

Lab 6: LU Decomposition

*Instructor: Manuel Charlemagne***SU2022**

Motivation

A very basic problem in Big Data is solving linear equations, e.g.

$$\begin{aligned} 2x_1 + x_2 + 3x_3 &= 5 \\ 4x_1 - x_2 + 3x_3 &= 7 \\ -2x_1 + 5x_2 + 5x_3 &= 3 \end{aligned}$$

We can write it as

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & -1 & 3 \\ -2 & 5 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \\ 3 \end{bmatrix}$$

or $\mathbf{Ax} = \mathbf{b}$. This is very slow when size of \mathbf{A} becomes large. We can think of a way that can write \mathbf{A} as $\mathbf{A} = \mathbf{LU}$, where \mathbf{L} is lower triangular and \mathbf{U} is upper triangular, such that the equation becomes

$$\mathbf{LUx} = \mathbf{b} \Rightarrow \text{solve } \mathbf{y} \text{ for } \mathbf{Ly} = \mathbf{b} \Rightarrow \text{solve } \mathbf{x} \text{ for } \mathbf{Ux} = \mathbf{y}.$$

which can be very easily done by backward substitution, as we can see.

$$\begin{bmatrix} 2 & 1 & 3 \\ 0 & -3 & -3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ -3 \\ 2 \end{bmatrix} \Rightarrow \mathbf{x} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

But how can this happen?

Basic LU Decomposition

Gaussian Elimination

A common solution to this problem is ==Gaussian Elimination==, take the above \mathbf{A} as an example,

- To make the first element on second line to be zero, we need $row_2 + = -2 * row_1$.

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & -1 & 3 \\ -2 & 5 & 5 \end{bmatrix} \sim \begin{bmatrix} 2 & 1 & 3 \\ 0 & -3 & -3 \\ -2 & 5 & 5 \end{bmatrix}$$

- To make the first element on third line to be zero, we need $row_3 + = row_1$.

$$\begin{bmatrix} 2 & 1 & 3 \\ 0 & -3 & -3 \\ -2 & 5 & 5 \end{bmatrix} \sim \begin{bmatrix} 2 & 1 & 3 \\ 0 & -3 & -3 \\ 0 & 6 & 8 \end{bmatrix}$$

- To make the second element on third line to be zero, we need $row_3 + = 2 * row_2$.

$$\begin{bmatrix} 2 & 1 & 3 \\ 0 & -3 & -3 \\ 0 & 6 & 8 \end{bmatrix} \sim \begin{bmatrix} 2 & 1 & 3 \\ 0 & -3 & -3 \\ 0 & 0 & 2 \end{bmatrix} = \mathbf{U}$$

And we have created an upper triangular matrix.

Elementary Matrices

We can use matrices to represent the Gaussian elimination steps. For example, doing $row_2 + = 2 * row_1$ is equivalent to doing a matrix multiplication,

$$\mathbf{E}_{(-2)1,2}\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} - & a_1 & - \\ - & a_2 & - \\ - & a_3 & - \end{bmatrix} = \begin{bmatrix} - & a_1 & - \\ - & -2 \times a_1 + a_2 & - \\ - & a_3 & - \end{bmatrix}$$

We can write that

$$\mathbf{E}_{(2)2,3}\mathbf{E}_{(1)1,3}\mathbf{E}_{(-2)1,2}\mathbf{A} = \mathbf{U}$$

We can then express \mathbf{A} using \mathbf{U} .

$$\begin{aligned} \mathbf{A} &= \mathbf{E}_{(-2)1,2}^{-1}\mathbf{E}_{(1)1,3}^{-1}\mathbf{E}_{(2)2,3}^{-1}\mathbf{U} \\ &= \mathbf{E}_{(2)1,2}\mathbf{E}_{(-1)1,3}\mathbf{E}_{(-2)2,3}\mathbf{U} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \mathbf{U} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -2 & 1 \end{bmatrix} \mathbf{U} \end{aligned}$$

And we get \mathbf{L} . Generally speaking, all invertible matrix has its LU decomposition.

Not all matrices can be done this way, and sometimes we need some other permutation matrix, but we won't go deep into that.

Cons: based on Gaussian elimination, maybe unstable and slow.

Cholesky Decomposition

If \mathbf{A} is symmetric positive definite,

$$\mathbf{A} = \mathbf{A}^T \quad \text{and} \quad \forall x \in \mathbb{R}^n \setminus \{0\} : x^T \mathbf{A} x > 0$$

Then we can write $\mathbf{A} = \mathbf{C}\mathbf{C}^T$.

$$\mathbf{A} = \begin{bmatrix} c_{11} & & & & \\ c_{21} & c_{22} & & & \\ c_{31} & c_{32} & c_{33} & & \\ \vdots & \vdots & \ddots & \ddots & \\ c_{n1} & c_{n2} & c_{n3} & \dots & c_{nn} \end{bmatrix} \begin{bmatrix} c_{11} & c_{21} & c_{31} & \dots & c_{n1} \\ & c_{22} & c_{32} & \dots & c_{n2} \\ & & c_{33} & \dots & c_{n3} \\ & & & \ddots & \vdots \\ & & & & c_{nn} \end{bmatrix}$$

So as we can see,

$$\begin{aligned}a_{11} &= c_{11}^2 \\a_{12} &= c_{11}c_{21} \\a_{22} &= c_{21}^2 + c_{22}^2\end{aligned}$$

and we can calculate the elements in **C** accordingly,

$$\begin{aligned}c_{11} &= \sqrt{a_{11}} \\c_{21} &= a_{12}/c_{11} \\c_{22} &= \sqrt{a_{22} - c_{21}^2}\end{aligned}$$

and so on... We can use the algorithm on the lab manual to calculate the whole matrix.

Submission

This is an individual lab. Please finish the implementation part. Submit your code and result on canvas before next week's lab.