4-3. Cholesky Decomposition

LU Decomposition (LU 분해)

LU 분해는 정사각행렬 A 를 두개의 삼각행렬,

- L (하삼각행렬, Lower Triangular matrix)
- U (상삼각행렬, Upper Triangular matrix)

로 나누는 분해입니다. → $A = LU$

연립 방정식 (2-3. Solving Systems of Linear Equations)

2-3 장에서는 연립방정식이 있다면, 이를 $Ax = b$ 형태로 표현할 수 있었습니다.

그리고 가우스 조던 소거법 (기본행연산)을 통해(식 4, 5, 7) x를 쉽게 구할 수 있었습니다.

$$\begin{cases} x + y + z = 6 \\ 2x + 3y - z = 5 \\ 2x + 3y + 3z = 17 \end{cases} \tag{3}$$

$$r_2 \rightarrow r_2 - 2r_1 \tag{4}$$

$$r_3 \rightarrow r_3 - 2r_1 \tag{5}$$

$$\begin{cases} x + y + z = 6 \\ 0x + y - 3z = -7 \\ 0x + y + z = 5 \end{cases} \tag{6}$$

$$r_3 \rightarrow r_3 - r_2 \tag{7}$$

$$\begin{cases} x + y + z = 6 \\ 0x + y - 3z = -7 \\ 0x + 0y + 4z = 12 \end{cases} \tag{8}$$

- 해당 예시에서 미지수 x, y, z 값을 구하는 과정에서 가장 마지막 미지수 z를 통해 y를, y를 통해 x를 구해나가는 방식을 **back substitution** 이라고 합니다.

해당 예시에서 구한 과정을 첨가 행렬로 표현해보면 다음과 같습니다.

- 식 (3)을 첨가행렬로 표현

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 2 & 3 & -1 & 5 \\ 2 & 3 & 3 & 17 \end{array}\right] \tag{9}$$

- 식 (4, 5, 7) 를 통해 얻은 결과

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 1 & -3 & -7 \\ 0 & 0 & 4 & 12 \end{array}\right] \tag{10}$$

- 얻게 된 과정 (기본행연산)을 기본 행렬로 표현

$$\overset{r_3 \to (r_3 - r_2)}{\begin{bmatrix}1&0&0\\0&1&0\\0&-1&1\end{bmatrix}} \underset{r_3 \to (r_3 - 2r_1)}{\begin{bmatrix}1&0&0\\0&1&0\\-2&0&1\end{bmatrix}} \overset{r_2 \to (r_2 - 2r_1)}{\begin{bmatrix}1&0&0\\-2&1&0\\0&0&1\end{bmatrix}} \underset{[A|b]}{\left[\begin{array}{ccc|c}1&1&1&6\\2&3&-1&5\\2&3&3&17\end{array}\right]} = \left[\begin{array}{ccc|c}1&1&1&6\\0&1&-3&-7\\0&0&4&12\end{array}\right]$$

해당 식을 자세히 살펴보면, 삼각행렬로 표현된다는 것을 보실 수 있습니다.

$$\overset{r_3 \to (r_3 - r_2)}{\begin{bmatrix}1&0&0\\0&1&0\\0&-1&1\end{bmatrix}} \underset{r_3 \to (r_3 - 2r_1)}{\begin{bmatrix}1&0&0\\0&1&0\\-2&0&1\end{bmatrix}} \overset{r_2 \to (r_2 - 2r_1)}{\begin{bmatrix}1&0&0\\-2&1&0\\0&0&1\end{bmatrix}} \underset{A}{\begin{bmatrix}1&1&1\\2&3&-1\\2&3&3\end{bmatrix}} = \begin{bmatrix}1&1&1\\0&1&-3\\0&0&4\end{bmatrix}$$

- 파란선 $\rightarrow L(Lower\ triangular)$
- 주황선 $\rightarrow U(Upper\ triangular)$

이를 식으로 표현하면 $L_1L_2L_3A = U$ 으로 표현할 수 있습니다.

$A = LU$ 로 표현하기 위해,

- $L_3^{-1}L_2^{-1}L_1^{-1}L_1L_2L_3A = L_3^{-1}L_2^{-1}L_1^{-1}U = LU$

해당 과정을 계산하면 다음과 같습니다.

$$\overset{A}{\begin{bmatrix}1 & 1 & 1\\ 2 & 3 & -1\\ 2 & 3 & 3\end{bmatrix}} = \begin{bmatrix}1 & 0 & 0\\ -2 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}^{-1}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -2 & 0 & 1\end{bmatrix}^{-1}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -1 & 1\end{bmatrix}^{-1}\begin{bmatrix}1 & 1 & 1\\ 0 & 1 & -3\\ 0 & 0 & 4\end{bmatrix}$$

$$= \begin{bmatrix}1 & 0 & 0\\ 2 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 2 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 1 & 1\end{bmatrix}\begin{bmatrix}1 & 1 & 1\\ 0 & 1 & -3\\ 0 & 0 & 4\end{bmatrix}$$

$$= \underset{L}{\begin{bmatrix}1 & 0 & 0\\ 2 & 1 & 0\\ 2 & 1 & 1\end{bmatrix}}\underset{U}{\begin{bmatrix}1 & 1 & 1\\ 0 & 1 & -3\\ 0 & 0 & 4\end{bmatrix}}$$

PLU decomposition (PLU 분해)

- P : permutation

어떠한 행렬 A는 행교환을 하지 않으면 LU분해가 되지 않는 경우가 있습니다. 앞서 LU 분해의 예시는 행교환을 하지 않는 예시 였습니다.

다음과 같은 행렬이 있다고 가정해보면, row addition 과 row scaling를 통해 상삼각행렬 (U)를 만들 수 없습니다.

$$A = \begin{bmatrix} 0 & 0 & 3 \\ 1 & 1 & 1 \\ 2 & 3 & -1 \end{bmatrix} \tag{15}$$

따라서 1행과 3행을 바꿔줌(행교환)으로써 상삼각행렬을 구할 수 있습니다.

전체 과정은 다음과 같습니다.

$$A = \begin{bmatrix} 0 & 0 & 3 \\ 1 & 1 & 1 \\ 2 & 3 & -1 \end{bmatrix} \tag{21}$$

$$= P_{13}LU = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 & -1 \\ 0 & -1/2 & 3/2 \\ 0 & 0 & 3 \end{bmatrix} \tag{22}$$

LDU decomposition (LDU 분해)

LDU분해에서 D는 대각행렬을 뜻합니다.

L(하삼각행렬)과 U(상삼각행렬)의 대각 성분(diagonal)을 모두 1로 만드는 방법입니다.
핵심 아이디어는 LU분해에서 U의 대각 성분을 D로 분리 하는 것입니다.

위 LU분해의 식을 보면 U의 대각 성분은 1이 아닙니다.

$$= \underset{L}{\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix}} \underset{U}{\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & -3 \\ 0 & 0 & 4 \end{bmatrix}}$$

해당 식을 대각행렬을 활용하여 다음과 같이 분해할 수 있습니다.

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix},\ D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 4 \end{bmatrix},\ U = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}$$

LU 분해의 활용

1. $Ax = b$ 해 구하기

1. $Ax = b \rightarrow LUx = b$
2. $Ux = y$ 의 꼴로 바꿔줍니다.
3. $Ly = b$ 를 통해 y를 구할 수 있습니다.
4. 이를 통해 back-substitution 을 이용해 $Ux = y$ 를 쉽게 구할 수 있습니다.

> LU분해를 활용하면 기존 해를 구하는 방식보다 더욱 쉽게 풀 수 있습니다.

2. 재사용성

만약 $Ax = b$ 에서 b가 바뀔 때마다 LU 한 번만 계산하면 새로운 x를 빠르게 구할 수 있습니다.

3. 행렬식 계산

$A = LU \rightarrow det(A) = det(L)det(U)$

L과 U는 삼각행렬이므로, 대각성분 곱만으로 행렬식을 계산할 수 있습니다.

$$det(A) = \prod_{i=1}^{n} l_{i,i} \prod_{j=1}^{n} u_{j,j} = \prod_{i=1}^{n} l_{i,i} u_{i,i}$$

4-3. Cholesky Decomposition

machine learning에서는 매우 많은 특별한 분해 방법이 있습니다.

대칭(symmetric), 양의 정부호(positive definite matrices) 행렬의 경우, 제곱근(square root)에 해당하는 다양한 연산 선택할 수 있습니다. 그 중에서 Cholesky(숄레스키) 분해(인수분해)는 대칭이고, 양의 정부호인 행렬에 대해 제곱근 계산을 제공합니다.

정리 4.18 (Cholesky Decomposition)

대칭이고 양의 정부호인 행렬 A는 다음과 같이 분해할 수 있습니다.

- $A = LL^T(L)$

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ l_{n1} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & \cdots & l_{n1} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & l_{nn} \end{bmatrix} . \qquad (4.44)$$

- L은 A의 Cholesky factor 이며, L은 유일합니다.

대칭 행렬 A의 조건

대칭행렬 A를 다음과 같이 분해(표현)할 수 있습니다. $A = LL^T = L^TL$

이는 LU 형태와 매우 유사한 구조로 볼 수 있습니다. 하지만 모든 대칭행렬에 대해 위 분해가 항상 가능한 것은 아닙니다.

위 분해가 가능하려면 $A = LL^T$ 형태에서 나온 하삼각행렬 L이 갖는 성질을 분석해야 합니다.

- 벡터 x에 대한 $L2$-Norm의 관점

  벡터의 $L2$-Norm 의 제곱은 다음과 같습니다.

  - $||Lx||^2 = (Lx)^T(Lx) = x^TL^TLx$

  여기에서 L^TL을 A라고 한다면, 다음을 만족해야합니다.

  - $x^TAx \geq 0$

  이 조건을 만족하는 행렬 A를 semi-positive definite 행렬이라고 합니다.

그래서 행렬 A가 다음 3가지 조건을 만족할 때만

$$A = LL^T = L^TL$$

Cholesky 분해가 가능합니다.

- square matrix (정방행렬)
- symmetric (대칭행렬)
- positive definite (양의 정부호) → (Cholesky 분해를 안전하고 유일하게 수행하기 위함)

예시 4.10 (Cholesky Factorization)

다음 대칭, 양의 정부호 행렬 $A \in \mathbb{R}^{3\times 3}$ 이 있습니다. 이 행렬의 Cholesky 분해 $A = LL^T$를 찾아보겠습니다.

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \boldsymbol{L}\boldsymbol{L}^\top = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix} . \quad (4.45)$$

위 식의 우항을 계산하면, 다음과 같습니다.

$$\boldsymbol{A} = \begin{bmatrix} l_{11}^2 & l_{21}l_{11} & l_{31}l_{11} \\ l_{21}l_{11} & l_{21}^2 + l_{22}^2 & l_{31}l_{21} + l_{32}l_{22} \\ l_{31}l_{11} & l_{31}l_{21} + l_{32}l_{22} & l_{31}^2 + l_{32}^2 + l_{33}^2 \end{bmatrix} . \quad (4.46)$$

위 식(4.46)을 식(4.45)와 비교하면, 대각성분 l_{ii}에서 간단한 패턴이 있는 것을 볼 수 있습니다.

$$l_{11} = \sqrt{a_{11}}\,, \quad l_{22} = \sqrt{a_{22} - l_{21}^2}\,, \quad l_{33} = \sqrt{a_{33} - (l_{31}^2 + l_{32}^2)}\,. \quad (4.47)$$

비슷하게, 대각 요소 아래에 있는 요소들도 다음과 같은 반복적인 패턴을 갖습니다.

$$l_{21} = \frac{1}{l_{11}} a_{21}\,, \quad l_{31} = \frac{1}{l_{11}} a_{31}\,, \quad l_{32} = \frac{1}{l_{22}} (a_{32} - l_{31}l_{21})\,. \quad (4.48)$$

해당 패턴을 일반화하면 다음과 같습니다.

$$L_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} L_{jk}^2}$$

$$L_{ij} = \frac{1}{L_{jj}}\left(a_{ij} - \sum_{k=1}^{j-1} L_{ik}L_{jk}\right) \quad \text{for } i > j$$

핵심적인 통찰은 행렬 A의 원소 a_{ij} 와 이미 계산된 l_{ij} 값을 바탕으로 하삼각행렬 L의 원소 l_{ij} 를 역으로 계산해낼 수 있다는 점입니다.

Cholesky 분해는 머신러닝의 수치 계산에서 매우 중요한 도구입니다.

머신러닝에서는 양의 정부호 행렬을 자주 다룹니다. 예를 들어 다변량 가우시안 변수의 공분산 행렬은 대칭이고 양의 정부호입니다. 이러한 공분산 행렬에 대해 Cholesky 분해를 적용하면,

- 가우시안 분포로 부터 샘플 생성
- 확률 변수에 대한 선형 변환 (VAE)
- 효율적인 행렬식 계산

$$\det(A) = \det(L)\det(L^\top) = \det(L)^2$$

$$\det(A) = \prod_i l_{ii}^2$$

등을 할 수 있습니다.

출처

- Mathmatics for Machine Learning (https://github.com/mml-book/mml-book.github.io)
- https://junstar92.github.io/mml-study-note/2022/07/14/ch4-3.html
- https://www.youtube.com/watch?v=z66pF_yiGVQ&list=PL_iJu012NOxdZDxoGsYidMf2_bERIQaP0&index=34
- https://angeloyeo.github.io/2021/06/16/LU_decomposition.html

- https://angeloyeo.github.io/2021/06/17/Cholesky_decomposition.html