TEXAS
The University of Texas at Austin

TOPIC 1 LINEAR ALGEBRA

Vectors

- A vector is just a list of numbers (order matters!!!)
  - v = (4, 12, π, 4)
  - w = (-3, 1.5, 3, 1.5)
- 2 vectors are equal *if and only if* all entries are equal
- What is the dimension of v? w?
  - rows x columns
- What is v_2? w_4?
  - Python is 0 based indexing, but math is 1 based indexing…
- What is v^T?
- What is w+v?
- What is w*v?

- v = (4, 12, π, 4), w = (-3, 1.5, 3, 1.5)

$$v^T = \begin{pmatrix} 4 \\ 12 \\ \pi \\ 4 \end{pmatrix}$$

$$v+w = (4+-3,\ 12+1.5,\ \pi+3,\ 4+1.5) = (1,\ 13.5,\ 3+\pi,\ 5.5)$$

$$v * w - \text{DNE}$$

Matrices

- A matrix is simply a 2-D list of numbers
  - 2x2 $A = \begin{pmatrix} 1 & 2 \\ 3 & 3 \end{pmatrix}$ $B = \begin{pmatrix} 5 & 2 \\ -1 & 1 \end{pmatrix}$ $C = \begin{pmatrix} 3 & 2 & 1 \\ 3 & 4 & 2 \end{pmatrix}$
- $A_{1,2}$? 2 $A_{2,1}$? 3 (row, column)
- A+B? $= \begin{pmatrix} 1+5, & 2+2 \\ 3+-1, & 3+1 \end{pmatrix} = \begin{pmatrix} 6 & 4 \\ 2 & 4 \end{pmatrix}$
- A+C? — DNE
- A*B?
- A*C?

Matrix Multiplication

- We want to multiply 2 matrices, A and B
  - Dimension of A is n_A x m_A
  - Dimension of B is n_B x m_B
- $(A * B)_{ij} = \sum_k A_{ik} B_{kj}$
  - We can only multiply A*B if $m_A = n_B$
  - A*B has dimension n_A x m_B
- $(B * A)_{ij} = \sum_k B_{ik} A_{kj}$
  - We can only multiply B*A if $m_B = n_A$
  - B*A has dimension n_B x m_A
- This implies that $A * B \neq B * A$!!!!!!!!!!!!!!!!!

$A * B$

$n_A \times m_A \; * \; n_B \times m_B$

- $A = \begin{pmatrix} 1 & 3 \\ 3 & 6 \end{pmatrix}$ $B = \begin{pmatrix} 5 & 2 \\ -1 & 1 \end{pmatrix}$ $C = \begin{pmatrix} 3 & 2 & 1 \\ 3 & 4 & 2 \end{pmatrix}$

$$A*B = \begin{pmatrix} 1*5+3*-1, & 1*2+3*1 \\ 3*5+6*-1, & 3*2+6*1 \end{pmatrix} = \begin{pmatrix} 2 & 5 \\ 9 & 12 \end{pmatrix}$$

$$A*C = \begin{pmatrix} 1*3+3*3, & 1*2+3*4, & 1*1+3*2 \\ 3*3+6*3, & 3*2+6*4, & 3*1+6*2 \end{pmatrix}$$

$$\begin{pmatrix} 12 & 14 & 7 \\ 27 & 30 & 15 \end{pmatrix}$$

Class Participation

- Let
  - $A = \begin{bmatrix} 1 & 1 \\ 3 & -2 \end{bmatrix}$ -2x2
  - $B = \begin{bmatrix} 1 & 0 \\ 2 & -1 \\ 3 & 7 \end{bmatrix}$ -3x2
- Which can you solve A*B or B*A?
- Solve whichever one you can

$$B*A = \begin{pmatrix} 1*1+0*3, & 1*1+0*-2 \\ 2*1+-1*3, & 2*1+-1*-2 \\ 3*1+7*3, & 3*1+7*-2 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 \\ -1 & 4 \\ 24 & -11 \end{pmatrix}$$

Matrix Identities

- Additive identity
  - $A + \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} = A$
  - $A + (-A) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$
- Multiplicative identity
  - $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} * A = A$
  - $A \; * A^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

$\begin{pmatrix} 1 & 2 \\ 3 & -4 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & -4 \end{pmatrix}$

$\begin{pmatrix} 1 & 2 \\ 3 & -4 \end{pmatrix} + \begin{pmatrix} -1 & -2 \\ -3 & 4 \end{pmatrix} =$

$I * A$

$\boxed{6} \times 6 * 6 \times 8$

Multiplicative Identity

- Let's double check the multiplicative identity

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} = \begin{pmatrix} \boxed{1} & 2 \\ 3 & \boxed{4} \\ \boxed{5} & 6 \end{pmatrix}$$

$3 \times 3 \quad 3 \times 2$

$A * I$

$3 \times 2 \quad 2 \times 2$

Systems of Equations

- Matrices and vectors are often used to express systems of linear equations
  - $3x + 2y = 8$
  - $x - y = 1$
- $\begin{pmatrix} 3 & 2 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 8 \\ 1 \end{pmatrix}$
- $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 1 & -1 \end{pmatrix}^{-1} \begin{pmatrix} 8 \\ 1 \end{pmatrix}$

$$\underbrace{\begin{pmatrix} 3 & 2 \\ 1 & -1 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} x \\ y \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} 8 \\ 1 \end{pmatrix}}_{b}$$

$$\begin{pmatrix} 3*x + 2*y \\ 1*x + (-1)*y \end{pmatrix} = \begin{pmatrix} 8 \\ 1 \end{pmatrix}$$

$$A^{-1} A x = A^{-1} b$$

$$x = A^{-1} b$$

Linear Algebra

- Let’s do this in python!

A Linear Algebra Example - Participation

- A total of \$50,000 is invested in three funds paying 6%, 8%, and 10% interest for 1 year. The year's total interest is \$3,700. Twice as much money is invested at 6% as invested at 10%. How much was invested in each of the funds.
- Formulate as A x =b
- Solve using $x=A^{-1}b$
  - In python you use np.linalg.solve(A,b)

$x = \$ @ 6\%$ $\quad y = \$ @ 8\%$ $\quad z = \$ @ 10\%$

$$x + y + z = 50$$

$$0.06x + 0.08y + 0.1z = 3.7$$

$$x = 2z \rightarrow x - 2z = 0$$

$$-x + 2z = 0$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 0.06 & 0.08 & 0.1 \\ -1 & 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 50 \\ 3.7 \\ 0 \end{pmatrix}$$

A Difficult Problem

- The PageRank algorithm is at the heart of what made Google the giant it is today and is used to determine the order in which search results are displayed. We're going to explore the PageRank algorithm.

- The goal of the PageRank algorithm is to sort webpages based on asking a simple question: if I were to start on a random webpage, and then randomly click a link on that page, and then randomly click a link on the next page, and so on, forever, with what frequency would I end up on each page? The pages that are clicked to frequently are considered 'important' and placed higher in the search results. This means that pages that are linked to by many other pages are considered important, but so are pages that are linked to by just a few important pages. It turns out the PageRank algorithm is just a huge linear algebra problem that Google has figured out how to solve very efficiently.

Page Rank

- The csv file on canvas has information on 100 webpages: which pages link to which other pages. The goal of this problem is to rank these 100 pages from most to least important.
- The csv file is a 100x100 matrix filled with True/False variables, let's call this matrix L. If L[i, j] == True, that means that page i has a link to page j. You can get from page i to page j with 1 click.
- Links are not necessarily reciprocal, so that does not necessarily mean that page j links to page i.

Page Rank

- First, we’ll create a matrix, P, such that P[i, j] is equal to the probability that if you start on page i you will click a link and end up on page j next.

Page Rank

- To start the PageRank algorithm, we must pick a webpage to start on and begin clicking links.
- If I had a row vector, v, with 100 entries (v is 1x100) such that all entries of v were greater than or equal to zero, and the sum of all entries of v were equal to 1, we could interpret v as the probability of starting on each page.
- Let's start with a simple v, such that all pages are equally likely, v[i] = 0.01 for all i. You can interpret this as tossing a 100-sided coin to pick which page to start clicking on.

Page Ranks

- If we were to multiply v*P, the resulting output would be the probability of being on each page after randomly starting on an initial page and randomly clicking one link on that initial page. Why?

$$v^1(17) = v^0(0)P_{0\,17} + v^0(1)P_{1\,17} + v^0(2)P_{2\,17} + \dots + v^0(99)P_{99\,17}$$

$$\left[v^0(0), v^0(1), v^0(2), \dots, v^0(99)\right]\begin{pmatrix} P_{0\,17} \\ P_{1\,17} \\ P_{2\,17} \\ \vdots \\ P_{99\,17} \end{pmatrix}$$

$$v^1 = v^0 P$$

Page Ranks

- The goal of the PageRank algorithm is to find a v such that v*P = v.

assume set v^0 st $v^0 * P = v^0$

what is prob of roll then 1 click get to page 17 → v^0_{17}

$v^1 = v^0 * P = v^0$

What about after 2 clicks → v^0_{47}

$v^2 = v^1 * P = v^0 * P = v^0$

Page Rank

- It is possible to write this as a system of linear equations, but it turns out this system of linear equations is not straightforward to solve, and it is actually a special type of problem called an Eigenvalue Problem.
- One way to approximate the solution to this problem is to start with an initial v, say v^1, and repeatedly multiply by P, so that
- $v^2 = v^1 * P$
- $v^3 = v^2 * P$
- …
- $v^{n+1} = v^n * P$
- The hope is that after enough multiplications v^{n+1} will be very close to v^n and we will have approximately solved the problem.

Order of Complexity

- Assume we have an n x n matrix, A, and n x 1 vector x
- Multiply: A * x => n^2 operations
- Solve: A * x = b => n^3
- Find: A^{-1} => n^4
- Someone mentioned after class that finding v is equivalent to solving a linear system…they were right
- But multiplying a few times is quicker than solving!

Big System of Equations

A Practice Problem

- We can change the equation v*P = v to be v*(P-I) = 0
- We can further change it to look more like the traditional form by taking the transpose
  - $(P^T - I) * v^T = 0$
- Assume you have found a v that satisfies this, will 17.7*v also satisfy this?

A Practice Problem

- This set of equations has 1 redundant equation
  - Let’s try to solve it anyway!
- Assume for a second that $v_{99} = 1$
  - But wait, then v can’t sum to 1!
  - Let’s solve for $v_{0:99}$ and then come back and fix v to sum to 1
- If we assume $v_{99} = 1$, we can remove the last equation from the system
- This is 99 equations and 99 unknowns
- Pose this problem and solve it for extra practice