Math 128A, Summer 2019
Lecture 25, 8/5/2019

CLASS ANNOUNCEMENTS: Linear Algebra (7 lectures):
- Direct methods for solving linear systems, $Ax = b$.
- Special Types of Matrices: (1) Diagonally Dominant, (2) Positive Definite
- Linear Algebra Under Floating Point Arithmetic (proving that it can work)
- Overcoming Limitations (Iterative Improvement) of Solution to Linear Systems

1 Review

We define a matrix $A_{n\times n}$ as a box of numbers, because we commit the sin of getting our hands dirty, away from pure mathematics. The box of numbers is with respect to some pair of bases, so sometimes we treat a matrix simply as a box of numbers without reference to a particular basis and linear operator.

The space of these matrices is usually called $\mathbb{R}^{n\times n} = \mathbb{R}^{n^2} = L(\mathbb{R}^n, \mathbb{R}^n)$ (linear operator).

We define a vector as a column vector (matrix). We write **Matrix-vector multiplication** as Ax (left-multiplication). We take the standard basis:

$$e_1 = \begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix}, \quad e_2 = \begin{bmatrix}0\\1\\.\\0\\0\end{bmatrix}, \quad \cdots, \quad e_n = \begin{bmatrix}0\\0\\0\\\vdots\\1\end{bmatrix}$$

We write vector multiplication as a dot product, taking the transpose:

$$e_i^T e_j = \begin{bmatrix}\cdot & \cdot & \cdots & \cdot\end{bmatrix} \begin{bmatrix}\cdot\\\cdot\\\vdots\\\cdot\end{bmatrix} = \delta_{ij} = \begin{cases}1, & i = j\\0, & i \neq j\end{cases}$$

, where δ_{ij} is the Kronecker delta.
Alternatively, we can take the other transpose:

$$e_i e_j^T$$

which gives a matrix full of 0's except for 1 at the i, jth entry.
We can write the identity matrix in terms of this basis (for $n \times n$ matrices):

$$I = \begin{bmatrix}1 & 0 & \cdots & 0\\0 & 1 & \cdots & 0\\\vdots & \vdots & \ddots & \vdots\\0 & 0 & \cdots & 1\end{bmatrix} = e_1e_1^T + e_2e_2^T + \cdots + e_ne_n^T = \sum_{i=1}^{n} e_ie_i^T = \sum_{j=1}^{n} e_je_j^T$$

So we can write any A as

$$A = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \left(e_i e_j^T \right)$$

$$A = AI = A \sum_{j=1}^{n} e_j e_j^T = \sum_{j=1}^{n} (Ae_j) \underbrace{e_j^T}_{},$$

putting this in column j of A, and putting the following in row i of A

$$IA = \sum_{i=1}^{n} \overbrace{e_i} \left(e_i^T A \right)$$

We can also write, for **extracting matrix entries** if given some black box:

$$IAI = \sum_i \sum_j e_i \left(e_i^T A e_j \right) e_i^T = \sum \left(e_i^T A e_j \right) e_i e_j^T$$

and our convention is to write the scalar inside the parenthesis out to the front of the expression (and lose our structure and intuition).

Now we define matrix-vector multiplication. First we write:

$$X = Ix = \sum_{i=1}^{n} e_i \left(e_i^T x \right) = \sum_{i=1}^{n} x_i e_i, \qquad e_i := e_i^T x$$

and for matrix-vector multiplication,

$$\begin{aligned} (Ax)_i &= \sum_{j=1}^{n} a_{ij} x_j, \qquad x = \sum_i x_i e_i \\ &= \sum_{j=1}^{n} e_i^T A e_j e_j^T x, \end{aligned}$$

and equivalently,

$$\begin{aligned} Ax = AIx &= A \sum_i \left(e_i^T x \right) e_i \\ &= \sum_j \underbrace{\left(e_j^T x \right)}_{x_j} \underbrace{Ae_j}_{j \text{ col of } A} \end{aligned}$$

and we define $R(A)$ as the 'range' of A (or 'column space') to bring us to our first interpretation of matrix-vector multiplication. Strain's remark is that in linear algebra, we usually have two or four interpretations of each thing. Equivalently, we have our second interpretation:

$$Ax = IAx = \sum_{i=1}^{n} \overbrace{e_i}^{\text{put in row } i} \underbrace{e_i^T A}_{\text{row vec}} \underbrace{x}_{\text{col vec}}$$

$$I = \sum_{i=1}^{n} \underbrace{e_i}_{\text{col vec}} \underbrace{e_i^T}_{\text{row vec}}$$

Now we define Matrix-Matrix multiplication:

$$MA = (IM)(AI) = \sum e_i \underbrace{(e_i^T M)}_{\text{row}} \sum \underbrace{(Ae_j)}_{\text{col}} e_j^T$$

equivalently, we can place the identities differently:

$$MA = M \sum (Ae_j) e_j^T = \sum \underbrace{(MAe_j)}\underbrace{e_j^T}$$

and we say M hits each **column** of A, and e_j^T puts this into column j of the result. This means M does **row** operations on A.
If we right-multiply A by B now, we can expect that B does column operations on A. To see this, we write:

$$AB = \left(B^T A^T\right)^T = \sum_i e_i \left(e_i^T A\right) B.$$

We usually use left-multiplication, but there is always the symmetric viewpoint.
We can also write:

$$AB = \sum_j Ae_j e_j^T \sum_i e_i e_i^T B = \sum_{ij} Ae_j \underbrace{e_j^T e_i}_{\delta_{ij}} e_i^T B = \sum_i \underbrace{(Ae_i)}_{\text{col } i \text{ of } A} \underbrace{\left(e_i^T B\right)}_{\text{row } i \text{ of } B},$$

and we express AB as a sum of rank-1 $n \times n$ matrices.

2 Solving $Ax = b$

We already know how to do this via Row operations (row echelon forms). Say

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 15 \\ 15 \end{bmatrix}$$

We may write something like:

$$\begin{bmatrix} \underline{1} & 2 & 3 \\ 0 & -3 & -6 \\ 0 & -6 & -21 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ -9 \\ -27 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & \underline{-2} & -6 \\ 0 & 0 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ -9 \\ 9 \end{bmatrix}$$

and we have a nice (easy) triangular system. We have: $\boxed{x_3 = 9}$, so $-3x_2 - 6x_3 = -9$ and thus $\boxed{x_2 = 1}$. Lastly, $x_1 + 2x_2 + 3x_3 = 6$, so we have $\boxed{x_3 = 1}$. The underlined main diagonal entries (pivots) must be nonzero for our system to have a (unique) solution.

Remark: Our algorithm (if possible) has to ensure nonzero pivots.

For an example of when this is not possible, if we have something like:

$$\begin{bmatrix} 0 & 1 & 4 \\ 0 & 2 & 5 \\ 0 & 3 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix},$$

then we cannot readily predict if we have a solution or if we have (infinitely) many solutions. It follows via linear algebra that here the kernel (null space) is 1-dimensional, so the solution space (range) is $(3-1)$-dimensional.
Strain: This seems like a good opportunity for a theorem.

Theorem 2.1. $\exists!$ solution x to $Ax = b$ if and only if:
(1) A is an intertible matrix, **independent** of choices of b, or
(2) $\det A \neq 0$, or
(3) no 0 pivots are encountered (with the correct sorting in Gaussian elimination, aka partial pivoting), or
(4) none of As eigenvalues is 0

(Aside) Strain: Applied math majors don't drink beer, they drink whiskey.

3 GEPP: Gaussian Elimination with Partial Pivoting

Now we say the real meaning behind **GEPP** (Gaussian Elimination with Partial Pivoting) is that if we take an equation $Ax = b$ and convert it into an upper-triangular form $Ux = \beta$, where U is upper-triangular with **nonzero entries on diagonal**. Recall that the entire first hour of lecture was spent on row operations. We say that there should be some operator M to enact row operations on A via MA to give $MA = U$. (We postpone discussion on pivoting until later, possibly tomorrow.) Our question is how do we find M that performs a specified row operation?
Strain (Aside): Suppose we are out drinking with a friend, and there's a bullet flying toward me, and I want to know what happens. I take my friend, place him in front of me and observe. We do this by inserting I:

$$MA = (MI)A$$

For example, to get from :

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix} \xmapsto{M_1} \begin{bmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & -27 & -42 \end{bmatrix},$$

we take

$$M_1 := \begin{bmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ -7 & 0 & 1 \end{bmatrix}$$

which we call **unit-lower-triangular**. Now we say

$$M_2 M_1 A = U$$

where U is upper-triangular and

$$M_2 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 9 & 1 \end{bmatrix}.$$

$$\begin{aligned} M_2 M_1 A &= U \\ A &= LU \\ L &:= (M_1^{-1} M_2^{-1}) \end{aligned}$$

In summary, we find that $A = LU$ is actually simply just a matrix-factorization, with $L := \left(M_1^{-1} M_2^{-1}\right)$, which we call **lower-triangular reduction to upper-triangular form**.
In our previous example, we write:

$$M_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -9 & 1 \end{bmatrix},$$

$$M_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 0 & 1 \end{bmatrix}$$

Try:

$$M_1^{-1}M_2^{-1} = \begin{bmatrix}1 & 0 & 0\\ 4 & 1 & 0\\ 7 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -9 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 4 & 1 & 0\\ 7 & -9 & 1\end{bmatrix},$$

and notice that this is **not** how matrix-matrix multiplication usually works. Here, we simply take items and place them into the final matrix. It turns out that L, which is apparently a relatively complicated object because we need to invert matrices, is actually a matrix of multipliers:

$$L = \begin{bmatrix}1 & 0 & \cdots & 0\\ m_{21} & \ddots & & 0\\ \vdots & & \ddots & 0\\ m_{n1} & \cdots & m_{n,m-1} & 0\end{bmatrix}$$

where m_{ij} is simply the row operations we perform for Gaussian elimination. For example, write:

$$A \mapsto M_1 A = \begin{bmatrix}a_1 1 & \cdots & a_{1n}\\ 0 & \ddots & \\ 0 & \cdots & \cdots\end{bmatrix}, \begin{bmatrix}a_{11} & c^T\\ d & A_{22}\end{bmatrix} \xrightarrow{e_i \frac{d_i}{a_{11}} c^T} \begin{bmatrix}a_{11} & c^T\\ \mathcal{M}(0) & A_{22} - \frac{d_i}{a_{11}}c^T\end{bmatrix},$$

where

$$M_1 := \begin{bmatrix}1 & \cdots & \cdots\\ \frac{-a_{21}}{a_{11}} & \ddots & \cdots\\ \vdots & \ddots & \cdots\\ \frac{-a_{n1}}{a_{11}} & \cdots & \cdots\end{bmatrix} = I - \frac{1}{a_{11}}\overbrace{(a_{:,1} - a_{11}e_1)}^{m_1} e_1^T,$$

where $e_1^T m_1 = 0$, where m_1 is the first column of A but setting the first entry (pivot) to 0. Notice that m_2 is the second column of **the result** after performing M_1 (to yield a different A).
Thus our theorem is essentially proven by a single step of recursion, where if A is invertible, then the smaller bottom-right parts are also invertible

Lecture ends here.

Next time we'll talk about **pivoting**. Strain jokes that usually once we learn pivoting, we end up trying to forget it quickly.