Linear Algebra

United College of Engineering and Research

Anubhav Kesarwani

January 2025

Preface

Motivation to learn Linear Algebra

Linear algebra is a branch of mathematics that is widely used throughout science and engineering. Linear algebra is a form of continuous mathematics.

Linear algebra is the branch of mathematics concerning linear equations, linear maps, and their representations in vector spaces and through matrices. Linear algebra is central to almost all areas of mathematics.

There are several prerequisites for a better understanding of Machine Learning and (especially) Deep Learning algorithms. In order to grasp the nitty-gritties of higher ML concepts, one needs to have a strong base in higher mathematics.

Linear Algebra is the mathematical foundation that solves the problem of representing data as well as computations in machine learning models.

Hence, the concepts of linear algebra are crucial for understanding the theory behind machine learning, especially for deep learning.

Contents

Chapter 1

Mathematical Objects

Mathematical objects are what we talk and write about when we do math. Numbers, functions, triangles, matrices, groups and more complicated things such as vector spaces and infinite series are all examples of mathematical objects. Mathematical objects are abstract objects. They are not physical objects, but we think about them and talk about them as if they actually existed. Mathematical objects have certain properties that other kinds of abstract objects may not have. In particular, unlike other kinds of abstract objects, math objects are inert:

- Math objects don't move or change over time.
- Math objects don't interact with other objects or with the real world.

In particular, we are concerned with the study of following mathematical objects in Linear Algebra amongst others:

1. Scalars
2. Vectors
3. Matrices
4. Tensors

Furthermore, it should be noted that higher mathematics is nothing but layers upon layers of abstraction. Intuitively, it would be right to consider **Sets** as the foundation of Linear Algebra. Therefore, an understanding of Set Theory is essential to grasp Linear Algebra with ease.

- a line is a set of points
- a plane is a set of lines
- a vector is a set of points
- a matrix is a set of vectors

1.1 Scalars

A scalar is just a single number, in contrast to most of the other objects studied in linear algebra, which are usually arrays of multiple numbers. When we introduce them, we specify what kind of number they are. For example, we might say:

Let $s \in \mathbb{R}$ be the slope of the line, while defining a real-valued scalar, or,

Let $n \in \mathbb{N}$ be the number of units, while defining a natural number scalar.

1.2 Vectors

A vector is an array of numbers. The numbers are arranged in order. We can identify each individual number by its index in that ordering.

$$\mathbf{x} := \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \ x_i \in \mathbb{R} \tag{1.1}$$

The first element of $\mathbf{x}$ is $\mathbf{x}_1$, the second element is $\mathbf{x}_2$, and so on. If $\mathbf{x}_i \in \mathbb{R} \ \forall \ i \in \{1, 2, 3, ..., n\}$, then the vector $\mathbf{x}$, having n dimensions, lies in the set formed by taking the Cartesian product of $\mathbb{R}$ n times, denoted as $\mathbf{x} \in \mathbb{R}^n$.

One can think of vectors as identifying points in space, with each element giving the coordinate along a different axis. Sometimes we need to index a set of elements of a vector. In this case, we define a set containing the indices and write the set as a subscript.

For example, to access $\mathbf{x}_1$, $\mathbf{x}_3$ and $\mathbf{x}_6$, we define the set $S = \{1, 3, 6\}$ and write $\mathbf{x}_S$. We use the $-$ sign to index the complement of a set. For example, $\mathbf{x}_{-1}$ is the vector containing all elements of $\mathbf{x}$ except for $\mathbf{x}_1$. Similarly, $\mathbf{x}_{-S}$ is the vector containing all elements of $\mathbf{x}$ except for $\mathbf{x}_1$, $\mathbf{x}_3$ and $\mathbf{x}_6$.

$$f(x) = y, \text{ where } x \in \mathbf{x} \text{ and } y \in \mathbf{y}. \tag{1.2}$$

The ultimate goal of Machine Learning is learning functions from data, i.e., transformations or mappings from the domain onto the range of a function.

The domain $\mathbf{x}$ is usually a vector of variables or features mapping onto a vector of target values.

1.3 Matrices

A matrix is a 2-D array of numbers, so each element is identified by two indices instead of just one.

$$\mathbf{A} := \begin{bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & \dots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m,1} & A_{m,2} & \dots & A_{m,n} \end{bmatrix}, \text{ where } A_{i,j} \in \mathbb{R} \tag{1.3}$$

If a real-valued matrix $\mathbf{A}$ has a height of m (rows) and a width of n (columns), then we say that $\mathbf{A} \in \mathbb{R}^{m \times n}$. The i-th row of $\mathbf{A}$ is denoted by $\mathbf{A}_{i,:}$ and the j-th column of $\mathbf{A}$ is denoted by $\mathbf{A}_{:,j}$. Suppose by applying some function f to $\mathbf{A}$, the resultant matrix is $\mathbf{B}$, then the (i, j)-th element of $\mathbf{B}$ is given by $f(\mathbf{A})_{i,j}$.

1.4 Tensors

A tensor is an array with more than two axes. In the general case, an array of numbers arranged on a regular grid with a variable number of axes is known as a tensor. Suppose a tensor named A has three dimensions, then the element of A at coordinates (i, j, k) is denoted as $\mathsf{A}_{i,j,k}$.

1.5 A note on the transpose of matrices

The transpose of a matrix is the mirror image of the matrix across its principal diagonal line. The transpose of a matrix $\mathbf{A}$ is denoted by $\mathbf{A}^\mathsf{T}$.

$$\mathbf{A} = \begin{bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & \dots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m,1} & A_{m,2} & \dots & A_{m,n} \end{bmatrix} \tag{1.4}$$

$$\mathbf{A}^\mathsf{T} = \begin{bmatrix} A_{1,1} & A_{2,1} & \dots & A_{m,1} \\ A_{1,2} & A_{2,2} & \dots & A_{m,2} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1,n} & A_{2,n} & \dots & A_{m,n} \end{bmatrix} \tag{1.5}$$

Vectors can be thought of as matrices that contain only one column. Thus, the transpose of a vector is therefore a matrix with only one row. This can be denoted as $\mathbf{x} = \big(x_1, x_2, x_3\big)^\mathsf{T}$.

A scalar can be thought of as a matrix with only a single entry. From this, we can see that a scalar is its own transpose: $a = a^\mathsf{T}$.

Chapter 2

Properties of Matrix Operations

2.1 Properties of Matrix Operations

The operations are as follows:

2.1.1 Addition

If $\mathbf{A}$ and $\mathbf{B}$ are matrices of the same size $m \times n$, then $\mathbf{A} + \mathbf{B}$, their sum, is a matrix of size $m \times n$.

2.1.2 Multiplication by Scalars

If $\mathbf{A}$ is a matrix of size $m \times n$ and α is a scalar, then $\alpha\mathbf{A}$ is a matrix of size $m \times n$.

2.1.3 Matrix Multiplication

If $\mathbf{A}$ is a matrix of size $m \times n$ and $\mathbf{B}$ is a matrix of size $n \times p$, then the product $\mathbf{AB}$ is a matrix of size $m \times p$.

2.1.4 Vectors

A vector of length n can be treated as a matrix of size $n \times 1$, and the operations of vector addition, multiplication by scalars, and multiplying a matrix by a vector agree with the corresponding matrix operations.

2.1.5 Transpose

If $\mathbf{A}$ is a matrix of size $m \times n$, then its transpose $\mathbf{A}^\mathsf{T}$ is a matrix of size $n \times m$.

2.1.6 Identity Matrix

An identity matrix is a matrix that does not change any vector when we multiply that vector by that matrix. We denote the identity matrix that preserves n-dimensional vectors as $\mathbf{I}_n$. Formally, $\mathbf{I}_n \in \mathbb{R}^{n\times n}$, and

$$\forall\ \mathbf{x} \in \mathbb{R}^n,\ \mathbf{I}_n\mathbf{x} = \mathbf{x}. \tag{2.1}$$

$\mathbf{I}_n$ is the $n \times n$ identity matrix; its principal diagonal elements are equal to 1 and its off-diagonal elements are equal to 0.

2.1.7 Zero Matrix

It is denoted by 0, the matrix of all zeroes (of relevant size).

2.1.8 Inverse

If $\mathbf{A}$ is a square matrix, then its inverse $\mathbf{A}^{-1}$ is a matrix of the same size. The matrices that have nonzero determinant have inverses and are called invertible.

For square matrices,

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_n \tag{2.2}$$

In many cases, we can treat addition and multiplication of matrices as addition and multiplication of numbers. However, there are some differences between operations with matrices and operations with numbers:

- Properties such as associative, distributive, and commutative are followed in scalar multiplication and matrix addition.
- Matrix multiplication does not commute.
- In general, $\mathbf{AB} \neq \mathbf{BA}$, even if $\mathbf{A}$ and $\mathbf{B}$ are both square matrices. If $\mathbf{AB} = \mathbf{BA}$, then we say that $\mathbf{A}$ and $\mathbf{B}$ commute.
- For a general matrix $\mathbf{A}$, we cannot say that $\mathbf{AB} = \mathbf{AC}$ yields $\mathbf{B} = \mathbf{C}$. However, if we know that $\mathbf{A}$ is invertible, then we can multiply both sides of the equation $\mathbf{AB} = \mathbf{AC}$ to the left by $\mathbf{A}^{-1}$ and get $\mathbf{B} = \mathbf{C}$.
- The equation $\mathbf{AB} = 0$ does not necessarily yield $\mathbf{A} = 0$ or $\mathbf{B} = 0$. For example, take:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{2.3}$$

2.2 A Note on the Methods of Solving a System of Linear Equations

Apart from the usual (direct) methods of solving a system of linear equations, which includes, Elimination Method, Substitution Method, and Cross Multiplication Method, there are various other methods of solving a system of linear equations:

- **Matrix Method**:
  - Cramer's Rule
  - Gaussian Elimination
  - Gauss-Jordan Method
  - Triangularization Method
  - Cholesky Method
  - Partition Method
- **Iterative Methods**:
  - Jacobi Iterative Method
  - Gauss-Seidel Iterative Method
  - SOR Method

It might seem overwhelming at first but as other properties of and special types of matrices are introduced, these methods will become easier to grasp. However, it must be noted that when it comes to solving a system of linear equations, in our case, we require a fast and efficient algorithm, so it would be fine not having some if not most of these methods on your fingertips.

Note: *This section will be expanded in the future.*

Chapter 3

System of Linear Equations

3.1 Introduction

Using the linear algebra notations and the now defined mathematical objects, we can write a system of linear equations as follows:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3.1}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a known matrix, $\mathbf{b} \in \mathbb{R}^{m}$ is a known vector, and $\mathbf{x} \in \mathbb{R}^{n}$ is a vector of unknown variables.

3.2 Matrix-Vector Product

In general, an element of vector $\mathbf{b}$ can be expressed as:

$$\mathbf{b}_i = \mathbf{A}_{i,:}\mathbf{x} \tag{3.2}$$

Matrix-vector product notation provides a more compact representation for equations of this form.

3.3 Solving the System of Equations

To solve for the unknown vector in the equation:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3.3}$$

Pre-multiply both sides of the equation by the inverse of matrix $\mathbf{A}$:

$$\begin{aligned} \mathbf{A}^{-1}\mathbf{A}\mathbf{x} &= \mathbf{A}^{-1}\mathbf{b} && (3.4) \\ \mathbf{I}_n\mathbf{x} &= \mathbf{A}^{-1}\mathbf{b} && (3.5) \\ \mathbf{x} &= \mathbf{A}^{-1}\mathbf{b} && (3.6) \end{aligned}$$

3.4 Existence of the Inverse

When $\mathbf{A}^{-1}$ exists, several different algorithms can find it in closed form. In theory, the same inverse matrix can then be used to solve the equation many times for different values of $\mathbf{b}$. $\mathbf{A}^{-1}$ is primarily useful as a theoretical tool, however, and should not actually be used in practice for most software applications. Because $\mathbf{A}^{-1}$ can be represented with only limited precision on a digital computer, algorithms that make use of the value of $\mathbf{b}$ can usually obtain more accurate estimates of $\mathbf{x}$.

3.5 Uniqueness and Existence of Solutions

For $\mathbf{A}^{-1}$ to exist, the equation $\mathbf{A}\mathbf{x} = \mathbf{b}$ must have *exactly one solution for every value of* $\mathbf{b}$, i.e., a unique value of $\mathbf{x}$ for every value of $\mathbf{b}$.

It is also possible for the system of equations to have no solutions or infinitely many solutions for some values of $\mathbf{b}$. It is not possible, however, to have more than one but less than infinitely many solutions for a particular $\mathbf{b}$; if both $\mathbf{x}$ and $\mathbf{y}$ are solutions, then:

$$\mathbf{z} = \alpha\mathbf{x} + (1-\alpha)\mathbf{y} \tag{3.7}$$

is also a solution for any real α.

3.6 Linear Combinations

To analyze how many solutions the equation has, think of the columns of $\mathbf{A}$ as specifying different directions we can travel in from the origin (the point specified by the vector of all zeros), then determine how many ways there are of reaching $\mathbf{b}$. In this view, each element of $\mathbf{x}$ specifies how far we should travel in each of these directions, with $\mathbf{x}_i$ specifying how far to move in the direction of column i:

$$\mathbf{A}\mathbf{x} = \sum_i \mathbf{x}_i \mathbf{A}_{:,i} \tag{3.8}$$

In general, this kind of operation is called a **Linear Combination**. Any n-vector can be represented as a linear combination of the standard unit vectors. For example:

$$\begin{aligned} \mathbf{x} &= \begin{pmatrix} a \\ b \\ c \end{pmatrix} && (3.9) \\ &= a \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + b \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + c \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} && (3.10) \end{aligned}$$

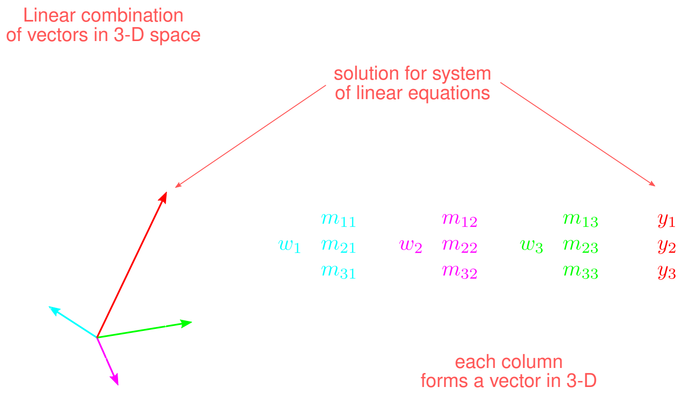

Figure 3.1: Linear combination of vectors

3.7 Span of Vectors

Formally, a linear combination of some set of vectors $\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}\}$ is given by multiplying each vector $\mathbf{v}$ by a corresponding scalar coefficient and adding the results:

$$\sum_i c_i \mathbf{v}^{(i)} \tag{3.11}$$

This is known as the **Span**. The span of a set of vectors is the set of all points obtainable by linear combination of the original vectors. This particular span is known as the **column space**, or the **range**, of **A**.

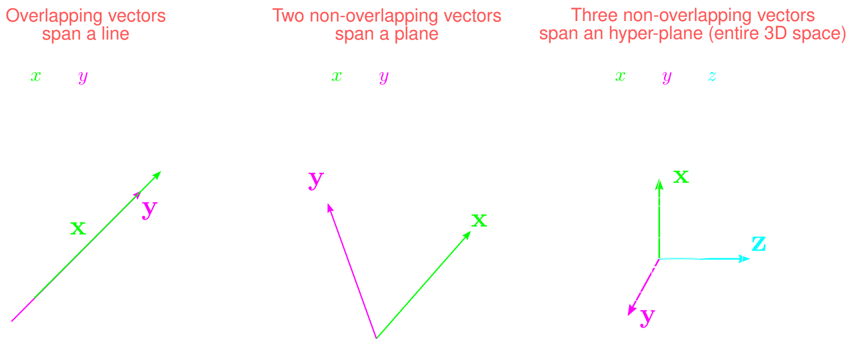
Overlapping vectors
span a line
x
y
x
y
Two non-overlapping vectors
span a plane
x
y
y
x
Three non-overlapping vectors
span an hyper-plane (entire 3D space)
x
y
z
x
z
y

Figure 3.2: Span of vectors

3.8 Column Space and Solutions

A solution of the equation $\mathbf{Ax} = \mathbf{b}$ exists if $\mathbf{b}$ lies in the span of the columns of $\mathbf{A}$. To have a solution for all values of $\mathbf{b} \in \mathbb{R}^m$, the column space of $\mathbf{A}$ must be all of $\mathbb{R}^m$, which implies that $\mathbf{A}$ must have at least m columns ($n > m$). Otherwise, the dimensionality of $\mathbf{A}$ would be reduced (for potential values of $\mathbf{b}$ there would be no solution). For example, consider a 3×2 matrix. The span of vectors results in a $2D$ plane within $\mathbb{R}^3$ but the target $\mathbf{b}$ is $3D$. Modifying the value of $\mathbf{x}$ at best enables us to trace out a $2D$ plane within $\mathbb{R}^3$. The equation has a solution if and only if $\mathbf{b}$ lies on that plane.

3.9 Linear Dependence and Independence

Now, $n > m$ is only a necessary condition for every point to have a solution. It is possible for some of the columns to be redundant (mutually linearly dependent; one vector being a linear combination of another vector), which implies that the addition of any linearly dependent vector to a set of vectors does not contribute any new points to the span of that set of vectors.

This means that for the column space of the matrix to encompass all of $\mathbb{R}^m$, the matrix must contain at least one set of *exactly* m linearly independent columns. This condition is both necessary and sufficient for solutions to all values of $\mathbf{b} \in \mathbb{R}^m$.

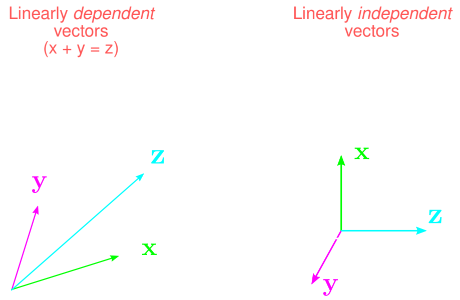
Linearly *dependent* vectors (x + y = z)
Linearly *independent* vectors
z
y
x
x
z
y

Figure 3.3: Linearly dependent and independent vectors

Set of two or more vectors are said to be linearly dependent if at least one of the vectors can be obtained as a linear combination of other vectors in the set. This is known as Linear Dependence.

3.10 Square Matrices and Singularity

Note: No set of m-dimensional vectors can have more than m mutually linearly independent columns, but a matrix with more than m columns may have more than one such set.

We also need to ensure that whether the equation is over-constrained; thus, depending on the nature of linear dependence of the columns of the matrix, the system of equations has either zero (when it is not possible to satisfy all equations simultaneously, in the case of mutually linearly independent columns) or infinitely many solutions (when it is possible to represent the general solution as the linear combination of vectors using arbitrary constants, in the case of mutually linearly independent columns). Thus, the matrix must have at most m columns, so that the equation has at most one solution for each value of $\mathbf{b}$.

3.11 Singular and Non-Singular Matrices

It is now evident that in the coefficient matrix of the equation $\mathbf{Ax} = \mathbf{b}$, all the columns of the matrix must be linearly independent and that the number of columns must be equal to the number of rows of the matrix, i.e., $n = m$. This means that the matrix $\mathbf{A}$ is a *square matrix*.

A square matrix with linearly dependent columns is known as *singular* (zero determinant). A square matrix has linearly independent columns if and only if the matrix is *non-singular*.

Thus, it is only possible to find the inverse of a non-singular square matrix, having linearly independent columns.

3.12 Matrix Inversion Method: Limitations

Note that the Matrix Inversion Method of solving a system of equations cannot be applied in cases where we cannot find the inverse of the coefficient matrix (obviously). Such as in cases where the matrix is not square or it is square but it is also singular (has a zero determinant). In such cases, we shall use methods other than the Matrix Inversion method in order to solve for a system of linear equations.

Chapter 4

Norms

4.1 Introduction

In Machine Learning, the size of vectors is measured using a function called a **norm**. Formally, the L^p norm is given by:

$$\|\mathbf{x}\|_p = \left(\sum_i |x_i|^p \right)^{\frac{1}{p}} \tag{4.1}$$

where $p \in \mathbb{R}, p \geq 1$.

The L^p norm is a *mapping function*. It is used to map vectors to non-negative values. It satisfies the following properties:

- $f(x) = 0 \implies x = 0$
- $f(x + y) \leq f(x) + f(y)$ (the **triangle inequality**)
- $\forall \alpha \in \mathbb{R}, f(\alpha x) = |\alpha| \, f(x)$

The norm of a vector $\mathbf{x}$ can be visualized as the distance of the point $\mathbf{x}$ from the origin in the vector space.

4.2 Euclidean Norm

For a vector $\mathbf{x}$, the L^2 norm, also known as the **Euclidean norm**, is given by:

$$\mathbf{x} = x\hat{\imath} + y\hat{\jmath} \tag{4.2}$$

Then the distance of the *point* $\mathbf{x}$ from the origin is given by the **Distance formula** as follows:

$$\|\mathbf{x}\| = \sqrt{x_1^2 + \cdots + x_n^2} \tag{4.3}$$

In essence, the calculated distance is **Euclidean distance** of the vector $\mathbf{x}$.

For a vector $\mathbf{x}$ of n dimensions,

$$L^2 = \|\mathbf{x}\|_2 = \sqrt{x_1^2 + \cdots + x_n^2} \tag{4.4}$$

In machine learning, the Euclidean norm is quite frequently used (and as seen from the coordinate geometry analogy), it is represented simply by $\|\mathbf{x}\|$.

$$\begin{align} L^2 = \|\mathbf{x}\|_2 = \|\mathbf{x}\| = \sqrt{x_1^2 + \cdots + x_n^2} \tag{4.5} \\ \left(L^2\right)^2 = \|\mathbf{x}\|_2^2 = \|\mathbf{x}\|^2 = x_1^2 + \cdots + x_n^2 \tag{4.6} \end{align}$$

Mathematically and computationally, the squared L^2 norm is more convenient to use. For example: the derivatives of the squared L^2 norm with respect to each element of $\mathbf{x}$ each depend only on the corresponding element of $\mathbf{x}$, while all of the derivatives of the L^2 norm depend on the entire vector.

For a vector $\mathbf{x}$:

$$\|\mathbf{x}\|^2 = \mathbf{x}^T\mathbf{x} \tag{4.7}$$

then L^2 norm can easily be calculated.

4.3 L1 Norm

However, in several Machine Learning applications, where it is necessary to differentiate between zero and non-zero elements, the L^1 norm is preferred as opposed to the L^2 norm (which grows very slowly near the origin).

$$\|\mathbf{x}\|_1 = \sum_i |x_i| \tag{4.8}$$

Every time an element of vector $\mathbf{x}$ is offset from the origin by ϵ, the L^1 norm increases by ϵ. The L^1 norm is also used as a substitute for the number of non-zero entries.

4.4 Special Norms

Two special norms which are used quite commonly in the context of machine learning and deep learning are:

- **Max norm**

$$\|\mathbf{x}\|_\infty = \max_i |x_i| \tag{4.9}$$

The Max norm of a vector gives the absolute value of the element having the largest value in the vector.

- **Frobenius Norm**

$$\|A\|_F = \sqrt{\sum_{i,j} A_{i,j}^2} \tag{4.10}$$

which is analogous to the L^2 norm of a vector.

4.5 Dot Product and Norms

The dot product of two vectors can be written in terms of norms. Specifically, using the L^2 norm as follows:

$$\mathbf{x}^T\mathbf{y} = \|\mathbf{x}\|_2\|\mathbf{y}\|_2 \cos\theta \tag{4.11}$$

where θ is the angle between $\mathbf{x}$ and $\mathbf{y}$.