 \mathbb{V}_1^T + \cdots + \sigma_k \mathbb{U}_k \mathbb{V}_k^T$$

- 이 결과는 모든 행렬에 적용된다.

축소된 SVD 예제

- 행렬 $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$ 의 축소된 특이값분해를 구하라.

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

$$= U_1 \Sigma V_1^T$$

$$= \begin{bmatrix} \frac{\sqrt{6}}{3} & 0 \\ \frac{\sqrt{6}}{6} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{6}}{6} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}$$

LSA (Latent Semantic Analysis)

- LSA(LSI)는 document data의 숨겨진 의미(hidden concept)를 찾아내는 기법이다.
- LSA는 각각의 문서(document)와 단어(word)를 벡터로 표현한다. 벡터내에서 각각의 element는 **숨겨진 의미**가 될 것이다.

LSA (Latent Semantic Analysis)

d_1 : Romeo and Juliet.

d_2 : Juliet: O happy dagger!

d_3 : Romeo died by dagger.

d_4 : "Live free or die", that's the motto of New-Hampshire

d_5 : Did you know, New-Hampshire is in New-England

Query : dies and dagger

- 3번 문서는 쿼리에 대해서 1등이 될 것이다.
 - 2번, 4번 문서는 그 다음이 될 것이다.
 - 1번, 5번 문서는?
 - ✓ 사람들이 인식하기로는 문서 1번이 문서 5번 보다 주어진 쿼리에 더 맞는 문서이다.
- 컴퓨터도 이러한 추론 과정을 할 수 있을까? 즉 **숨겨진 의미**를 찾을 수 있을까?

LSA (Latent Semantic Analysis)

matrix A :

	live	die	free	new-hampshire
d_1	[1 1 0 0 0 0 0 0]			
d_2	[0 1 1 1 0 0 0 0]			
d_3	[1 0 0 1 0 1 0 0]			
d_4	[0 0 0 0 1 1 1 1]			
d_5	[0 0 0 0 0 0 0 1]			

Diagram illustrating the matrix A with rows labeled by documents d_1 through d_5 and columns labeled by words "live", "die", "free", and "new-hampshire". The matrix entries are binary values indicating the presence or absence of words in each document. Arrows point from the word labels to their corresponding columns in the matrix.

Document labels: d_1, d_2, d_3, d_4, d_5

Word labels: live, die, free, new-hampshire

Document words: romeo, juliet, happy, dagger

doc-doc matrix

1번 문서에는 romeo, juliet, 2번 문서에는 juliet, happy, dagger

즉 겹쳐지는 것이 1개이므로 $B[1,2] = B[2,1] = 1$

matrix A :

[[1 1 0 0 0 0 0 0]
[0 1 1 1 0 0 0 0]
[1 0 0 1 0 1 0 0]
[0 0 0 0 1 1 1 1]
[0 0 0 0 0 0 0 1]]

5×8

matrix A^T :

[[1 0 1 0 0]
[1 1 0 0 0]
[0 1 0 0 0]
[0 1 1 0 0]
[0 0 0 1 0]
[0 0 1 1 0]
[0 0 0 1 0]
[0 0 0 1 1]]

8×5

matrix $B = AA^T$ doc-doc matrix

문서 i 와 문서 j 가 b 개의 공통 단어를 가지고 있으면 $B[i,j] = b$

matrix $AA^T(B)$:

[[2, 1, 1, 0, 0],
[1, 3, 1, 0, 0],
[1, 1, 3, 1, 0],
[0, 0, 1, 4, 1],
[0, 0, 0, 1, 1]]

5×5

word-word matrix

juliet은 1번, 2번 문서에서 나오고, dagger는 2, 3번 문서에서 나온다.

즉 겹쳐지는 것이 1개이므로 $C[2,4] = B[4,2] = 1$

matrix $C = A^T A$ word-word matrix

즉, 단어 i 와 단어 j 가 c 개의 문서에서 함께 발생했으면 $C[i,j] = c$

matrix A^T :

[[1 0 1 0 0]
[1 1 0 0 0]
[0 1 0 0 0]
[0 1 1 0 0]
[0 0 0 1 0]
[0 0 1 1 0]
[0 0 0 1 0]
[0 0 0 1 1]]

8×5

matrix A :

[[1 1 0 0 0 0 0 0]
[0 1 1 1 0 0 0 0]
[1 0 0 1 0 1 0 0]
[0 0 0 0 1 1 1 1]
[0 0 0 0 0 0 0 1]]

5×8

matrix $A^T A(C)$:

[[2, 1, 0, 1, 0, 1, 0, 0] ,
[1, 2, 1, 1, 0, 0, 0, 0] ,
[0, 1, 1, 1, 0, 0, 0, 0] ,
[1, 1, 1, 2, 0, 1, 0, 0] ,
[0, 0, 0, 1, 1, 1, 1, 1] ,
[1, 0, 0, 1, 2, 1, 1, 1] ,
[0, 0, 0, 1, 1, 1, 1, 1] ,
[0, 0, 0, 0, 1, 1, 1, 2]]

8×8

LSA (Latent Semantic Analysis)

SVD 사용!

$A = U\Sigma V^T$, U 는 B 의 eigenvectors이고, V 는 C 의 eigenvectors이다.

$$\Sigma = \begin{bmatrix} 2.285 & 0 & 0 & 0 & 0 \\ 0 & 2.010 & 0 & 0 & 0 \\ 0 & 0 & 1.361 & 0 & 0 \\ 0 & 0 & 0 & 1.118 & 0 \\ 0 & 0 & 0 & 0 & 0.797 \end{bmatrix}$$



singular value

LSA (Latent Semantic Analysis)

Reduced SVD 사용!

$A_k = S_k \Sigma_k U_k^T$, 모든 singular value를 사용할 수 없고, 작은 것들은 제외한다.

k 개의 특이값만 남기는 것이다. 즉 k 개의 "hidden concepts"만 남긴다.

LSA (Latent Semantic Analysis)

$$\Sigma_2 = \begin{bmatrix} 2.285 & 0 \\ 0 & 2.010 \end{bmatrix}$$

$$V_2^T =$$

```
array([[ 0.39615277,  0.28005737],  
       [ 0.31426806,  0.44953214],  
       [ 0.17823952,  0.26899154],  
       [ 0.43836375,  0.36850831],  
       [ 0.26388058, -0.34592143],  
       [ 0.52400482, -0.24640466],  
       [ 0.26388058, -0.34592143],  
       [ 0.32637322, -0.45966878]])
```

romeo
juliet
happy
dagger
live
die
free
new-hampshire

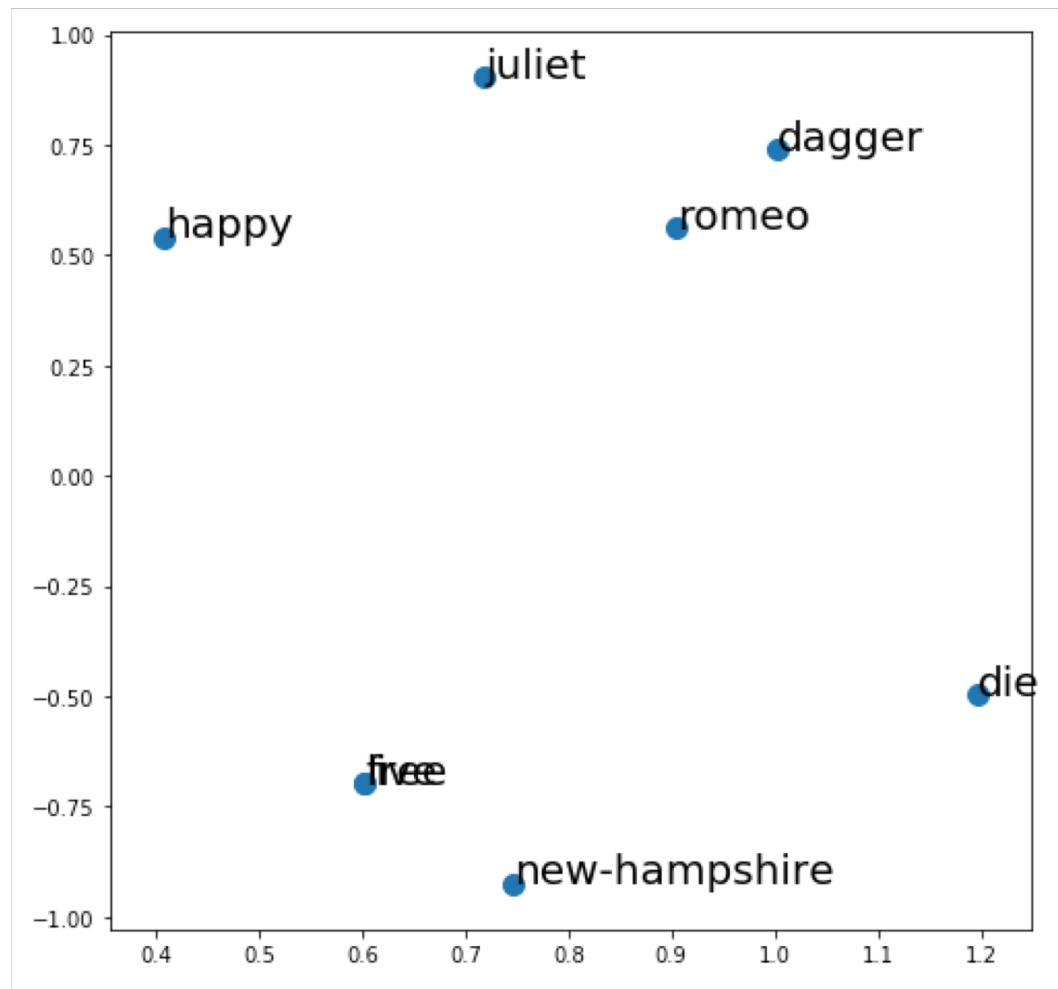
Word vector

$$\Sigma_2 V_2^T =$$

```
array([[ 0.90532712,  0.56298763],  
       [ 0.71819615,  0.90367568],  
       [ 0.40733041,  0.54074246],  
       [ 1.00179178,  0.74079687],  
       [ 0.60304575, -0.6953914 ],  
       [ 1.19750713, -0.49533699],  
       [ 0.60304575, -0.6953914 ],  
       [ 0.74586005, -0.92405295]])
```

romeo
juliet
happy
dagger
live
die
free
new-hampshire

LSA (Latent Semantic Analysis)



Word vector의 scatter

LSA (Latent Semantic Analysis)

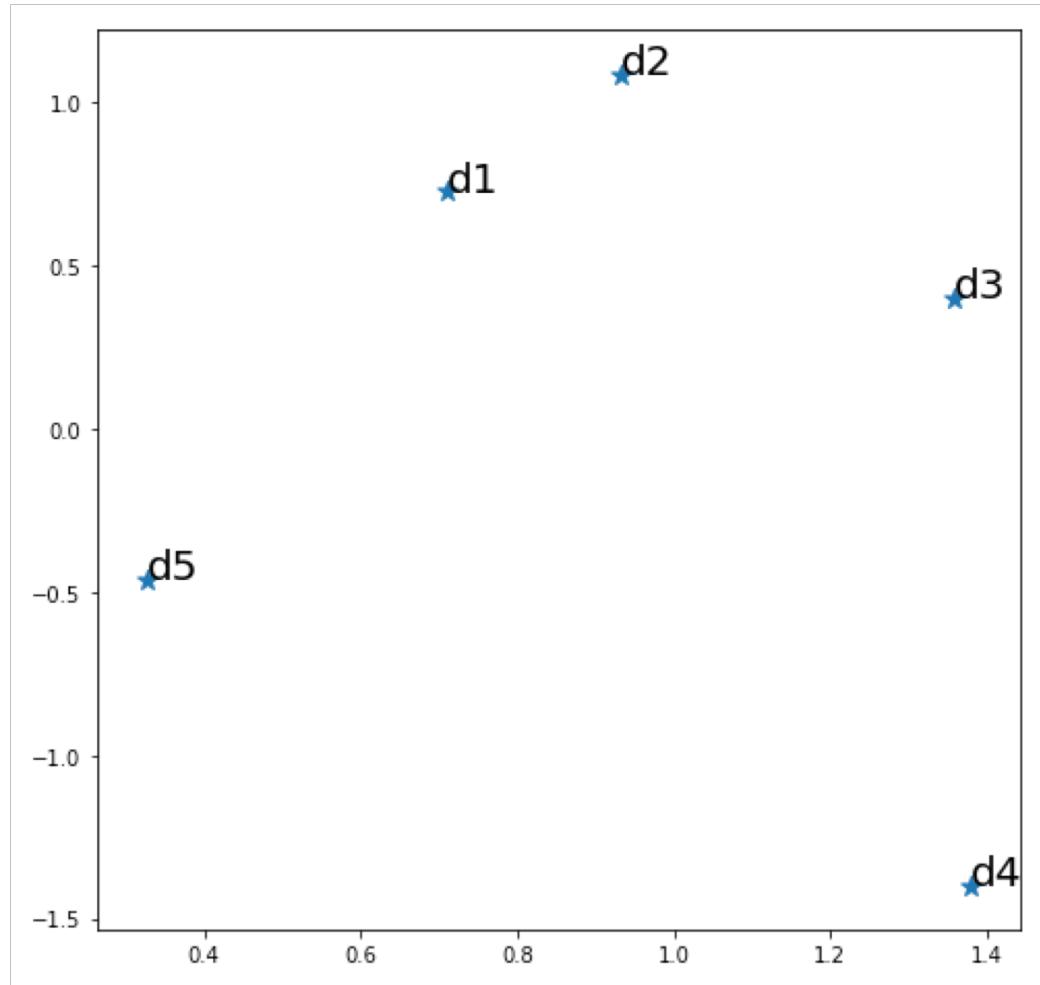
$$\Sigma_2 = \begin{bmatrix} 2.285 & 0 \\ 0 & 2.010 \end{bmatrix}$$

$U_2 = \text{array}([[0.31086574, 0.36293322, } d_1 \\ [0.40733041, 0.54074246, } d_2 \\ [0.59446137, 0.20005441, } d_3 \\ [0.60304575, -0.6953914, } d_4 \\ [0.1428143, -0.22866156, } d_5$

Document vector

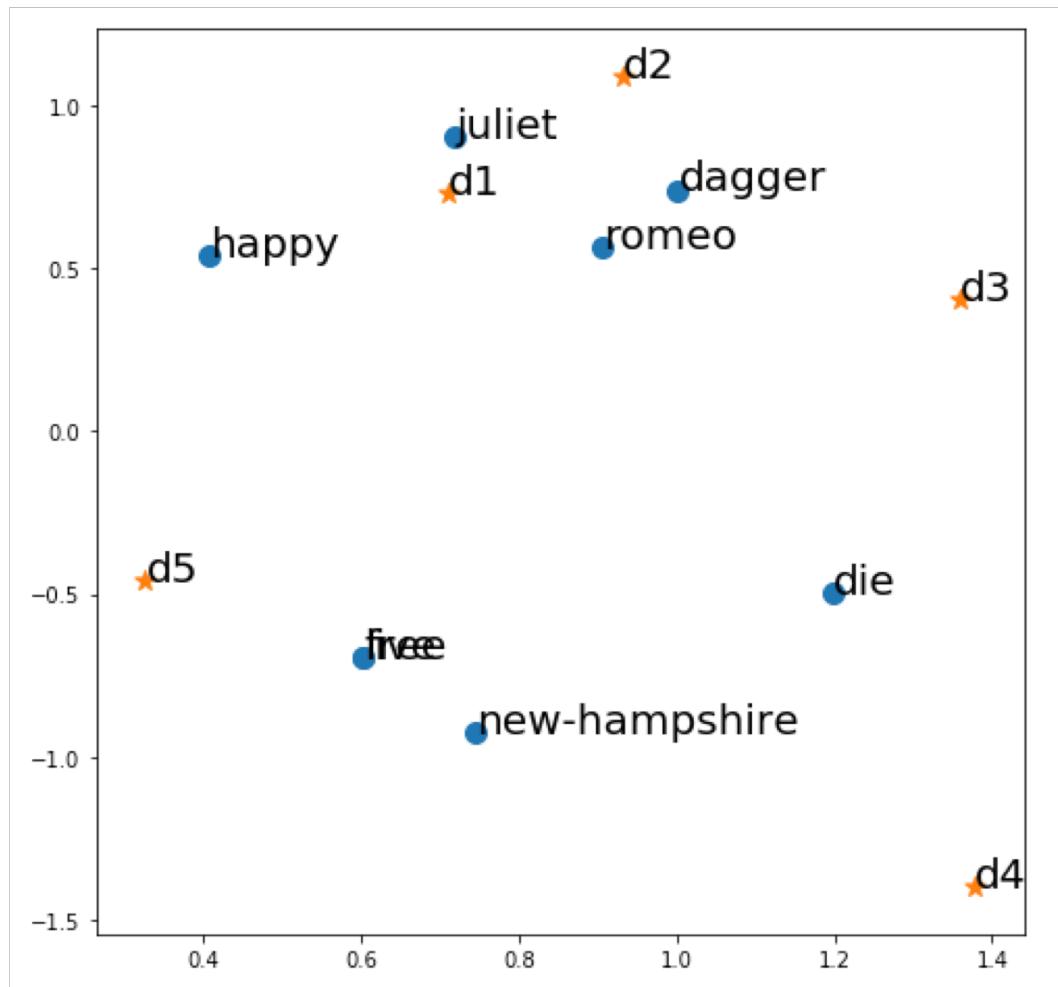
$U_2\Sigma_2 = \text{array}([[0.71042084, 0.7295895, } d_1 \\ [0.93087134, 1.08703198, } d_2 \\ [1.35852135, 0.40216102, } d_3 \\ [1.37813921, -1.39791629, } d_4 \\ [0.32637322, -0.45966878]]) } d_5$

LSA (Latent Semantic Analysis)



Document vector \ominus scatter

LSA (Latent Semantic Analysis)



Word / Document vector \ominus scatter

LSA (Latent Semantic Analysis)

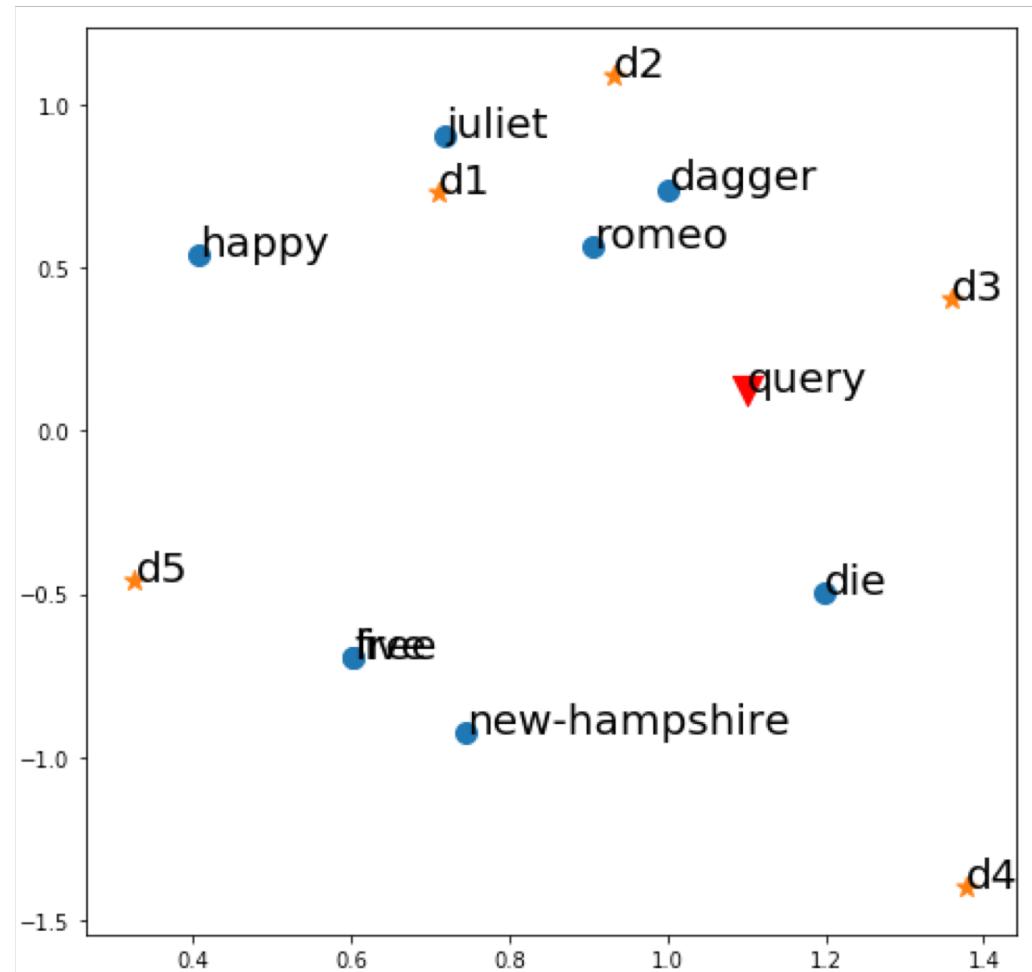
$$q = \frac{q_1 + q_2}{2}$$

$$\text{cosine similarity} = \frac{d_i \cdot q}{|d_i||q|}$$

query : dagger, die

```
result : [(0.9844359912676067, 'Romeo die by dagger.'),  
          (0.7727964887537556, 'Romeo and Juliet.'),  
          (0.7306768205359726, 'Juliet: O happy dagger!'),  
          (0.6187306127613211, "'Live free or die', that's the motto of New-Hampshire"),  
          (0.4849183185073821, 'Did you know, New-Hampshire is in New-England')]
```

LSA (Latent Semantic Analysis)



Word / Document / Query vector의 scatter