PLU Decomposition

발표자: 양유진

링크:https://youtu.be/sOuOfkmFGUA

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB

선형방정식의 행렬 표기법

선형시스템(Linear System)에서는 **선형 연립 1차 방정식**을 **행렬**로 표현할 수 있음.

$$\begin{cases} x_1 + 2x_2 + x_3 = 5 \\ x_1 + x_3 = 3 \end{cases} \rightarrow \begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$$

계수행렬 상수항 벡터

가우시안 소거법(Gaussian Elimination)

정의

- 미지수가 n개인 연립 1차 방정식을 나타낸 행렬을 행 사다리꼴 행렬(Row Echelon Form of Matrix)로 만들어 해를 구하는 방법.
- 연립 1차 방정식 풀이와 역행렬을 구할 때 사용되는 알고리즘.

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

단계

[STEP1] 전진 소거(forward elimination)
- 첨가행렬(Augmented matrix) 생성 후, 행 사다리꼴 형태(REF)로 만듦

[STEP2] 후진대입(Back substitution)
- 후진대입법을 사용하여 해를 구함

가우시안 소거법(Gaussian Elimination)

[STEP1] 전진 소거(forward elimination)

$$\begin{bmatrix}1 & 0 & 1\\0 & 1 & -1\\2 & 2 & 3\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}2\\1\\3\end{bmatrix}\rightarrow\begin{bmatrix}1 & 0 & 1 & 2\\0 & 1 & -1 & 1\\2 & 2 & 3 & 3\end{bmatrix}$$

- Augmented matrix: [계수행렬|상수항 벡터]
- pivot p : 대각행렬
- multiplier m : $\frac{pivot\text{ 아래 원소}}{pivot}$

$$p_1 = 1 \quad \rightarrow \quad m_{2,1} = \frac{0}{p_1} = 0, \qquad m_{3,1} = \frac{2}{p_1} = 2$$

$$R(2) - R(1) \times m_{2,1} \rightarrow R(2)$$
$$R(3) - R(1) \times m_{3,1} \rightarrow R(3)$$

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

$$p_2 = 1 \quad \rightarrow \quad m_{3,2} = \frac{2}{p_2} = 2$$

$$R(3) - R(2) \times m_{3,2} \rightarrow R(3)$$

REF 행렬

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

가우시안 소거법(Gaussian Elimination)

[STEP2] 후진 대입(Back substitution)

$$\begin{bmatrix}1 & 0 & 1\\0 & 1 & -1\\2 & 2 & 3\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}2\\1\\3\end{bmatrix}$$

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

$$3x_3 = -3 \quad \rightarrow \quad x_3 = -1$$

$$x_2 = x_3 + 1 \quad \rightarrow \quad x_2 = -1 + 1 = 0$$

$$x_1 = -x_3 + 2 \quad \rightarrow \quad x_2 = -(-1) + 2 = 3$$

$$\therefore x = \begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix} = \begin{bmatrix}3\\0\\-1\end{bmatrix}$$

LU-Decomposition 란?

LU-Decomposition

:임의의 행렬 A 가 존재할 때, A 를 두 행렬의 곱으로 분해하는 것

$$\text{“}\boldsymbol{A = LU}\text{”}$$

→ Gaussian Elimination을 이용하여 구할 수 있음. (without pivoting)

→ 행교환 없는 가우시안 소거법

LU-Decomposition 란?

*삼각행렬(triangular matrix): 주대각선을 기준으로 대각항의 위 or 아래쪽 항들의 값이 모두 0인 경우

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

L ower-triangular matrix **하삼각행렬**: 주대각선을 기준으로 대각항의 **위쪽 항**들의 값이 **모두** 0인 경우

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 \\ 4 & 5 & 6 & 0 \\ 7 & 8 & 9 & 8 \end{bmatrix}$$

U pper-triangular matrix **상삼각행렬**: 주대각선을 기준으로 대각항의 **아래쪽 항**들의 값이 **모두** 0인 경우

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 5 & 6 & 7 \\ 0 & 0 & 8 & 9 \\ 0 & 0 & 0 & 8 \end{bmatrix}$$

LU-Decomposition 란?

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 2 & 2 & 3 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad c = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$$

Upper-triangular matrix **상삼각행렬:** 첨가행렬을 생성하기 않고 가우시안 소거법 중 1단계 전진소거를 적용함.

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

Lower-triangular matrix **하삼각행렬:** $m_{2,1} = \frac{0}{p_1} = 0, \quad m_{3,1} = \frac{2}{p_1} = 2, \quad m_{3,2} = \frac{2}{p_2} = 2$

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

LU-Decomposition 란? [검증]

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 2 & 2 & 3 \end{bmatrix},\ x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix},\ c = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix},\quad L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix},\ U = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 3 \end{bmatrix}$$

Lower-triangular matrix 하삼각행렬:

$$\boldsymbol{Ly} = \boldsymbol{c} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$$

$$y_1 = 2$$
$$y_2 = 1$$
$$y_3 = 3 - (2y_1 + 2y_2) = 3 - 4 - 2 = -3$$

$$\therefore \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix}$$

Upper-triangular matrix 상삼각행렬:

$$\boldsymbol{Ux} = \boldsymbol{y} \rightarrow \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix}$$

$$x_3 = -1$$
$$x_2 = 1 + x_3 = 0$$
$$x_1 = 2 - x_3 = 2 - (-1) = 3$$

$$\therefore \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ -1 \end{bmatrix}$$

$$1 * 3 + 1 * (-1) = 2\ [True]$$
$$1 * 0 + (-1) * (-1) = 1\ [True]$$
$$2 * 3 + 2 * 0 + 3 * (-1) = 3\ [True]$$

PLU-Decomposition 란?

Permutation **치환행렬:** 순서가 부여된 임의의 행렬을 의도된 다른 순서로 뒤섞어주는 연산 행렬

[STEP1] 단위행렬 I를 재배치시킨 후 P로 선언함.

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

permutation $\sigma = \{4, 2, 3, 1\}$

[STEP2] 치환행렬 P와 재배열하고자 하는 임의의 행렬 A를 곱함.

$$A = \begin{array}{cc} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} & \\ \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}$$

열의 재배열

$$P \cdot A = \begin{bmatrix} d & b & c & a \\ h & f & g & e \\ l & j & k & i \\ p & n & o & m \end{bmatrix}$$

행의 재배열

$$A \cdot P = \begin{bmatrix} m & n & o & p \\ e & f & g & h \\ i & j & k & l \\ a & b & c & d \end{bmatrix}$$

SageMath

System for **A**lgebra and **G**eometry **E**xperimentation

- 수학의 다양한 분야의 기능을 갖춘 컴퓨터 대수학 시스템.

(대수학, 조합론, 그래프 이론, 수치해석, 수론, 미적분학, 통계학)

- MATLAB, Mathematica, Maple, Magma를 대체하는 오픈 소스 소프트웨어 개발을 목적으로 개발되었음. (released in 2005)

- SAGE 명령어 익히지 않아도 **CAS 도구**들의 언어를 그대로 SAGE에서 사용할 수 있음

- Python(기본언어), cython, C, C++, fortran

SageMath를 사용하여 PLU 행렬 구하기

https://sagecell.sagemath.org/

```
A = matrix([[a,b,c],[d,e,f],[g,h,i]])
print A

P, L, U = A.LU()
print "P = "  # 치환 행렬
print P
print
print "L = "  # 하삼각행렬
print L
print
print "U = "  # 상삼각행렬
print U
print
print A == P*L*U
```

알파벳 - 숫자(행렬의 성분/요소) 넣어야 함

P, L, U 구해주는 함수

PLU 분해가 잘 되었는지 검증

출처: http://matrix.skku.ac.kr/2018-album/LU-decom.html

감사합니다