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)