Lecture 1
Matrix Terminology and Notation

- matrix dimensions
- column and row vectors
- special matrices and vectors

Matrix dimensions

a *matrix* is a rectangular array of numbers between brackets

examples:

$$A = \begin{bmatrix} 0 & 1 & -2.3 & 0.1 \\ 1.3 & 4 & -0.1 & 0 \\ 4.1 & -1 & 0 & 1.7 \end{bmatrix}, \qquad B = \begin{bmatrix} 3 & -3 \\ 12 & 0 \end{bmatrix}$$

dimension (or size) always given as (numbers of) rows $\times$ columns

- A is a 3×4 matrix, B is 2×2
- the matrix A has four columns; B has two rows

$m \times n$ matrix is called *square* if $m = n$, *fat* if $m < n$, *skinny* if $m > n$

Matrix coefficients

coefficients (or entries) of a matrix are the values in the array

coefficients are referred to using double subscripts for row, column

A_{ij} is the value in the ith row, j column of A; also called i, j entry of A

i is the *row index* of A_{ij}; j is the *column index* of A_{ij}

(here, A is a matrix; A_{ij} is a number)

example: for $A = \begin{bmatrix} 0 & 1 & -2.3 & 0.1 \\ 1.3 & 4 & -0.1 & 0 \\ 4.1 & -1 & 0 & 1.7 \end{bmatrix}$, we have:

$A_{23} = -0.1$, $A_{22} = 4$, but A_{41} is meaningless

the row index of the entry with value -2.3 is 1; its column index is 3

Column and row vectors

a matrix with one column, *i.e.*, size $n \times 1$, is called a (column) *vector*

a matrix with one row, *i.e.*, size $1 \times n$, is called a *row vector*

'vector' alone usually refers to column vector

we give only one index for column & row vectors and call entries *components*

$$v = \begin{bmatrix} 1 \\ -2 \\ 3.3 \\ 0.3 \end{bmatrix} \qquad w = \begin{bmatrix} -2.1 & -3 & 0 \end{bmatrix}$$

- v is a 4-vector (or 4×1 matrix); its third component is $v_3 = 3.3$
- w is a row vector (or 1×3 matrix); its third component is $w_3 = 0$

Matrix equality

$A = B$ means:

- A and B have the same size
- the corresponding entries are equal

for example,

- $\begin{bmatrix} -2 \\ 3.3 \end{bmatrix} \neq \begin{bmatrix} -2 & -3.3 \end{bmatrix}$ since the dimensions don't agree
- $\begin{bmatrix} -2 \\ 3.3 \end{bmatrix} \neq \begin{bmatrix} -2 \\ 3.1 \end{bmatrix}$ since the 2nd components don't agree

Zero and identity matrices

$0_{m \times n}$ denotes the $m \times n$ **zero matrix**, with all entries zero

I_n denotes the $n \times n$ **identity matrix**, with

$$I_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

$$0_{2\times 3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$0_{n\times 1}$ called *zero vector*; $0_{1\times n}$ called *zero row vector*

convention: usually the subscripts are dropped, so you have to figure out the size of 0 or I from context

Unit vectors

e_i denotes the ith **unit vector**: its ith component is one, all others zero

the three unit 3-vectors are:

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \qquad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \qquad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

as usual, you have to figure the size out from context

unit vectors are the columns of the identity matrix I

some authors use $\mathbf{1}$ (or e) to denote a vector with all entries one, sometimes called the **ones vector**

the ones vector of dimension 2 is $\mathbf{1} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Lecture 2
Matrix Operations

- transpose, sum & difference, scalar multiplication
- matrix multiplication, matrix-vector product
- matrix inverse

Matrix transpose

transpose of $m \times n$ matrix A, denoted A^T or A', is $n \times m$ matrix with

$$\left(A^T\right)_{ij} = A_{ji}$$

rows and columns of A are transposed in A^T

example: $\left[\begin{array}{cc} 0 & 4 \\ 7 & 0 \\ 3 & 1 \end{array}\right]^T = \left[\begin{array}{ccc} 0 & 7 & 3 \\ 4 & 0 & 1 \end{array}\right].$

- transpose converts row vectors to column vectors, vice versa
- $\left(A^T\right)^T = A$

Matrix addition & subtraction

if A and B are both $m \times n$, we form $A + B$ by adding corresponding entries

example: $$\begin{bmatrix} 0 & 4 \\ 7 & 0 \\ 3 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 0 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 6 \\ 9 & 3 \\ 3 & 5 \end{bmatrix}$$

can add row or column vectors same way (but never to each other!)

matrix subtraction is similar: $$\begin{bmatrix} 1 & 6 \\ 9 & 3 \end{bmatrix} - I = \begin{bmatrix} 0 & 6 \\ 9 & 2 \end{bmatrix}$$

(here we had to figure out that I must be 2×2)

Properties of matrix addition

- commutative: $A + B = B + A$
- associative: $(A+B)+C = A+(B+C)$, so we can write as $A+B+C$
- $A + 0 = 0 + A = A$; $A - A = 0$
- $(A + B)^T = A^T + B^T$

Scalar multiplication

we can multiply a number (a.k.a. *scalar*) by a matrix by multiplying every entry of the matrix by the scalar

this is denoted by juxtaposition or $\cdot$, with the scalar on the left:

$$(-2)\begin{bmatrix} 1 & 6 \\ 9 & 3 \\ 6 & 0 \end{bmatrix} = \begin{bmatrix} -2 & -12 \\ -18 & -6 \\ -12 & 0 \end{bmatrix}$$

(sometimes you see scalar multiplication with the scalar on the right)

- $(\alpha + \beta)A = \alpha A + \beta A$; $(\alpha\beta)A = (\alpha)(\beta A)$
- $\alpha(A + B) = \alpha A + \alpha B$
- $0 \cdot A = 0$; $1 \cdot A = A$

Matrix multiplication

if A is $m \times p$ and B is $p \times n$ we can form $C = AB$, which is $m \times n$

$$C_{ij} = \sum_{k=1}^{p} a_{ik}b_{kj} = a_{i1}b_{1j} + \cdots + a_{ip}b_{pj}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

to form AB, $\#$cols of A must equal $\#$rows of B; called **compatible**

- to find i, j entry of the product $C = AB$, you need the ith row of A and the jth column of B
- form product of corresponding entries, $e.g.$, third component of ith row of A and third component of jth column of B
- add up all the products

Examples

example 1: $\begin{bmatrix} 1 & 6 \\ 9 & 3 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} -6 & 11 \\ -3 & -3 \end{bmatrix}$

for example, to get $1,1$ entry of product:

$$C_{11} = A_{11}B_{11} + A_{12}B_{21} = (1)(0) + (6)(-1) = -6$$

example 2: $\begin{bmatrix} 0 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 6 \\ 9 & 3 \end{bmatrix} = \begin{bmatrix} -9 & -3 \\ 17 & 0 \end{bmatrix}$

these examples illustrate that matrix multiplication is not (in general) commutative: we don't (always) have $AB = BA$

Properties of matrix multiplication

- $0A = 0$, $A0 = 0$ (here 0 can be scalar, or a compatible matrix)
- $IA = A$, $AI = A$
- $(AB)C = A(BC)$, so we can write as ABC
- $\alpha(AB) = (\alpha A)B$, where α is a scalar
- $A(B + C) = AB + AC$, $(A + B)C = AC + BC$
- $(AB)^T = B^T A^T$

Matrix-vector product

very important special case of matrix multiplication: $y = Ax$

- A is an $m \times n$ matrix
- x is an n-vector
- y is an m-vector

$$y_i = A_{i1}x_1 + \cdots + A_{in}x_n, \quad i = 1, \ldots, m$$

can think of $y = Ax$ as

- a function that transforms n-vectors into m-vectors
- a set of m linear equations relating x to y

Inner product

if v is a row n-vector and w is a column n-vector, then vw makes sense, and has size 1×1, $i.e.$, is a scalar:

$$vw = v_1 w_1 + \cdots + v_n w_n$$

if x and y are n-vectors, $x^T y$ is a scalar called *inner product* or *dot product* of x, y, and denoted $\langle x, y \rangle$ or $x \cdot y$:

$$\langle x, y \rangle = x^T y = x_1 y_1 + \cdots + x_n y_n$$

(the symbol $\cdot$ can be ambiguous — it can mean dot product, or ordinary matrix product)

Matrix powers

if matrix A is square, then product AA makes sense, and is denoted A^2

more generally, k copies of A multiplied together gives A^k:

$$A^k = \underbrace{A\,A\,\cdots\,A}_{k}$$

by convention we set $A^0 = I$

(non-integer powers like $A^{1/2}$ are tricky — that's an advanced topic)

we have $A^k A^l = A^{k+l}$

Matrix inverse

if A is square, and (square) matrix F satisfies $FA = I$, then

- F is called the *inverse* of A, and is denoted A^{-1}
- the matrix A is called *invertible* or *nonsingular*

if A doesn't have an inverse, it's called *singular* or *noninvertible*

by definition, $A^{-1}A = I$; a basic result of linear algebra is that $AA^{-1} = I$

we define negative powers of A via $A^{-k} = \left(A^{-1}\right)^k$

Examples

example 1: $\begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix}^{-1} = \frac{1}{3}\begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix}$ (you should check this!)

example 2: $\begin{bmatrix} 1 & -1 \\ -2 & 2 \end{bmatrix}$ does not have an inverse; let's see why:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} 1 & -1 \\ -2 & 2 \end{bmatrix} = \begin{bmatrix} a-2b & -a+2b \\ c-2d & -c+2d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

. . . but you can't have $a - 2b = 1$ and $-a + 2b = 0$

Properties of inverse

- $\left(A^{-1}\right)^{-1} = A$, *i.e.*, inverse of inverse is original matrix (assuming A is invertible)
- $(AB)^{-1} = B^{-1}A^{-1}$ (assuming A, B are invertible)
- $\left(A^T\right)^{-1} = \left(A^{-1}\right)^T$ (assuming A is invertible)
- $I^{-1} = I$
- $(\alpha A)^{-1} = (1/\alpha)A^{-1}$ (assuming A invertible, $\alpha \neq 0$)
- if $y = Ax$, where $x \in \mathbf{R}^n$ and A is invertible, then $x = A^{-1}y$:

$$A^{-1}y = A^{-1}Ax = Ix = x$$

Inverse of 2×2 matrix

it's useful to know the general formula for the inverse of a 2×2 matrix:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

provided $ad - bc \neq 0$ (if $ad - bc = 0$, the matrix is singular)

there are similar, but much more complicated, formulas for the inverse of larger square matrices, but the formulas are rarely used

Lecture 3
Linear Equations and Matrices

- linear functions
- linear equations
- solving linear equations

Linear functions

function f maps n-vectors into m-vectors is *linear* if it satisfies:

- *scaling*: for any n-vector x, any scalar α, $f(\alpha x) = \alpha f(x)$
- *superposition*: for any n-vectors u and v, $f(u + v) = f(u) + f(v)$

example: $f(x) = y$, where $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$, $y = \begin{bmatrix} x_3 - 2x_1 \\ 3x_1 - 2x_2 \end{bmatrix}$

let's check scaling property:

$$f(\alpha x) = \begin{bmatrix} (\alpha x_3) - 2(\alpha x_1) \\ 3(\alpha x_1) - 2(\alpha x_2) \end{bmatrix} = \alpha \begin{bmatrix} x_3 - 2x_1 \\ 3x_1 - 2x_2 \end{bmatrix} = \alpha f(x)$$

Matrix multiplication and linear functions

general example: $f(x) = Ax$, where A is $m \times n$ matrix

- scaling: $f(\alpha x) = A(\alpha x) = \alpha Ax = \alpha f(x)$
- superposition: $f(u+v) = A(u+v) = Au + Av = f(u) + f(v)$

so, matrix multiplication is a linear function

converse: *every* linear function $y = f(x)$, with y an m-vector and x and n-vector, can be expressed as $y = Ax$ for some $m \times n$ matrix A

you can get the coefficients of A from $A_{ij} = y_i$ when $x = e_j$

Composition of linear functions

suppose

- m-vector y is a linear function of n-vector x, $i.e.$, $y = Ax$ where A is $m \times n$
- p-vector z is a linear function of y, $i.e.$, $z = By$ where B is $p \times m$.

then z is a linear function of x, and $z = By = (BA)x$

so *matrix multiplication* corresponds to *composition* of linear functions, $i.e.$, linear functions of linear functions of some variables

Linear equations

an equation in the variables $x_1, \ldots, x_n$ is called *linear* if each side consists of a sum of multiples of x_i, and a constant, $e.g.$,

$$1 + x_2 = x_3 - 2x_1$$

is a linear equation in $x_1,\ x_2,\ x_3$

any set of m linear equations in the variables $x_1, \ldots, x_n$ can be represented by the compact matrix equation

$$Ax = b,$$

where A is an $m \times n$ matrix and b is an m-vector

Example

two equations in three variables $x_1,\ x_2,\ _3$:

$$1 + x_2 = x_3 - 2x_1, \quad x_3 = x_2 - 2$$

step 1: rewrite equations with variables on the lefthand side, lined up in columns, and constants on the righthand side:

$$\begin{array}{cccccc} 2x_1 & +x_2 & -x_3 & = & -1 \\ 0x_1 & -x_2 & +x_3 & = & -2 \end{array}$$

(each row is one equation)

step 2: rewrite equations as a single matrix equation:

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

- ith row of A gives the coefficients of the ith equation
- jth column of A gives the coefficients of x_j in the equations
- ith entry of b gives the constant in the ith equation

Solving linear equations

suppose we have n linear equations in n variables $x_1, \ldots, x_n$

let's write it in compact matrix form as $Ax = b$, where A is an $n \times n$ matrix, and b is an n-vector

suppose A is invertible, *i.e.*, its inverse A^{-1} exists

multiply both sides of $Ax = b$ on the left by A^{-1}:

$$A^{-1}(Ax) = A^{-1}b.$$

lefthand side simplifies to $A^{-1}Ax = Ix = x$, so we've solved the linear equations: $x = A^{-1}b$

so multiplication by *matrix inverse* solves a set of linear equations

some comments:

- $x = A^{-1}b$ makes solving set of 100 linear equations in 100 variables *look* simple, but the notation is hiding alot of work!

- fortunately, it's very easy (and fast) for a computer to compute $x = A^{-1}b$ (even when x has dimension 100, or much higher)

many scientific, engineering, and statistics application programs

- from user input, set up a set of linear equations $Ax = b$
- solve the equations
- report the results in a nice way to the user

when A isn't invertible, $i.e.$, inverse doesn't exist,

- one or more of the equations is redundant
  ($i.e.$, can be obtained from the others)
- the equations are inconsistent or contradictory

(these facts are studied in linear algebra)

in practice: A isn't invertible means you've set up the wrong equations, or don't have enough of them

Solving linear equations in practice

to solve $Ax = b$ ($i.e.$, compute $x = A^{-1}b$) by computer, we don't compute A^{-1}, then multiply it by b (but that would work!)

practical methods compute $x = A^{-1}b$ directly, via specialized methods (studied in numerical linear algebra)

standard methods, that work for any (invertible) A, require about n^3 multiplies & adds to compute $x = A^{-1}b$

but modern computers are very fast, so solving say a set of 500 equations in 500 variables takes only a few seconds, even on a small computer

. . . which is simply **amazing**

Solving equations with sparse matrices

in many applications A has many, or almost all, of its entries equal to zero, in which case it is called *sparse*

this means each equation involves only some (often just a few) of the variables

sparse linear equations can be solved by computer very efficiently, using *sparse matrix techniques* (studied in numerical lineae algebra)

it's not uncommon to solve for hundreds of thousands of variables, with hundreds of thousands of (sparse) equations, even on a small computer

. . . which is **truly amazing**

(and the basis for many engineering and scientific programs, like simulators and computer-aided design tools)