Handbook

Mark Gao

2023

Contents

I Math 3

1 Linear Algebra . . . 3
1.1 Matrix Operations . . . 3
1.2 Inverse . . . 3
1.3 Determinant . . . 3
1.4 Square Matrices . . . 3
1.5 Eigenvalues and Eigenvectors . . . 4

2 Calculus . . . 4
2.1 Derivative and integral foundations . . . 4
2.2 Multivariable Topics . . . 4

3 Probability . . . 4
3.1 Probabilitly foundations . . . 4
3.2 Markov Chains . . . 4
3.3 Renewal Processes . . . 4

II Computer Science 5

4 Object Oriented Programming . . . 5
4.1 Java foundations . . . 5
4.2 Scope & Encapsulation . . . 5

5 Data Structures & Algorithms . . . 5

6 Machine Learning . . . 5

III Finance 6

7 Fixed Income . . . 6

Part I
Math

1 Linear Algebra

1.1 Matrix Operations

Use the matrices A and B for the examples to follow.

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

The addition and subtraction of matrices happens element-wise.

$$A + B = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} \\ a_{21} + b_{21} & a_{22} + b_{22} \end{pmatrix} \tag{1}$$

The multiplication operation is known as the **dot product**.

$$A \cdot B = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{pmatrix} \tag{2}$$

The division operation does not exist for matrices. There is no inverse operation to the dot product.

Note that **left-multiplication** and **right-multiplication** results in different outcomes. To isolate B in the equation $ABC = D$, right-multiply C then left-multiply A:

$$AB = DC$$

$$B = ADC$$

The **cross product** is a new operation to matrices. For a 2×2 matrix, the cross product is the same as its determinant.

$$A \times B = \det A - \det B = a_{11}a_{22} - a_{12}a_{21} - (bb - bb) \tag{3}$$

The cross product is more meaningful for matrices 3×3 or bigger.

$$A \times B =)i(+)j(+)k(\tag{4}$$

The **transpose** of a matrix is flipped along its diagonal. Where the matrix is not square, this diagonal is imagined. For example given the 2×3 matrix C:

$$\begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{pmatrix}^T = \begin{pmatrix} c_{11} & c_{21} \\ c_{12} & c_{22} \\ c_{13} & c_{23} \end{pmatrix} \tag{5}$$

1.2 Inverse

The **inverse** of a matrix should satisfy $AA^{-1} = I$. To compute A^{-1},

$$AI = B \tag{6}$$

1.3 Determinant

1.4 Square Matrices

A **diagonal matrix** is a square matrix which is comprised of non-zero values along the diagonal and zeros everywhere else. The **identity matrix** is a square matrix with 1's on the diagonal.

$$I = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{7}$$

1.5 Eigenvalues and Eigenvectors

2 Calculus

2.1 Derivative and integral foundations

2.2 Multivariable Topics

3 Probability

3.1 Probabilitly foundations

3.2 Markov Chains

3.3 Renewal Processes

Part II

Computer Science

4 Object Oriented Programming

This section will mostly be covered in Java, with analogies drawn to Python. This section will cover:

- Scope and encapsulation
- Polymorphism

4.1 Java foundations

```java
class MyClass{
    private int x = 123;

    public MyClass(int a) {
        ...
    }
}
```

wtf? hello

4.2 Scope & Encapsulation

5 Data Structures & Algorithms

6 Machine Learning

Part III
Finance

7 Fixed Income