

# 1장 부록

## 보충 설명: 1.2.1절

**정리 1A.1. [LU 분해(LU decomposition)  $A = LU$ ]**

행 교환이 필요하지 않은 경우, 모든  $m \times m$  행렬  $A$ 는 유일한  $A = LU$ , 즉 **LU 분해(LU decomposition)**를 갖는다. 여기서  $L$  행렬은 하삼각행렬이며 대각성분은 1, 대각 아래 성분은 소거 과정에서 얻은 승수이다.  $U$ 는 상삼각행렬로, 전진 소거(forward elimination) 이후와 후진 대입(back-substitution) 이전에 나타나며, 대각 성분들이 피벗(pivot)이다.

(증명) 예를 들어,  $4 \times 4$  행렬을 생각하자. (자세한 증명은 1 또는 2 을 참조하라.)

$$\underbrace{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}}_A \xrightarrow{M_1} \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}}_{M_1 A} \xrightarrow{M_2} \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & 0 & \times & \times \\ & 0 & \times & \times \end{bmatrix}}_{M_2 M_1 A} \xrightarrow{M_3} \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & 0 & \times \end{bmatrix}}_{M_3 M_2 M_1 A} \quad (1)$$

$m - 1$  steps 이후의 행렬은 상삼각행렬  $U$ 가 된다.

$$\underbrace{M_{m-1} \cdots M_2 M_1}_{L^{-1}} A = U \quad (2)$$

$k$  step의 시작에서,  $\mathbf{x}_k = \tilde{A}_k(:, k)$ 라 하자.

이는 행렬의  $k$ 번째 열을 나타낸다.

$$\tilde{A}_k = M_{k-1} \cdots M_2 M_1 A = \begin{bmatrix} \tilde{A}_k(1:k-1, 1:k-1) & \tilde{A}_k(1:k-1, k:m) \\ \mathbf{0} & \tilde{A}_k(k:m, k:m) \end{bmatrix} \quad (3)$$

이 때,  $\tilde{A}_k(1:k-1, 1:k-1)$ 는 상삼각행렬이다. 만약,

$$M_k = I - \mathbf{l}_k \mathbf{e}_k^T \quad \text{where} \quad \mathbf{e}_k = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \mathbf{l}_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ l_{k+1,k} \\ \vdots \\ l_{m,k} \end{bmatrix}, \quad l_{jk} = \frac{x_{jk}}{x_{kk}}, \quad j = k+1, \dots, m \quad (4)$$

이러면  $M_k^{-1} = I + \mathbf{l}_k \mathbf{e}_k^T$ 이다. 참고로  $\mathbf{l}_k(k+1:m) = \tilde{A}_k(k+1:m, k)/\tilde{A}_k(k, k)$ 이다.  
 $k$  step 이후에, 다음을 얻을 수 있다.

$$\tilde{A}_k(:, k) = \mathbf{x}_k = \begin{bmatrix} x_{1k} \\ \vdots \\ x_{kk} \\ x_{k+1,k} \\ \vdots \\ x_{m,k} \end{bmatrix} \implies \tilde{A}_{k+1}(:, k) = M_k \mathbf{x}_k = \mathbf{x}_k - \mathbf{l}_k (\mathbf{e}_k^T \mathbf{x}_k) = \begin{bmatrix} x_{1k} \\ \vdots \\ x_{kk} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (5)$$

그리고  $j = k+1, \dots, m$ 에 대해서,

$$\tilde{A}_{k+1}(:, j) = M_k \mathbf{x}_j = \mathbf{x}_j - \mathbf{l}_k (\mathbf{e}_k^T \mathbf{x}_j) = \mathbf{x}_j - \mathbf{l}_k x_{kj} = \begin{bmatrix} x_{1j} \\ \vdots \\ x_{kj} \\ \times \\ \vdots \\ \times \end{bmatrix} \quad (6)$$

$$\implies \tilde{A}_{k+1}(k+1:m, j) = \tilde{A}_k(k+1:m, j) - \mathbf{l}_k(k+1:m) * \tilde{A}_k(k, j)$$

왜냐하면  $M_k^{-1} M_{k+1}^{-1} = I + l_k \mathbf{e}_k^T + l_{k+1} \mathbf{e}_{k+1}^T$ 이며, 다음을 얻는다.

$$L = M_1^{-1} \dots M_{m-1}^{-1} = \begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ l_{m1} & l_{m2} & \dots & l_{m,m-1} & 1 \end{bmatrix} \quad (7)$$

삼각 행렬분해(triangular factorization)은  $L$ 과  $U$ 가 대각에 1을 가지고  $D$ 가 피벗의 대각 행렬일 때 종종 쓰인다. ◀

이 알고리즘에서,  $L$ 과  $U$ 는  $A$ 와 똑같은 행렬에 덮어 씌우며 컴퓨터의 메모리 사용을 최소화 할 수 있다.

가우시안 소거법의 경우  $LU$  분해 계산 비용은  $\sim \frac{2}{3}m^3$  flops이다.

## 피벗팅 가우시안 소거법 (Gaussian Elimination with Pivoting)

$LU$  분해는 가우시안 소거법을 통해 가장 잘 이해할 수 있다. 가우시안 소거법에서, 행렬은 행 연산을 통해 수정되어 상삼각행렬  $U$ 를 만든다. 이 행 연산들을 따라가면,  $L$  행렬을 찾을 수 있다. 행 연산(row operation)은 한 행을 다른 행들과 선형 결합(linear combination)한 결과로 대체하는 것이다.  $U$ 의 대각성분들은 **피벗(pivots)**라고 하며,  $A$ 가 diagonally dominant하다면 피벗팅은 피해야 한다.

### 정리 1A.2 [ $PA = LU$ ]

모든  $m \times m$  행렬  $A$ 에 대해  $PA = LU$ 를 만족하는 치환행렬(permutation matrix)  $P$ , 단위 대각행렬을 갖는 하삼각행렬  $L$ ,  $m \times m$  상삼각행렬  $U$ 가 있으며, 이를 **피벗팅 LU 분해(LU decomposition with pivoting)**라 한다.

(증명) 예를 들어, 행렬  $A$ 를 생각하자. (자세한 증명은 1 또는 2 을 참조하라.)

$$\underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ x_{ik} & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}}_{\text{Pivot selection}} \xrightarrow{P_1} \underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}}_{\text{Row interchange}} \xrightarrow{M_1} \underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & 0 & \times & \times & \times \\ & 0 & \times & \times & \times \\ & 0 & \times & \times & \times \end{bmatrix}}_{\text{Elimination}} \quad (8)$$

$m - 1$  steps 후에, 행렬은 상삼각행렬  $U$ 가 된다.

$$M_{m-1}P_{m-1} \cdots M_2P_2M_1P_1A = U \quad (9)$$

이 때,  $P_2M_1 = (P_2M_1P_2^{-1})P_2$ 이며,  $P_2M_1P_2^{-1}$ 는 다음과 같다.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \tau_1 & 1 & 0 & 0 & 0 \\ \tau_2 & 0 & 1 & 0 & 0 \\ \tau_3 & 0 & 0 & 1 & 0 \\ \tau_4 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \tau_3 & 1 & 0 & 0 & 0 \\ \tau_2 & 0 & 1 & 0 & 0 \\ \tau_1 & 0 & 0 & 1 & 0 \\ \tau_4 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$$M'_k = P_{m-1} \cdots P_{k+1} M_k P_{k+1}^{-1} \cdots P_{m-1}^{-1} \quad (11)$$

라 하자.

그렇다면  $M'_k$ 는 단위 하삼각행렬이며 쉽게 역연산이 가능하다. (피벗팅을 하지 않는 가우시안 소거법처럼 negating the sub-diagonal entries을 통해)

따라서,

$$(M'_{m-1} \cdots M'_2 M'_1)(P_{m-1} \cdots P_2 P_1)A = U \quad (12)$$

$L = (M'_{m-1} \cdots M'_2 M'_1)^{-1}$ ,  $P = P_{m-1} \cdots P_2 P_1$  라 하면,

$$PA = LU \quad (13)$$

이다.

이 때,  $P^{-1} = P^T$ 이다. ◀

피벗팅의 목적은 가우시안 소거법을 모든 행렬에 대해 적용 가능하고 안정되게 만드는 것이다. 안정성의 측면에서, 피벗팅은 일반적으로  $\|L\|$  을 order 1로 보장하며,  $\|U\|$  를 order of  $\|A\|$  로 보장한다. 하지만, 특정 행렬  $A$ 에 대해서는  $\|U\|/\|A\|$  가 매우 클 수도 있다. 예를 들어 피벗팅이 일어나지 않는  $PA = LU$  분해에서,

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 & 0 \\ -1 & -1 & -1 & 1 & 0 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 & 8 \\ 0 & 0 & 0 & 0 & 16 \end{bmatrix} \quad (14)$$

이 패턴은 다음과 같이 임의 차원의  $u_{mm} = 2^{m-1}$ 을 갖는 임의의  $m$ 차원의 행렬로 계속 이어질 수 있다.

그러나 이러한 예에도 불구하고 부분 피벗팅은 실제로 매우 안정적이다. 만약 무작위로 수억 개의 행렬들중에서  $A$ 를 임의로 고른다면, 대부분의 행렬에서 이런 현상을 보지 못할 것이다.

## 보충 설명: 1.3절

행렬식은 다음의 세 가지 가장 기본적인 속성으로 정의할 수 있다.

### 정의 1A.2. [행렬식; Determinant]

- 행렬식은 첫 번째 행에 선형적으로 의존한다.
- 행렬식은 두 행이 교환될 때 부호가 바뀐다.
- 단위 행렬의 행렬식은 1이다.

위 행렬식 정의로부터, 다음이 성립함을 쉽게 보일수 있다.

- $A$ 의 두 행이 같으면,  $\det(A) = 0$ .
- 한 행의  $n$ 배를 다른 행에서 빼는 연산은 행렬식을 변화시키지 않는다.
- $A$ 가 0인 행이나 열을 가지면,  $\det(A) = 0$ .
- $A$ 의 전치 행렬의 행렬식은  $A$  자신의 행렬식과 같다  
 $\det(A^T) = \det(A)$ .

**정리 1A.3 [행렬식의 공식]** 정방행렬(square matrix)  $A = (a_{ij}) \in \mathfrak{R}^{n \times n}$ 에 대해

1. 행렬식은 피벗에 대한 수식을 제공한다. 즉,  $A$ 가 가역 행렬이면,  $A = P^{-1}LU$ 로 부터

$$\det(A) = \det(P^{-1}LU) = \pm(\text{products of the pivots}) \quad (15)$$

가 성립하여, 소거(elimination) 순서와는 상관 없이 모든 피벗의 곱은 부호를 제외하면 일정하게 유지된다. 이때,  $P$ 의 행렬식은 행이 짝수번 또는 홀수번 교환되었는지에 따라  $+1$  또는  $-1$ 을 갖고,  $\det(L) = 1$  이고  $\det(U) = u_{11} \dots u_{nn}$ 를 만족한다.

$$2. \det(A) = \sum_{\sigma} a_{1\sigma_1} a_{2\sigma_2} \dots a_{n\sigma_n} \det P_{\sigma}$$

이 때, 총합은  $(1, \dots, n)$ 까지의 모든  $n!$  permutations  $\sigma$  이다.

3.  $A$ 의 행렬식은  $i$  행과  $i$ 행의 여인수(cofactor)의 조합이다.

$$\det(A) = a_{i1}A_{i1} + a_{i2}A_{i2} + \dots + a_{in}A_{in} \quad (16)$$

이 때, 여인수  $A_{ij}$ 는  $M_{ij}$ 의 부호를 갖는 행렬식이며,

$$A_{ij} = (-1)^{i+j} \det(M_{ij}) \quad (17)$$

위의  $M_{ij}$ 는  $A$ 에서  $i$ 행과  $j$ 열을 지운 행렬이다.

정방행렬(square matrix)  $A = (a_{ij}) \in \mathfrak{R}^{n \times n}$ 의 수반행렬(Adjoint matrix)  $\text{adj}(A) = (A_{ji}) \in \mathfrak{R}^{n \times n}$ 는 다음과 같이 정의된다.

$$\text{adj}(A) := \begin{bmatrix} A_{11} & A_{21} & \dots & A_{n1} \\ A_{12} & A_{22} & \dots & A_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1n} & A_{2n} & \dots & A_{nn} \end{bmatrix} \quad (18)$$

## 행렬식의 활용(Applications of determinants)

$A^{-1}$ 의 계산,  $Ax = b$ 의 해, 평행다면체의 부피, 피벗의 공식은 모두 다음 정리에 바탕을 둔다.

**정리 1A.4. [행렬식의 활용]** 정방행렬(square matrix)  $A = (a_{ij}) \in \Re^{n \times n}$ 에 대해

(1)  $A$ 가 가역이면,  $\det(A) \neq 0$  이고

$$A^{-1} = \frac{\text{adj}(A)}{\det(A)} \quad (19)$$

이다.

(2) 크래머 공식(Cramer's rule): 각각의 원소  $b$ 에 대한  $A^{-1}b$ 의 의존도를 측정한다. 만약 한 변수가 실험에서 바뀌거나 관찰값이 수정되면,  $x = A^{-1}b$ 의 "영향 계수(influence coefficient)"가 행렬식의 비율과 동일하다. 즉,

$x = A^{-1}b$ 의  $j$ 번째 성분은,

$$x_j = \frac{\det(B_j)}{\det(A)}, \text{ where } B_j = \begin{bmatrix} a_{11} & \cdots & b_1 & \cdots & a_{1n} \\ a_{21} & \cdots & b_2 & \cdots & a_{2n} \\ \vdots & & \vdots & & \vdots \\ a_{n1} & \cdots & b_n & \cdots & a_{nn} \end{bmatrix} \quad (20)$$

$B_j$ 는 벡터  $b$ 로  $A$  행렬의  $j$ 번째 열의 값을 바꾼 행렬이다.

(3) 가우시안 소거법으로  $A$ 를 행간의 교환이나 치환 행렬 연산 없이 수행할 수 있다는 것은 선행되는 부분행렬  $A_1, \dots, A_n$ 이 비특이행렬(nonsingular)이라는 것과 필요충분조건이다.

**(증명)** (1) 아래의 식을 보자.

$$\begin{aligned}
A \cdot \text{adj}(A) &= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} A_{11} & A_{21} & \cdots & A_{n1} \\ A_{12} & A_{22} & \cdots & A_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1n} & A_{2n} & \cdots & A_{nn} \end{bmatrix} \\
&= \begin{bmatrix} \det(A) & 0 & \cdots & 0 \\ 0 & \det(A) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \det(A) \end{bmatrix} = \det(A)I
\end{aligned} \tag{21}$$

대각 성분을 제외한 모든 곳에서 성분이 0인 것을 증명하기 위하여  $j \neq i$ 이며  $i$ 번째 행이  $B$ 의  $j$ 번째 행에 복사되는 행만을 제외하고  $B$ 를  $A$ 와 동일하게 설정하자. 그러면,

$$\det(B) = 0 = a_{i1}A_{j1} + a_{i2}A_{j2} + \cdots + a_{in}A_{jn}, \quad \forall i \neq j \tag{22}$$

(2)  $\det(B_j) = (\text{adj}(A)b)_j$ .

(3)  $A$ 가 LDU로 분해된다면, 좌상단 코너(upper left corners)에서는 다음을 만족한다.

$$A_k = L_k D_k U_k \tag{23}$$

모든  $k$ 에 대해, 부분행렬  $A_k$ 는 가우시안 소거법을 거친다. 특히 피벗  $d_k$ 는 행렬식의 비율로 표현할 수 있다. 즉,

$$d_k = \frac{\det(A_k)}{\det(A_{k-1})} \tag{24}$$

## 행렬식과 역행렬 공식(Matrix Inversion and Determinant Formula)

주로 사용되는 대각합(traces)와 행렬식(determinants)의 곱의 법칙과 항등식은 다음과 같다.

### 보조정리 1A.5 [행렬식 공식]

(1) 대각합(trace)와 행렬식(determinant)의 곱의 법칙:

$$\begin{aligned}
\text{tr}(AB) &= \text{tr}(BA), & A \in \mathfrak{R}^{n \times p}, B \in \mathfrak{R}^{p \times n} \\
\det(AB) &= \det(A)\det(B) = \det(BA), & A \in \mathfrak{R}^{n \times n}, B \in \mathfrak{R}^{n \times n}
\end{aligned} \tag{25}$$

(2) 실베스터 행렬항등식 (Sylvester's determinant identity):

$$\det(I_m + AB^T) = \det(I_n + B^T A) \quad \text{where } A, B \in \mathfrak{R}^{m \times n} \tag{26}$$

(증명) (1) 증명하기 쉽다.

(2)  $P$ 와  $Q$ 를 다음과 같은 네 개의 블록으로 구성된 행렬이라 하면,

$$P = \begin{bmatrix} \mathbf{I}_m & -\mathbf{A} \\ \mathbf{B}^T & \mathbf{I}_n \end{bmatrix}, \quad Q = \begin{bmatrix} \mathbf{I}_n & \mathbf{B}^{I'} \\ -\mathbf{A} & \mathbf{I}_m \end{bmatrix} \quad (27)$$

블록 행렬  $P$ 의 행렬식은 다음과 같다.

$$\begin{aligned} \det(P) &= \det \begin{bmatrix} \mathbf{I}_m & -\mathbf{A} \\ \mathbf{B}^T & \mathbf{I}_n \end{bmatrix} = \det \begin{bmatrix} \mathbf{I}_m & -\mathbf{A} \\ \mathbf{0} & \mathbf{I}_n - \mathbf{B}^T(-\mathbf{A}) \end{bmatrix} \\ &= \det(\mathbf{I}_m) \det(\mathbf{I}_n + \mathbf{B}^T \mathbf{A}) = \det(\mathbf{I}_n + \mathbf{B}^T \mathbf{A}) \end{aligned} \quad (28)$$

이와 비슷하게  $Q$ 도 구할 수 있다.

$$\det(\mathbf{I}_n + \mathbf{B}^T \mathbf{A}) = \det(P) = \det(Q) = \det(\mathbf{I}_m + \mathbf{A} \mathbf{B}^T) \quad (29)$$

◀

### 예제 1A.2

슈어 보수행렬 보조정리를 통해

(i) 모든 대칭 행렬  $A \succ 0, C \succeq 0$ 에 대해:

$$C \succeq B^T A^{-1} B \iff \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succeq 0 \quad (30)$$

(ii) 모든 대칭 행렬  $A \succ 0, C \succeq 0$ 에 대해:

$$\left. \begin{array}{l} \mathcal{R}(B) \perp \mathcal{N}(A) \\ C \succeq B^T A^{-1} B \end{array} \right\} \iff \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succeq 0 \quad (31)$$

*Hint:* Use

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} = \begin{bmatrix} R^T & 0 \\ B^T R^{-1} & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & C - B^T A^{-1} B \end{bmatrix} \begin{bmatrix} R & R^{-T} B \\ 0 & I \end{bmatrix} \quad (32)$$

### 각주

This document is confidential and is intended solely for the use of the class.

Copyright © 2020 Prof. Jaewook Lee, Seoul National University



ALL RIGHTS RESERVED. No part of this work covered by the copyright herein may be reproduced, transmitted, stored or used in any form or by any means graphic, electronic, or mechanical, including but not limited to photocopying, recording, scanning, digitizing, taping, Web distribution, information networks, or information storage and retrieval systems without the prior written permission of the author.

---

1. [Trefethen and Bau. (1997)] L. N. Trefethen and D. Bau Numerical Linear Algebra. SIAM, Philadelphia, 1997. [↩](#) [↩](#)
2. [Golub et al. (1995)] G. H. Golub, C. F. Van Loan, Matrix Computations. Johns Hopkins University Press, Baltimore, 1995 [↩](#) [↩](#)