20. Linear algebra II: Gaussian elimination and LU factorization

Last time

- Review of linear algebra
- Vectors
- Linear maps
- Matrices

Goals for today

- Solving systems of linear equations
- Gaussian elimination
- LU factorization

Systems of linear equations

- Recall from last lecture that the following are equivalent:
  - Vector equation: $x_1\mathbf{a_1} + x_2\mathbf{a_2} = \mathbf{b}$

Systems of linear equations

- Recall from last lecture that the following are equivalent:
  - Vector equation: $x_1\mathbf{a_1} + x_2\mathbf{a_2} = \mathbf{b}$
  - Matrix equation: $\mathsf{A}\,\mathbf{x} = \mathbf{b}$

  where $\mathsf{A} = \begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix}$ and $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

Systems of linear equations

- Recall from last lecture that the following are equivalent:
  - Vector equation: $x_1\mathbf{a_1} + x_2\mathbf{a_2} = \mathbf{b}$
  - Matrix equation: $\mathsf{A}\,\mathbf{x} = \mathbf{b}$

    where $\mathsf{A} = \begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix}$ and $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$
  - System of linear equations

$$\begin{aligned} a_{1,1}x_1 + a_{1,2}x_2 &= b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 &= b_2 \end{aligned}$$

Systems of linear equations

- Recall from last lecture that the following are equivalent:
  - Vector equation: $x_1\mathbf{a_1} + x_2\mathbf{a_2} = \mathbf{b}$
  - Matrix equation: $\mathrm{A}\,\mathbf{x} = \mathbf{b}$

    where $\mathrm{A} = \begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix}$ and $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$
  - System of linear equations

$$\begin{aligned} a_{1,1}x_1 + a_{1,2}x_2 &= b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 &= b_2 \end{aligned}$$

- Matrix notation is best in higher dimensions

Solving linear equations

- Consider the system

$$
\begin{aligned}
x + 3y &= 7 \\
2x - y &= 0
\end{aligned}
$$

- We want to **solve** this system

Solving linear equations

- Consider the system

$$\begin{aligned} x + 3y &= 7 \\ 2x - y &= 0 \end{aligned}$$

- We want to **solve** this system

- **Solution**: vector **x** $= (x, y)$ satisfying both equations simultaneously

Solving linear equations

- Consider the system

$$\begin{aligned} x + 3y &= 7 \\ 2x - y &= 0 \end{aligned}$$

- We want to **solve** this system

- **Solution**: vector $\mathbf{x} = (x, y)$ satisfying both equations simultaneously

- **Idea**: **Eliminate** one of the variables

Elimination / row reduction

- Want to **eliminate** (remove) one variable from one equation
- Number equations as E_1, E_2, etc.

Elimination / row reduction

- Want to **eliminate** (remove) one variable from one equation
- Number equations as E_1, E_2, etc.

- Eliminate x by adding multiple of E_1 to E_2
- Form new equation $E_2' := E_2 + \alpha E_1$

Elimination / row reduction

- Want to **eliminate** (remove) one variable from one equation
- Number equations as E_1, E_2, etc.

- Eliminate x by adding multiple of E_1 to E_2
- Form new equation $E_2' := E_2 + \alpha E_1$

- Choose α to make coefficient of x in the result equal to 0
- Gives **equivalent** system

Elimination II

- Act on matrix A *and* right-hand side b in *same* way

Elimination II

- Act on matrix A *and* right-hand side b in *same* way
- Form **augmented matrix** by adjoining b to A:

$$\left[\begin{array}{cc|c} 1 & 3 & 7 \\ 2 & -1 & 0 \end{array}\right]$$

Elimination II

- Act on matrix A *and* right-hand side b in *same* way
- Form **augmented matrix** by adjoining b to A:

$$\left[\begin{array}{cc|c} 1 & 3 & 7 \\ 2 & -1 & 0 \end{array}\right]$$

- Take

$$\alpha = -\frac{a_{2,1}}{a_{1,1}}$$

so that $a'_{2,1} = a_{2,1} + \alpha\, a_{1,1} = 0$

Elimination III

- Applying the above **row operation** gives

$$\left[\begin{array}{cc|c} 1 & 3 & 7 \\ 0 & -7 & -14 \end{array}\right]$$

Elimination III

- Applying the above **row operation** gives

$$\left[\begin{array}{cc|c} 1 & 3 & 7 \\ 0 & -7 & -14 \end{array}\right]$$

- 2nd row shows that $-7y = -14$, so $y = 2$
- Then **backsubstitute** to find x:

  $x + 3y = 7$, so $x + 6 = 7$, so $x = 1$

Elimination IV

- Generalise to matrix of any size

Elimination IV

- Generalise to matrix of any size

- Apply operations until get **upper-triangular matrix** U
- i.e. all entries below main diagonal are 0

Elimination IV

- Generalise to matrix of any size

- Apply operations until get **upper-triangular matrix** U
- i.e. all entries below main diagonal are 0

- Backsubstitution effectively does row operations to introduce zeros in upper triangular part

Several right-hand sides

- Suppose that we wish to solve $Ax = b$ with the *same* matrix A for several right-hand sides b
- We will execute *exact same sequence* of row operations to reduce to upper-triangular form

Several right-hand sides

- Suppose that we wish to solve $Ax = b$ with the *same* matrix A for several right-hand sides b
- We will execute *exact same sequence* of row operations to reduce to upper-triangular form

- Can we somehow use this fact by just recording the sequence of row operations?

Row operations as elementary matrices

- Applying a row operation to an (augmented) matrix A produces new (augmented) matrix
- Express as \$L_1 , A for suitable matrix L_1:

Row operations as elementary matrices

- Applying a row operation to an (augmented) matrix A produces new (augmented) matrix
- Express as \$L_1 , A for suitable matrix L_1:

- E.g. the row operation $E_2 \leftarrow E_2 + \alpha E_1$ is

$$L_1 = \begin{pmatrix} 1 & 0 \\ \alpha & 1 \end{pmatrix}$$

Row operations as elementary matrices

- Applying a row operation to an (augmented) matrix A produces new (augmented) matrix
- Express as $L_1 , A for suitable matrix L_1:

- E.g. the row operation $E_2 \leftarrow E_2 + \alpha E_1$ is

$$L_1 = \begin{pmatrix} 1 & 0 \\ \alpha & 1 \end{pmatrix}$$

- For a vector x, have $Lx = x_1 \ell_1 + x_2 \ell_2$ – lin. comb. of *columns* of L

Sequence of row operations

- Sequence of n row operations on matrix A gives

$$L_n L_{n-1} \cdots L_1 A = U$$

if the row operations perform row reduction

Sequence of row operations

- Sequence of n row operations on matrix A gives

$$L_n L_{n-1} \cdots L_1 A = U$$

  if the row operations perform row reduction

- Hence

$$A = L_1^{-1} L_2^{-1} \cdots L_n^{-1} U$$

- So $A = LU$ where L is lower-triangular

Sequence of row operations

- Sequence of n row operations on matrix A gives

$$L_n L_{n-1} \cdots L_1 A = U$$

  if the row operations perform row reduction

- Hence

$$A = L_1^{-1} L_2^{-1} \cdots L_n^{-1} U$$

- So $A = LU$ where L is lower-triangular
- Note that $L_1^{-1} = \begin{pmatrix} 1 & 0 \\ -\alpha & 1 \end{pmatrix}$

Structure of L

- L_k has 1s on main diagonal
- And nonzero entries below diagonal only on kth column

$$(L_k)_{i,k} = -\frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}}$$

where $A^{(k-1)}$ is matrix after $(k-1)$ steps

Structure of L

- L_k has 1s on main diagonal
- And nonzero entries below diagonal only on kth column

$$(L_k)_{i,k} = -\frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}}$$

where $A^{(k-1)}$ is matrix after $(k-1)$ steps

- $L = $L_n \cdots L_1$: below-diagonal entries are those of the individual Ls!

LU factorization

- Any square matrix has such an **LU factorization** (decomposition)
- If have calculated $LU = A$ then to solve:
  - Solve $L(Ux) = b$
  - Solve $Ly = b$
  - Then solve $Ux = y$
- Solving triangular systems is easy by forward- or back-substitution

Pivoting

- In elimination algorithm may need to divide by $a_{i,i} = 0$
- Clearly we cannot do this

Pivoting

- In elimination algorithm may need to divide by $a_{i,i} = 0$
- Clearly we cannot do this
- Solution: **pivot**: swap with row with largest $a_{j,i}$

Pivoting

- In elimination algorithm may need to divide by $a_{i,i} = 0$
- Clearly we cannot do this
- Solution: **pivot**: swap with row with largest $a_{j,i}$
- Must track which rows we swap using **permutation matrix** P:

$$PA = LU$$

Pivoting

- In elimination algorithm may need to divide by $a_{i,i} = 0$
- Clearly we cannot do this
- Solution: **pivot**: swap with row with largest $a_{j,i}$
- Must track which rows we swap using **permutation matrix** P:

$$PA = LU$$

- In fact for numerical stability we should *always* pivot

Summary

- Can solve linear equations by elimination / row reduction
- Equivalent to LU factorization