Matrices and Tensors

- Types of matrices
- Matrices with structure
- Special matrices
- A few words on tensors

Types of (square) matrices

- Symmetric $A^T = A$.
- Skew-symmetric $A^T = -A$.
- Hermitian $A^H = A$.
- Skew-Hermitian $A^H = -A$.

- Normal $A^H A = AA^H$.

- Nonnegative $a_{ij} \geq 0, \; i, j = 1, \ldots, n$

- Similarly for nonpositive, positive, and negative matrices

- Unitary $Q^H Q = I$. (for complex matrices)

[Note: Common useage restricts this definition to complex matrices. An *orthogonal matrix* is a unitary *real* matrix – not very natural]

- Orthogonal $Q^T Q = I$ [orthonormal columns]

In this class: W'll call unitary matrix a square matrix with orthonormal columns, whether real or complex. [so: Orthogonal + square = unitary]

➤ The term "orthonormal" matrix is rarely used.

✍1 What is the inverse of a unitary (complex / real) matrix?

✍2 What can you say about the diagonal entries of a skew-symmetric (real) matrix?

✍3 What can you say about the diagonal entries of a Hermitian (complex) matrix?

✍4 What can you say about the diagonal entries of a skew-Hermitian (complex) matrix?

✍5 Which matrices of the following type are also normal: real symmetric, real skew-symmetric, Hermitian, skew-Hermitian, complex symmetric, complex skew-symmetric matrices.

✍6 Find all real 2×2 matrices that are normal.

✍7 Show that a triangular matrix that is normal is diagonal.

Matrices with structure

- Diagonal $a_{ij} = 0$ for $j \neq i$. Notation :

$$A = \text{diag}\,(a_{11}, a_{22}, \ldots, a_{nn})\,.$$

- Upper triangular $a_{ij} = 0$ for $i > j$.
- Lower triangular $a_{ij} = 0$ for $i < j$.
- Upper bidiagonal $a_{ij} = 0$ for $j \neq i$ and $j \neq i + 1$.
- Lower bidiagonal $a_{ij} = 0$ for $j \neq i$ and $j \neq i - 1$.
- Tridiagonal $a_{ij} = 0$ when $|i - j| > 1$.

- Banded $a_{ij} \neq 0$ only when $i - m_l \leq j \leq i + m_u$, 'Bandwidth' = $m_l + m_u + 1$.

- Upper Hessenberg $a_{ij} = 0$ when $i > j + 1$. Lower Hessenberg matrices can be defined similarly.

- Outer product $A = uv^T$, where both u and v are vectors.

- Block tridiagonal generalizes tridiagonal matrices by replacing each nonzero entry by a square matrix.

Special matrices

Vandermonde : Given a column of entries $[x_0, x_1, \cdots, x_n]^T$ put its (component-wise) powers into the columns of a matrix V:

$$V = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^2 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix}$$

✍8 Try the matlab function *vander*

✍9 What does the matrix-vector product Va represent?

✍10 Interpret the solution of the linear system $Va = y$ where a is the unknown. Sketch a 'fast' solution method based on this.

Toeplitz :

➤ Entries are constant along diagonals, i.e., $a_{ij} = r_{j-i}$.

➤ Determined by $m + n - 1$ values r_{j-i}.

$$T = \underbrace{\begin{pmatrix} r_0 & r_1 & r_2 & r_3 & r_4 \\ r_{-1} & r_0 & r_1 & r_2 & r_3 \\ r_{-2} & r_{-1} & r_0 & r_1 & r_2 \\ r_{-3} & r_{-2} & r_{-1} & r_0 & r_1 \\ r_{-4} & r_{-3} & r_{-2} & r_{-1} & r_0 \end{pmatrix}}_{Toeplitz}$$

➤ Toeplitz systems ($m = n$) can be solved in $O(n^2)$ ops.

➤ The whole inverse (!) can be determined in $O(n^2)$ ops.

✍ 11 Explore *toeplitz(c,r)* in matlab.

Hankel : Entries are constant along anti-diagonals, i.e., $a_{ij} = h_{j+i-1}$.
Determined by $m+n-1$ values h_{j+i-1}.

$$H = \underbrace{\begin{pmatrix} h_1 & h_2 & h_3 & h_4 & h_5 \\ h_2 & h_3 & h_4 & h_5 & h_6 \\ h_3 & h_4 & h_5 & h_6 & h_7 \\ h_4 & h_5 & h_6 & h_7 & h_8 \\ h_5 & h_6 & h_7 & h_8 & h_9 \end{pmatrix}}_{Hankel}$$

✍12 Explore *hankel(c,r)* in matlab.

Circulant : Entries in a row are cyclically right-shifted to form next row. Determined by n values.

$$C = \underbrace{\begin{pmatrix} c_1 & c_2 & c_3 & c_4 & c_5 \\ c_5 & c_1 & c_2 & c_3 & c_4 \\ c_4 & c_5 & c_1 & c_2 & c_3 \\ c_3 & c_4 & c_5 & c_1 & c_2 \\ c_2 & c_3 & c_4 & c_5 & c_1 \end{pmatrix}}_{Circulant}$$

✍13 How can you generate a circulant matrix in matlab?

✍14 If C is circulant (real) and symmetric, what can be said about the c_i's?

➤ A simple and important circulant matrix is the up-shift matrix S_n

$$S_5 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

✍15 What is the result of multiplying S_n by a vector? What are the powers of S_n? What is the inverse of S_n?

✍16 Show that

$$C = c_1 I + c_2 S_n + c_3 S_3^2 + \cdots + c_n S_n^{n-1}$$

As a result show that all circulant matrices of the same size commute.

✍17 (Continuation) Use the result of the previous exercise to show that the product of two circulant matrices is circulant.

Sparse matrices

➤ Matrices with very few nonzero entries – so few that this can be exploited.

➤ Many of the large matrices encountered in applications are sparse.

➤ Main idea of “sparse matrix techniques” is not to represent the zeros.

➤ This will be covered in some detail at the end of the course.

A few words on tensors

- A *tensor* is a multidimensional array
- An order N tensor requires N indices:

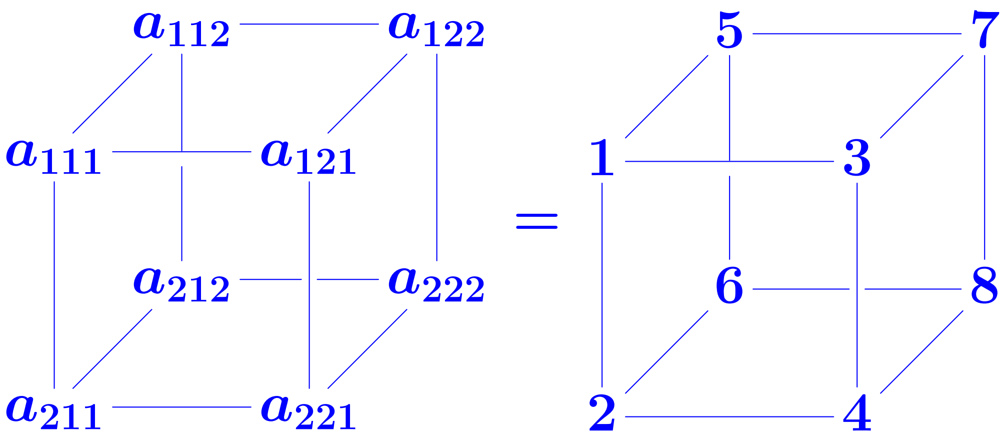
$$\mathcal{A} = (a_{i_1,i_2,\ldots,i_N}) \in \mathbb{R}^{d_1\times d_2\times\cdots\times d_N}$$

- Each d_n is the n-th *dimension* of $\mathcal{A}$.
- A vector is an order-1 tensor and a matrix is an order-2 tensor.
- Order 3 tensor indexed by 3 indices i_1, i_2, i_3 or i, j, k.
- For each fixed i_3 (or k) $a_{:,:,i_3}$ is a matrix - a frontal slice
- $a(i_1, i_2, \cdots, i_{n-1}, :, i_{n+1}, \cdots, i_n)$ is a fiber (a vector) in n-th mode

✍18 How many mode n fibers are there?

➤ Illustration [1st and 2nd indices: top to bottom and left to right, 3rd: front to back]

$\mathcal{A} =$



➤ For above example, frontal slices are the matrices

$$A_1 = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \quad A_2 = \begin{pmatrix} 5 & 7 \\ 6 & 8 \end{pmatrix}$$

➤ The mode-2 fibers are:

$$f_1 = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \; f_2 = \begin{pmatrix} 2 \\ 4 \end{pmatrix}, \; f_3 = \begin{pmatrix} 5 \\ 7 \end{pmatrix}, \; f_4 = \begin{pmatrix} 6 \\ 8 \end{pmatrix}$$

➤ Note order: $a(x, :, y) : a(1, :, 1), a(2, :, 1), a(1, :, 2), a(2, :, 2)$

Unfolding and mode-n products Useful for visualizing tensors of order $N > 3$.

➤ Unfolding of a tensor along mode n is a matrix $A_{(n)}$ of dimension

$$d_n \times (d_1 \cdots d_{n-1} d_{n+1} \cdots d_N).$$

➤ Columns of $A_{(n)}$ are all mode n fibers of $\mathcal{A}$

➤ For the above example tensor $\mathcal{A}$, the three mode-n unfoldings are

$$A_{(1)} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{bmatrix}, \quad A_{(2)} = \begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{bmatrix}, \quad A_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}.$$

➤ Note: In Python/Pytorch tensors are stored differently from multidimensional arrays in matlab

Mode-n Product or tensor-matrix product Given a tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$ and a matrix $M \in \mathbb{R}^{c_n \times d_n}$, the mode-$n$ product is a tensor

$$\mathcal{B} = \mathcal{A} \otimes_n M \in \mathbb{R}^{d_1 \times \cdots \times d_{n-1} \times c_n \times d_{n+1} \cdots \times d_N} \quad \text{where}$$

$$b(i_1, \dots, i_{n-1}, j_n, i_{n+1}, \dots, i_N) = \sum_{i_n=1}^{d_n} a(i_1, \dots, i_{n-1}, i_n, i_{n+1}, \dots, i_N)\, m(j_n, i_n) \text{ for } j_n = 1, 2, \dots, c_n$$

➤ Matrix interpretation : $\boxed{B_{(n)} = M A_{(n)}.}$

➤ Here are a couple of properties of tensor-matrix multiplication:

- For $m \neq n$, and matrices F and G of appropriate dimensions,

$$(\mathcal{A} \otimes_n F) \otimes_m G = (\mathcal{A} \otimes_m G) \otimes_n F.$$

- For any n, and for matrices F and G of appropriate dimensions,

$$(\mathcal{A} \otimes_n F) \otimes_n G = \mathcal{A} \otimes_n (GF).$$

➤ for a general n, the mode-n product of two matrices is not defined, can safely omit the parentheses and write $(\mathcal{A} \otimes_n F) \otimes_m G$ as $\mathcal{A} \otimes_n F \otimes_m G$.