Review of linear algebra

Saturday, November 21, 2020 9:28 AM

Some review of what you learned in linear algebra, some new stuff

Parts from ch 6.1 & ch 6.3 but mostly misc. material

Matrices and vectors

$A = \begin{bmatrix} 3 & 4 & 6 \\ 2 & 1 & 8 \end{bmatrix}$ is a 2×3 matrix, and sometimes write $A = [a_{ij}]$ to denote $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$

usually capital letters for matrices

Indices a_{ij} a nearly universal convention is that i is the row index, j is the column index

Vectors: a "vector" in a vector space V is an abstract object that you should have learned about in your lin. alg. class.

For us now, "vector" is a specific representation of a vector in a finite dimensional vector space, given a specific basis

i.e., just a list of numbers $x = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \in \mathbb{R}^3$

Sometimes, not always, write as $\vec{x}$ or $\overset{\rightharpoonup}{x}$ or $\underline{x}$ or (when typeset) in bold. Usually lowercase.

For computation, "vector" = "column vector" though sometimes we write it as a row vector just because it's easier to typeset

(a rigorous way to do this is $x^T = [3, 4, 5]$)

transpose

$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \quad A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$

Numpy distinguishes a vector (1D) from a row or column vector ($1 \times n$ or $n \times 1$)

If $A = A^T$, we call the matrix symmetric

A^* means $\bar{A}^T$ (aka A^+ in physics notation, though in numerical analysis, "dagger" is used for pseudo-inverse)

"adjoint"   complex conjugate   "dagger"

"Matlab notation"

"end" keyword in Matlab is useful

$A(2,:)$ means the 2^{nd} row of A (":" means "everything")

$A(:,4)$ means the 4^{th} column of A

Numpy has similar, but not identical, syntax

How is a matrix stored on a computer?

Let's start w/ a <u>vector</u>. In python, an array $x = [2, 3, 9]$ might be stored as a linked list. Good for <u>appending</u>, horrible for numerical computation.

Numpy and matlab store as a contiguous array, much like in C

$x = [2, 3, 9]$ ⟶ stores metadata, like length

index a in memory

index $a + \Delta$

$a + 2\Delta$

where Δ is size of the datatype (eg. 8 bytes)

Now a <u>matrix</u>

$$A = \begin{bmatrix} 2 & 3 & 9 \\ 4 & 6 & -2 \end{bmatrix}$$

Store metadata (# rows, columns)

then 2 main classes: <u>row-major order</u>: $\mathrm{vec}(A) = [2\ 3\ 9\ \ 4\ 6\ {-2}]$ and store this vector

Numpy, C

or <u>column-major-order</u>: $\mathrm{vec}(A) = \begin{bmatrix} 2 \\ 4 \\ 3 \\ 6 \\ 9 \\ -2 \end{bmatrix}$ and store this vector

Fortran, Matlab

Sparse matrix Storage

A "sparse" matrix is one that has many 0 entries, like

$$A = \begin{bmatrix} 0 & 0 & 4 \\ 0 & 1 & 0 \end{bmatrix}.$$

Mathematically, "many 0's" is vague.

Computationally, it is not <u>vague</u>: you must tell your system (Numpy...) to use a sparse format (eg. Matlab: $A = \mathrm{sparse}(A)$)

Otherwise it stores the 0's.

won't go into detail. It's not always efficient: need a <u>lot</u> of 0 entries to be worth it.

(google compressed sparse row (CSR)

or compressed sparse column (CSC))

Matlab uses

Special matrices / types-of-matrices

- Diagonal
$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ 0 & 0 & \ddots & 0 \\ 0 & 0 \cdots & 0 & a_{nn} \end{bmatrix}$$
often use notation D

- Lower triangular
$$A = \begin{bmatrix} a_{11} & 0 & & 0 \\ a_{21} & a_{22} & \ddots & \\ \vdots & & & 0 \\ a_{n1} & \cdots & & a_{nn} \end{bmatrix}$$
often use L

Upper triangular is the opposite
$$A = \begin{bmatrix} & & \\ 0 & & \\ & & \end{bmatrix}$$
often use U

- Identity
$$I = \begin{bmatrix} 1 & & 0 \\ & 1 & \\ 0 & & \ddots & \\ & & & 1 \end{bmatrix}$$
or $I_{3\times3}$ to make size clear if needed
$$I_{3\times3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- All ones vector $\mathbb{1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$

- All ones matrix $(\neq I)$ is $\mathbb{1}\mathbb{1}^T = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & & & & \\ \vdots & & & & \\ 1 & & \cdots & & 1 \end{bmatrix}$

- Canonical basis vectors aka unit vectors
$$e_j = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow j^{th} \text{ spot}$$
<u>ex</u>: in $\mathbb{R}^3$, $e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, $e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$, $e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

- Diagonally dominant if $\forall i$, $|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|$

if $>$ then "strictly" diag. dom.

e.g. $A = \begin{pmatrix} 13 & 2 & 1 \\ 5 & 19 & -2 \\ 3 & 6 & -20 \end{pmatrix}$

check: $13 > 2+1$ ✓
$19 > 5 + |-2|$ ✓
$|-20| > 3+6$ ✓

so A is strictly diag. dominant

- 0 matrix $0 = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & & \\ 0 & \cdots & 0 \end{bmatrix}$ usually the size is determined by the context

- Block matrices
$$A = \left[\begin{array}{cc|cc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ \hline 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{array}\right]$$
can be written as $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$

we do this a lot w/ 0 blocks $\boxed{A_{11}} = \begin{bmatrix} 1 & 2 \\ 5 & 6 \end{bmatrix}$, $A_{12} = \dots$

- $A = A^T$ is symmetric
  $A = A^* \,(:= \bar{A}^T)$ is self-adjoint = Hermitian
  } same thing if all $\mathbb{R}$ entries and not $\mathbb{C}$

Matrix Algebra

The dot product aka inner product of two vectors $\vec{x}$ and $\vec{y}$ in $\mathbb{R}^n$

$$\vec{x}^T\vec{y} = \sum_{i=1}^{n} x_i y_i \qquad \vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad \vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Sometimes written $\langle x, y \rangle$ or $\langle x | y \rangle$ in physics

and if $\vec{x}, \vec{y} \in \mathbb{C}^n$, it is $\vec{x}^* \vec{y} = \sum_{i=1}^{n} \bar{x}_i y_i$

$$(x_1 \;\; x_2 \; \dots \; x_n) \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

[Note: the outer product is $\vec{x} \cdot \vec{y}^T$ — physics $|x\rangle\langle y|$

i.e. $\begin{bmatrix} 3 \\ 1 \end{bmatrix}\begin{bmatrix} 2 & 4 \end{bmatrix} = \begin{bmatrix} 6 & 12 \\ 2 & 4 \end{bmatrix}$]

If a matrix A is $m \times n$ w/ rows $A = \begin{bmatrix} - \vec{a}_1^T - \\ \vdots \\ - \vec{a}_m^T - \end{bmatrix}$

then for $\vec{y} \in \mathbb{R}^n$, $A\vec{y} \in \mathbb{R}^m$, $(A\vec{y})_i = \vec{a}_i^T \vec{y}$ "matrix-vector product" mat-vec

If a matrix B is $n \times k$, w/ colums $B = \begin{bmatrix} | & & | \\ \vec{b}_1 & \dots & \vec{b}_k \\ | & & | \end{bmatrix}$

then $(AB)_{ik} = \vec{a}_i^T \vec{b}_k = \underbrace{\sum_{j=1}^{n} a_{ij} b_{jk}}$

you might have learned via indices like this but I encourage you to think of via dot products and higher-level concepts

Ex $\begin{bmatrix} 2 & 3 & 4 \\ 1 & 2 & 6 \end{bmatrix} \cdot \begin{bmatrix} -1 & 0 \\ 2 & 1 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$

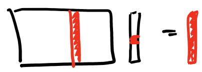
$c_{21} = \begin{bmatrix} 1 & 2 & 6 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix} = -1 + 4 + 0 = 3$

Remember, if $\underset{m \times k}{C} = \underset{m \times n}{A} \times \underset{n \times k}{B}$ (n and n must match)

Facts: • In general, matrix multiplication is not commutative

$AB \neq BA$ (in fact, the sizes might not even work!)

- $A \cdot e_j = A(:,j)$ i.e., j^{th} column



- $A \cdot I = A$ and $I \cdot A = A$

- $A = B \Rightarrow CA = CB$ (if C is the right size)

and $A = B \Rightarrow AC = BC$ (if C is the right size)

- If A is a <u>square matrix</u> (ie $n \times n$),

  we say it is invertible aka non-singular if there is a matrix (which we denote as A^{-1}) such that $A \cdot A^{-1} = I$ ← identity

  (and it turns out this implies $A^{-1}A = I$ as well)

  $(A^{-1})^{-1} = A$

  If not invertible, we call it singular

- $(AB)^T = B^T A^T$ and $(AB)^* = B^* A^*$

- $(AB)^{-1} = B^{-1}A^{-1}$ if AB is invertible

- Sherman-Morrison / Woodbury / Matrix-Inversion Lemma

  $$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1} U)^{-1} V^T A^{-1}$$

  eg., if u, v are vectors

  $$(A + \underbrace{\varepsilon u v^T}_{\text{rank 1 perturbation}}) = A^{-1} - \frac{A^{-1} u v^T A^{-1}}{1/\varepsilon + v^T A^{-1} u}$$

- Matrix multiplication <u>is</u> associative and distributive

  $$A(BC) = ABC = (AB)C$$

  $$A(B+C) = AB + AC$$

  $$(A+B)C = AC + BC$$

- The inverse of a lower (upper) triangular matrix, if it exists, is also lower (upper) triangular

- The product of two lower (upper) triangular matrices is also lower (upper) triangular

- Not going over rank, determinant, trace, etc. but hopefully you've seen in linear algebra