Math 449: Numerical Applied Mathematics
Lecture 09

09/20/2017
Today's topic: LU Decomposition

Today: LU decomposition

Last time: Gaussian Elimination $A\vec{x} = \vec{b}$

1. Reduce system to upper triangular form
2. Solve upper triangular - system by back substitution.

$$\begin{pmatrix} u_{11} & \cdots & u_{1n} \\ & & \vdots \\ & & u_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

$$x_i = \frac{1}{u_{ii}} \left(b_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) \quad \text{for } x = 2, \ldots \qquad n^2 \text{ total operations}$$

$u_{ii} \neq 0$

1st idea: Divide-and-Conquer

Suppose n is even, $AB = C$, split A, B, C into $\frac{n}{2} \times \frac{n}{2}$ blocks

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

We can multiply blockwise just like 2x2 matrices.

$$C_{11} = A_{11} B_{11} + B_{12} \cdot B_{21} \qquad C_{21} = A_{21} B_{11} + A_{22} \cdot B_{21}$$

$$C_{12} = A_{11} B_{12} + A_{12} \cdot B_{22} \qquad C_{22} = A_{21} B_{12} + A_{22} \cdot B_{22}$$

Reduce $n \times n$ matrix multiplication to

8 $\frac{n}{2} \times \frac{n}{2}$ matrix mults

4 $\frac{n}{2} \times \frac{n}{2}$ adds

$4\left(\frac{n}{2}\right)^2 = n^2$ ordering adds

$T(n)$: # of ops to multiply 2 $n \times n$ matrices.

For simplicity, assume n is a power of 2.

Recurrence relation $\begin{cases} T(n) = 8T\left(\frac{n}{2}\right) + n^2 \\ T(1) = 1 \end{cases}$

Special case of the master theorem

Lemma $\begin{cases} T(n) = aT\left(\frac{n}{b}\right) + cn^d \\ T(1) = 1 \end{cases}$

if $\log_b a > d$, then $T(n) = O(n^{\log_b a})$

Proof Sketch

$$T(n) = aT\left(\frac{n}{b}\right) + cn^d$$
$$= a\left(aT\left(\frac{n}{b^2}\right) + c\left(\frac{n}{b}\right)^d\right] + cn^d$$
$$= a^2T\left(\frac{n}{b^2}\right) + cn^d\left(1 + \frac{a}{b^d}\right)$$
$$= a^3T\left(\frac{n}{b^3}\right) + cn^d\left(1 + \frac{a}{b^d} + \left(\frac{a}{b^d}\right)^2\right)$$
$$= a^{\log_b n}\, T(1) + cn^d\left(1 + \frac{a}{\cancel{b^d}} + \cdots + \left(\frac{a}{b^d}\right)^{\log_b n - 1}\right)$$

1^{st} term

$$a^{\log_b n} = n^{\log_b a}$$

2^{nd} term

$$O\left(n^d\left(\frac{a}{b^d}\right)^{\log_b n}\right)$$

$$n^d\left(\frac{a}{b^d}\right)^{\log_b n} = n^d\,\frac{a^{\log_b n}}{n^d} = n^{\log_b a}. \quad \boxtimes$$

Divide + Conquer Algorithm

$$\log_b a = \log_2 8 = 3 > 2 = d \qquad O(n^3)$$

Gauss Algorithm for multiplying complex number

$$(a+ib)(c+id) = (ac + bd) + i(ad+bc)$$
$$= \left((a+b)c - b(c+d)\right) + i\left(a(d-c) + (a+b)c\right)$$

Note: very useful for $(A+iB)(C+iD)$

Strassen's Algo:

$M_1 = (A_{11} + A_{22})(B_{11} + B_{22})$
$M_2 = (A_{21} + A_{22})B_{11}$
$M_3 = A_{11}(B_{12} - B_{22})$
$M_4 = A_{22}(B_{21} - B_{11})$
$M_5 = (A_{11} + A_{12})B_{22}$
$M_6 = (A_{21} - A_{11})(B_{11} + B_{12})$
$M_7 = (A_{12} - A_{22})(B_{21} + B_{22})$

$C_{11} = M_1 + M_4 - M_5 + M_7$
$C_{12} = M_3 + M_5$
$C_{21} = M_2 + M_4$
$C_{22} = M_1 - M_2 + M_3 + M_6$

7 matrix mult.
18 adds/subs

$$T(n) = 7T\left(\frac{n}{2}\right) + 18\left(\frac{n}{2}\right)^2 \quad \frac{9}{2}n^2$$
$$T(1) = 1$$

$$\log_2 7 \approx 2.8074 > 2 \Rightarrow \text{Strassen is } O(n^{\log_2 7}) \approx O(n^{2.807})$$

Current Record $\approx O(n^{2.3728639})$ F. LeGall 2014.

Conj. $O(n^{2+\varepsilon})$ for any $\varepsilon > 0$

Fast Matrix Multiplication

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \qquad 2\times2+2\times2=8$$

Strassen 1969

Cost of Gaussian Elimination

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \longrightarrow \begin{pmatrix} a_{11} & a_{12} \\ 0 & a_{22}-\frac{a_{12}\cdot a_{21}}{a_{11}} \end{pmatrix}$$

unit - lower triangular matrix.

$$L=\begin{pmatrix} 1 & 0 \\ \frac{a_{21}}{a_{11}} & 1 \end{pmatrix} \qquad L^{-1}=\begin{pmatrix} 1 & 0 \\ -\frac{a_{21}}{a_{11}} & 1 \end{pmatrix}$$

$$U=L^{-1}A$$

$$A=LU$$

more generally, $n\times n$ matrix

$U = L_{(N)}\cdots L_{(1)}A$ where each $L_{(i)}$ climates 1 entry below diagonal

$$L_{(i)}=\begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix}$$

$$A=(L_N\cdots L_1)^{-1}\cdot U = L\cdot U$$

Note: LU decomposition only works if we never encounter a 0 on the diagonal.

Example. $\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$

$$\begin{pmatrix} 1 & 0 \\ l_{21} & 1 \end{pmatrix}\begin{pmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} \\ l_{21}u_{11} & l_{21}u_{12}+u_{22} \end{pmatrix}$$

Let $u_{11}=0$ $u_{12}=1$ but then $l_{21}\cdot u_{11}=0\neq 1$

Once we have $A=LU$, it's easy to solve $A\vec{x}=\vec{b}$

$\underbrace{LU\vec{x}}_{\vec{y}}=b$ Solve $L\vec{y}=\vec{b}$ by substitution $O(n^2)$

$U\vec{x}=\vec{y}$ by back-substitution

Reminder: not every matrix has an LU decomp.