HW4

Exam 1 starts Friday

Out of town 9/26, lecture as normal

Feedback
Computational Cost

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

\[ O(n^3) \]

- \( u_{11} = a_{11}, \ u_{12}^T = a_{12}^T \).
- \( \ell_{21} = a_{21}/u_{11} \).
- \( L_{22}U_{22} = A_{22} - \ell_{21}u_{12}^T \).

\[ \text{Cost} (n) = \frac{2}{3} n^3 + O(n) \]

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?

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

\[
\begin{pmatrix}
1 & 0 \\
2 & 1
\end{pmatrix}
\begin{pmatrix}
\nu & \nu_2 \\
0 & \nu
\end{pmatrix}
\]

\[
\nu_{11} = 0
\]

\[
\nu = 2 \\
0
\]

All 0,1 exactly one 1 per row/col

PA = LU

Permutation mat., permutes rows
Saving the LU Factorization

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

For numerical stability:

- Swap so that \( a_{11} \) is as big as possible.
- Typically sufficient if \( \frac{a_{11}}{\max_{j=2}^{n} a_{j1}} \leq \epsilon \).

\[
PA = LU
\]

PA = LU will row permutations; 

PAQ = LU  row col. permutations: "complete" pivoting

\( L \) adds a leading order \( O(n^3) \) cost

Demo: LU Factorization with Partial Pivoting [cleared]
Saving the LU Factorization

What can be done to get something \textit{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 \textit{pivot}.

- Swapping rows to get a bigger pivot is called \textit{partial pivoting}.
- Swapping rows \textit{and columns} to get an even bigger pivot is called \textit{complete pivoting}. (downside: additional $O(n^2)$ cost \textit{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?

\[ A = L L^T \]

\[
\begin{pmatrix}
    l_{11} & 0 \\
    l_{21} & l_{22}
\end{pmatrix}
\begin{pmatrix}
    a_{11} & a_{12}^T \\
    a_{21} & a_{22}
\end{pmatrix}
\]

**Symm.:** \( A = A^T \)

**PD:** \( \forall x \in \mathbb{R} \setminus \{0\} \) \( x^T A x > 0 \)

**Symm. matrices:**

**PD:** Eigen values lead to:

- If \( a_{11} < 0 \) \( \Rightarrow \) not SPD

\[ A = L L^T \]

\[ x^T A x = x^T L L^T x \]

\[ = (L^T x)^T (L x) \]

\[ = \| L x \|_2 \]

\[ \| x \|_2 = \| L \|_2 \]

\( O(n^3) \)
More cost concerns

What’s the cost of solving $Ax = b$?

1. LU factor $A \rightarrow O(n^3)$
2. $Fw / bw$ sub $\rightarrow O(n^2)$

What’s the cost of solving $Ax = b_1, b_2, \ldots, b_n$?

2. $n \times Fw / bw$ sub $\rightarrow O(n^3)$

What’s the cost of finding $A^{-1}$?

\[
\begin{align*}
A \cdot A^{-1} &= I \\
A \cdot x &= I \quad \left(\text{solve col-by-col}\right) \rightarrow O(n^3)
\end{align*}
\]
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**  
$\mathbf{z} = \alpha \mathbf{x} + \mathbf{y}$  
vector-vector operations  
$O(n)$  
?axpy

**Level 2**  
$\mathbf{z} = \mathbf{A}\mathbf{x} + \mathbf{y}$  
matrix-vector operations  
$O(n^2)$  
?gemv

**Level 3**  
$\mathbf{C} = \mathbf{A}\mathbf{B} + \beta \mathbf{C}$  
matrix-matrix operations  
$O(n^3)$  
?gemm, ?trsm

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

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

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