SI231b: Matrix Computations

Lecture 7: LU Factorization with Pivoting

Yue Qiu

qiuyue@shanghaitech.edu.cn

School of Information Science and Technology

ShanghaiTech University

Oct. 9, 2021

Recap: LU Factorization with Partial Pivoting

Step k of LU factorization

1. row exchange: $\tilde{A}^{(k-1)} = P_k A^{(k-1)}$
2. Gaussian elimination: $A^{(k)} = M_k \tilde{A}^{(k-1)}$

In general, the procedure follows

$$M_{n-1}P_{n-1}M_{n-2}P_{n-2}\cdots M_1P_1A = U.$$

Denote

$$\begin{aligned}
\tilde{M}_{n-1} &= M_{n-1}, \\
\tilde{M}_{n-2} &= P_{n-1}M_{n-2}P_{n-1}^T, \\
\vdots \quad &= \quad \vdots \\
\tilde{M}_k &= P_{n-1}P_{n-2}\cdots P_{k+1}M_kP_{k+1}^T\cdots P_{n-2}^TP_{n-1}^T
\end{aligned}$$

Note: $\tilde{M}_k$ has the same structure with M_k (recall the structure of M_k)

A Simple Example

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

Step 1, 1st row $\longleftrightarrow$ 3rd row of A, then perform Gaussian elimination

$$\tilde{A}^{(0)} = P_1 A = \begin{bmatrix} & & 1 & \\ & 1 & & \\ 1 & & & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix}$$

$$A^{(1)} = M_1 \tilde{A}^{(0)} = \begin{bmatrix} 1 & & & \\ -\frac{1}{2} & 1 & & \\ -\frac{1}{4} & & 1 & \\ -\frac{3}{4} & & & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \end{bmatrix}$$

Step 2: 2nd row $\longleftrightarrow$ 4th row of $A^{(1)}$, then repeat Gaussian elimination

$$\tilde{A}^{(1)} = P_2 A^{(1)} = \begin{bmatrix} 1 & & & \\ & & & 1 \\ & & 1 & \\ & 1 & & \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \end{bmatrix}$$

$$A^{(2)} = M_2 \tilde{A}^{(1)} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & \frac{3}{7} & 1 & \\ & \frac{2}{7} & & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{2}{7} & \frac{4}{7} \\ & & -\frac{6}{7} & -\frac{2}{7} \end{bmatrix}$$

Now, it's your turn to give P_3, M_3 and the final P, L, and U

A Simple Example

$$\underbrace{\begin{bmatrix} & & 1 & \\ & & & 1 \\ & 1 & & \\ 1 & & & \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & & & \\ \frac{3}{4} & 1 & & \\ \frac{1}{2} & -\frac{2}{7} & 1 & \\ \frac{1}{4} & -\frac{3}{7} & \frac{1}{3} & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & & \frac{2}{3} \end{bmatrix}}_{U}$$

In practice, the permutation matrix P

- is not represented explicitly as a matrix or the product of permutation matrices
- an equivalent effect can be achieved via a permutation vector

Note: $|\ell_{ij}| \leq 1$ for $i \geq j$

Following the aforementioned procedure,

$$\mathrm{PA} = \mathrm{LU},$$

where

- $\mathrm{P} = \mathrm{P}_{n-1}\mathrm{P}_{n-2}\cdots\mathrm{P}_1$ is again a permutation matrix (why?)
- $\mathrm{L} = \left(\tilde{\mathrm{M}}_{n-1}\tilde{\mathrm{M}}_{n-2}\cdots\tilde{\mathrm{M}}_1\right)^{-1}$ is a lower-triangular matrix with unit diagonals

$$\tilde{\mathrm{M}}_k = \tilde{\mathrm{P}}_{k+1}\mathrm{M}_k\tilde{\mathrm{P}}_{k+1}^T = \mathrm{I} - \tilde{\mathrm{P}}_{k+1}\tau^{(k)}\mathrm{e}_k^T = \mathrm{I} - \tilde{\mathrm{P}}_{k+1}\mathrm{M}_k \qquad \textit{why?}$$

Here $\tilde{\mathrm{P}}_{k+1} = \mathrm{P}_{n-1}\mathrm{P}_{n-2}\cdots\mathrm{P}_{k+1}$

Then, we obtain

$$\tilde{\mathrm{M}}_k^{-1} = \mathrm{I} + \tilde{\mathrm{P}}_{k+1}\mathrm{M}_k$$

$$\mathrm{L} = \mathrm{I} + \sum_{i=1}^{n-1}\tilde{\mathrm{P}}_{k+1}\mathrm{M}_k$$

An Alternative Approach for LU Factorization with Partial Pivoting

For $A \in \mathbb{R}^{n\times n}$, and a permutation matrix P_1

$$P_1 A = \left[\begin{array}{c|c} a_{11}^{(0)} & v^T \\ \hline u & A_1' \end{array}\right] = \underbrace{\left[\begin{array}{c|c} 1 & 0 \\ \hline 1/a_{11}^{(0)} u & I_{n-1} \end{array}\right]}_{L_1} \underbrace{\left[\begin{array}{c|c} a_{11}^{(0)} & v^T \\ \hline 0 & A_1' - 1/a_{11}^{(0)} uv^T \end{array}\right]}_{U_1}$$

Then repeat the above procedure to $A_1' - 1/a_{11}^{(0)} uv^T$, i.e.,

$$\begin{aligned} P_2' \left(A_1' - 1/a_{11}^{(0)} uv^T\right) &= \left[\begin{array}{c|c} a_{22}^{(1)} & w^T \\ \hline s & A_2' \end{array}\right] \\ &= \left[\begin{array}{c|c} 1 & 0 \\ \hline 1/a_{22}^{(1)} s & I_{n-2} \end{array}\right] \left[\begin{array}{c|c} a_{22}^{(1)} & w^T \\ \hline 0 & A_2' - 1/a_{22}^{(1)} sw^T \end{array}\right] \end{aligned}$$

Denote $P_2 = \begin{bmatrix} 1 & \\ & P_2' \end{bmatrix}$, we obtain (next page)

$$\mathrm{P}_2\mathrm{P}_1\mathrm{A} = \underbrace{\begin{bmatrix} 1 & & \\ & 1 & \\ \frac{1}{a_{11}^{(0)}}\mathrm{P}_2'\mathrm{u} & \frac{1}{a_{22}^{(1)}}\mathrm{s} & \mathrm{I}_{n-2} \end{bmatrix}}_{\mathrm{L}_2} \underbrace{\begin{bmatrix} a_{11}^{(0)} & & \mathrm{v}^T \\ & a_{22}^{(1)} & \mathrm{w}^T \\ & & \mathrm{A}_2' - \frac{1}{a_{22}^{(1)}}\mathrm{s}\mathrm{w}^T \end{bmatrix}}_{\mathrm{U}_2}$$

- following the above notations, $\mathrm{L} = \mathrm{L}_{n-1}$, $\mathrm{U} = \mathrm{U}_{n-1}$
- P_k only acts on the first $(k-1)$ columns of L_k
- algorithm style, suitable for computer implementation

Remark:

- Gaussian elimination tells why you can perform an LU factorization, and when does it exist
- the recursive approach tells how you can compute the LU factorization on a modern computer

Example

Please compute an LU factorization with partial pivoting using the method introduced in the last page for

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

LU with complete pivoting:

In matrix form, at each stage before Gaussian elimination

- permutation of rows with P_k on the left
- permutation of columns with Q_k on the right

$$M_{n-1}P_{n-1}M_{n-2}P_{n-2}\cdots M_1P_1AQ_1Q_2\cdots Q_{n-1} = U.$$

By

- using the same definition of L, P with LU factorization with partial pivoting,
- denoting $Q = Q_1Q_2\cdots Q_{n-1}$,

the LU factorization with complete pivoting can be represented by

$$PAQ = LU$$

Too computationally expensive, why?

LU Factorization without Pivoting:

U = A, L = I;
for k = 1 : n-1
  for j = k+1 : n
    $\ell_{jk} = u_{jk} / u_{kk}$
    $u_{j,k:n} = u_{j,k:n} - \ell_{jk} u_{k,k:n}$
  end
end
U = triu(U)

Operations count:

- $\mathcal{O}\left(\frac{2}{3}n^3\right)$ flops

Please give your own explanation

LU Factorization with Partial Pivoting:

```
U = A, L = I, P = I;
for k = 1 :  n-1
     select i ≥ k to maximize |u_ik|
     u_{k,k:m} ↔ u_{i,k:m}  (exchange of rows)
     ℓ_{k,1:k-1} ↔ ℓ_{i,1:k-1}
     p_{k,:} ↔ p_{i,:}
     for j = k+1 :  n
          ℓ_jk = u_jk / u_kk
          u_{j,k:n} = u_{j,k:n} - ℓ_jk u_{k,k:n}
     end
end
U = triu(U)
```

Operations count:

- $\mathcal{O}\left(\frac{2}{3}n^3\right)$ flops, flops count of partial pivoting?