September 17, 2024

Announcements

- HW3
- Quizzes due on time
- Exam 1

Goals

- Solving → LU

Review

- $A = U \Sigma V^T$

  orth (U), orth (V)

  Σ: singular values ≥ 0, diag

  $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$

- $\|QA\|_2 = \|A\|_2$

- $\|A\|_2 = \|\cancel{U}\Sigma \cancel{V^T}\|_2 = \|\Sigma\|_2$

  $= \sigma_1$

Computing the 2-Norm

$A = U \Sigma V^T$

$A^{-1} = V \Sigma^{-1} U^T$

Using the SVD of A, identify the 2-norm.

$$\|A\|_2 = \|\cancel{U} \Sigma \cancel{V^T}\|_2 = \|\Sigma\|_2$$
$$= \sigma_1$$

Express the matrix condition number $\text{cond}_2(A)$ in terms of the SVD:

$$\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2 = \|\Sigma\|_2 \|\Sigma^{-1}\|_2 = \sigma_1 / \sigma_n$$

m $\left\| \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{pmatrix} \right\|_2$

Not a matrix norm: Frobenius

The 2-norm is very costly to compute. Can we make something simpler?

$$\|A\|_F = \sqrt{\sum_{i,j} A_{ij}^2}$$

Σ

What about its properties?

Not a matrix norm: Frobenius

The 2-norm is very costly to compute. Can we make something simpler?

$$\|A\|_F = \sqrt{\sum_{ij} A_{ij}^2}$$

What about its properties?

Satisfies the mat. norm properties.

- definiteness
- scaling
- triangle inequality
- submultiplicativity (proof via Cauchy-Schwarz)

$$\|A\| = \max_{\|x\|=1} \|Ax\|$$

Frobenius Norm: Properties

$$\|QA\|_F = \|A\|_F$$

Is the Frobenius norm induced by any vector norm?

$$\|I\|_F = \sqrt{n}$$

How does it relate to the SVD?

$$\|A\|_F = \|U\Sigma V^T\|_F = \|\Sigma\|_F = \sqrt{\sum \sigma_i^2}$$

Solving Systems: Simple cases

Solve $D\boldsymbol{x} = \boldsymbol{b}$ if D is diagonal. (Computational cost?)

$x_i = b_i / d_{ii}$ $\qquad O(n)$

Solve $Q\boldsymbol{x} = \boldsymbol{b}$ if Q is orthogonal. (Computational cost?)

$\cancel{Q^T Q}\, x = Q^T b = x$ $\qquad O(n^2)$

Given SVD $A = U\Sigma V^T$, solve $A\boldsymbol{x} = \boldsymbol{b}$. (Computational cost?)

$U \Sigma V^T \vec{x} = \vec{b}$

$\Sigma V^T \vec{x} = U^T \vec{b}$ $\qquad O(n^2)$

$V^T \vec{x} = \Sigma^{-1} U^T \vec{b}$

$\vec{x} = V \Sigma^{-1} U^T \vec{b}$

Solving Systems: Triangular matrices

Solve

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ & a_{22} & a_{23} & a_{24} \\ & & a_{33} & a_{34} \\ & & & a_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}.$$

$a_{33}z + a_{34}w = b_3$

$w = b_4/a_{44}$

Demo: Coding back-substitution [cleared]

What about non-triangular matrices?

Gauss elim.

Gaussian Elimination

$A = LU$

Demo: Vanilla Gaussian Elimination [cleared]

What do we get by doing Gaussian Elimination?

RREF

How is that different from being upper triangular?

LU not guaranteed to provide RREF.

What if we do not just eliminate downward but also upward?

$\left[\begin{array}{c|c} A & b \end{array}\right]$

LU Factorization

What is the LU factorization?

"$A = LU$"

L lower $\diagdown$ — $\operatorname{diag}(L) = \mathbb{1}$

U upper

Solving $A\boldsymbol{x} = \boldsymbol{b}$

Does LU help solve $A\boldsymbol{x} = \boldsymbol{b}$?

$\vec{y} = U\vec{x}$

$A\vec{x} = \vec{b}$

$LU\vec{x} = \vec{b} \quad \Leftrightarrow \quad L\vec{y} = \vec{b}$ (solve by fw. subst.) $O(n^2)$

$U\vec{x} = \vec{y}$ (solve by bw. subst.) $O(n^2)$

$O(n^2)$

Determining an LU factorization

$$A = \begin{pmatrix} a_{11} & \vec{a}_{12}^T \\ \vec{a}_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} 1 & \\ \vec{l}_{21} & L_{22} \end{pmatrix} \begin{pmatrix} u_{11} & \vec{u}_{12}^T \\ & U_{22} \end{pmatrix}$$

$$\begin{pmatrix} 1 & \\ \vec{l}_{21} & L_{22} \end{pmatrix} \begin{pmatrix} u_{11} & \vec{u}_{12}^T \\ & U_{22} \end{pmatrix} \qquad \begin{pmatrix} a_{11} & \vec{a}_{12}^T \\ \vec{a}_{21} & A_{22} \end{pmatrix}$$

$\cancel{l_{11}}\,1 \cdot u_{11} = a_{11}$

$\vec{u}_{12}^T = \vec{a}_{12}^T$

$u_{11} \cdot \vec{l}_{21} = \vec{a}_{21}$

$\vec{l}_{21} = \vec{a}_{21}/u_{11}$?

pivot

$\underbrace{\vec{l}_{21}\vec{u}_{12}^T}_{} + L_{22}U_{22} = A_{22}$

$L_{22}U_{22} = A_{22} - \vec{l}_{21}\vec{u}_{12}^T$

Demo: LU Factorization [cleared]

Computational Cost

What is the computational cost of multiplying two $n \times n$ matrices?

$O(n^3)$

- $u_{11} = a_{11}$, $\boldsymbol{u}_{12}^T = \boldsymbol{a}_{12}^T$.
- $\ell_{21} = \boldsymbol{a}_{21} / u_{11}$.
- $L_{22} U_{22} = A_{22} - \ell_{21} \boldsymbol{u}_{12}^T$.

What is the computational cost of carrying out LU factorization on an $n \times n$ matrix?

$O(n^3)$

Demo: Complexity of Mat-Mat multiplication and LU [cleared]

LU: Failure Cases?

Is LU/Gaussian Elimination bulletproof?

$$A = \begin{pmatrix} 0 & 1 \\ 2 & 1 \end{pmatrix}$$

$$\begin{pmatrix} u_{11} & u_{12} \\ & u_{22} \end{pmatrix}$$

$$\begin{pmatrix} 1 & \\ l_{21} & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 2 & 1 \end{pmatrix}$$

$$u_{11} = 0$$

$$0 = u_{11} \cdot l_{21} + 1 \cdot 0 = 2$$

Saving the LU Factorization

What can be done to get something *like* an LU factorization?

Partial pivoting $PA = LU$

Demo: LU Factorization with Partial Pivoting [cleared]

Saving the LU Factorization

What can be done to get something *like* an LU factorization?

Idea from linear algebra class: In Gaussian elimination, simply swap rows, equivalent linear system.

- Good idea: Swap rows if there's a zero in the way
- Even better idea: Find the largest entry (by absolute value), swap it to the top row.

The entry we divide by is called the *pivot*.

- Swapping rows to get a bigger pivot is called partial pivoting.
- Swapping rows *and columns* to get an even bigger pivot is called complete pivoting. (downside: additional $O(n^2)$ cost *per step* to find the pivot!)

Demo: LU Factorization with Partial Pivoting [cleared]