LU: Failure Cases?

Is LU/Gaussian Elimination bulletproof?

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

$$\begin{bmatrix} 1 & \\ l_{21} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ & u_{22} \end{bmatrix} = A$$

$$\Rightarrow u_{11} = 0, \quad \underline{\underline{u_{11}}} \cdot l_{21} + 1 \cdot 0 = 2$$

$$\Rightarrow 0 = 2$$

Saving the LU Factorization

$PA = LU$

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

- Same as Gauss elim: swap rows to move 0s out of the way
- In particular, find $\underbrace{\text{argmax}}_{\uparrow}(\text{abs}(A[:,1]))$ pivot.
- partial pivoting + $O(n^2)$
- complet pivoting + $O(n^3)$.

Demo: LU Factorization with Partial Pivoting [cleared]

More cost concerns

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

- LU: $O(n^3)$
- $O(n^2) + O(n^2)$

$> O(n^3)$

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

- LU: $O(n^3)$
- $n \cdot (O(n^2) + O(n^2)) = O(n^3)$

$> O(n^3)$

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

$A^{-1} A = I$

$A^T A^{-T} = I$

$O(n^3)$

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`

? = s, d, c, z

Show (using perf): `numpy` `matmul` calls BLAS `dgemm`

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

ge : general matrix

Sample routine names:

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

LU on Blocks: The Schur Complement

Given a matrix

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix},$$

can we do ‘block LU’ to get a *block triangular matrix*?

$$\begin{bmatrix} I & \\ -CA^{-1} & I \end{bmatrix}\begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} A & B \\ 0 & \underbrace{-CA^{-1}B+D}_{\text{Schur complement.}} \end{bmatrix}$$

LU: Special cases

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

$$\underset{\sim}{P}\,\underline{\underline{A}} = \underset{\sim}{L}\,\underline{\underline{U}}$$

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

- $\boxed{A}$ $m>n$: $\boxed{L}\,\boxed{U}$
- $\boxed{A}$ $m<n$: $\boxed{L}\,\boxed{U}$

Round-off Error in LU without Pivoting

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

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

$$fl(U) = \begin{bmatrix} \varepsilon & 1 \\ 0 & -1/\varepsilon \end{bmatrix} \qquad L \cdot fl(U) = \begin{bmatrix} \varepsilon & 1 \\ 1 & 0 \end{bmatrix}$$

$$err_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Round-off Error in LU with Pivoting

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

$$L = \begin{bmatrix} 1 & 0 \\ \varepsilon & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 1 \\ 0 & \underline{\underline{1-\varepsilon}} \end{bmatrix}$$

$$fl(U) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$L \cdot fl(U) = \begin{bmatrix} 1 & 1 \\ \varepsilon & 1+\varepsilon \end{bmatrix} \Rightarrow err_2 = \begin{bmatrix} 0 & 0 \\ 0 & \varepsilon \end{bmatrix}$$

$$err_1 = 1/\varepsilon \cdot err_2$$

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?

$$\hat{A} = A + uv^T$$

Sherman-Morrison formula,

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}$$

Demo: Sherman-Morrison [cleared]

In-Class Activity: LU

In-class activity: LU and Cost