Week 6. 행렬분해 (Matrix decomposition)

- 인공지능은 데이터를 기반으로 자료를 분류하거나 미래를 예측하는 등 합리적 의사결정을 한다.
- 그러나 이를 실제로 구현하기 위해서는 프로그래밍 언어를 거쳐 코딩을 해야한다.
- 즉 우리는 컴퓨터가 이해하고 계산하는 방식을 이해해야 한다.
- 지금까지 학습한 행렬연산을 실제로 구현하기 위해 사용되는 행렬 분해기법을 간단히 소개한다.
  - ▷ 먼저 LU 분해와 QR 분해에 대하여 학습한다.
  - ▷ 실제 응용성이 높은 특잇값분해(SVD)에 대하여 학습한다.

6.1 LU 분해 (LU decomposition)

하나의 정사각행렬 A

$$A = \begin{bmatrix} l_1 & 0 & 0 & 0 \\ * & l_2 & 0 & 0 \\ * & * & l_3 & 0 \\ * & * & * & l_4 \end{bmatrix} \begin{bmatrix} u_1 & * & * & * \\ 0 & u_2 & * & * \\ 0 & 0 & u_3 & * \\ 0 & 0 & 0 & u_4 \end{bmatrix} = LU$$

하삼각행렬 L(lower)과 상삼각행렬 U(upper)의 곱 $A = LU$로 표현하는 것을 행렬의 LU분해라고 한다. LU 분해는 보통 선형연립방정식을 풀 때 많이 쓰인다.

— 계수행렬이 삼각행렬인 경우, 해를 구하기 위한 계산에 대입만 필요하므로 계산량 훨씬 적기 때문.

$$b = Ax = \begin{bmatrix} l_1 & 0 & 0 & 0 \\ * & l_2 & 0 & 0 \\ * & * & l_3 & 0 \\ * & * & * & l_4 \end{bmatrix} \begin{bmatrix} u_1 & * & * & * \\ 0 & u_2 & * & * \\ 0 & 0 & u_3 & * \\ 0 & 0 & 0 & u_4 \end{bmatrix} = LUx = Ly$$

이면, $b = Ly$를 먼저 쉽게 풀어 y를 구한 후, $Ux = y$를 풀어서 $b = Ax$를 만족하는 x를 구하면 된다.

| 단계 1 | $Ax = b$를 $LUx = b$로 쓰고 우선 $Ly = b$를 만족하는 y를 구한다

| 단계 2 | 위에서 구한 y를 이용하여 $Ux = y$를 만족하는 x를 구한다.

요약 → $A = LU$ 또는 $A = PLU$ (P는 치환행렬이며 $P^TP = I_n$)로 분해하여, 선형연립방정식 $Ax = b$ (또는 $A'x = b'$ 여기 $A' = LU$)를 푸는 것!

6.2 QR분해

— 최소제곱문제를 해결할 때 활용

$m \times n$ $(m \geq n)$ 행렬 A에 대하여 정규직교벡터들을 열로 하는 $m \times n$ 행렬 Q와 $n \times n$의 상삼각행렬 R의 곱 $A = QR$로 표현하는것을 QR분해라고 한다.

* 정규직교벡터 = 자신의 크기가 모두 1이면서 서로 다른 벡터들끼리 내적이 0인 벡터
따라서 Q는 $Q^TQ = I_n$을 만족한다.

$\boxed{A = QR}$이 주어졌을 때, 최소제곱문제 $\min \|Ax-b\|$는 다음과 같이 구할 수 있다.
앞서 $\hat{x} = (A^TA)^{-1}A^Tb$ 이 최소제곱문제의 해이므로, 관계식

$$\min \|Ax-b\| \Leftrightarrow A^TAx = A^Tb$$

$$\Leftrightarrow (R^TQ^T)(QR)x = R^TQ^Tb \Leftrightarrow R^TRx = R^TQ^Tb$$

$$\Leftrightarrow Rx = Q^Tb$$

으로부터 $\boxed{\hat{x} = R^{-1}Q^Tb}$ 임을 쉽게 알 수 있다.

6.3 SVD (Singular Value Decomposition, 특잇값 분해)

* 임의의 크기를 가진 행렬의 SVD는 모두 존재 → 선형대수학에서 가장 많이 응용

크기가 $m \times n$인 행렬 A에 대하여 $m \times m$의 직교행렬 (orthogonal matrix) U와 $n \times n$의 직교행렬 V 및 $m \times n$의 대각선 행렬 Σ가 존재하여 다음을 만족한다

$$A = U\Sigma V^T = U\begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} V^T$$

singular value

$$= \underbrace{\begin{bmatrix} u_1 & u_2 & \cdots & u_k & u_{k+1} & \cdots & u_m \end{bmatrix}}_{U} \underbrace{\left[\begin{array}{cccc|ccc} \sigma_1 & & & 0 & 0 & \cdots & 0 \\ & \sigma_2 & & & 0 & \cdots & 0 \\ & & \ddots & & \vdots & & \vdots \\ 0 & & & \sigma_k & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{array}\right]}_{\Sigma} \underbrace{\begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{bmatrix}}_{V}$$

left singular vector

right singular vector

직교행렬은 $U^TU = I_m = UU^T$, $V^TV = I_n = VV^T$을 의미하고, Σ_1은 주대각선분이 모두 단조감소의 순서로 배열된 양수 ($\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \sigma_k > 0$)인 가역 대각선행렬이고, O는 영행렬이다.

위 행렬 Σ_1의 대각선 성분들을 행렬 A의 특잇값 (singular value)이라 하고, U의 열들을 A의 좌 특이벡터 (left singular vector), V의 열들을 우특이벡터 (right singular vector)라 한다.

SVD Key Idea

- 행렬 대각화하는 한 방법
- 모든 크기의 행렬에 적용 가능
- 실수행렬 및 복소수 성분 행렬에 대해서도 성립
- 큰 크기의 행렬을 작은 크기의 행렬로 바꿔 인공지능처리에 용이하게 한다.

* 행렬분해의 개념, 구체적 유도 과정 및 응용 사례

A 고윳값과 행렬의 대각화 등 개념은 아래 참고

〈선형대수학〉, 이상구·김재만·김경원, 빅북총서

- Chapter 4. 행렬식
- Chapter 8. 행렬의 대각화

def.

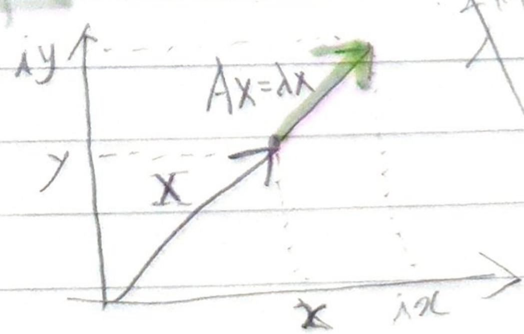
고유값 eigenvalue

고유벡터 eigenvector

A를 n차의 정사각행렬이라 하자. 0이 아닌 벡터 $x \in \mathbb{R}^n$가 적당한 스칼라 λ에 대하여 다음을 만족하면 λ를 A의 고유값 (eigenvalue)이라 하고, x를 λ에 대응하는 A의 고유벡터 (eigenvector)라고 한다

$$\boxed{Ax = \lambda x}$$

고유값 / 고유벡터 원래 행렬과 관계



(background)

"Ax가 x와 평행이 되게 하는 0 아닌 벡터 x가 존재하는가?"

* 고유값을 구하는 일반적인 방법

$Ax = \lambda x \Longleftrightarrow Ax = \lambda I_n x \Longleftrightarrow (\lambda I_n - A)x = 0$ 이고,
또한 $x \neq 0$ 이므로 동차연립방정식 $(\lambda I_n - A)x = 0$ 은
0 아닌 해를 가져야 한다. 따라서 특성방정식 (characteristic equation)
$\boxed{|\lambda I_n - A| = 0 \text{ 이 성립}}$해야 한다.

Theorem A가 $n \times n$ 행렬이고, λ가 스칼라이면 다음 명제들은 동치이다.
(1) λ는 A의 고유값이다.
(2) λ는 특성방정식 $\det(\lambda I_n - A) = 0$의 해가 된다.
(3) 동차선형 연립방정식 $(\lambda I_n - A)x = 0$은 자명하지 않은 해를 갖는다.

행렬의 대각화

A가 어떤 대각선행렬과 닮음행렬일 때, 즉 적당한 가역행렬 P가 존재하여 $P^{-1}AP$가 대각선행렬일 때, A를 대각화 가능한 (diagonalized) 행렬이라고 하며, 이 때 행렬 P를 'A를 대각화하는 (diagonalizing)' 행렬이라 한다.

* $P^{-1}AP = D$
P - 가역행렬
A - 정사각행렬
D - 대각선행렬

행렬의 닮음?

정사각행렬 A, B에 대하여 다음을 만족하는 가역행렬 P가 존재할 때
B는 A와 닮은 (similar) 행렬이라고 한다.

$$B = P^{-1}AP$$

이 때, $B \sim A$라 쓴다.

* 대각화 필요충분조건
A가 n개의 일차독립 고유벡터를 갖는 것
(이 때, A는 자신의 고유값 $\lambda_1, \lambda_2, \cdots \lambda_n$을 주대각성분으로 갖는 대각선행렬 D와 닮은행렬)

Theorem n차의 정사각행렬 A, B, C에 대하여 다음이 성립
(1) $A \sim A$
(2) $B \sim A \Rightarrow A \sim B$
(3) $B \sim A,\ A \sim C \Rightarrow B \sim C$

n차의 두 정사각행렬 A, B가 $A \sim B$이면,
(1) $\det(A) = \det(B)$
(2) $tr(A) = tr(B)$

// 고유벡터의 일차독립성) $x_1, x_2, \cdots x_k$를 행렬 $A = [a_{ij}]_{n \times n}$의 서로 다른 고유값 $\lambda_1, \lambda_2, \cdots \lambda_k$에 대응하는 고유벡터라 하면 $\{x_1, \cdots x_k\}$는 일차독립이

// n차의 정사각행렬 A가 n개의 서로 다른 고유값을 가지면 A는 대각화 가능하다.