Matrices

Stephen Boyd

EE103
Stanford University

October 9, 2017

Outline

Matrices

- a *matrix* is a reactangular array of numbers, *e.g.*,

$$\begin{bmatrix} 0 & 1 & -2.3 & 0.1 \\ 1.3 & 4 & -0.1 & 0 \\ 4.1 & -1 & 0 & 1.7 \end{bmatrix}$$

- its *size* is given by (row dimension) $\times$ (column dimension)
  e.g., matrix above is 3×4
- *entries* also called *coefficients* or *elements*
- B_{ij} is i, j entry of matrix B
- i is the *row index*, j is the *column index*; indexes start at 1
- two matrices are *equal* (denoted with $=$) if they are the same size and corresponding entries are equal

Matrix shapes

an $m \times n$ matrix A is

- *tall* if $m > n$
- *wide* if $m < n$
- *square* if $m = n$

Column and row vectors

- we consider an $n \times 1$ matrix to be an n-vector
- we consider a 1×1 matrix to be a number
- a $1 \times n$ matrix is called a *row vector*, *e.g.*,

$$\left[\begin{array}{cccc} 1.2 & -0.3 & 1.4 & 2.6 \end{array} \right]$$

which is *not* the same as the (column) vector

$$\left[\begin{array}{r} 1.2 \\ -0.3 \\ 1.4 \\ 2.6 \end{array} \right]$$

Columns and rows of a matrix

- suppose A is an $m \times n$ matrix with entries

$$A_{ij}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

- its jth *column* is (the m-vector)

$$\begin{bmatrix} A_{1j} \\ \vdots \\ A_{mj} \end{bmatrix}$$

- its ith *row* is (the n-row-vector)

$$\begin{bmatrix} A_{i1} & \cdots & A_{in} \end{bmatrix}$$

- *slice* of matrix: $A_{p:q,r:s}$: $(q-p+1) \times (s-r+1)$ matrix with entries A_{ij} with $p \leq i \leq q$, $r \leq j \leq s$

Block matrices

- we can form *block matrices*, whose entries are matrices, such as

$$A = \begin{bmatrix} B & C \\ D & E \end{bmatrix}$$

  where B, C, D, and E are matrices
- B, C, D, and E are *submatrices* or *blocks* of A
- matrices in each block row must have same height (row dimension)
- matrices in each block column must have same width (column dimension)

Column and row representation of matrix

- A is an $m \times n$ matrix
- can express as block matrix with its (m-vector) columns $a_1, \ldots, a_n$

$$A = \left[\begin{array}{cccc} a_1 & a_2 & \cdots & a_n \end{array} \right]$$

- or as block matrix with its (n-row-vector) rows $b_1, \ldots, b_m$

$$A = \left[\begin{array}{c} b_1 \\ b_2 \\ \vdots \\ b_m \end{array} \right]$$

Examples

- *image*. X_{ij} is i, j pixel value in a monochrome image
- *rainfall data*. A_{ij} is rainfall at location i on day j
- *multiple asset returns.* R_{ij} is return of asset j in period i
- *contingency table.* A_{ij} is number of objects with first attribute i and second attribute j
- *feature matrix.* X_{ij} is value of feature i for entity j

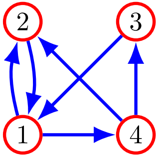
- in each of these, what do the rows and columns mean?

Graph or relation

- a *relation* is a set of pairs of *objects*, labeled $1, \ldots, n$, such as

$$\mathcal{R} = \{(1,2),\ (1,3),\ (2,1),\ (2,4),\ (3,4),\ (4,1)\}$$

- same as *directed graph*



- can represent as $n \times n$ matrix with $A_{ij} = 1$ if $(i,j) \in \mathcal{R}$

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

Special matrices

- $m \times n$ *zero matrix* has all entries zero, written as $0_{m \times n}$ or just 0
- *identity matrix* is square matrix with $I_{ii} = 1$ and $I_{ij} = 0$ for $i \neq j$, *e.g.*,

$$\left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right], \qquad \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

- *sparse matrix*: many entries are zero
  - examples: 0 and I
  - can be stored and manipulated efficiently
  - $\mathbf{nnz}(A)$ is number of nonzero entries

Diagonal and triangular matrices

- *diagonal matrix*: square matrix with $A_{ij} = 0$ when $i \neq j$
- $\mathbf{diag}(a_1, \ldots, a_n)$ is diagonal matrix with $A_{ii} = a_i$, $i = 1, \ldots, n$
- *lower triangular matrix*: $A_{ij} = 0$ for $i < j$
- *upper triangular matrix*: $A_{ij} = 0$ for $i > j$
- examples:

$$\mathbf{diag}(0.2, -3, 1.2) = \left[\begin{array}{rrr} 0.2 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 1.2 \end{array}\right], \qquad \left[\begin{array}{rr} -0.6 & 0 \\ -0.3 & 3.5 \end{array}\right]$$

Transpose

- the *transpose* of an $m \times n$ matrix A is denoted A^T, and defined by

$$(A^T)_{ij} = A_{ji}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, m$$

- for example,

$$\begin{bmatrix} 0 & 4 \\ 7 & 0 \\ 3 & 1 \end{bmatrix}^T = \begin{bmatrix} 0 & 7 & 3 \\ 4 & 0 & 1 \end{bmatrix}$$

- transpose converts column to row vectors (and vice versa)
- $(A^T)^T = A$

Addition, subtraction, and scalar multiplication

- (just like vectors) we can add or subtract matrices of the same size:

$$(A+B)_{ij} = A_{ij} + B_{ij}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

  (subtraction is similar)

- scalar multiplication:

$$(\alpha A)_{ij} = \alpha A_{ij}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

- many obvious properties, *e.g.*,
  - $A + B = B + A$
  - $\alpha(A + B) = \alpha A + \alpha B$
  - $(A + B)^T = A^T + B^T$

Matrix norm

- for $m \times n$ matrix A, we define

$$\|A\| = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}^2 \right)^{1/2}$$

- agrees with vector norm when $n = 1$
- satisfies norm properties:

$$\begin{array}{ll} \|\alpha A\| = |\alpha| \|A\|, & \|A + B\| \leq \|A\| + \|B\|, \\ \|A\| \geq 0, & \|A\| = 0 \text{ only if } A = 0 \end{array}$$

- distance between two matrices: $\|A - B\|$
- (there are other matrix norms, which we won't use)

Outline

Matrix-vector product

- *matrix-vector product* of $m \times n$ matrix A, n-vector x, denoted $y = Ax$, with

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

- for example,

$$\left[\begin{array}{rrr} 0 & 2 & -1 \\ -2 & 1 & 1 \end{array}\right] \left[\begin{array}{r} 2 \\ 1 \\ -1 \end{array}\right] = \left[\begin{array}{r} 3 \\ -4 \end{array}\right]$$

Row interpretation

- $y = Ax$ can be expressed as

$$y_i = b_i^T x, \quad i = 1, \ldots, m$$

  where $b_1^T, \ldots, b_m^T$ are rows of A
- so $y = Ax$ is a 'batch' inner product of all rows of A with x
- example: $A\mathbf{1}$ is vector of row sums of matrix A

Column interpretation

- $y = Ax$ can be expressed as

$$y = x_1 a_1 + x_2 a_2 + \cdots + x_n a_n$$

  where $a_1, \ldots, a_n$ are columns of A
- so $y = Ax$ forms linear combination of columns of A, with coefficients $x_1, \ldots, x_n$
- important example: $Ae_j = a_j$
- columns of A are linearly independent if $Ax = 0$ implies $x = 0$

Outline

Matrices

Matrix-vector multiplication

Examples

General examples

- $0x = 0$, *i.e.*, multiplying by zero matrix gives zero
- $Ix = x$, *i.e.*, multiplying by identity matrix does nothing
- inner product $a^T b$ is matrix-vector multiplication with $1 \times n$ matrix a^T, n-vector b
- $\tilde{x} = Ax$ is de-meaned version of x, with

$$A = \begin{bmatrix} 1-1/n & -1/n & \cdots & -1/n \\ -1/n & 1-1/n & \cdots & -1/n \\ \vdots & & \ddots & \vdots \\ -1/n & -1/n & \cdots & 1-1/n \end{bmatrix}$$

Difference matrix

- $(n-1) \times n$ *difference matrix* is

$$D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ & & \ddots & \ddots & & & \\ & & & \ddots & \ddots & & \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}$$

  $y = Dx$ is $(n-1)$-vector of differences of consecutive entries of x:

$$Dx = \begin{bmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{bmatrix}$$

- $\|Dx\|^2$ (Laplacian) is measure of wiggliness for x a time series

Return matrix – portfolio vector

- R is $T \times n$ matrix of asset returns
- R_{ij} is return of asset j in period i (say, in percentage)
- n-vector h gives portfolio (investments in the assets)
- T-vector Rh is time series of the portfolio return
- $\mathbf{avg}(Rh)$ is the portfolio (mean) return, $\mathbf{std}(Rh)$ is its risk

Feature matrix – weight vector

- $X = [x_1 \ \cdots \ x_N]$ is $n \times N$ *feature matrix*
- column x_j is feature n-vector for object or example j
- X_{ij} is value of feature i for example j
- n-vector w is weight vector
- $s = X^T w$ is vector of scores for each example; $s_j = x_j^T w$

Input – output matrix

- A is $m \times n$ matrix
- $y = Ax$
- n-vector x is *input* or *action*
- m-vector y is *output* or *result*
- A_{ij} is the factor by which y_i depends on x_j
- A_{ij} is the *gain* from input j to output i
- *e.g.*, if A is lower triangular, then y_i only depends on $x_1, \ldots, x_i$

Complexity

- $m \times n$ matrix stored A as $m \times n$ array of numbers (for sparse A, store only $\mathbf{nnz}(A)$ nonzero values)
- matrix addition, scalar-matrix multiplication cost mn flops
- matrix-vector multiplication costs $m(2n-1) \approx 2mn$ flops (for sparse A, around $2\mathbf{nnz}(A)$ flops)