September $\cancel{17}$ 24, 2024

Announcements

- HW3

Goals

- LU
- Linear systems
- maybe LSQ?

Review

$A = LU$

$Ax = b \rightsquigarrow \underbrace{LU x}_{y} = b$

Saving the LU Factorization

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

$PA = LU \leftarrow$ partial

vs complete

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]

Cholesky: LU for Symmetric Positive Definite

LU *can* be used for SPD matrices. But can we do better?

$$A = LL^T \qquad A = A^T$$

$$\begin{array}{c} 1 \\ n-1 \end{array}\begin{pmatrix} l_{11} & \\ \vec{l}_{21} & L_{22} \end{pmatrix}\begin{pmatrix} l_{11} & \vec{l}_{21}^T \\ & L_{22} \end{pmatrix} = \begin{pmatrix} a_{11} & \vec{a}_{21}^T \\ \vec{a}_{21} & A_{22} \end{pmatrix}$$

$$\underbrace{\quad}_{1} \quad \underbrace{\quad}_{n-1}$$

$$l_{11}^2 = a_{11} \Leftrightarrow l_{11} = \sqrt{a_{11}}$$

$$l_{11}\vec{l}_{21} = \vec{a}_{21} \Leftrightarrow \vec{l}_{21} = \vec{a}_{21} / l_{11}$$

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

More cost concerns

What's the cost of solving $A\boldsymbol{x} = \boldsymbol{b}$?

$O(n^3)$ LU + fw/bw $O(n^2)$

What's the cost of solving $A\boldsymbol{x} = \boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n$?

$O(n^3)$ LU + $n \times$ (fw/bw) $O(n \cdot n^2)$

What's the cost of finding A^{-1}?

$AX = I$

$O(n^3)$

Cost: Worrying about the Constant, BLAS

$O(n^3)$ really means

$$\alpha \cdot n^3 + \beta \cdot n^2 + \gamma \cdot n + \delta.$$

All the non-leading and constants terms swept under the rug. But: at least the leading constant ultimately matters.

Shrinking the constant: surprisingly hard (even for 'just' matmul)

Idea: Rely on library implementation: *BLAS* (Fortran)

Level 1	$\boldsymbol{z} = \alpha \boldsymbol{x} + \boldsymbol{y}$	vector-vector operations $O(n)$ `?axpy`
Level 2	$\boldsymbol{z} = A\boldsymbol{x} + \boldsymbol{y}$	matrix-vector operations $O(n^2)$ `?gemv`
Level 3	$C = AB + \beta C$	matrix-matrix operations $O(n^3)$ `?gemm`, `?trsm`

Demo: BLAS Level 2 vs Level 3 [cleared]

LAPACK

LAPACK: Implements 'higher-end' things (such as LU) using BLAS
Special matrix formats can also help save const significantly, e.g.

- banded
- sparse
- symmetric
- triangular

Sample routine names:

- `dgesvd`, `zgesdd`
- `dgetrf`, `dgetrs`

LU on Blocks: The Schur Complement

Given a linear system

n $\quad$ A square

$$m \left[\begin{array}{cc|c} A & B & \boldsymbol{b}_1 \\ \cancel{C} & D & \boldsymbol{b}_2 \end{array}\right], \qquad - CA^{-1} \qquad 0 \quad 0$$

$$\left[\begin{array}{c|c} A & B \\ C & D \end{array}\right] x =$$

can we do 'block Gaussian elimination' to get a *block triangular matrix*?

$$\left[\begin{array}{cc|c} A & B & \boldsymbol{b}_1 \\ 0 & D - CA^{-1}B & \boldsymbol{b}_2 - CA^{-1}\boldsymbol{b}_1 \end{array}\right]$$

Schur complement.

$A: k \times k$

$C: (m-k) \times k$

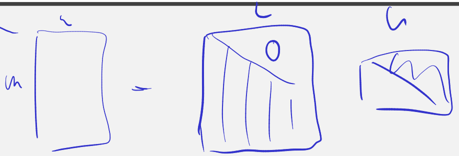
LU: Special cases

What happens if we feed a square non-invertible matrix to LU?

$PA = LU$ inv.

What happens if we feed LU an $m \times n$ non-square matrices?

"tall skinny"



$m > n$ $\quad (m \times n) \times (n \times n)$

Round-off Error in LU without Pivoting

Consider factorization of $\begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix}$ where $\epsilon < \epsilon_{\text{mach}}$:

$$L = \begin{pmatrix} 1 & \\ 1/\varepsilon & 1 \end{pmatrix} \qquad U = \begin{pmatrix} \varepsilon & 1 \\ 0 & 1 - \frac{1}{\varepsilon} \end{pmatrix}$$

$$fl(U) = \begin{pmatrix} \varepsilon & 1 \\ 0 & -\frac{1}{\varepsilon} \end{pmatrix}$$

$$L\, fl(U) = \begin{pmatrix} \varepsilon & 1 \\ 1 & 0 \end{pmatrix} \qquad \text{bw. err:} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

Round-off Error in LU with Pivoting

Permuting the rows of A in partial pivoting gives $PA = \begin{bmatrix} 1 & 1 \\ \epsilon & 1 \end{bmatrix}$

Changing matrices

Seen: LU cheap to re-solve if RHS changes. (Able to keep the expensive bit, the LU factorization) What if the *matrix* changes?

$$\tilde{A} = A + \vec{u}\vec{v}^T$$

$A = LU$

compute $A^{-1}\vec{z}$ in $O(n^2)$

Sherman-Morrison

$$(\underbrace{A + \vec{u}\vec{v}^T}_{\tilde{A}})^{-1} = A^{-1} - \frac{A^{-1}\vec{u}\vec{v}^T A^{-1}}{1 + v^T A^{-1} \vec{u}}$$

$$\tilde{A}\vec{x} = \vec{b} \iff \vec{x} = \tilde{A}^{-1}\vec{b}$$

$$A^{-1}\vec{b} - \frac{(A^{-1}\vec{u})\vec{v}^T(A^{-1}\vec{b})}{1 + v^T(A^{-1}\vec{u})}$$

$O(n^2)$

if parenthesized right.

Demo: Sherman-Morrison [cleared]

In-Class Activity: LU

In-class activity: LU and Cost