SI231b: Matrix Computations

Lecture 8: Special LU Factorization and Computational Complexity

Yue Qiu

qiuyue@shanghaitech.edu.cn

School of Information Science and Technology

ShanghaiTech University

Oct. 8, 2022

Recap: LU Factorization with Partial Pivoting Through Recursion

For $\mathbf{A} \in \mathbb{R}^{n\times n}$, and a permutation matrix $\mathbf{P}_1$

$$\mathbf{P}_1\mathbf{A} = \left[\begin{array}{c|c} a_{11}^{(0)} & \mathbf{v}^T \\ \hline \mathbf{u} & \mathbf{A}_1' \end{array}\right] = \underbrace{\left[\begin{array}{c|c} 1 & 0 \\ \hline 1/a_{11}^{(0)}\mathbf{u} & \mathbf{I}_{n-1} \end{array}\right]}_{\mathbf{L}_1} \underbrace{\left[\begin{array}{c|c} a_{11}^{(0)} & \mathbf{v}^T \\ \hline 0 & \mathbf{A}_1' - 1/a_{11}^{(0)}\mathbf{u}\mathbf{v}^T \end{array}\right]}_{\mathbf{U}_1}$$

Then repeat the above procedure to $\mathbf{A}_1' - 1/a_{11}^{(0)}\mathbf{u}\mathbf{v}^T$, i.e.,

$$\begin{aligned} \mathbf{P}_2'\left(\mathbf{A}_1' - 1/a_{11}^{(0)}\mathbf{u}\mathbf{v}^T\right) &= \left[\begin{array}{c|c} a_{22}^{(1)} & \mathbf{w}^T \\ \hline \mathbf{s} & \mathbf{A}_2' \end{array}\right] \\ &= \left[\begin{array}{c|c} 1 & 0 \\ \hline 1/a_{22}^{(1)}\mathbf{s} & \mathbf{I}_{n-2} \end{array}\right] \left[\begin{array}{c|c} a_{22}^{(1)} & \mathbf{w}^T \\ \hline 0 & \mathbf{A}_2' - 1/a_{22}^{(1)}\mathbf{s}\mathbf{w}^T \end{array}\right] \end{aligned}$$

Denote $\mathbf{P}_2 = \begin{bmatrix} 1 & \\ & \mathbf{P}_2' \end{bmatrix}$, we obtain (next page)

Recap: LU Factorization with Partial Pivoting Through Recursion

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

- following the above notations, $\mathbf{L} = \mathbf{L}_{n-1}$, $\mathbf{U} = \mathbf{U}_{n-1}$
- $\mathbf{P}_k$ only acts on the first $(k-1)$ columns of $\mathbf{L}_k$
- 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

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

Answer:

$$\begin{bmatrix} & & 1 \\ 1 & & \\ & 1 & \end{bmatrix} \begin{bmatrix} 2 & 4 & 5 \\ -3 & 1 & 4 \\ 4 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & & \\ \frac{1}{2} & 1 & \\ -\frac{3}{4} & \frac{5}{6} & 1 \end{bmatrix} \begin{bmatrix} 4 & 2 & 3 \\ & 3 & \frac{7}{2} \\ & & \frac{10}{3} \end{bmatrix}$$

LU with complete pivoting:

In matrix form, at each stage before Gaussian elimination

- permutation of rows with $\mathbf{P}_k$ on the left
- permutation of columns with $\mathbf{Q}_k$ on the right

$$\mathbf{M}_{n-1}\mathbf{P}_{n-1}\mathbf{M}_{n-2}\mathbf{P}_{n-2}\cdots\mathbf{M}_1\mathbf{P}_1\mathbf{A}\mathbf{Q}_1\mathbf{Q}_2\cdots\mathbf{Q}_{n-1} = \mathbf{U}.$$

By

- using the same definition of $\mathbf{L}$, $\mathbf{P}$ with LU factorization with partial pivoting,
- denoting $\mathbf{Q} = \mathbf{Q}_1\mathbf{Q}_2\cdots\mathbf{Q}_{n-1}$,

the LU factorization with complete pivoting can be represented by

$$\mathbf{PAQ} = \mathbf{LU}$$

Too computationally expensive, why?

LU Factorization without Pivoting:

```
U = A, L = I;
for k = 1 : n-1
     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

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?

$\mathbf{LDL}^T$ Factorization for Symmetric Matrices

Theorem

If $\mathbf{A} \in \mathbb{R}^{n \times n}$ is symmetric and nonsingular, and every leading principal sub-matrix $\mathbf{A}_{\{1,\ldots,k\}}$ satisfies

$$\det(\mathbf{A}_{\{1,\ldots,k\}}) \neq 0,$$

for $k = 1, 2, \cdots, n-1$, then there exists a lower-triangular matrix $\mathbf{L}$ with unit entries and a diagonal matrix

$$\mathbf{D} = \mathrm{diag}(d_1, d_2, \cdots, d_n),$$

where $d_i \neq 0$ for $i = 1, 2, \cdots, n$, such that $\mathbf{A} = \mathbf{LDL}^T$. The factorization is unique.

Proof: making use of the LU factorization

Computational complexity: not surprisingly $\mathcal{O}\left(\frac{n^3}{3}\right)$

Symmetry is preferred

If $\mathbf{A}$ is symmetric, and $\mathbf{P}_1$ is a permutation matrix

- $\mathbf{P}_1\mathbf{A}$ is not symmetric
- $\mathbf{P}_1\mathbf{A}\mathbf{P}_1^T$ is symmetric

Consider the following

$$\begin{aligned}\mathbf{P}_1\mathbf{A}\mathbf{P}_1^T &= \begin{bmatrix} \alpha & \mathbf{v}^T \\ \mathbf{v} & \mathbf{A}_1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & \\ 1/\alpha\mathbf{v} & \mathbf{I}_{n-1} \end{bmatrix} \begin{bmatrix} \alpha & \\ & \tilde{\mathbf{A}}_1 \end{bmatrix} \begin{bmatrix} 1 & 1/\alpha\mathbf{v}^T \\ & \mathbf{I}_{n-1} \end{bmatrix},\end{aligned}$$

with $\tilde{\mathbf{A}}_1 = \mathbf{A}_1 - 1/\alpha\mathbf{v}\mathbf{v}^T$ also symmetric.

Note: with symmetric pivoting, α is some diagonal entry a_{ii}, why?

When the procedure terminates, $\mathbf{PAP}^T = \mathbf{LDL}^T$ where

$$\mathbf{P} = \mathbf{P}_{n-1} \cdots \mathbf{P}_2\mathbf{P}_1$$

Symmetric Positive Definite (SPD)

$\mathbf{M} = \mathbf{M}^T \in \mathbb{R}^{n\times n}$ is SPD iff (if and only if)

$$\mathbf{x}^T\mathbf{M}\mathbf{x} > 0, \quad \forall \mathbf{x} \in \mathbb{R}^n \backslash \mathbf{0}$$

Properties of SPD Matrices:

- real positive eigenvalues
- positive diagonal entries
- all principle sub-matrices are SPD
- $\mathbf{A} \in \mathbb{R}^{n\times n}$ is SPD and $\mathbf{X} \in \mathbb{R}^{n\times r}$ has full rank, then $\mathbf{X}^T\mathbf{A}\mathbf{X}$ is also SPD

Recursive Factorization

For an SPD matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$,

$$\mathbf{A} = \begin{bmatrix} a_{11} & \mathbf{w}^T \\ \mathbf{w} & \mathbf{A}_1 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} \sqrt{a_{11}} & \\ 1/\sqrt{a_{11}}\mathbf{w} & \mathbf{I}_{n-1} \end{bmatrix}}_{\mathbf{L}_1} \underbrace{\begin{bmatrix} 1 & \\ & \mathbf{A}_1 - 1/a_{11}\mathbf{w}\mathbf{w}^T \end{bmatrix}}_{\mathbf{D}_1} \underbrace{\begin{bmatrix} \sqrt{a_{11}} & 1/\sqrt{a_{11}}\mathbf{w}^T \\ & \mathbf{I}_{n-1} \end{bmatrix}}_{\mathbf{L}_1^T}$$

Require: the (1, 1) entry of $\left(\mathbf{A}_1 - 1/a_{11}\mathbf{w}\mathbf{w}^T\right)$ should be positive to continue.

Note: $\left(\mathbf{A}_1 - 1/a_{11}\mathbf{w}\mathbf{w}^T\right)$ is a principle sub-matrix of $\mathbf{L}_1^{-1}\mathbf{A}\mathbf{L}_1^{-T}$.

Following the same principle, when the procedure terminates,

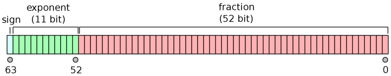
- $\mathbf{L}_n = \mathbf{L}$, $\mathbf{D}_n = \mathbf{I}_n$
- $\mathbf{A} = \mathbf{L}\mathbf{L}^T$: Cholesky factorization
- $\mathcal{O}\left(\frac{1}{3}n^3\right)$ flops, half of LU factorization

IEEE Standard for Floating-Point Arithmetic (IEEE 754)

- single format, 32 bit
- double format, 64 bit

Take the double format for example,

- 1 bit for sign;
- 52 bits for the mantissa;
- 11 bits for the exponent;



IEEE standard stipulates that each arithmetic operation be correctly rounded, meaning that the computed result is the rounded version of the exact result.

Machine Precision

Resolution is traditionally summarized by a number known as machine epsilon, i.e., ε_m

$$\varepsilon_m = \frac{1}{2} \times (\text{gap between 1 and next largest floating point number})$$

- $\varepsilon_m \approx 5.96 \times 10^{-8}$ for single format
- $\varepsilon_m \approx 1.11 \times 10^{-16}$ for double format

Try the eps command in Matlab to get ε_m

Property

$$\forall x \in \mathbb{R}, \text{there exists } x' \in \mathbb{F}, \text{ such that } |x - x'| < \varepsilon_m |x|$$

where $\mathbb{F}$ represents the set of floating point numbers. Or equivalently,

$$\forall x \in \mathbb{R}, \text{ there exists } \varepsilon \text{ with } |\varepsilon| \leq \varepsilon_m, \text{ such that } fl(x) = x(1+\varepsilon)$$

Matrix Condition Number

Consider solving the linear equation $\mathbf{Ax} = \mathbf{b}$ using direct methods, such as LUP/Cholesky factorization, which can be represented by

$$(\mathbf{A} + \sigma\mathbf{A})\,(\mathbf{x} + \sigma\mathbf{x}) = \mathbf{b}.$$

Making use of $\mathbf{Ax} = \mathbf{b}$ and dropping out the product $\sigma\mathbf{A}\sigma\mathbf{x}$, we obtain

$$\frac{\|\sigma\mathbf{x}\|}{\|\mathbf{x}\|} \Big/ \frac{\|\sigma\mathbf{A}\|}{\|\mathbf{A}\|} \le \|\mathbf{A}\|\|\mathbf{A}^{-1}\|$$

where $\|\mathbf{A}\|\|\mathbf{A}^{-1}\|$ defines the condition number of the matrix $\mathbf{A}$ and is often denoted by $\kappa(\mathbf{A})$.

The linear equation $\mathbf{Ax} = \mathbf{b}$ is

- well-conditioned if small $\sigma\mathbf{A}$ leads to small $\sigma\mathbf{x}$ (small $\kappa(\mathbf{A})$)
- ill-conditioned if small $\sigma\mathbf{A}$ leads to large $\sigma\mathbf{x}$ (large $\kappa(\mathbf{A})$)

Note: here the meaning of ''small'' and''large'' depends on the application.

Readings

You are supposed to read

- Gene H. Golub and Charles F. Van Loan. *Matrix Computations*, Johns Hopkins University Press, 2013.

  Chapter 2.6 – 2.7, Chapter 4.1 – 4.4

- Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*, SIAM, 1997.

  Lecture 12 – 13, 23