Mathematical basics

Dr. Tushar Sandhan

sandhan@iitk.ac.in

Contents

Matrix multiplication

- "When A is a *mxn* matrix & B is a *kxl* matrix, AB is only possible if *n=k*. The result will be an *mxl* matrix"

$$
\begin{array}{cc}
 & \mathbf{n} \\
\mathbf{m} & \begin{matrix} A_1 & A_2 & A_3 \\ A_4 & A_5 & A_6 \\ A_7 & A_8 & A_9 \\ A_{10} & A_{11} & A_{12} \end{matrix}
\end{array}
\quad \text{x} \quad
\begin{array}{cc}
\mathbf{l} & \\
\begin{matrix} B_{13} & B_{14} \\ B_{15} & B_{16} \\ B_{17} & B_{18} \end{matrix} & \mathbf{k}
\end{array}
\quad = \mathbf{m} \text{ x } \mathbf{l} \text{ matrix}
$$

Number of columns in A = Number of rows in B

Vector Products

- **Two vectors:**

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

- **Inner product** = scalar

$$\mathbf{x}^T \mathbf{y} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = x_1 y_1 + x_2 y_2 + x_3 y_3 = \sum_{i=1}^{3} x_i\, y_i$$

- **Outer product** = matrix

$$xy^T = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} = \begin{bmatrix} x_1 y_1 & x_1 y_2 & x_1 y_3 \\ x_2 y_1 & x_2 y_2 & x_2 y_3 \\ x_3 y_1 & x_3 y_2 & x_3 y_3 \end{bmatrix}$$

Properties of matrix multiplication

- In this table *A, B*, and *C* are *n x n* matrices, *I* is the *n x n* identity matrix, and *O* is the *n x n* zero matrix.

Property	Example
The commutative property of multiplication does not hold!	$AB \neq BA$
Associative property of multiplication	$(AB)C = A(BC)$
Distributive properties	$A(B + C) = AB + AC$
	$(B + C)A = BA + CA$
Multiplicative identity property	$IA = A$ and $AI = A$
Multiplicative property of zero	$OA = O$ and $AO = O$
Dimension property	The product of an $m \times n$ matrix and an $n \times k$ matrix is an $m \times k$ matrix.

sandhan@iitk.ac.in

Identity matrix

- An identity matrix is **a square matrix in which all the elements of principal diagonals are one, and all other elements are zeros**. It is denoted by the notation "I_n" or simply "I".

$$I_n = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}.$$

  - For any nxn matrix $\boldsymbol{A}$, we have $\boldsymbol{A} \boldsymbol{I_n} = \boldsymbol{I_n} \boldsymbol{A} = \boldsymbol{A}$
  - For any nxm matrix $\boldsymbol{A}$, we have $\boldsymbol{I_n} \boldsymbol{A} = \boldsymbol{A}$, and $\boldsymbol{A} \boldsymbol{I_m} = \boldsymbol{A}$ (so 2 possible matrices)

sandhan@iitk.ac.in

Inverse matrix

- **Notation.** A common notation for the inverse of a matrix ***A*** is $\boldsymbol{A}^{-1}$.

$$\boxed{A\, A^{-1} = A^{-1}\, A = I_n\, .}$$

- The inverse matrix is unique when it exists. So if ***A*** is invertible, then $\boldsymbol{A}^{-1}$ is also invertible and then $(A^T)^{-1} = (A^{-1})^T$

sandhan@iitk.ac.in

Inverse matrix

- For a *X x X* square matrix: $$A = \begin{pmatrix} x_{1,1} & \dots & x_{1,j} \\ \vdots & \ddots & \vdots \\ x_{i,1} & \dots & x_{i,j} \end{pmatrix}$$
- The inverse matrix is: $$A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} \operatorname{cof}(A, x_{1,1}) & \dots & \operatorname{cof}(A, x_{1,j}) \\ \vdots & \ddots & \vdots \\ \operatorname{cof}(A, x_{i,1}) & \dots & \operatorname{cof}(A, x_{i,j}) \end{pmatrix}^T$$
- E.g.: 2x2 matrix $$\mathbf{A}^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

- **Properties:**
  - $(A^{-1})^{-1} = A$
  - $(A \times B)^{-1} = B^{-1} \times A^{-1}$
  - $(A^T)^{-1} = (A^{-1})^T$

Diagonal Matrix:

- A square matrix in which every element except the principal diagonal elements is zero is called a Diagonal Matrix.

$$A = \begin{bmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{bmatrix}_{n \times n}$$

Examples $\begin{bmatrix} 3 & 0 \\ 0 & 4 \end{bmatrix}$ $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -4 \end{bmatrix}$

Transpose Matrix

- **Properties:** Transpose matrix is obtained by interchanging the rows and columns.
    - The transpose of matrix A is denoted as A^T.

- $(A^T)^T = A$
- $(A+B)^T = A^T + B^T$
- $(A \times B)^T = B^T \times A^T$
- $(kA)^T = kA^T$

Trace matrix

- The trace of an n × n square matrix A is defined as

$$\mathrm{tr}(\mathbf{A}) = \sum_{i=1}^{n} a_{ii} = a_{11} + a_{22} + \cdots + a_{nn}$$

where aii denotes the entry on the *ith* row and *ith* column of A. The entries of A can be real numbers or (more generally) complex numbers.

- The trace is not defined for non-square matrices.

sandhan@iitk.ac.in

Determinant of matrix

- Determinants can only be found for square matrices.
- For a 2x2 matrix A, **det(A) = ad-bc**. Lets have at closer look at that:

$$\det(A) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc$$

- A matrix A has an inverse matrix A^{-1} if and only if **det(A)≠0**.

Eigendecomposition

- Eigenvectors and eigenvalues are vectors and numbers associated to square matrices. Together they provide the eigen-decomposition of a matrix which analyzes the structure of this matrix.

- There are several ways to define eigenvectors and eigenvalues, the most common approach defines an eigenvector of the matrix *A* as a vector *u* that satisfies the following equation:

$$Au = \lambda u \ .$$

- when rewritten, the equation becomes:

$$(A - \lambda I)u = 0 \ ,$$

- where λ is a scalar called the eigenvalue associated to the eigenvector.

- In a similar manner, we can also say that a vector *u* is an eigenvector of a matrix *A* if the length of the vector (but not its direction) is changed when it is multiplied by *A*.

Eigendecomposition

- Example:

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix}$$

has the eigenvectors:

$$\mathbf{u}_1 = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \quad \text{with eigenvalue} \quad \lambda_1 = 4$$

and

$$\mathbf{u}_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad \text{with eigenvalue} \quad \lambda_2 = -1$$

The length of u1 and u2 is changed when one of these two vectors is multiplied by the matrix A.

Eigendecomposition

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix} = 4 \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 12 \\ 8 \end{bmatrix}$$

and

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = -1 \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

For most applications we normalize the eigenvectors (i.e., transform them such that their length is equal to one):

$$\mathbf{u}^{\mathsf{T}}\mathbf{u} = 1$$

Eigendecomposition

- Traditionally, we put together the set of eigenvectors of A in a matrix denoted U. Each column of U is an eigenvector of A. The eigenvalues are stored in a diagonal matrix (denoted Λ), where the diagonal elements gives the eigenvalues (and all the other values are zeros). We can rewrite the first equation as:

$$\mathbf{AU} = \mathbf{U\Lambda}$$

or also as:

$$\mathbf{A} = \mathbf{U\Lambda U}^{-1} \, .$$

For the previous example we obtain:

$$\mathbf{A} = \mathbf{U\Lambda U}^{-1}$$

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

$$= \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} .$$

- It is important to note that not all matrices have eigenvalues.
- For example, the matrix $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ does not have eigenvalues.

sandhan@iitk.ac.in

Eigendecomposition with *Python Code*

```
import numpy as np
from numpy.linalg import eig
```

```
A = np.array([ [2, 3], [2, 1] ])
print (A)
```

```
[[2 3]
 [2 1]]
```

```
#getting the eigenvalues and eigenvvector of M
Lambda, U = np.linalg.eig(A)
```

```
print(U)
```

```
[[ 0.83205029 -0.70710678]
 [ 0.5547002   0.70710678]]
```

```
print(Lambda)
```

```
[ 4. -1.]
```

```
# getting U inverse
inv_U = np.linalg.inv(U)
inv_U
```

```
array([[ 0.72111026,  0.72111026],
       [-0.56568542,  0.84852814]])
```

```
Λ = np.diag(Lambda)
Λ
```

```
array([[ 4.,  0.],
       [ 0., -1.]])
```

```
def round(values, decs=0): # we don't want to include
  return np.round(values*10**decs)/(10**decs)
vec = np.dot(U,np.dot(Λ, inv_U)) # taking the product
round(vec)
```

```
array([[2., 3.],
       [2., 1.]])
```

Eigendecomposition with *Matlab Code*

```
clc
clear all
A = [2 3; 2 1];
Lambda = eig(A)

Lambda =
     4
    -1

[U,D] = eig(A)

U =
    0.8321   -0.7071
    0.5547    0.7071
```

```
D =
     4     0
     0    -1

U=[0.8321 -0.7071; 0.5547 0.7071];
U_inv = inv(U)

U_inv =
    0.7211    0.7211
   -0.5657    0.8486

A1=U*D*inv(U)

A1 =
    2.0001    3.0001
    1.9999    0.9999
```

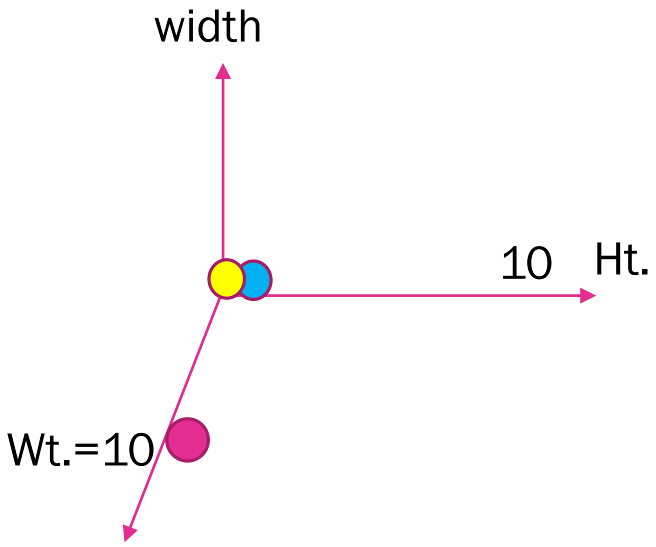
- Quiz1 = total weightage of the course is 10%

	Wt.	Ht	width
Sample#1:	10	1.2	0.5
Sample#2:	15	0.9	0.8

⋮

0.5 0.9 0.8

Thank you



- To make zero mean: subtract the mean from **all**
- To make it unit var: divide **all** by std. dev.
- Do it independently on each feature = each column

sandhan@iitk.ac.in