Lecture 7

Solving systems of linear equations III

Dr. Zhiguo Long

Zonghelou 718

Please contact me via QQ or Email

Contents

The cost of Gaussian Elimination

- Gaussian elimination (GE) is unnecessarily expensive when it is applied to many systems of equations with the same matrix A but different right-hand sides $\vec{b}$.
  - The forward elimination process is the most computationally expensive part at $O(n^3)$ but is exactly the same for any choice of $\vec{b}$.
  - In contrast, the solution of the resulting upper triangular system only requires $O(n^2)$ operations.
- We can use this information to improve the way in which we solve multiple systems of equations with the same matrix A but different right-hand sides $\vec{b}$.

Elementary row operations (EROs)

Note that the EROs discussed in the last lecture can be produced by left multiplication with a suitable matrix:

- Row swap:

$$\begin{pmatrix}1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 0\end{pmatrix}\begin{pmatrix}a & b & c\\ d & e & f\\ g & h & i\end{pmatrix}=\begin{pmatrix}a & b & c\\ g & h & i\\ d & e & f\end{pmatrix}$$

- Row swap:

$$\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}a & b & c & d\\ e & f & g & h\\ i & j & k & l\\ m & n & o & p\end{pmatrix}=\begin{pmatrix}a & b & c & d\\ i & j & k & l\\ e & f & g & h\\ m & n & o & p\end{pmatrix}$$

Elementary row operations (cont.)

- Multiply row by α:

$$\begin{pmatrix} \alpha & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = \begin{pmatrix} \alpha a & \alpha b & \alpha c \\ d & e & f \\ g & h & i \end{pmatrix}$$

- $\alpha \times \text{row } p + \text{row } q$:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \alpha & 0 & 1 \end{pmatrix} \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ \alpha a + g & \alpha b + h & \alpha c + i \end{pmatrix}$$

LU factorisation

- Recall from the last lecture that Gaussian elimination (GE) is just a sequence of EROs.
- Each of these EROs is equivalent to (left) multiplication by a suitable matrix, E say.
- Hence, forward elimination applied to the system $A\vec{x} = \vec{b}$ can be expressed as

$$(E_m \cdots E_1)A\vec{x} = (E_m \cdots E_1)\vec{b}, \tag{1}$$

  where m is the number of EROs required to reduce to the upper triangular form.
- Let $U = (E_m \cdots E_1)A$ and $L = (E_m \cdots E_1)^{-1}$.

LU factorisation (cont.)

- Now the original system $A\vec{x} = \vec{b}$ is equivalent to

$$LU\vec{x} = b \tag{2}$$

  where U is *upper triangular* (by construction) and L may be shown to be lower triangular (provided the EROs do not include any row swaps).

- Once L and U are known it is easy to solve (2):
  - Solve $L\vec{z} = \vec{b}$ in $O(n^2)$ operations.
  - Solve $U\vec{x} = \vec{z}$ in $O(n^2)$ operations.

- L and U may be found in $O(n^3)$ operations by performing GE and saving the E_i matrices, however it is more convenient to find them directly (also $O(n^3)$ operations).

Computing L and U

Consider a general 4×4 matrix A and its factorisation LU:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix}$$

For the first column,

$$\begin{aligned} a_{11} &= (1,0,0,0)(u_{11},0,0,0)^T &&= u_{11} &&\to u_{11} = a_{11} \\ a_{21} &= (l_{21},1,0,0)(u_{11},0,0,0)^T &&= l_{21}u_{11} &&\to l_{21} = a_{21}/u_{11} \\ a_{31} &= (l_{31},l_{32},1,0)(u_{11},0,0,0)^T &&= l_{31}u_{11} &&\to l_{31} = a_{31}/u_{11} \\ a_{41} &= (l_{41},l_{42},l_{43},1)(u_{11},0,0,0)^T &&= l_{41}u_{11} &&\to l_{41} = a_{41}/u_{11} \end{aligned}$$

The second, third and fourth columns follow in a similar manner, giving all the entries in L and U.

Notes

- L is assumed to have 1's on the diagonal, to ensure that the factorisation is unique.
- The process involves division by the diagonal entries u_{11}, u_{22}, etc., so they **must** be non-zero.
- In general the factors l_{ij} and u_{ij} are calculated for each column j in turn, i.e.,

```
for j in range(n):
  for i in range(j+1):
      # Compute factors u_{ij}
      ...
  for i in range(j+1, n):
      # Compute factors l_{ij}
      ...
```

Example 1

Use LU factorisation to solve the linear system of equations given by

$$\begin{pmatrix} 2 & 1 & 4 \\ 1 & 2 & 2 \\ 2 & 4 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 12 \\ 9 \\ 22 \end{pmatrix}.$$

This can be rewritten in the form $A = LU$ where

$$\begin{pmatrix} 2 & 1 & 4 \\ 1 & 2 & 2 \\ 2 & 4 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}.$$

Example 1

$$\begin{pmatrix} 2 & 1 & 4 \\ 1 & 2 & 2 \\ 2 & 4 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}.$$

Example 1

$$\begin{pmatrix} 2 & 1 & 4 \\ 1 & 2 & 2 \\ 2 & 4 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}.$$

Column 1 of A gives

$$\begin{aligned} A_{11} = 2 = & \quad u_{11} \to & \quad u_{11} = 2 \\ A_{21} = 1 = & \quad l_{21}u_{11} \to & \quad l_{21} = 0.5 \\ A_{31} = 2 = & \quad l_{31}u_{11} \to & \quad l_{31} = 1. \end{aligned}$$

Example 1

$$\begin{pmatrix} 2 & 1 & 4 \\ 1 & 2 & 2 \\ 2 & 4 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}.$$

Column 2 of A gives

$$\begin{aligned} A_{12} = 1 = & \quad & u_{12} \rightarrow & \quad & u_{12} = 1 \\ A_{22} = 2 = & \quad & l_{21}u_{12} + u_{22} \rightarrow & \quad & u_{22} = 1.5 \\ A_{32} = 4 = & \quad & l_{31}u_{12} + l_{32}u_{22} \rightarrow & \quad & l_{32} = 2. \end{aligned}$$

Example 1

$$\begin{pmatrix} 2 & 1 & 4 \\ 1 & 2 & 2 \\ 2 & 4 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}.$$

Column 3 of A gives

$$\begin{aligned} A_{13} = 4 = &\quad u_{13} \rightarrow &\quad u_{13} = 4 \\ A_{23} = 2 = &\quad l_{21}u_{13} + u_{23} \rightarrow &\quad u_{23} = 0 \\ A_{33} = 6 = &\quad l_{31}u_{13} + l_{32}u_{23} \rightarrow &\quad u_{33} = 2. \end{aligned}$$

Example 1

Solve the lower triangular system $L\vec{z} = \vec{b}$:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} 12 \\ 9 \\ 22 \end{pmatrix} \rightarrow \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} 12 \\ 3 \\ 4 \end{pmatrix}$$

Solve the upper triangular system $U\vec{x} = \vec{z}$:

$$\begin{pmatrix} 2 & 1 & 4 \\ 0 & 1.5 & 0 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \begin{pmatrix} 12 \\ 3 \\ 4 \end{pmatrix} \rightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}.$$

Example 2

Rewrite the matrix A as the product of lower and upper triangular matrices where

$$A = \begin{pmatrix} 4 & 2 & 0 \\ 2 & 3 & 1 \\ 0 & 1 & 2.5 \end{pmatrix}.$$

The link

The first example gives

$$\begin{pmatrix} 2 & 1 & 4 \\ 1 & 2 & 2 \\ 2 & 4 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 4 \\ 0 & 1.5 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

Note that

- the matrix U is the same as the fully eliminated upper triangular form produced by Gaussian elimination;
- L contains the multipliers that were used at each stage to eliminate the rows.

Further reading

- Wikipedia: LU decomposition
- Wikipedia: Matrix decomposition (Other examples of decompositions).
- Nick Higham: What is an LU factorization? (a very mathematical treatment with additional references)

Note that these implementations use additional pivoting to achieve better results. We tackle this in the next section.

- LAPACK: `dgetrf()`. (Implementation of LU factorisation from LAPACK).
- Scipy: `scipy.linalg.lu`