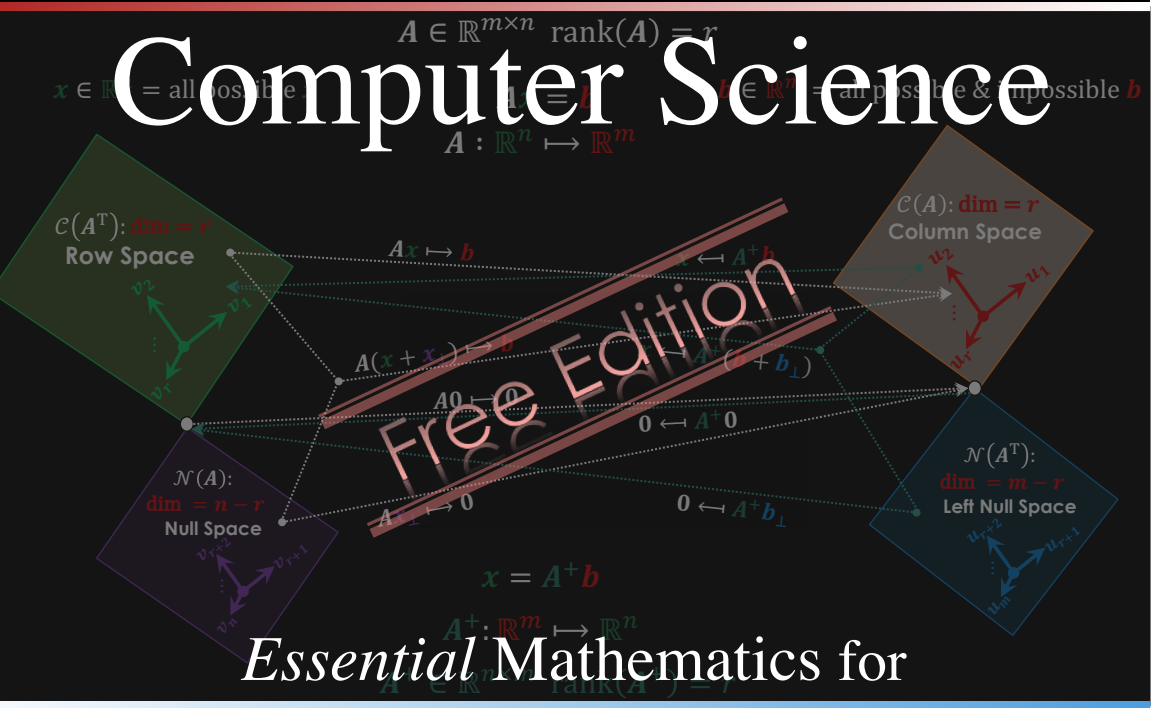
Linear Algebra for Computer Science



Essential Mathematics for Computer and Data Scientists

Manoj Thulasidas

Linear Algebra for Computer Science

Linear Algebra for Computer Science

Essential Mathematics for Computer Scientists

Manoj Thulasidas

ASIAN BOOKS
Singapore

ASIAN BOOKS

Manoj Thulasidas
Associate Professor of Computer Science (Education)
80 Stamford Road, School of Computing and Information Systems,
Singapore Management University, Singapore 178902

https://www.thulasidas.com
https://smu.sg/manoj
https://LA4CS.com

ISBN: 978-981-18-2045-8 (ebook)

Published in Singapore

"You can't learn too much linear algebra."

—Prof. Benedict Gross, Harvard

"Matrices act. They don't just sit there."

—Prof. Gilbert Strang, MIT

Contents

Part IV Advanced Topics

List of Figures

List of Tables

Acknowledgments

This book is the final metamorphosis of a collection of lecture notes, written specifically to help my students of *Linear Algebra for Computing Applications* at Singapore Management University. It would not have existed but for the opportunity to teach that course, and for the encouragement of my friends, colleagues, and students. I am grateful to them for their help and support, and especially to the ones who proofread the drafts and suggested edits.

M.T.

Preface

This book has its origin in my experience teaching Linear Algebra to Computer Science students at Singapore Management University. Traditionally, Linear Algebra is taught as a pure mathematics course, almost as an afterthought, not fully integrated with any other applied curriculum. That certainly was how it was taught to me. The course I was teaching, however, had a definite pedagogical objective of bringing out the applicability and usefulness of Linear Algebra in Computer Science, which is essentially applied mathematics. In the era of machine learning and artificial intelligence, Linear Algebra is the branch of mathematics that holds the most relevance to computing.

One question I got from one of my brightest students the first time I taught the course was why they were forced to learn this particular branch of mathematics. It was not a defiant or rebellious question, but one of pure curiosity. I did not have a ready answer then, but I think I have one now. When we embark on any profession, we have to start with the tools of the trade. For instance, if we want to be a musician, we have to learn the notes before we can perform. If we want to be a writer, we need to know the vocabulary and grammar of our chosen language first. Similarly, in order to be a computer scientist, we have to have the necessary mathematical skills. And Linear Algebra is arguably the most critical expertise needed, especially when it comes to dealing with large quantities of data efficiently.

Linear Algebra is a well-established field, and we can find several resources freely available on the Internet. In writing this book, I have made use of many of them. In fact, the whole process of writing can be thought of as an exercise in curating the right pieces of information at the right level, and standardizing them with consistent notations to form a coherent and fluid narrative.

Mathematics books often tend to be dry, unreadable collections of facts, theorems, proofs and problems. This type of discourse is understandable, given the nature of the subject that demands accuracy and completeness, often at the expense of readability. My objective was to write a book that would be read. For this reason, I set practicality as my goal, and even used, dare I say this, humor to keep my reader engaged.

I employed two more tricks to improve the readability. The first one is to restrict the field (over which our vectors and matrices are defined) to real numbers ($\mathbb{R}$) because of its relevance to computer science. The second trick is to pepper the text with "boxes," which are curious applications, background information, or other tidbits that are topic-adjacent to the subject matter under discussion.

From experience, I know that a book of this length takes about a year to write and another year to polish and publish. The first version of this book was done and dusted in about three months, almost ten times faster than normal. For this reason, it is continuously updated, corrected and improved upon over the last couple of years. How my students respond to the course on which the book is based will also inspire further revisions. These new versions and/or editions are made available periodically.

MANOJ THULASIDAS

Singapore

August 7, 2025

Introduction

I cannot teach anybody anything, I can only make them think.

—Socrates

This book is for an introductory course in Linear Algebra. It teaches the mathematical foundations of Linear Algebra to illustrate their relevance to computer science and applications. It also prepares the students for advanced numerical methods in computing, especially in machine learning and data analytics. Designed for a first course in Linear Algebra, this book will cover the basic concepts and techniques of Linear Algebra and provide an appreciation of the wide application of this discipline within the field of computer science.

The book will require development of some theoretical results, with proofs and consequences employing some level of mathematical rigor, algebraic manipulation, geometry and numerical algorithms. However, the main focus will be on the computational aspects and the applications of Linear Algebra.

Regardless of how application-oriented we would like to make it, Linear Algebra is a branch of pure mathematics. And in a textbook written for a first course introducing it, we will have a hard time staying away from the purely theoretical; however, Linear Algebra touches upon so many aspects of the technologies that enhance our modern life that we may discover interesting connections and applications even from the most basic of its concepts. This book will bring out such connections wherever possible, thereby attempting to be an intellectual treat to its readers.

I.1 Why Learn Linear Algebra?

From a purely mathematical perspective, Linear Algebra is a branch that takes us to *mathematical maturity*, whereby we start to appreciate the interconnections among its various branches, such as algebra (as in solving equations) and geometry (such as vector spaces), advanced algorithms and mathematical intuitions behind them and so on.

We have a few major branches of mathematics that are highly relevant to scientific computing, numerical methods and technology in general. The most important ones among them are Calculus (of the multivariable kind) and Linear Algebra. Roughly speaking, Calculus corresponds to the analog world–concerned with continuous variables, their evolution, effects on others etc. It plays an enormously important role in all branches of engineering and physical sciences. However, the world is digital now. And, the mathematics of digital technology, at a high level, is Linear Algebra.

For computer science specifically, we have one more relevant branch of mathematics, which is discrete math. This one is a lot of fun, full of puzzles and brain-teasers. It involves number theory, game theory and other fun stuff.

Relevant to all fields where data and number crunching are involved (ranging from social sciences to medicine to data science) is Statistics and Probability. Later on, we will see some of the statistical quantities (covariance and its principal components, for instance) and even certain quintessentially calculus operations (error minimization in linear regression, viewed as a projection operation) coming out of operations specific to Linear Algebra in an elegant fashion.

While we can list a number of impressive applications of Linear Algebra in computer science (such as the page rank algorithm that made Google what it is, and certain image and data compression algorithms), the real role of this branch of mathematics is similar to that of the alphabet or vocabulary or grammar in learning a language. If we want to be a writer, for instance, we have to be good at all these aspects of the language of our choice. Having these background skills alone is not enough; it will not make us a writer. But what is absolutely certain is that *without* these skills, we will *never* be a writer, not a good one at any rate.

Linear Algebra, in much the same way, is really the basic backdrop of several of the pivotal numerical algorithms in computer science.

I.2 Learning Objectives and Competencies

The learning objectives of the book include imparting the knowledge and the foundational concepts of Linear Algebra, as well as a set of skills and an appreciation for its application in computing. Upon the successful completion of a course based on this book, students should have a sound understanding the following:

- The concept of linearity as it applies to expressions, functions and transformations.
- The notion of vectors and matrices, their operations.
- Important characteristics of matrices, such as its four fundamental subspaces, rank, determinant, eigenvalues and eigenvectors, different factorizations, etc.
- How to use the characteristics of a matrix to solve a linear system of equations using algorithms such as Gaussian and Gauss-Jordan eliminations.
- Important concepts of vector spaces such as independence, basis, dimensions, orthogonality, Gram-Schmidt process for orthonormalization, etc.
- Properties of special categories of matrices such as symmetric, positive definite, etc.
- Eigenvalue, singular value and other decompositions, and their applications.

- Several algorithms in and using Linear Algebra, like linear regression, QR and power iteration for eigenanalysis, Cholesky decomposition etc.

The learning objectives listed above translate to the following learning outcomes or competencies in the students. Upon the successful completion of a course based on this book, students should be able to perform the following (both manually and in software tools such as `SageMath`):

1. Determine the existence and uniqueness of the solution of a linear system, and find its complete solution by choosing an effective method such as Gaussian elimination, factorization, diagonalization, etc.
2. Test for linear independence of vectors, orthogonality of vectors and vector spaces.
3. Compute the rank, determinant, inverse, Gram-Schmidt orthogonalization and different factorizations of a matrix.
4. Visualize the four fundamental subspaces of a matrix, and identify their relation to systems of linear equations, and find their dimension and basis.
5. Identify special properties of a matrix, such as symmetry, positive definiteness, etc., and use this information to facilitate the calculation of matrix characteristics.
6. Describe the use of mathematical techniques from linear algebra as applied to computer applications.
7. Compute eigenvalues and eigenvectors of a matrix, use them for diagonalizing, taking its powers, and applying them to solve advanced problems. [Optional Topic]
8. Perform diagonalization and the singular value decomposition of a matrix, and identify its principal components. [Optional Topic]

I.3 Organization

The book is organized in four parts, as listed below. For courses based on this book, the same topic flow is recommended, although

it may be necessary to shuffle some of the sub-topics for clarity and ease of understanding.

Part I: Numerical Computations

In the first part, we will go over the basics of Linear Algebra as it is usually taught in an undergraduate curriculum. Here, we will be working with vectors and matrices as represented by arrays of numbers, and their basic operations.

1. Functions, Equations and Linearity
2. Vectors, Matrices and Operations
3. Transposes and Determinants

Part II: Algebraic View

After covering the basic operations, we move on to the view of matrices and vectors as encoding linear equations, and ways of solving them.

4. Gaussian Elimination
5. Ranks and Inverses of Matrices

Part III: Geometric View

In the third part, we will look at the beautiful geometry that arises from vectors and matrices, from the perspective of the spaces they define, and their properties and operations.

6. Vector Spaces, Basis and Dimensions
7. Change of Basis, Orthogonality and Gram-Schmidt
8. Review and Recap
9. The Four Fundamental Spaces
10. Projection, Least Squares and Linear Regression

Part IV: Advanced Topics

In the last part, we will discuss the operations on matrices such as eigenvalue and singular value decompositions, their significance and applications.

11. Eigenvalue Decomposition and Diagonalization
12. Special Matrices, Similarity and Algorithms
13. Singular Value Decomposition

Part IV of the book, **Advanced Topics**, although significant as the basis of many algorithms, may be too difficult in an introductory undergraduate course in Linear Algebra. It may be revisited later on in the curriculum.

Even though the emphasis in each part will be as declared in its title, we will see that it is difficult to segment Linear Algebra into watertight compartments like Numeric, Algebraic, Geometric etc. We will necessarily see some overlap.

The book will also list some of the tools and resources to be used, especially to illustrate the computing applications. It will also show that most of the numeric calculations that used to be performed by hand are now carried out using mathematical programs such as `Matlab` or **`SageMath`**, the latter being our tool of choice.

I.3.1 Our Focus

The value we can find in learning Linear Algebra is not in developing arithmetic dexterity in performing numeric computations, for we can find all such numerical computations neatly implemented in tools like **`SageMath`**. Even some symbolic manipulations are found in **`SageMath`** or other tools such as Mathematica. What we focus on, therefore, are the aspects of Linear Algebra that will improve our intuitive insights and conceptual understanding. Our hope is that these aspects will help us be better computer and data scientists. For this reason, the exercises that follow every chapter after a chapter summary are very different from what we usually find in books in Linear Algebra; they are mainly objective type questions testing our grasp of the theoretical, conceptual and intuitive aspects.

Part I

Numerical Computations

1

Functions, Equations and Linearity

It is a monstrous thing to force a child to learn Latin or Greek or mathematics on the ground that they are an indispensable gymnastic for the mental powers. It would be monstrous even if it were true.

—George Bernard Shaw

1.1 Linearity

Formally, Linear Algebra deals with objects (mathematical entities) that transform in a *linear* fashion. After all, it has the word "Linear" in its name. Let's define linearity, along with what "transform" means. In order to do that, however, we have to take a step back and look at some of the foundations.

1.1.1 Expressions and Functions

When we combine one or more mathematical *variables* in a valid way (using operations like addition, multiplication, exponentiation *etc.*, or using *functions* like sine, logarithm etc.), we get a mathematical

expression. Here are some examples of expressions: $3x, \sin x, x + y, \boldsymbol{Ax}, \ln x$.

A *variable* is a symbol (like a, x, $\boldsymbol{A}$ etc.) that is a placeholder or a container for a *value*. Note that the value can be a single number, or a complicated one like a vector or a matrix, although we have not defined them yet. For now, however, we are thinking of the variables as placeholders of single, real values. For a variable x, we will state that it is real using this mathematical jargon: $x \in \mathbb{R}$ where $\mathbb{R}$ represents the set of all real numbers and $\in$ says, "is a member of."

When we have an *expression* (which is a combination of variables), we can also think of it as a relationship between its value and the set of variables it contains, which is what we mean by a *function*. For example, $4x + y$ is an expression; $f(x, y) = 4x + y$ is a function. We usually write a function of a single variable as $f(x) =$ an expression in x. Some examples of single-variable functions are: $f(x) = 3x, f(x) = e^{-x^2}$. We require the functions to be *single-valued*.

In other words, we can think of an expression as *defining* or *specifying* a *function*, with the additional constraint that one set of inputs (corresponding to the variables in the expression) gives one unique output (which is the value of the expression). Examples: $\sqrt{x}$ is an expression, but $f(x) = \sqrt{x}$ is *not* a function by our definition because it gives two values for every x. However, $f(x) = |\sqrt{x}\,|$ (where $|y|$ stands for the absolute value of y) is a function.

We can think of a function as a *transformation*: A mathematical object that transforms the inputs to the outputs. Although we are thinking of the inputs (x, y etc.) and the output ($f(x, y)$) as numeric values, they can be other mathematical entities, like vectors and matrices (to be defined). Again, for now, let's limit ourselves to real values of the inputs and outputs for our discussion now.

Linearity

Definition: We call a transformation (or a function) of a single, real value ($x \in \mathbb{R}$) *linear* if it satisfies the following two conditions:

- **Homogeneity**: When the value of x is multiplied by a real number, the value of the function also gets multiplied by the same number.

$$f(sx) = sf(x) \qquad \forall\, s, x \in \mathbb{R}$$

This property of multiplicative scaling behavior is known as *homogeneity*.

- **Additivity**: The output of the function for the sum of two inputs is the sum of the corresponding outputs.

$$f(x + x') = f(x) + f(x') \qquad \forall\, x, x' \in \mathbb{R}$$

Here are some examples of linear functions: $f(x) = 3x, f(x) = 0$. Some functions that are not linear would be: $f(x) = x^2, f(x) = 5$. Note that $f(x) = mx + c$ is *not* linear if $c \neq 0$ because it does not satisfy the linearity conditions listed above. This fact may look weird to us because we know that $y = mx + c$ is the equation to a line. We will take a closer look at it very soon.

1.1.2 Equations and Equality

In order to understand why $f(x) = mx + c$ is *not* linear, we have to understand what an *equation* is, and, even more fundamentally, what equality (or the equal sign, $=$) means. When we write $f(x) = mx+c$, we are *defining* a function. The equal sign ($=$) in $f(x) = mx + c$ is one of assignment: We are *assigning* the symbol f to the function of one variable $mx + c$, which is not a difficult concept for computer scientists with exposure to programming languages[1]. This function, indeed, is not a linear function according to our definition of linearity.

When we write $y = mx + c$, on the other hand, we do not mean an assignment or a definition. It is a statement of truth, or a recipé for computing any y given a value for x. It is an *equation*. In fact, it is an equation in *two* variables, $x, y \in \mathbb{R}$. In other words, the equal sign ($=$) in $y = mx + c$ is an assertion of truth or a condition. We call such assertions of truth equations. Our preferred form of equations is: expression $=$ constant, or function (which is a placeholder for an expression) $=$ constant. Therefore, we will write $y = mx + c$ as $-mx + y = c$.

To summarize, equations in one variable have the form $f(x) = b$ (where $x, b \in \mathbb{R}$, in our context). If the function $f(x)$ (which is a

[1] Programming languages such as Python are procedural, where we specify assignments and operations (or steps) to be performed on them. We have another class of programming languages called functional, where we list statements of truth and specify mathematical operations on the variables. Haskell is one of them.

proxy for some, hitherto unspecified expression) is linear, we consider the equation $f(x) = b$ to be linear as well. In fact, in the case of expressions (or functions) of one variable, we know a bit more than that because we have defined what we mean by linearity: The only possible form $f(x)$ can take so that it is linear is ax. So the most general linear equation of one variable is simply $ax = b$.

How do we define linearity for two (or more) variables? In particular, is $y = mx + c$ (or, equivalently, $-mx + y = c$) a linear *equation*? We know that $mx + c$ is *not* a linear *expression*, and $f(x) = mx + c$ is *not* a linear *function*. But, $-mx + y$ is a function of *two* variables, $f(x, y)$. In order to determine whether it is linear, the linearity conditions listed earlier (homogeneity and additivity) need to be generalized for multiple variables.

1.1.3 Linearity of Multivariable Functions

Let's generalize the homogeneity property to two variables. For the one variable case, it was $f(sx) = sf(x) \forall s, x \in \mathbb{R}$. For a two-variable function $f(x, y)$, let's say that homogeneity is satisfied when *both* variables are multiplied by the *same* scaling factor (or scalar).

$$f(sx, sy) = sf(x, y) \quad \forall s, x, y \in \mathbb{R}$$

For one variable, the additivity property was $f(x + x') = f(x) + f(x') \forall x, x' \in \mathbb{R}$. Let's generalize it as follows:

$$f(x + x', y + y') = f(x, y) + f(x', y') \quad \forall x, y, x', y' \in \mathbb{R}$$

If we define a new mathematical entity, consisting of two numbers (elements) in a specified order, for which scaling is defined as scaling each element, and addition is defined as the addition of the corresponding elements, we could reuse our original (one-variable) linearity conditions for the two-variable case as well, or indeed for the n-variable case. This mathematical entity (a group of numbers in a specified order) is a **vector**. Let's define it as a column of numbers, with scaling and addition operations.

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} \quad s\boldsymbol{x} = \begin{bmatrix} sx \\ sy \end{bmatrix} \quad \boldsymbol{x'} = \begin{bmatrix} x' \\ y' \end{bmatrix} \quad \boldsymbol{x} + \boldsymbol{x'} = \begin{bmatrix} x + x' \\ y + y' \end{bmatrix}$$

Our vector $\boldsymbol{x}$ has two real numbers in it, and we write $\boldsymbol{x} \in \mathbb{R}^2$. We can easily generalize the vectors such as $\boldsymbol{x}$ to n dimensions. All we have

to do is write its n elements (say, $x_1, x_2, \cdots, x_n$) in a column. We would then write $\boldsymbol{x} \in \mathbb{R}^n$. With these definitions and generalization, we can restate the two linearity conditions as follows:

- **Homogeneity**: $f(s\boldsymbol{x}) = sf(\boldsymbol{x}) \quad \forall\, s \in \mathbb{R}, \boldsymbol{x} \in \mathbb{R}^n$
- **Additivity**: $f(\boldsymbol{x} + \boldsymbol{x'}) = f(\boldsymbol{x}) + f(\boldsymbol{x'}) \quad \forall\, \boldsymbol{x}, \boldsymbol{x'} \in \mathbb{R}^n$

For the sake of completeness, let's verify whether $f(x, y) = -mx + y$ is indeed linear. If we scale x and y by s, the value of the function gets scaled by the same factor:

$$f(sx, sy) = -msx + sy = s(-mx + y) = sf(x, y)$$

Therefore, the homogeneity condition is satisfied. For the additivity condition:

$$\begin{aligned} f(x + x', y + y') &= -m(x + x') + y + y' \\ &= (-mx + y) + (-mx' + y') \\ &= f(x, y) + f(x', y') \end{aligned}$$

Therefore, the second, additivity, condition also is satisfied, and $f(x, y) = -mx + y$ is indeed linear. And the *equation* $-mx + y = c$ is linear as well. To be clear, earlier we said $mx + c$, as an expression in (or $f(x) = mx + c$ as a function of) one variable is *not* linear, which is not in contradiction with the statement that $f(x, y) = -mx + y$ *is* linear as a function of (or $-mx + y$ as an expression in) *two variables*. And $-mx + y = c$ is a linear equation in two variables.

1.1.4 Vectors and Matrices

We stated earlier that the most general linear equation of one variable was $ax = b$. We start with the linear equation $-mx + y = c$. What is the most general linear equation (or system of linear equations) in two or n dimensions?

Let's start by defining a multiplication operation between two vectors: The product of two vectors is the sum of the products of the corresponding elements of each of them. With this definition, and a notational trick[2] of writing the first of the two vectors horizontally,

[2]The reason for this trick is to have a generalized definition of the product of two matrices, of which this vector multiplication will become a special case. We will go through matrix multiplication in much more detail in the next chapter.

we write:

$$\begin{bmatrix} a_1 & a_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = a_1 x + a_2 y \implies \begin{bmatrix} -m & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = -mx + y$$

Our linear equation $-mx+y = c$ and a more general $a_1x+a_2y = b$ then becomes:

$$\begin{bmatrix} -m & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = c \qquad \begin{bmatrix} a_1 & a_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = b$$

If we had one more equation, $a_{21}x + a_{22}y = b_2$, we could add another row in the compact notation above. In fact, the notation is capable of handling as many equations as we like. For instance, if we had three equations:

1. $a_{11}x + a_{12}y = b_1$
2. $a_{21}x + a_{22}y = b_2$
3. $a_{31}x + a_{32}y = b_3$

We could write the system of three equations, more compactly,

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad \text{or} \quad \boldsymbol{Ax} = \boldsymbol{b} \tag{1.1}$$

Here, the table of numbers we called $\boldsymbol{A}$ is a **matrix**. What we have written down as $\boldsymbol{Ax} = \boldsymbol{b}$ is a system of three linear equations (in 2-dimensions, but readily extended to n dimensions).

1.1.5 Linear Transformations

Now that we got a glimpse of vectors and matrices, and defined what linearity means in the context of Linear Algebra, let's quickly summarize the discussion with linear transformations. A function $f(x)$ can be thought of as a transformation, or a mapping[3]. When we write $y = f(x)$, what we are saying is that we can transform for every $x \in \mathbb{R}$ using $f(x)$ and get a value of $y \in \mathbb{R}$ to which it is mapped. f is a mapping from $\mathbb{R}$ to $\mathbb{R}$. Of all possible functions $f(x)$, the simplest one is a linear transformation, $f(x) = ax$. In fact, it is the only form

[3]For our purposes, there is very little difference between functions, transformations and mapping.

of linear transformation in one dimension—the only one that respects linearity as we defined it.

$$f(x) = ax, \quad x \in \mathbb{R}. \quad f : \mathbb{R} \mapsto \mathbb{R}$$

And the most general form of equation (in the form expression or function = constant) in one dimension would be:

$$ax = b, \quad a, b \in \mathbb{R}$$

What is the simplest possible transformation from a multi-dimensional space $\mathbb{R}^n$ to a *different* multi-dimensional space $\mathbb{R}^m$? This turns out to be the linear transformation encoded in a matrix.

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \quad \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{y} \in \mathbb{R}^m. \quad \boldsymbol{A} : \mathbb{R}^n \mapsto \mathbb{R}^m$$

We can prove (as indeed we will, later on) that every linear transformation $\mathbb{R}^n \mapsto \mathbb{R}^m$ has a *unique* matrix $\boldsymbol{A}$ associated with it. The matrix $\boldsymbol{A}$, therefore, represents a mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$, taking the vector $\boldsymbol{x}$ and giving us the vector $\boldsymbol{y}$. When we define an equation (as *expression* = *constant*), we get the deceptively simple equation:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

As we saw earlier, the equation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ also represents a system of linear equations. The properties of such systems as encoded in the matrix $\boldsymbol{A}$, the conditions under which the system can be solved, and the geometry and intuitions behind it will become the rest of this book.

1.2 The Big Picture

For our purposes in computer science, the most appropriate set of numbers to build vectors and matrices is that of real numbers, arranged in columns or rows or tables. A scalar is a member of the set of real numbers, $s \in \mathbb{R}$. It can also be thought of as a 1×1 matrix. A vector is a $n \times 1$ matrix, $\boldsymbol{x} \in \mathbb{R}^n$. Vectors of fewer than four components live in spaces[4] that we can understand and visualize: $\mathbb{R}^2$ is a plane and

[4] As we get more sophisticated later on, we will qualify this statement and draw a distinction between coordinate spaces where we live and vector spaces where vectors live.

$\mathbb{R}^3$ is the three-dimensional space we live in. We can extrapolate and understand, if not visualize, dimensions higher than three, namely $\mathbb{R}^n, n > 3$, as well.

Matrices (such as data sets), on the other hand, live in spaces that are harder to visualize: $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. However, as we will see, the properties of these spaces are not unlike the spaces of vectors.

Although we will deal with vectors and matrices as arrays (columns or tables) of numbers, it is important to keep in mind that they are, in fact, abstract entities defined only by the operations (such as scaling, addition, multiplication) specified on them.

In computer science, the representation of vectors and matrices as arrays of numbers is the one that makes most sense because our data sets tend to comprise numbers. However, if we can find other entities on which we can specify the requisite set of operations, we are allowed to treat them as vectors as well. This generalization, though not critical in computer science, plays an important role in the applications of Linear Algebra in other fields, most notably in physics.

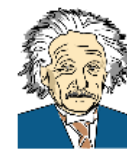
The whole point of Linear Algebra is that we do not have to think of the representation; we can work with the properties of the underlying entities. So, as the book progresses, we will stop worrying about the individual elements of matrices or vectors. Towards the end, in Part IV, we will go over some advanced applications of Linear Algebra, where, our intuition will become completely abstract and independent of any representation. We will then be in a position to apply such intuitions to come up with efficient algorithms and computational techniques.

2

Vectors, Matrices and Their Operations

So far as the theories of mathematics are about reality, they are not certain; so far as they are certain, they are not about reality.

—Albert Einstein



Vectors and matrices, along with the operations defined on them, form the basis of Linear Algebra as we will learn in this book. In the previous chapter, we got a glimpse of them. Now, let us look at them in more detail and in a formal fashion.

2.1 Vectors

For our purposes in Computer Science, vectors are an *ordered* list of *numbers*. The numbers are of the same type. In our case, typically they are real numbers because data tends to be full of real numbers. We would formally call the vectors by the type of the numbers. In other words, our vectors are vectors over the *field* of real numbers.

The last statement calls for a digression to describe what a *field* is, which is in the box below. If we have a m dimensional vector $\boldsymbol{x}$, we indicate its field by saying, $\boldsymbol{x} \in \mathbb{R}^m$. In our tool, **SageMath**,

Groups, Rings and Fields

***Group*:** A *Group* is a set of mathematical entities with one binary operation, the identity of the operation and an inverse for every element with respect to that operation. The operation needs to be commutative.

The set of all integers with addition defined will be an example. Zero is the identity, for every number, its negative is the additive inverse.

***Ring*:** A *Ring* is a set of mathematical entities with two binary operations (generalized versions of the arithmetic operations of addition and multiplication) with the key properties:

- Addition is associative and commutative
- There is an additive identity, a zero
- Every element has an additive inverse
- Multiplication is associative
- Not necessary that multiplication be commutative
- Multiplication distributes over addition
- It need not have an multiplicative inverse

Classic examples of *Rings* are:

- Integers
- Integers modulo some Natural number greater than one

***Field*:** A *Field* has (in addition to what *Rings* have):

- Multiplication is commutative
- Every nonzero element has a multiplicative inverse

Classic examples of *Fields* are:

- Rational numbers
- Real numbers
- Complex numbers
- Integers modulo a Prime number

A *Field* is a *Ring* with extra properties. And a *Ring* is a *Group* with extra properties.

we will see that vectors (and matrices) can be over the *ring* (see the box titled **Groups, Rings and Fields**) of integers ($\mathbb{Z}$, called `ZZ` in **`SageMath`**), or the *field* of rationals ($\mathbb{Q}$, `QQ`), real ($\mathbb{R}$, `RR`) or complex ($\mathbb{C}$, `CC`) numbers[1].

In order to build a realistic example of a vector that we might come across in computer science, let's think of a data set where we have multiple observations with three quantities:

[1] See `https://www.mathsisfun.com/sets/number-types.html` for common sets of numbers in mathematics.

1. w: Weight in kg

2. h: Height in cm

3. a: Age in years

A sample data point could be a list of three values: $(w, h, a) = (76, 175, 51)$. We can think of this data point as a vector (over the field[2] of real numbers). Our vectors are always Column Vectors, or numbers arranged in a column. Therefore, we write:

$$\boldsymbol{x} = \begin{bmatrix} 76 \\ 175 \\ 51 \end{bmatrix} \in \mathbb{R}^3 \tag{2.1}$$

A column vector is, in fact, a matrix of size $m \times 1$. Note that although the numbers of the vector (also known as its *elements*, *entries* or *components*) belong to the same field mathematically, they do not have to signify the same physical quantity[3]. They do not need to have the same physical significance. However, the order of the elements in a vector is significant.

2.2 Vector Operations

2.2.1 Scalar Multiplication

We define the *scalar multiplication* of a vector with a number as the follows: The product of the multiplication is a new vector with each of its elements multiplied by the number. In other words, the number (called *scalar*) *scales* the vector.

Scalar Multiplication

Definition: For any vector $\boldsymbol{x}$ and any scalar s, the result of scalar

[2]In computer science, the difference between using the rational *ring* instead of real *field* is too subtle to worry about. It has to do with the definition of the *norm* (or the size) of a vector.
[3]In fact, the whole machinery of Linear Algebra is a big syntactical engine, yielding us important insights into the underlying structure of the numbers under study. But it has little semantic content or correspondence to the physical world, other than the insights themselves.

multiplication ($s\boldsymbol{x}$) is defined as follows:

$$\forall\ \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n \text{ and } s \in \mathbb{R}, \quad s\boldsymbol{x} \stackrel{\text{def}}{=} \begin{bmatrix} sx_1 \\ sx_2 \\ \vdots \\ sx_n \end{bmatrix} \in \mathbb{R}^n \tag{2.2}$$

We say that the set of vectors is *closed* under scalar multiplication, which is a fancy way of saying that if we scale a vector, we get another vector. Note that for a vector over a certain field (like $\mathbb{R}$), the scalar also belongs to the same field. In fact, it can be any number in that field, including zero.

The fact that the scaling factor can be zero, and that the set of vectors is closed under scalar multiplication has something to say about the zero vector (denoted by $\mathbf{0}$, which is a vector whose elements are all zeros). It has to be a vector too, because by scaling any vector with the scalar zero, we get an entity with all zeros, and because the set of vectors is closed under scalar multiplication, this entity has to be a vector. The zero vector is special, just like zero is a special number.

Scalar multiplication is commutative: $s\boldsymbol{x} = \boldsymbol{x}s$. It is not a precondition, but a consequence of the definition.

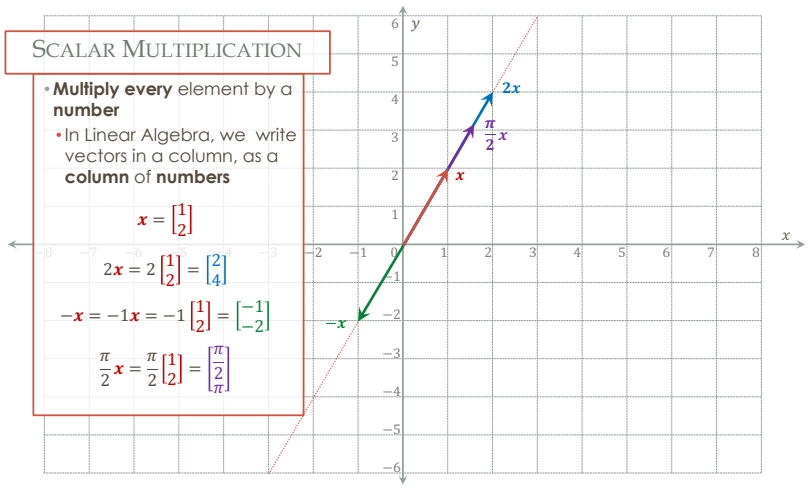


Fig. 2.1 Example of scalar multiplication of a vector $\boldsymbol{x} \in \mathbb{R}^2$. Note that all the scaled versions lie on the same line defined by the original vector.

Figure 2.1 shows an example of scalar multiplication. The original vector $\boldsymbol{x} \in \mathbb{R}^2$ has elements 1 and 2, which we plot with one unit

along the x axis, and two units along the y axis. The vector then is an arrow from the origin to the point $(1, 2)$, shown in red. Note how the scaled versions (in blue, green and purple) of the vector are all collinear with the original one. Note also that the line defined by the original vector and its scaled versions all go through the origin.

2.2.2 Vector Addition

We define the addition of two vectors such that the result of the addition is another vector whose elements are the sum of the corresponding elements in the two vectors. Note that the sum of two vectors also is vector. In other words, the set of vectors is closed under addition as well.

Vector Addition

Definition: For any two vector $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, the result of addition $(\boldsymbol{x_1} + \boldsymbol{x_2})$ is defined as follows:

$$\begin{aligned} \forall\, \boldsymbol{x_1} = \begin{bmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{bmatrix} \text{ and } \boldsymbol{x_2} = \begin{bmatrix} x_{12} \\ x_{22} \\ \vdots \\ x_{n2} \end{bmatrix} \in \mathbb{R}^n, \\ \boldsymbol{x_1} + \boldsymbol{x_2} \stackrel{\text{def}}{=} \begin{bmatrix} x_{11} + x_{12} \\ x_{21} + x_{22} \\ \vdots \\ x_{n1} + x_{n2} \end{bmatrix} \in \mathbb{R}^n \end{aligned} \tag{2.3}$$

Vector addition also is commutative ($\boldsymbol{x_1} + \boldsymbol{x_2} = \boldsymbol{x_2} + \boldsymbol{x_1}$), as a consequence of its definition. Moreover, we can only add vectors of the same number of elements (which is called the dimension of the vector). Although it is not critical to our use of Linear Algebra, vectors over different fields also should not be added. We will not see the latter restriction because our vectors are all members of $\mathbb{R}^n$. Even if we come across vectors over other fields, we will not be impacted because we have the hierarchy integers ($\mathbb{Z}$) $\subseteq$ rationals ($\mathbb{Q}$) $\subseteq$ real ($\mathbb{R}$) $\subseteq$ complex ($\mathbb{C}$) numbers. In case we happen to add a vector over the field of integers to another one over complex numbers, we will get a sum vector over the field of complex numbers; we may not realize that we are committing a Linear Algebra felony.

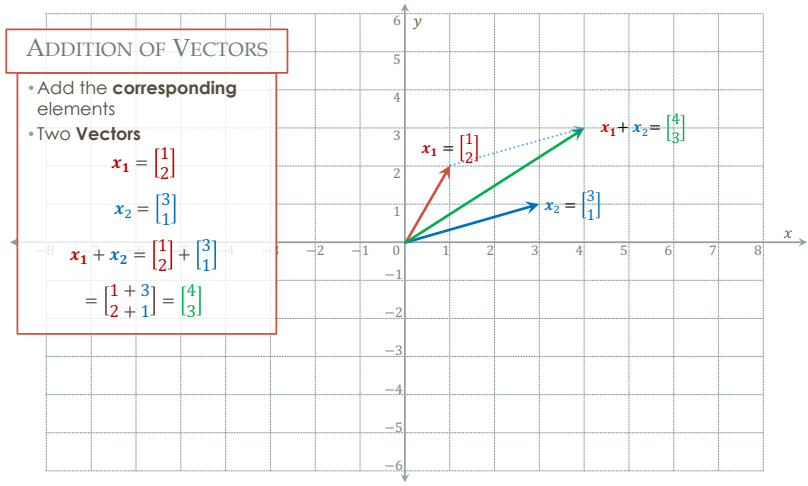


Fig. 2.2 Adding two vectors $\boldsymbol{x_1}, \boldsymbol{x_2} \in \mathbb{R}^2$. We add the element of $\boldsymbol{x_2}$ to the corresponding elements of $\boldsymbol{x_1}$, which we can think of as moving $\boldsymbol{x_2}$ (in blue) to the tip of $\boldsymbol{x_1}$ (in red), so that we get the arrow drawn with dashed blue line. The sum of the two vectors is in green, drawn from the origin to the tip of the dashed arrow.

Figure 2.2 shows the addition of two vectors $\boldsymbol{x_1}, \boldsymbol{x_2} \in \mathbb{R}^2$. We can think of the addition operation as transporting $\boldsymbol{x_2}$ (in blue) to the tip of $\boldsymbol{x_1}$ (in red), so that we get the arrow drawn with dashed blue line, parallel to the original $\boldsymbol{x_2}$, and with the same size. The sum of the two vectors is in green, drawn from the origin to the tip of the dashed arrow. We could, of course, $\boldsymbol{x_1}$ (the red vector) to the tip of $\boldsymbol{x_2}$ (blue) and perform the addition that way as well. In other words, we could complete the parallelogram with $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ as adjacent sides, and think of the diagonal (from the origin) as the sum[4].

Now that we have defined addition and scaling, we can ask the question whether each vector has an *additive inverse*. In other words, for each vector $\boldsymbol{x} \in \mathbb{R}^n$, can we find another vector $\boldsymbol{x'}$ such that $\boldsymbol{x} + \boldsymbol{x'} = \mathbf{0}$. Note that $\mathbf{0}$ is not the number 0, but a vector $\mathbf{0} \in \mathbb{R}^n$. We can see that if we scale $\boldsymbol{x}$ by -1 so that $\boldsymbol{x'} = -1 \times \boldsymbol{x}$, by our definition of the addition of vectors, we have $\boldsymbol{x} + \boldsymbol{x'} = \mathbf{0}$. Therefore, every vector in $\mathbb{R}^n$ has an additive inverse.

[4]In Linear Algebra as taught in this book, our vectors are always drawn from the origin, as a general rule. This description of the addition of vectors is the only exception to the rule, when we think of a vector transported to the tip of another one.

2.2.3 Linear Combinations

Once we have scalar multiplication and addition, we can perform them both on our vectors, and be sure that what we get are valid vectors because of the closure property of the set of vectors under the two operations. If we have two vectors in $\mathbb{R}^n$, $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, and any two scalars, $s_1, s_2 \in \mathbb{R}$, we can construct a new vector $\boldsymbol{y} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2}$. We know that $\boldsymbol{y} \in \mathbb{R}^n$ is a valid vector.

For example, if we look at the two vectors in Figure 2.2, taking $s_1 = 2$ and $s_2 = -1$, we can get

$$\boldsymbol{y} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2} = 2\begin{bmatrix}1\\2\end{bmatrix} + (-1)\begin{bmatrix}3\\1\end{bmatrix} = \begin{bmatrix}2\\4\end{bmatrix} + \begin{bmatrix}-3\\-1\end{bmatrix} = \begin{bmatrix}-1\\3\end{bmatrix}$$

By taking different values for s_1 and s_2, we can get all sorts of $\boldsymbol{y}$ vectors. This notion of linear combinations of vectors is foundational to Linear Algebra, and we will have much more to say about it later on.

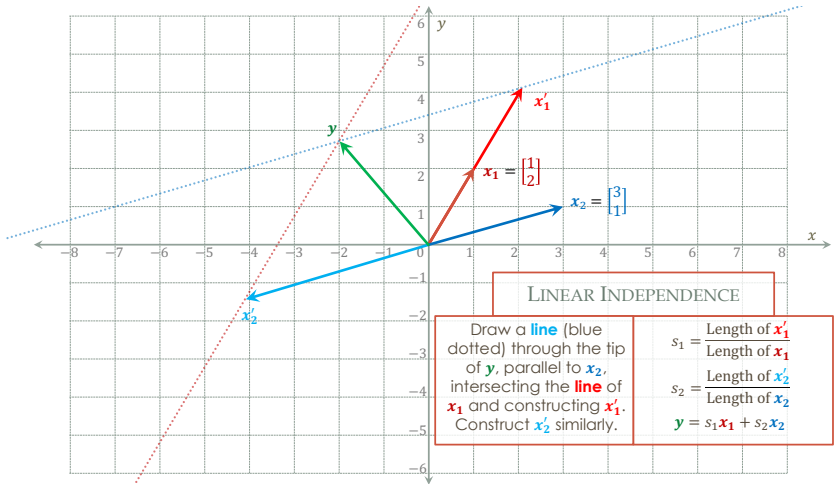


Fig. 2.3 Given *any* green vector $\boldsymbol{y}$ and the two vectors $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, how to find the scaling factors s_1 and s_2 in $\boldsymbol{y} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2}$? This geometric construction shows that any vector can be written as a linear combination of $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$.

Is it possible to have two sets of s_1 and s_2 for the same $\boldsymbol{y}$? The answer is no, and we can probably already see why. Suppose we want to find the scaling factors for the green vector in Figure 2.3 so that it can be written as $s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2}$. Here is one way to think about it: Draw a line through the tip of the green vector, parallel to the blue

vector $\boldsymbol{x_2}$. Let it intersect the line of the red vector, giving us the light red vector $\boldsymbol{x_1'}$. The (signed) length of this vector $\boldsymbol{x_1'}$ divided by the length of $\boldsymbol{x_1}$ will be s_1. The sign is positive if $\boldsymbol{x_1}$ and $\boldsymbol{x_1'}$ are in the same direction, and negative otherwise. By a similar construction, we can get s_2 as well. From the definition of the addition of two vectors (as the diagonal from $\mathbf{0}$ of the parallelogram of which the two vectors are sides), we can see that $\boldsymbol{y} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2}$, as shown in Figure 2.3. Since two lines (that are not parallel to each other) can intersect only in one point (in $\mathbb{R}^2$), the lengths and the scaling factors are unique.

The second question is whether we can get the zero vector

$$\boldsymbol{y} = \mathbf{0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2}$$

without having $s_1 = s_2 = 0$ (when $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are what we will call linearly independent in a minute). The answer is again no, as a corollary to the geometric "proof" for the uniqueness of the scaling factors. But we will formally learn the real reasons (*à la* Linear Algebra) in a later chapter dedicated to vector spaces and the associated goodness.

2.3 Linear Independence of Vectors

In Figure 2.3, we saw that we can get any general vector $\boldsymbol{y}$ as a linear combination of $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$. In other words, we can always find s_1 and s_2 such that $\boldsymbol{y} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2}$. Later on, we will say that $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ span $\mathbb{R}^2$, which is another way of saying that all vectors in $\mathbb{R}^2$ can be written as a unique linear combination of $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$.

The fact that $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ span $\mathbb{R}^2$ brings us to another pivotal concept in Linear Algebra: Linear Independence, which we will get back to, in much more detail in Chapter 6. $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are indeed two linearly independent vectors in $\mathbb{R}^2$.

A set of vectors are linearly independent of each other if none of them can be expressed as a linear combination of the rest. For $\mathbb{R}^2$ and two vectors $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, it means $\boldsymbol{x_1}$ is not a scalar multiple of $\boldsymbol{x_2}$. Another equivalent statement is that $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are linearly independent if the only s_1 and s_2 we can find such that $\mathbf{0} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2}$ is $s_1 = 0$ and $s_2 = 0$.

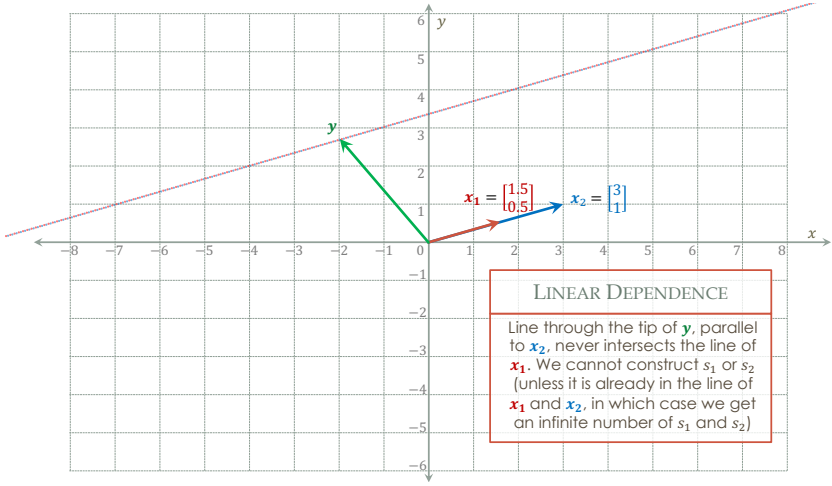


Fig. 2.4 Two linearly dependent vectors $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, which cannot form a linear combination such that the green $\boldsymbol{y} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2}$.

Figure 2.3 shows two $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ that are linearly independent. In Figure 2.4, we have another pair, $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, that are linearly *dependent*. When we try to construct $\boldsymbol{y} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2}$ out of it, we fail, unless $\boldsymbol{y}$ happens to be in the same line as $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$. Even in that case, we do not get a unique pair of s_1 and s_2, but an infinite number of them.

Note that in Figure 2.4, $\boldsymbol{x_2} = 2\boldsymbol{x_1}$, or $2\boldsymbol{x_1} - \boldsymbol{x_2} = \boldsymbol{0}$: The zero vector can be written as a linear combination of $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$. Thus, our test for linear independence fails for these two vectors, indicating that they are, indeed, linearly *dependent*.

2.3.1 Vector Dot Product

Given some scalars or vectors, now we know how to multiply a vector by a scalar and add two vectors. The operations between two scalars are not interesting, and a scalar cannot add to a (nontrivial) vector. So we are left with only one operation left to define, multiplication of a vector by another vector. Let's define a dot product between two vectors.

Dot Product

Definition: For any two vector $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, the dot product $(\boldsymbol{x_1} \cdot \boldsymbol{x_2})$

is defined as follows:

$$\begin{aligned} \forall\, \boldsymbol{x_1} = \begin{bmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{bmatrix} \text{ and } \boldsymbol{x_2} = \begin{bmatrix} x_{12} \\ x_{22} \\ \vdots \\ x_{n2} \end{bmatrix} \in \mathbb{R}^n, \\ \boldsymbol{x_1} \cdot \boldsymbol{x_2} \stackrel{\text{def}}{=} x_{11}x_{12} + x_{21}x_{22} + \cdots + x_{n1}x_{n2} \\ = \sum_1^n x_{i1}x_{i2} \in \mathbb{R} \end{aligned} \tag{2.4}$$

In other words, we compute the dot product between two vectors by multiplying their corresponding elements and summing up the products. Clearly, if the numbers of elements are different for the two vectors, the dot product is not defined. We cannot multiply vectors of different dimensions.

The dot product of two vectors is also known as their *scalar product* because the result of the operation is a scalar. This is not to be confused with scalar multiplication, which is about scaling a vector using a scalar. Another commonly used name for the dot product is *inner product*.

Norm of a Vector

Definition: For a vector $\boldsymbol{x}$, its norm, $\|\boldsymbol{x}\|$, is defined as follows:

$$\begin{aligned} \forall\, \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n,\ \|\boldsymbol{x}\| \stackrel{\text{def}}{=} \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} \\ = \sqrt{\sum_1^n x_i^2} \quad \in \mathbb{R} \end{aligned} \tag{2.5}$$

By the definition of the dot product, we can see that $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x} \cdot \boldsymbol{x}}$. Related to the dot product, the norm of a vector is a measure of its size. Also note that if we scale a vector by a factor s, the norm also scales by the same factor: $\|s\boldsymbol{x}\| = s\|\boldsymbol{x}\|$. If we divide a vector by its norm (which means we perform scalar multiplication by the reciprocal of the norm), the resulting vector has unit length, or is *normalized*. We

Cosine Similarity

In text analytics, in what they call Vector-Space Model, documents are represented as vectors. We would take the *terms* in all documents and call each a direction in some space. A document then would be a vector having components equal to the frequency of each term. Before creating such *term-frequency* vectors, we may want to clean up the documents by normalizing different forms of words (by *stemming* or *lemmatization*) removing common words like articles (the, a, *etc.*) and prepositions (in, on, *etc.*), which are considered *stop words*. We may also want to remove or assign lower weight to words that are common to all documents, using the so-called *inverse document frequency* (IDF) instead of raw term frequency (TF). If we treat the chapters in this book as documents, words like linear, vector, matrix, *etc.* may be of little distinguishing value. Consequently, they should get lower weights.

Once such document vectors are created (either using TF or TF-IDF), one common task is to quantify similarity, for instance, for plagiarism detection or document retrieval (searching). How would we do it? We could use the norm of the difference vector, but given that the documents are likely to be of different length, and document length is not a metric by which we want to quantify similarity, we may need another measure. The cosine of the angle between the document vectors is a good metric, and it is called the Cosine Similarity, computed exactly as we described in this section.

$$\cos\theta = \frac{\sum_{i=1}^{m} x_{i1}x_{i2}}{\|\boldsymbol{x_1}\|\|\boldsymbol{x_2}\|}$$

How accurate would the cosine similarity measure be? It turns out that it would be very good. If, for instance, we compare one chapter against another one in this book as opposed to one from another book on Linear Algebra, the former is likely to have a higher cosine similarity. Why? Because we tend to use slightly flowery (albeit totally appropriate) language because we believe it makes for a nuanced treatment of the intricacies of this elegant branch of mathematics. How many other Linear Algebra textbooks are likely to contain words like albeit, flowery, nuance, intricacy *etc.*?

use the notation $\hat{x}$ to indicate unit vectors.

$$\hat{\boldsymbol{x}} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$$

The norm we defined above is the Euclidean Norm. Other norms are possible, defined as below for various values of p:

$$\|\boldsymbol{x}\|_p \stackrel{\text{def}}{=} \sqrt[p]{\sum_{i=1}^{n} |x_i|^p} \quad \in \mathbb{R} \tag{2.6}$$

This so-called p-norm becomes the Euclidean norm we defined above for $p = 2$. For $p = 1$, it becomes what is known as the Manhattan (or Taxicab) norm. As $p \to \infty$, the p-norm becomes the maximum

absolute value of the elements of the vector (x_i), known as the infinity norm, or simply the maximum norm.

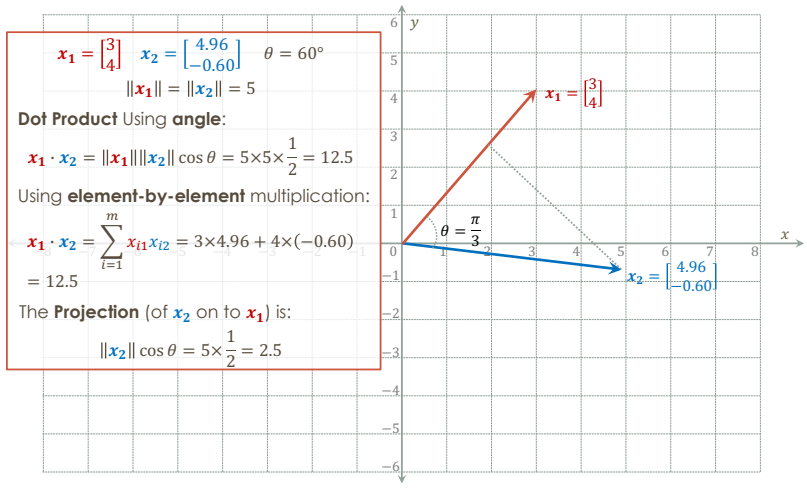


Fig. 2.5 Dot product between two vectors $\boldsymbol{x_1}, \boldsymbol{x_2} \in \mathbb{R}^2$. We compute the dot product using the elements of the vectors as well as the angle between them and show that we get the same result.

Dot Product as Projection: The dot product can also be defined using the angle between the two vectors. Let's consider two vectors $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ with an angle θ between them. Then,

$$\boldsymbol{x_1} \cdot \boldsymbol{x_2} \stackrel{\text{def}}{=} \|\boldsymbol{x_1}\|\|\boldsymbol{x_2}\| \cos\theta$$

Rearranging this definition, we can see that what it is describing is the projection of one vector in the direction of the other.

$$\|\boldsymbol{x_2}\| \cos\theta = \frac{\boldsymbol{x_1} \cdot \boldsymbol{x_2}}{\|\boldsymbol{x_1}\|} = \frac{\boldsymbol{x_1}}{\|\boldsymbol{x_1}\|} \cdot \boldsymbol{x_2} = \hat{\boldsymbol{x}}_1 \cdot \boldsymbol{x_2}$$

$\hat{\boldsymbol{x}}_1$ is the unit vector in the direction of the first vector. $\|\boldsymbol{x_2}\| \cos\theta$ is the projection of the second vector onto this direction. As a consequence, if the angle between the two vectors $\theta = \frac{\pi}{2}$, the projection is zero. If the angle is zero, the projected length is the same as the length of the second vector.

Quantum Computing

The backbone of Quantum Mechanics is, in fact, Linear Algebra. By the time we finish Chapter 11 (Eigenvalue Decomposition), we will have learned everything we need to deal with the mathematics of QM. However, we may not be able to understand the lingo because physicists use a completely different and alien-looking notation. Let's go through this so-called Dirac (AKA "bra-ket") notation for two good reasons. Firstly, we may come across it in our online searches on Linear Algebra topics. Secondly, closer to our field, Quantum Computing is gathering momentum. And the lingo used in the description of its technical aspects is likely to be the one from physics.

A vector in QM is a *ket* vector. What we usually write as $\boldsymbol{x}$ would appear as $|x\rangle$ in this notation. The transpose of a vector is the *bra* vector, written as $\boldsymbol{y}^\mathsf{T} \equiv \langle y|$. Therefore, a dot product $\boldsymbol{x} \cdot \boldsymbol{y} = \boldsymbol{x}^\mathsf{T} y \equiv \langle x|y\rangle$.

Now that we got started with QM, let's go ahead and complete the physics story. The QM vectors are typically the *wave functions* of the *probability amplitudes*. So, if we have an electron with a wave function $|\psi\rangle = \psi(x)$ (x being its location in a one dimensional problem), what it is describing is the probability amplitude of finding the electron at x. The corresponding probability is the square of the norm of this vector, which is $\langle\psi|\psi\rangle$.

There are a couple of complications here: Since $|\psi\rangle$ is actually a function $\psi(x)$, it has a value at each point x, and if we are going to think of it as a vector, it is an infinite-dimensional vector. Secondly, the values of $|\psi\rangle$ can be (and typically are) complex numbers. So when we take the norm, since we like the norm to be positive, we cannot merely take the transpose, we have to take the complex-conjugate transpose (called the Hermitian transpose, or simply conjugate transpose). Lastly, the analog of summation of elements, when we have an infinity of them, is going to be an integral. The space in which such wave functions live is called the *Hilbert Space*.

Finally, toward the end of this book, we will come across expressions like $\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x}$ which will look like $\langle\psi|H|\psi\rangle$. The matrix $\boldsymbol{A}$ has become an *operator* H corresponding to a physical observable (in this case, the energy, if H is the so-called *Hamiltonian*), and the values we can get are, in fact, its eigenvalues, which can be thought of as the reason why the observable can take only discrete, quantized values. That, in a nutshell, is how we get the various allowed energy levels in a Hydrogen atom.

The two definitions of the dot product are equivalent, which is a remarkable fact. In other words,

$$\sum_{1}^{n} x_{1i}x_{2i} = \|\boldsymbol{x_1}\|\|\boldsymbol{x_2}\| \cos\theta$$

It can be proven using the Law of Cosines. Figure 2.5 shows the equivalence of the two definitions using an example of two vectors in $\mathbb{R}^2$, one projecting onto the other.

2.4 Matrices

We introduced the notion of a matrix when we explored the conditions of linearity for functions and expressions with two variables in §1.1.4, page 15. Now it is time to describe it more completely and formally. Exactly like vectors, matrices are tables of numbers. In Computer Science, the numbers we are dealing with are typically real (which we call `floating` or `double` in various programming languages). We will, therefore, talk about matrices over the field of real numbers, and write $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, which says $\boldsymbol{A}$ is a matrix over the field of reals, with m rows and n columns (rows first, always).

2.5 Matrix Operations

The first two operations on matrices that we will define are identical to the ones for vectors.

2.5.1 Scalar Multiplication

Definition: For any matrix $\boldsymbol{A}$ and any scalar s, the result of scalar multiplication ($s\boldsymbol{A}$) is defined as follows:

$$\begin{aligned} \forall\ \boldsymbol{A} &= \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & a_{ij} & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n} \text{ and } s \in \mathbb{R}, \\ s\boldsymbol{A} &\stackrel{\text{def}}{=} \begin{bmatrix} sa_{11} & \cdots & sa_{1n} \\ \vdots & sa_{ij} & \vdots \\ sa_{m1} & \cdots & sa_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n} \end{aligned} \tag{2.7}$$

In other words, when we multiply a matrix by a scalar, we multiply each of its elements by the scalar. The resulting matrix is of the same dimension: If $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, then $s\boldsymbol{A} \in \mathbb{R}^{m \times n}$.

2.5.2 Matrix Addition

Definition: For any two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, their sum (the result of their addition, $\boldsymbol{A}+\boldsymbol{B}$) is defined as follows:

$$\begin{gathered}\forall\ \boldsymbol{A}=\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & a_{ij} & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \text{ and } \boldsymbol{B}=\begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & b_{ij} & \vdots \\ b_{m1} & \cdots & b_{mn} \end{bmatrix} \in \mathbb{R}^{m\times n} \\ \boldsymbol{A}+\boldsymbol{B} \stackrel{\text{def}}{=} \begin{bmatrix} a_{11}+b_{11} & \cdots & a_{1n}+b_{1n} \\ \vdots & a_{ij}+b_{ij} & \vdots \\ a_{m1}+b_{m1} & \cdots & a_{mn}+b_{mn} \end{bmatrix} \in \mathbb{R}^{m\times n}\end{gathered} \tag{2.8}$$

To add two matrices, we simply add the corresponding elements. Naturally, the two matrices we are trying to add should have the same dimensions.

As we can see, although the notations are slightly different, vectors and matrices have the same operations, at least when it comes to scalar multiplication and addition. Soon, we will see that the dot product, a quintessentially vector-type operation (with the cosine of the angle between them) is also, in fact, a matrix operation. The reason for this connection is simple: A vector in Linear Algebra is a matrix with just one column; it is a column matrix. Everything we specify for matrices applies to vectors as well.

It may be worth our time to compare the first two operations that we defined for vectors and matrices (namely, scalar multiplication and addition) to the conditions of linearity (namely, homogeneity and additivity, described in §1.1.3, page 15). The fact that they seem parallel is not an accident, but points to the underlying structure of Linear Algebra.

2.6 Properties of Scaling and Addition

We have to keep in mind that in matrix (and therefore vector) addition, the dimensions should match: $\boldsymbol{A}+\boldsymbol{B}$ is defined if and only if they both have the same number of rows and columns. In particular, a vector cannot add to a matrix (of more than one column). Vectors of different lengths cannot interact with each other; they live in different spaces. A zero vector in $\mathbb{R}^3$ (our 3-D space) is not the same as the zero vector in $\mathbb{R}^2$ (a plane).

When dealing with matrices (or vectors) of compatible sizes, we can list the mathematical properties of scalar multiplication and addition. Both these operations are:

Commutativity

The order in which the matrices (or vectors) appear in the operations does not matter.

$$s\boldsymbol{x} = \boldsymbol{x}s; \quad s\boldsymbol{A} = \boldsymbol{A}s$$
$$\boldsymbol{x_1} + \boldsymbol{x_2} = \boldsymbol{x_2} + \boldsymbol{x_1}; \quad \boldsymbol{A} + \boldsymbol{B} = \boldsymbol{B} + \boldsymbol{A}$$

Associativity

We can group and perform the operations in any order we want.

$$s_1 s_2 \boldsymbol{x} = s_1(s_2\boldsymbol{x}) = (s_1 s_2)\boldsymbol{x}; \quad s_1 s_2 \boldsymbol{A} = (s_1 s_2)\boldsymbol{A} = s_1(s_2\boldsymbol{A})$$
$$\boldsymbol{x_1} + \boldsymbol{x_2} + \boldsymbol{x_3} = (\boldsymbol{x_1} + \boldsymbol{x_2}) + \boldsymbol{x_3} = \boldsymbol{x_1} + (\boldsymbol{x_2} + \boldsymbol{x_3})$$
$$\boldsymbol{A} + \boldsymbol{B} + \boldsymbol{C} = (\boldsymbol{A} + \boldsymbol{B}) + \boldsymbol{C} = \boldsymbol{A} + (\boldsymbol{B} + \boldsymbol{C})$$

Distributivity

Scalar multiplication distributes over matrix addition.

$$s(\boldsymbol{x_1} + \boldsymbol{x_2}) = s\boldsymbol{x_1} + s\boldsymbol{x_2}; \quad s(\boldsymbol{A} + \boldsymbol{B}) = s\boldsymbol{A} + s\boldsymbol{B}$$
$$(s_1 + s_2)\boldsymbol{x} = s_1\boldsymbol{x} + s_2\boldsymbol{x}; \quad (s_1 + s_2)\boldsymbol{A} = s_1\boldsymbol{A} + s_2\boldsymbol{A}$$

These properties are not arbitrarily imposed, but the consequences of the definitions of the operations of scalar multiplication and matrix addition. In other words, they can be proven to be true starting from the definitions.

2.7 Matrix Multiplication

We will now define and describe how matrices multiply. Two matrices can be multiplied to get a new matrix, but only under certain conditions. Matrix multiplication, in fact, forms the backbone of much of subject matter to follow. For that reason, we will look at it carefully, and from different perspectives.

2.7.1 Conformance Requirement

In order to multiply one matrix with another, the number of *columns* of the *first* has to be the same as the number of *rows* of the *second*. The product will have the same number of rows as the first matrix, and the same number of columns as the second one. In other words, if $\boldsymbol{A} \in \mathbb{R}^{m\times k}$ and $\boldsymbol{B} \in \mathbb{R}^{k\times n}$, then they can be multiplied to give $\boldsymbol{AB} \in \mathbb{R}^{m\times n}$.

Note that this conformance requirement is different from the one for matrix addition, where the individual matrices and the sum are of the *same* size. In contrast, for matrix multiplication, if the individual matrices are of the same size, they *cannot* be multiplied, unless they both have the same number of rows and columns (in which case, we will call the square matrices).

Element-wise Multiplication

Definition: For conformant matrices, we define the matrix multiplication as follows: The element in the i^{th} row and the j^{th} column of the product is the sum of the products of the elements in i^{th} row of the first matrix and the j^{th} column of them second matrix.

More formally, for any two conformant matrices $\boldsymbol{A} \in \mathbb{R}^{m\times k}$ and $\boldsymbol{B} \in \mathbb{R}^{k\times n}$, their product (the result of their multiplication, $\boldsymbol{C} = \boldsymbol{AB}$) is defined as follows:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & a_{il} & \vdots \\ a_{m1} & \cdots & a_{mk} \end{bmatrix} \text{ and } \boldsymbol{B} = \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & b_{lj} & \vdots \\ b_{k1} & \cdots & b_{kn} \end{bmatrix},$$
$$\boldsymbol{AB} = \boldsymbol{C} = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & c_{ij} & \vdots \\ c_{m1} & \cdots & c_{mn} \end{bmatrix} \in \mathbb{R}^{m\times n} \text{ where} \tag{2.9}$$
$$c_{ij} \overset{\text{def}}{=} a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ik}b_{kj} = \sum_{l=1}^{k} a_{il}b_{lj}$$

Since the definition of is bit cumbersome, let us take a look at what it is saying using Figure 2.6.

Note that matrix multiplication is not commutative. $\boldsymbol{AB} \neq \boldsymbol{BA}$, in general. In fact, for general (non-square) matrices, if they are conformant for the multiplication $\boldsymbol{AB}$, they cannot be conformant for $\boldsymbol{BA}$ and the product is not even defined.

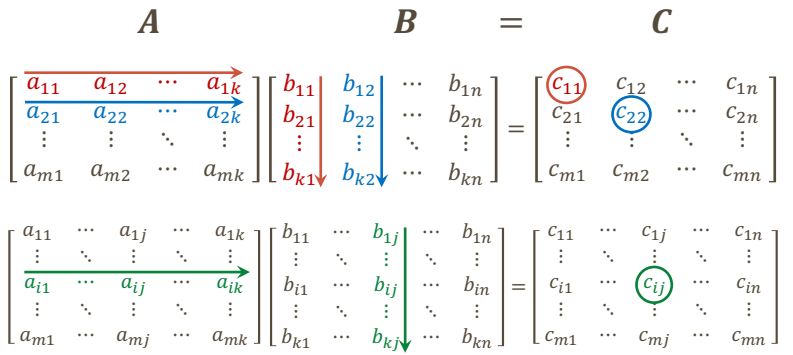


Fig. 2.6 Illustration of matrix multiplication $\boldsymbol{AB} = \boldsymbol{C}$. In the top panel, the element c_{11} (in red) of the product $\boldsymbol{C}$ is obtained by taking the elements in the first row of $\boldsymbol{A}$ and multiplying them with the elements in the first column of $\boldsymbol{B}$ (shown in red letters and arrows) and summing them up. c_{22} (blue) is the sum-product of the second row of $\boldsymbol{A}$ and the second column of $\boldsymbol{B}$ (blue). In the bottom panel, we see a general element, c_{ij} (green) as the sum-product of the i^{th} row of $\boldsymbol{A}$ and the j^{th} column of $\boldsymbol{B}$ (in green).

2.7.2 Vector Dot Product

Now that we have defined the general matrix multiplication $\boldsymbol{AB} = \boldsymbol{C}$, with $\boldsymbol{A} \in \mathbb{R}^{m\times k}$, $\boldsymbol{B} \in \mathbb{R}^{k\times n}$, and $\boldsymbol{C} \in \mathbb{R}^{m\times n}$, let's consider a special case when $m = n = 1$. Both $\boldsymbol{A}$ and $\boldsymbol{B}$ have k numbers in them, arranged horizontally and vertically.

$$\begin{aligned} \boldsymbol{A} &= \begin{bmatrix} a_1 & a_2 & \cdots & a_k \end{bmatrix} = \boldsymbol{a}^{\mathsf{T}} \quad \boldsymbol{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{bmatrix} = \boldsymbol{b} \\ \boldsymbol{C} &= [c_{11}] = a_1 b_1 + a_2 b_2 + \cdots + a_k b_k = \sum_{i=1}^{k} a_i b_i = s \end{aligned} \tag{2.10}$$

In this case, $\boldsymbol{C}$ has become a matrix of one row and one column, which is the same as a scalar. We can, therefore, use the symbol s to represent it. $\boldsymbol{B}$ is a column matrix with k rows, which is what we earlier called a vector; remember, our vectors are all column vectors. Let's use the symbol $\boldsymbol{b}$ instead of $\boldsymbol{B}$ to stay consistent in our

notations. Now, $\boldsymbol{A}$ is a special matrix with k elements arranged in one row. Some people call it a row vector, but let's call it the *transpose* of a column vector, $\boldsymbol{a}$, and denoted it by $\boldsymbol{a}^\mathsf{T}$. We get the transpose of a matrix switching its rows and columns. In the case of a column matrix (a vector), which is a sequence of numbers standing vertically, its transpose happens when we make them lie down horizontally.

What we have written down above is mathematically identical to our definition of vector dot product in Eqn (2.4), with $\boldsymbol{a} = \boldsymbol{x_1}$ and $\boldsymbol{b} = \boldsymbol{x_2}$. In other words, the vector dot product $\boldsymbol{a} \cdot \boldsymbol{b}$ is identical to the matrix product $\boldsymbol{a}^\mathsf{T}\boldsymbol{b}$. Note that $\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{b} \cdot \boldsymbol{a}$ and $\boldsymbol{a}^\mathsf{T}\boldsymbol{b} = \boldsymbol{b}^\mathsf{T}\boldsymbol{a}$, as we can see from the summation in Equations (2.4) and (2.10). Since this dot product is something we may need to cross-reference later on, let's restate it and give it a new equation number:

$$\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}^\mathsf{T}\boldsymbol{b} = \boldsymbol{b}^\mathsf{T}\boldsymbol{a} = \boldsymbol{b} \cdot \boldsymbol{a} \tag{2.11}$$

As we see clearly now, the vector dot product is a special case of matrix multiplication. In fact, all the operations we defined for vectors are special cases of the corresponding operations for matrices, which is not surprising because our vectors are merely matrices with one column. We actually used matrix multiplication way back in Eqn (1.1), where we called it a notational trick.

Multiplication as Dot Products

Definition: In $\boldsymbol{AB} = \boldsymbol{D}$ (with $\boldsymbol{A} \in \mathbb{R}^{m\times k}$, $\boldsymbol{B} \in \mathbb{R}^{k\times n}$, and $\boldsymbol{D} \in \mathbb{R}^{m\times n}$)[5], $\boldsymbol{A}$ consists of m rows $\boldsymbol{r_i}^\mathsf{T}$, each of which have k elements and $\boldsymbol{B}$ consists of n column vectors $\boldsymbol{c_j}$ with k elements each. Then the element d_{ij} of the product is the dot product of $\boldsymbol{r_i}$ and $\boldsymbol{c_j}$.

$$d_{ij} = \boldsymbol{r_i} \cdot \boldsymbol{c_j} = \boldsymbol{r_i}^\mathsf{T}\boldsymbol{c_j}$$

Here, we are restating the matrix multiplication using the definition of vector dot product, rather cyclically. We think of $\boldsymbol{A}$ consisting of m vectors $\boldsymbol{r_i}$ arranged horizontally. To be more precise, $\boldsymbol{A}$ consists of m rows $\boldsymbol{r_i}^\mathsf{T}$, since our vectors are column vectors, as are the n columns in the matrix $\boldsymbol{B}$ which we call $\boldsymbol{c_j}$. Because of the matrix dimensions, all $\boldsymbol{r_i}^\mathsf{T}$ and $\boldsymbol{c_j}$ have k elements each, and we are allowed to take their dot products.

[5] We are using $\boldsymbol{D}$ (instead of $\boldsymbol{C}$) as the product in order to avoid possible confusion between the symbols for column vectors $\boldsymbol{c_j}$ and the elements of the product matrix.

2.7.3 Block-wise Matrix Multiplication

We can perform matrix multiplication in a block-wise fashion, where we segment the matrices we are multiplying into blocks that are conformant sub-matrices, and multiply block-by-block. While this statement may be of dubious usefulness, we have already used it above, in defining matrix multiplication as dot product of the rows of the first and the columns of the second. What we did was to divide up the matrices into smaller blocks that are conformant for multiplication and define the product in terms of the products of the smaller blocks.

The block-wise multiplication brings out the recursive nature of the operation. In the product $\boldsymbol{C} = \boldsymbol{AB}$, if $\boldsymbol{A}$ is segmented into $m \times k$ blocks $\boldsymbol{A}_{il}$ and $\boldsymbol{B}$ into $\boldsymbol{B}_{lj}$ ($k \times n$ of them), we can write the block $\boldsymbol{C}_{ij}$ as:

$$\boldsymbol{C}_{ij} = \sum_{l=1}^{k} \boldsymbol{A}_{il}\boldsymbol{B}_{lj}$$

provided that the block $\boldsymbol{A}_{il}$ is conformant for multiplication with $\boldsymbol{B}_{lj}$. Compare this equation to Eqn (2.9) and we can immediately see that the latter is a special case of partitioning $\boldsymbol{A}$ and $\boldsymbol{B}$ into blocks of single elements.

In general, segmenting matrices into comformant blocks may not be trivial, but we have advanced topics in Linear Algebra where it comes in handy. For our purposes, we can think of few cases of simple segmentation of a matrix and perform block-wise multiplication.

Special Cases of Block-wise Multiplication

Here is a situation where block-wise multiplication makes sense: If we have $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n\times 2n}$, we can certainly multiply them and get $\boldsymbol{AB} \in \mathbb{R}^{n\times 2n}$. We can also think of $\boldsymbol{B}$ as composed of two matrices, $\boldsymbol{B_1}$ and $\boldsymbol{B_2}$, both of size $\mathbb{R}^{n\times n}$, arranged side-by-side. In other words, $\boldsymbol{B}$ consists of two blocks of square matrices and can be written as $\boldsymbol{B} = [\boldsymbol{B_1} \mid \boldsymbol{B_2}]$. The notation $[\boldsymbol{C} \mid \boldsymbol{D}]$ stands for a new matrix in which we have the columns of $\boldsymbol{C}$ and $\boldsymbol{D}$ side-by-side. Later on, we will call such matrices "augmented matrices." Now,

$$\boldsymbol{AB} = \boldsymbol{A}[\boldsymbol{B_1} \mid \boldsymbol{B_2}] = [\boldsymbol{AB_1} \mid \boldsymbol{AB_2}]$$

which is like distributing the matrix multiplier $\boldsymbol{A}$ over the operation of segmenting the matrix into blocks. We will use this type of block-wise matrix multiplication in understanding some of the techniques later. We have keep it in mind that the order in which the matrices appear in the product is significant because matrix multiplication is not commutative. In particular, $\boldsymbol{BA} \neq [\boldsymbol{B_1A} \mid \boldsymbol{B_2A}]$. In fact, the $\boldsymbol{B}$ and $\boldsymbol{A}$ are not even conformant for the multiplication $\boldsymbol{BA}$.

Another case where block-wise multiplication makes sense is when we think of the second matrix as blocks of column vectors. In other words, we have $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n \times k}$ and if we think of $\boldsymbol{B}$ as k column vectors $\boldsymbol{b_i}$ standing side-by-side, which we write as $\boldsymbol{B} = [\boldsymbol{b_i}]$. Then we have the block-wise multiplication,

$$\boldsymbol{AB} = \boldsymbol{A}[\boldsymbol{b_i}] = \boldsymbol{A}[\boldsymbol{b_1} \mid \boldsymbol{b_2} \mid \cdots \mid \boldsymbol{b_k}] = [\boldsymbol{Ab_1} \mid \boldsymbol{Ab_2} \mid \cdots \mid \boldsymbol{Ab_n}]$$

Earlier, we spoke of $\boldsymbol{Ax} = \boldsymbol{b}$ as set of m linear equations on n unknowns. We can think of the product $\boldsymbol{AB}$ as shown in this block-wise multiplication as a collection of k such sets. We will circle back to this notion in Chapter 5.

2.7.4 Column and Row Pictures of Matrix Multiplication

We can think of our first matrix as composed of column vectors, and the product as a linear combination of these vectors, which gives us the column picture of matrix multiplication. This column picture is a view that underpins several critical concepts in Linear Algebra, and we will come back to it time and again in this book.

Remember, we defined the Linear Combinations of vectors in §2.2.3 (page 25). Let's now restate the definition of matrix multiplication using the notion of the linear combinations of the columns of the first matrix, scaled by the elements of the second matrix. To keep it simple, let's first consider the multiplication of a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ by a column vector $\boldsymbol{x} \in \mathbb{R}^n$. The matrices are conformant, and the multiplication is allowed. We will get a vector as the product, which we will call $\boldsymbol{b} \in \mathbb{R}^m$. We can think of $\boldsymbol{A}$ as being composed of n column vectors ($\boldsymbol{c_i} \in \mathbb{R}^m$) standing side-by-side. The product $\boldsymbol{Ax}$ is then the linear combination of the columns of $\boldsymbol{A}$, scaled by the components of $\boldsymbol{x}$.

$$\boldsymbol{A} = \begin{bmatrix} | & | & & | \\ \boldsymbol{c_1} & \boldsymbol{c_2} & \cdots & \boldsymbol{c_n} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{m \times n} \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n \tag{2.12}$$

$$\boldsymbol{Ax} = \boldsymbol{b} = x_1\boldsymbol{c_1} + x_2\boldsymbol{c_2} + \cdots + x_n\boldsymbol{c_n} \in \mathbb{R}^m$$

Similarly, multiplication on the left gives us the row picture. Take $\boldsymbol{x}^\mathsf{T}$ to be a matrix of single row ($\boldsymbol{x}^\mathsf{T} \in \mathbb{R}^{1\times m}$), which means $\boldsymbol{x}$ is column vector $\boldsymbol{x} \in \mathbb{R}^m$. And $\boldsymbol{A} \in \mathbb{R}^{m\times n}$. Now the product $\boldsymbol{x}^\mathsf{T}\boldsymbol{A}$ is a row matrix $\boldsymbol{b}^\mathsf{T} \in \mathbb{R}^{1\times n}$. The row picture says that $\boldsymbol{x}^\mathsf{T}\boldsymbol{A}$ is a linear combination of the rows of the matrix $\boldsymbol{A}$.

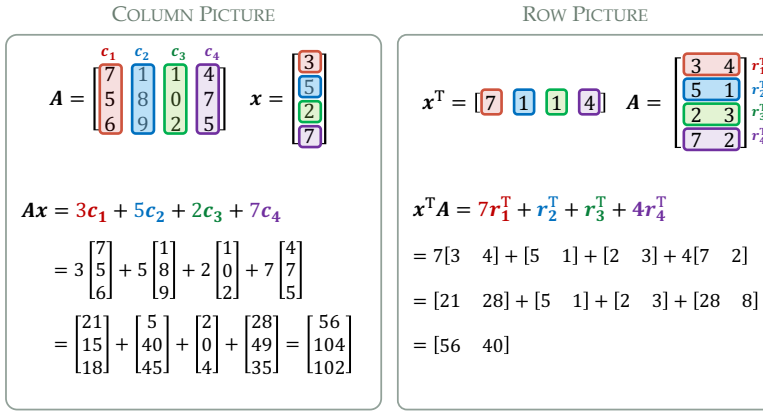


Fig. 2.7 Illustration of matrix multiplication as column and row pictures. On the left, we have $\boldsymbol{Ax}$, where the product (which is a column vector we might call $\boldsymbol{b}$) is the linear combination of the columns of $\boldsymbol{A}$. On the right, we have $\boldsymbol{x}^\mathsf{T}A$, where the product (say $\boldsymbol{b}^\mathsf{T}$) is a linear combination of the rows of $\boldsymbol{A}$.

Since the column and row pictures are hard to grasp as concepts, we illustrate them in Figure 2.7 using example matrices with color-coded rows and columns. We can use the basic element-wise multiplication (Eqn (2.9)) of matrices and satisfy ourselves that the column and row pictures indeed give the same numeric answers as element-by-element multiplication. To put it as a mnemonic, matrix multiplication is the linear combination of the *columns* of the matrix on the *left* and of the *rows* of the matrix on the *right*.

The column and row pictures are, in fact, special cases of the block-wise multiplication we discussed earlier. In the column picture, we are segmenting the first matrix into blocks that are columns, which are conformant for multiplication by single-element blocks (or scalars) of the second matrix. The sum of the individual products is then the linear combinations of the columns of the first matrix. We can visualize the row picture also as a similar block-wise multiplication.

If, instead of a vector $\boldsymbol{x}$, we had a multi-column second matrix, then the product also would have multiple columns. Each column of the product would then be the linear combinations of the columns of $\boldsymbol{A}$, scaled by the corresponding column of the second matrix. In other words, in $\boldsymbol{AB}, \boldsymbol{A} \in \mathbb{R}^{m \times k}, \boldsymbol{B} \in \mathbb{R}^{k \times n}$, the product has n columns, each of which is a linear combination of the columns of $\boldsymbol{A}$.

Considering this a teachable moment, let's look at it from the perspective of block-wise multiplication once more. Let's think of $\boldsymbol{B}$ as composed on of n column matrices stacked side-by-side as $\boldsymbol{B} = [\boldsymbol{b_1} \mid \boldsymbol{b_2} \mid \cdots \mid \boldsymbol{b_n}]$. Then, by block-wise multiplication, we have:

$$\boldsymbol{AB} = \boldsymbol{A}[\boldsymbol{b_1} \mid \boldsymbol{b_2} \mid \cdots \mid \boldsymbol{b_n}] = [\boldsymbol{Ab_1} \mid \boldsymbol{Ab_2} \mid \cdots \mid \boldsymbol{Ab_n}]$$

which says that each column of the product is $\boldsymbol{Ab_i}$, which, by the column picture of matrix multiplication, is a linear combination of the columns of $\boldsymbol{A}$ using the coefficients from $\boldsymbol{b_i}$.

2.7.5 Vector Inner and Outer Products

Vector dot product is commutative: $\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{b} \cdot \boldsymbol{a}$. This property can be easily verified by examining its definition in Eqn (2.10). When written using the matrix notation, commutativity means: $\boldsymbol{a}^\mathsf{T}\boldsymbol{b} = \boldsymbol{b}^\mathsf{T}\boldsymbol{a}$. However, note that matrix multiplication by itself is *not* commutative. In particular, $\boldsymbol{a}^\mathsf{T}\boldsymbol{b} \neq \boldsymbol{a}\boldsymbol{b}^\mathsf{T}$ (although both multiplications are between conformant matrices and therefore allowed). The latter is, in fact, not a scalar, but a matrix of size $k \times k$ if $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^k$. The matrix $\boldsymbol{a}\boldsymbol{b}^\mathsf{T}$ is called the *outer* product (as opposed to the dot product, which is the *inner* product) of the two vectors. The outer product has certain properties that will become important to us later in the book.

Other Operations

Although we have covered the mathematical operations in traditional Linear Algebra, there are a couple of more that are commonly used in Computer/Data Science.
***Hadamard Product*:** The Hadamard product, also known as the element-wise product or entry-wise product, is a mathematical operation that takes two matrices or vectors of the same dimensions and produces a new matrix or vector by multiplying corresponding elements together. In other words, each element in the resulting matrix/vector is obtained by multiplying the elements at the same position in the input matrices/vectors. This operation is often denoted by $\odot$ and is commonly used in various fields, including linear algebra, signal processing, and machine learning, for tasks such as pointwise multiplication of data or combining features.

For $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m\times n}$, the Hadamard product $\boldsymbol{C} = \boldsymbol{A} \odot \boldsymbol{B}$ has elements $c_{ij} = a_{ij}b_{ij}$. As we can see, $\boldsymbol{C} \in \mathbb{R}^{m\times n}$ as well.

***Scalar Addition*:** Scalar addition involves adding a scalar to each element of a matrix. This operation is performed by simply adding the scalar value to every element of the matrix, resulting in a new matrix with the same dimensions as the original. Scalar addition is used to shift or adjust the values in the matrix uniformly in some algorithms in CS.

For $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $s \in \mathbb{R}$, scalar addition can be formally written as $\boldsymbol{B} = \boldsymbol{A} + s$ has elements $b_{ij} = a_{ij} + s$. Again, $\boldsymbol{B} \in \mathbb{R}^{m\times n}$. Note that this operation is not really allowed in pure Linear Algebra. We cannot add entities belonging to $\mathbb{R}^{m\times n}$ and $\mathbb{R}$ together. But we can think of the operation as $\boldsymbol{B} = \boldsymbol{A} + s\boldsymbol{O}$, where $\boldsymbol{O} \in \mathbb{R}^{m\times n}$ is matrix with all ones as its elements: $o_{ij} = 1$.

We can ask whether these operations are linear. When we ask such a question, we have to carefully define what we mean. For the Hadamard product, let's think of it as a transformation. Rewriting it using our usual notations, we have $\boldsymbol{A} \odot \boldsymbol{X} = \boldsymbol{B}$, which is a transformation of $\boldsymbol{X}$ to $\boldsymbol{B}$. $\boldsymbol{A} : \boldsymbol{X} \mapsto \boldsymbol{B}$ or $\boldsymbol{A} : \mathbb{R}^{m\times n} \mapsto \mathbb{R}^{m\times n}$. Is this operation linear? Similarly, for scalar addition, think of it as $a + \boldsymbol{X} = \boldsymbol{B}$, which is a transformation $a : \mathbb{R}^{m\times n} \mapsto \mathbb{R}^{m\times n}$. Is it linear?

2.8 Generalized Vectors

We should keep in mind that, in its generality, vectors are defined only by their operations, namely scalar multiplication, addition, and the inner product. Any set of mathematical entities, literally any at all, for which we can consistently define these operations with the right commutative, additive and associative properties can be treated as vectors. Once we do that, the vast machinery of Linear Algebra stands ready to help us ensure consistency, derive insights and deepen our understanding further, which is the way it is used in modern physics, most notably in quantum mechanics and special relativity.

Convolution

Convolution in image processing involves sliding a small matrix (kernel) over an image. At each position, the kernel's values are multiplied with the underlying image pixels, and the results are summed to form a new pixel value in the output image. This operation is used for tasks like blurring, edge detection, and feature extraction.

Here's how convolution works in the context of image processing:

- **Input Image**: A two-dimensional matrix representing an image's pixel values.
- **Kernel/Filter**: A smaller matrix with numerical values defining the convolution operation.
- **Sliding**: The kernel is systematically moved over the image in small steps.
- **Element-Wise Multiplication**: Values in the kernel and underlying pixels are multiplied.
- **Summation**: The products are added up at each kernel position.
- **Output**: The sums are placed in the output matrix, which is also known as the feature map or convolved image.

Convolution is used for various image processing tasks:

- **Blurring/Smoothing**: By using a kernel with equal values, convolution can smooth an image, reducing noise and sharp transitions.
- **Edge Detection**: Specific kernels can detect edges in an image by highlighting areas with rapid intensity changes.
- **Feature Extraction**: Convolution with various filters can extract specific features from images, such as texture or pattern information.
- **Sharpening**: Convolution with a sharpening filter enhances edges and details in an image.

It is left as an exercise to the student to look up how exactly the convolution is performed, and whether it is linear.

Tensors and Tensor Products in Quantum Computing

***Tensors*:** At a basic level, a **tensor** is just a container of numbers, arranged in one or more dimensions:

- A **scalar** is a single number – a tensor of rank 0.
- A **vector** is a 1D list of numbers – a tensor of rank 1.
- A **matrix** is a 2D grid of numbers – a tensor of rank 2.
- A **rank-3 tensor** is like a 3D block of numbers – think of a cube of values.
- In general, a **tensor** is a multi-dimensional array.

In a general sense, a tensor is a multilinear object that can have more than two indices. For example, a rank-3 tensor T_{ijk} can be thought of as a 3-dimensional array of numbers. In computer science, tensors are used heavily in machine learning (especially deep learning) to represent data like images, audio, and video. In quantum computing, tensors are crucial for representing quantum states and quantum operations – especially when dealing with multiple qubits. To understand how quantum states scale with the number of qubits, we need to go beyond standard matrix multiplication and introduce tensor products.

***Vectors and Tensor Products*:** Let $\boldsymbol{u} \in \mathcal{C}^m$ and $\boldsymbol{v} \in \mathcal{C}^n$ be two vectors over the field of complex numbers. Their tensor (AKA Kronecker) product is denoted by $\boldsymbol{u} \otimes \boldsymbol{v}$ and defined as:

$$\boldsymbol{u} \otimes \boldsymbol{v} = \begin{bmatrix} u_1\boldsymbol{v} \\ u_2\boldsymbol{v} \\ \vdots \\ u_m\boldsymbol{v} \end{bmatrix} \in \mathcal{C}^{mn}$$

Each term like $u_1\boldsymbol{v}$ is, in fact, a scaled version of $\boldsymbol{v} \in \mathcal{C}^n$. So this tensor product $\boldsymbol{u} \otimes \boldsymbol{v}$ lies in $\mathcal{C}^{mn}$. Note that tensor products of vectors are not commutative: $\boldsymbol{u} \otimes \boldsymbol{v} \neq \boldsymbol{v} \otimes \boldsymbol{u}$, which is conceptually important in QC.

***Tensor Products of Matrices*:** If $\boldsymbol{A} \in \mathcal{C}^{m \times n}$ and $\boldsymbol{B} \in \boldsymbol{C}^{p \times q}$, their tensor product is defined as:

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \cdots & a_{1n}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \cdots & a_{2n}\boldsymbol{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}\boldsymbol{B} & a_{m2}\boldsymbol{B} & \cdots & a_{mn}\boldsymbol{B} \end{bmatrix} \in \mathcal{C}^{mp \times nq}$$

***Why Tensor Products Matter in QC*:** The state space of n qubits is the 2^n-dimensional complex vector space $\mathcal{C}^{2^n}$, formed by tensoring n copies of $\mathcal{C}^2$. This exponential scaling underlies the power of quantum computing – and the difficulty of simulating it classically. Tensor products are thus not just a mathematical convenience, but a foundational structure in quantum theory and – by extension, quantum computing.

3

Transposes and Determinants

I would rather have questions that can't be answered than answers that can't be questioned.

—Richard Feynman

Now that we have defined matrices and mastered the basic operations on them, let's look at another operation that comes up very often in Linear Algebra, namely taking the transpose of a matrix. We will also introduce the concept of the determinant of a matrix, which a single number with a lot of information about the matrix, and with a nice geometrical interpretation. Also in this chapter, we will go over the nomenclature of various special matrices and entities related to matrices with a view to familiarizing ourselves with the lingo of Linear Algebra. This familiarity will come in handy in later chapters.

3.1 Transpose of a Matrix

We get the transpose of a matrix by flipping it over its *main diagonal*. Before defining it more formally, let's first look at an example in

Eqn (3.1), where the so-called *main diagonal* is highlighted in bold. The transpose of a matrix is obtained, basically, by switching the rows and columns.

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{7} & 1 & 1 & 4 \\ 8 & \mathbf{8} & 0 & 7 \\ 6 & 9 & \mathbf{2} & 8 \end{bmatrix} \in \mathbb{R}^{3\times 4} \quad \boldsymbol{A}^{\mathsf{T}} = \begin{bmatrix} \mathbf{7} & 8 & 6 \\ 1 & \mathbf{8} & 9 \\ 1 & 0 & \mathbf{2} \\ 4 & 7 & 8 \end{bmatrix} \in \mathbb{R}^{4\times 3} \tag{3.1}$$

The *main diagonal* is the elements whose row number is the same as the column number: a_{ii}. Flipping a matrix over it means we take the element a_{ij} and put in the location of a_{ji}.

To make future definitions easier to write, let's first introduce a shorthand notation for a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ as $\left[a_{ij}\right]$, where we expect ourselves to understand, from the context, that we mean a_{ij} to stand for a typical element in $\boldsymbol{A}$, with the row index $i, 1 \leq i \leq m$ and the column index $j, 1 \leq j \leq n$. With this notation, we can state the definition of the transpose of a matrix as follows:

Matrix Transpose

Definition: For any matrix $\boldsymbol{A} = \left[a_{ij}\right] \in \mathbb{R}^{m\times n}$, its transpose is defined as $\boldsymbol{A}^{\mathsf{T}} \stackrel{\text{def}}{=} \left[a_{ji}\right] \in \mathbb{R}^{n\times m}$.

3.1.1 Properties of Transposes

From the definitions of transposes and the basic operations of matrices, we can prove the following properties.

1. **Transpose of a Transpose:** If we take the transpose twice, we get the original matrix back.
$$\left(\boldsymbol{A}^{\mathsf{T}}\right)^{\mathsf{T}} = \boldsymbol{A}$$
2. **Transpose of a Sum:** The transpose of the sum of two matrices is the sum of the transposes of the individual matrices.
$$(\boldsymbol{A} + \boldsymbol{B})^{\mathsf{T}} = \boldsymbol{A}^{\mathsf{T}} + \boldsymbol{B}^{\mathsf{T}}$$
3. **Scalar Multiplication:** The operation of taking the transpose of a matrix commutes with scalar multiplication.
$$(s\boldsymbol{A})^{\mathsf{T}} = s\boldsymbol{A}^{\mathsf{T}}$$

4. **Transpose of a Scalar:** Scalars can be considered a matrix of one row and one column. Therefore, it is its own transpose.

$$s \equiv [s] \implies s^{\mathsf{T}} = [s]^{\mathsf{T}} = [s] = s$$

3.1.2 Product Rule

The transpose of the product of two matrices is the product of the transposes, taken in the reverse order.

$$(\boldsymbol{AB})^{\mathsf{T}} = \boldsymbol{B}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}}$$

This product rule also can be proven by looking at the (i, j) element of the product matrices on the left and right hand sides, although it is a bit tedious to do so. Before proving the product rule, let's look at the dimensions of the matrices involved in the product rule and easily see why the rule makes sense.

Let's consider $\boldsymbol{A} \in \mathbb{R}^{m\times k}$ and $\boldsymbol{B} \in \mathbb{R}^{k\times n}$ so that $\boldsymbol{AB} \in \mathbb{R}^{m\times n}$. The dimensions have to be this way by the conformance requirement of matrix multiplication, which says that the number of columns of the first matrix has to be the same as the number of rows of the second one. Otherwise, we cannot define the product. In particular, for $m \neq n$, $\boldsymbol{BA}$ is not defined.

By the definition of transpose, we have

$$\boldsymbol{A} \in \mathbb{R}^{m\times k} \implies \boldsymbol{A}^{\mathsf{T}} \in \mathbb{R}^{k\times m} \text{ and } \boldsymbol{B} \in \mathbb{R}^{k\times n} \implies \boldsymbol{B}^{\mathsf{T}} \in \mathbb{R}^{n\times k}$$

Note that the number of *columns* of $\boldsymbol{B}^{\mathsf{T}}$ is the same as number of *rows* of $\boldsymbol{A}^{\mathsf{T}}$. Therefore, the product $\boldsymbol{B}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}}$ is well defined, while $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{B}^{\mathsf{T}}$ cannot be defined (unless $m = n$). Furthermore, $\boldsymbol{B}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}} \in \mathbb{R}^{n\times m}$.

And, from the definition of the transpose of a matrix again, we know that

$$\boldsymbol{AB} \in \mathbb{R}^{m\times n} \implies (\boldsymbol{AB})^{\mathsf{T}} \in \mathbb{R}^{n\times m}$$

which is the same as the dimensions of product of the transposes in the reverse order: $\boldsymbol{B}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}} \in \mathbb{R}^{n\times m}$. Therefore, at least from the perspective of conformance of matrix multiplication, the product rule of transposes makes sense.

Let's illustrate this product rule of transposes using an example with two simple matrices $\boldsymbol{A}$ and $\boldsymbol{B}$:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \text{ and } \boldsymbol{B} = \begin{bmatrix} 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$\boldsymbol{AB} = \begin{bmatrix} 34 & 40 \\ 79 & 94 \end{bmatrix} \implies (\boldsymbol{AB})^{\mathsf{T}} = \begin{bmatrix} 34 & 79 \\ 40 & 94 \end{bmatrix}$$

$$\boldsymbol{B}^{\mathsf{T}} = \begin{bmatrix} 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix} \text{ and } \boldsymbol{A}^{\mathsf{T}} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

$$\boldsymbol{B}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}} = \begin{bmatrix} 34 & 79 \\ 40 & 94 \end{bmatrix} = (\boldsymbol{AB})^{\mathsf{T}}$$

We have postponed the proof for the product rule of transposes for as long as possible. Now, we will present two proofs.

Proof 1: Here, we will use the element-wise multiplication of matrices to prove the product rule. In $\boldsymbol{C} = \boldsymbol{AB}$, where $\boldsymbol{A} \in \mathbb{R}^{m \times k}$ and $\boldsymbol{B} \in \mathbb{R}^{k \times n}$, let's denote $\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{A'} = \left[a'_{ij}\right]$, $\boldsymbol{B}^{\mathsf{T}} = \boldsymbol{B'} = \left[b'_{ij}\right]$ and $\boldsymbol{C}^{\mathsf{T}} = \boldsymbol{C'} = \left[c'_{ij}\right]$.

(1) Element-wise matrix multiplication:	$c_{ij} = \sum_{p=1}^{k} a_{ip} b_{pj}$
(2) Definition of transpose:	$c'_{ij} = c_{ji}$
(3) Using (2), and $i \leftrightarrow j$:	$= \sum_{p=1}^{k} a_{jp} b_{pi}$
(4) Definition of transpose:	$= \sum_{p=1}^{k} a'_{pj} b'_{ip}$
(5) Rearranging:	$c'_{ij} = \sum_{p=1}^{k} b'_{ip} a'_{pj}$
(6) Recognizing (5) as matrix product:	$\boldsymbol{C}^{\mathsf{T}} = \boldsymbol{B}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}}$

Proof 2: Here, we will use the dot-product view of matrix multiplication to prove the product rule. In $\boldsymbol{C} = \boldsymbol{AB}$, let's denote $\boldsymbol{C}^{\mathsf{T}} = \boldsymbol{C'}$. Noting that the element c_{ij} of $\boldsymbol{C}$ is the dot product of i^{th} row of $\boldsymbol{A}$

and j^{th} column of $\boldsymbol{B}$, we can write:

(1) The element c_{ij} as dot product:	$c_{ij} = \boldsymbol{a_i} \cdot \boldsymbol{b_j}$
(2) Definition of transpose:	$c'_{ij} = c_{ji}$
(3) Using (1), and switching $i \leftrightarrow j$:	$c_{ji} = \boldsymbol{a_j} \cdot \boldsymbol{b_i}$
(4) Since dot product is commutative:	$c_{ji} = \boldsymbol{b_i} \cdot \boldsymbol{a_j}$
(5) Using (2) and the definition of transposes:	$c'_{ij} = \boldsymbol{b_i} \cdot \boldsymbol{a_j} = \boldsymbol{b'_i} \cdot \boldsymbol{a'_j}$
(6) Writing (5) in matrix form:	$\boldsymbol{C'} = \boldsymbol{C^\mathsf{T}} = \boldsymbol{B^\mathsf{T} A^\mathsf{T}}$

In (5), we used the fact that the i^{th} row of a matrix is the same as the i^{th} column of its transpose. In both proofs, we have shown that $\boldsymbol{C^\mathsf{T}} = (\boldsymbol{AB})^\mathsf{T} = \boldsymbol{B^\mathsf{T} A^\mathsf{T}}$.

3.2 Definitions and Matrices with Special Properties

Main Diagonal: The line of elements in a matrix with the same row and column indexes is known as the main diagonal. It may be referred to as the leading diagonal as well. In other words, it is the line formed by the elements a_{ii} in $\boldsymbol{A} = \begin{bmatrix} a_{ij} \end{bmatrix} \in \mathbb{R}^{m \times n}$. Note that number of rows and columns the matrix $\boldsymbol{A}$ does not have to be equal.

Square Matrix: When the number of rows in a matrix is the same as the number of columns, we call it a square matrix. If $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ then $\boldsymbol{A}$ is a square matrix.

Here is an interesting fact: For any matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{AA^\mathsf{T}} \in \mathbb{R}^{m \times m}$ and $\boldsymbol{A^\mathsf{T} A} \in \mathbb{R}^{n \times n}$. Therefore, for any matrix (of any size), its product with its transpose (multiplying on the right or left) is always a square matrix.

Symmetric Matrix: A matrix is symmetric when its transpose is identical to itself. $\boldsymbol{A^\mathsf{T}} = \boldsymbol{A} \implies \boldsymbol{A}$ is symmetric. This condition cannot be satisfied by a non-square matrix, and therefore all symmetric matrices are square matrices. For a symmetric matrix, $a_{ij} = a_{ji}$.

For a square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, $\boldsymbol{A} + \boldsymbol{A^\mathsf{T}}$ is always a symmetric matrix, which is easily proved by the fact that the transpose of the sum of two matrices is the sum of their transposes.

Skew-Symmetric Matrix: A matrix is skew symmetric (AKA anti-symmetric) when its transpose is the negative of itself. $\boldsymbol{A}^\mathsf{T} = -\boldsymbol{A} \implies \boldsymbol{A}$ is skew symmetric. Like in the previous case, this condition also cannot be met by a non-square matrix, and therefore all skew-symmetric matrices are also square matrices. For a skew-symmetric matrix, $a_{ij} = -a_{ji}$.

For a square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{A} - \boldsymbol{A}^\mathsf{T}$ is always a skew-symmetric matrix, which can be proven using the same technique as for $\boldsymbol{A} + \boldsymbol{A}^\mathsf{T}$ being symmetric.

Gram Matrix: Another interesting fact: For any matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, what is $\left(\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\right)^\mathsf{T}$? By the product rule of matrix transposes,

$$\left(\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\right)^\mathsf{T} = \boldsymbol{A}^\mathsf{T}\left(\boldsymbol{A}^\mathsf{T}\right)^\mathsf{T} = \boldsymbol{A}^\mathsf{T}\boldsymbol{A}$$

In other words, $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$ is symmetric. So is $\boldsymbol{A}\boldsymbol{A}^\mathsf{T}$, by the same argument. $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$ is called the Gram Matrix.

Diagonal Matrix: A matrix that has zero elements everywhere other than the main diagonal elements is called a diagonal matrix. It is usually a square matrix, but we can call a non-square matrix also a diagonal matrix without ambiguity because we have a clear definition for the main diagonal. Using symbols, $\boldsymbol{A} = \left[a_{ij}\right] \in \mathbb{R}^{m\times n}$ is a diagonal matrix if $a_{ij} = 0$ for all $i \neq j$. Note that we do not specify the values of the diagonal elements in anyway: They may or may not be zero. In particular, a zero matrix (with all zero elements) is also a diagonal matrix.

Identity Matrix: The identity matrix is a square, diagonal matrix with ones along the diagonal and zeros everywhere else. $\boldsymbol{A} = \boldsymbol{I}$ if and only if $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $a_{ii} = 1$ and $a_{ij} = 0$ for $i \neq j$. We use the symbol $\boldsymbol{I}$ or $\boldsymbol{I}_n$ to refer to the identity matrix.

Unit Vectors We can think of the columns of the identity matrix as unit vectors defining directions in space. For $\boldsymbol{I}_3 \in \mathbb{R}^{3\times 3}$, the unit vectors ($\hat{i}, \hat{j}$ and $\hat{k}$, as some people, especially physicists, denote them) would be:

$$\boldsymbol{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \hat{i} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \hat{j} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \hat{k} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Now, we can write any general vector, say,

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \text{as} \quad \boldsymbol{x} = x\hat{i} + y\hat{j} + z\hat{k}$$

Note that the unit vectors, as defined here, all have unit length. They are also perpendicular to each other because the dot product of any distinct pair of them is zero.

Upper and Lower Triangular Matrices: An upper triangular matrix ($\boldsymbol{U} = \begin{bmatrix} u_{ij} \end{bmatrix} \in \mathbb{R}^{m \times n}$) is the one with all the elements *below* the main diagonal zero.

$$u_{ij} = 0 \text{ for } i > j$$

Similarly, a lower triangular matrix ($\boldsymbol{L} = \begin{bmatrix} l_{ij} \end{bmatrix} \in \mathbb{R}^{m \times n}$) is the one with all the elements *above* the main diagonal zero.

$$l_{ij} = 0 \text{ for } i < j$$

Here are some examples:

$$\boldsymbol{L} = \begin{bmatrix} a & 0 \\ c & d \end{bmatrix} \in \mathbb{R}^{2 \times 2} \quad \boldsymbol{U} = \begin{bmatrix} a & b \\ 0 & d \end{bmatrix} \in \mathbb{R}^{2 \times 2}$$

Again, note that we do not specify anything about nonzero elements in the definitions. They can be zeros as well, and a diagonal matrix, in principle, is both an upper and a lower triangular matrix at the same time.

Inverse of a Matrix: A (square) matrix ($\boldsymbol{A}$), when multiplied by its inverse ($\boldsymbol{A^{-1}}$) will result in the identity matrix ($\boldsymbol{I}$): $\boldsymbol{A^{-1}A} = \boldsymbol{AA^{-1}} = \boldsymbol{I}$. For non-square matrices, we can have *left* and *right* inverses, but they will be different from each other.

Singular Matrix: Not all square matrices have inverses. If a matrix is *noninvertible*, it is called singular.

3.3 Determinant of a Matrix

The determinant of a square matrix is a scalar value derived out of its elements. It contains a large amount of information about

Morphisms

Earlier (in §1.1.5, page 16), we talked about how a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ as encoding a linear transformations between $\mathbb{R}^n$ and $\mathbb{R}^m$. $\boldsymbol{A} : \mathbb{R}^n \mapsto \mathbb{R}^m$. Other names for a linear transformation include linear map, linear mapping, homomorphism etc. If $n = m$, then $\boldsymbol{A}$ maps $\mathbb{R}^n$ to itself: Every vector ($\boldsymbol{x}$) in $\mathbb{R}^n$, when multiplied by $\boldsymbol{A}$, gives us another vector $\boldsymbol{b} \in \mathbb{R}^n$. The name used for it is a linear **endomorphism**. Note that two *different* vectors $\boldsymbol{x}_1, \boldsymbol{x}_2$ do not *necessarily* have to give us two different vectors. If they do, then the transformation is called an **automorphism**. Such a transformation can be *inverted*: We can find another transformation that will reverse the operation.

To complete the story of morphisms, a transformation $\mathbb{R}^n \mapsto \mathbb{R}^m$ (with n and m not necessarily equal) is called an **isomorphism** if it can be reversed, which means that it is a one-to-one *and* onto mapping, also known as bijective. If two vectors in $\mathbb{R}^n$ ($\boldsymbol{x}_1$ and $\boldsymbol{x}_2$) map to the same vector in $\mathbb{R}^m$ ($\boldsymbol{b}$), then given $\boldsymbol{b}$, we have no way of knowing which vector ($\boldsymbol{x}_1$ or $\boldsymbol{x}_2$) it came from, and we cannot invert the operation.

Of course, these morphisms are more general: They are not necessarily between the so-called Euclidean spaces $\mathbb{R}^n$, but between any mathematical structure. We, for our purposes in computer science, are interested only in $\mathbb{R}^n$ though.

Let's illustrate the use of the idea of isomorphisms with an (admittedly academic) example. We know that the number of points in a line segment between 0 and 1 is infinite. So is the number of points in a square of side 1. Are these two infinities the same?

If we can find an isomorphism from the square (in $\mathbb{R}^2$) to the line ($\mathbb{R}$), then we can argue that they are. Here is such an isomorphism: For any point in the square, take its coordinates, $0 < x, y < 1$. Express them as decimal numbers. Create a new number x' by taking the first digit of x, then the first digit of y, followed by the second digit of x, second digit of y and so on, thereby interleaving the coordinates into a new number. As we can see, $0 < x' < 1$, and this transformation $\boldsymbol{T}$ is a one-to-one mapping and onto, or an isomorphism: $\boldsymbol{T} : (x, y) \mapsto x' : \mathbb{R}^2 \mapsto \mathbb{R}$. It is always possible to reverse the operation and find x and y given any x', $\boldsymbol{T^{-1}} : x' \mapsto (x, y) : \mathbb{R} \mapsto \mathbb{R}^2$. Therefore the infinities have to be equal.

Another transformation that is definitely not an isomorphism is the projection operation: Take any point (x, y) and project it to the x axis, so that the new $x' = x$. $\boldsymbol{P} : (x, y) \mapsto x : \mathbb{R}^2 \mapsto \mathbb{R}$ maps multiple points to the same number, and it cannot be reversed. $\boldsymbol{P}^{-1}$ does not exist.

Interestingly, $\boldsymbol{P}$ is a linear transformation, while $\boldsymbol{T}$ is not.

the matrix and the linear transformation (see §1.1.5, page 16) that it represents. We use the symbol $\det(\boldsymbol{A}), \Delta\boldsymbol{A}$ or $|\boldsymbol{A}|$ to denote the determinant of the square matrix $\boldsymbol{A}$.

Before actually defining the determinant, let's formally state what we said in the previous paragraph.

$$|\boldsymbol{A}| = f(a_{ij}) \quad \text{where} \quad \boldsymbol{A} = \left[a_{ij}\right] \in \mathbb{R}^{n \times n}$$

which says that the determinant is a function (a hitherto unknown one) of the elements of the square matrix. We will soon specify what the function is. Before defining it, however, let us list some interesting facts about the determinant.

Thinking of $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ as a linear transformation, we can say that $|\boldsymbol{A}| = 0$ means that it is *not* a one-to-one mapping or an isomorphism. In other words, $\boldsymbol{A}$ takes multiple vectors in $\mathbb{R}^n$ to the *same* transformed vector (again in $\mathbb{R}^n$). See the box titled "**Morphisms**" for the formal nomenclature and some more details.

If the determinant is not zero, then $\boldsymbol{A}$ represents an isomorphism, and we can invert its transformation. We can find another matrix to do the reversal, and we will call it the inverse of $\boldsymbol{A}$, and denote it using the symbol $\boldsymbol{A^{-1}}$, which is indeed the same inverse we defined on page 51. The transformations that are not isomorphisms cannot be reversed, and the corresponding matrices are noninvertible; they are singular matrices. To use the right mathematical lingo, the determinant $|\boldsymbol{A}|$ being nonzero is a necessary and sufficient condition for the matrix $\boldsymbol{A}$ to be invertible.

As we shall see later, when the matrix $\boldsymbol{A}$ is invertible, the system of linear equations $\boldsymbol{Ax} = \boldsymbol{b}$ has a unique solution. We can think of this characteristic of the determinant an algebraic property because it says something about the solutions of linear equations. In addition, we will find a geometric property as well, stating that the determinant behaves like the volume of a parallelepiped that the matrix represents, which we will explore once we define what the determinant is. Parallelepiped, by the way, is the higher-dimensional generalization of a parallelogram, which is a diamond-shaped, sheared rectangle in 2-D.

3.3.1 2×2 Matrices

If we have a matrix $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$ as in the equation below, its determinant is defined as $|\boldsymbol{A}| = ad - bc$. To restate it formally,

$$\boldsymbol{A} = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \in \mathbb{R}^{2\times 2} \quad |\boldsymbol{A}| = \begin{vmatrix} a & c \\ b & d \end{vmatrix} \stackrel{\text{def}}{=} ad - bc \in \mathbb{R} \tag{3.2}$$

We can think of $\boldsymbol{A}$ as having two column vectors ($\boldsymbol{c_1}$ and $\boldsymbol{c_2}$) standing side-by-side.

$$\boldsymbol{A} = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \in \mathbb{R}^{2\times 2} \quad \boldsymbol{c_1} = \begin{bmatrix} a \\ b \end{bmatrix}, \boldsymbol{c_2} = \begin{bmatrix} c \\ d \end{bmatrix}, \in \mathbb{R}^2$$

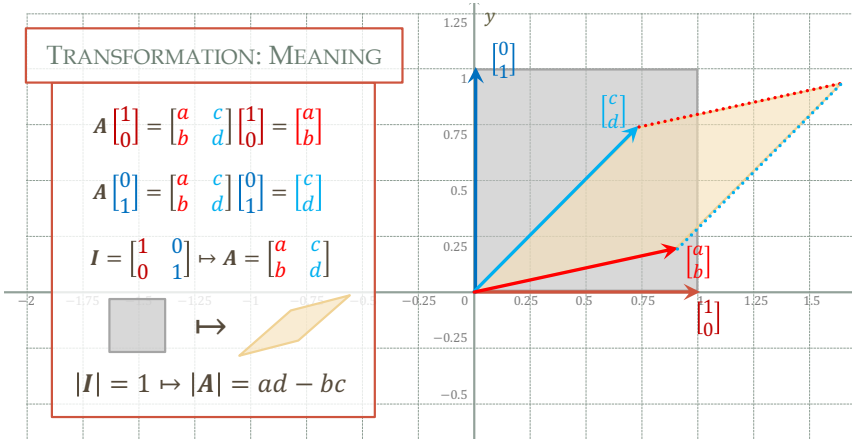


Fig. 3.1 The matrix $\boldsymbol{A}$ transforms the unit vectors to its columns, $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$, thus transforming the unit square to a parallelogram.

How does $\boldsymbol{A}$ transform the unit vectors? (The i^{th} unit vector is the i^{th} column of the identity matrix $\boldsymbol{I}$). The transformed versions are just $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$, the columns of $\boldsymbol{A}$.

$$\boldsymbol{A}\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}a & c\\b & d\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}a\\b\end{bmatrix} = \boldsymbol{c_1} \quad \boldsymbol{A}\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}a & c\\b & d\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}c\\d\end{bmatrix} = \boldsymbol{c_2}$$

It is easiest to use the column picture of matrix multiplication (§2.7.4, page 39) to understand this fact: The first product above is a trivial linear combination of the columns of $\boldsymbol{A}$, taking one of the first column and zero of the second column ($1 \times \boldsymbol{c_1} + 0 \times \boldsymbol{c_2}$), giving us the first column $\boldsymbol{c_1}$ of $\boldsymbol{A}$ back. Similarly the second unit vector transforms to the second column $\boldsymbol{c_2}$ of $\boldsymbol{A}$.

As we can see in Figures 3.1 and 3.2, the transformation that $\boldsymbol{A}$ performs on the unit square with vertices at $(0,0), (1,0), (1,1)$ and $(0,1)$ takes it to a parallelogram with vertices at $(0,0), (a,b), (a+c, b+d)$ and (c,d). And the area of this parallelogram, as Figure 3.2 proves without words[1], is indeed the determinant as defined in Eqn (3.2). In $\mathbb{R}^n$, instead of the area, we get the (signed) volume of hyper-parallelepiped. Note that we have proved only the absolute value of

[1] The proof is recreated from Mathematics StackExchange, attributed to Solomon W. Golomb. If the geometrical version of the proof is hard to digest, there is an algebraic version as well.

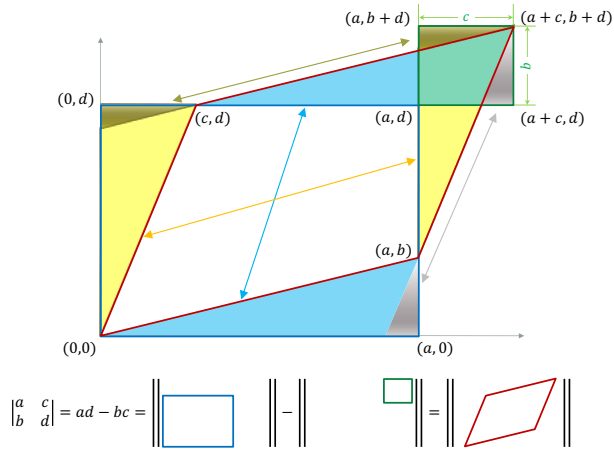


Fig. 3.2 The parallelogram that results from the transformation of the unit square by the action of $\boldsymbol{A}$. The picture proves, without words, that its area is the same as $|\boldsymbol{A}|$.

the area in Figure 3.2; we will learn more about the sign part in the following examples.

In Figure 3.3, we have three different examples of $\boldsymbol{A}$ in $\mathbb{R}^2$ and their determinants. In the left panel, we see how $\boldsymbol{A}$ transforms the red and blue unit vectors (which have unit elements in the first and second dimension respectively). The red unit vector gets transformed to the first column of $\boldsymbol{A}$ (shown in a lighter shade of red), and the blue one to the second column (light blue vector). If we complete the parallelogram, its area is 2, which is the determinant, $|\boldsymbol{A}|$.

In the middle panel of Figure 3.3, we have a different $\boldsymbol{A}$, which does something strange to the red and blue unit vectors: They get transformed to a line, which is a collapsed parallelogram with zero area. And by the definition of $|\boldsymbol{A}|$, it is indeed zero. Looking at the column vectors in $\boldsymbol{A}$, we can see that they are scalar multiples of each other; they are not linearly independent.

In the right panel of the same figure, we have shuffled the columns of $\boldsymbol{A}$, so that parallelogram is the same as in the left panel, but the determinant now is negative. We, therefore, call the determinant the *signed* area of the parallelogram formed with the column vectors as adjacent sides. Why is the area negative in this case? It is because $\boldsymbol{A}$ has flipped the order of the transformed vectors: Our blue unit vector is to the left of the red one. In the left panel, the transformed blue

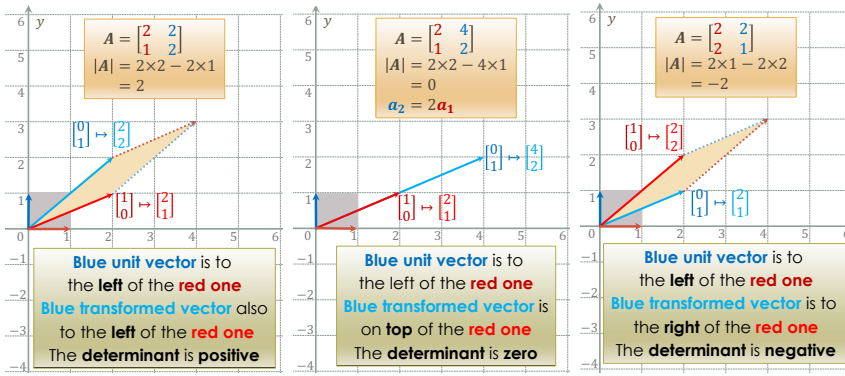


Fig. 3.3 Determinants as areas: The matrix $\boldsymbol{A}$ transforms the unit square (shaded grey) into the amber parallelogram. The determinant $|\boldsymbol{A}|$ is the signed area of this parallelogram. The sign is negative when the transformed unit vectors "flip."

vector is still to the left. But in the right panel, the blue one has gone to the *right* of the red one after transformation, thereby attributing a negative sign to the determinant.

To use a more formal language, to go from the first (red) unit vector to the second (blue) one, we go in the counterclockwise direction. The direction is the same for the transformed versions in the left panel of Figure 3.3, in which case the determinant is positive. So is the area. In the right panel, the direction for the transformed vectors is clockwise, opposite of the unit vectors. In this case, the determinant and the signed area are negative.

3.3.2 3×3 Matrices

We have defined the determinant of a 2×2 matrix in Eqn (3.2). We now extend it to higher dimensions recursively. For $\boldsymbol{A} = \begin{bmatrix} a_{ij} \end{bmatrix} \in \mathbb{R}^{3\times 3}$, we have:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \in \mathbb{R}^{3\times 3} \tag{3.3}$$

$$|\boldsymbol{A}| \overset{\text{def}}{=} a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \in \mathbb{R}$$

Since each of the 2×2 determinants in Eqn (3.3) are defined in Eqn (3.2), we have a recursive formula for the determinant of $\boldsymbol{A} \in$

$\mathbb{R}^{3\times 3}$. Notice that the first term has a positive sign, the second one a negative sign and the third a positive sign again. This pattern of alternating signs extends to higher dimensions as well.

Minors and Cofactors: The first submatrix whose determinant appears in Eqn (3.3) (multiplying a_{11}) is obtained by removing the first row and column from $\boldsymbol{A}$. Generalizing, the one multiplying a_{ij} is the determinant of the submatrix obtained by removing the i^{th} row and the j^{th} column. The determinants of such submatrices are called the *minors* of $\boldsymbol{A}$, denoted by M_{ij}.

The minor with the associated sign is called the cofactor, $C_{ij} = -1^{i+j}M_{ij}$. What we did in Eqn (3.3) was to expand the determinant along the first row. We could have done it along the first column as well. In fact, we can compute the determinant by expanding along *any* row or column and summing up the cofactors. With these definitions of minors and cofactors, we rewrite Eqn (3.3) more compactly as follows (where we are expanding $|\boldsymbol{A}|$ along the i^{th} row):

$$|\boldsymbol{A}| = \sum_{j=1}^{3}(-1)^{i+j}a_{ij}M_{ij} = \sum_{j=1}^{3} a_{ij}C_{ij} \tag{3.4}$$

Determinant as Volume: For 2×2 matrices, we saw that the determinant was the (signed) area of the parallelogram to which the unit square transformed. In the 3×3 case, it becomes the volume of the parallelepiped that is the transformed version of the unit cube. It is also signed: If two unit vectors flip orientation when transformed, the determinant gets multiplied by -1, which means if one more unit vector flips, the sign reverts back to the original.

3.3.3 $n \times n$ Matrices

We can extend the notion of volume to $\mathbb{R}^n$. If we think of $\boldsymbol{A}$ as being composed of n column vectors,

$$\boldsymbol{A} = \begin{bmatrix} | & | & & | \\ \boldsymbol{c_1} & \boldsymbol{c_2} & \cdots & \boldsymbol{c_n} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n\times n} \quad \boldsymbol{c_i} \in \mathbb{R}^n$$

the determinant, $|\boldsymbol{A}|$, is the *signed volume* of the n-dimensional parallelepiped formed with edges $\boldsymbol{c_i}$.

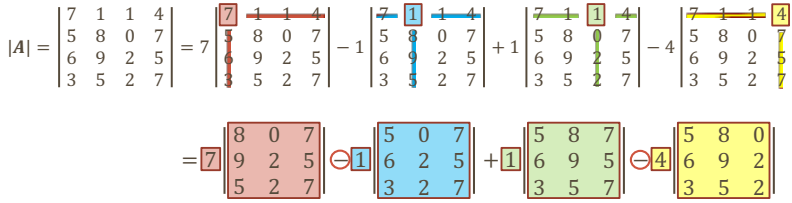


Fig. 3.4 Illustration of minors and cofactors using an example 4×4 matrix. The minor is the determinant of the submatrix obtained by removing the row and column corresponding to each element, as shown. Notice the sign in the summation. The minor with the associated sign is the cofactor.

Laplace Formula: What we wrote down in Eqn (3.4) is, in fact, the general version of the recursive formula for computing the determinant, expanding over the i^{th} row.

$$|\boldsymbol{A}| = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} M_{ij} = \sum_{j=1}^{n} a_{ij} C_{ij} \tag{3.5}$$

This is the Laplace formula (or expansion). We could do the expansion over the j^{th} column as well.

$$|\boldsymbol{A}| = \sum_{i=1}^{n} (-1)^{i+j} a_{ij} M_{ij} = \sum_{i=1}^{n} a_{ij} C_{ij}$$

In fact, one of the properties of determinants is that they are invariant under row-column transposition, which is to say, when taking the transpose of the matrix. In other words $|\boldsymbol{A}| = \left|\boldsymbol{A}^{\mathsf{T}}\right|$.

3.3.4 Properties of Determinants

We went over some of the characteristics and properties of determinants. Let's list them all here for completeness.

1. Identity matrices (of any dimension, $\boldsymbol{I} \in \mathbb{R}^{n \times n}$) have a determinant of one.

   $$|\boldsymbol{I}| = 1$$

   This property is consistent with the fact that a hypercube of unit sides has unit (hyper)volume.

2. If we exchange two rows (or two columns) the determinant changes sign. (As a consequence, it should be impossible to get back to the same matrix by performing an odd number of row or column exchanges.)

3. If we multiply a row (or a column) by a scalar s, the determinant gets multiplied by the same factor:

$$\boldsymbol{A} = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \quad \boldsymbol{A}_1 = \begin{bmatrix} sa & sc \\ b & d \end{bmatrix} \quad \boldsymbol{A}_2 = \begin{bmatrix} sa & c \\ sb & d \end{bmatrix}$$
$$\implies |\boldsymbol{A}_1| = |\boldsymbol{A}_2| = s\,|\boldsymbol{A}|$$

   Since determinants behave like volumes, we can see that if we double one side of a hypercube or a parallelepiped, its volume gets doubled. If we double *all* sides, then its volume gets multiplied by 2^n.

4. If we can express all elements in one row (or a column) as sums of two numbers each, we can split the whole determinant as sum of two determinants.

$$\begin{vmatrix} a+a' & c+c' \\ b & d \end{vmatrix} = \begin{vmatrix} a & c \\ b & d \end{vmatrix} + \begin{vmatrix} a' & c' \\ b & d \end{vmatrix}$$

5. If one row (or columns) is all zeros, then the determinant is zero. This statement should be self-evident from the definition of determinants (Eqn (3.5)): We expand the Laplace formula over the row or column with all zeros to get $|\boldsymbol{A}| = 0$.

6. The determinant does not change if we add or subtract any multiple of one row (or column) from any other row (or column). While this property may sound a bit mysterious, it corresponds to the fact that the matrix equations (such as $\boldsymbol{Ax} = \boldsymbol{b}$) actually encode systems of linear equations, which we can add and subtract without affecting their solvability or solutions.

7. If two rows (or columns) of the matrix are the same, the determinant is zero, which is a corollary of the previous two properties.

8. For a triangular matrix (*e.g.*, upper triangular matrix, $\boldsymbol{U} : u_{ij} = 0$ for $i > j$), the determinant is the product of the diagonal

Deriving $|A|$

It is possible to start from the properties (listed in §3.3.4, page 58) and derive the formula for the determinant for a 2×2 matrix, which is an interesting exercise.

- Start with the identity matrix in $\mathbb{R}^2$. By Property (1), we have: $\begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} = 1$
- Property (3): Scale the first row by a, the determinant scales by a: $\begin{vmatrix} a & 0 \\ 0 & 1 \end{vmatrix} = a$
- Using the same property again, this time to scale the second row by d,

$$\begin{vmatrix} a & 0 \\ 0 & d \end{vmatrix} = ad \tag{3.6}$$

- Similarly, by multiplying the rows of $\boldsymbol{I}$ by b and c, we get: $\begin{vmatrix} b & 0 \\ 0 & c \end{vmatrix} = bc$
- Property (2): Swap the rows, the determinant changes sign, which means:

$$\begin{vmatrix} 0 & c \\ b & 0 \end{vmatrix} = - \begin{vmatrix} b & 0 \\ 0 & c \end{vmatrix} = -bc \tag{3.7}$$

- Property (4): Express the elements of the first row as sums, we can write:

$$\begin{vmatrix} a & c \\ b & d \end{vmatrix} = \begin{vmatrix} a+0 & 0+c \\ b & d \end{vmatrix} = \begin{vmatrix} a & 0 \\ b & d \end{vmatrix} + \begin{vmatrix} 0 & c \\ b & d \end{vmatrix} \tag{3.8}$$

- We can express the first term in Eqn (3.8), again by using Property (4) as:

$$\begin{vmatrix} a & 0 \\ b & d \end{vmatrix} = \begin{vmatrix} a & 0 \\ 0+b & d+0 \end{vmatrix} = \begin{vmatrix} a & 0 \\ 0 & d \end{vmatrix} + \begin{vmatrix} a & 0 \\ b & 0 \end{vmatrix} = \begin{vmatrix} a & 0 \\ 0 & d \end{vmatrix}$$

  The second determinant is zero by Property (5) because it contains a column of zeros.
- Similarly, the second term becomes: $\begin{vmatrix} 0 & c \\ b & d \end{vmatrix} = \begin{vmatrix} 0 & c \\ b & 0 \end{vmatrix}$
- Therefore we get, using Equations (3.6) and (3.7) in Eqn (3.8),

$$\begin{vmatrix} a & c \\ b & d \end{vmatrix} = \begin{vmatrix} a & 0 \\ 0 & d \end{vmatrix} + \begin{vmatrix} 0 & c \\ b & 0 \end{vmatrix} = ad - bc$$

What this derivation is telling us is that the properties of the determinant are not merely a consequence of its definition, but also its origin. In other words, if we are looking for a number associated with a matrix with the specified set of properties, the determinant turns out to be that number.

elements. We can appreciate the veracity of this property by expanding the determinant over the first column for an upper triangular matrix, or the first row of a lower triangular matrix.

9. If the determinant is zero, the matrix cannot be inverted. We have not defined or discussed matrix inversion, but the inverse of a matrix is something that reverses its transformation. If

$\boldsymbol{Ax} = \boldsymbol{b}$, and $\boldsymbol{x} = \boldsymbol{A^{-1}b}$, we call $\boldsymbol{A^{-1}}$ the inverse of $\boldsymbol{A}$. What this property says is that if $|\boldsymbol{A}| = 0$, we cannot find $\boldsymbol{A^{-1}}$. As we saw earlier, such noninvertible matrices are called singular matrices, and have deep implications in much of what we will learn form now on.

10. The determinant of the product of two matrices (of the same size) is the product of their determinants: $|\boldsymbol{AB}| = |\boldsymbol{A}|\,|\boldsymbol{B}|$.

11. The determinant of the transpose of a matrix is the same as that of the matrix. We could have used this last property to avoid specifying "or column" in most of the properties above.

3.4 Numerical Computations

This first part of the book (comprising Chapters 1, 2 and 3) was meant to be about numerical computations involving vectors and matrices. The moment we start speaking of vectors, however, we are already thinking in geometrical terms. In this chapter, we also saw how determinants had a geometric meaning as well.

In the first chapter, in order to provide a motivation for the idea of matrices, we introduced linear equations, which is what we will expand on, in the next part on the algebraic view of Linear Algebra. We will go deeper into systems of linear equations. While discussing the properties of determinants, we hinted at their connection with linear equations again.

As we can see, although we might want to keep the algebra and geometry separated, it may not be possible (nor is it perhaps advisable) to do so because we are dealing with different views of the same subject of Linear Algebra.

Part II

Algebraic View

4

Gaussian Elimination

Confidence is what you have before you understand the problem.

—Woody Allen

If we have a system of linear equations, we may or may not have solutions, we may have a unique solution or an infinite number of solutions. As we will see very soon, it is not easy to determine the solvability of a system even for relatively small number of equations. We can, of course, perform a series of manipulations on the equations (like adding them, eliminating variables, substituting the solved ones back etc.) to arrive at their solutions. How do we get a computer to solve the system of equations though? We need an algorithm, both to determine the solvability conditions and to actually find the solutions. Such an algorithm is called *Gaussian elimination* or *row reduction*.

4.1 Solvability of System of Linear Equations

Let's start with a simple system of two linear equations, and see what the problem really is. Here, we will have only two variables, x and

Table 4.1 Various permutations of simultaneous equations in two variables, showing solvability

	Equations	Solution	Comment
1	$x + y = 5$ $x - y = 1$	$x = 3$ $y = 2$	Unique solution
2	$x + y = 5$	$x = t$ $y = 5 - t$	Infinity of solutions Only one equation
3	$x + y = 5$ $2x + 2y = 10$	$x = t$ $y = 5 - t$	Infinity of solutions Really, only one equation
4	$x + y = 5$ $x + y = 1$	–	No solutions Inconsistent equations
5	$x + y = 5$ $x - y = 1$ $3x - y = 7$	$x = 3$ $y = 2$	Unique solution Really only two equations
6	$x + y = 5$ $x - y = 1$ $3x - y = 9$	–	No solutions Inconsistent equations

y. We are told that if we have two equations and two unknowns, we can solve them, as though the equality of the number of variables to the number of equations is the necessary and sufficient condition for solvability. However, as Table 4.1 shows, it is only part of the story, and not always true.

In the first row of the table, we see the ideal case: We have two good equations for our two unknowns and we can easily find their solutions.

In the second row, we have too few equations. What happens is not that we do not have a solution. Any x and y satisfying the one equation we have is a solution, and we have infinitely many of them.

In the third row of Table 4.1, we do have two equations, but the second equation is *derived* from the first, and is therefore not *independent*. In effect, we have only one equation and an infinite number of solutions, as in the second row.

In the fourth row, we have two equations, but they are not *consistent* with each other. They both cannot be true at the same time for any pair of values of x and y.

Things get complicated in the fifth row, where we seem to have three equations. But the third one can be derived from the first two. It is, in fact, Eq.1 + 2×Eq.2, which means, in reality, we only have two good equations for two unknowns. We therefore get a good solution, much like the first row of Table 4.1.

The sixth row looks similar to the fifth, but the third equation there is different on the right hand side. It cannot be derived from the other two, and is inconsistent with them. Therefore, we have no solutions.

In light of these results, we can state the solvability condition, in a general case as follows: If we have a system of n *independent* and *consistent* linear equations on n unknowns, we can find a *unique* solution for them. We are yet to define the concepts of independence and consistency though.

Independence

Definition: An equation in a system of linear equations is considered independent if it cannot be derived from the rest using algebraic manipulations.

If we multiply an equation with a scalar, or add two equations, the new equation we get is *not* independent. Again, notice the similarity of the dependence of equations with our requirements for linearity (§1.1.3, page 15).

Consistency

Definition: An inconsistent system of linear equations is the one with no solutions.

The concept of consistency is harder to pin down. For now, we are defining it rather cyclically as in the statement above. It is possible, however, to visualize why some equations are inconsistent with others. In fourth row of Table 4.1, for instance, the lines described by the two equations are parallel to each other.

4.1.1 Visualizing Equations

In Table 4.1, our equations represent line because we have only two variables (x and y) and we are dealing with $\mathbb{R}^2$, which is a plane.

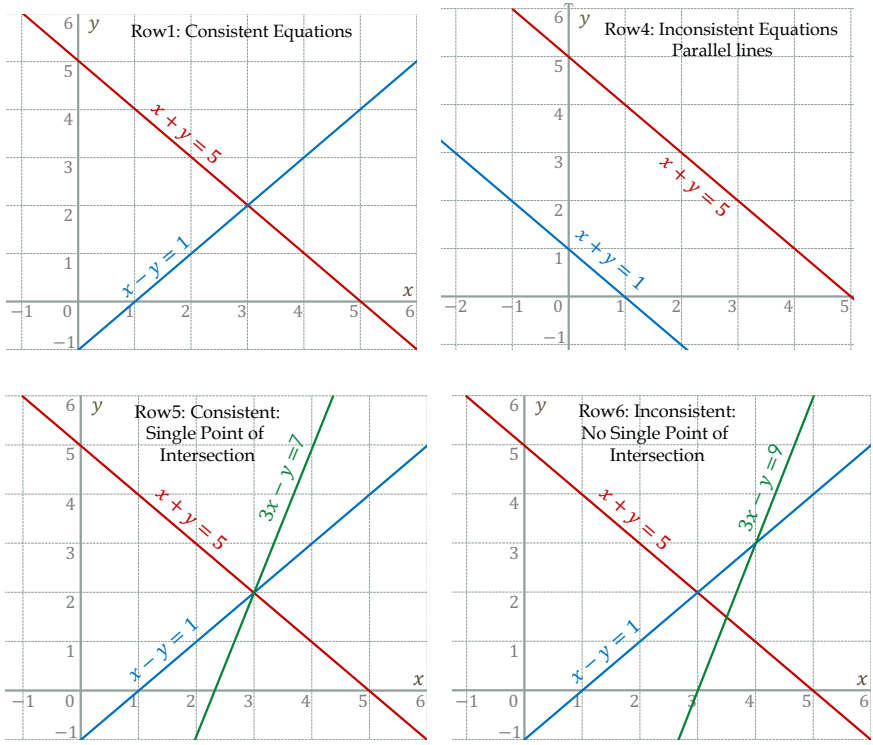


Fig. 4.1 Visualizing equations listed in Table 4.1. Clockwise from top-left: Row 1, 4, 5 and 6 in the table.

In Figure 4.1, we can see the lines corresponding to the equations in the various rows in the table. For the first row, we have two good (independent and consistent) equations, corresponding to the red and blue lines in the top-left panel. This system has nice solution, where the two lines intersect, at $x = 3, y = 2$. When people say "as many equations as unknowns," what they have in mind is this type of independent and consistent system.

In the second row of Table 4.1, we have only one equation, which corresponds to the red one in all panels of Figure 4.1. Every point on the line is a solution to the equation, which goes for the third row too.

The fourth row of the table is shown in the top-right panel of Figure 4.1 as two parallel lines, which will never meet. No point in the red line will ever be a point in the blue line, and the equations do not have a solution, which is why they are inconsistent.

In the fifth row of Table 4.1, we have an extra third equation, shown as a green line in the bottom-left panel of Figure 4.1. It is consistent with the other two equations because it goes through the same solution point $x = 3, y = 2$. The equations are, in fact, not independent, any one of them can be derived from the other two. The three equations are consistent with each other and any two of them would be enough to fully solve the system.

The sixth row shows the case where the third equation is not consistent. Note that in this case, we do not have parallel lines, but three lines not having a common point (which would have been the solution). For future use, it may be interesting to wonder what an approximate, or *best possible*, solution would be. Is the centroid of the triangle formed by the three lines, perhaps, a good candidate as the best possible solution?

4.1.2 Generalizing to $\mathbb{R}^n$

The representation of the one-dimensional space $\mathbb{R}$, as we learn in school, is the number line. An equation, such as $x = 1$ or, in general, $ax = b$ defines a point on this line.

When we move to $\mathbb{R}^2$, the same equation, $x = 1$ defines a vertical line. In general, however, we have an equation, $a_1x + a_2y = b$, which defines a line with a slope. If we have two such equations, then we get two lines, with the possibility of them intersecting and giving us a solution to the system of the two equations.

Similarly, in $\mathbb{R}^3$, a single linear equation, such as $x = 1$ defines a plane. It is a plane parallel to the yz plane, at unit distance from it. $x = 0$ would be the yz plane. A general equation of the form $a_1x + a_2y + a_3z = b$ gives us a plane with some orientation. Another linear equation defines another plane. If the equations are consistent and independent, the planes will intersect, giving us the solution, which is a line. Since the solution is a line, we have an infinity of solutions with two linear equations in $\mathbb{R}^3$. If we have one more linear equation, we have another plane, potentially intersecting this solution line at a point, giving us a unique solution.

With this picture in mind, let's list the behavior of systems of linear equations and their solutions in $\mathbb{R}^2$, $\mathbb{R}^3$ and extrapolate to $\mathbb{R}^n$, as in Table 4.2.

Summarizing the insights from Table 4.2,

Table 4.2 Properties and behavior of linear equations and solutions

Equations	$\mathbb{R}^2$	$\mathbb{R}^3$	$\mathbb{R}^n : n > 3$
One equation	A line Infinite solutions	A plane Infinite solutions	An $n-1$ subspace Infinite solutions
Two independent and consistent equations	A point of intersection Unique solution	A line of intersection Infinite solutions	An $n-2$ subspace as intersection. ∞ solutions
Three independent and consistent equations	Cannot happen	A point of intersection Unique solution	An $n-3$ subspace as intersection. ∞ solutions
n independent and consistent equations	Cannot happen	Cannot happen	A point of intersection. Unique solution
Two independent, but inconsistent equations	Parallel lines No intersection No solutions	Parallel planes No intersection No solutions	Parallel subspaces No intersection No solutions
Three independent, but inconsistent equations	Lines that make $\|\|\|, \nparallel, \triangle$[1] No solutions	Planes making $\|\|\|, \nparallel, \triangle$ in cross section[2] No solutions	Hard to visualize No common intersection No solutions
n independent, but inconsistent equations	Lines that make $\|\|\|, \nparallel, \triangle$[3] No solutions	Planes making $\|\|\|, \nparallel, \triangle$ in cross section[4] No solutions	Hard to visualize No common intersection No solutions

[1,3] These symbols represent three parallel lines, two parallel lines plus an intersecting line, or three lines making a triangle.
[2,4] These symbols represent three planes, which, when sliced perpendicularly by a plane, show up as three parallel lines, two parallel lines plus an intersecting line, or three lines making a triangle. In other words, the three planes are parallel, two parallel with one intersecting or the three making a triangular tube respectively.

- The number of *independent* equations in a system of *consistent* linear equations can never be greater than the number of unknowns.
- If a system of linear equations has more *independent* equations than unknowns, it is necessarily *inconsistent* and has no solutions.

As we can see, the behavior of systems of equations is much more complicated than what we learned in highschool. Furthermore, it looks hard to generalize to higher dimensions, and harder still to understand, how a given system is going to behave. Imagine if we have 50 unknowns and 100 equations–a relatively small system that we may come across in our computer/data science career. Do they have solutions? Are the equations independent? Consistent? Are there too many? Or too few? We clearly need a systematic way, an algorithm, to tell us these things.

4.2 Gaussian Elimination

The critical step in the process to determine the solvability and to actually solve systems of linear equations is *Gaussian Elimination*,

also known as *Row Reduction*. Gaussian elimination is the algorithm to transform a matrix by applying a series of *elementary row operations* to arrive at its *Row-Echelon Form* (REF). When working with systems of linear equations and their solutions, we apply Gaussian elimination on the *augmented matrix* of the system. (We will soon define the terms in *italics*.) Gaussian elimination, however, has applications other than solving equations, such as computing determinants, finding inverses, determining ranks of matrices, and so on.

4.2.1 Matrix as System of Linear Equations

The first step in applying Gaussian elimination is to cast our linear equations in a compact matrix form. We, in fact, did this in Chapter 1 (§1.1.4, page 15), when we introduced vectors and matrices. Let's do it again, looking at the deceptively simple equation, $\boldsymbol{Ax} = \boldsymbol{b}$, that will become the mainstay of our discussion for the rest of the book. Writing it out explicitly:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & a_{ij} & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m\times n} \text{ and } \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n, \tag{4.1}$$

$$\boldsymbol{Ax} = \boldsymbol{b}, \quad \text{where} \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \in \mathbb{R}^m$$

Now that we know matrix multiplication, we can see that $\boldsymbol{Ax} = \boldsymbol{b}$ represents a system of m linear equations of the kind

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i \text{ where } 1 \leq i \leq m$$

on n unknowns, $x_j, 1 \leq j \leq n$.

On these equations, we can perform algebraic operations, like adding or subtracting them, multiplying by a scalar, etc. In fact, we can perform operations similar to taking linear combinations, described in §2.2.3 (page 25). Notice the similarity between these algebraic manipulations of equations and some of the properties of determinants (§3.3.4, page 58)? We will make use of this similarity in using Gaussian elimination for determinant calculation.

4.2.2 Elementary Row Operations

Analogous to the algebraic manipulation of equations, let's first define a set of *elementary row operations*, as listed below:

- Swap any two rows.
- Multiply any row by a scalar.
- Add a multiple of any row to another.

We will use these row operations to build an algorithm to solve the system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, much the same way we would use the corresponding algebraic operations to solve the equations symbolically. The advantage in the matrix formulation is that we are dealing with numbers, and we can program a computer to perform the operations once the algorithm is ready.

4.2.3 Augmented Matrix

We have to keep in mind that what we are doing when performing row operation on $\boldsymbol{A}$ is basically the same as algebraic manipulation of equations. Therefore, we have to apply the same operation to the right hand side, $\boldsymbol{b}$ as well. For this reason, it may be best to add an extra column to $\boldsymbol{A}$ with the elements of $\boldsymbol{b}$. Such a matrix is called the *augmented matrix*. It is merely a convenient bookkeeping technique, which turns out to be useful when implementing row operations in a computer program.

$$\left[\boldsymbol{A} \mid \boldsymbol{b}\right] = \left[\begin{array}{ccc|c} a_{11} & \cdots & a_{1n} & b_1 \\ a_{21} & \cdots & a_{2n} & b_2 \\ \vdots & a_{ij} & \vdots & b_i \\ a_{m1} & \cdots & a_{mn} & b_m \end{array}\right] \tag{4.2}$$

Now that we are naming matrices, let's also call $\boldsymbol{A}$ (either as part of $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$ or by itself) the *coefficient matrix*, and the $\boldsymbol{b}$ part the *constant matrix* or *constant vector*.

Before describing the algorithm of Gaussian elimination, let's look at the endpoint of the algorithm, which is the form in which we would like to have our matrix $\boldsymbol{A}$ or $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$.

Row-Echelon form

Definition: A matrix is considered to be in its row-echelon form (REF) if it satisfies the following two conditions:

1. All rows with zero elements are at the bottom of the matrix.

2. The *first nonzero element* in any row is strictly to the right of the first nonzero element in the row above it.

Pivots

Definition: The leading nonzero element in a row of a matrix in its row-echelon form is called a pivot. The corresponding column is called the pivot column. Pivots are also called the *leading coefficient*.

The largest number of pivots a matrix can have is the smaller of its dimensions (numbers of rows and columns). In other words, for $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, the largest number of pivots would be $\min(m, n)$.

Rank

Definition: The number of pivots of a matrix (in its REF) is its rank. We will have a better definition of rank later on. If a matrix has its largest possible rank (which is the largest possible number of pivots, $\min(m, n)$) is called a *full-rank* matrix. If a matrix is not full rank, we call it rank deficient, and its rank deficiency is $\min(m, n) - \text{rank}(\boldsymbol{A})$.

We have a few examples of matrices in their REF in Eqn (4.3) below, where the pivots are shown in **bold**. The first matrix shows a square matrix of size 4×4, and it has four pivots, and is therefore full rank. The second matrix is 2×3, and has two pivots–the largest possible number. It is also full-rank. The third one is a 4×4 matrix, but has only three pivots. It is rank deficient by one. The fourth matrix also has a rank deficiency of one.

$$\begin{bmatrix} \mathbf{5} & 2 & 11 & 3 \\ 0 & \mathbf{3} & 7 & 13 \\ 0 & 0 & \mathbf{17} & 2 \\ 0 & 0 & 0 & \mathbf{13} \end{bmatrix} \quad \begin{bmatrix} \mathbf{5} & 11 & 3 \\ 0 & 0 & \mathbf{13} \end{bmatrix} \quad \begin{bmatrix} \mathbf{5} & 2 & 11 & 3 \\ 0 & 0 & \mathbf{17} & 2 \\ 0 & 0 & 0 & \mathbf{13} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} \mathbf{5} & 2 & 11 & 3 \\ 0 & 0 & \mathbf{17} & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.3}$$

4.2.4 The Algorithm

With the definitions of REF, pivots and ranks in place, we can state the Gaussian elimination algorithm, running on a matrix $\boldsymbol{A} = \left[a_{ij}\right] \in \mathbb{R}^{m\times n}$ as follows:

Gaussian Elimination: Algorithm

Since this is a book aimed at computer and data scientists, it is probably worth our time stating the algorithm of Gaussian Elimination as an algorithm.

Input: $\boldsymbol{A} = \left[a_{ij}\right] \in \mathbb{R}^{m \times n}$
Output: RRF$(\boldsymbol{A}) \in \mathbb{R}^{m \times n}$

1: REF $\leftarrow \boldsymbol{A}$; nrows $\leftarrow m$
2: **while** $m > 0$ **and** $n > 0$ **do**
If the first row does not start with a nonzero element, try to find a row that does:
3: **if** $a_{11} = 0$ **then**
4: **repeat**
5: **if** $a_{11} = 0$ **then**
6: **for** $i = 1$ to m **do**
7: **if** $a_{i1} \neq 0$ **then**
Swap row 1 with row i:
8: $r_1 \leftrightarrow r_i$
9: Exit **repeat** loop
10: **end if**
11: **end for**
Could not find a row with nonzero element in the first column
12: $\boldsymbol{A} \leftarrow \boldsymbol{A}\left[a_{22} : a_{mn}\right], m \leftarrow m-1, n \leftarrow n-1$
13: **end if**
14: **until** $a_{11} \neq 0$
15: **else**
Subtract scaled first row from other rows to get zero first column
16: **for** $i = 2$ **to** m **do**
17: $r_i \leftarrow r_i - \frac{a_{i1}}{a_{11}} r_1$
18: **end for**
19: **end if**
Save the current row in the REF
20: REF(nrows$-m$) $\leftarrow \boldsymbol{A}\left[a_{11} : a_{1n}\right]$
21: $\boldsymbol{A} \leftarrow \boldsymbol{A}\left[a_{22} : a_{mn}\right], m \leftarrow m-1, n \leftarrow n-1$
22: **end while**
23: **return** REF

Note that instead of finding the first row with nonzero element (starting at line 5), it may be a better numerical strategy to unconditionally locate the row with the largest absolute value at the first element (starting at line 3), and swapping it with the first row. Most programs implement Gaussian Elimination that way.

1. If $a_{11} = 0$, loop over the rows of $\boldsymbol{A}$ to find the row that has a nonzero element in the first column.

2. If found, swap it with the first row. If not, ignore the first row and column and move on to the second row (calling it the first).

3. Multiply the first row with $-\frac{a_{i1}}{a_{11}}$ and add it to the i^{th} row to get zeros in the first column of all rows other than the first one.

4. Now consider the submatrix from the second row, second column to the last row, last element (*i.e.*, from a_{22} to a_{mn}) as the new matrix:
$$\boldsymbol{A} \leftarrow \boldsymbol{A}[a_{22} : a_{mn}], m \leftarrow m - 1, n \leftarrow n - 1$$

5. Loop back to step 1 and perform all the steps until all rows or columns are exhausted.

Note that in step 3, we are subtracting a number after dividing it by a_{11}, which may cause numerical instability in the algorithm when $a_{11} \to 0$. For this reason, we may want to modify the first step to find the row with the largest absolute value in the first column. (In other words, modify the first step to read, "Loop over the rows of $\boldsymbol{A}$ to find the row that has the *largest absolute value* ($\neq 0$) in the first column," which is the way Gaussian elimination is implemented in most programs, including **`SageMath`**.)

4.3 Applications of Gaussian Elimination

Although the primary objective of Gaussian elimination is to solve systems of linear equations, we also use it for a couple of other purposes.

We saw that the rank of a matrix is the same as the number of pivots, which we can get directly from Gaussian elimination (AKA row reduction, as a reminder).

At the end of Gaussian elimination, we have an upper triangular matrix. If we start with a square matrix, whose determinant we need to compute, we can get it (or its absolute value) by simply taking the product of the diagonal elements (which are the pivots) because the elementary row operations we perform in Gaussian elimination do not change the absolute value of the determinant.

For these applications, we can also perform Gaussian elimination by column (as column reduction) instead of row. However, for the main application of solving a system of linear equations, we can do it only row-wise. Besides, column-wise Gaussian elimination is the same as row reduction on the transpose of the matrix anyway.

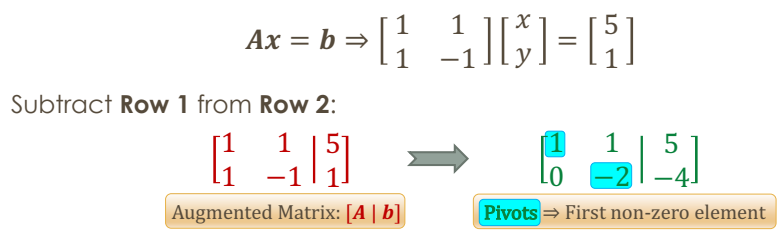


Fig. 4.2 Gaussian elimination on a simple augmented matrix, showing the pivots.

4.3.1 Solution to System of Linear Equations

To solve a system of linear equations, we apply the algorithm of Gaussian elimination to its augmented matrix. Let's first look at a simple example from the first row in Table 4.1. As shown in Figure 4.2, the system of linear equations, $x + y = 5, x - y = 1$, translates to an augmented matrix $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$ with two rows, and single step Gaussian elimination, after which, we get two pivots, 1 and -2. Although it is a simple system, we can make a few observations about it:

- The rank of the 2×2 matrix $\boldsymbol{A}$ is two, and it is full-rank matrix.
- Since the coefficient matrix ($\boldsymbol{A}$) is square and full rank, we can infer that the system has a unique solution.
- The determinant, $|\boldsymbol{A}|$ is the product of the pivots $= -2$. It can also be calculated directly: $|\boldsymbol{A}| = a_{11}a_{22} - a_{21}a_{12} = (1 \times -1) - (1 \times 1) = -2$.
- The last row of the REF form of $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$ stands for the equation $-2y = -4$, which gives us $y = 2$.
- We can *back-substitute* this value of y in the second last row (which is the first row) to solve for the other variable, giving us $x + 2 = 5 \implies x = 3$.

Back Substitution

Definition: The process of solving a system of linear equations from the REF of the augmented matrix of the system is known as back substitution. The equation corresponding to the last nonzero row is

Table 4.3 Illustration of solvability conditions based on the characteristics of REF

	Equations	$[\boldsymbol{A} \mid \boldsymbol{b}]$	REF($[\boldsymbol{A} \mid \boldsymbol{b}]$)	Observations[1]
1	$x + y = 5$ $x - y = 1$	$\left[\begin{array}{cc\|c} 1 & 1 & 5 \\ 1 & -1 & 1 \end{array}\right]$	$\left[\begin{array}{cc\|c} 1 & 1 & 5 \\ 0 & -2 & -4 \end{array}\right]$	REF has no $0 = b_i \implies$ Solvable Rank = # Vars $\implies$ Unique Solution
2	$x + y = 5$	$\left[\begin{array}{cc\|c} 1 & 1 & 5 \end{array}\right]$	$\left[\begin{array}{cc\|c} 1 & 1 & 5 \end{array}\right]$	REF has no $0 = b_i \implies$ Solvable Rank < # Vars $\implies$ Infinity of Solns
3	$x + y = 5$ $2x + 2y = 10$	$\left[\begin{array}{cc\|c} 1 & 1 & 5 \\ 2 & 2 & 10 \end{array}\right]$	$\left[\begin{array}{cc\|c} 1 & 1 & 5 \\ 0 & 0 & 0 \end{array}\right]$	REF has no $0 = b_i \implies$ Solvable Rank < # Vars $\implies$ Infinity of Solns
4	$x + y = 5$ $x + y = 6$	$\left[\begin{array}{cc\|c} 1 & 1 & 5 \\ 1 & 1 & 6 \end{array}\right]$	$\left[\begin{array}{cc\|c} 1 & 1 & 5 \\ 0 & 0 & 1 \end{array}\right]$	REF has $0 = b_i \implies$ Inconsistency Not Solvable
5	$x + y = 5$ $x - y = 1$ $3x - y = 7$	$\left[\begin{array}{cc\|c} 1 & 1 & 5 \\ 1 & -1 & 1 \\ 3 & -1 & 7 \end{array}\right]$	$\left[\begin{array}{cc\|c} 1 & 1 & 5 \\ 0 & -2 & -4 \\ 0 & 0 & 0 \end{array}\right]$	REF has no $0 = b_i \implies$ Solvable Rank = # Vars $\implies$ Unique Solution
6	$x + y = 5$ $x - y = 1$ $3x - y = 9$	$\left[\begin{array}{cc\|c} 1 & 1 & 5 \\ 1 & -1 & 1 \\ 3 & -1 & 9 \end{array}\right]$	$\left[\begin{array}{cc\|c} 1 & 1 & 5 \\ 0 & -2 & -4 \\ 0 & 0 & 2 \end{array}\right]$	REF has $0 = b_i \implies$ Inconsistency Not Solvable

[1] Note: When we say $0 = b_i$ in the observations, we mean with $b_i \neq 0$.

solved first, and the solution is substituted in the one for the row above and so on, until all the variables are solved.

With Gaussian elimination and back substitution, we can solve a system of linear equations is as fully as possible. Moreover, we can say a lot about the solvability of the system by looking at the pivots, as we shall illustrate using the examples in Table 4.1. We have the augmented matrices for these equations, their REF, and our observations on solvability of the system based on the properties of the REF in Table 4.3. These observations are, in fact, general statements about the solvability of systems of linear equations, as listed below.

Solvability Conditions of a system of linear equations based on the properties of the REF of its augmented matrix:

1. If we have a row in the REF (of the augmented matrix $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$) with all zeros in the coefficient ($\boldsymbol{A}$) part and a nonzero element in the constant ($\boldsymbol{b}$) part, the system is not solvable.

2. If the number of pivots in the REF is the same as the number of unknowns, we have a unique solution, provided the system is solvable by the first condition.

3. If the number of pivots is smaller than the number of variables, we have an infinity of solutions, providing the system is solvable (by the first condition).

The first solvability condition above can be stated in a variety of ways:

- If a certain linear combination of the rows of the coefficient matrix gives all zeros, the same combination of the elements of the constant vector also should give zero. Else, the system is inconsistent and not solvable.
- If the rank of the augmented matrix is greater than the rank of the coefficient matrix, the system is inconsistent and not solvable.
- If the pivot in any row of the REF of the augmented matrix is in the augmented column (meaning, in the constants column, coming from $\boldsymbol{b}$), the system is not solvable.

Knowing that the number of pivots in the REF is the same as the rank of the matrix, we can restate the solvability conditions more formally (albeit absolutely equivalently and therefore superfluously) as follows: For a system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ with m equations and n unknowns (in other words, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$), with $\text{rank}(\boldsymbol{A}) = r$ and $\text{rank}(\left[\boldsymbol{A} \mid \boldsymbol{b}\right]) = r'$, we have:

1. If $r' > r$, the system is inconsistent and unsolvable.
2. If $n = r = r'$, the system has a unique solution.
3. If $n > r$, the system has an infinity of solutions.

Notes:

- The largest value the rank of any matrix can have is the smaller of its dimensions.
- The rank r of the coefficient matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ can never be larger than the rank r' of $\left[\boldsymbol{A} \mid \boldsymbol{b}\right] \in \mathbb{R}^{m\times(n+1)}$.
- As a corollary, if $n > m$, we have too few equations. The system, if solvable, will always have infinitely many solutions.

We will look at some more examples to illustrate the solvability conditions when we discuss the elementary matrices (the matrices

associated with the elementary operations) after the next important topic on finding the complete solution of a system of linear equations when we have an infinite number of solutions.

4.3.2 Complete Solution

Looking at the solvability of the systems of equations in Table 4.3, we see that in several of them, we have either a unique solution or no solutions at all. In the case of no solution, we give up right away. When we have a unique solution, we can get at it using back substitution. What is more complicated is when we have infinitely many solutions as in rows 2 and 3 in Table 4.1, but the solution was still trivial to find because it was just a line representing one of the equations in the system.

Here is a system of equations with nontrivial infinity of solutions, along with its augmented matrix and its REF.

$$\begin{matrix} x+y+z=6 \\ 2x+2y+z=9 \\ x+y=3 \end{matrix} \implies \left[\boldsymbol{A} \mid \boldsymbol{b}\right] = \left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 2 & 2 & 1 & 9 \\ 1 & 1 & 0 & 3 \end{array}\right] \xrightarrow{\text{REF}} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 0 & -1 & -3 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

From the REF above, we can start back substituting: Row 3 does not say anything. Row 2 says:

$$-z = -3 \implies z = 3$$

Substituting it in the row above, we get:

$$x + y + 3 = 6 \quad \text{or} \quad x + y = 3$$

By convention, we take the variable corresponding to the non-pivot column (which is column 2 in this case, corresponding to the variable y) as a *free variable*. It can take any value $y = t$. Once y takes a value, x is fixed: $x = 3 - t$. So the complete solution to this system of equations is:

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3-t \\ t \\ 3 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 3 \end{bmatrix} + t \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$$

Note that a linear equation (such as $z = 3$) in $\mathbb{R}^3$ defines a plane. $x + y = 3$ also defines a plane. (The pair $x = 3 - t$ and $y = t$ is

a parametric equation to the same plane.) The intersection of these two planes is a line. And any point in this line is a solution to our set of linear equations.

Let's take one more example, this time with four variables x_1, x_2, x_3 and x_4:

$$x_1 + x_2 + x_3 + 2x_4 = 6$$
$$2x_1 + 2x_2 + x_3 + 7x_4 = 9$$

$$\left[\boldsymbol{A} \mid \boldsymbol{b}\right] = \left[\begin{array}{cccc|c} 1 & 1 & 1 & 2 & 6 \\ 2 & 2 & 1 & 7 & 9 \end{array}\right] \xrightarrow{\text{REF}} \left[\begin{array}{cccc|c} 1 & 1 & 1 & 2 & 6 \\ 0 & 0 & -1 & 3 & -3 \end{array}\right]$$

Columns 1 and 3 have pivots, which means x_2 and x_4 are free variables. Assigning values t_1 and t_2 to them, the last row gives us the equation: $-x_3 + 3t_2 = -3$ or $x_3 = 3 + 3t_2$. The first row stands for the equation:

$$x_1 + x_2 + x_3 + 2x_4 = 6$$

Now we know $x_2 = t_1$, $x_3 = 3 + 3t_2$ and $x_4 = t_2$. Substituting, we get:

$$x_1 + t_1 + 3 + 3t_2 + 2t_2 = 6$$

This equation gives us the parametrized value of $x_1 = 3 - t_1 - 5t_2$ and the complete solution:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 - t_1 - 5t_2 \\ t_1 \\ 3 + 3t_2 \\ t_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 3 \\ 0 \end{bmatrix} + t_1 \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + t_2 \begin{bmatrix} -5 \\ 0 \\ 3 \\ 1 \end{bmatrix} \tag{4.4}$$

We have chosen to decompose the complete solution in this way (as the sum of one vector and a linear combination of two others) because it is the right form for a later topic. Just as a preview, we will call the first vector a *particular solution* and the linear combination the *null space* later on.

General Solution The complete solution in Eqn (4.4) is also called the general solution. Let's take a closer look at it. It has the form $\boldsymbol{x}_p + t_1\boldsymbol{x}_{s_1} + t_2\boldsymbol{x}_{s_2}$. The first vector, $\boldsymbol{x}_p$ is the so-called *particular solution*. We can get it by setting all the free variables (the ones corresponding to the non-pivot columns, which are the second and fourth variables in our example above) to the value zero. The second and third terms in the solution form a linear combination of two

vectors $\boldsymbol{x}_{s_1}$ and $\boldsymbol{x}_{s_2}$. These vectors are called the special solutions of the system. We can see that they are, in fact, the solutions to the equations when the right hand side, the constants part, is zero, which is to say $\boldsymbol{b} = \boldsymbol{0}$.

When a system of linear equations is of the form $\boldsymbol{Ax} = \boldsymbol{0}$, it is called a homogeneous system because all the terms in the system are of the same order one in the variables (as opposed to some with order zero if we had a nonzero $\boldsymbol{b}$). Therefore, the special solutions are also called homogeneous solutions.

Lastly, the linear combinations of the special solutions in our example, $t_1\boldsymbol{x}_{s_1} + t_2\boldsymbol{x}_{s_2}$, define a plane in $\mathbb{R}^4$, as two linearly independent vectors always form a plane going through the origin $\boldsymbol{0}$. What the addition of $\boldsymbol{x}_p$ does is to shift the plane to its tip, namely the coordinate point $(3, 0, 3, 0)$, if we allow ourselves to visualize it in a coordinate space. In other words, the complete solution is any vector whose tip is on this plane defined by the special solutions $\boldsymbol{x}_{s_1}$ and $\boldsymbol{x}_{s_2}$, and shifted by the particular solution $\boldsymbol{x}_p$.

4.3.3 Elementary Matrices

Each of the row operations that we perform on a matrix $\boldsymbol{A}$ in Gaussian elimination can be thought of as a matrix multiplying $\boldsymbol{A}$ on the left. This insight comes from the row picture of matrix multiplication (§2.7.4, page 39). For easy reference, here are the elementary row operations once more:

1. Swap any two rows.

   e.g., Swap the second and third rows ($r_3 \leftrightarrow r_2$).

2. Multiply any row by a scalar.

   e.g., Multiply the second row by 3 ($r_2 \leftarrow 3r_2$).

3. Add a multiple of any row to another.

   e.g., Subtract three times the first row from the third ($r_3 \leftarrow -3r_1 + r_3$).

For any $\boldsymbol{A} \in \mathbb{R}^{3\times n}$, here are the so-called elementary matrices (or operators) that would implement the examples listed above:

$$\boldsymbol{E_1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \boldsymbol{E_2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{E_3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix}$$

By the row picture of matrix multiplication, $\boldsymbol{U} = \boldsymbol{E_1 A}$ can be stated as follows, denoting the rows of $\boldsymbol{A}$ by r_i and the rows of $\boldsymbol{U}$ by r'_i:

- r'_1 is a linear combination of the rows of $\boldsymbol{A}$:
  $r'_1 = 1r_1 + 0r_2 + 0r_3 = r_1$
- r'_2 is another linear combination: $r'_2 = 0r_1 + 0r_2 + 1r_3 = r_3$
- r'_3 is another linear combination: $r'_3 = 0r_1 + 1r_2 + 0r_3 = r_2$

As we can see, $\boldsymbol{E_1}$ implements the swap of the second and third rows of any $\boldsymbol{A} \in \mathbb{R}^{3\times n}$. In particular, it does it for the identity matrix $\boldsymbol{I}$ in $\mathbb{R}^{3\times 3}$. Therefore, we can see that the elementary matrices differ from the identity matrix of the same size by one elementary row operation.

Summarizing, we can capture each elementary row operation that we apply to a matrix in Gaussian elimination as a matrix multiplying it on the left. This matrix is called the elementary matrix or elementary operator, denoted by $\boldsymbol{E}$. Since we are reducing a matrix $\boldsymbol{A}$ to its REF, it is customary to think of its rows being replaced by linear combinations other rows: In other words, we usually write $r_i \leftarrow$ (rather than $r'_i =$) a specified linear combination, as we see in our examples.

Figure 4.3 shows the elementary matrix (in blue) that reduced our augmented matrix $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$ for the system of linear equations, $x + y = 5, x - y = 1$ to REF. We needed only one elementary operation in this example. We have more examples of elementary matrices in Tables 4.6 to 4.10.

4.3.4 LU Decomposition

As we have seen, the Row-Echelon Form (REF), the end result of Gaussian elimination, is always an upper triangular matrix. We hinted at this by calling it $\boldsymbol{U}$ in one of our equations. Since Gaussian elimination can be applied to any matrix, we can always get an upper triangular matrix from any matrix by applying the elementary row

$$\boldsymbol{Ax} = \boldsymbol{b} \Rightarrow \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$$

Subtract **Row 1** from **Row 2**:

$$\left[\begin{array}{cc|c} 1 & 1 & 5 \\ 1 & -1 & 1 \end{array}\right] \Rightarrow \left[\begin{array}{cc|c} 1 & 1 & 5 \\ 0 & -2 & -4 \end{array}\right]$$

Augmented Matrix: $[\boldsymbol{A} \mid \boldsymbol{b}]$ — Pivots ⇒ First non-zero element

The ***Elementary Row Operation*** as a Matrix:

$$\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \left[\begin{array}{cc|c} 1 & 1 & 5 \\ 1 & -1 & 1 \end{array}\right] = \left[\begin{array}{cc|c} 1 & 1 & 5 \\ 0 & -2 & -4 \end{array}\right] = \boldsymbol{U}$$

Using the **row picture** for the first matrix multiplication above:

- First Row of the product $\boldsymbol{U}$ = 1× **Row 1** + 0× **Row 2** of $[\boldsymbol{A} \mid \boldsymbol{b}]$
- Second Row of the product $\boldsymbol{U}$ = −1× **Row 1** + 1× **Row 2** of $[\boldsymbol{A} \mid \boldsymbol{b}]$

Fig. 4.3 Gaussian elimination on a simple augmented matrix, showing the elementary operator that implements the row reduction.

operations listed earlier (in §4.2.2, page 70). Table 4.4 shows an example of this process of getting a $\boldsymbol{U}$ out of a matrix by finding its REF.

We have more examples coming up (in Tables 4.4 to 4.10), where we will write down the elementary matrix that implemented each row operation. We will call these elementary matrices $\boldsymbol{E_i}$. Referring to Table 4.4, we can therefore write the REF form as:

$$\boldsymbol{U} = \boldsymbol{E_2 E_1 A} = \boldsymbol{EA}$$

Table 4.4 $\boldsymbol{A = LU}$ Decomposition

$[\boldsymbol{A}] \rightarrow$ REF	Row Op.	$\boldsymbol{E_i}$	Inv. Op.	$\boldsymbol{E_i^{-1}}$
$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 1 & 3 & 0 \end{bmatrix}$	$r_2 \leftarrow -2r_1 + r_2$ $r_3 \leftarrow -r_1 + r_3$	$\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	$r_2 \leftarrow 2r_1 + r_2$ $r_3 \leftarrow r_1 + r_3$	$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -1 \\ 0 & 2 & -1 \end{bmatrix}$	$r_3 \leftarrow 2r_2 + r_3$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$	$r_3 \leftarrow -2r_2 + r_3$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -1 \\ 0 & 0 & -3 \end{bmatrix}$	$\boldsymbol{L} = \boldsymbol{E_1^{-1} E_2^{-1}} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -2 & 1 \end{bmatrix}$		$\boldsymbol{U}$ = REF = $\begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -1 \\ 0 & 0 & -3 \end{bmatrix}$	

Let's focus on one elementary row operation, $\boldsymbol{E_2}$ (second row, under the column $\boldsymbol{E_i}$). This operation replaces row 3 with the sum of twice row 2 and row 3. The inverse of this operation would be to replace the row 3 with row 3 − twice row 2, as shown in the column Inv. Op. The elementary operation for this inverse operation is, in fact, the inverse of $\boldsymbol{E_2}$, which we can find in the last column, under $\boldsymbol{E_i^{-1}}$.

Note the order in which the elementary matrices appear in the product. We perform the first row operation $\boldsymbol{E_1}$ on $\boldsymbol{A}$, get the product $\boldsymbol{E_1 A}$ and apply the second operation $\boldsymbol{E_2}$ on this product, and so on.

We have not yet fully discussed the inverse of matrices, but the idea of the inverse undoing what the original matrix does is probably clear enough at this point. Now, starting from the upper triangular matrix $\boldsymbol{U}$ that is the REF of $\boldsymbol{A}$, we can write the following:

$$\boldsymbol{U} = \boldsymbol{EA} = \boldsymbol{E_2 E_1 A} \implies \boldsymbol{A} = \boldsymbol{E_1^{-1} E_2^{-1} U} = \boldsymbol{LU}$$

where we have called the matrix $\boldsymbol{E_1^{-1} E_2^{-1}}$, which is a lower triangular matrix (because each of the $\boldsymbol{E_i^{-1}}$ is a lower triangular matrix) $\boldsymbol{L}$. Again notice the order in which the inverses $\boldsymbol{E_i}$ appear in the equation: We undo the last operation first before moving on to the previous one.

This statement is the famous $\boldsymbol{LU}$ decomposition, which states that any matrix can be written as the product of a lower triangular matrix $\boldsymbol{L}$ and an upper triangular matrix $\boldsymbol{U}$. The algorithm we use to perform the decomposition is Gaussian elimination. Notice that all the elementary matrices and their inverses have unit determinants, and the row operations we carry out do not change the determinant of $\boldsymbol{A}$, such that $|\boldsymbol{A}| = |\boldsymbol{U}|$.

Permutation Matrices: In discussing the example in Table 4.4, we cleverly glossed over one fact: We did not use the first elementary row operation (namely, swapping rows). Row exchanges introduce symmetric matrices as their elementary matrices. They are not lower triangular. The matrix implementing row exchanges is called a *permutation matrix*, and is denoted by $\boldsymbol{P}$.

Every permutation matrix (of single row exchange) differs from the identity matrix of the same size by one or more row exchange[5]. And it is its own inverse (because exchanging row with another is undone by the same exchange): $\boldsymbol{P}^2 = \boldsymbol{I}$. To introduce another name, the matrices that are their own inverses are called involutory. And the permutation matrices implementing a single row exchange are involutory.

Here are all possible permutation matrices that swap one or more pairs of rows in $\mathbb{R}^2$ and $\mathbb{R}^3$:

[5]In principle, we can have permutation matrices that perform multiple row exchanges, but for the purposes of this book, we focus on one row exchange at a time.

Table 4.5 $\boldsymbol{A = PLU}$ Decomposition

$[\boldsymbol{A}] \to$ REF	Row Op.	$\boldsymbol{E_i}$	Inv. Op.	$\boldsymbol{E_i^{-1}}$
$\begin{bmatrix}1&1&1\\2&2&1\\1&3&0\end{bmatrix}$	$r_2 \leftrightarrow r_3$	$\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}$	$r_2 \leftrightarrow r_3$	$\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}$
$\begin{bmatrix}1&1&1\\1&3&0\\2&2&1\end{bmatrix}$	$r_2 \leftarrow -r_1 + r_2$ $r_3 \leftarrow -2r_1 + r_3$	$\begin{bmatrix}1&0&0\\-1&1&0\\-2&0&1\end{bmatrix}$	$r_2 \leftarrow r_1 + r_2$ $r_3 \leftarrow 2r_1 + r_3$	$\begin{bmatrix}1&0&0\\1&1&0\\2&0&1\end{bmatrix}$
$\begin{bmatrix}1&1&1\\0&2&-1\\0&0&-1\end{bmatrix}$	$\boldsymbol{P} = \begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}$	$\boldsymbol{L} = \boldsymbol{E_2^{-1}} = \begin{bmatrix}1&0&0\\1&1&0\\2&0&1\end{bmatrix}$	$\boldsymbol{U}$ = REF = $\begin{bmatrix}1&1&1\\0&2&-1\\0&0&-1\end{bmatrix}$	

$r_1 \leftrightarrow r_2$	$r_1 \leftrightarrow r_2$	$r_2 \leftrightarrow r_3$	$r_3 \leftrightarrow r_1$	$r_1 \leftrightarrow r_2$ $r_2 \leftrightarrow r_3$
$\begin{bmatrix}0&1\\1&0\end{bmatrix}$	$\begin{bmatrix}0&1&0\\1&0&0\\0&0&1\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&1&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}$

Note that the last one does multiple row exchanges and consequently $\boldsymbol{P}^2 \neq \boldsymbol{I}$: It is not involutory.

4.3.5 $\boldsymbol{PLU}$ Decomposition

Since the elementary operation of row exchanges breaks our decomposition of $\boldsymbol{A = LU}$, we keep the permutation part separate, and come up with the general, unbreakable, universally applicable decomposition $\boldsymbol{A = PLU}$. We can see an example of this decomposition in Table 4.5.

4.3.6 Computing Determinants

As breifly mentioned earlier, we can use Gaussian elimination to compute the determinant of a square matrix. Once we get the REF ($\boldsymbol{U}$), the determinant is simply the product of diagonal elements. The reason is that the elementary row operations do not change the absolute value of the determinant. Or rather, they change the determinant in a way we can keep track of.

- Swapping two rows will result in the determinant flipping sign (Property 1 in §3.3.4, page 58). We therefore have to keep track of how many times we swapped rows when applying Gaussian elimination to compute determinants.

- Although we defined it as an elementary row operation (for future use), we do not scale rows in Gaussian elimination. If we did, we would have to keep track of it as well because the determinant then would be multiplied by the same scaling factor (Property 3 in §3.3.4, page 58)

- The third elementary operation (adding/subtracting a multiple of a row from another one) does not change the determinant (by Property 6 in §3.3.4, page 58).

Why would we want to compute determinants this way rather than using the Laplace expansion (Eqn (3.4))? The computational complexity of the Laplace expansion is $\mathcal{O}(n!)$ while that of the Gaussian-elimination way is only $\mathcal{O}(n^3)$, which is a huge gain for large values of n. As a special case, if the matrix does not have a full set of pivots, the determinant is zero, which means we do not even have to complete Gaussian elimination; we can stop the moment we get a row without a pivot.

4.4 More Examples

In Table 4.6, we have the first example of a system of three equations: We have as many equations as unknowns. After Gaussian elimination, we can see that we have a full set of pivots, and $\boldsymbol{A}$ is indeed full rank. Therefore, we get a unique solution through back substitution, as shown after the table.

We can restate the determinant computation more formally in terms of the $\boldsymbol{PLU}$ decomposition we discussed above. If we start with a square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, we can make the following statements:

1. By the product rule of determinants (Property 10 in §3.3.4, page 58, $|\boldsymbol{AB}| = |\boldsymbol{A}|\,|\boldsymbol{B}|$), we can see that $|\boldsymbol{A}| = |\boldsymbol{P}|\,|\boldsymbol{L}|\,|\boldsymbol{U}|$.

2. $|\boldsymbol{P}| = \pm 1$. (Exercise: Prove this statement.)

3. $|\boldsymbol{L}| = 1$. (Exercise: Prove this statement.)

4. Therefore $|\boldsymbol{A}| = |\boldsymbol{P}|\,|\boldsymbol{U}| = \pm\,|\boldsymbol{U}|$, with the sign given by $|\boldsymbol{P}|$.

Table 4.6 Example 1: Gaussian elimination and solvability: Unique solution

Equations	$\left[\boldsymbol{A} \mid \boldsymbol{b}\right] \rightarrow$ REF	Row Operation	Elementary Matrix $\boldsymbol{E_i}$
$x+y+z=6$ $2x+2y+z=9$ $x+3y=7$	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 2 & 2 & 1 & 9 \\ 1 & 3 & 0 & 7 \end{array}\right]$	$r_2 \leftarrow -2r_1 + r_2$ $r_3 \leftarrow -r_1 + r_3$	$\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$
	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 0 & -1 & -3 \\ 0 & 2 & -1 & 1 \end{array}\right]$	$r_2 \leftrightarrow r_3$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$
	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 2 & -1 & 1 \\ 0 & 0 & -1 & -3 \end{array}\right]$	REF	

Back Substitution:
The last row of the REF($\boldsymbol{A}$) says $-z = -3 \implies z = 3$.
Substituting it in the row above, $2y - 3 = 1 \implies y = 2$.
Substituting z and y in the first row, $x + 2 + 3 = 6 \implies x = 1$.
The complete and unique solution is: $(x, y, z) = (1, 2, 3)$

5. Since $\boldsymbol{U}$ is triangular, its determinant is the product of its diagonal elements.

$$|\boldsymbol{U}| = \prod_{i=1}^{n} u_{ii}$$

6. In particular, if $\boldsymbol{U}$ does not have full set of pivots (and is rank deficient), at least one of $u_{ii} = 0 \implies |\boldsymbol{A}| = 0$.

In the second example, Table 4.7, we have modified the third equation such that it is no longer consistent with the first two. We again have as many equations as unknowns. However, after Gaussian elimination, we can see that the last row reads $\left[\begin{array}{ccc\|c} 0 & 0 & 0 & 3 \end{array}\right]$. It translates to an equation $0 = 3$, which indicates that the equations are inconsistent. Note that rank$(\left[\boldsymbol{A} \mid \boldsymbol{b}\right]) = 3 >$ rank$(\boldsymbol{A}) = 2$.

Table 4.7 Example 2: Gaussian elimination and solvability: No solutions

Equations	$\left[\boldsymbol{A} \mid \boldsymbol{b}\right] \rightarrow$ REF	Row Operation	Elementary Matrix $\boldsymbol{E_i}$
$x+y+z=6$ $2x+2y+z=9$ $x+y=6$	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 2 & 2 & 1 & 9 \\ 1 & 1 & 0 & 6 \end{array}\right]$	$r_2 \leftarrow -2r_1 + r_2$ $r_3 \leftarrow -r_1 + r_3$	$\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$
	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 0 & -1 & -3 \\ 0 & 0 & -1 & 0 \end{array}\right]$	$r_3 \leftarrow -r_2 + r_3$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}$
	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 0 & -1 & -3 \\ 0 & 0 & 0 & 3 \end{array}\right]$	REF	

The last row of the REF($\boldsymbol{A}$) is $\left[\begin{array}{ccc\|c} 0 & 0 & 0 & 3 \end{array}\right]$, saying $0 = 3 \implies$ The system is inconsistent.

Table 4.8 Ecample 3: More equations than unknowns, but solvable

Equations	$\left[\boldsymbol{A} \mid \boldsymbol{b}\right] \rightarrow$ REF	Row Operation	Elementary Matrix $\boldsymbol{E_i}$
$x+y+z=6$ $2x+2y+z=9$ $x+3y=7$ $3x+y+z=8$	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 2 & 2 & 1 & 9 \\ 1 & 3 & 0 & 7 \\ 3 & 1 & 1 & 8 \end{array}\right]$	$r_2 \leftarrow -2r_1 + r_2$ $r_3 \leftarrow -r_1 + r_3$ $r_4 \leftarrow -3r_1 + r_4$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -3 & 0 & 0 & 1 \end{bmatrix}$
	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 0 & -1 & -3 \\ 0 & 2 & -1 & 1 \\ 0 & -2 & -2 & -10 \end{array}\right]$	$r_2 \leftrightarrow r_3$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 2 & -1 & 1 \\ 0 & 0 & -1 & -3 \\ 0 & -2 & -2 & -10 \end{array}\right]$	$r_4 \leftarrow r_2 + r_4$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$
	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 2 & -1 & 1 \\ 0 & 0 & -1 & -3 \\ 0 & 0 & -3 & -9 \end{array}\right]$	$r_4 \leftarrow -3r_3 + r_4$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -3 & 1 \end{bmatrix}$
	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 2 & -1 & 1 \\ 0 & 0 & -1 & -3 \\ 0 & 0 & 0 & 0 \end{array}\right]$	REF	

The last row of the REF($\boldsymbol{A}$), $\left[\begin{array}{ccc\|c} 0 & 0 & 0 & 0 \end{array}\right]$,says $0 = 0 \implies$ The system is consistent.

The third example in Table 4.8, we have added one more equation to the system in Table 4.6. Thus, we have four equations for three unknowns. But the augmented matrix reduces with the last row reading $0 = 0$, which means the fourth equation is consistent with

Table 4.9 Example 4: More equations than unknowns, with no solutions

Equations	$\left[\boldsymbol{A} \mid \boldsymbol{b}\right] \rightarrow$ REF	Row Operation	Elementary Matrix $\boldsymbol{E_i}$
$x+y+z=6$ $2x+2y+z=9$ $x+3y=7$ $3x+y+z=11$	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 2 & 2 & 1 & 9 \\ 1 & 3 & 0 & 7 \\ 3 & 1 & 1 & 11 \end{array}\right]$	$r_2 \leftarrow -2r_1 + r_2$ $r_3 \leftarrow -r_1 + r_3$ $r_4 \leftarrow -3r_1 + r_4$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -3 & 0 & 0 & 1 \end{bmatrix}$
	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 0 & -1 & -3 \\ 0 & 2 & -1 & 1 \\ 0 & -2 & -2 & -7 \end{array}\right]$	$r_2 \leftrightarrow r_3$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 2 & -1 & 1 \\ 0 & 0 & -1 & -3 \\ 0 & -2 & -2 & -7 \end{array}\right]$	$r_4 \leftarrow r_2 + r_4$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$
	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 2 & -1 & 1 \\ 0 & 0 & -1 & -3 \\ 0 & 0 & -3 & -6 \end{array}\right]$	$r_4 \leftarrow -3r_3 + r_4$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -3 & 1 \end{bmatrix}$
	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 2 & -1 & 1 \\ 0 & 0 & -1 & -3 \\ 0 & 0 & 0 & 3 \end{array}\right]$	REF	

The last row of the REF($\boldsymbol{A}$) is $\left[\begin{array}{ccc\|c} 0 & 0 & 0 & 3 \end{array}\right]$, saying $0 = 3 \implies$ The system is inconsistent.

Table 4.10 Example 5: Gaussian elimination and solvability: Infinity of solutions

Equations	$\left[\boldsymbol{A} \mid \boldsymbol{b}\right] \rightarrow$ REF	Row Operation	Elementary Matrix $\boldsymbol{E_i}$
$x+y+z=6$ $2x+2y+z=9$ $x+y=3$	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 2 & 2 & 1 & 9 \\ 1 & 1 & 0 & 3 \end{array}\right]$	$r_2 \leftarrow -2r_1 + r_2$ $r_3 \leftarrow -r_1 + r_3$	$\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$
	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 0 & -1 & -3 \\ 0 & 0 & -1 & -3 \end{array}\right]$	$r_3 \leftarrow -r_2 + r_3$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}$
	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 0 & -1 & -3 \\ 0 & 0 & 0 & 0 \end{array}\right]$	REF	

The last row of the REF($\boldsymbol{A}$) is $\left[\begin{array}{ccc\|c} 0 & 0 & 0 & 0 \end{array}\right]$, indicating consistency. With two independent equations for three unknowns, we get an infinity of solutions.

the rest. And $\text{rank}(\left[\boldsymbol{A} \mid \boldsymbol{b}\right]) = \text{rank}(\boldsymbol{A}) = 3$, same as the number of unknowns. Hence unique solution.

The fourth example is similar to the third one; we have added a fourth equation in Table 4.9 as well. However, the new equation is not consistent with the rest.

In the last example in Table 4.10 (which we used to illustrate the complete solution with free variables), we have as many equations as unknowns. After Gaussian elimination, we get the last row reading $\left[\begin{array}{ccc\|c} 0 & 0 & 0 & 0 \end{array}\right]$; no zero=nonzero row, indicating that the equations are consistent. However $\text{rank}(\left[\boldsymbol{A} \mid \boldsymbol{b}\right]) = \text{rank}(\boldsymbol{A}) = 2$ with three unknowns, which means we have infinitely many combinations of (x, y, z) that can satisfy these equations.

4.5 Beyond Gaussian Elimination

In this chapter, we looked at the applications of Gaussian elimination in computing determinants and solving equations. We briefly introduced matrix inverses. The interplay between ranks, determinants and inverses will be further explored in the next chapter.

5

Ranks and Inverses of Matrices

Everything is simpler than you think and at the same time more complex than you imagine.

—Johann Wolfgang von Goethe

Along with determinants, the rank of a matrix is a number that says a lot about the properties of the matrix. Unlike determinants, however, rank is defined for all matrices, not merely square ones. And rank is an integer, while determinant is a function of the elements of the matrix, and belongs to the same field over which it is defined. For square matrices, ranks and determinants are related to each other. In fact, most things in Linear Algebra tend to be interconnected. In this chapter, we will summarize what we already learned about ranks, and expand on it. The full discussion of ranks, however, will have to wait until we look at the underlying geometry of matrices and vectors.

5.1 Rank of a Matrix

Earlier, we stated that the rank of a matrix is the number of pivots in its row echelon form (REF), which we get through Gaussian elimination. This algorithm is merely a series of elementary row operations, each of which is about taking a linear combination of the rows of the matrix. And, right from the opening chapters where we introduced vectors and matrices (in §2.2.3, page 25), we talked about linear combinations.

If we can combine a bunch of rows in a matrix so as to get a different row, then the rows are not linearly independent. To state it without ambiguity using mathematical lingo, for a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, we say that a nonzero i^{th} row ($\boldsymbol{r_i^\mathsf{T}}$) is a linear combination of the other rows if we can write:

$$\boldsymbol{r_i^\mathsf{T}} = \sum_{j=1, j \neq i}^{m} s_j \boldsymbol{r_j^\mathsf{T}}, \quad \boldsymbol{r_j} \in \mathbb{R}^n, \quad s_j \in \mathbb{R}$$

When we can write one row as the linear combination of the rest, we say that that row is *not* linearly independent. Note that any row that is not linearly independent can be reduced to zero using the elementary row operations. Once a row is reduced to zero, it does not have a pivot. Conversely, all nonzero rows do have pivots, and they are linearly independent. Therefore, we can state that the rank of a matrix is the number of linearly independent rows.

5.1.1 Row and Column Ranks

Strictly speaking, the number of linearly independent rows would be the row rank of the matrix. Similarly, we can define a column rank, which is the number of linearly independent columns of the matrix. Or, the number of pivots when Gaussian elimination is performed column-wise, using column-reduction operations.

We have a theorem that states that the row rank is the same as the column rank. It is fairly easy to prove this fact. We will provide some hints and leave it as an exercise after looking at various possible shapes of matrices, the Gauss-Jordan algorithm and the canonical form it produces (in §5.2.2, page 95). We also have the full proof in one of the advanced topics, in §12.6 (page 239).

5.1.2 Shapes of Matrices

A general matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ can be tall if the number of rows is greater than the number of columns: $m > n$. Or it can be wide (or, less charitably, fat) if it has more columns than rows: $m < n$. When $m = n$, what we have is a square matrix.

For a square matrix, the best possible situation is that we may have a pivot in every row/column. We cannot have more pivots than that. Therefore, the rank r of the matrix has a maximum value of $m = n$.

$$\boldsymbol{A} \in \mathbb{R}^{n \times n} \implies \text{rank}(\boldsymbol{A}) \leq n$$

For a "tall" matrix, the best scenario is when we have a pivot in each of the columns. We cannot have more than one pivot in the same column. So we have $r \leq n$ when $\boldsymbol{A} \in \mathbb{R}^{m \times n}, n < m$. Similarly, for a "wide" matrix, we have, $r \leq m$ when $\boldsymbol{A} \in \mathbb{R}^{m \times n}, n > m$. Combining these two limits, we can write:

$$\boldsymbol{A} \in \mathbb{R}^{m \times n} \implies \text{rank}(\boldsymbol{A}) \leq \min(m, n)$$

If the rank $r = \min(m, n)$, we call $\boldsymbol{A}$ a full-rank matrix; else, it is rank deficient. If $r = m$, it is often called a full-row-rank (or if $r = n$, full-column-rank) matrix.

5.1.3 Properties of Ranks

Here are some of the properties of ranks of matrices. Although listed without any formal proof, the properties are either obvious, or derived from the fact that the rank is the number of linearly independent columns or rows, and that matrix multiplication is about taking their linear combinations (as in the row and column pictures). The proofs will have to wait till we learn more about the geometry associated with matrices because ranks are closely linked to it.

For $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, we have:

1. The only matrix that has a rank of zero is the zero matrix, which is a matrix that has all entries as zeros.

2. If $\boldsymbol{B} \in \mathbb{R}^{n \times k}$, conformant for multiplication with $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ on the right,

$$\text{rank}(\boldsymbol{AB}) \leq \min(\text{rank}(\boldsymbol{A}), \text{rank}(\boldsymbol{B}))$$

3. If $\boldsymbol{B} \in \mathbb{R}^{n\times k}$, conformant for multiplication with $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ on the right, with $\text{rank}(\boldsymbol{A}) = r$ and $\text{rank}(\boldsymbol{B}) = n$ (full-row rank $\implies k \geq n$),

$$\text{rank}(\boldsymbol{AB}) = \text{rank}(\boldsymbol{A}) = r$$

This property follows from the column picture of matrix multiplication and the definition of rank as the number of linearly independent columns: The product $\boldsymbol{AB}$ has columns that are linear combinations of the r independent columns of $\boldsymbol{A}$. Note that $m \geq r$ and $n \geq r$ (since $\text{rank}(\boldsymbol{A}) = r$), $k \geq n$ (since $\text{rank}(\boldsymbol{B}) = n$) $\implies k \geq r$. In other words, the product $\boldsymbol{AB} \in \mathbb{R}^{m\times k}$ has enough rows and columns to accommodate the rank of r. Or, stating it using still other words, while multiplying on the right, $\boldsymbol{B}$ has enough rows and columns to preserve the rank of $\boldsymbol{A}$, subject to the condition $\text{rank}(\boldsymbol{AB}) \leq \min(\text{rank}(\boldsymbol{A}), \text{rank}(\boldsymbol{B}))$.

4. Similarly, if $\boldsymbol{B} \in \mathbb{R}^{k\times m}$, conformant for multiplication with $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ on the left, with $\text{rank}(\boldsymbol{B}) = m$ (full-column rank),

$$\text{rank}(\boldsymbol{BA}) = \text{rank}(\boldsymbol{A})$$

This property follows from the row picture of matrix multiplication and the definition of rank as the number of linearly independent rows. The explanation is the transpose case of the previous case. Keep in mind that these two properties (3 and 4) do not violate the condition in property (2).

5. If $\boldsymbol{B} \in \mathbb{R}^{m\times n}$, same dimensions as $\boldsymbol{A}$ so that we can add them,

$$\text{rank}(\boldsymbol{A} + \boldsymbol{B}) \leq \text{rank}(\boldsymbol{A}) + \text{rank}(\boldsymbol{B})$$

6. As a corollary of the previous property, we can say that if $\text{rank}(\boldsymbol{A}) = r$, we can write it as a sum of r matrices of rank one, and we do not need more matrices. Although stated as an innocuous-looking factoid, this property has important implications in computer science, in the context of data compression, as we shall see later on.

7. A very important property that we will prove later on (much later, in §12.6, page 239):

$$\text{rank}\big(\boldsymbol{AA}^{\mathsf{T}}\big) = \text{rank}\big(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\big) = \text{rank}(\boldsymbol{A}) = \text{rank}\big(\boldsymbol{A}^{\mathsf{T}}\big)$$

$\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$ is called the Gram matrix, and has properties that make it important in machine learning.

5.2 Gauss-Jordan Elimination

While discussing the Gaussian elimination algorithm in the previous chapter, we ended up with the $\boldsymbol{A} = \boldsymbol{PLU}$ decomposition. The $\boldsymbol{U}$ matrix had the same determinant as $\boldsymbol{A}$ (and $\boldsymbol{P}$ contained the information about the sign of the determinant).

If we are not really concerned about computing the determinant, but only want solve the system of linear equations $\boldsymbol{Ax} = \boldsymbol{b}$, we can keep applying elementary row operations instead of the back substitution we discussed earlier. This algorithm to solve linear equations is known as *Gauss-Jordan Elimination*.

As a reminder to ourselves, we have the *elementary row operations*, as listed below:

- Swap any two rows.
- Multiply any row by a scalar.
- Add a multiple of any row to another.

In the previous chapter, although we listed the three elementary operations, we did not use the second one, namely scaling a row. In the Gauss-Jordan algorithm, we will use it as well.

Before stating the steps involved in Gauss-Jordan elimination, let's look at the example in Figure 5.1. Here, we start from the REF of the augmented matrix of the equations $x + y = 5, x - y = 1$ (see Figure 4.2), and keep applying the elementary row operations, which do not affect the solution of the system of equations.

In this case, the REF of the augmented matrix, $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$, becomes the identity matrix in the coefficient ($\boldsymbol{A}$) side, at which point the constants column contains the solution. Remember, each row of the augmented matrix (whether row reduced or not) represents an equation. What equation does the top row of final matrix from Gauss-Jordan elimination represent? It reads: $1x + 0y = 3$ or $x = 3$. Similarly, the second row gives us the equation $y = 2$.

The aim of the Gauss-Jordan algorithm is to get to an identity matrix in the coefficient part of the augmented matrix. In other

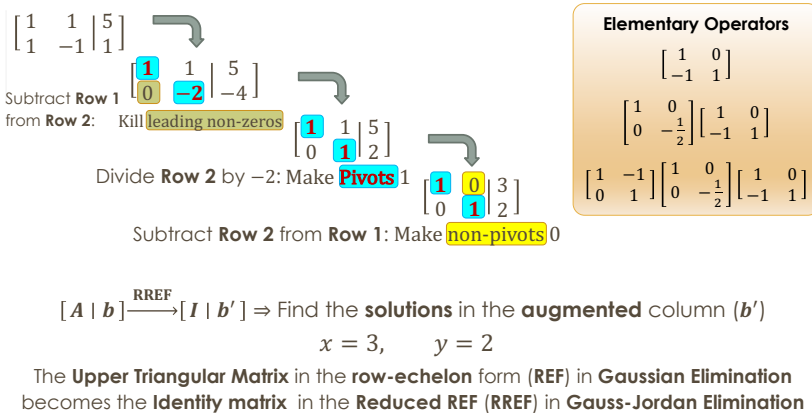


Fig. 5.1 Example of Gauss-Jordan elimination to solve a simple system of linear equations

words, try to make $\boldsymbol{A}$ in $\boldsymbol{Ax} = \boldsymbol{b}$ an $\boldsymbol{I}$, failing which, get as close to the identity matrix as possible. This form of the coefficient matrix is called the *Reduced Row Echelon Form*, or RREF, also known as the *Canonical Form*. For a full-rank, square matrix, its canonical form is the identity matrix of the same size, as we saw in the example in Figure 5.1. We will soon discuss other combinations of shapes and ranks and their canonical forms.

Reduced Row Echelon Form

Definition: A matrix is in its Reduced Row Echelon or Canonical Form if it has the following characteristics:

1. It is in REF. (See §4.2.3, page 70)
2. Every nonzero row is scaled such that the pivot value is one.
3. In every pivot column, the only nonzero element is the pivot.

As we can see, the *Reduced* Row Echelon Form (RREF) is the Row Echelon Form (REF) with some special properties, as its name indicates. The actual definition of REF and RREF are a bit fluid: Different textbooks may define them differently. For instance, some may suggest that REF should have all pivots equalling unity, through scaling. Such differences, however, do not affect the conceptual significance of the forms and perhaps not even their applications.

Gauss-Jordan Elimination: Algorithm

We can state the Gauss-Jordan algorithm also using pseudo-code so that the steps may be clearer to the students of computer science.

Input: $\boldsymbol{A} = \left[a_{ij}\right] \in \mathbb{R}^{m \times n}$
Output: RRRF($\boldsymbol{A}$) $\in \mathbb{R}^{m \times n}$

Get the REF using Gaussian Elimination
$\boldsymbol{A} \leftarrow$ REF($\boldsymbol{A}$); pivots $\leftarrow$ pivots of $\boldsymbol{A}$
for $i = 1$ to m **do**
  if pivots$_i = 0$ **then**
    Exit **for** loop
  end if
  Scale the row to get unit pivot
  $r_i \leftarrow \frac{r_i}{pivots_i}$
  Subtract scaled pivot row from other rows to get zero pivot column
  for $j = 1$ to m **do**
    if $j \neq i$ **then**
      $k \leftarrow$ pivot column index
      $r_j \leftarrow r_j - a_{jk} r_i$
    end if
  end for
end for
return $\boldsymbol{A}$ as RREF

Note that while the REF (which is the result of Gaussian elimination) can have different shapes depending on the order in which the row operations are performed, RREF (the result of Gauss-Jordan) is immutable: A matrix has a unique RREF. In this sense RREF is more fundamental than REF, and points to the underlying characteristics of the matrix.

5.2.1 The Algorithm

With the insight from the example in Figure 5.1, we can state the steps in the Gauss-Jordan algorithm (for a matrix $\boldsymbol{A} = \left[a_{ij}\right] \in \mathbb{R}^{m \times n}$) as follows:

1. Run the Gaussian elimination algorithm (§4.2.4, page 71) to get REF($\boldsymbol{A}$).

2. Loop (with index i) over the rows of REF($\boldsymbol{A}$) and scale the i^{th} row by $\frac{1}{a_{ik}}$ (where k is the column where Pivot$_i$ appears) so that Pivot$_i = 1$.

3. Loop (with index j) over all the elements of the pivot column from 1 to $i-1$, multiply the i^{th} row with $-a_{jk}$ and add it to the j^{th} row to get zeros in the k^{th} for all rows above the i^{th} row.

4. Loop back to step 2 (with $i \leftarrow i+1$) and iterate until all rows or columns are exhausted.

Let's look at two examples to illustrate the outputs of Gauss-Jordan algorithm. Starting from the REF of the augmented matrices in Tables 4.7 and 4.10, we find their RREF. The examples are in Tables 5.1 and 5.2.

Table 5.1 Gauss-Jordan on a full-rank, square coefficient matrix, giving us the unique solution

Equations	$[\boldsymbol{A} \mid \boldsymbol{b}]$	REF	RREF
$x+y+z=6$ $2x+2y+z=9$ $x+3y=7$	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 2 & 2 & 1 & 9 \\ 1 & 3 & 0 & 7 \end{array}\right]$	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 2 & -1 & 1 \\ 0 & 0 & -1 & -3 \end{array}\right]$	$\left[\begin{array}{ccc\|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{array}\right]$

Here, we start from the REF form of $[\boldsymbol{A} \mid \boldsymbol{b}]$ (as in Table 4.7), and do further row reduction to get $\boldsymbol{I}$ in the coefficient part. In other words, in total, we apply the Gauss-Jordan algorithm. We can then read out the solution $(x, y, z) = (1, 2, 3)$ from the augmented part of the RREF.

If we started from the REF of $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$ in Table 4.9, where we have inconsistent equations, the RREF would be an uninteresting 4×4 identity matrix. What it says is that we have a rank of four in $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$, with only three unknowns, indicating that we have no solutions.

Notice that we did not write down the elementary matrices for the Gauss-Jordan algorithm, as we diligently did for Gaussian elimination? The reason for doing that in Gaussian elimination was to arrive at the $\boldsymbol{A} = \boldsymbol{PLU}$ decomposition so as to compute its determinant. We do not use Gauss-Jordan to compute determinants, although we could, if we wanted to.

5.2.2 Shapes of Canonical Forms

We looked at what we called "tall" and "wide" matrices in §5.1.2. What are the canonical forms of matrices of these various shapes? Remember, canonical form is an alias for RREF.

Table 5.2 Gauss-Jordan on a rank-deficient coefficient matrix, resulting in an infinity of solutions

Equations	$[\boldsymbol{A} \mid \boldsymbol{b}]$	REF	RREF
$x + y + z = 6$ $2x + 2y + z = 9$ $x + y = 3$	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 2 & 2 & 1 & 9 \\ 1 & 1 & 0 & 3 \end{array}\right]$	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 0 & -1 & -3 \\ 0 & 0 & 0 & 0 \end{array}\right]$	$\left[\begin{array}{ccc\|c} 1 & 1 & 0 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 \end{array}\right]$

This second example starts from the REF form of $[\boldsymbol{A} \mid \boldsymbol{b}]$ from Table 4.10. The Gauss-Jordan algorithm gets us as close to the identity matrix as possible in the coefficient part of $[\boldsymbol{A} \mid \boldsymbol{b}]$. The solution is: $z = 3$ and $x + y = 3$, which is usually written as $(x, y, z) = (3 - t, t, 3)$.

Square Matrices Let's start with a square matrix of full rank: $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, $\text{rank}(\boldsymbol{A}) = n$. Since all the rows have pivots, and since we can scale the pivots to get ones, we will end up with the identity matrix $\boldsymbol{I}$ as the canonical form. In fact, this is what we did in Table 5.1, for the coefficient matrix.

$$\boldsymbol{A} \in \mathbb{R}^{n \times n},\ \text{rank}(\boldsymbol{A}) = n \implies \boldsymbol{A} \xrightarrow{\text{RREF}} \boldsymbol{I}_n \in \mathbb{R}^{n \times n}$$

where the symbol $\boldsymbol{I}_n$ stands for the identity matrix in $\mathbb{R}^{n \times n}$.

What happens if the matrix is rank deficient? If $\text{rank}(\boldsymbol{A}) = r < n$, we have only r pivots, which the Gauss-Jordan algorithm will place in the first r rows. The last $n - r$ rows of the RREF, therefore, will be zeros. The top r rows will contain the identity matrix $\boldsymbol{I}$ because the pivots will be scaled to one, and they will be used to eradicate other elements in the pivot columns.

The pivot columns, however, may not be ideally placed to give us an $\boldsymbol{I}_r$ matrix in the top-left corner, but may appear interspersed among other pivot-less columns. As we remember from solving linear equations, columns with no pivots result in free variables, and we may call these columns part of a matrix $\boldsymbol{F}$. Thus the canonical form of a rank-deficient square matrix looks like:

$$\boldsymbol{A} \in \mathbb{R}^{n \times n},\ \text{rank}(\boldsymbol{A}) = r < n \implies \boldsymbol{A} \xrightarrow{\text{RREF}} \left[\begin{array}{c} \boldsymbol{I}_r \oplus \boldsymbol{F}_{r \times (n-r)} \\ \hline \boldsymbol{0}_{(n-r) \times n} \end{array}\right] \in \mathbb{R}^{n \times n}$$

We use the expression $\boldsymbol{I} \oplus \boldsymbol{F}$ to indicate that $\boldsymbol{I}$ and $\boldsymbol{F}$ are shuffled in together. Note that $\oplus$ is *not* a standard notation for this purpose.

In Table 5.2, we have an example of such a system: The coefficient matrix is a square matrix with rank deficiency. Looking at the RREF column of the table, we can see that pivot columns do make an identity matrix, but the pivot-less, free-variable columns are interspersed

among it. The pivot columns are 1 and 3, and the free variable is in column 2. The bottom $m - r = 1$ row is a zero row.

Tall Matrices For a "tall" matrix of full (column) rank, we will get pivots for the first n rows, and the rest $m - n$ rows will be zeros. The n pivots can be normalized so that the top part of the canonical form becomes $\boldsymbol{I}_n$, and we will have:

$$\boldsymbol{A} \in \mathbb{R}^{m\times n},\ \text{rank}(\boldsymbol{A}) = n < m \implies \boldsymbol{A} \xrightarrow{\text{RREF}} \left[\begin{array}{c} \boldsymbol{I}_n \\ \hline \boldsymbol{0}_{(m-n)\times n} \end{array}\right] \in \mathbb{R}^{m\times n}$$

If the tall matrix is rank deficient, we again have an $\boldsymbol{I}$ and $\boldsymbol{F}$ in the top r rows, followed by zeros. The pivot and pivot-less columns may get interspersed among each other, as in the case of the square matrix, giving us a canonical form that may see identical to the one for the rank-deficient square matrix, but the indices are different: It has one instance of n replaced by m.

$$\boldsymbol{A} \in \mathbb{R}^{m\times n},\ \text{rank}(\boldsymbol{A}) = r < n < m \implies \boldsymbol{A} \xrightarrow{\text{RREF}} \left[\begin{array}{c} \boldsymbol{I}_r \oplus \boldsymbol{F}_{r\times(n-r)} \\ \hline \boldsymbol{0}_{(m-r)\times n} \end{array}\right] \in \mathbb{R}^{m\times n}$$

Wide Matrices Moving on to "wide" matrices (with more columns than rows), if they have full (row) rank, they will get a canonical form that may not cleanly separate into $\boldsymbol{I}$ and $\boldsymbol{F}$. We can see an example in Table 5.3, which has two independent and consistent equations, and therefore a full-row-rank $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$. The first and third columns have pivots, and make an identity matrix, but the free-variable column appears in between. The general shape of a full-row-rank, wide matrix, therefore, looks like:

$$\boldsymbol{A} \in \mathbb{R}^{m\times n},\ \text{rank}(\boldsymbol{A}) = m < n \implies \boldsymbol{A} \xrightarrow{\text{RREF}} \left[\boldsymbol{I}_m \oplus \boldsymbol{F}_{m\times(n-m)}\right] \in \mathbb{R}^{m\times n}$$

For a rank-deficient wide matrix also, the canonical form is similar to the other rank-deficient cases. What we had for the coefficient matrix (or for the whole augmented matrix $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$) in Table 5.2 was a wide matrix (not enough independent equations) with rank deficiency. Looking at the RREF column of the table, we can see that pivot elements do make an identity matrix, but the pivot-less, free variable columns are interspersed among it. The pivot columns are 1

Table 5.3 Gauss-Jordan on a rank-deficient "wide" coefficient matrix

Equations	$[\boldsymbol{A} \mid \boldsymbol{b}]$	REF	RREF
$x + y + z = 6$ $2x + 2y + z = 9$	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 2 & 2 & 1 & 9 \end{array}\right]$	$\left[\begin{array}{ccc\|c} 1 & 1 & 1 & 6 \\ 0 & 0 & -1 & -3 \end{array}\right]$	$\left[\begin{array}{ccc\|c} 1 & 1 & 0 & 3 \\ 0 & 0 & 1 & 3 \end{array}\right]$

This third example is similar to the one in Table 5.2, but we removed the third equation. The Gauss-Jordan algorithm gets us as close to the identity matrix as possible in the coefficient part of $[\boldsymbol{A} \mid \boldsymbol{b}]$. Notice how the columns of the identity matrix and the pivot-less columns are interspersed.

and 3, and the free variable is in column 2. The rest $m - r$ rows are zeros. We represent this as:

$$\boldsymbol{A} \in \mathbb{R}^{m \times n},\ \text{rank}(\boldsymbol{A}) = r < m < n \implies \boldsymbol{A} \xrightarrow{\text{RREF}} \left[\begin{array}{c} \boldsymbol{I}_r \oplus \boldsymbol{F}_{r \times (n-r)} \\ \hline \boldsymbol{0}_{(m-r) \times n} \end{array}\right] \in \mathbb{R}^{m \times n}$$

Summary We see that the Gauss-Jordan algorithm tries very hard to get to an identity matrix as the RREF. If the matrix has full rank, it succeeds fully in the case of square and tall matrices, with a clean identity matrix visible in the RREF. In the case of a wide matrix, it may not be able to cleanly separate the identity matrix from the pivot-less columns. If the matrix is rank deficient, in general, the RREF will have the columns of the identity matrix (which are pivot columns) interspersed with the free-variable columns. In any case, the RREF is unique for a matrix, unlike REF, the output of Gaussian elimination, which depends on the order in which the elementary row operations are performed.

Earlier, we stated that the row and column ranks were the same for any matrix, and that we would provide a hint to prove it. This whole section is the hint, plus the fact that elementary row operations do not change row or column rank. Neither do elementary column operations. There is nothing that prevents us from doing both, one after the other, if we want, and then asserting the fact that the row and column ranks do not change.

5.3 Inverse of a Matrix

We came across the inverse of matrices in a few previous occasions, starting with the nomenclature, where we defined it as that matrix which, when multiplied with the original gives us the identity matrix.

We again saw it in the properties of determinants: If a matrix has zero determinant, it cannot be inverted. While talking about elementary matrices, we spoke of their inverses as those matrices that would reverse the effect of multiplying with the original matrices. Let's recap and expand on what we have learned about inverses.

5.3.1 Invertibility

A square matrix is invertible if we can compute its inverse, obviously. Invertibility can be expressed in a variety of ways, as listed below. For a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, if $\boldsymbol{A^{-1}}$ exists, we may say:

- $\boldsymbol{A}$ is invertible.
- $\boldsymbol{A}$ is *not* singular.
- $\boldsymbol{A}$ is *not* degenerate.
- $\boldsymbol{A}$ has n pivots in its REF.
- $\boldsymbol{A}$ has full rank: $\text{rank}(\boldsymbol{A}) = n$.
- The system of linear equations $\boldsymbol{Ax} = \boldsymbol{b}$ has a unique solution.
- $\boldsymbol{A}$ represents a linear automorphism. (See §3.3, page 52.)

These are the statements about the invertibility of $\boldsymbol{A}$ that we have learned so far. We have quite a few more of such statements, related to the linear independence of the rows and columns of $\boldsymbol{A}$, the geometry of its so-called fundamental spaces, its eigenvalues *etc.* We will list them later, when appropriate.

5.3.2 Properties of Inverses

In addition to the various statements about invertibility, we also have a list of some interesting properties of inverses.

1. Inverse of an inverse is the original matrix: $\left(\boldsymbol{A^{-1}}\right)^{-1} = \boldsymbol{A}$
2. **Scaling**: When we multiply a matrix by a scalar, its inverse gets multiplied by the reciprocal of the scalar:

$$(s\boldsymbol{A})^{-1} = s^{-1}\boldsymbol{A^{-1}} \text{ because}$$
$$(s\boldsymbol{A})\, s^{-1}\boldsymbol{A^{-1}} = (ss^{-1})\boldsymbol{A}\boldsymbol{A^{-1}} = 1\,\boldsymbol{I} = \boldsymbol{I}$$

3. **Product Rule:** Similar to the product rule of transposes, the inverse of a product is the product of the inverses of the factors in the reverse order:

$$(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1} \text{ because } (\boldsymbol{AB})\boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$$
$$= \boldsymbol{A}\left(\boldsymbol{BB}^{-1}\right)\boldsymbol{A}^{-1} = \boldsymbol{AIA}^{-1} = \boldsymbol{AA}^{-1} = \boldsymbol{I}$$

We used this property in constructing the lower triangular matrix $\boldsymbol{L}$ (on $\boldsymbol{A} = \boldsymbol{LU}$) from the inverses of elementary matrices. (See §4.3.4, page 81.)

4. **Involutory Matrix:** A matrix whose inverse is the same as itself is called an involutory matrix. If $\boldsymbol{P}$ is involutory, $\boldsymbol{P}^2 = \boldsymbol{I}$, as the single-row-exchange permutation matrices.

5. The determinant of the inverse of a matrix is the reciprocal of its determinant:

$$\left|\boldsymbol{A}^{-1}\right| = \frac{1}{|\boldsymbol{A}|}$$

5.3.3 Gauss-Jordan Method to Compute Inverse

The inverse of $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, if it exists, is easily computed as:

$$\left[\boldsymbol{A} \mid \boldsymbol{I}\right] \xrightarrow{\text{RREF}} \left[\boldsymbol{I} \mid \boldsymbol{A}^{-1}\right]$$

We have two ways of seeing why this works. The first explanation below is more elegant and less complicated than the second one that follows.

A Using Block Multiplication

The first and most elegant explanation of why Gauss-Jordan elimination gives $\boldsymbol{A}^{-1}$ uses the **Special Case of Block-wise Multiplication** we discussed on page 38. If we have a block matrix $\left[\boldsymbol{A} \mid \boldsymbol{I}\right] \in \mathbb{R}^{n\times 2n}$, we can multiply it by another matrix $\boldsymbol{E} \in \mathbb{R}^{n\times n}$ as:

$$\boldsymbol{E}\left[\boldsymbol{A} \mid \boldsymbol{I}\right] = \left[\boldsymbol{EA} \mid \boldsymbol{EI}\right] = \left[\boldsymbol{EA} \mid \boldsymbol{E}\right]$$

If $\boldsymbol{E}$ happens to be the elementary matrix that takes $\boldsymbol{A}$ to its RREF, then the process of multiplying by $\boldsymbol{E}$ on the left is the same as performing Gauss-Jordan elimination: $\boldsymbol{E}\left[\boldsymbol{A} \mid \boldsymbol{I}\right] = \text{RREF}\left(\left[\boldsymbol{A} \mid \boldsymbol{I}\right]\right)$.

And since we start with a full-rank, invertible $\boldsymbol{A}$, its RREF is nothing but $\boldsymbol{I}$. Putting it all together, we write:

$$\text{RREF}\left(\left[\boldsymbol{A} \mid \boldsymbol{I}\right]\right) = \left[\boldsymbol{EA} \mid \boldsymbol{E}\right] = \left[\boldsymbol{I} \mid \boldsymbol{E}\right]$$

Comparing the left blocks in the equation above, we get $\boldsymbol{EA} = \boldsymbol{I} \implies \boldsymbol{E} = \boldsymbol{A^{-1}}$, and we can see $\boldsymbol{E}$ sitting as the right block in the final term. In other words,

$$\left[\boldsymbol{A} \mid \boldsymbol{I}\right] \xrightarrow{\text{RREF}} \left[\boldsymbol{I} \mid \boldsymbol{A^{-1}}\right]$$

B Using Super-Augmented Matrices

The previous explanation was elegant because we worked at the level of matrices (using matrix-algebra, as it were), rather than descend to the column/row/element level. If we need another explanation, we can get it by looking at $\boldsymbol{A}$ as encoding one system of equations $\boldsymbol{Ax} = \boldsymbol{b}$, and Gauss-Jordan giving us the solution. If we have multiple systems of equations with the same coefficient matrix, we can perform Gauss-Jordan elimination on all of them in one go.

Limiting ourselves to the simple case of full-rank, square coefficient matrix $\boldsymbol{A}$ so that $\boldsymbol{A^{-1}}$ exists, if we have $\boldsymbol{Ax} = \boldsymbol{b_1}$ and $\boldsymbol{Ax} = \boldsymbol{b_2}$, we could write a "super" augmented matrix like, $\left[\boldsymbol{A} \mid [\boldsymbol{b_1}\, \boldsymbol{b_2}]\right]$. If the RREF of this augmented matrix becomes $\left[\boldsymbol{I} \mid [\boldsymbol{b_1'}\, \boldsymbol{b_2'}]\right]$, we know that $\boldsymbol{b_1'}$ is the solution to the first system, and $\boldsymbol{b_2'}$ that of the second one. In other words:

$$\left[\boldsymbol{A} \mid [\boldsymbol{b_1}\, \boldsymbol{b_2}]\right] \xrightarrow{\text{RREF}} \left[\boldsymbol{I} \mid [\boldsymbol{b_1'}\, \boldsymbol{b_2'}]\right] \implies \boldsymbol{Ab_1'} = \boldsymbol{b_1} \text{ and } \boldsymbol{Ab_2'} = \boldsymbol{b_2}$$

More generally, we could write this complicated super system as (again, limiting ourselves to $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{B'} \in \mathbb{R}^{n \times n}$):

$$\boldsymbol{AX} = \boldsymbol{B} \implies \left[\boldsymbol{A} \mid \boldsymbol{B}\right] \xrightarrow{\text{RREF}} \left[\boldsymbol{I} \mid \boldsymbol{B'}\right]$$

Noting that $\boldsymbol{X}$ is a symbolic matrix (x_{ij} are symbols, not numbers) and $\boldsymbol{B'}$ is the solution that satisfies the equations, we can confidently write:

$$\boldsymbol{AB'} = \boldsymbol{B}$$

We know that there is nothing special about this $\boldsymbol{B}$. In other words, if we have a $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, we can build any number of systems of linear equations by an arbitrary choice of $\boldsymbol{B}$. Let's therefore choose

$\boldsymbol{B} = \boldsymbol{I}$, and investigate what it says. With this choice, we get:

$$\boldsymbol{AB'} = \boldsymbol{I}$$

What does this equation mean? It says that $\boldsymbol{B'}$ is the matrix that, when multiplied with $\boldsymbol{A}$ results in the identity matrix $\boldsymbol{I}$, which is the definition of the inverse of $\boldsymbol{A}$. In other words, $\boldsymbol{B'} = \boldsymbol{A^{-1}}$

Remembering again that $\boldsymbol{B'}$ is merely the part of the result of Gauss-Jordan running on $\big[\boldsymbol{A} \mid \boldsymbol{B}\big]$, and that $\boldsymbol{B}$ now is $\boldsymbol{I}$, we get the same cryptic recipé for $\boldsymbol{A^{-1}}$:

$$\big[\boldsymbol{A} \mid \boldsymbol{I}\big] \xrightarrow{\text{RREF}} \big[\boldsymbol{I} \mid \boldsymbol{A^{-1}}\big]$$

Elaborating on it: To find the $\boldsymbol{A^{-1}}$,

1. Construct the augmented matrix $\big[\boldsymbol{A} \mid \boldsymbol{I}\big]$.
2. Run the Gauss-Jordan algorithm on it.
3. If we get the identity matrix $\boldsymbol{I}$ in the place of the original matrix $\boldsymbol{A}$ in $\big[\boldsymbol{A} \mid \boldsymbol{I}\big]$, the augmented part will now contain the inverse $\boldsymbol{A^{-1}}$.
4. If we cannot get $\boldsymbol{I}$, it means $\boldsymbol{A}$ is singular.

We see that Gauss-Jordan elimination (which is an extension of Gaussian elimination) gives us a recipé for computing the inverse of any invertible matrix $\boldsymbol{A}$. It so happens that the Gauss-Jordan method of finding the inverse is the one that is computationally most efficient.

5.3.4 Cofactors to Compute Inverse

In addition to the Gauss-Jordan algorithm, we also have an analytic formula, similar to the Laplace expansion for determinant computation, Eqn (3.5), to directly compute the inverse, using the same cofactors we defined in §3.3.2, page 57.

Refreshing our memory, every element in a matrix $\boldsymbol{A} = \big[a_{ij}\big]$ has *cofactor* C_{ij}, which is the determinant of the submatrix obtained by deleting the i^{th} row and j^{th} column of $\boldsymbol{A}$, with an associated sign. We can put these cofactors (being just numbers) in a matrix of the same size to get $\boldsymbol{C} = [C_{ij}]$. The *transpose* of this matrix of cofactors goes by many names, such as the *adjugate*, the *classical adjoint* and the *adjoint* matrix.

Once we have the matrix of cofactors, we can compute its transpose (which is the adjugate matrix) and then the inverse as:

$$\boldsymbol{A^{-1}} = \frac{1}{|\boldsymbol{A}|}\boldsymbol{C^{\intercal}} \tag{5.1}$$

Although this equation looks compact, it is horribly expensive computationally. As we can see from the formula, when $|\boldsymbol{A}| = 0$, we cannot compute $\boldsymbol{A^{-1}}$ and the matrix $\boldsymbol{A}$ is singular.

5.3.5 Inverse of a 2×2 Matrix

For a 2×2 matrix, we can apply Eqn (5.1) and write down its inverse.

$$\boldsymbol{A} = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \implies |\boldsymbol{A}| = ad - bc \text{ and } \boldsymbol{A^{-1}} = \frac{1}{ad-bc}\begin{bmatrix} d & -c \\ -b & a \end{bmatrix}$$

Remember this formula as: Swap the diagonal elements and switch the sign of the non-diagonal elements. And then divide the determinant.

5.4 Left and Right Inverses

Left inverse of a matrix $\boldsymbol{A}$ is that matrix which produces $\boldsymbol{I}$ when multiplied on the left of $\boldsymbol{A}$. Similarly for the right inverse, it is the multiplication on the right. In other words, if $\boldsymbol{BA} = \boldsymbol{I}$, then $\boldsymbol{B} = \boldsymbol{A^{-1}_{\text{Left}}}$. And, if $\boldsymbol{AC} = \boldsymbol{I}$, then $\boldsymbol{C} = \boldsymbol{A^{-1}_{\text{Right}}}$.

Although we did not specify it, the inverse $\boldsymbol{A^{-1}}$ we have been talking about is a double-sided inverse: $\boldsymbol{AA^{-1}} = \boldsymbol{I} = \boldsymbol{A^{-1}A}$. For a full-rank, square matrix, we can prove that if $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \in \mathbb{R}^{n \times n}$, $\text{rank}(\boldsymbol{A}) = n$, and $\boldsymbol{AB} = \boldsymbol{CA} = \boldsymbol{I}$, then $\boldsymbol{B} = \boldsymbol{C}$.

For "tall" or "wide" matrices (as described in §5.1.2, page 90), the situation is a bit more complicated, even if they are of full rank. Remember, for tall matrices full rank means full column rank, and for wide ones, it means full row rank. To state it with deadly mathematical precision:

$$\begin{aligned} \text{Tall, full rank} &\implies \boldsymbol{A} \in \mathbb{R}^{m \times n}, m > n, \text{rank}(\boldsymbol{A}) = r = n \\ \text{Wide, full rank} &\implies \boldsymbol{A} \in \mathbb{R}^{m \times n}, m < n, \text{rank}(\boldsymbol{A}) = r = m \end{aligned}$$

For tall matrices, what is the shape of $\boldsymbol{A^{\intercal}A}$? It is a square matrix of size $n \times n = r \times r$. It has a rank of r, as we stated in Property 7, while

listing the properties of ranks in §5.1.3, page 90. When $\boldsymbol{A}$ has full column-rank, $r = n$, and $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$ is full rank as well. This so-called Gram matrix is therefore an invertible matrix, and $\left(\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\right)^{-1}$ exists. It is also symmetric because $\left(\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\right)^\mathsf{T} = \boldsymbol{A}^\mathsf{T}\boldsymbol{A}$. With a bit of matrix algebra wizardry, we can write:

$$\left(\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\right)^{-1}\left(\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\right) = \boldsymbol{I} \implies \left(\left(\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^\mathsf{T}\right)\boldsymbol{A} = \boldsymbol{I} \tag{5.2}$$

This is of the form $\boldsymbol{B}\boldsymbol{A} = \boldsymbol{I}$, and therefore $\left((\boldsymbol{A}^\mathsf{T}\boldsymbol{A})^{-1}\boldsymbol{A}^\mathsf{T}\right)$ is something like an inverse of $\boldsymbol{A}$, when multiplied on the left. We will call this the left inverse of $\boldsymbol{A}$, $\boldsymbol{A}^{-1}_{\textbf{Left}}$. Remember, $\boldsymbol{A} \in \mathbb{R}^{m\times n}, m > n$ is not square. Although we cannot multiply $\boldsymbol{A}$ on the right with $\boldsymbol{A}^{-1}_{\textbf{Left}}$ (because of conformance requirements is met), what we will get is a matrix in $\mathbb{R}^{m\times m}$ with a rank of $n < m$. And the product cannot be an identity matrix because it is rank-deficient.

In data science, our matrices tend to be "tall," with $m \gg n$, and $\boldsymbol{A}^\mathsf{T}\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is a nice, small matrix. Moreover, it tends to be a full-rank matrix because data points are usually measurements, and one measurement is very unlikely to be a linear combination of others. $\boldsymbol{A}\boldsymbol{A}^\mathsf{T} \in \mathbb{R}^{m\times m}$, on the other hand, is a horrible, huge matrix with a rank $r = n \ll m$, which means it is hopelessly rank deficient.

When we have a "wide" matrix, however, it is the other Gram matrix, $\boldsymbol{A}\boldsymbol{A}^\mathsf{T} \in \mathbb{R}^{m\times m}$, that is the nice matrix. All we wrote down above for the left inverse will work for the right inverse with obvious and trivial changes, and altogether we get the following results:

$$\begin{aligned}
\text{Tall, full rank}: \boldsymbol{A} \in \mathbb{R}^{m\times n}, m > n, r = n &\implies \boldsymbol{A}^{-1}_{\textbf{Left}} = (\boldsymbol{A}^\mathsf{T}\boldsymbol{A})^{-1}\boldsymbol{A}^\mathsf{T} \\
\text{Wide, full rank}: \boldsymbol{A} \in \mathbb{R}^{m\times n}, m < n, r = m &\implies \boldsymbol{A}^{-1}_{\textbf{Right}} = \boldsymbol{A}^\mathsf{T}(\boldsymbol{A}\boldsymbol{A}^\mathsf{T})^{-1}
\end{aligned}$$

5.5 Cramer's Rule

Related to the matrix of cofactors and the analytic formulas for determinant and inverse of matrices (Equations (3.5) and (5.1)) is a beautifully compact and elegant (albeit computationally useless) formula for solving a system of linear equations. This formula is known as the Cramer's Rule. It states that for a system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ with a square, invertible $\boldsymbol{A}$ and $\boldsymbol{x} = [x_j]$, the solution is

given by:

$$x_j = \frac{|\boldsymbol{A_j}|}{|\boldsymbol{A}|} \tag{5.3}$$

where $\boldsymbol{A_j}$ is the matrix formed by replacing the j^{th} column of $\boldsymbol{A}$ with $\boldsymbol{b}$.

To recap, we have learned four ways of solving the system of linear equations $\boldsymbol{Ax} = \boldsymbol{b}$.

1. Do Gaussian elimination on the augmented matrix $\big[\boldsymbol{A} \mid \boldsymbol{b}\big]$, followed by back substitution.
2. Perform Gauss-Jordan elimination on the augmented matrix $\big[\boldsymbol{A} \mid \boldsymbol{b}\big]$ (which is not really different from the previous method).
3. If $\boldsymbol{A}$ is a full-rank, square matrix $\boldsymbol{Ax} = \boldsymbol{b} \implies \boldsymbol{x} = \boldsymbol{A^{-1}b}$. This is often written in some programs as $\boldsymbol{x} = \boldsymbol{A}\backslash\boldsymbol{b}$, indicating something like dividing by $\boldsymbol{A}$ on the left.
4. Use Cramer's rule, again if $\boldsymbol{A}$ is a full-rank, square matrix.

As we can see, although the methods (3) and (4) are elegant, they work only for the ideal situation of unique solutions. Besides, they are (especially the Cramer's rule method is) prohibitively expensive to compute for nontrivial matrix sizes. To top it off, their numeric stability also is questionable.

5.6 Algebraic View of Linear Algebra

We have completed algebraic view of Linear Algebra, by which we meant determining the solvability conditions of systems of linear equations and characterizing the solutions. Along the way, we touched upon the concepts of ranks and determinants, inverses and a couple of decompositions.

While looking at linear equations, we did speak of their shapes in coordinate spaces: a linear equation is a line in $\mathbb{R}^2$ and a plane in $\mathbb{R}^3$, and their solutions are the intersections of lines or planes. While this view is undoubtedly geometric, the geometric view of Linear Algebra we will start in the next part of the book is the deeper and richer structure of vector spaces and subspaces and their properties. The insights from such geometry, as we shall see, will have direct

relevance to certain algorithms, and a wider relevance to how we discuss and present our ideas in computer science.

Part III

Geometric View

6

Vector Spaces, Basis and Dimensions

To those who do not know mathematics, it is difficult to get across a real feeling as to the beauty, the deepest beauty, of nature. . . If you want to learn about nature, to appreciate nature, it is necessary to understand the language that she speaks in.

—Richard Feynman

The notion of linear combinations is central to Linear Algebra, which is why we started seeing it from the very first chapter of this book. Now that we are taking a geometric view to understanding the subject, we will deal with this notion and the associated concepts in much more detail and in a formal way.

6.1 Linear Combinations

Let's start by reminding ourselves of the basic operations on vectors (and indeed matrices):

- Vectors can be scaled. A vector multiplied by a scalar is another vector, which means the collection of vectors is closed under scalar multiplication.

- Vectors can be added. The set of vectors is closed under vector addition as well: When we add vectors, we get other vectors.

We can do both these operations together: Take a bunch of vectors, scale them and add the scaled versions, which is exactly what we mean by a linear combination. As we can see from the basic operations, the set of vectors is closed under linear combinations as well; when we take linear combinations, what we get are other vectors. Stating this rather obvious fact formally:

$$\begin{aligned} &\forall \boldsymbol{x_i} \in \mathbb{R}^n, s_i \in \mathbb{R} \text{ and any nonnegative integer } k, \\ &\qquad \boldsymbol{z} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2} + \cdots + s_k\boldsymbol{x_k} \\ &\qquad = \sum_{i=1}^{k} s_i\boldsymbol{x_i} \in \mathbb{R}^n \text{ is a linear combination} \end{aligned} \tag{6.1}$$

Let's look a few examples of linear combinations:

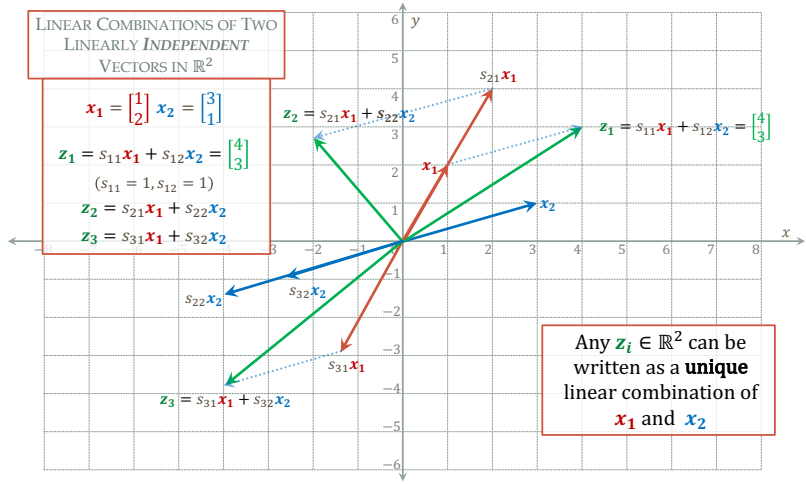


Fig. 6.1 Linear combinations of two vectors in $\mathbb{R}^2$. We can create all possible vectors in $\mathbb{R}^2$ starting from our $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$.

1. Two vectors in $\mathbb{R}^2$: In Figure 6.1, we start with two vectors $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$. The first linear combination we make is the sum $\boldsymbol{z_1} = \boldsymbol{x_1} + \boldsymbol{x_2}$, which is the same as $\boldsymbol{z_1} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2}$, with the scalar multipliers s_1 and s_2 set to 1. Then we change the values of the scalars s_1 and s_2 so as to get other vectors. As we can see, we can get to any vector in $\mathbb{R}^2$ by the appropriate choice of s_1 and s_2.

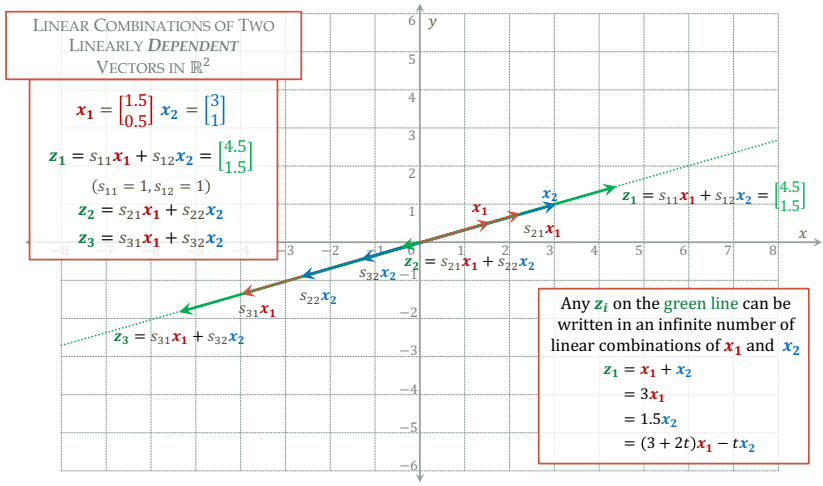


Fig. 6.2 Linear combinations of two vectors in $\mathbb{R}^2$. Starting from our $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, we cannot get out of the line defined by the vectors, no matter what scaling factors we try. The vectors are not *linearly independent*.

2. In Figure 6.2, we again have two vectors. In this case, however, when we take all possible linear combinations, we do not seem be able get to all vectors in $\mathbb{R}^2$; we are confined to one line. In order see why, we have to first notice that the two vectors we defined were not independent of each other. Our second vector, $\boldsymbol{x_2}$ is a scalar multiple of $\boldsymbol{x_1}$: $\boldsymbol{x_2} = 2\boldsymbol{x_1}$. Therefore, when we take a linear combination $\boldsymbol{z} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2}$, what we are actually doing, in effect, is only scaling one vector.

$$\boldsymbol{z} = s_1\boldsymbol{x_1} + 2s_2\boldsymbol{x_1} = (s_1 + 2s_2)\boldsymbol{x_1} = s\boldsymbol{x_1}$$

   All the scaled versions of $\boldsymbol{x_1}$ fall in the same direction as $\boldsymbol{x_1}$.

3. In the third example. Figure 6.3, we move on to $\mathbb{R}^3$, but with vectors very similar to the first case, but with a third component (because vectors in in $\mathbb{R}^3$ have three components) equal to zero. When we take all possible linear combinations $\boldsymbol{z} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2}$, we can see that the third component of $\boldsymbol{z}$ can never be anything other than zero, and therefore all such linear combinations live on the xy plane.

4. In Figure 6.4, we have added some nonzero values as the third component of $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$. Still, the linear combinations of

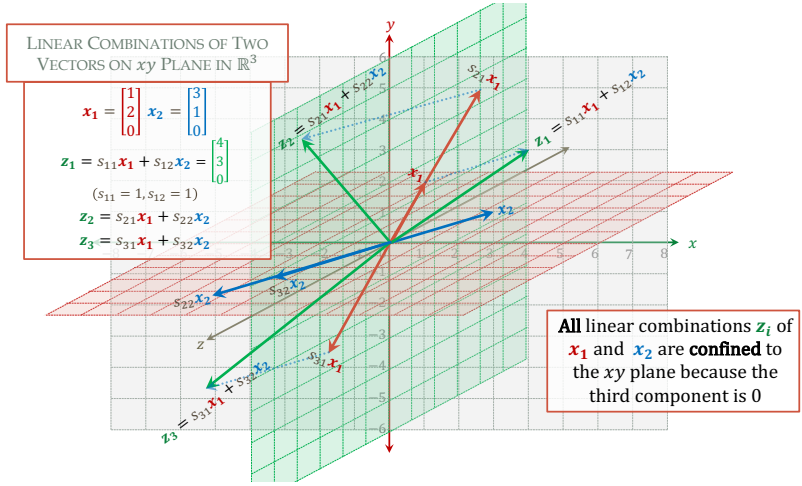


Fig. 6.3 Linear combinations of two vectors in $\mathbb{R}^3$. All the vectors we create starting from our $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ live on the xy plane. We do not have enough vectors to *span* the whole of $\mathbb{R}^3$.

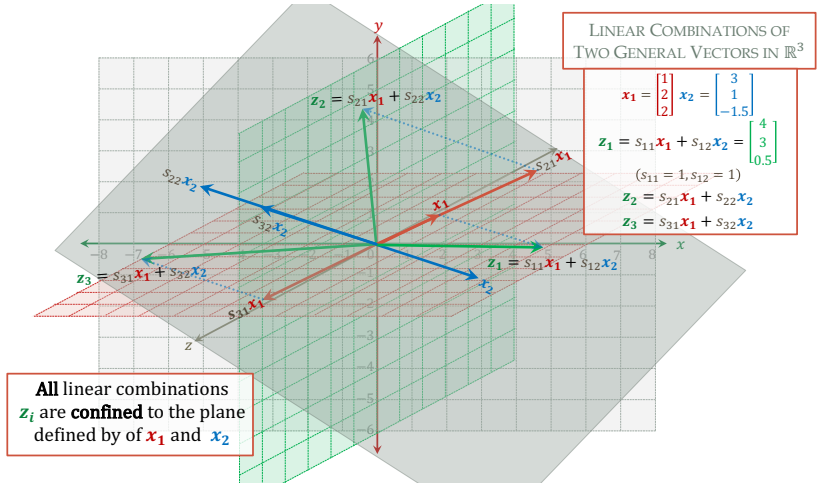


Fig. 6.4 Linear combinations of two vectors in $\mathbb{R}^3$. Here again, we do not have enough vectors to span $\mathbb{R}^3$. The linear combinations of $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ live in a plane defined by the two vectors. (Not drawn to scale.)

the two vectors, $\boldsymbol{z} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2}$, all fall on a plane defined by the two vectors, or, equivalently, by the three points $(0, 0, 0)$, $(1, 2, 2)$ and $(3, 1, -1.5)$. Two vectors are not enough to span $\mathbb{R}^3$.

5. As we can also see from Figure 6.4, we have a hard time depicting to general vectors in $\mathbb{R}^3$. In this fifth example, we are going to move to three vectors, but will not attempt to draw them. If we have the following three vectors, we still cannot get all vectors in $\mathbb{R}^3$:

$$\boldsymbol{x_1} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} \boldsymbol{x_2} = \begin{bmatrix} 3 \\ 1 \\ 0 \end{bmatrix} \boldsymbol{x_3} = \begin{bmatrix} 11 \\ 17 \\ 0 \end{bmatrix}$$

   In addition to the two vectors in Figure 6.3, we have added a third vector, but with its third component equal to 0. Now we take all possible linear combinations $\boldsymbol{z} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2} + s_3\boldsymbol{x_3}$. The third component of $\boldsymbol{z}$ will have to be zero no matter what scaling factors we try, and we are confined to the xy plane. Although it may not be obvious, $\boldsymbol{x_3}$ is a linear combination of $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, which we can see by solving for s_1 and s_2 in the equation $\boldsymbol{x_3} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2}$.

6. In the next example, we start with the two vectors in Figure 6.4 and add a third vector, which we know is a linear combination of the two. We have these three vectors in $\mathbb{R}^3$:

$$\boldsymbol{x_1} = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} \boldsymbol{x_2} = \begin{bmatrix} 3 \\ 1 \\ -1.5 \end{bmatrix} \boldsymbol{x_3} = \begin{bmatrix} 9 \\ 8 \\ 3 \end{bmatrix} = 3\boldsymbol{x_1} + 2\boldsymbol{x_2}$$

   The three vectors, being linear combinations of each other, are on the same plane in $\mathbb{R}^3$. All other linear combinations we can make with them will fall on the same plane.

7. In the last example, we will start from the simpler vectors in Figure 6.3, which live on the xy plane in $\mathbb{R}^3$. We add a simple third vector with a component in the third direction.

$$\boldsymbol{x_1} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} \boldsymbol{x_2} = \begin{bmatrix} 3 \\ 1 \\ 0 \end{bmatrix} \boldsymbol{x_3} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

   Note that none of these three vectors can be written as the linear combination of the other two. Now we are no longer confined to any plane. We can indeed create any vector in $\mathbb{R}^3$ as a linear combination of these three.

We have listed seven examples with many permutations of linear combinations of vectors above. In order to make sense of them, and to introduce some terms that we will define very soon, let's take stock of what we have done. In the first example, we had two *linearly independent* vectors in $\mathbb{R}^2$. Two linearly independent vectors are enough to *span* $\mathbb{R}^2$, which is the collection of all possible vectors with two components. We will soon call $\mathbb{R}^2$ a *vector space*. In the second example, we had two vectors, but they were not linearly independent. What they spanned was a subset of $\mathbb{R}^2$, which we will call a *subspace*. In the third example, we went on to $\mathbb{R}^3$, but with two linearly independent vectors, which were not enough to span the whole vector space. Instead, they spanned a subspace that was the xy plane. In the fourth example, we again had only two vectors in $\mathbb{R}^3$, spanning a subspace. In the fifth and sixth examples, we did add the third vector, but it was not linearly independent of the other two, and therefore, the spans of the three vectors were still only subspaces. Finally, in the seventh example, we had three good, linearly independent vectors that could span the entirety of $\mathbb{R}^3$.

Our job now is to define the unfamiliar terms in the previous paragraph, express them in mathematical language, and generalize them to higher dimensions, $\mathbb{R}^n$. Clearly, we will not be able to visualize them, but that inability will not prevent us from developing insights and intuitions that will apply to all dimensions and datasets.

6.1.1 Spans of Vectors

Definition: The set of all possible linear combinations of a given set of vectors is their span. To write it mathematically,

$$\text{Given } k \text{ vectors } \boldsymbol{x_i} \in \mathbb{R}^n, \text{span}(\{\boldsymbol{x_i}\}) = \left\{ \boldsymbol{z} \mid \boldsymbol{z} = \sum_{i=1}^{k} s_i \boldsymbol{x_i} \right\} \quad (6.2)$$

$$\text{for any } k \text{ scalars } s_i \in \mathbb{R}$$

In Figure 6.1, the span of the red and blue vectors is all possible vectors in $\mathbb{R}^2$. In Figure 6.2, the span of the two vectors is a much smaller subset: Only those vectors that are in the same direction (the dotted line in the figure) as the original two, collinear vectors.

6.1.2 Cardinality

Definition: The number of vectors in the span is called its cardinality. Cardinality is just a fancy name for the size or the number of elements in a set. It applies to our notion of span of vectors as well as the set of vectors that define a span, as in Eqn (6.2) for instance. The cardinality of the former is indeed infinity (as there are an infinity of linear combinations we can form), while the cardinality of the latter (the vectors of which we are forming the linear combinations) is k.

6.1.3 Linear Independence

Definition: The vectors in a set are linearly independent if we cannot write any one of them as a linear combination of the others. An equivalent definition, again in mathematical lingo, is that the necessary and sufficient condition for a given set of k vectors $\boldsymbol{x_i} \in \mathbb{R}^n$ to be linearly independent is that there be no set of scalars $s_i \in \mathbb{R}$ that are not all zero such that:

$$\sum_{i=1}^{k} s_i \boldsymbol{x_i} = \mathbf{0} \tag{6.3}$$

Clearly, with all $s_i = 0$, the sum is always the zero vector. What we are looking for is a set of scaling factors with at least some nonzero numbers such that the sum is $\mathbf{0}$. If we can find such a set, the vectors are not linearly independent. They are linearly dependent.

Looking at the example in Figure 6.2, we have $\boldsymbol{x_2} = 2\boldsymbol{x_1} \implies \boldsymbol{x_2} - 2\boldsymbol{x_1} = \mathbf{0}$, the zero vector as a linear combination with nonzero scalars, which means $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are not linearly independent. Similarly, for the sixth example in the list of examples above, $\boldsymbol{x_3} = 3\boldsymbol{x_1} + 2\boldsymbol{x_2}$ or $3\boldsymbol{x_1} + 2\boldsymbol{x_2} - \boldsymbol{x_3} = \mathbf{0}$, again the zero vector as a a linear combination with nonzero scalars, implying linear dependence.

6.2 Vector Spaces and Subspaces

For our purpose in this book, we will define a vector space as a set of all possible vectors. In fact, while defining vectors, their operations and properties, we were indirectly defining vector spaces as well. For instance, we said when we scale a vector or add two vectors, we get another vector, which is the same as saying that the *set* of vectors

is closed under scalar multiplication and vector addition. We also listed the associativity, commutativity and distributivity properties of vector operations, which are, in fact properties of the underlying vector space. Although mathematical puritans may look askance at such sloppiness, we will consider it a pardonable foible of computer scientists. We will, however, present the full and precise definition in all its mathematical glory in §8.1, page 145.

6.2.1 Vector Space

Definition: The set $\mathcal{V}$ of all vectors $\boldsymbol{x} \in \mathbb{R}^n$ is a vector space (AKA linear space) if the operations of a scalar multiplication and a vector addition (with the right properties as specified earlier) are defined for $\boldsymbol{x}$ and $\mathcal{V}$ is closed under the operations.

As we mentioned earlier, any mathematical object that can fulfill the requirements of vectors is a vector, as far as Linear Algebra is concerned. And a closed set of any such vectors would be a vector space. Here are some examples of vector spaces:

Coordinate Space: The usual Euclidean space of points is a vector space, treating the numbers and coordinates in them as vectors. For instance, the good old number line (same as $\mathbb{R}$) is a vector space. If we treat the (x, y) coordinates in R^2 as vectors, the two-dimensional plane, being a collection of such coordinates, is a vector space. Such spaces ($\mathbb{R}^n$) are called coordinate spaces, naturally.

Space of Matrices: Matrices also have operations identical to vectors, as we defined them. Vectors are indeed single-column matrices. Therefore, we can treat a set of matrices (of the same size) as a vector space. For instance, if we create a set $\mathcal{M} = \{\boldsymbol{A} \mid \boldsymbol{A} \in \mathbb{R}^{m \times n}\}$, it would be a vector space.

Complex Numbers: The set of complex numbers ($\mathbb{C}$) is a vector space as well, very similar to the coordinate space of $\mathbb{R}^2$.

6.2.2 Subspace

Definition: Given a vector space $\mathcal{V}$, a subset of it is a vector (or linear) subspace $\mathcal{S}$ if it is closed under the same operations of a scalar multiplication and a vector addition defined in $\mathcal{V}$. In other words, for

any vectors $\boldsymbol{x_i} \in \mathcal{S}$, the same operations (as defined in $\mathcal{V}$) applied on them results in other vectors $\boldsymbol{y_i} \in \mathcal{S}$.

In practice, we can say that all linear combinations of the vectors in $\mathcal{S}$ have to be in $\mathcal{S}$ for it to qualify as a subspace. Therefore, a practical definition of a subspace $\mathcal{S} \subseteq \mathcal{V}$ is that it is the span of a given set of vectors in $\mathcal{V}$.

In Figure 6.2, we have two vectors, $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, that were scaled versions of each other. All their linear combinations (which are, in fact, just scaled versions of $\boldsymbol{x_1}$) form a line in $\mathbb{R}^2$. This line represents a subspace because, when we add any two vectors in this line, we get another vector that is collinear with it. Note that the zero vector has to be a part of any subspace because the trivial linear combination of the vectors (that define the span, which is the subspace) with zero scalars is still a linear combination and has to be in the subspace.

If we have two subspaces, their intersection is a subspace, but their union is not. Let's formally prove the intersection part of this statement.

1. Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be the subspaces and $\mathcal{I} = \mathcal{S}_1 \cap \mathcal{S}_2$, with $\boldsymbol{x_1}, \boldsymbol{x_2} \in \mathcal{I}$.
2. By the definition of the intersection, $\boldsymbol{x_1}, \boldsymbol{x_2} \in \mathcal{S}_1$ and $\boldsymbol{x_1}, \boldsymbol{x_2} \in \mathcal{S}_2$.
3. Since $\mathcal{S}_1$ is a subspace, for any two scalars, s_1 and s_2, $s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2} \in \mathcal{S}_1$.
4. Since $\mathcal{S}_2$ is a subspace, for the same scalars, $s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2} \in \mathcal{S}_2$ as well.
5. Therefore, $s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2} \in \mathcal{I}$, since it is in both $\mathcal{S}_1$ and $\mathcal{S}_2 \implies \mathcal{I} = \mathcal{S}_1 \cap \mathcal{S}_1$ is a subspace.

6.2.3 Orthogonal Subspaces

Orthogonality in Linear Algebra is a generalization of perpendicularity in geometry; the latter applies to lines or other shapes where the concept of angles makes sense. Two lines are perpendicular when the angle between them is $90°$. On the other hand, two vectors are orthogonal when their dot product is zero. In our definition of vectors for the purpose, there is little difference between orthogonal vectors and perpendicular vectors, but there can be, with other definitions of vectors and their dot products. We shall, therefore, never again talk of perpendicular vectors.

We can extend the concept of orthogonality to vector subspaces as well. Two subspaces are orthogonal to each other when every vector in one subspace is perpendicular to every vector in the other one.

Here are two subspaces: The xy plane and the yz plane in $\mathbb{R}^3$. Are they orthogonal subspaces? From the geometric shapes, they look like two planes at right angles to each other. But is every vector in the xy subspace orthogonal to those in the yz subspace? Clearly not, because we have a whole line of vectors along the y axis that are in both subspaces. They are clearly not orthogonal to themselves. Any two subspaces with nontrivial (meaning more than the zero vector, $\mathbf{0}$) intersection, therefore, cannot be orthogonal to each other. Note that all subspaces have to contain the zero vector.

To look at a positive example, the xy plane and the z axis can be considered two subspaces. They are indeed orthogonal to each other. Every vector on the xy plane is orthogonal to every vector along the z axis. Here is a generalized proof using symbolic vectors.

1. A general vector on the xy plane and a general vector along the z axis have the form:

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \quad \boldsymbol{z} = \begin{bmatrix} 0 \\ 0 \\ z \end{bmatrix}$$

2. The dot product between any vector on the xy plane and any vector along the z axis is therefore:

$$\boldsymbol{x} \cdot \boldsymbol{z} = \boldsymbol{x}^{\mathsf{T}} \boldsymbol{z} = 0 \implies \boldsymbol{x} \perp \boldsymbol{z}$$

3. Therefore, by the definition of orthogonal subspaces, xy plane $\perp$ z axis.

Both perpendicularity and orthogonality are indicated by the same symbol $\perp$.

6.3 Basis and Dimensions

6.3.1 Basis

Definition: A basis of a vector space or a vector subspace is a set of vectors that meet two criteria:

1. That they span the vector space or the vector subspace.
2. That they are linearly independent.

Notational Abuse

For our purposes in Linear Algebra, $\mathbb{R}^2, \mathbb{R}^3, \mathbb{R}^n$ *etc.* are *vector spaces*, which means they are collections or sets of all possible vectors of the right dimensions. $\mathbb{R}^2$, for instance, is a collection of all vectors of the kind

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \text{ with } x_1, x_2 \in \mathbb{R}$$

and nothing else.

However, as we saw, $\mathbb{R}^2$ also is a coordinate space—the normal and familiar two-dimensional plane where we have points with (x, y) coordinates. Because of such familiarity, we may switch lightly between the vector space that is $\mathbb{R}^2$ and the coordinate space that is $\mathbb{R}^2$. We did it, for instance, when speaking of the span of a single vector which is a *line*. In a vector space, there is no line, there are no points, only vectors.

Similarly when we spoke of vector subspaces, we spoke of the xy plane in $\mathbb{R}^3$ without really distinguishing it from the vector space $\mathbb{R}^3$.

Since the vectors in $\mathbb{R}^2$ or $\mathbb{R}^3$, as we described them so far, all have components (or elements) that are identical to the coordinates of the points in the coordinate space, this abuse of notation goes unnoticed. Soon, we will see that the coordinates are artifacts of the basis that we choose. It just so happens that the most natural and convenient basis vectors are indeed the ones that will give coordinates as components.

Ultimately, however, it may be a difference without a distinction, but it is still good to know when we are guilty of notational abuse so that we may be careful to avoid mistakes arising from it.

The vectors in the basis are called basis vectors, obviously. As we can see, the concept of basis builds on the related concepts of linear combinations and spans of vectors. We can define basis in a variety of ways. Here is another definition: A basis of a vector space or vector subspace is the minimum set of vectors that span it. Yet another one: A set of vectors in a vector space (or subspace) is called a basis if every element in it can be written as a unique and finite linear combination of them. All these definitions are indeed equivalent.

Illustrating it with a couple of examples and counterexamples:

- In Figure 6.1 illustrating linear combinations, we have two vectors and linear combinations like $\boldsymbol{z}$:

$$\boldsymbol{x_1} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \text{ and } \boldsymbol{x_2} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \quad \text{Test vector: } \boldsymbol{z} \in \mathbb{R}^2 = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$$

  $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ span all of $\mathbb{R}^2$ and are linearly independent. Therefore they form a basis for the vector space $\mathbb{R}^2$. Given any

vector $\boldsymbol{z} \in \mathbb{R}^2$, we can find unique scalars s_1 and s_2 such that $\boldsymbol{z} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2}$.

- In Figure 6.2, we have two candidate basis vectors $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$:

$$\boldsymbol{x_1} = \begin{bmatrix}1.5\\0.5\end{bmatrix} \text{ and } \boldsymbol{x_2} = \begin{bmatrix}3\\1\end{bmatrix} \quad \text{Test vector: } \boldsymbol{z} \in \mathbb{R}^2 = \begin{bmatrix}5\\1\end{bmatrix}$$

  $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ do not span all of $\mathbb{R}^2$ and are not linearly independent. Therefore they are not a basis for the vector space $\mathbb{R}^2$. Given our test vector $\boldsymbol{z}$, we are not able to find scalars s_1 and s_2 such that $\boldsymbol{z} = s_1\boldsymbol{x_1} + s_2\boldsymbol{x_2}$.

- However, considering the subspace indicated by the line in the same Figure 6.2, are the two vectors in the previous example a basis? The vectors $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ do span this subspace, but they are still not a basis because they are not linearly independent. For this reason, a new vector $\boldsymbol{z_2}$, which is in this subspace, can be expressed as an infinite number of linear combinations of $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$.

- The reason why the two vectors did not form a basis for the subspace in the previous example was that we had too many vectors: for a subspace formed as a span of one vector (which is a line of vectors), we need only one vector in the basis. Similarly, for a subspace that looks like a plane (span of two vectors) as in Figure 6.4, we need exactly two vectors in the basis. If we had a third vector, it is necessarily a linear combination of the other two that are already in the basis, and is therefore not linearly independent.

6.3.2 Dimension

Definition: The dimension of a vector space or subspace is the number of vectors in its basis. Or, in fancier language, the dimension is the cardinality of any basis of the space or subspace.

Note that a vector space or vector subspace (which we may as well start calling spaces and subspaces, dropping the vector part, now that we know them well enough) may have many different bases, but all of them will have the same cardinality. The dimension of a space or a subspace is an immutable property.

Earlier, we spoke of the orthogonal subspaces. We can restate the orthogonality in terms of their bases: Two subspaces are orthogonal to each other if all the basis vectors of one are orthogonal to all the basis vectors of the other.

In $\mathbb{R}^3$, we spoke of the xy plane as a subspace being orthogonal to the z axis (taking the liberty to mix vector and coordinate spaces). The dimension of the first space is 2 and the second one is 1, adding up to the dimension of the containing space $\mathbb{R}^3$. We saw that the xy plane and the yz plane do not form orthogonal subspaces, and one reason is that their dimensions add up to 4, which is more than the dimension of $\mathbb{R}^3$.

In $\mathbb{R}^4$, however, we will be able to find two-dimensional subspaces that are orthogonal to each other. Remember that subspaces cannot have nontrivial (nonzero) intersections. Therefore, we can have two planes intersecting at a point in four dimensions!

6.3.3 Orthogonal Complements

Definition: If we have a subspace $\mathcal{S} \subset \mathbb{R}^n$ of dimension r, then its orthogonal complement $\mathcal{S}^\perp$ is the collection of all vectors $\boldsymbol{x} \in \mathbb{R}^n$ that are orthogonal to *all* vectors in $\mathcal{S}$.

$$\mathcal{S}^\perp = \left\{ \boldsymbol{x} \mid \boldsymbol{x}^\mathsf{T} \boldsymbol{y} = 0 \ \forall \ \boldsymbol{y} \in \mathcal{S} \right\}$$

As a consequence, the dimension of $\mathcal{S}^\perp = n - r$ so that the dimensions of the subspace and its orthogonal complement add up to the dimension of the containing space.

In the earlier example of orthogonal subspaces, the xy plane and the z axis (in the coordinate space $\mathbb{R}^3$) are, in fact, more than orthogonal to each other. They are orthogonal complements, which means there are no other dimensions left in the space beyond what is accounted for by these two. To look at a counter example, the x and z axes define orthogonal subspaces, but not orthogonal complements.

Keep in mind that although all the dimensions are accounted for when we think of the orthogonal subspaces (such as the xy plane and the z axis in $\mathbb{R}^3$), there are still plenty of vectors not in either. In fact, most vectors in $\mathbb{R}^3$ are not on the xy plane or the z axis; they are linear combinations of the ones in these to subspaces.

6.4 Geometry of Linear Equations

Earlier, we listed a set of equations in Table 4.1 and visualized some of them in Figure 4.1, as part of algebraic view of solving equations. Our linear equations formed lines in two-dimensional xy planes (which are *coordinate* spaces, not *vector* spaces, underscoring the need for care in distinguishing between them as described in the box on "**Notational Abuse**"). Now we want to look at the equations again as vectors in some vector space, where we present the geometry in a completely different way.

Looking at the system of linear equations $\boldsymbol{Ax} = \boldsymbol{b}$ one row at a time, as an equation making a shape in the coordinate space is the row picture. Our equations in Figure 4.1 have two variables each, and the row picture produces visualizations in the coordinate space of $\mathbb{R}^2$. In general, for $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, the row picture works in $\mathbb{R}^n$, n being the number of variables. Our new way of looking at the equations, which we will call the column picture, will mean that we are working in $\mathbb{R}^m$, of dimension the same as the number of rows in $\boldsymbol{A}$, or, equivalently, the number of equations in the system. In our examples in Figures 4.1 and 6.5 to 6.7, if the row picture looks simpler than the column picture, it is only because we have two variables, and potentially more than two equations. As the coefficient matrix $\boldsymbol{A}$ becomes large, both pictures become equally challenging to visualize, but the column view gives us deeper insights.

In Figure 6.5, we have our favorite system of two equations, this time shown as a linear combination of vectors that form the columns of the coefficient matrix $\boldsymbol{A}$. Here is the system:

$$\begin{aligned} x + y &= 5 \\ x - y &= 1 \end{aligned} \qquad \boldsymbol{Ax} = \boldsymbol{b} \quad \boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \ \boldsymbol{b} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$$

By the column picture of matrix multiplication, we know that the product $\boldsymbol{Ax}$ is a linear combination of the columns of $\boldsymbol{A}$, taken with the scaling factors in $\boldsymbol{x}$, which are x and y. The system of equations $\boldsymbol{Ax} = \boldsymbol{b}$ says that the solution is the right values for the scalars in the linear combination of columns of $\boldsymbol{A}$ that will give is $\boldsymbol{b}$. How we find these right values as depicted in Figure 6.5. Note that we had to relabel the axes (in Figures 6.5 to 6.7) as Directions 1 and 2 because they are indicators in a vector space now, not in the coordinate space. In particular, the x and y in our equations have nothing to do with

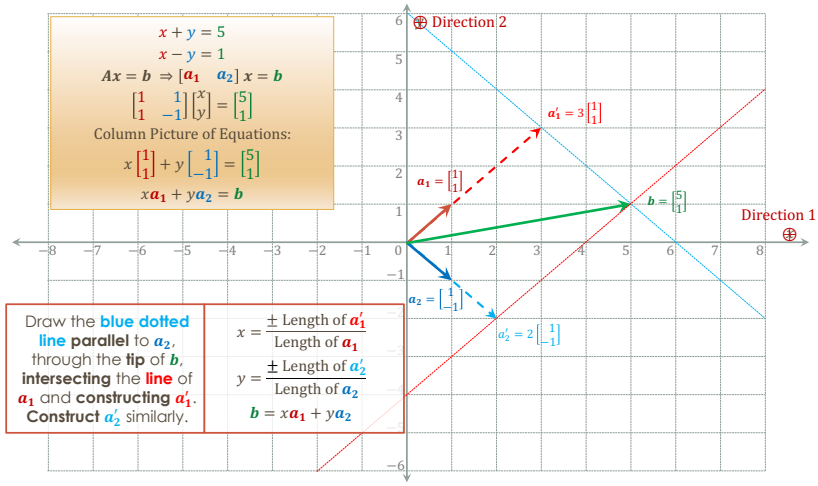


Fig. 6.5 Two linear equations on two unknowns with a unique solution. The scalars for the column vectors of $\boldsymbol{A}$ to produce $\boldsymbol{b}$ as linear combinations are the solution to the system of equations. Note that the length of $\boldsymbol{a_i'}$ is signed (as indicated by $\pm$): It is positive if $\boldsymbol{a_i'}$ is in the same direction $\boldsymbol{a_i}$, negative otherwise.

these directions, once again highlighting the need for care described in the box above on notational abuse.

To convince ourselves that this system does have a solution in this case, let's outline, as a series of steps, or an algorithm of sorts, how we can get to the right values for the scalars[1] referring to Figure 6.5:

1. Draw a line parallel to the second blue vector (the second column of $\boldsymbol{A}$, $\boldsymbol{a_2}$), going through the tip of the green $\boldsymbol{b}$ vector. It is shown in blue as a thin dashed line.

2. Scale the first red vector (the first column of $\boldsymbol{A}$, $\boldsymbol{a_1}$) to reach this line. The scaling required tells us what the scalar should be. For our equations, the scalar for $\boldsymbol{a_1}$ is $x = 3$ in $\boldsymbol{b} = x\boldsymbol{a_1} + y\boldsymbol{a_2}$.

3. Similarly, draw the red dashed line parallel to $\boldsymbol{a_1}$, going through the tip of the green $\boldsymbol{b}$ vector.

[1] In listing these steps, we break our own rule about notational abuse: We talk about drawing lines in the vector space, which we cannot do. A vector space contains only vectors; it does not contain lines. It is coordinate spaces that contain lines. This predicament of ours shows the difficulty in staying absolutely rigorous about concepts. Perhaps pure mathematical rigor for its own sake is not essential, especially for an applied field like computer science.

4. Scale $\boldsymbol{a_2}$ to reach this line, which gives us the scalar $y = 2$ in $\boldsymbol{b} = x\boldsymbol{a_1} + y\boldsymbol{a_2}$.

Following the steps listed above, we can see that the solution is the scalars needed for $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$, giving us $(x, y) = (3, 2)$. Although we presented our steps above as an "algorithm," we should point out that it is never used as a method for solving the equations. It is an algorithm only for mentally constructing the requisite scalars for the right linear combination, illustrating that it is possible to do so, and that the scalars are unique in this case of linear equations with a unique solution.

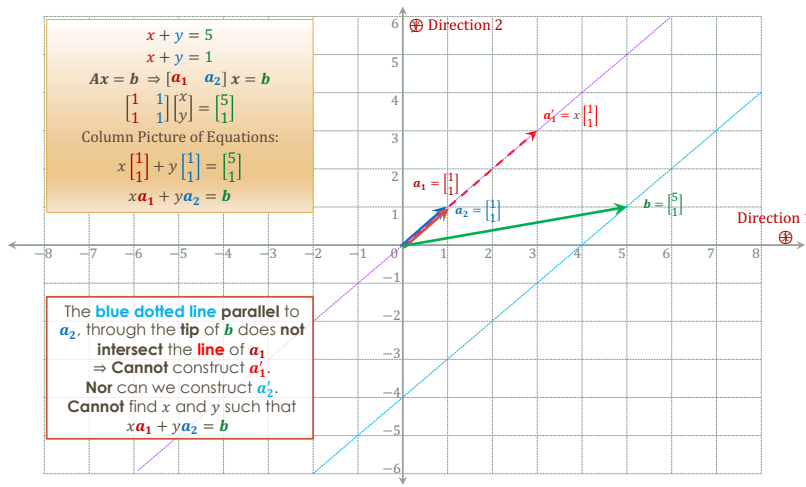


Fig. 6.6 Two inconsistent linear equations on two unknowns with no solution. All linear combination of the column vectors of $\boldsymbol{A}$ fall on the purple line, and the right hand side does not. Therefore, $\boldsymbol{b}$ is not in the span of $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$. Hence no solution. Note that we have drawn the red and blue vectors slightly offset from each other for visibility; they are supposed to be on top of each other.

Let's now move on to the case where we have two linear equations that are not consistent with each other: $x + y = 5$ and $x + y = 1$. In Figure 4.1, we saw that they were parallel lines, which would never meet. What do they look like in our advanced geometric view in the vector space? The column vectors of the coefficient matrix are now identical. As we now know, all possible linear combinations of the two vectors $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$ fall in a subspace that is a line defined by the direction of either of them. Their span, to use the technical term, is only a subspace. The green vector $\boldsymbol{b}$ that we would like to create is

not along this line, which means no matter what scalars we try, we will never be able to get $\boldsymbol{b}$ out of $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$. The system has no solutions.

If we try the steps of our little construction algorithm above, we see that while we can draw a line parallel to $\boldsymbol{a_2}$ going through the tip of $\boldsymbol{b}$, there is scaling factor (other than ∞, to be absolutely rigorous) that will take $\boldsymbol{a_1}$ to this line.

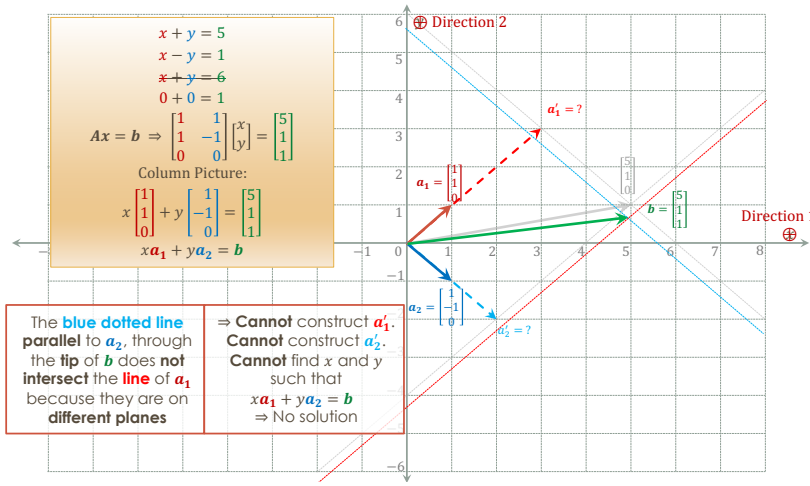


Fig. 6.7 Three linear equations on two unknowns with no solution. Here, $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$ are not collinear, but span the plane of the page. The right hand side, $\boldsymbol{b}$, has a component in the third direction, perpendicular to the page coming toward us, indicated only by the shadow that $\boldsymbol{b}$ casts.

In Figure 6.7, we have three equations, with the third one inconsistent with the first two. The geometric view is in three dimensions, as opposed to the algebraic visualization of the equations, which still stays in two dimensions because of the number of variables. In other words, the geometric view is based on the column vectors of $\boldsymbol{A}$, which have as many elements as equations, or number of rows of $\boldsymbol{A}$. The algebraic view is based on the number of unknowns, which is the same as the number of columns of $\boldsymbol{A}$.

The fact that the vector space now has three directions makes it harder for us to visualize it. We have simplified it: First, we reduced the third equation to a simpler form. Secondly, we indicate the third direction (assumed to be roughly perpendicular to the page, coming

toward us) only by the shadows that the vector and the dashed lines would cast if we were to shine light on them from above the page.

Here are the equations, the columns of $\boldsymbol{A}$ and the RHS:

$$\begin{aligned} &x + y = 5 \\ &x - y = 1 \\ &x + y = 6 \text{ (reduced to 0 = 1)} \end{aligned} \qquad \boldsymbol{a_1} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \boldsymbol{a_2} \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} 5 \\ 1 \\ 1 \end{bmatrix}$$

Running our construction algorithm on this system:

1. Drawing a line parallel to $\boldsymbol{a_2}$ (blue vector), going through the tip of the green $\boldsymbol{b}$, we get the thin dashed line in blue. This line is one unit above the plane of the page (because the third component of $\boldsymbol{b}$ is one).

2. Trying to scale the first red vector ($\boldsymbol{a_1}$) to reach this blue dashed line, we fail because the scaled versions go under the blue line.

3. Similarly, we fail trying to scale the blue $\boldsymbol{a_2}$ as well, for the same reason. We cannot find x and y such that $\boldsymbol{b} = x\boldsymbol{a_1} + y\boldsymbol{a_2}$, because of the pesky, nonzero third component in $\boldsymbol{b}$.

4. However, we can see that the shadows of these red and blue dashed lines (shown in grey) on the plane of the page do meet at the tip of the shadow of $\boldsymbol{b}$. Think of these shadows as projections and we have a teaser for a future topic.

In Figure 6.7, we took a coefficient matrix such that the third components of its column vectors were zero so that we could visualize the system relatively easily: Most of the action was taking place on the xy-plane. In a general case of three equations on two unknowns, even when we have nonzero third components, the two column vectors still make a plane as their span. If the RHS vector is in the span of the two vectors, we get a unique solution. If not, the equations are inconsistent and we get no solution.

The geometric view of Linear Algebra, much like the algebraic view, concerns itself with the solvability conditions, the characteristics of the solutions *etc.*, but from the geometry of the vector spaces associated with the coefficient matrix (as opposed to row-reduction type of operations in the algebraic view). As we saw in this chapter, and as we will appreciate even more in later chapters, the backdrop of the geometric view is the column picture of matrix multiplication.

7

Change of Basis, Orthogonality and Gram-Schmidt

Mathematics takes us still further from what is human, into the region of absolute necessity, to which not only the world, but every possible world, must conform.

—Bertrand Russell

Right from the start of this book, we wrote vectors as column matrices, with numbers arranged in a column. These numbers are the components of the vector. Where do the components come from? How do we get them? They are, in fact, the byproduct of the underlying basis that we did not hitherto talk much about. In this chapter, we will expand on our understanding of bases, learn how the components change when we change bases and explore some of the desirable properties of basis vectors.

7.1 Basis and Components

Basis of a space[1] (such as $\mathbb{R}^n$) is the minimal set of vectors that span it. Minimal implies that the basis vectors are linearly independent. In $\mathbb{R}^n$, we can have a maximum of n linearly independent vectors $\boldsymbol{a_i} \in \mathbb{R}^n$. Any set of such n linearly independent vectors would be a basis for $\mathbb{R}^n$. Once we have the basis, we can express any vector $\boldsymbol{x} \in \mathbb{R}^n$ as a *unique* linear combination of the basis vectors. The components (or the coordinates) of $\boldsymbol{x}$ are the coefficients (or the scaling factors) of this linear combination. Since the linear combination is unique, so are the components.

7.1.1 Components of a Vector

Definition: For any $\boldsymbol{x} \in \mathbb{R}^n$, if the set of n vectors $\boldsymbol{a_i}$ (which are columns of a matrix $\boldsymbol{A}$) form a basis for $\mathbb{R}^n$ and

$$\boldsymbol{x} = \sum_{i=1}^{n} x_{i|\boldsymbol{A}} \boldsymbol{a_i} \tag{7.1}$$

then the n scalars $x_{i|\boldsymbol{A}} \in \mathbb{R}$ are the components (or coordinates) of $\boldsymbol{x}$ in the basis $\boldsymbol{a_i}$, or in the basis matrix $\boldsymbol{A}$.

If we were to place the basis vectors as the columns of a matrix, we would get a square matrix $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a_i} \end{bmatrix} \in \mathbb{R}^{n \times n}$. Now that we know that we can place a set of basis vectors as the columns of a matrix, we will start calling the matrices themselves the basis of spaces and subspaces, which is why we call the components of $\boldsymbol{x}$ in the $\boldsymbol{A}$ basis as $x_{i|\boldsymbol{A}}$ in Eqn (7.1).

7.1.2 Identity Matrix as a Basis

To answer the question that we started this chapter with: We already had components to our vectors. Where do they come from? These components are, in fact, with reference to the identity matrix as the basis . Let's look at an example to illustrate it. Consider a vector

[1] From this chapter onward, when we say space or subspace, we mean a vector space or a vector subspace.

$\boldsymbol{x} \in \mathbb{R}^2$.

$$\boldsymbol{x} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \boldsymbol{I}\boldsymbol{x} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 3 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 2\boldsymbol{q_1} + 3\boldsymbol{q_2}$$

where we called the basis vectors $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$. Some textbooks, especially the ones on physics, may call the *unit* vectors $\hat{i}$ and $\hat{j}$, so that $\boldsymbol{x} = 2\hat{i} + 3\hat{j}$. In either case, the components of the vector are 2 in the first direction and 3 in the second one. Clearly, this example of how a two-dimensional vector gets its components from an $\boldsymbol{I_2}$ extends to higher dimensions as well. For $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{I_n}$ is the basis that gives it the components.

Identity matrices are pretty much the perfect basis we could ask for. First of all, the column vectors of $\boldsymbol{I}$ all have their norm equal to one, which is why we call them *unit* vectors in the previous paragraph. Secondly, each column is orthogonal to the rest. And lastly, the matrix is diagonal, which says that the dimension of the space of which it is a basis is the sum of the diagonal elements. This sum is also called the trace (usually written as Tr($\boldsymbol{A}$) or trace($\boldsymbol{A}$) or Tr. $\boldsymbol{A}$) of the matrix. Since it is an important concept in theoretical Linear Algebra, let's define it:

Trace

Definition: For any $\boldsymbol{A} = \begin{bmatrix} a_{ij} \end{bmatrix} \in \mathbb{R}^{n \times n}$, its trace is defined as

$$\text{trace}(\boldsymbol{A}) = \sum_{i=1}^{n} a_{ii}$$

When we use the identity matrix as the basis (as we almost always do), what we get as the components of vectors are, in fact, the coordinates of the points where the tips of the vectors lie. For this reason, the identity basis may also be referred to as the *coordinate basis*. The components of a vector may be called *coordinates*. And a vector (as we define and use it in Linear Algebra, as starting from the origin) may be called a *position vector* to distinguish it from other vectors (such as the electric or magnetic field strength, which may have a specific value and direction at any point in the coordinate space).

7.1.3 Orthogonal Basis

We sneaked in orthogonality earlier in the previous chapter, although we were planning a fuller treatment of the topic here in this chapter. Orthogonal vectors are the ones whose dot product is zero: $\boldsymbol{x}$ and $\boldsymbol{y}$ are orthogonal if $\boldsymbol{x}^\mathsf{T}\boldsymbol{y} = 0$. If our basis vectors are orthogonal to each other, then we have an orthogonal basis.

7.1.4 Orthonormal Basis

In addition to being orthogonal to each other, if our basis vectors are all of unit length, then we have an orthonormal basis. The identity matrix as a basis is a nice orthonormal basis, but it is not the only one. Before looking at some examples, let's study orthonormal bases in their generality.

Let's call our orthonormal basis matrix $\boldsymbol{Q} = [\boldsymbol{q_i}] \in \mathbb{R}^{n\times n}$, with column vectors $\boldsymbol{q_i} \in \mathbb{R}^n$. The fact that the columns $\boldsymbol{q_i}$ form an orthonormal basis for means two things: The dot product of any two distinct columns is zero, and the norm of each column is one. Both these characteristics can be stated compactly as:

$$\boldsymbol{q}_i^\mathsf{T}\boldsymbol{q}_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{7.2}$$

As a consequence, we get some interesting properties for the matrix $\boldsymbol{Q}$.

1. The inverse of an orthonormal matrix is its transpose:

$$\boldsymbol{Q}^{-1} = \boldsymbol{Q}^\mathsf{T} \implies \boldsymbol{Q}^\mathsf{T}\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^\mathsf{T} = \boldsymbol{I} \tag{7.3}$$

   We can easily prove this by looking at the product $\boldsymbol{Q}\boldsymbol{Q}^\mathsf{T}$ as the dot products of the rows of $\boldsymbol{Q}$ with the columns of $\boldsymbol{Q}^\mathsf{T}$. Because of Eqn (7.2) above, only the elements where the row number is the same as the column number survive, giving us the identity matrix.

2. An orthonormal matrix does not change the norm of a vector:

$$\|\boldsymbol{Q}\boldsymbol{x}\| = \|\boldsymbol{x}\| \tag{7.4}$$

Proof: Consider $\|\boldsymbol{Qx}\|^2$:

$$\begin{aligned}\|\boldsymbol{Qx}\|^2 &= (\boldsymbol{Qx})^{\mathsf{T}}\boldsymbol{Qx} = \boldsymbol{x}^{\mathsf{T}}\boldsymbol{Q}^{\mathsf{T}}\boldsymbol{Qx} \\ &= \boldsymbol{x}^{\mathsf{T}}\boldsymbol{Q}^{-1}\boldsymbol{Qx} = \boldsymbol{x}^{\mathsf{T}}\boldsymbol{Ix} \\ &= \boldsymbol{x}^{\mathsf{T}}\boldsymbol{x} = \|\boldsymbol{x}\|^2 \\ \implies \|\boldsymbol{Qx}\| &= \|\boldsymbol{x}\|\end{aligned}$$

3. Orthonormal matrices have unit determinants: $|\boldsymbol{Q}| = \pm 1$, which follows conceptually from the fact that they do not change the size of the vector it is multiplying with. But it can be easily proven by the properties of determinants:

$$\begin{aligned}\boldsymbol{Q}^{-1} = \boldsymbol{Q}^{\mathsf{T}} \implies \left|\boldsymbol{Q}^{-1}\right| = \left|\boldsymbol{Q}^{\mathsf{T}}\right| = |\boldsymbol{Q}| \implies \frac{1}{|\boldsymbol{Q}|} = |\boldsymbol{Q}| \\ \implies |\boldsymbol{Q}|^2 = 1 \implies |\boldsymbol{Q}| = \pm 1\end{aligned}$$

Keep in mind that although we made a distinction between orthogonal and orthonormal matrices, most authors write the former to mean the latter. We also will adapt this sloppy practice soon.

7.2 Change of Basis

One of the important applications of Linear Algebra uses the change of basis: Going from one basis to another. We see this application in multi-player video games where the underlying world is rendered from the perspective each player. We also see it in the perspective correction apps in our smart phones.

Let's start with some examples as shown in Table 7.1 and drawn in Figure 7.1. In the first row of the vector $\boldsymbol{x}$ has components 7 and 5 in the usual basis we are used to, which may be called identity or coordinate basis. In the table, it is written as $[\boldsymbol{x}]_{\boldsymbol{I}}$.

In the second row of Table 7.1, we are using the matrix $\boldsymbol{A}'$ as the basis. Notice that the basis vectors $\boldsymbol{a}'_1$ and $\boldsymbol{a}'_2$ are still in the same direction as the corresponding unit vectors from the identity basis $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$. But they have been scaled up. Looking at the components of the same vector in this basis $\boldsymbol{A}'$, as we see in $[\boldsymbol{x}]_{\boldsymbol{A}'}$, we see that they are smaller than the coordinates: $7 \rightarrow 2$ and $5 \rightarrow 2$. What it is saying is that since the basis vectors are bigger, we need to take

Table 7.1 Examples of change of basis

Basis Matrix	Basis Vectors	Linear Combination	Vector Components
$\boldsymbol{A} = \boldsymbol{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\boldsymbol{q_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\ \boldsymbol{q_2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\boldsymbol{x} = 7\boldsymbol{q_1} + 5\boldsymbol{q_2}$	$[\boldsymbol{x}]_{\boldsymbol{I}} = \begin{bmatrix} 7 \\ 5 \end{bmatrix}$
$\boldsymbol{A'} = \begin{bmatrix} 3.5 & 0 \\ 0 & 2.5 \end{bmatrix}$	$\boldsymbol{a'_1} = \begin{bmatrix} 3.5 \\ 0 \end{bmatrix}\ \boldsymbol{a'_2} = \begin{bmatrix} 0 \\ 2.5 \end{bmatrix}$	$\boldsymbol{x} = 2\boldsymbol{a'_1} + 2\boldsymbol{a'_2}$	$[\boldsymbol{x}]_{\boldsymbol{A'}} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$
$\boldsymbol{A''} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$	$\boldsymbol{a''_1} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}\ \boldsymbol{a''_2} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\boldsymbol{x} = 2\boldsymbol{a''_1} + 3\boldsymbol{a''_2}$	$[\boldsymbol{x}]_{\boldsymbol{A''}} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

How the same vector $\boldsymbol{x}$ is represented in various bases. The notation of square brackets around a vector with a matrix as subscript stands for the vector represented in the basis of the matrix: $[\boldsymbol{x}]_{\boldsymbol{A}}$ is the representation of $\boldsymbol{x}$ using the columns of $\boldsymbol{A}$ as the basis vectors.

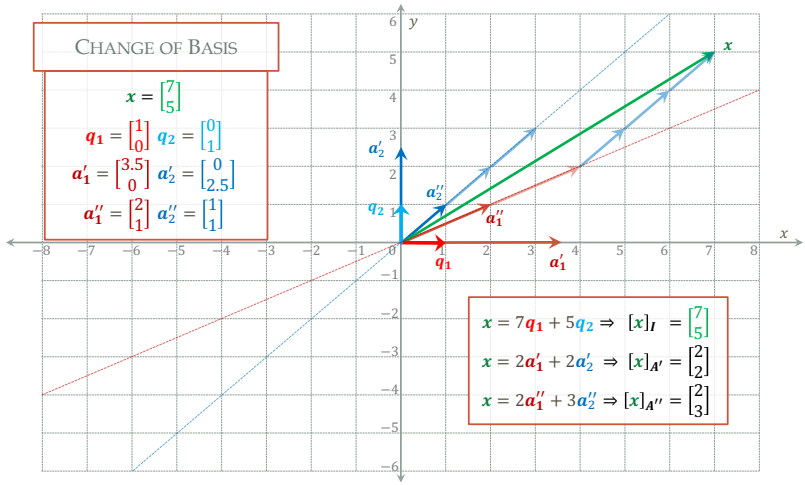


Fig. 7.1 Visualizing the change-of-basis examples listed in Table 7.1. The green vector is represented in three different bases, giving vastly different values for its components.

only two in each direction to get to the tip of our vector $\boldsymbol{x}$. When the basis vectors become bigger, the components get smaller, which should indicate to us that the change of basis probably has the inverse of the basis matrix involved in some fashion.

In the third row of Table 7.1, things get really complicated. Now we have the basis vectors in some random direction (not orthogonal to each other) with some random size (pretty far from unity). The first component now is 2 and the second 3, as described in Table 7.1 and illustrated in Figure 7.1.

How do we compute the new components given the original vector and the new basis vectors? In other words, how do we perform the change of basis in a general way? Let's call our original basis is the coordinate basis $\boldsymbol{I}_n \in \mathbb{R}^{n\times n}$ and the new basis $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. The column vectors of $\boldsymbol{A} = \boldsymbol{a_i} \in \mathbb{R}^n$ are also specified in the identity basis. Eqn (7.1) tells us how to arrive the components of $\boldsymbol{x}$ (which are x_i) in this basis. Remember, we denoted the representation of $\boldsymbol{x}$ in $\boldsymbol{A}$ as $[\boldsymbol{x}]_{\boldsymbol{A}}$. In order to find the components of $\boldsymbol{x}$ in the $\boldsymbol{A}$ basis, we have to find the vector $[\boldsymbol{x}]_{\boldsymbol{A}}$ such that $\boldsymbol{x} = \boldsymbol{A}[\boldsymbol{x}]_{\boldsymbol{A}}$, which is the same as solving the system of linear equations $\boldsymbol{x} = \boldsymbol{A}[\boldsymbol{x}]_{\boldsymbol{A}}$ as shown in Eqn (7.5) below. Since we know that $\boldsymbol{A}$ is a basis matrix, it is square, full rank and is therefore invertible. We can, therefore, get the components in the new basis $\boldsymbol{A}$ through matrix inversion:

$$\boldsymbol{x} = \sum_{i=1}^{n} x_{i|A}\boldsymbol{a_i} \implies \boldsymbol{x} = \boldsymbol{A}[\boldsymbol{x}]_{\boldsymbol{A}} \implies [\boldsymbol{x}]_{\boldsymbol{A}} = \boldsymbol{A^{-1}x} \tag{7.5}$$

Let's now verify Eqn (7.5) using the most complicated example we did, namely the third row of Table 7.1. Remembering the prescription for the inverse of a 2×2 matrix (swap the diagonal elements, switch the sign of the off-diagonal elements and divide by the determinant), we have:

The original vector in the coordinate basis: $\boldsymbol{x} = \begin{bmatrix} 7 \\ 5 \end{bmatrix}$

The new basis: $\boldsymbol{A} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \implies |\boldsymbol{A}| = 1$ and $\boldsymbol{A^{-1}} = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$

The vector in the new basis: $[\boldsymbol{x}]_{\boldsymbol{A}} = \boldsymbol{A^{-1}x} = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 7 \\ 5 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

Note that we have used the symbol $\boldsymbol{A}$ for the new basis rather than $\boldsymbol{A}''$ as in Eqn (7.5). Comparing our $[\boldsymbol{x}]_{\boldsymbol{A}}$ with $[\boldsymbol{x}]_{\boldsymbol{A}''}$ in the Table 7.1, we can satisfy ourselves that the matrix equation in Eqn (7.5) does work.

7.2.1 Consequences of Basis Change

When we change the basis to $\boldsymbol{A}$, what happens to vector dot products, and therefore, to the norm of vector? Using the definition of dot products way back in Eqn (2.11) and the change of basis in Eqn (7.5),

we can write:

$$\boldsymbol{x} \cdot \boldsymbol{y} = \boldsymbol{x}^{\mathsf{T}}\boldsymbol{y} = \left(\boldsymbol{A}[\boldsymbol{x}]_{\boldsymbol{A}}\right)^{\mathsf{T}} \boldsymbol{A}[\boldsymbol{y}]_{\boldsymbol{A}} = [\boldsymbol{x}]_{\boldsymbol{A}}^{\mathsf{T}} \left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right) [\boldsymbol{y}]_{\boldsymbol{A}}$$

If the dot product is to remain unchanged, we need $\boldsymbol{x} \cdot \boldsymbol{y} = \boldsymbol{x}^{\mathsf{T}}\boldsymbol{y} = [\boldsymbol{x}]_{\boldsymbol{A}}^{\mathsf{T}}[\boldsymbol{y}]_{\boldsymbol{A}}$, or $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A} = \boldsymbol{I}$ for the new basis $\boldsymbol{A}$. What type of matrices satisfy this condition? As we see in Eqn (7.3), orthogonal matrices do. Therefore, preservation of the norm of the vector under basis change also requires the basis matrix to be orthogonal. (Orthonormal, in fact, but as promised, we will let it slide from now on.)

7.3 Basis of Subspaces

Whatever we said about bases and components for spaces also applies to subspaces, but with one important and interesting difference. As we know, subspaces live inside a bigger space. For example, we can have a subspace of dimension r inside $\mathbb{R}^n$ with $r < n$. For this subspace, we will need r basis vectors, each of which is a n-dimensional vector: $\boldsymbol{a_i} \in \mathbb{R}^n$. If we were to place these r vectors in a matrix, we would get $\boldsymbol{A} \in \mathbb{R}^{n \times r}$, not a square matrix, but a "tall" one.

Remember, this subspace of dimension r is *not* $\mathbb{R}^r$. In particular, a two-dimensional subspace (a plane going through the origin) inside $\mathbb{R}^3$ is *not* the same as $\mathbb{R}^2$. Let's take an example, built on the third row of Table 7.1 again, to illustrate it. Let's take the two vectors in the example, make them three-dimensional by adding a third component. The subspace we are considering is the span of these two vectors, which is a plane in the coordinate space $\mathbb{R}^3$: All linear combinations of the two vectors lie on this plane. We will use the same two vectors as the basis $\boldsymbol{A}$ and write our vector $\boldsymbol{x}$ (old, coordinate basis) as $[\boldsymbol{x}]_{\boldsymbol{A}}$ (new basis for the subspace). We have:

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \quad \boldsymbol{x} = 2\boldsymbol{a_1} + 3\boldsymbol{a_2} = \begin{bmatrix} 7 \\ 5 \\ 2 \end{bmatrix} \implies [\boldsymbol{x}]_{\boldsymbol{A}} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

Note that our $[\boldsymbol{x}]_{\boldsymbol{A}}$ has only two components because the subspace has a dimension of two. Why is that? Although all the vectors in the subspace are in $\mathbb{R}^3$, they are all linear combinations of the two column vectors of $\boldsymbol{A}$. The two scaling factors required in taking the

linear combination of the basis vectors are the two components of the vectors in this basis.

In the case of full spaces $\mathbb{R}^n$, we had a formula in Eqn (7.5) to compute $[\boldsymbol{x}]_{\boldsymbol{A}}$, which had $\boldsymbol{A}^{-1}$ in it. For a subspace, however, what we have is a "tall" matrix with more rows (m) than columns ($r < m$). What is the equivalent of Eqn (7.5) in this case? Here's where we will use the left inverse as defined in Eqn (5.2). Note that $\boldsymbol{A}$ is a tall matrix with full column rank because its r columns (being basis vectors for a subspace) are linearly independent. $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}$ is a full-rank, square matrix of size $r \times r$, whose inverse figures in the left inverse. As a reminder, here is how we defined it:

$$\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right) = \boldsymbol{I} \implies \boldsymbol{A}_{\text{left}}^{-1} = \left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}}$$

Coming back to the change-of-basis problem, we get the components in the new basis $\boldsymbol{A}$ following the same arguments used earlier in deriving Eqn (7.5):

$$\boldsymbol{A}[\boldsymbol{x}]_{\boldsymbol{A}} = \boldsymbol{x} \implies [\boldsymbol{x}]_{\boldsymbol{A}} = \boldsymbol{A}_{\text{Left}}^{-1}\,\boldsymbol{x} = \left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{x} \tag{7.6}$$

Once again, let's verify the veracity of this prescription using our example.

$$\boldsymbol{x} = \begin{bmatrix}7\\5\\2\end{bmatrix} \quad \boldsymbol{A} = \begin{bmatrix}2 & 1\\1 & 1\\1 & 0\end{bmatrix} \implies \boldsymbol{A}^{\mathsf{T}} = \begin{bmatrix}2 & 1 & 1\\1 & 1 & 0\end{bmatrix} \text{ and } \boldsymbol{A}^{\mathsf{T}}\boldsymbol{A} = \begin{bmatrix}6 & 3\\3 & 2\end{bmatrix}$$

$$\left|\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right| = 3 \text{ and } \left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1} = \frac{1}{3}\begin{bmatrix}2 & -3\\-3 & 6\end{bmatrix} = \begin{bmatrix}\frac{2}{3} & -1\\-1 & 2\end{bmatrix}$$

$$\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{A}_{\text{Left}}^{-1} = \begin{bmatrix}\frac{2}{3} & -1\\-1 & 2\end{bmatrix}\begin{bmatrix}2 & 1 & 1\\1 & 1 & 0\end{bmatrix} = \begin{bmatrix}\frac{1}{3} & -\frac{1}{3} & \frac{2}{3}\\0 & 1 & -1\end{bmatrix}$$

$$[\boldsymbol{x}]_{\boldsymbol{A}} = \boldsymbol{A}_{\text{Left}}^{-1}\,\boldsymbol{x} = \begin{bmatrix}\frac{1}{3} & -\frac{1}{3} & \frac{2}{3}\\0 & 1 & -1\end{bmatrix}\begin{bmatrix}7\\5\\2\end{bmatrix} = \begin{bmatrix}2\\3\end{bmatrix}$$

It is a tedious calculation, but it works out to be exactly the scalars in the linear combination we started from: $\boldsymbol{x} = 2\boldsymbol{a_1} + 3\boldsymbol{a_2}$. Happily, we will never have to do such calculations by hand; we have **SageMath** to do it for us. For a deeper understanding of this left inverse, we will have to wait for the projection operation coming up two chapters down the line, but based on what we will discuss next.

7.4 Orthogonality

7.4.1 Orthogonal Vectors

We came across orthogonal vectors in a couple of places earlier in this book. If we imagine vectors as line segments in *coordinate* spaces with arrows at their tips, we can say that orthogonal vectors are simply perpendicular vectors. We are clearly more sophisticated than that by now, and our *vector* spaces do not have lines and arrows. Orthogonal vectors are, therefore, those vectors that have their inner (or dot) product equal to zero.

Orthogonal Vectors

Definition: Two vectors, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ are orthogonal to each other if and only if

$$\boldsymbol{x}^\mathsf{T}\boldsymbol{y} = \boldsymbol{y}^\mathsf{T}\boldsymbol{x} = 0$$

Note that the vectors in the inner product can commute. Note also that the less sophisticated definition of the inner product (namely $\boldsymbol{x} \cdot \boldsymbol{y} = \|\boldsymbol{x}\|\|\boldsymbol{y}\| \cos\theta$) shows that the inner (or dot) product is zero when $\boldsymbol{x}$ and $\boldsymbol{y}$ are perpendicular to each other because the angle θ between them is $\frac{\pi}{2}$ and $\cos\theta = 0$.

We still stay away from the definition of the inner product using the angle because, by now, we know that the machinery of Linear Allegra may be applied to vector-like objects where we may not be able to talk about directions and angles. For example, in Fourier transforms or the wave functions in quantum mechanics, vectors are functions with the inner product defined with no reference any kind of angles. We can still have *orthogonal* "vectors" in such vector spaces when the inner product is zero. Clearly, we cannot have *perpendicular* vectors without abusing the notion and notation a bit too much for our (or at least, the author's) liking.

Earlier, in Eqn (2.5), we defined the Euclidean norm of a vector $\|\boldsymbol{a}\|$:

$$\|\boldsymbol{a}\|^2 = \boldsymbol{a}^\mathsf{T}\boldsymbol{a}$$

Using this definition, we can prove that the inner product of orthogonal vectors has to be zero, albeit not completely devoid of the lack of sophistication associated with perpendicularity.

If we have $\boldsymbol{a} \perp \boldsymbol{b}$, then we know that $\|\boldsymbol{a}\|$and $\|\boldsymbol{b}\|$ make the sides of a right-angled triangle (which is where the pesky perpendicularity

rears its ugly head again) whose hypotenuse is of length $\|\boldsymbol{a}+\boldsymbol{b}\|$. Then, by the Pythagorean theorem, we can write:

$$\begin{aligned}
\|\boldsymbol{a}\|^2 + \|\boldsymbol{b}\|^2 &= \|\boldsymbol{a}+\boldsymbol{b}\|^2 \\
\boldsymbol{a}^\mathsf{T}\boldsymbol{a} + \boldsymbol{b}^\mathsf{T}\boldsymbol{b} &= (\boldsymbol{a}+\boldsymbol{b})^\mathsf{T}(\boldsymbol{a}+\boldsymbol{b}) \\
&= \left(\boldsymbol{a}^\mathsf{T}+\boldsymbol{b}^\mathsf{T}\right)(\boldsymbol{a}+\boldsymbol{b}) \\
&= \boldsymbol{a}^\mathsf{T}\boldsymbol{a} + \boldsymbol{b}^\mathsf{T}\boldsymbol{b} + \boldsymbol{b}^\mathsf{T}\boldsymbol{a} + \boldsymbol{a}^\mathsf{T}\boldsymbol{b} \\
\implies 0 &= \boldsymbol{b}^\mathsf{T}\boldsymbol{a} + \boldsymbol{a}^\mathsf{T}\boldsymbol{b} \\
\implies \boldsymbol{b}^\mathsf{T}\boldsymbol{a} = \boldsymbol{a}^\mathsf{T}\boldsymbol{b} &= 0 \text{ since } \boldsymbol{b}^\mathsf{T}\boldsymbol{a} = \boldsymbol{a}^\mathsf{T}\boldsymbol{b}
\end{aligned}$$

7.4.2 Orthogonalization

Given that the orthonormal (we may as well call it orthogonal because everybody does it) basis is the best possible basis we can ever hope to have, we may want to have an algorithm to make any matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ orthogonal. An orthogonal matrix is the one in which the columns are normalized and orthogonal to one another. In other words, it is matrix that could be a basis matrix as described in §7.1.4 with the associated properties. And orthogonalization is the process or algorithm that can make a square matrix orthogonal.

The first question we may want to ask ourselves is why we would want to do this; why orthogonalize? We know that a perfect basis for $\mathbb{R}^{n\times n}$ is $\boldsymbol{I}_n$, the identity matrix. Why not just use it? We have two reasons for doing it. The first one is pedagogical: We get to see how projection works in a general way, which we will use later. The second reason is a practical one from a computer science perspective: Certain numerical algorithms use the decomposition that results from the orthogonalization process. Another reason, again from our neck of the woods, is that when we know that a matrix $\boldsymbol{Q}$ is orthonormal, we know that the transformation it performs on a vector is numerically stable: $\boldsymbol{Q}^n\boldsymbol{x}$ does not suffer from overflow or underflow errors because the norm of $\boldsymbol{x}$ is not modified by the multiplication with $\boldsymbol{Q}$.

7.4.3 Projection

Earlier, in Chapter 2, we looked at vector dot product as projection using the cosine of the angle between them, in Figure 2.5. To remind

ourselves, if we have two vectors $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$ with an angle θ between them, the projection of $\boldsymbol{a_2}$ onto $\boldsymbol{a_1}$ has the length $\|\boldsymbol{a_2}\| \cos\theta$. From the definition of the dot product using the angle, we can write:

$$\|\boldsymbol{a_2}\| \cos\theta = \frac{\boldsymbol{a_1} \cdot \boldsymbol{a_2}}{\|\boldsymbol{a_1}\|} = \frac{\boldsymbol{a_1}}{\|\boldsymbol{a_1}\|} \cdot \boldsymbol{a_2}$$

To move to a totally matrix notation, we can use the matrix definition of dot products, $\boldsymbol{a_1} \cdot \boldsymbol{a_2} = \boldsymbol{a_1}^\mathsf{T} \boldsymbol{a_2} = \boldsymbol{a_2}^\mathsf{T} \boldsymbol{a_1}$, which changes the formula above to:

$$\|\boldsymbol{a_2}\| \cos\theta = \frac{\boldsymbol{a_1}^\mathsf{T}}{\|\boldsymbol{a_1}\|} \boldsymbol{a_2} = \boldsymbol{a_2}^\mathsf{T} \frac{\boldsymbol{a_1}}{\|\boldsymbol{a_1}\|}$$

This formula gives us the *length* of the projection, which we call x in Figure 7.2. For our purposes, we want a *vector* as the projection, which would be a vector in the direction of $\boldsymbol{a_1}$, with its norm equal to the length of the projection x. Let's call this projection vector $\boldsymbol{a_{2\|}}$. The direction of $\boldsymbol{a_1}$ is specified by the unit vector $\boldsymbol{q_1}$:

$$\boldsymbol{q_1} = \frac{\boldsymbol{a_1}}{\|\boldsymbol{a_1}\|}$$

Putting it all together, we write:

$$\boldsymbol{a_{2\|}} = \left(\boldsymbol{a_2}^\mathsf{T} \boldsymbol{q_1}\right) \boldsymbol{q_1} \tag{7.7}$$

This form of the projection is what we will use in the Gram-Schmidt process coming up in the next section.

However, we want to go a bit further and with this Eqn (7.7) and develop what we will call a projection matrix. Knowing the definition of the norm of a vector:

$$\|\boldsymbol{a_1}\|^2 = \boldsymbol{a_1}^\mathsf{T} \boldsymbol{a_1}$$

we can write the projection in a form that we will use later, when we start projecting vectors onto subspaces rather than other vectors. The derivation of this form of projection is shown in Figure 7.2, which we will repeat as a recasting of Eqn (7.7) merely to get used to the vector/matrix manipulations, if nothing else.

Rewriting Eqn (7.7) using definitions of the norm ($\|\boldsymbol{a_1}\|$) and the direction ($\boldsymbol{q_1}$):

$$\begin{aligned} \boldsymbol{a_{2\|}} &= \boldsymbol{q_1} \left(\boldsymbol{q_1}^\mathsf{T} \boldsymbol{a_2}\right) = \frac{\boldsymbol{a_1}}{\|\boldsymbol{a_1}\|} \left(\frac{\boldsymbol{a_1}^\mathsf{T}}{\|\boldsymbol{a_1}\|} \boldsymbol{a_2}\right) \\ &= \boldsymbol{a_1} \left(\frac{\boldsymbol{a_1}^\mathsf{T} \boldsymbol{a_2}}{\|\boldsymbol{a_1}\|^2}\right) = \left(\frac{\boldsymbol{a_1} \boldsymbol{a_1}^\mathsf{T}}{\boldsymbol{a_1}^\mathsf{T} \boldsymbol{a_1}}\right) \boldsymbol{a_2} \end{aligned} \tag{7.8}$$

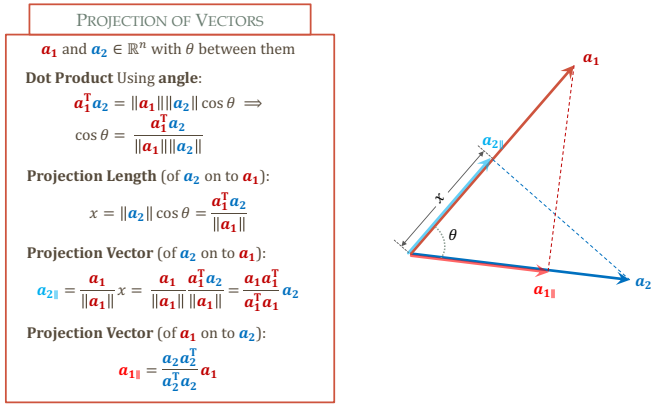


Fig. 7.2 Dot product between two vectors $\boldsymbol{a_1}, \boldsymbol{a_2} \in \mathbb{R}^n$, shown on the plane defined by the two vectors.

In the relatively complicated manipulations in Eqn (7.8), we have a couple of observations to write down:

- The entities in the parentheses in all but the last RHS are scalars. If we were to call it, say s, it commutes with matrix (and vector) multiplication:

$$s\boldsymbol{a_1}\boldsymbol{a_2} = \boldsymbol{a_1} s \boldsymbol{a_2} = \boldsymbol{a_1}\boldsymbol{a_2} s$$

- In the first RHS, we have the projection of $\boldsymbol{a_2}$ equal to a scalar times the direction of $\boldsymbol{a_1}$, which makes sense.
- This pattern repeats itself in all the RHS until the very last one: The projection vector is $\boldsymbol{a_1}$ scaled.
- In the last and final RHS (where we used the associativity of matrix multiplication to put the parenthesis where we wanted), the entity multiplying $\boldsymbol{a_2}$ is a matrix: It is $\boldsymbol{a_1}\boldsymbol{a_1^T} \in \mathbb{R}^{n\times n}$ multiplied by a scalar $\left(\boldsymbol{a_1^T}\boldsymbol{a_1}\right)^{-1}$.
- We can think of this scaled matrix as an operator, which, when applied to any vector $\boldsymbol{a_2}$, gives its projection to $\boldsymbol{a_1}$

We will call the scaled matrix $\boldsymbol{P}$, the projection matrix. Let's write it down once more as the definition of the projection matrix, which we

will revisit in a couple of chapters.

$$\boldsymbol{P} = \boldsymbol{a_1}\left(\boldsymbol{a_1^\intercal a_1}\right)^{-1}\boldsymbol{a_1^\intercal} \tag{7.9}$$

7.5 Gram-Schmidt Process

The Gram-Schmidt process is an algorithm to turn a square, invertible matrix into an orthonormal one through a series of steps. These steps are, in fact, elementary column operations, akin to the row operations we performed in Gaussian and Gauss-Jordan eliminations.

At the end of the Gram-Schmidt process, starting from $\boldsymbol{A}$, we will get a matrix $\boldsymbol{Q}$ that is a rotated version of the identity matrix $\boldsymbol{I}$ with some of the rows permuted. As we will see, the algorithm keeps the direction of the first column vector and the order of the rest fixed. In other words, if the second column vector of $\boldsymbol{A}$ is to the "right" of the first one, we will have the second column of $\boldsymbol{Q}$ also to the right of the first one. (To be more precise, by "right," we mean that we have to turn clockwise to go from the first column to the second.)

7.5.1 The Algorithm

Since our objective is to get an orthonormal basis matrix, we will normalize each column vector and ensure that it is orthogonal to the rest through this iterative process, starting from the first column. In order to describe the algorithm (which is what a step-by-step process is), let's set the stage by specifying our symbols. We will start from a square matrix $\boldsymbol{A}$ and end up with another square matrix $\boldsymbol{Q}$.

$$\boldsymbol{A} = \begin{bmatrix}\boldsymbol{a_i}\end{bmatrix}, \boldsymbol{Q} = \begin{bmatrix}\boldsymbol{q_i}\end{bmatrix} \in \mathbb{R}^{n\times n}, \boldsymbol{A} \xrightarrow{\text{G-S}} \boldsymbol{Q} \implies \boldsymbol{q_i^\intercal q_j} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

Here are the steps in the Gram-Schmidt process, as illustrated in Figure 7.3:

1. Take the first column of $\boldsymbol{A}$ and normalize it to get the first column of $\boldsymbol{Q}$.
$$\boldsymbol{q_1} = \frac{\boldsymbol{a_1}}{\|\boldsymbol{a_1}\|}$$
2. Use $\boldsymbol{q_1}$ to get $\boldsymbol{q_2}$ (or $\boldsymbol{q_j}$ with $j = 2$) using the steps below:

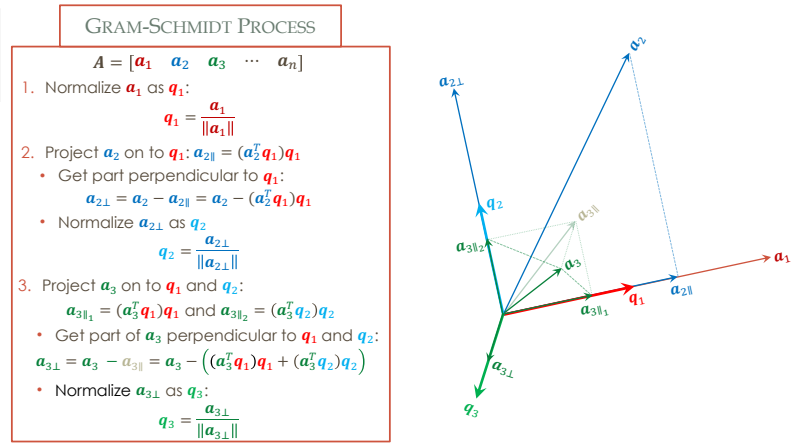


Fig. 7.3 Illustration of the Gram-Schmidt process running on a matrix $\boldsymbol{A}$. The first column (red) is normalized to get the unit vector $\boldsymbol{q_1}$, which is then used to create $\boldsymbol{q_2}$ from the second (blue) column. Both $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ are used in projecting the third (green) column and computing $\boldsymbol{q_3}$.

(a) Take the second column of $\boldsymbol{A}$ and project it onto $\boldsymbol{q_1}$ to get the part of $\boldsymbol{a_2}$ parallel to it, using Eqn (7.7).

$$\boldsymbol{a_{2\|}} = \left(\boldsymbol{a_2^{\mathsf{T}} q_1}\right) \boldsymbol{q_1}$$

(b) Subtract $\boldsymbol{a_{2\|}}$ from $\boldsymbol{a_2}$ to get the perpendicular part.

$$\boldsymbol{a_{2\perp}} = \boldsymbol{a_2} - \boldsymbol{a_{2\|}} = \boldsymbol{a_2} - \left(\boldsymbol{a_2^{\mathsf{T}} q_1}\right) \boldsymbol{q_1}$$

(c) Normalize $\boldsymbol{a_{2\perp}}$ to get the second column of $\boldsymbol{Q}$,

$$\boldsymbol{q_2} = \frac{\boldsymbol{a_{2\perp}}}{\|\boldsymbol{a_{2\perp}}\|}$$

3. Now get $\boldsymbol{q_j}$ using the $j-1$ vector's normalized so far, $\boldsymbol{q_i}$, $1 \leq i < j$, following the steps below:

(a) Take $\boldsymbol{a_j}$, the next column of $\boldsymbol{A}$, project it successively to $\boldsymbol{q_i}$ to get the part of $\boldsymbol{a_j}$ parallel to it, using Eqn (7.7). $\boldsymbol{a_{j\|i}} = \left(\boldsymbol{a_j^{\mathsf{T}} q_i}\right) \boldsymbol{q_i}$

(b) Subtract all parallel parts $\boldsymbol{a_{j\|i}}$ from $\boldsymbol{a_j}$ to get the part perpendicular to all $\boldsymbol{q_i}$.

$$\boldsymbol{a_{j\perp}} = \boldsymbol{a_j} - \sum_{i=1}^{j-1} \boldsymbol{a_{j\|i}} = \boldsymbol{a_j} - \sum_{i=1}^{j-1} \left(\boldsymbol{a_j^{\mathsf{T}} q_i}\right) \boldsymbol{q_i}$$

(c) Normalize $\boldsymbol{a_{j\perp}}$ to get the next column of $\boldsymbol{Q}$, $\boldsymbol{q_j}$.

$$\boldsymbol{q_j} = \frac{\boldsymbol{a_{j\perp}}}{\|\boldsymbol{a_{j\perp}}\|}$$

4. Iterate until we run out of columns, which is when $j = n$.

Note that step 2 in the algorithm above is actually the same as step 3, with $j = 2$, but we chose to spell it out in order to make the process clear. In fact, even the first step can be thought of as the same as the third step, making it easy to list the steps as an algorithm, and indeed to write code based on them.

7.5.2 Numerical Considerations

When we look at the steps of the Gram-Schmidt process, we see that we are normalizing a vector, using it to compute the next one, normalizing it and working our way through the whole matrix $\boldsymbol{A}$. As we know, floating point operations on a computer have an inherent precision, and the first step necessarily incurs a certain error. Since it is being used in the next step, the errors accumulate. For this reason, the process may be numerically unstable.

The modified version of the algorithm mitigates this problem by breaking down the summation in the step 3(b) above iteratively as:

$$\begin{aligned}
\boldsymbol{a}_{\boldsymbol{j}\perp}^{(1)} &= \boldsymbol{a_j} - \left(\boldsymbol{a}_{\boldsymbol{j}}^{\mathsf{T}}\boldsymbol{q_1}\right)\boldsymbol{q_1} \\
\boldsymbol{a}_{\boldsymbol{j}\perp}^{(i+1)} &= \boldsymbol{a}_{\boldsymbol{j}\perp}^{(i)} - \left(\boldsymbol{a}_{\boldsymbol{j}\perp}^{(i)\mathsf{T}}\boldsymbol{q_i}\right)\boldsymbol{q_i} \quad 1 < i < j-1 \\
\boldsymbol{a_{j\perp}} &= \boldsymbol{a}_{\boldsymbol{j}\perp}^{(j+1)}
\end{aligned} \tag{7.10}$$

In other words, instead of projecting $\boldsymbol{a_j}$ onto all the currently available orthonormal vectors $\boldsymbol{q_i}$, $1 \leq i < j$ and subtracting the sum in step 3(b), we project it onto $\boldsymbol{q_1}$ first, subtract it from $\boldsymbol{a_j}$ to get the perpendicular part $\boldsymbol{a}_{\boldsymbol{j}\perp}^{(1)}$. Then instead of projecting $\boldsymbol{a_j}$ onto $\boldsymbol{q_1}$ again, we project $\boldsymbol{a}_{\boldsymbol{j}\perp}^{(1)}$ instead, thereby disrupting the accumulation of rounding errors.

7.5.3 QR Decomposition

Since the Gram-Schmidt algorithm is about taking linear combinations of the columns of the matrix, it should be possible to write it as

a matrix multiplication on the right.

$$\boldsymbol{A}, \boldsymbol{Q} \in \mathbb{R}^{n \times n}, \boldsymbol{A} \xrightarrow{\text{G-S}} \boldsymbol{Q} \implies \boldsymbol{AX} = \boldsymbol{Q}, \boldsymbol{X} \in \mathbb{R}^{n \times n}$$

Furthermore, since $\boldsymbol{q_1}$ is a scaled version of $\boldsymbol{a_1}$, the first column of $\boldsymbol{X}$ has only one nonzero element, x_{11}. In general, $\boldsymbol{q_i}$ is a linear combination of all $\boldsymbol{a_j}$, $1 \leq j \leq i$. Therefore, $\boldsymbol{X}$ is an upper triangular matrix. We are more interested in the inverse of this matrix, which we will call $\boldsymbol{R}$. It is also an upper triangular matrix: a fact that can be appreciated by noticing that $\boldsymbol{a_i}$ is a linear combination of all $\boldsymbol{q_j}$, $1 \leq j \leq i$. It is also a fact that can be proven in general, and is left as an exercise.

With these notations, we can write:

$$\boldsymbol{AX} = \boldsymbol{Q} \implies \boldsymbol{A} = \boldsymbol{QR} \implies \boldsymbol{R} = \boldsymbol{Q^{-1}A} = \boldsymbol{Q^{\mathsf{T}}A}$$

Where we used the fact that $\boldsymbol{Q}$ is orthogonal, and therefore its inverse exists, and $\boldsymbol{Q^{-1}} = \boldsymbol{Q^{\mathsf{T}}}$.

As we can see, the Gram-Schmidt algorithm leads to a decomposition of a square, full-rank matrix into an orthogonal matrix $\boldsymbol{Q}$ and an upper triangular matrix $\boldsymbol{R}$, which is called the $\boldsymbol{QR}$ decomposition, naturally.

7.6 Rotation Matrices

Thinking of matrix multiplication of the type $\boldsymbol{Ax} = \boldsymbol{b}$ as a transformation $\boldsymbol{A} : \boldsymbol{x} \mapsto \boldsymbol{b}$, we can appreciate that $\boldsymbol{Q}$ has to be a transformation that either rotates or reflects a vector, or shuffles its elements.

An important class of orthonormal matrices are rotation matrices. They are orthonormal matrices because they do not change the norm of the vector being rotated. As shown in Figure 7.4, we can easily derive the rotation matrix in $\boldsymbol{Q_\theta}$ in $\mathbb{R}^2$. Note that the inverse of the rotation matrix would be a matrix that would rotate in the opposite direction: $\boldsymbol{Q_\theta^{-1}}$ should be $\boldsymbol{Q_{-\theta}}$. And, being an orthonormal matrix, it should also be the same as $\boldsymbol{Q_\theta}^{\mathsf{T}}$.

$$\boldsymbol{Q_\theta} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad |\boldsymbol{Q_\theta}| = \cos^2\theta + \sin^2\theta = 1$$

$$\boldsymbol{Q_\theta^{-1}} = \boldsymbol{Q_{-\theta}} = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} = \boldsymbol{Q_\theta^{\mathsf{T}}}$$

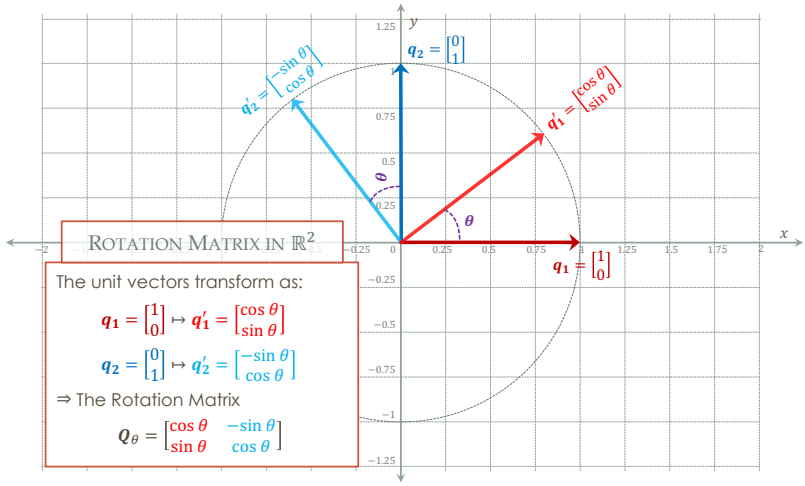


Fig. 7.4 The rotation matrix in $\mathbb{R}^2$ can be written down by looking at where the unit vectors go under a rotation through a specified angle.

Rotation in $\mathbb{R}^3$ is defined by three angles, the pitch, roll and yaw, as they are known in flying. The matrix can actually be written as the product of three independent rotations, but it is probably of not much interest in computer science, except, perhaps for developing flight simulators.

Linearity in Relativity

We talked about rotation matrices. In $\mathbb{R}^3$, for instance, if we have a vector $\boldsymbol{x}$ that gets rotated by a yaw (ψ), pitch (θ) and roll (ϕ) and ends up as $\boldsymbol{x}'$, we have a linear transformation: $\boldsymbol{x}' = \boldsymbol{R}(\psi, \theta, \phi)\boldsymbol{x}$. If all the rotation angles are zero ($\psi = \theta = \phi = 0$), then clearly $\boldsymbol{R} = \boldsymbol{I}_3$. Since the physical world we live in is $\mathbb{R}^3$, these vectors ($\boldsymbol{x}$ and $\boldsymbol{x}'$) represent the *position* of a point and how its coordinates change in a rotated frame. These are the so-called position vectors.

Then Einstein came along and said we should be thinking of *events* rather that *positions*. An event takes place at a position and at a time, which would be represented by a four-dimensional vector $\begin{bmatrix} \boldsymbol{x} \\ t \end{bmatrix}$. Then Einstein proceeded to write a "rotation" between two events as something like: $\begin{bmatrix} \boldsymbol{x}' \\ t' \end{bmatrix} = \boldsymbol{L} \begin{bmatrix} \boldsymbol{x} \\ t \end{bmatrix}$ Here, if we think of time as universal (meaning, independent of the position or the state of motion of an observer), then $t' = t$, which says the fourth-row, fourth-column element of this matrix $\boldsymbol{L}$ is 1, the rest being identical to those in $\boldsymbol{R}$. And if the rotation angles are all zero, $\boldsymbol{L} = \boldsymbol{I}_4$.

So far so good. But what comes next is the jaw-dropping, god-level genius of Albert Einstein when he said the matrix $\boldsymbol{L}$ depends on the speed v of the observer. In other words, it is a function of the rotation angles as well as the speed, $\boldsymbol{L}(\psi, \theta, \phi, v)$. This is the so-called Lorentz transformation. If we assume there is no rotation, and the motion is along the z axis, then we get the Lorentz matrix as:

$$\boldsymbol{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \gamma & -\gamma\beta \\ 0 & 0 & -\gamma\beta & \gamma \end{bmatrix}$$

What γ and β are is not so important for our discussion here, but $\beta = \frac{v}{c}$, the velocity as a fraction of the speed of light and $\gamma = \frac{1}{\sqrt{1-\beta^2}}$, the Lorentz factor. What is important to note is that the z coordinate (which is the length along the direction of motion) and time are interconnected, which leads to length contraction and time dilation in such a way as to keep the speed of light a constant. How this transformation is derived and what the physical reasons behind them form the initial part of the paper that literally changed all of physics and our understanding of the universe forever.

Once the transformation is written as a linear transformation, Einstein had all of linear algebra standing behind his equations. There was never going to be a mathematical inconsistency in all the crazily counter-intuitive predications of special relativity. But there is this crucial assumption of linearity, which was introduced on page six, third line in this English translation of the original paper: *"In the first place it is clear that the equations must be linear on account of the properties of homogeneity which we attribute to space and time."*

Although I never explicitly stated it this way so far, I have taken issue with this assumption of linearity, which formed the basis of my first book, The Unreal Universe, with its key finding described in this short video.

Importance of Linearity in Relativity *á la* ChatGPT

One of the foundational assumptions in *special relativity* is the **linearity** of the transformations between inertial reference frames. This assumption plays a crucial role in deriving the Lorentz transformations and establishing the consistency of the theory. Here, we explore why this assumption is both natural and necessary.

In special relativity, inertial frames are related by transformations that preserve the uniformity of space and time. If $x^\mu = (t, x, y, z)$ represents coordinates in one frame and $x'^\mu = (t', x', y', z')$ in another, the transformation $x'^\mu = f(x^\mu)$ must:

1. **Preserve the Principle of Relativity**: All inertial frames are equivalent, meaning the laws of physics are the same in each.
2. **Preserve Homogeneity and Isotropy**: Space and time are uniform and directionally invariant.

These conditions imply that the transformation must act uniformly on all points in spacetime. This uniformity necessitates **linearity**, as nonlinear transformations would introduce position-dependent effects, violating homogeneity.

A linear transformation can be written as:

$$x'^\mu = \Lambda^\mu_{\ \nu} x^\nu,$$

where $\Lambda^\mu_{\ \nu}$ is a constant matrix. Linearity ensures that the spacetime interval:

$$s^2 = -c^2t^2 + x^2 + y^2 + z^2$$

is preserved up to a constant factor, ensuring consistency with the invariant speed of light c.

The linearity assumption simplifies the interplay between space and time while ensuring causality and consistency with experimental observations. Without linearity, transformations would introduce arbitrary distortions, undermining the predictive power of the theory.

In summary, the assumption of linearity in special relativity arises naturally from the symmetry and uniformity of spacetime, enabling the elegant derivation of the Lorentz transformations and the unification of space and time.

8

Review and Recap

The story so far:
In the beginning the Universe was created.
This has made a lot of people very angry and been widely regarded as a bad move.

—Douglas Adams

8.1 A Generalization

We have come about halfway in our Linear-Algebra journey. The way we discussed our topics in this book was explicitly geared toward computer science. Specifically, our vectors and matrices were over the field of real numbers, $\mathbb{R}$. We could generalize it to any field, $\mathbb{F}$ or $\mathbb{K}$, and almost all of our statements and discussions will still stand. However, for computer scientists, the most appropriate field to work with, we believe, is $\mathbb{R}$.

We also presented certain definitions, such as the pivotal one of vector space, incrementally: We defined vectors first, their operations and the properties thereof, and finally said a vector space are merely

a complete collection or closed set of all possible vectors of the same dimension.

We may find some value in taking a step back and defining the concept of vector space in a general and formal way, which starts from scratch and lists all the properties and operations we are looking for. Here is such a definition, which also provides a good summary of our initial, introductory chapters.

Vector Space

Definition: A vector space over a field of $\mathbb{K}$ is a set of elements that have two operations defined on them. We will call the elements "vectors" and use the symbol $\mathcal{S}$ for the set.

1. **Addition** (denoted by $+$): For any two vectors, $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{S}$, addition assigns a third (not necessarily distinct) vector (called the sum) in $\boldsymbol{z} \in \mathcal{S}$. We will write $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y}$.
2. **Scalar Multiplication**: For an element $s \in \mathbb{K}$ (which we will call a scalar) and a vector $\boldsymbol{x} \in \mathcal{S}$, scalar multiplication assigns a new (not necessarily distinct) vector $\boldsymbol{z} \in \mathcal{S}$ such that $\boldsymbol{z} = s\boldsymbol{x}$.

These two operations have to satisfy the properties listed below:

Commutativity: Addition should respect associativity, which means the order in which the vectors appear in the operation does not matter. For any two vectors $\boldsymbol{x_1}, \boldsymbol{x_2} \in \mathcal{S}$,

$$\boldsymbol{x_1} + \boldsymbol{x_2} = \boldsymbol{x_2} + \boldsymbol{x_1}$$

We can make scalar multiplication also commutative by defining $s\boldsymbol{x} = \boldsymbol{x}s$.

Associativity: Both operations should respect associativity, which means we can group and perform the operations in any order we want. For any two scalars $s_1, s_2 \in \mathbb{K}$ and a vector $\boldsymbol{x} \in \mathcal{S}$,

$$s_1 s_2 \boldsymbol{x} = s_1(s_2\boldsymbol{x}) = (s_1 s_2)\boldsymbol{x}$$

and for any three vectors $\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3} \in \mathcal{S}$,

$$\boldsymbol{x_1} + \boldsymbol{x_2} + \boldsymbol{x_3} = (\boldsymbol{x_1} + \boldsymbol{x_2}) + \boldsymbol{x_3} = \boldsymbol{x_1} + (\boldsymbol{x_2} + \boldsymbol{x_3})$$

Distributivity: Scalar multiplication distributes over vector addition. For any scalar $s \in \mathbb{K}$ and any two vectors $\boldsymbol{x_1}, \boldsymbol{x_2} \in \mathcal{S}$,

$$s(\boldsymbol{x_1} + \boldsymbol{x_2}) = s\boldsymbol{x_1} + s\boldsymbol{x_2}$$

and for any two scalars $s_1, s_2 \in \mathbb{K}$ and a vector $\boldsymbol{x} \in \mathcal{S}$,

$$(s_1 + s_2)\boldsymbol{x} = s_1\boldsymbol{x} + s_2\boldsymbol{x}$$

Identity of Addition: The set $\mathcal{S}$ has an identity of addition (called the zero vector) $\mathbf{0} \in \mathcal{S}$ such that $\boldsymbol{x} + \mathbf{0} = \mathbf{0} + \boldsymbol{x} = \boldsymbol{x}$.

Additive Inverse: For every $\boldsymbol{x} \in \mathcal{S}$, the set $\mathcal{S}$ has an inverse of addition $-\boldsymbol{x}$ such that $\boldsymbol{x} + (-\boldsymbol{x}) = \mathbf{0}$, the identity of addition.

Identity of Scalar Multiplication: The field $\mathbb{K}$ has a unit element, 1 (called the multiplicative identity) such that for every $x \in \mathcal{S}$, $1\boldsymbol{x} = \boldsymbol{x}1 = \boldsymbol{x}$.

If $\mathcal{S}$ is a set that satisfies all these conditions, then it is a vector space.

As we can see, we do not say anything about what the "vectors" are. In particular, we are free to call our matrices by the name vectors also, and consider a collection of matrices that satisfy these conditions a vector space. What are such matrices? They are matrices of the same dimensions, of which our vectors are examples.

Note that the existence of a third operation, vector-vector multiplication (such as dot products, or matrix multiplication, more generally) is not required for our set of vectors $\mathcal{S}$ to be considered a vector space.

8.2 Product Rules: Transposes and Inverses

The product rule of transposes and inverses is similar: When we transpose (or invert) a product, we get the operations in the reverse order on the factor matrices.

$$(\boldsymbol{AB})^\mathsf{T} = \boldsymbol{B}^\mathsf{T}\boldsymbol{A}^\mathsf{T} \quad \text{and} \quad (\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$$

The product rule for transposes gives us two interesting symmetric matrices from any odd matrix:

$$(\boldsymbol{A}^\mathsf{T}\boldsymbol{A})^\mathsf{T} = \boldsymbol{A}^\mathsf{T}(\boldsymbol{A}^\mathsf{T})^\mathsf{T} = \boldsymbol{A}^\mathsf{T}\boldsymbol{A} \quad \text{and} \quad (\boldsymbol{A}\boldsymbol{A}^\mathsf{T})^\mathsf{T} = (\boldsymbol{A}^\mathsf{T})^\mathsf{T}\boldsymbol{A}^\mathsf{T} = \boldsymbol{A}\boldsymbol{A}^\mathsf{T}$$

$\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$ is the Gram matrix. $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$, $\boldsymbol{A}\boldsymbol{A}^\mathsf{T}$ and $\boldsymbol{A}^\mathsf{T}$ all have the same rank as $\boldsymbol{A}$. In particular, if $\boldsymbol{A}$ has full column rank, $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$ is a full-rank, square matrix. And if $\boldsymbol{A}$ has full row rank, $\boldsymbol{A}\boldsymbol{A}^\mathsf{T}$ is a full-rank, square matrix.

Combining the invertibility of $A^\mathsf{T}A$ and AA^T with the product rule for inverses, we can come up with left and right inverses.

$$(A^\mathsf{T}A)^{-1}A^\mathsf{T}A = I \implies A^{-1}_{\text{Left}} = (A^\mathsf{T}A)^{-1}A^\mathsf{T}$$

In writing the first part of the expression above, we assume that $A^\mathsf{T}A$ (the Gram matrix) is invertible, which means A has full column rank. Similarly, for full-row-rank matrix A has a right inverse:

$$AA^\mathsf{T}(AA^\mathsf{T})^{-1} = I \implies A^{-1}_{\text{Right}} = A^\mathsf{T}(AA^\mathsf{T})^{-1}$$

Keep in mind that A is not necessarily square, and does not, therefore, have a double-sided inverse A^{-1}. But the left inverse can take the place of an inverse, as it did in Eqn (7.6).

While describing the Gram-Schmidt process, we wrote down the projection operator in a special form in Eqn (7.8):

$$a_{2\|} = a_1 \left(a_1^\mathsf{T} a_1\right)^{-1} a_1^\mathsf{T} a_2$$

Notice a similar pattern emerging? The projection of a_2 onto a_1 is the *left* inverse of $a_1 = a^{-1}_{1\,\text{left}} \in \mathbb{R}^{1\times n}$ multiplying $a_2 \in \mathbb{R}^n$ on the *left*, giving us a scalar (which then scales a_1 to make the projection a vector).

8.3 Column Picture of Matrix Multiplication

While describing matrix multiplication, we presented many different ways of understanding it. All these different pictures are indeed equivalent. First, we started with an element-wise definition. We then saw that vectors, being matrices of single columns, obeyed the same multiplication rules, and realized that vector dot product is a matrix multiplication. We then redefined matrix multiplication as dot products of the rows of the first matrix with the columns of the second.

We also looked at the row and column pictures of matrix multiplication: Multiplication on the left is the same as the linear combinations of the rows of the second matrix, and multiplication on the right is the same as the linear combinations of the columns of the first. Of all these different views of matrix multiplication, the column picture is most useful one. Let's look at it once more, and apply it to our

understanding of some of the crucial and basic concepts of Linear Algebra.

In the product $\boldsymbol{AX} = \boldsymbol{B}$ with $\boldsymbol{A} \in \mathbb{R}^{m \times k}$, $\boldsymbol{X} \in \mathbb{R}^{k \times n}$ and consequently $\boldsymbol{B} \in \mathbb{R}^{m \times n}$, the columns of the product $\boldsymbol{B}$ are linear combinations of the columns of $\boldsymbol{A}$, taken with the scalars in each column of $\boldsymbol{X}$.

More specifically, in our favorite equation $\boldsymbol{Ax} = \boldsymbol{b}$, $\boldsymbol{b}$ is a linear combination of the columns of $\boldsymbol{A}$ scaled by the components of $\boldsymbol{x}$. This view makes it possible for us to unearth a system of linear equations hiding behind the definitions of span and linear independence of vectors.

8.3.1 Span of Vectors

Earlier, we defined the span of vectors in Eqn (6.2). For our own nefarious ulterior motives, let's recast the definition using other notations and symbols as in the following equation:

$$\text{Given } n \text{ vectors } \boldsymbol{a_i} \in \mathbb{R}^m, \text{span}(\{\boldsymbol{a_i}\}) = \left\{ \boldsymbol{b} \mid \boldsymbol{b} = \sum_{i=1}^{n} x_i \boldsymbol{a_i} \right\} \quad \text{for any } n \text{ scalars } x_i \in \mathbb{R} \tag{8.1}$$

We switched from our previous notation of vectors (from $\boldsymbol{x_i} \rightarrow \boldsymbol{a_i}$) and scaling factors (from $s_i \rightarrow x_i$). Can we unearth our favorite matrix equation $\boldsymbol{Ax} = \boldsymbol{b}$ from this new definition?

As we can see, we may redefine the span of a set of vectors as follows: If we arrange a set of vectors as the columns of a matrix $\boldsymbol{A}$, then their span is all possible vectors we can get as the product $\boldsymbol{Ax} = \boldsymbol{b}$ for all possible vectors $\boldsymbol{x}$. In the very next chapter, we will have a lot more to say about this way of looking at the span.

8.3.2 Linear Independence

The definition of linear independence in Eqn (6.3) also hides within it a system of linear equations. In order to rebrand linear dependency also as a matrix equation that we will have the pleasure of meeting again later on, let's consider it for the columns $\boldsymbol{a_i} \in \mathbb{R}^m$ of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, where the scalars are called $x_i \in \mathbb{R}$. The condition now

The Zero Vector

The zero vector $\mathbf{0} \in \mathbb{R}^n$ has the same role as the number zero in $\mathbb{R}$: It is the identity of vector addition. It has few special properties. It is the only vector that is parallel and perpendicular (or, to be more precise, collinear and orthogonal) to all other vectors *at the same time*.

As we know by now, a vector is collinear with a scaled version of itself. And, since $\mathbf{0} = s\boldsymbol{x}$ for $s = 0$ for any $\boldsymbol{x}$, we can say that $\mathbf{0}$ is collinear with any $\boldsymbol{x}$. Furthermore, if $\boldsymbol{x} \cdot \boldsymbol{y} = 0$, $\boldsymbol{x}$ is orthogonal to $\boldsymbol{y}$. Sure enough, $\boldsymbol{x} \cdot \mathbf{0} = 0$ for any $\boldsymbol{x}$, proving that $\mathbf{0} \perp \boldsymbol{x}$.

When it comes to vector spaces, $\mathbf{0} \in \mathbb{R}^0$ is the smallest vector space we can think of. It has only one vector. Scale it by any $\boldsymbol{s} \in \mathbb{R}$, we get $\mathbf{0}$. Add any "two" members of this tiny vector space, we get $\mathbf{0} + \mathbf{0} = \mathbf{0}$. Therefore, this vector space (let's call it $\mathcal{Z}$) is indeed closed under scalar multiplication and vector addition, as all vector spaces are supposed to be.

The strange thing about it is the dimension of $\mathcal{Z}$. It has to be zero because, as we know, a vector space of dimension one is a line, not a point. What is the basis of $\mathcal{Z}$? We might think that it is the set $\{\mathbf{0}\}$, in which case the cardinality of the set has to be zero, because the dimension of a vector space defined as the cardinality of its basis. It looks as though $\mathbf{0}$ does not count. Does it mean that a set of real numbers $\{0\}$ has zero members in it? Or that $\{0, 1\}$ has only one member, despite the fact that we can clearly see two of them? Are our eyes deceiving us?

The weirdness of the zero vector extends to the notion of vector subspaces as well. All vector subspaces contain the zero vector, as do all vector spaces. Otherwise they would not be closed under scalar multiplication (with $s = 0$). Therefore, the intersection of any two subspaces is at least $\{\mathbf{0}\}$. Are we justified in calling it a null set because we know that its cardinality is zero?

These contradictions lead to the inescapable conclusion that the vector space containing only the zero vector $\mathcal{Z} = \{\mathbf{0}\}$ does not need a basis at all. Remember, the basis set $\{\mathbf{0}\}$ of $\mathcal{Z}$ needs to satisfy two conditions: (1) It should space the space (which it does), (2) The vectors in the basis set need to be linearly independent. But the zero vector is not linearly independent of itself, and therefore cannot form a basis of anything!

There is much more to zero than meets the eye.

becomes:

$$\sum_{i=1}^{n} x_i \boldsymbol{a_i} = \mathbf{0} \implies \boldsymbol{A}\boldsymbol{x} = \mathbf{0} \tag{8.2}$$

In other words, if we can find nontrivial solutions (meaning $\boldsymbol{x} \neq \mathbf{0}$) for the matrix equation $\boldsymbol{A}\boldsymbol{x} = \mathbf{0}$, then the columns of $\boldsymbol{A}$ are *not* linearly independent.

8.4 Named Algorithms

As we have encountered a few algorithms (by which we mean step-by-step instructions to achieve an objective), it may be useful now to compare them in terms of their purpose and end products, which is what we have in Table 8.1.

When we started solving systems of linear equations, we encountered the Gaussian-elimination algorithm. Its objective is to get to the row echelon form (REF) of the matrix (typically, the augmented matrix $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$ of the system $\boldsymbol{Ax} = \boldsymbol{b}$). Once we have the REF, we can count the number of pivots as $\text{rank}(\boldsymbol{A})$, and fully solve the system by back substitution.

As a byproduct, Gaussian elimination produces a decomposition as well, $\boldsymbol{A} = \boldsymbol{PLU}$. $\boldsymbol{P}$, being a permutation matrix, has a determinant of ± 1, while $\boldsymbol{L}$, being the inverse of a bunch of elementary matrices, has a determinant of 1. Therefore, $|\boldsymbol{A}|$ can be computed from its REF, which is the upper triangular matrix $\boldsymbol{U}$ as the product of its diagonal elements, multiplied by $|\boldsymbol{P}| = \pm 1$.

Table 8.1 Named algorithms and their uses

Name	Type of Operation	Uses	Decomposition
Gaussian Elimnation	Elementary Row Operations	Row Echelon Form (REF). Solve equations using back substitution. Determine rank. Compute determinants.	$\boldsymbol{PLU}$
Gauss-Jordan Elimination	Elementary Row Operations	Reduced Row Echelon Form (RREF). Solve equations. Determine rank. Compute inverses.	None
Gram-Schmidt Process	Elementary Column Operations	Orthonormal basis matrix. Works only on square, invertible $\boldsymbol{A}$.	$\boldsymbol{QR}$

Gauss-Jordan elimination is an extension of Gaussian elimination, where we scale the pivot rows to get the pivots to have a value of one. We also get rid of all the matrix elements above the pivot in each pivot column by subtracting appropriate multiples of the pivot row. Our objective is to get an identity matrix if we were to look only at the

pivot rows and columns. The resulting form is called the reduced row echelon form or RREF. Once we have the RREF of the augmented matrix of a system of linear equations, we can read off the solutions from the pivot rows. Gauss-Jordan elimination also may produce a decomposition, but we are not really concerned about it.

The RREF that Gauss-Jordan elimination produces can also tell us about the rank, which is the number of pivots (also equal to the sum of pivots because they all have value equal to one). But the real interesting usage of this algorithm is in inverting a matrix, $\boldsymbol{A}$. For this purpose, we augment it with $\boldsymbol{I}$ and run the G-J algorithm on $[\boldsymbol{A} \mid \boldsymbol{I}]$. When the $\boldsymbol{A}$ part of the augmented matrix becomes $\boldsymbol{I}$, we have the inverse of $\boldsymbol{A}$ where $\boldsymbol{I}$ used to be.

$$[\boldsymbol{A} \mid \boldsymbol{I}] \xrightarrow{\text{RREF}} [\boldsymbol{I} \mid \boldsymbol{A}^{-1}]$$

If the algorithm fails to produce $\boldsymbol{I}$, it means that it could not find a pivot in at least one row, and the matrix is not invertible; it is rank-deficient.

The third algorithm, the Gram-Schmidt process, is different from the first two in the sense that it does elementary column operations (not row ones as the other two). However, it is not such a big difference because a column operation is a row operation on the transpose. The purpose of Gram-Schmidt is to produce a matrix that is orthogonal. In the process, it creates the so-called $\boldsymbol{QR}$ decomposition, where $\boldsymbol{R}$, an upper triangular matrix, is the inverse of all the elementary operations performed by the matrix. But the process does not really have to keep track of the operations and take its inverse because of the basic property of orthogonal matrices, $\boldsymbol{Q}^{-1} = \boldsymbol{Q}^{\mathsf{T}}$. Therefore,

$$\boldsymbol{A} = \boldsymbol{QR} \implies \boldsymbol{R} = \boldsymbol{Q}^{-1}\boldsymbol{A} = \boldsymbol{Q}^{\mathsf{T}}\boldsymbol{A}$$

Once $\boldsymbol{Q}$ is obtained, $\boldsymbol{R}$ is only a matrix multiplication away.

We have created a neat little Table 8.1 to compare these three algorithms. It may be useful as a quick reference to their steps, purposes and usages.

8.5 Pivots, Ranks, Inverses and Determinants

Another topic worthy of a recap is the interconnected story of pivots, ranks, inverses and determinants of matrices, and how it relates to the

solvability of the underlying system of linear equations. We looked at each of these topics at various points in our discussions so far. Because of the intimate interconnections among them, it is best to summarize it all as we do in Figure 8.1.

Matrices come in different shapes: Square, "tall" and "wide," as we described way back in §5.1.2 (page 90). Each shape of matrix can be either full-rank or rank-deficient, thereby giving us six possible permutations. When we run Gauss-Jordan elimination on these matrices, depending on their rank, we will get different canonical forms, as we discussed in §5.2.2 (page 95). Figure 8.1 is an attempt to put all these things together, and comment on the solvability of the underlying system of linear equations.

Corresponding to each permutation[1] of the matrix shape and its rank status, we have a row in Figure 8.1. Referring to these rows, we can make the following statements.

The only matrix that can be inverted is in the first row. It has a nonzero determinant, and it leads to a unique solution for the system of equations that it represents. The matrix in the second row also has a determinant, but it is zero. The matrix is not invertible. Note that its RREF has at least one zero row at the bottom. If the entry on the augmented part of the matrix is also zero, then the equations are consistent, but we do not have enough of them. Therefore, we get an infinite number of solutions. We can see it because the RREF has $\boldsymbol{F}$ in it, which indicates columns with no pivots and therefore free variables. On the other hand, if the constant in this zero row is nonzero, we have inconsistency and hence no solution.

We can make a similar statement about the third row also: The system can be consistent (with $0 = 0$ in the bottom rows), in which case we get a unique solution, or it can be inconsistent, with no solutions. Note the absence of $\boldsymbol{F}$ in the RREF, indicating that we will never get an infinity of solutions.

The fourth row has a matrix with RREF with both free variables and zero rows. So we have the possibilities of no solution or an infinite number of solutions.

[1] To be very pedantic about it, it is not a permutation but an element of the Cartesian product of the sets of matrix shapes and rank statuses.

	Shape	Rank	RREF	Solvability
1	Square, Full-rank: $\boldsymbol{A} \in \mathbb{R}^{n\times n}$	n	$\boldsymbol{I}_n$	Unique Solution
2	Square, Rank-deficient: $\boldsymbol{A} \in \mathbb{R}^{n\times n}$	$r < n$	$\begin{bmatrix} \boldsymbol{I}_r \oplus \boldsymbol{F}_{r\times(n-r)} \\ \hline \boldsymbol{0}_{(n-r)\times n} \end{bmatrix}$	No Solution or Infinity of Solutions
3	Tall, Full-rank: $\boldsymbol{A} \in \mathbb{R}^{m\times n}, m > n$	n	$\begin{bmatrix} \boldsymbol{I}_n \\ \hline \boldsymbol{0}_{(m-n)\times n} \end{bmatrix}$	No Solution or Unique Solution
4	Tall, Rank-deficient: $\boldsymbol{A} \in \mathbb{R}^{m\times n}, m > n$	$r < n$	$\begin{bmatrix} \boldsymbol{I}_r \oplus \boldsymbol{F}_{r\times(n-r)} \\ \hline \boldsymbol{0}_{(m-r)\times n} \end{bmatrix}$	No Solution or Infinity of Solutions
5	Wide, Full-rank: $\boldsymbol{A} \in \mathbb{R}^{m\times n}, m < n$	m	$\begin{bmatrix} \boldsymbol{I}_m \oplus \boldsymbol{F}_{m\times(n-m)} \end{bmatrix}$	Infinity of Solutions
6	Wide, Rank-deficient: $\boldsymbol{A} \in \mathbb{R}^{m\times n}, m < n$	$r < m$	$\begin{bmatrix} \boldsymbol{I}_r \oplus \boldsymbol{F}_{r\times(n-r)} \\ \hline \boldsymbol{0}_{(m-r)\times n} \end{bmatrix}$	No Solution or Infinity of Solutions

Fig. 8.1 Properties and solvability of the linear equations $\boldsymbol{Ax} = \boldsymbol{b}$ based on the RREF of the coefficient matrix $\boldsymbol{A}$,

When it comes to the full-rank "wide" matrix, we have no zero row, which means we will never have inconsistency. But we never have enough equations either (as indicated by the free variable part in the RREF), and we always have an infinity of solutions.

Rank-deficient "wide" matrix leads to situations similar to rows 2 and 4, with no solution or infinite number of solutions. To reiterate, when a matrix is rank-deficient (as in rows 2, 4 or 6), regardless of its shape, we either have no solution (inconsistent equations) or an infinite number of solutions (not enough equations).

Figure 8.1 is color-coded for easy reference: Whenever we see free variables (corresponding to columns with no pivots) in RREF, shown in purple, an infinity of solutions is a possibility, obviated only by the existence of inconsistent equations (shown in red) as indicated by zero rows in the $\boldsymbol{A}$ part of $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$, with the $\boldsymbol{b}$ part nonzero. The only instances where we get a unique solution is when we can have an $\boldsymbol{I}_n$ in the RREF, shown in green. We can also see that a tall matrix can never have an infinity of solutions and a wide one can never have a unique solution.

8.6 Two Geometries

When we started visualizing linear equations (Figure 4.1), we were doing it in the coordinate space, where we have points and lines and planes. A linear equation in the *coordinate space* of $\mathbb{R}^2$, for instance, is a line. As we know, each row of the augmented matrix $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$ represents an equation, which has a shape in the coordinate space. In other words, when we think of equations, we are thinking rows, and we can consider the thinking process the row picture of $\boldsymbol{Ax} = \boldsymbol{b}$. The dimension of this coordinate space is the same as the number of unknowns.

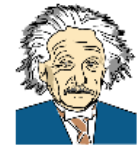
We have a wholly separate *vector* space $\mathbb{R}^2$ as well, where we have all possible vectors with two components and nothing else. When we think of $\boldsymbol{Ax} = \boldsymbol{b}$ as a system of equations demanding that $\boldsymbol{b}$ be the right linear combination of the *columns* of $\boldsymbol{A}$ as specified by $\boldsymbol{x}$, we are working with the latter. If we have two rows, our column vectors have two elements, and we are working with the vector space $\mathbb{R}^2$, regardless of how many unknowns we may have. If we have three equations, as in Figure 6.7, we are in the vector space $\mathbb{R}^3$, even though we have only two unknowns. This mode of thinking can be considered the column picture of systems of linear equations. This column picture is the more sophisticated, university-level thinking we are trying to foster in this course.

9

The Four Fundamental Spaces

Time and space are modes by which we think and not conditions in which we live.

—Albert Einstein



Our geometric view started with the notion that we can think of the solution to the system of linear equations $\boldsymbol{Ax} = \boldsymbol{b}$ as a quest for that special $\boldsymbol{x}$ whose components become the coefficients in taking the linear combination of the columns of $\boldsymbol{A}$ to give us $\boldsymbol{b}$. Since this opening statement was a bit too long and tortured, let's break it down. Here is what it means:

- The product of the matrix multiplication $\boldsymbol{Ax}$ is a linear combination of the columns $\boldsymbol{a_i}$ of $\boldsymbol{A}$.

- The linear combination is taken with coefficients equal to the components of $\boldsymbol{x} = x_i$. In other words, the product is:

$$\boldsymbol{Ax} = \sum_{i=1}^{n} x_i \boldsymbol{a_i}$$

- Our system of linear equations $\boldsymbol{Ax} = \boldsymbol{b}$, therefore becomes the condition:

$$\sum_{i=1}^{n} x_i \boldsymbol{a_i} = \boldsymbol{b}$$

Solving this system means finding the right x_i so that the linear combination satisfies the condition.

We also learned that the collection of all possible linear combinations of a set of vectors is called the span of those vectors, and it is a vector subspace, which is a subset of a vector space. For example, $\mathbb{R}^m$ is a space[1] and n vectors $\boldsymbol{a_i} \in \mathbb{R}^m$ span a subspace contained within $\mathbb{R}^m$. Let's call this subspace $\mathcal{C} \subseteq \mathbb{R}^m$. If, among the n vectors that span $\mathcal{C}$, only $r \leq n$ of them are linearly independent, then those r vectors form a basis for $\mathcal{C}$ and the dimension of this subspace $\mathcal{C}$ is indeed r (which is the cardinality of the basis). In fact, even if $n > m$, the n vectors span only a subspace $\mathcal{C} \subset \mathbb{R}^m$ if the number of independent vectors $r < m$. Note that r can never be greater than m, the number of components of each of our vectors $\boldsymbol{a_i} \in \mathbb{R}^m$.

9.1 Column Space

The choice of the symbols in the previous paragraph was not an accident. We are ready to introduce the concept of the Column Space of a matrix.

Column Space

Definition: The column space $\mathcal{C}$ of a matrix is the span of its columns.

$$\text{For a matrix } \boldsymbol{A} = \left[\boldsymbol{a_i}\right] \in \mathbb{R}^{m \times n},\ \mathcal{C}(\boldsymbol{A}) \stackrel{\text{def}}{=} \left\{ \boldsymbol{z} \mid \boldsymbol{z} = \sum_{i=1}^{n} x_i \boldsymbol{a_i} \right\}$$

where $x_i \in \mathbb{R}$. Note that $\mathcal{C}(\boldsymbol{A}) \subseteq \mathbb{R}^m$, where m is the number of *rows* of $\boldsymbol{A}$ because each of its column vectors $\boldsymbol{a_i} \in \mathbb{R}^m$.

Are the columns of $\boldsymbol{A}$ ($\boldsymbol{a_i}$) a basis for $\mathcal{C}(\boldsymbol{A})$? Remembering that the basis of a space or subspace is the *minimal* set of vectors that span it, we can see that a decent basis for $\mathcal{C}(\boldsymbol{A})$ would be the linearly

[1] Once again, we will be dropping the ubiquitous "vector" prefix from spaces and subspaces.

independent columns of $\boldsymbol{A}$. How many of them are there? We saw earlier that the number of linearly independent columns of $\boldsymbol{A}$ is its rank, rank$(\boldsymbol{A})$. To be more precise, we should call it the column-rank of $\boldsymbol{A}$, but we also saw that the column and row ranks were the same. Finally, the dimension of a space or subspace is the number of vectors in its basis. Therefore, the dimension of the column space is the rank. Conversely, we can define rank$(\boldsymbol{A})$ as the dimension of $\mathcal{C}(\boldsymbol{A})$.

9.1.1 Significance of Column Space

As we said in the introduction to this chapter, the solution to $\boldsymbol{Ax} = \boldsymbol{b}$ is that $\boldsymbol{x}$ which produces the linear combination of the columns of $\boldsymbol{A}$ equalling $\boldsymbol{b}$. We then defined the column space $\mathcal{C}(\boldsymbol{A})$ as *all* possible linear combinations of the columns of $\boldsymbol{A}$. In other words, if $\boldsymbol{b} \notin \mathcal{C}(\boldsymbol{A})$, we will never be able to find a solution vector $\boldsymbol{x}$.

We can, in fact, turn the last statement about the solvability of $\boldsymbol{Ax} = \boldsymbol{b}$ and $\mathcal{C}(\boldsymbol{A})$ into an alternative definition of column space: $\mathcal{C}(\boldsymbol{A})$ is the set of all vectors $\boldsymbol{b}$ if there exists $\boldsymbol{x} \neq \boldsymbol{0}$ such that $\boldsymbol{Ax} = \boldsymbol{b}$.

Note that the elementary column operations, which we used in the Gram-Schmidt process, do not change the column space because they are all about taking linear combinations of the columns or scaling them. The span of the columns is unaffected by such linear operations. Therefore, in $\boldsymbol{A} \xrightarrow{\text{G-S}} \boldsymbol{Q}$, both $\boldsymbol{A}$ and its orthogonal cousin $\boldsymbol{Q}$ have the same column space: $\mathcal{C}(\boldsymbol{A}) = \mathcal{C}(\boldsymbol{Q})$.

On the other hand, the elementary row operations in Gaussian and Gauss-Jordan eliminations do affect the column space. What is unaffected would be the linear combination of the rows, which we will call the row space. But before getting to it, let's look at the null space.

9.2 Null Space

So far, we have concerned ourselves mostly with the solution of $\boldsymbol{Ax} = \boldsymbol{b}$. Let's now take a look at the so-called homogeneous system of linear equations $\boldsymbol{Ax} = \boldsymbol{0}$. In fact, we did come across this system when we were talking about linear independence in the last chapter, rebranding its condition as a matrix equation in Eqn (8.2). If $\boldsymbol{Ax} = \boldsymbol{0}$

for some nonzero vector $\boldsymbol{x}$ (in other words, if the homogeneous system has a nontrivial solution), then the columns of $\boldsymbol{A}$ are not linearly independent.

The collection of *all* vectors $\boldsymbol{x}$ that solve $\boldsymbol{Ax} = \boldsymbol{0}$ (called the solution set) is the null space of $\boldsymbol{A}$, denoted by $\mathcal{N}(\boldsymbol{A})$. Let's put down this statement as the definition of the null space.

Null Space

Definition: The null space $\mathcal{N}$ of a matrix $\boldsymbol{A}$ is the complete set of all vectors that form the solutions to the homogeneous system of linear equations $\boldsymbol{Ax} = \boldsymbol{0}$.

$$\mathcal{N}(\boldsymbol{A}) \stackrel{\text{def}}{=} \{\boldsymbol{x} \mid \boldsymbol{Ax} = \boldsymbol{0}\}$$

Note that if we have two vectors in the null space, then all their linear combinations are also in the null space. $\boldsymbol{x_1}, \boldsymbol{x_2} \in \mathcal{N}(\boldsymbol{A}) \implies \boldsymbol{x} = s_1\boldsymbol{x_1} + \boldsymbol{s_2}\boldsymbol{x_2} \in \mathcal{N}(\boldsymbol{A})$. If $\boldsymbol{Ax_1} = 0$ and $\boldsymbol{Ax_2} = 0$, then $\boldsymbol{A}(s_1\boldsymbol{x_1} + \boldsymbol{s_2}\boldsymbol{x_2}) = 0$ because of the basic linearity condition we learned way back in the first chapter. What all this means is that the null space is indeed a vector subspace. Furthermore, $\mathcal{N}(\boldsymbol{A})$ is complete, by definition. In other words, we will not find a vector $\boldsymbol{x}$ that is not in $\mathcal{N}(\boldsymbol{A})$ such that $\boldsymbol{Ax} = \boldsymbol{0}$. In the cold hard language of mathematics, $\boldsymbol{x} \in \mathcal{N}(\boldsymbol{A}) \iff \boldsymbol{Ax} = \boldsymbol{0}$.

For any and all vector $\boldsymbol{x} \in \mathcal{N}(\boldsymbol{A})$, in the null space of $\boldsymbol{A}$, its dot product with the rows of $\boldsymbol{A}$ are zero. If $\boldsymbol{r}_i^{\mathsf{T}}$ is a row of $\boldsymbol{A}$, we have:

$$\boldsymbol{Ax} = \boldsymbol{0} \implies \begin{bmatrix} \vdots \\ \boldsymbol{r}_i^{\mathsf{T}} \\ \vdots \end{bmatrix} \boldsymbol{x} = \boldsymbol{0} \implies \boldsymbol{r}_i^{\mathsf{T}}\boldsymbol{x} = \boldsymbol{0}\ \forall\, i \tag{9.1}$$

The dot product being zero means that the vectors in $\mathcal{N}(\boldsymbol{A})$ and the rows of $\boldsymbol{A}$ are orthogonal. It then follows that any linear combinations of the rows of $\boldsymbol{A}$ are also orthogonal to the vectors in $\mathcal{N}(\boldsymbol{A})$. And, the collection of the linear combinations of the rows of $\boldsymbol{A}$ is indeed their span, which is a subspace we will soon call the row space of $\boldsymbol{A}$.

9.3 Row Space

The set of all possible linear combinations of the rows of a matrix is called its row space, with one little complication. Since our vectors

are all column vectors, and the rows of a matrix are definitely not columns, we think of the transpose of the rows as the vectors whose linear combinations make the row space. Equivalently, we may transpose the matrix first and call the column space of the transpose the row space of the original. That is to say, the row space of $\boldsymbol{A}$ is $\mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$, which is the notation we will use for row space. Since the row rank is the same as column rank, the number of linearly independent rows in $\boldsymbol{A}$ is its rank, which is the same as the dimension of $\mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$.

9.3.1 Significance of Row Space

If all vectors in a subspace are orthogonal to all vectors in another subspace, then we call these two subspaces orthogonal. Earlier, we established that the row and null spaces of a matrix are orthogonal to each other. In fact, the row and null spaces are orthogonal complements of each other, which means that all the vectors in the containing space that are orthogonal to all the vectors in the null space are in the row space. Again, this complex statement needs some dismantling (and convincing): What this statement says, in mathematical language, is the following:

- We have a matrix $\boldsymbol{A}$, with a row space $\mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$ and null space $\mathcal{N}(\boldsymbol{A})$.
- $\boldsymbol{x} \notin \mathcal{N}(\boldsymbol{A})$, $\boldsymbol{y} \in \mathcal{N}(\boldsymbol{A})$, and $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{y} = 0 \implies \boldsymbol{x} \in \mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$.

The proof of this assertion is probably beyond the remit of this book on applied Linear Algebra, and is, therefore, included as an optional box on "**Orthogonal Complementarity**." What it means, however, is important for us to know. Let's first illustrate it using an example.

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \implies \text{Basis for } \mathcal{C}(\boldsymbol{A}^{\mathsf{T}}) = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \right\}$$

Clearly, $\in \mathbb{R}^{3\times 3}$, $\text{rank}(\boldsymbol{A}) = 2$ and therefore $\mathcal{C}(\boldsymbol{A}^{\mathsf{T}}) \subset \mathbb{R}^3$ with the dimension 2, which is why we have two basis vectors, the pivot rows of $\boldsymbol{A}$. In this case, $\boldsymbol{A}$ is already in its RREF.

In order to compute $\mathcal{N}(\boldsymbol{A})$, we have to find the solutions to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$. For our $\boldsymbol{A}$, we can see that all vectors with the first two components zero and a nonzero third component will give $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$. We can

Orthogonal Complementarity

Before attempting to prove that $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$ and $\mathcal{N}(\boldsymbol{A})$ are orthogonal complements, it is perhaps best to spell out what it means. It means that all the vectors in $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$ are orthogonal to all the vectors in $\mathcal{N}(\boldsymbol{A})$, which is the orthogonal part. It also means if there is a vector orthogonal to *all* vector in $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$, it is in $\mathcal{N}(\boldsymbol{A})$, which is the complement part. Here are the two parts in formal lingo:

1. $\boldsymbol{y} \in \mathcal{C}(\boldsymbol{A}^\mathsf{T})$ and $\boldsymbol{x} \in \mathcal{N}(\boldsymbol{A}) \implies \boldsymbol{x}^\mathsf{T}\boldsymbol{y} = 0$
2. (a) $\boldsymbol{y} \neq \boldsymbol{0} \in \mathcal{C}(\boldsymbol{A}^\mathsf{T})$ and
   (b) $\boldsymbol{x}^\mathsf{T}\boldsymbol{y} = 0 \,\forall\, \boldsymbol{y} \in \mathcal{C}(\boldsymbol{A}^\mathsf{T}) \implies \boldsymbol{x} \in \mathcal{N}(\boldsymbol{A})$
3. Or, conversely,
   (a) $\boldsymbol{x} \in \mathcal{N}(\boldsymbol{A})$ and
   (b) $\boldsymbol{x}^\mathsf{T}\boldsymbol{y} = 0 \,\forall\, \boldsymbol{x} \in \mathcal{N}(\boldsymbol{A}) \implies \boldsymbol{y} \in \mathcal{C}(\boldsymbol{A}^\mathsf{T})$

The first part is easy to prove, and we did it in Eqn (9.1), which just says that each element of $\boldsymbol{Ax}$ has to be zero for $\boldsymbol{Ax} = \boldsymbol{0}$ to be true.

To prove the second part, let's think of the matrix $\boldsymbol{A}^\mathsf{T}$ as being composed of *columns* $\boldsymbol{c_i}$. If $\boldsymbol{y} \in \mathcal{C}(\boldsymbol{A}^\mathsf{T})$, as in condition 2(a), we can write $\boldsymbol{y}$ as a linear combination of $\boldsymbol{c_i}$:

$$\boldsymbol{y} = \sum s_i \boldsymbol{c_i} = \boldsymbol{A}^\mathsf{T}\boldsymbol{s}$$

for some s_i, which we put together as a column vector $\boldsymbol{s}$. If $\boldsymbol{x}^\mathsf{T}\boldsymbol{y} = 0$ as condition 2(b) says, then $\boldsymbol{x}^\mathsf{T}\boldsymbol{A}^\mathsf{T}\boldsymbol{s} = 0 \implies (\boldsymbol{Ax})^\mathsf{T}\boldsymbol{s} = 0$. We know that $\boldsymbol{s} \neq \boldsymbol{0}$ because it is the linear combination that gives $\boldsymbol{y} \neq \boldsymbol{0}$. The only way $(\boldsymbol{Ax})^\mathsf{T}\boldsymbol{s} = 0$ can be true for *all* $\boldsymbol{y} \in \mathcal{C}(\boldsymbol{A}^\mathsf{T})$ is if $\boldsymbol{Ax} = 0 \implies \boldsymbol{x} \in \mathcal{N}(\boldsymbol{A})$.

The third part says $\boldsymbol{Ax} = \boldsymbol{b}$ and $\boldsymbol{y} \perp \boldsymbol{x}$. Let's assume that $\boldsymbol{y} \notin \mathcal{C}(\boldsymbol{A}^\mathsf{T})$, with the hope that it leads to a contradiction. $\boldsymbol{y} \notin \mathcal{C}(\boldsymbol{A}^\mathsf{T}) \implies \boldsymbol{Ay} = \boldsymbol{0}$, because otherwise $\boldsymbol{y}$ would be in the column space. But if $\boldsymbol{y} \notin \mathcal{C}(\boldsymbol{A}^\mathsf{T}), \boldsymbol{y} \in \mathcal{N}(\boldsymbol{A})$, which means it is a linear combination of the vectors in $\mathcal{N}(\boldsymbol{A})$. And it means $\boldsymbol{y}$ cannot be orthogonal to *all* of them, as 3(b) requires. Hence our starting assumption has to be wrong, and $\boldsymbol{y} \in \mathcal{C}(\boldsymbol{A}^\mathsf{T})$.

therefore write:

$$\mathcal{N}(\boldsymbol{A}) \subset \mathbb{R}^3, \text{ of one dimension, with the basis } \left\{ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

Thinking in terms of the coordinate space, we see that the row space is actually the xy-plane and the null space is the z-axis. They are indeed orthogonal complements.

Let's go back to talking about the general case, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\text{rank}(\boldsymbol{A}) = r$. We know that by definition, all vectors $\boldsymbol{x} \in \mathcal{N}(\boldsymbol{A})$ get transformed to $\boldsymbol{0}$ by $\boldsymbol{Ax} = \boldsymbol{0}$. Because of the orthogonal complementarity, we also know that all *orthogonal* vectors are in $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$, the row space

of $\boldsymbol{A}$, as illustrated above with our toy example. All the vectors in the row space get transformed to nonzero vectors in $\mathbb{R}^m$. What about the rest of the vectors? After all, most vectors in $\mathbb{R}^n$ are in neither $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$ nor $\mathcal{N}(\boldsymbol{A})$: They are linear combinations of these two sets of vectors. Let's take such a vector $\boldsymbol{x} = \boldsymbol{x}_\parallel + \boldsymbol{x}_\perp$ where $\boldsymbol{x}_\parallel \in \mathcal{C}(\boldsymbol{A}^\mathsf{T})$ and $\boldsymbol{x}_\perp \in \mathcal{N}(\boldsymbol{A})$.

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}(\boldsymbol{x}_\parallel + \boldsymbol{x}_\perp) = \boldsymbol{A}\boldsymbol{x}_\parallel + \boldsymbol{A}\boldsymbol{x}_\perp = \boldsymbol{A}\boldsymbol{x}_\parallel + \boldsymbol{0} = \boldsymbol{A}\boldsymbol{x}_\parallel$$

What this equation tells is that all vectors not in $\mathcal{N}(\boldsymbol{A})$ also end up in $\mathcal{C}(\boldsymbol{A})$ by $\boldsymbol{A}\boldsymbol{x}$. Moreover, multiple vectors $\boldsymbol{x} \notin \mathcal{N}(\boldsymbol{A})$ get transformed to the same $\boldsymbol{b} \in \mathcal{C}(\boldsymbol{A})$: It is a many-to-one (surjective) mapping.

What is special about the row space is that the mapping from $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$ to $\mathcal{C}(\boldsymbol{A})$ is a one-to-one (injective) mapping. Since it is an important point, let's state it mathematically:

$$\boldsymbol{x} \in \mathcal{C}(\boldsymbol{A}^\mathsf{T}) \text{ and } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \implies \text{for a given } \boldsymbol{x}, \boldsymbol{b} \text{ is unique.}$$

In order to see it, we have to appreciate that $\boldsymbol{b}$ is a linear combination of the linearly independent columns of $\boldsymbol{A}$. For a specific $\boldsymbol{x}$, it is a linear combination with the specified coefficients. As we know, probably from the first chapter, there is only one such linear combination; we cannot get two different linear combinations with the same set of linearly independent vectors and coefficients.

A consequence of $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$ being $\mathcal{N}(\boldsymbol{A})^\perp$ is that their dimensions add up to the dimension of the containing space. Or, $n = r + \dim(\mathcal{N}(\boldsymbol{A})) \implies \dim(\mathcal{N}(\boldsymbol{A})) = n - r$. The dimension of the null space of $\boldsymbol{A}$ is called its nullity, and the statement that the rank and nullity add up to the number of columns of $\boldsymbol{A}$ is the famous rank-nullity theorem.

9.4 Left Null Space

To complete the picture and bring out the beautiful symmetry of the whole system, we will define one more null space, which is $\mathcal{N}(\boldsymbol{A}^\mathsf{T})$, which is also called the left null space. It lives on the same side as the column space, $\mathcal{C}(\boldsymbol{A})$, and no vector in the input space $\mathbb{R}^n$ (other than the zero vector) can reach this space.

9.5 Computing the Four Fundamental Subspaces

If we want to "find" or "compute" the column, row or null space of a matrix, how do we proceed? First of all, we need to clarify what "finding" means, when it comes to a subspace. Here is the clarification: Find the basis for the subspace, state its dimension, and the dimension of the space of which it is a subspace. For instance, we could say, "$\mathcal{C}(\boldsymbol{A}) \subset \mathbb{R}^m, \dim(\mathcal{C}(\boldsymbol{A})) = r$, with the basis $= \{\boldsymbol{v_i}\}$" specifying the three pieces of information we are after.

9.5.1 Row Space

We start with the computation of the row space because it is the easiest one. To find the rank and basis of $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$, we will fall back on Gaussian (or Gauss-Jordan) elimination, to get to the REF (or RREF) of $\boldsymbol{A}$. Remembering that row operations do not change the row space, a good basis for the row space would be the pivot rows of RREF. The dimension of the row space is the number of pivots, AKA rank$(\boldsymbol{A})$ and the containing space is $\mathbb{R}^n$, where n is the number of columns of $\boldsymbol{A}$. The full specification of the row space of $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, rank$(\boldsymbol{A}) = r$ would therefore be $\mathcal{C}(\boldsymbol{A}^\mathsf{T}) \subseteq \mathbb{R}^n$ with dimension $= r$ and basis = the set of r pivot rows.

9.5.2 Column Space

For a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ of rank r, the dimension of $\mathcal{C}(\boldsymbol{A})$ is r, the dimension of the containing space is m, the number of rows. Or, more compactly, $\mathcal{C}(\boldsymbol{A}) \subset \mathbb{R}^m, \dim(\mathcal{C}(\boldsymbol{A})) = r$.

How do we find the basis, or the linearly independent columns of $\boldsymbol{A}$? The pivot columns are the ones that are linearly independent in the reduced forms. It turns out that the corresponding columns in the *original matrix* $\boldsymbol{A}$ are the ones that are linearly independent, and therefore, they form a good basis for $\mathcal{C}(\boldsymbol{A})$. Let's see why.

In order to make the discussion easier, we are going to call the RREF of $\boldsymbol{A}$ by the name $\boldsymbol{R}$. As we know, the solution set of the homogeneous system $\boldsymbol{Ax} = \boldsymbol{0}$ can be obtained from its augmented matrix $\left[\boldsymbol{A} \mid \boldsymbol{0}\right]$, by finding its RREF, which is $\left[\boldsymbol{R} \mid \boldsymbol{0}\right]$. Note that the constants part of the augmented matrices are still $\boldsymbol{0}$, which is different from what happens if we do $\left[\boldsymbol{A} \mid \boldsymbol{b}\right] \xrightarrow{\text{RREF}} \left[\boldsymbol{R} \mid \boldsymbol{b'}\right]$ where $\boldsymbol{b} \neq \boldsymbol{b'}$ in

general. But when $\boldsymbol{b} = \mathbf{0}$, the constants part does not get affected by row operations. Since the solution of a system of linear equations is not affected by row operations, the solution sets for $\boldsymbol{A}$ and $\boldsymbol{R}$ are the same.

Next, we can see that in $\boldsymbol{R}$, the pivot columns are linearly independent of each other by design. They are the only ones with a nonzero element (actually 1) in the pivot positions, and there is no way we can create a 1 by taking finite combinations of 0. Therefore, a linear combination of the pivot columns in $\boldsymbol{R}$ will be $\mathbf{0}$ if and only if the coefficients multiplying them are all zero. Since the solution sets for $\boldsymbol{A}$ and $\boldsymbol{R}$ are the same, the same statement applies to $\boldsymbol{A}$ as well. Therefore, the columns in $\boldsymbol{A}$ that correspond to the pivot columns in $\boldsymbol{R}$ are linearly independent. And indeed, they form a basis for the column space of $\boldsymbol{A}$.

9.5.3 Null Spaces

The null space of a matrix would be the complete solution of the homogeneous system of equations $\boldsymbol{Ax} = \mathbf{0}$. When we write down the augmented matrix, it becomes $\left[\boldsymbol{A} \mid \mathbf{0}\right]$, and whatever row operations we perform on it will not change the constants or RHS part because it is $\mathbf{0}$. We can therefore solve $\boldsymbol{Ax} = \mathbf{0}$ by performing Gauss-Jordan elimination on the matrix $\boldsymbol{A}$ itself. If we find columns with no pivots, we will have free variables and null space.

To compute the left null space, we will follow the same procedure to find the complete solution set of the homogeneous system of linear equations $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{x} = \mathbf{0}$. After all, the left null space is $\mathcal{N}(\boldsymbol{A}^{\mathsf{T}})$, which is the null space of $\boldsymbol{A}^{\mathsf{T}}$.

9.6 Summary of the Four Spaces

We are now ready to summarize the four fundamental subspaces defined by a matrix. We will do it as a nice picture first, discuss it in the text here and then present it again in two tables. It is that important.

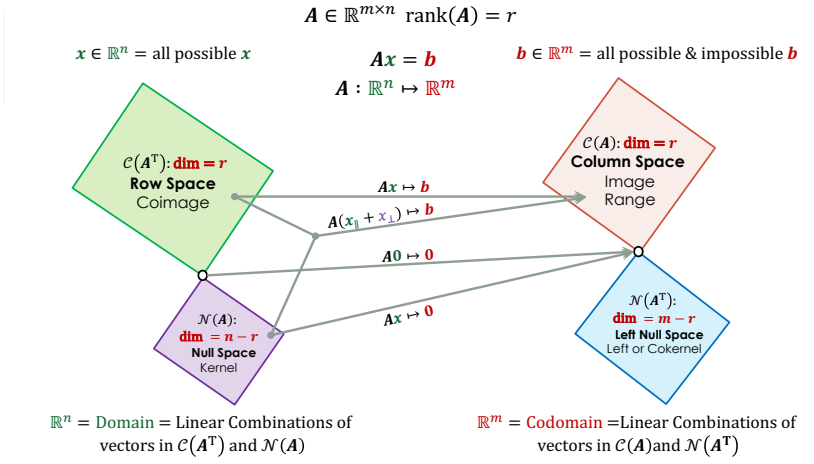


Fig. 9.1 A pictorial representation of the four fundamental subspaces defined by a matrix.

9.6.1 General Properties

In Figure 9.1, we have a pictorial representation of what the four fundamental subspaces mean in terms of the mapping $\boldsymbol{A} : \mathbb{R}^n \mapsto \mathbb{R}^m$ in the system $\boldsymbol{Ax} = \boldsymbol{b}$. They basically carve out different segments of the two vector spaces, $\mathbb{R}^n$ and $\mathbb{R}^m$, with specific properties.

The row space and null space are orthogonal complements. So are the column space and the left null space. All the vectors in the mull space ($\mathcal{N}(\boldsymbol{A})$) map to the zero vector $\mathbf{0}$. All other vectors in the row space get mapped to the column space $\mathcal{C}(\boldsymbol{A})$. All other vectors are linear combinations of the vectors in $\mathcal{N}(\boldsymbol{A})$ and $\mathcal{C}(\boldsymbol{A}^\mathbf{T})$. They also go to $\mathcal{C}(\boldsymbol{A})$, was we saw earlier. Since $\mathcal{C}(\boldsymbol{A}^\mathbf{T})$ and $\mathcal{N}(\boldsymbol{A})$ are orthogonal complements, we have nothing left in $\mathbb{R}^n$ to go to the left null space $\mathcal{N}(\boldsymbol{A}^\mathbf{T})$, other that the zero vector $\mathbf{0} \in \mathbb{R}^n$.

9.6.2 Matrix Shapes and the Spaces

As we saw, the geometric view of the fundamental spaces defined by a matrix are closely connected to its rank and the solvability of the system of linear equations it represents. It may be worth our time to look at the shapes of the matrix to see what its fundamental spaces look like. The four subspaces are closely related to the RREF of the matrix, and so are the solvability of the underlying system of linear

Table 9.1 Summary of the four fundamental spaces

Name, Space, Dim.	Basis	Description
Column Space $\mathcal{C}(\boldsymbol{A}) \subseteq \mathbb{R}^m$ $\dim(\mathcal{C}(\boldsymbol{A})) = r$	Pivot columns of $\boldsymbol{A}$	Range (Image) of $\boldsymbol{A} : \mathbb{R}^n \mapsto \mathbb{R}^m$ $\boldsymbol{b} \notin \mathcal{C}(\boldsymbol{A}) \implies$ no solution
Row Space $\mathcal{C}(\boldsymbol{A}^\mathsf{T}) \subseteq \mathbb{R}^n$ $\dim(\mathcal{C}(\boldsymbol{A}^\mathsf{T})) = r$	Pivot rows of $\boldsymbol{R}$ or $\boldsymbol{A}$	Coimage of $\boldsymbol{A} : \mathbb{R}^n \mapsto \mathbb{R}^m$ $\boldsymbol{x} \in \mathcal{C}(\boldsymbol{A}^\mathsf{T}) \implies \boldsymbol{x} \mapsto \boldsymbol{b} \neq \boldsymbol{0}$
Null Space (AKA Kernel) $\mathcal{N}(\boldsymbol{A}) \subseteq \mathbb{R}^n$ $\dim(\mathcal{N}(\boldsymbol{A})) = n - r$	Solution set of $\boldsymbol{Ax} = \boldsymbol{0}$ or $\boldsymbol{Rx} = \boldsymbol{0}$	$\boldsymbol{x} \in \mathcal{N}(\boldsymbol{A}) \implies \boldsymbol{x} \mapsto \boldsymbol{0}$ If $\boldsymbol{Ax} = \boldsymbol{b}$ and $\boldsymbol{x}_\perp \in \mathcal{N}(\boldsymbol{A})$, $\boldsymbol{A}(\boldsymbol{x} + \boldsymbol{x}_\perp) = \boldsymbol{b}$
Left Null Space $\mathcal{N}(\boldsymbol{A}^\mathsf{T}) \subseteq \mathbb{R}^m$ $\dim(\mathcal{N}(\boldsymbol{A}^\mathsf{T})) = m - r$	Solution set of $\boldsymbol{A}^\mathsf{T}\boldsymbol{x} = \boldsymbol{0}$ or $\boldsymbol{R}'\boldsymbol{x} = \boldsymbol{0}$	Unreachable, AKA Cokernel $\boldsymbol{b} \in \mathcal{N}(\boldsymbol{A}^\mathsf{T}) \implies \boldsymbol{Ax} \neq \boldsymbol{b}$

The table shows how the four fundamental subspaces of a matrix are related to the linear equations it represents. $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\text{rank}(\boldsymbol{A}) = r$ in $\boldsymbol{Ax} = \boldsymbol{b}$ with $\boldsymbol{A} \xrightarrow{\text{RREF}} \boldsymbol{R}$ and $\boldsymbol{A}^\mathsf{T} \xrightarrow{\text{RREF}} \boldsymbol{R}'$

equations. Keep in mind that while the REF (which is the result of Gaussian elimination) can have different shapes depending on the order in which the row operations are performed, RREF (the result of Gauss-Jordan) is immutable: A matrix has a unique RREF.

Table 9.2 The fundamental spaces of matrices of different shapes

Matrix Shape	RREF	Observations on the Four Spaces
Square, full-rank $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ $\text{rank}(\boldsymbol{A}) = n$	$\boldsymbol{I}_n$	$\mathcal{C}(\boldsymbol{A}), \mathcal{C}(\boldsymbol{A}^\mathsf{T}) = \mathbb{R}^n$ $\mathcal{N}(\boldsymbol{A}), \mathcal{N}(\boldsymbol{A}^\mathsf{T}) = \{\boldsymbol{0}\}$
"Tall," full-rank $\boldsymbol{A} \in \mathbb{R}^{m\times n},\ m > n$ $\text{rank}(\boldsymbol{A}) = n$	$\begin{bmatrix} \boldsymbol{I}_n \\ \hline \boldsymbol{0}_{(m-n)\times n} \end{bmatrix}$	$\mathcal{C}(\boldsymbol{A}) \subset \mathbb{R}^m, \dim(\mathcal{C}(\boldsymbol{A})) = n$ $\mathcal{C}(\boldsymbol{A}^\mathsf{T}) = \mathbb{R}^n$ $\mathcal{N}(\boldsymbol{A}) = \{\boldsymbol{0}\}$ $\mathcal{N}(\boldsymbol{A}^\mathsf{T}) \subset \mathbb{R}^m, \dim(\mathcal{N}(\boldsymbol{A}^\mathsf{T})) = m - n$
"Wide," full-rank $\boldsymbol{A} \in \mathbb{R}^{m\times n},\ m < n$ $\text{rank}(\boldsymbol{A}) = m$	$\begin{bmatrix} \boldsymbol{I}_m & \oplus & \boldsymbol{F}_{m\times(n-m)} \end{bmatrix}$	$\mathcal{C}(\boldsymbol{A}) = \mathbb{R}^m$ $\mathcal{C}(\boldsymbol{A}^\mathsf{T}) \subset \mathbb{R}^n, \dim(\mathcal{C}(\boldsymbol{A}^\mathsf{T})) = m$ $\mathcal{N}(\boldsymbol{A}) \subset \mathbb{R}^n, \dim(\mathcal{N}(\boldsymbol{A})) = n - m$ $\mathcal{N}(\boldsymbol{A}^\mathsf{T}) = \{\boldsymbol{0}\}$
Rank-deficient $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ $\text{rank}(\boldsymbol{A}) = r < r_m$ $r_m = \min(m, n)$	$\begin{bmatrix} \boldsymbol{I}_r \oplus \boldsymbol{F}_{r\times(n-r)} \\ \hline \boldsymbol{0}_{(m-r)\times n} \end{bmatrix}$	$\mathcal{C}(\boldsymbol{A}) \subset \mathbb{R}^m, \dim(\mathcal{C}(\boldsymbol{A})) = r$ $\mathcal{C}(\boldsymbol{A}^\mathsf{T}) \subset \mathbb{R}^n, \dim(\mathcal{C}(\boldsymbol{A}^\mathsf{T})) = r$ $\mathcal{N}(\boldsymbol{A}) \subset \mathbb{R}^n, \dim(\mathcal{N}(\boldsymbol{A}^\mathsf{T})) = n - r$ $\mathcal{N}(\boldsymbol{A}^\mathsf{T}) \subset \mathbb{R}^m, \dim(\mathcal{N}(\boldsymbol{A})) = m - r$

We have presented, in Table 9.2, a comparison of the various possible shapes and ranks of the matrices and the corresponding variations in their fundamental spaces. We should note the following points:

1. The column and row spaces always have the same dimension, and it is equal to the rank of the matrix. In fact, it is another definition for the rank.

$$\dim\left(\mathcal{C}(\boldsymbol{A})\right) = \dim\left(\mathcal{C}(\boldsymbol{A}^{\mathsf{T}})\right) = \text{rank}(\boldsymbol{A})$$

2. The dimension of the column space (which is the same as the rank) and the dimension of the null space (which is also called nullity) add up to the dimension of the domain. This is the famous rank-nullity theorem[2].

$$\text{For } \boldsymbol{A} \in \mathbb{R}^{m\times n} : \mathbb{R}^n \mapsto \mathbb{R}^m, \dim\left(\mathcal{C}(\boldsymbol{A})\right) + \dim\left(\mathcal{N}(\boldsymbol{A})\right) = n$$

9.7 Computing the Spaces

It is probably best to illustrate the computation of the four fundamental spaces of a matrix using an example. Let's compute all four fundamental subspaces of the following matrix:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 & 6 \\ 2 & 2 & 1 & 9 \\ 1 & 1 & 0 & 3 \end{bmatrix}$$

In fact, this matrix and an associated system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ are very similar to something we solved way back in Chapter 4, in Eqn (4.4), when we were still in Linear Algebra primary school. The difference now is that we have added one more row, and removed the RHS of the equations in $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$; we are working with just $\boldsymbol{A}$ now.

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 & 6 \\ 2 & 2 & 1 & 9 \\ 1 & 1 & 0 & 3 \end{bmatrix} \xrightarrow{\text{REF}} \begin{bmatrix} 1 & 1 & 1 & 6 \\ 0 & 0 & -1 & -3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{\text{RREF}} \begin{bmatrix} 1 & 1 & 0 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \boldsymbol{R}$$

[2]Wikipedia describes it as: "The rank-nullity theorem is a theorem in linear algebra, which asserts that the dimension of the domain of a linear map is the sum of its rank (the dimension of its image) and its nullity (the dimension of its kernel)."

Looking at $\boldsymbol{R}$, we can see that we have pivots in columns 1 and 3. The rank of the matrix is 2. Let's write down all the subspaces.

9.7.1 Column Space

The column space lives in $\mathbb{R}^3$ because the columns have three components. The dimension of the column space = the rank = the number of pivots = 2. The basis (which is a set) consists of the column vectors in $\boldsymbol{A}$ (*not* in $\boldsymbol{R}$) corresponding to the pivot columns, namely 1 and 3. So here is our answer:

$$\mathcal{C}(\boldsymbol{A}) \subset \mathbb{R}^3;\ \dim\left(\mathcal{C}(\boldsymbol{A})\right) = 2;\ \text{Basis} = \left\{ \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \right\}$$

9.7.2 Row Space

For the row space, we can take the pivot *rows* of $\boldsymbol{R}$ as our basis. Note that the row space lives in $\mathbb{R}^4$ because the number of columns of $\boldsymbol{A}$ is 4. It also has a dimension of 2, same as the rank.

$$\mathcal{C}(\boldsymbol{A}^{\mathsf{T}}) \subset \mathbb{R}^4;\ \dim\left(\mathcal{C}(\boldsymbol{A}^{\mathsf{T}})\right) = 2;\ \text{Basis} = \left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 3 \end{bmatrix} \right\}$$

Notice how we have been careful to write the basis as a set of column vectors, although we are talking about the row space. Our vectors are always columns.

9.7.3 Null Space

For the null space, we will first solve the underlying $\boldsymbol{Ax} = \boldsymbol{0}$ equations completely, highlight a pattern, and present it as a possible shortcut, to be used with care.

We saw earlier that $\boldsymbol{Ax} = \boldsymbol{0}$ and $\boldsymbol{Rx} = \boldsymbol{0}$ have the same solution set, which is the null space. Writing down the equations explicitly,

we have

$$\boldsymbol{Rx} = \begin{bmatrix} 1 & 1 & 0 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \mathbf{0} \tag{9.2}$$

$$\implies x_1 + x_2 + 3x_4 = 0 \text{ and } x_3 + 3x_4 = 0$$

Noting that x_2 and x_4 are free variables (because the corresponding columns have no pivots), we solve for x_1 and x_3 as $x_3 = -3x_4$ and $x_1 = -x_2 - 3x_4$. Therefore the complete solution becomes:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -x_2 - 3x_4 \\ x_2 \\ -3x_4 \\ x_4 \end{bmatrix} = x_2 \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + x_4 \begin{bmatrix} -3 \\ 0 \\ -3 \\ 1 \end{bmatrix}$$

The complete solution is a linear combination of the two vectors above because x_2 and x_4, being free variables, they can take any value in $\mathbb{R}$. In other words, the complete solution is the span of these two vectors, which is what the null space is. We give our computation of the null space as follows:

$$\mathcal{N}(\boldsymbol{A}) \subset \mathbb{R}^4;\ \dim\left(\mathcal{N}(\boldsymbol{A})\right) = 2;\ \text{Basis} = \left\{ \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -3 \\ 0 \\ -3 \\ 1 \end{bmatrix} \right\}$$

9.7.4 Left Null Space

The steps to compute $\mathcal{N}(\boldsymbol{A}^\mathbf{T})$ are identical to the ones for $\mathcal{N}(\boldsymbol{A})$, except that we start with $\boldsymbol{A}^\mathbf{T}$ instead of $\boldsymbol{A}$, naturally. We will not go through them here, but, as promised earlier, we will share a shortcut for computing null spaces in the box on "**Null Spaces: A Shortcut**," and use the left-null-space computation of this $\boldsymbol{A}$ as an example.

From the examples worked out, we can see that the null-space computations and the complete solutions of the underlying system of linear equations have a lot in common. It is now time to put these two topics on an equal footing, which also gives us an opportunity to review the process of completely solving a system of linear equations and present it as a step-by-step algorithm.

Null Spaces: A Shortcut

Once we have the RREF of a matrix, we can write down the null space by looking at it. Here are the steps:

1. Identify the free-variable columns (which are the ones without pivots).
2. Copy these columns with a negative sign for the entries, while leaving a blank space for the free variables. Ignore zero rows at the bottom of the RREF.
3. Type in 1 for the free variable and zero for all others in the vectors created in (2). This is our basis.
4. The dimension of the containing space is the number of variables. The dimension of the null space is the number of basis vectors in (3)

Illustrating it with the left-null-space computation of the example in the text:

$$\boldsymbol{A}^{\mathsf{T}} = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \\ 6 & 9 & 3 \end{bmatrix}; \quad \text{RREF}(\boldsymbol{A}^{\mathsf{T}}) = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Using our shortcut,

1. The only free variable is the third one (call it x_3).
2. Copying it, with the negative sign and a blank ◯ in the third position (because our free variable is x_3), we get:

$$\begin{bmatrix} 1 \\ -1 \\ \bigcirc \end{bmatrix}$$

3. In the blank position, indicated by ◯, of the first basis vector created in (2) we type in 1.
4. The final answer is:

$$\mathcal{N}(\boldsymbol{A}^{\mathsf{T}}) \subset \mathbb{R}^3; \ \dim \mathcal{N}(\boldsymbol{A}^{\mathsf{T}}) = 1; \ \text{Basis} = \left\{ \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \right\}$$

A bit of thinking should convince us that this shortcut is, in fact, the same as what we did in computing the null space in the text, by writing down the complete solution. We can easily verify that we get the same answer by applying this shortcut on the matrix $\boldsymbol{R}$ in Eqn (9.2).

9.8 Review: Complete Solution

The computation of the null space of a matrix is, in fact, the same as finding the *special* solutions of the underlying system of linear equations. When we found the *complete* solution earlier in §4.3.2 (page 77) and when we computed the null space above, the procedures

may have looked ad-hoc. Now that we know all there is to know about the fundamental spaces of a matrix, it is time to finally put to rest the tentativeness in the solving procedure and present it like an algorithm with unambiguous steps. Let's start by defining the terms.

In $\boldsymbol{Ax} = \boldsymbol{b}$, the *complete* solution is the sum of the *particular* solution and the *special* solutions. We can see an example in Eqn (4.4), where the first vector is the particular solution and the second and third terms making a linear combination of two vectors is the special solution. The linear combination is, in fact, the null space of $\boldsymbol{A}$. As we saw earlier, if $\boldsymbol{x}_\parallel$ is a solution to $\boldsymbol{Ax} = \boldsymbol{b}$, then $\boldsymbol{x}_\parallel + \boldsymbol{x}_\perp$ also is a solution for any $\boldsymbol{x}_\perp \in \mathcal{N}(\boldsymbol{A})$ because, as we saw earlier,

$$\boldsymbol{Ax} = \boldsymbol{A}(\boldsymbol{x}_\parallel + \boldsymbol{x}_\perp) = \boldsymbol{Ax}_\parallel + \boldsymbol{Ax}_\perp = \boldsymbol{Ax}_\parallel + \boldsymbol{0} = \boldsymbol{Ax}_\parallel$$

$\boldsymbol{x}_\parallel$ is a particular solution (which may be called $\boldsymbol{x}_p$) and the set of $\boldsymbol{x}_\perp$ (AKA $\boldsymbol{x}_s$) is the special solution.

Since the special solution, the set of $\boldsymbol{x}_\perp$, is a subspace, we can select any full set of linearly independent vectors as the basis to specify it. Let's work through an example to illustrate whatever we stated so far, before listing the algorithm for completely solving a system of linear equations.

9.8.1 An Example

We will reuse the example in Eqn (4.4) (on page 78), where we started with these equations:

$$\begin{aligned} x_1 + x_2 + x_3 + 2x_4 &= 6 \\ 2x_1 + 2x_2 + x_3 + 7x_4 &= 9 \end{aligned}$$

from which we got the augmented matrix:

$$\left[\boldsymbol{A} \mid \boldsymbol{b}\right] = \left[\begin{array}{cccc|c} 1 & 1 & 1 & 2 & 6 \\ 2 & 2 & 1 & 7 & 9 \end{array}\right] \xrightarrow{\text{REF}} \left[\begin{array}{cccc|c} 1 & 1 & 1 & 2 & 6 \\ 0 & 0 & -1 & 3 & -3 \end{array}\right]$$

and ended up with the complete solution:

$$\boldsymbol{x} = \begin{bmatrix} 3 \\ 0 \\ 3 \\ 0 \end{bmatrix} + t_1 \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + t_2 \begin{bmatrix} -5 \\ 0 \\ 3 \\ 1 \end{bmatrix} \tag{9.3}$$

The null space of the coefficient matrix has the basis vectors appearing as the linear combination in the last two terms above. $\mathcal{N}(\boldsymbol{A})$ is a plane in $\mathbb{R}^4$, and we can use any two linearly independent vectors on it as its basis. The exact basis vectors we wind up with depend on the actual elimination steps we use, but they all specify the same plane of vectors, and indeed the same subspace. As we can see, in solving the system of equations above, we started with the row-echelon form of the augmented matrix $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$. The REF (the output of Gaussian elimniation) of a matrix is not unique; it is the RREF (from Gauss-Jordan elimination) that is unique.

Let's solve the system again. This time, we will start by finding the RREF (the output of Gauss-Jordan) because it is unique for any given matrix.

$$\left[\boldsymbol{A} \mid \boldsymbol{b}\right] = \left[\begin{array}{cccc|c} 1 & 1 & 1 & 2 & 6 \\ 2 & 2 & 1 & 7 & 9 \end{array}\right] \xrightarrow{\text{RREF}} \left[\begin{array}{cccc|c} 1 & 1 & 0 & 5 & 3 \\ 0 & 0 & 1 & -3 & 3 \end{array}\right]$$

In order to find a particular solution, we first set values of the free variables to zero, knowing that they are free to take any values. This step, in effect, ignores the free variables for the moment. In the example above, the free variables are x_2 and x_4, corresponding to the columns with no pivots. Ignoring these pivot-less columns, what we see is an augmented matrix $\left[\boldsymbol{A} \mid \boldsymbol{b}\right] = \left[\boldsymbol{I} \mid \boldsymbol{b}'\right]$, giving us the particular solution[3] as below:

$$\left[\begin{array}{cc|c} 1 & 0 & 3 \\ 0 & 1 & 3 \end{array}\right] \implies \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \implies \boldsymbol{x}_p = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 3 \\ 0 \end{bmatrix}$$

The special solutions are from the null space, and we have to solve the homogeneous system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$, for which the augmented matrix is $\left[\boldsymbol{A} \mid \boldsymbol{0}\right]$. What we do, in practice, is to ignore the $\boldsymbol{b}$ part of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, or set it to zero, and work with the coefficient part. We then cyclically set one free variable to one, and the rest to zero in the equations specified by RREF($\left[\boldsymbol{A} \mid \boldsymbol{0}\right]$):

$$\left[\begin{array}{cccc|c} 1 & 1 & 0 & 5 & 0 \\ 0 & 0 & 1 & -3 & 0 \end{array}\right] \implies \begin{aligned} x_1 + x_2 + 5x_4 &= 0 \\ x_3 - 3x_4 &= 0 \end{aligned}$$

[3]Note that the particular solution obtained using this prescription does not have to be in the row space of the coefficient matrix because we are taking zero values for the free variables.

Setting $x_2 = 1$ and $x_4 = 0$:

$$\begin{array}{rcl} x_1 + x_2 + 5x_4 &= 0 \\ x_3 - 3x_4 &= 0 \end{array} \implies \begin{array}{rcl} x_1 + 1 &= 0 \\ x_3 &= 0 \end{array} \implies \boldsymbol{x}_{\perp 1} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Now, setting $x_2 = 0$ and $x_4 = 1$:

$$\begin{array}{rcl} x_1 + x_2 + 5x_4 &= 0 \\ x_3 - 3x_4 &= 0 \end{array} \implies \begin{array}{rcl} x_1 + 5 &= 0 \\ x_3 - 3 &= 0 \end{array} \implies \boldsymbol{x}_{\perp 2} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -5 \\ 0 \\ 3 \\ 1 \end{bmatrix}$$

Putting it all together, we can write down the complete solution as:

$$\boldsymbol{x} = \begin{bmatrix} 3 \\ 0 \\ 3 \\ 0 \end{bmatrix} + t_1 \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + t_2 \begin{bmatrix} -5 \\ 0 \\ 3 \\ 1 \end{bmatrix} \tag{9.4}$$

Comparing Eqn (9.4) above to Eqn (9.3), we can see that we got identical solutions in our example. However, in principle, the vectors in the linear combination in the last two terms may differ. Our assertion is that what we may have in Eqn (9.4), if they were different from the ones in Eqn (9.3), would merely be another pair of linearly independent vectors in the same planar subspace specified by the latter.

How do we make sure though? We know enough Linear Algebra to answer this question: If we were to place these four vectors in a matrix, we would have only two linearly independent columns, and the rank of the matrix would be two. How do we know the rank? We perform row-reduction on it and count the number of pivots.

Squeezing out the last teachable moment from this review of completely solving linear systems, the rank is also the trace of the RREF of the coefficient matrix because the pivots are all normalized to one.

9.8.2 The Algorithm

Here is a description of what we did in the example above, but using the most general case. We start with $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\text{rank}(\boldsymbol{A}) = r \leq \min(m, n)$. We know the shape of the canonical

form, RREF:

$$\boldsymbol{A} \xrightarrow{\text{RREF}} \boldsymbol{R} = \left[\begin{array}{c|c} \boldsymbol{I}_r \oplus \boldsymbol{F}_{r\times(n-r)} & \boldsymbol{b'} \\ \hline \boldsymbol{0}_{(m-r)\times n} & \end{array}\right]$$

Note that we use the symbol $\oplus$ to indicate that the columns of $\boldsymbol{I}_r$ and $\boldsymbol{F}_{r\times(n-r)}$ may be shuffled in; we may not have the columns of $\boldsymbol{F}$ neatly to the right of $\boldsymbol{I}$. With this picture in mind, let's describe the algorithm for completely solving the system of linear equations:

1. Find RREF through Gauss-Jordan on the augmented matrix $\left[\boldsymbol{A} \mid \boldsymbol{b}\right] \to \boldsymbol{R}$
2. Ignore the free variables by setting them to zero, which is the same as deleting the pivot-less columns in $\boldsymbol{R}$ and zero rows, giving us $\boldsymbol{I}_r$
3. Get the particular solution, $\boldsymbol{x}_{\parallel}$ with the r values in $\boldsymbol{b'}$, and zeros for the $n-r$ free variables
4. For each free variable, with the RREF of the homogeneous augmented matrix $\left[\boldsymbol{A} \mid \boldsymbol{0}\right]$ (which is the same $\boldsymbol{R}$ as above, but with $\boldsymbol{0}$ instead of $\boldsymbol{b'}$ in the augmenting column):
   - Set its value to one, and the values of all others to zero
   - Solve the resulting equations to get one special solution, $\boldsymbol{x}_{\perp i}$
   - Iterate over all free variables
5. Write down the complete solution:

$$\boldsymbol{x} = \boldsymbol{x}_{\parallel} + \sum_{i=1}^{n-r} t_i \boldsymbol{x}_{\perp i}$$

6. Know that we have computed the null space as well:

$$\mathcal{N}(\boldsymbol{A}) \subset \mathbb{R}^n, \quad \dim\left(\mathcal{N}(\boldsymbol{A})\right) = n - r, \quad \text{Basis} = \{\boldsymbol{x}_{\perp i}\}$$

9.9 Other Names

In order to see why these fundamental subspaces go by their aliases, we have to look at functions again. In normal algebra, when we

have a function $f(x) : \mathbb{R} \mapsto \mathbb{R}$, the totality of the possible values of x (which would be all of $\mathbb{R}$ since x is the so-called independent variable) is called the *domain* and that of $y = f(x)$ is called the *codomain*. Not all of the codomain may be reached though. For instance, if $f(x) = x^2$, the codomain is $\mathbb{R}$, but the part of the codomain that is reachable is $y > 0$. This part is called the *range*. Some people call the range the *image* (because it is what x is reflected into, if we were to guess). Not to be partial to y, they also call the part of the domain that is reflected the *coimage*.

Thinking of $\boldsymbol{A}$ as a mapping or function $\boldsymbol{A} : \mathbb{R}^n \mapsto \mathbb{R}^m$, the domain is $\mathbb{R}^n$ and the codomain is $\mathbb{R}^m$. The column space may be called the range or image, and the row space is the coimage, although we will stay away from such abominations. Not to be outdone, the null spaces also go by names such as kernel and right kernel. This naming is a bit more mysterious, but everything in the kernel gets mapped to $\mathbf{0}$, and nothing in the domain $\mathbb{R}^n$ ever gets mapped to the left kernel.

Regardless of what names they go by, the four fundamental subspaces bring together almost everything we learned so far. They form the basis of the advanced topics in Linear Algebra and are therefore critical for furthering our understanding.

10

Projection, Least Squares and Linear Regression

It was one thing to use computers as a tool, quite another to let them do your thinking for you.

—Tom Clancy

Now that we learned the four fundamental subspaces defined by a matrix, we can look at one of its most widely used applications in machine learning, namely linear regression. For this, we will be expanding on our notion of projection of one vector onto others. We will be projecting to subspaces instead.

10.1 Projection Revisited

As a preparation for orthonormalization and the Gram-Schmidt process, we looked at projection earlier in §7.4.3 (page 135). Our approach was to fall back on the trigonometric definition of the dot product using the cosine of the angle between the vectors. This approach was perhaps ill-advised for we are not learning trigonometry

here, but the much superior Linear Algebra. The right approach to be taken, as we shall see here, is much more elegant.

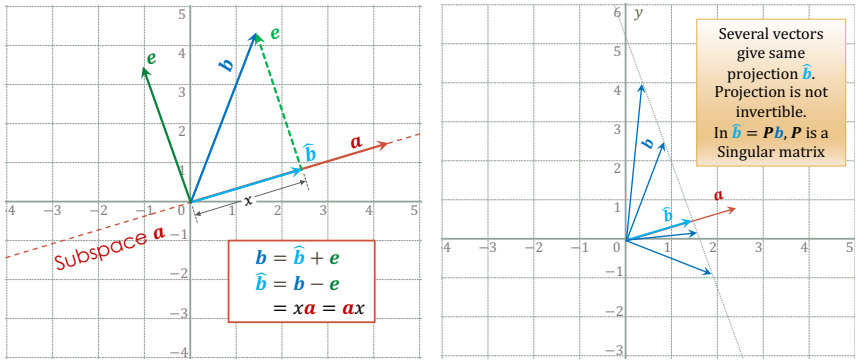


Fig. 10.1 Projection of one vector ($\boldsymbol{b}$ in blue) onto another ($\boldsymbol{a}$ in red). The projection $\hat{\boldsymbol{b}}$ is shown in light blue, and the error vector in green. The right panel shows that many different vectors (blue ones) can all have the same projection. The projection operation is many-to-one, and cannot be inverted.

In Figure 10.1, in the left panel, we have a blue vector $\boldsymbol{b}$ being projected onto the red $\boldsymbol{a}$. The projection is another vector, $\hat{\boldsymbol{b}}$ in a brighter shade of blue. Of all the vectors along $\boldsymbol{a}$, why is this the right vector to call the projection? One way to look it is to imagine that we are shining some light from the top, perpendicular to $\boldsymbol{a}$, in the plane containing $\boldsymbol{a}$ and $\boldsymbol{b}$. The projection then is the shadow cast by the blue vector on the red one. Another way is to think of the difference vector, shown in green as $\boldsymbol{e}$, to be a minimum. The projection is that vector along the direction $\boldsymbol{a}$ which is closest to $\boldsymbol{b}$, where closeness is defined as the distance between the tips of $\boldsymbol{b}$ and its projection $\hat{\boldsymbol{b}}$, which is the same as the norm $\|\boldsymbol{e}\|$. We can call this green vector the error (hence the name $\boldsymbol{e}$), and we are trying to minimize this error.

And when is $\boldsymbol{e}$ the smallest? It is when $\boldsymbol{a}$ and $\boldsymbol{e}$ are orthogonal to each other, or when $\boldsymbol{a} \perp \boldsymbol{e}$. We know the condition for orthogonality: The dot product $\boldsymbol{a}^\mathsf{T}\boldsymbol{e} = 0$. Since $\hat{\boldsymbol{b}}$ lies along the direction of $\boldsymbol{a}$, we know that it has to be a scalar multiple: $\hat{\boldsymbol{b}} = \boldsymbol{a}x$, where we called the scalar x (and put it after $\boldsymbol{a}$ for our own dark purposes). For each value of x, we will get a different candidate for the projection. The right candidate, the one that minimizes the error, is $\hat{\boldsymbol{b}}$, for which $x = \hat{x}$. Now the problem of computing the projection boils down to

computing $\hat{x}$.

$$\begin{aligned} \boldsymbol{a}^\mathsf{T}\boldsymbol{e} = \boldsymbol{a}^\mathsf{T}(\boldsymbol{b}-\hat{\boldsymbol{b}}) = \boldsymbol{a}^\mathsf{T}(\boldsymbol{b}-\boldsymbol{a}\hat{x}) = 0 \\ \boldsymbol{a}^\mathsf{T}\boldsymbol{a}\hat{x} = \boldsymbol{a}^\mathsf{T}\boldsymbol{b} \implies \hat{x} = \frac{\boldsymbol{a}^\mathsf{T}\boldsymbol{b}}{\boldsymbol{a}^\mathsf{T}\boldsymbol{a}} \\ \hat{\boldsymbol{b}} = \boldsymbol{a}\hat{x} = \boldsymbol{a}\frac{\boldsymbol{a}^\mathsf{T}\boldsymbol{b}}{\boldsymbol{a}^\mathsf{T}\boldsymbol{a}} = \boldsymbol{a}(\boldsymbol{a}^\mathsf{T}\boldsymbol{a})^{-1}\boldsymbol{a}^\mathsf{T}\boldsymbol{b} \\ \boldsymbol{P} = \boldsymbol{a}(\boldsymbol{a}^\mathsf{T}\boldsymbol{a})^{-1}\boldsymbol{a}^\mathsf{T} \end{aligned} \tag{10.1}$$

Here, in addition to computing the projection value $\hat{x}$, the projection vector $\hat{\boldsymbol{b}}$, we have also defined a projection matrix $\boldsymbol{P}$ which we can multiply with any vector and gets its projection onto $\boldsymbol{a}$. It is indeed a matrix because $\boldsymbol{a}\boldsymbol{a}^\mathsf{T}$ is of a column matrix $(n \times 1)$ multiplying a row matrix $(1 \times n)$, giving as an $n \times n$ matrix. The factor $(\boldsymbol{a}^\mathsf{T}\boldsymbol{a})^{-1}$ in between is just a scalar, which does not change the shape.

Comparing the derivation of the projection matrix above in Eqn (10.1) to the one we did earlier in Eqn (7.9), we can appreciate that they are identical. It is just that the Linear Algebra way is much more elegant. In this derivation, we took certain facts to be self-evident, such as when two vectors are orthogonal, their dot product is zero. We can indeed prove it, as shown in the box on "**Orthogonality and Dot Product**."

10.1.1 Why Project?

What we did was to project a vector $\boldsymbol{b}$ onto another one $\boldsymbol{a}$, which is to say we found the vector closest to $\boldsymbol{b}$ in the subspace spanned by $\boldsymbol{a}$. We now want to expand on this notion on projecting to a subspace, but before that, we may want to ask ourselves why we want to project to subspaces.

The motivation, as with most things in Linear Algebra, is connected to the solvability of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. Let's take the situation where we have too many equations, which means our coefficient matrix, $\boldsymbol{A}$ is "tall." We are particularly interested in such matrices because our datasets in computer science tends to be of that shape.

In general, "tall" data matrices with real data tend to be full rank because data points, which make up the rows of such matrices, tend to be linearly independent due to measurement errors and statistical fluctuations. The RREF of a "tall" full-rank matrix is $\boldsymbol{I}$ near the top,

and zero rows below it. Such matrices, as coefficient matrices in $\boldsymbol{Ax} = \boldsymbol{b}$, have solutions only if the RHS, $\boldsymbol{b}$ is in the column space of $\boldsymbol{A}$, which in general, it will not be.

What do we do in such a situation? We still want to solve the system even when $\boldsymbol{b} \notin \mathcal{C}(\boldsymbol{A})$. In this case, the best we can do is to find the vector $\hat{\boldsymbol{b}}$ in the column space that is closest to $\boldsymbol{b}$, which will be its projection onto $\mathcal{C}(\boldsymbol{A})$.

10.2 Projection to Subspace

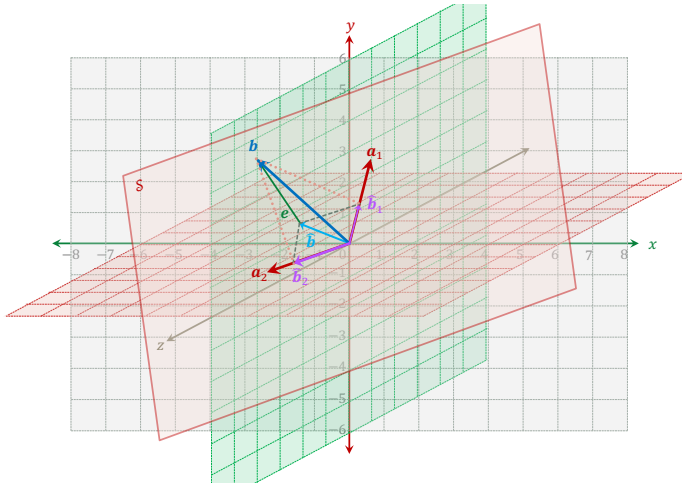


Fig. 10.2 Projecting the blue $\boldsymbol{b}$ onto the subspace that is the span of $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$, the two red vectors. The subspace is a plane, shown in light red. The projection, $\hat{\boldsymbol{b}}$ (in bright blue) of $\boldsymbol{b}$ is in the subspace, and is therefore a linear combination of its basis vectors, $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$, shown in red.

In order to project to a subspace, we need to specify what the subspace is, which we do in Linear Algebra by specifying the basis. If our basis vectors are $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$, the subspace onto which we are projecting is a plane, as shown in Figure 10.2. Exactly as we did in Eqn (10.1), we want the formulas for $\hat{x}$, the length of the project, or $\hat{\boldsymbol{b}}$, the projected vector and $\boldsymbol{P}$, the projection matrix. In the case of a plane with two basis vectors, we will have two lengths $\hat{x}_1$ and $\hat{x}_2$.

Since our projected vector $\hat{\boldsymbol{b}}$ is on the plane spanned by the basis vectors, it is a linear combinations of them, and the coefficients of the linear combination are what we call $\hat{x}_1$ and $\hat{x}_2$. Referring to

Figure 10.2, where we are projecting the blue $\boldsymbol{b}$ onto the red plane which is the span of $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$, we have the projections along each basis vector, $\hat{\boldsymbol{b}}_{\boldsymbol{i}} = \boldsymbol{a}_i \hat{x}_i$. Arranging the basis vectors as the columns of a matrix, we can write our linear combination for $\hat{\boldsymbol{b}}$ as:

$$\boldsymbol{A} = \begin{bmatrix} | & | \\ \boldsymbol{a_1} & \boldsymbol{a_2} \\ | & | \end{bmatrix}, \; \hat{\boldsymbol{b}} = \boldsymbol{A} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = \boldsymbol{A}\hat{\boldsymbol{x}} = \hat{\boldsymbol{b}}$$

which recovers the form of our favorite equation $\boldsymbol{Ax} = \boldsymbol{b}$. All we need to do is to find $\hat{\boldsymbol{x}}$ for which we have the fact that the green error vector $\boldsymbol{e} = \boldsymbol{b} - \hat{\boldsymbol{b}}$ is orthogonal to the red plane in Figure 10.2. When a vector is perpendicular to a subspace, it is perpendicular to every vector in it. In particular, $\boldsymbol{e}$ is orthogonal to every basis vector $\boldsymbol{a}_i$, which means we have $\boldsymbol{a}_i^{\mathsf{T}} \boldsymbol{e} = 0$. We can again write these two equations (for $i = 1$ and 2) as a matrix equation: $\boldsymbol{A}^{\mathsf{T}} \boldsymbol{e} = \boldsymbol{0}$.

$$\boldsymbol{A}^{\mathsf{T}} \boldsymbol{e} = \boldsymbol{0} \implies \boldsymbol{A}^{\mathsf{T}} (\boldsymbol{b} - \hat{\boldsymbol{b}}) = \boldsymbol{0} \implies \boldsymbol{A}^{\mathsf{T}} \boldsymbol{b} = \boldsymbol{A}^{\mathsf{T}} \hat{\boldsymbol{b}} = \boldsymbol{A}^{\mathsf{T}} \boldsymbol{A} \hat{\boldsymbol{x}}$$

Recasting the last part of the previous equation in the form $\boldsymbol{Ax} = \boldsymbol{b}$, we write our equation which will solve all our problems from now on:

$$\boldsymbol{A}^{\mathsf{T}} \boldsymbol{A} \hat{\boldsymbol{x}} = \boldsymbol{A}^{\mathsf{T}} \boldsymbol{b} \implies \hat{\boldsymbol{x}} = \left(\boldsymbol{A}^{\mathsf{T}} \boldsymbol{A} \right)^{-1} \boldsymbol{A}^{\mathsf{T}} \boldsymbol{b} \tag{10.2}$$

What this equation tells us is that even if $\boldsymbol{Ax} = \boldsymbol{b}$ does not have a solution (because $\boldsymbol{A}$ is full-column-rank with inconsistent RHS), multiplying both sides of it with $\boldsymbol{A}^{\mathsf{T}}$ magically makes it solvable, at least in an approximate sense.

This approximation that gives us the best possible solution $\hat{\boldsymbol{x}}$ is, in fact, identical to the least square minimization of the sum of squared errors, if we were to do it in the old, calculus way. A numerical, iterative minimization of the calculus kind may or may not converge, may not converge to the global minimum, and may have computational complexity issues. In the elegant world of Linear Algebra, it is only a couple of matrix multiplications and an inversion.

Using our recently acquired knowledge of the four fundamental space of $\boldsymbol{A}$, we can immediately see that $\boldsymbol{e} \in \mathcal{N}(\boldsymbol{A}^{\mathsf{T}})$. And, $\hat{\boldsymbol{b}} \in \mathcal{C}(\boldsymbol{A})$ because it is a linear combination of the columns of $\boldsymbol{A}$. $\boldsymbol{A}$ is guaranteed to be full-rank because its columns span a subspace, and $\boldsymbol{A}\hat{\boldsymbol{x}} = \hat{\boldsymbol{b}}$ is guaranteed to have a unique solution.

The Hat Matrix

As we saw, the projection matrix $\boldsymbol{P}$ takes $\boldsymbol{b}$ to its estimated, "hatted" counterpart, $\hat{\boldsymbol{b}}$. For this reason, some textbooks and articles, especially the ones on statistics, when dealing with linear regression, call the projection matrix $\boldsymbol{P}$ the hat matrix $\boldsymbol{H}$.

While deriving the projection matrix, we also looked the error vector $\boldsymbol{e} = \boldsymbol{b} - \hat{\boldsymbol{b}}$ when projecting $\boldsymbol{b}$ onto the column space, $\mathcal{C}(\boldsymbol{A})$ where $\hat{\boldsymbol{b}} = \boldsymbol{P}\boldsymbol{b}$ was the projection. We can then write

$$\boldsymbol{e} = \boldsymbol{b} - \hat{\boldsymbol{b}} = \boldsymbol{b} - \boldsymbol{P}\boldsymbol{b} = (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{b}$$

Some statisticians consider this a second projection matrix, $\boldsymbol{P_2} = \boldsymbol{I} - \boldsymbol{P}$. Remembering that $\boldsymbol{e} \in \mathcal{N}(\boldsymbol{A}^\intercal)$, $\boldsymbol{P_2}$ is the operator that will project a vector to the left null space of $\boldsymbol{A}$.

The last thing we need to do in this section is to write down the projection matrix $\boldsymbol{P}$ such that $\boldsymbol{P}\boldsymbol{b} = \hat{\boldsymbol{b}}$. Knowing that $\boldsymbol{A}\hat{\boldsymbol{x}} = \hat{\boldsymbol{b}}$ and using the formula for $\hat{\boldsymbol{x}}$ from Eqn (10.2), the projection matrix is:

$$\boldsymbol{P} = \boldsymbol{A}\left(\boldsymbol{A}^\intercal\boldsymbol{A}\right)^{-1}\boldsymbol{A}^\intercal \tag{10.3}$$

10.2.1 Properties of Projection Matrix

The first property of the projection matrix is that applying it multiple times on a vector is the same as applying it once. It makes sense from the meaning of projection—once we project a vector onto another, the projection is already collinear with it. Projecting a vector collinear with another one onto the latter does not do anything. This property is called idempotence and the projection matrix is idempotent, which we can easily verify:

$$\begin{aligned}
\boldsymbol{P} = \boldsymbol{A}\left(\boldsymbol{A}^\intercal\boldsymbol{A}\right)^{-1}\boldsymbol{A}^\intercal \implies \boldsymbol{P}^2 &= \boldsymbol{A}\left(\boldsymbol{A}^\intercal\boldsymbol{A}\right)^{-1}\boldsymbol{A}^\intercal\,\boldsymbol{A}\left(\boldsymbol{A}^\intercal\boldsymbol{A}\right)^{-1}\boldsymbol{A}^\intercal \\
&= \boldsymbol{A}\left(\left(\boldsymbol{A}^\intercal\boldsymbol{A}\right)^{-1}\boldsymbol{A}^\intercal\boldsymbol{A}\right)\left(\boldsymbol{A}^\intercal\boldsymbol{A}\right)^{-1}\boldsymbol{A}^\intercal \\
&= \boldsymbol{A}\boldsymbol{I}\left(\boldsymbol{A}^\intercal\boldsymbol{A}\right)^{-1}\boldsymbol{A}^\intercal = \boldsymbol{A}\left(\boldsymbol{A}^\intercal\boldsymbol{A}\right)^{-1}\boldsymbol{A}^\intercal \\
&= \boldsymbol{P}
\end{aligned}$$

Another property is that $\boldsymbol{P}$ has to be symmetric. This property comes from the fact that dot product of two vectors is the same as the dot product of one with the projection of the other onto it. Referring

to Figure 10.1, it means:

$$\boldsymbol{a}^{\mathsf{T}}\boldsymbol{b} = \boldsymbol{a}^{\mathsf{T}}\hat{\boldsymbol{b}} = \boldsymbol{a}^{\mathsf{T}}\boldsymbol{P}\boldsymbol{b}$$

On the other hand, $\boldsymbol{a}$ projected onto itself is, of course, $\boldsymbol{a}$. Therefore,

$$\boldsymbol{a}^{\mathsf{T}}\boldsymbol{b} = (\boldsymbol{P}\boldsymbol{a})^{\mathsf{T}}\hat{\boldsymbol{b}} = \boldsymbol{a}^{\mathsf{T}}\boldsymbol{P}^{\mathsf{T}}\boldsymbol{b}$$

With the two expressions for $\boldsymbol{a}^{\mathsf{T}}\boldsymbol{b}$, we can equate them and write,

$$\boldsymbol{a}^{\mathsf{T}}\boldsymbol{P}\boldsymbol{b} = \boldsymbol{a}^{\mathsf{T}}\boldsymbol{P}^{\mathsf{T}}\boldsymbol{b} \implies \boldsymbol{P} = \boldsymbol{P}^{\mathsf{T}}$$

Let's verify if our $\boldsymbol{P}$ in Eqn (10.3) is indeed symmetric as it should be.

$$\begin{aligned}
\boldsymbol{P}^{\mathsf{T}} &= \left(\boldsymbol{A}\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}}\right)^{\mathsf{T}} \\
(1) \qquad &= \left(\boldsymbol{A}^{\mathsf{T}}\right)^{\mathsf{T}}\left(\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\right)^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}} \\
(2) \qquad &= \boldsymbol{A}\left(\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{\mathsf{T}}\right)^{-1}\boldsymbol{A}^{\mathsf{T}} \\
(3) \qquad &= \boldsymbol{A}\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}} \\
&= \boldsymbol{P}
\end{aligned} \tag{10.4}$$

where we used (1) the product rule of transposes, (2) the fact that the inverse of a transpose is the transpose of the inverse and (3) the symmetry of $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}$.

10.3 Meaning of Projection

The projection operation is a many-to-one mapping, as shown in Figure 10.1, on its right panel. Since many vectors like $\boldsymbol{b}$ can be projected onto $\boldsymbol{a}$ to have the same $\hat{\boldsymbol{b}}$ as shown, given $\hat{\boldsymbol{b}}$, we just cannot figure out what $\boldsymbol{b}$ it came from. Such mappings are called injective and they cannot be inverted. In other words, the projection operation throws away some information.

In the world of Linear Algebra, the matrix corresponding to such operations would be a rank-deficient one. When projecting to a subspace defined by one vector, $\boldsymbol{P}$ is a rank-one matrix. For a subspace of dimension two in $\mathbb{R}^n$, it is a rank-two matrix. In general,

for any subspace, $\boldsymbol{P}$ is singular, which is the reason why we cannot apply the product rule for inverses to $(\boldsymbol{A}^\intercal \boldsymbol{A})^{-1}$.

$$\boldsymbol{P} = \boldsymbol{A}(\boldsymbol{A}^\intercal \boldsymbol{A})^{-1}\boldsymbol{A}^\intercal \neq \boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{A}^{\intercal -1}\boldsymbol{A}^\intercal = \boldsymbol{I}\boldsymbol{I} = \boldsymbol{I}$$

We should not do this expansion and the product rule does not apply here because $\boldsymbol{A}^{-1}$ is not defined, which was the whole point of embarking on this projection trip to begin with.

What happens if we take the full space and try to project onto it? In other words, we try to project a vector onto a "subspace" of dimension n in $\mathbb{R}^n$. In this case, we get a full-rank projection matrix, and the expansion of the inverse above is indeed valid, and the projection matrix really is $\boldsymbol{I}$, which is an invertible matrix because every vector gets "projected" onto itself. Note that the two properties we were looking for in $\boldsymbol{P}$ are satisfied by $\boldsymbol{I}$: It is idempotent because $\boldsymbol{I}^2 = \boldsymbol{I}$ and of course $\boldsymbol{I}^\intercal = \boldsymbol{I}$.

For rank-deficient matrices, $\boldsymbol{P}$ is $\boldsymbol{A}\boldsymbol{A}^{-1}_{\text{Left}}$, the left inverse multiplying on the right, almost like an attempt to get as close to $\boldsymbol{I}$ as possible.

10.4 Linear Regression

One of the interesting applications of the idea of projection is linear regression, where we try to model our data using linear functions. Before going to real data with multiple variables and several observations, let's start with a toy example.

10.4.1 Simple Linear Regression

When we have only one variable y depending linearly on another x, we get the normal line fitting, which is called simple linear regression, as opposed to multiple linear regression when we have multiple independent variables x_i

Figure 10.3 shows a data table with just two variables and five data points. We can think of y as a dependent variable, and x the independent one on which y depends. The dependence is modeled as a line, $y = mx + c$, which is the simple linear regression model. With the data shown in Figure 10.3, we can easily type in the numbers and get a "trendline" from our popular spreadsheet applications, which

x	y
1	1
2	2
3	2
4	5

$y = 1.2x - 0.5$
$R^2 = 0.8$

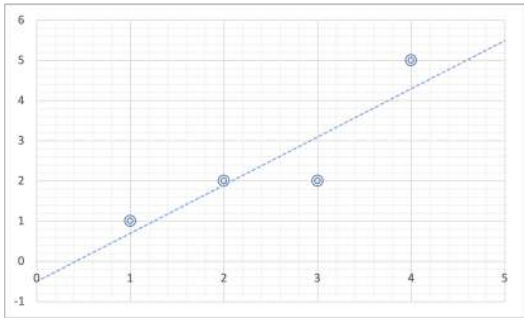


Fig. 10.3 An example of simple linear regression. The data points in the table on the left are plotted in the chart on the right, and a "trendline" is estimated and drawn.

gives us the values of $m = 1.2$ and $c = -0.5$ as shown. Let's look at how these data points become a system of linear equations, and how their solution then become a projection problem.

$$\begin{aligned} y &= mx + c \\ 1 &= m + c \\ 2 &= 2m + c \\ 2 &= 3m + c \\ 5 &= 4m + c \end{aligned} \quad \boldsymbol{Ax} = \boldsymbol{b} \quad \boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{bmatrix} \quad \boldsymbol{x} = \begin{bmatrix} m \\ c \end{bmatrix} \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 5 \end{bmatrix}$$

Since we are modeling our data as $y = mx + c$ and we have five (x, y) pairs, we get five equations as shown above, which we massage into the $\boldsymbol{Ax} = \boldsymbol{b}$ form. Notice how $\boldsymbol{A}$ has two columns, the first for m and another one for c full of ones. If we had written our model as $y = c + mx$, the column for the intercept c would have been the first one.

As we can see, we have five equations, and if we were to do Gauss-Jordan elimination on $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$, the third row would indicate an inconsistent equation $0 = 1$, and the system is not solvable, which is fine by us at this point in our Linear Algebra journey. We will get the best possible solution $\hat{\boldsymbol{x}}$, which will give us our best estimates for the slope and the intercept as $\hat{m}$ and $\hat{c}$. The steps are shown below.

$$\boldsymbol{A}^{\mathsf{T}} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A} = \begin{bmatrix} 30 & 10 \\ 10 & 4 \end{bmatrix} \quad \left|\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right| = 20$$

$$\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1} = \begin{bmatrix} \frac{1}{5} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{3}{2} \end{bmatrix}$$

$$\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}} = \begin{bmatrix} -\frac{3}{10} & -\frac{1}{10} & \frac{1}{10} & \frac{3}{10} \\ 1 & \frac{1}{2} & 0 & -\frac{1}{2} \end{bmatrix}$$

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{b} = \begin{bmatrix} \frac{6}{5} \\ -\frac{1}{2} \end{bmatrix} \implies \hat{m} = \frac{6}{5} \text{ and } \hat{c} = -\frac{1}{2}$$

As we can see, our linear regression model becomes $y = \hat{m}x + \hat{c} = 1.2x - 0.5$, same as the trendline that the spreadsheet application computed in Figure 10.3.

We also have the error vector $\boldsymbol{e} = \boldsymbol{b} - \hat{\boldsymbol{b}}$. $\boldsymbol{b}$ is what we project onto the column space, $\mathcal{C}(\boldsymbol{A})$ and $\hat{\boldsymbol{b}} = \boldsymbol{P}\boldsymbol{b}$ is the projection. Let's go ahead and compute $\boldsymbol{P}$ and $\hat{\boldsymbol{b}}$ as well. Using the formula for the projection matrix from Eqn (10.3), we get:

$$\boldsymbol{P} = \boldsymbol{A}\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}} = \begin{bmatrix} \frac{7}{10} & \frac{2}{5} & \frac{1}{10} & -\frac{1}{5} \\ \frac{2}{5} & \frac{3}{10} & \frac{1}{5} & \frac{1}{10} \\ \frac{1}{10} & \frac{1}{5} & \frac{3}{10} & \frac{2}{5} \\ -\frac{1}{5} & \frac{1}{10} & \frac{2}{5} & \frac{7}{10} \end{bmatrix}$$

$$\hat{\boldsymbol{b}} = \boldsymbol{P}\boldsymbol{b} = \begin{bmatrix} \frac{7}{10} \\ \frac{19}{10} \\ \frac{31}{10} \\ \frac{43}{10} \end{bmatrix} \quad \hat{\boldsymbol{e}} = \boldsymbol{b} - \hat{\boldsymbol{b}} = \begin{bmatrix} \frac{3}{10} \\ \frac{1}{10} \\ -\frac{11}{10} \\ \frac{7}{10} \end{bmatrix} \quad \|\boldsymbol{e}\| = \frac{9}{5}$$

The norm of the error vector is the variance ($\|\boldsymbol{e}\|$) in the data that is explained by the model. The total variance is computed independently (using its formula from statistics) as $\sigma^2 = 2.25$. The coefficient of determination R^2 is the fraction of the variance in the data that is explained by our model, and it is 0.8, just as the trendline from the spreadsheet application reports it in Figure 10.3.

10.4.2 Multiple Linear Regression

When we have one dependent variable and one independent variable, it is a simple linear regression (SLR). The statistical model in SLR is a line, which means we are saying that the data points all should have been on a line, but for some reason, they may wander and fluctuate; therefore let's look for the best fitting line.

When we have multiple independent variables, we have multiple linear regression (MLR). For two independent variables, the model is a plane, and we are trying to find the best plane that fits the data. For n independent variables, the model is an n-dimensional subspace in the *coordinate* space (*not* a *vector* subspace) that best describes the data.

Height (cm)	Weight (kg)	Hair Len. (cm)	Age (years)	Sex (M/F)
169.0	60.0	10.0	21.2	M
174.0	73.0	12.0	26.8	M
163.0	58.0	22.0	21.0	F
163.0	55.0	18.0	20.0	F
174.0	67.0	15.0	22.3	M
171.0	69.0	3.0	23.9	M
162.5	50.0	40.0	22.5	F
179.5	74.0	7.0	25.7	M
180.0	76.0	5.0	24.2	M
170.0	65.0	8.0	24.6	M
176.0	74.0	7.0	23.1	M
165.0	46.0	20.0	22.6	F
161.0	50.0	23.0	22.2	F
145.0	42.0	40.0	23.1	F
163.0	46.0	27.0	23.9	F
160.0	50.0	25.0	23.0	F
160.0	53.0	27.0	21.0	F
160.0	47.0	30.0	23.0	F
160.0	51.0	30.0	21.0	F
156.0	49.0	25.0	20.5	F
160.0	56.0	28.0	21.6	F
158.0	60.0	40.0	22.4	F
182.0	75.0	5.0	25.3	M
179.0	62.0	8.0	22.2	M
165.0	59.0	5.1	23.1	M
175.0	59.0	12.0	23.1	M
⋮	⋮	⋮	⋮	⋮

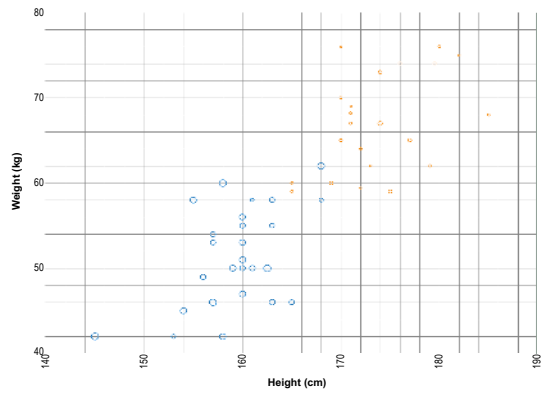


Fig. 10.4 Example of a data matrix and its visualization in multiple linear regression.

In Figure 10.4, we have the first 26 rows of a dataset of 127 observations of weight, height, length of hair, age and sex, arranged as a matrix. This so-called Young Adult dataset was used for an unrelated research project. The visualization shows Height in the x axis and Weight in y. The size of the point encodes Hair Len., which we can think of on the z axis coming towards us. The bigger bubbles indicate larger Hair Len., because they are closer to us. Finally the color indicates the last column, Sex: Blue for F and orange for M.

Treating this as a teachable moment, we are going to switch from our standard $\boldsymbol{Ax} = \boldsymbol{b}$ notation to what is generally used in the

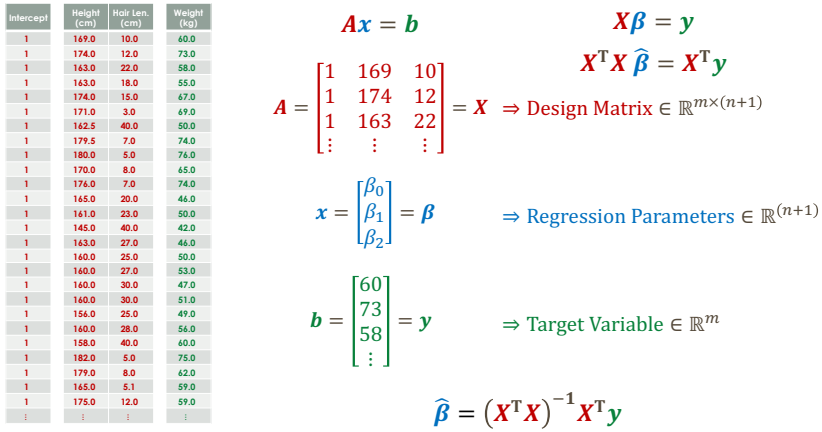


Fig. 10.5 The standard notations used in multiple linear regression. Weight is the dependent (or target or output) variable. Height and Hair Len. are the independent (or predictor or input) variables.

literature in the context of MLR, as shown in Figure 10.5, color coded for easy comprehension by our tired brains. Notice that in the case of SLR, we had the model $y = mx + c$, with the intercept introducing a column of ones in our $\boldsymbol{A}$ matrix. In MLR, we are going to keep our intercept as the first parameter, and the column of ones will be the first in our matrix, which we will now call $\boldsymbol{X}$ with the fancy name Design Matrix. Our model is

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \quad \text{or} \quad \text{Weight} = \beta_0 + \beta_1 \text{ Height} + \beta_2 \text{ Hair Len.}$$

Following the same matrix equations, now with the new notations as in Figure 10.5, we get the best estimate for the parameter vector $\hat{\beta}$, so that our model (coming from the 26 data points shown in Figures 10.4 and 10.5) becomes:

$$\begin{aligned} \text{Weight} &= \hat{\beta}_0 + \hat{\beta}_1 \text{ Height} + \hat{\beta}_2 \text{ Hair Len.} \\ &= -74.06 + 0.814 \text{ Height} - 0.151 \text{ Hair Len.} \end{aligned}$$

Although it is not easy to visualize the model and the points, even in the simple intuitive three-dimensional space, we have attempted to show this model in Figure 10.5. What is perhaps more important is to understand the model in terms of its parameters: We can see that

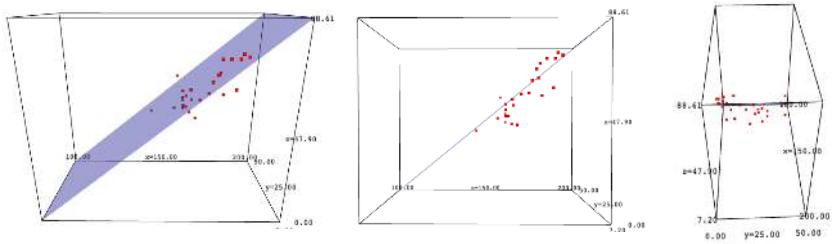


Fig. 10.6 Attempt to visualize a three dimensional MLR model for Weight. All three panels show the model (which is a plane) and the associated data points, but from different perspectives. The middle one shows the dependency of Weight on Height, and the last one shows that on Hair Len.

$\hat{\beta}_1 = 0.814$, a positive number indicating that the weight increases as the height increases. On the other hand, $\hat{\beta}_2 = -0.151$, which tells us that the weight decreases as the hair length increases, which is consistent with the fact that women tend to have longer hair and they tend to be smaller and lighter.

As in the case of SLR, we could compute R^2 and analyze whether it is appropriate to have the hair length in the model and so on, which is what we might expect to see in a book on data analytics. This book, however, is written for Linear Algebra, and this is probably an appropriate point to stop our work on its geometric aspects.

Part IV

Advanced Topics

11

Eigenvalue Decomposition and Diagonalization

Would it save you a lot of time if I just gave up and went mad now?

—Douglas Adams

We have completed the basics of Linear Algebra. We may have gone a bit beyond the basics in its algebraic and geometric aspects. Now it is time to switch gears and look at some topics that have enormous impact in computer science as well as more classical sciences. Eigenvalues are eigenvectors are the entry point to such topics. Although it appears in the "Advanced Topics" part of this book, eigenvalue decomposition and the associated discussion usually appear toward the end of all undergraduate-level courses on Linear Algebra.

The word "eigen" is German, and it means *own* or *characteristic*. We may, therefore, see some people calling the eigenvectors the characteristic vectors, although it is not common. What is much more common is to call the expression that gives us the eigenvalues of a matrix its *characteristic polynomial*.

11.1 Definition and Notation

We talked about $\boldsymbol{Ax} = \boldsymbol{b}$ as a transformation $\boldsymbol{A} \in \mathbb{R}^{m \times n} : \mathbb{R}^n \mapsto \mathbb{R}^m$. If we consider square matrices $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, then the mapping is from $\mathbb{R}^n$ to $\mathbb{R}^n$. We can then say that the matrix $\boldsymbol{A}$ is a mapping from a space to itself: It takes vectors in one space and transforms them to other vectors in the *same* space.

Are there vectors in the space that get transformed to a scalar multiple of itself? If there are, such vectors are called eigenvectors. Writing this statement in symbols, we come up with their definition.

Eigenvectors and eigenvalues

Definition: For $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, $\boldsymbol{s} \in \mathbb{R}^n \neq \boldsymbol{0}$ is an eigenvector if $\boldsymbol{As} = \lambda \boldsymbol{s}$ with the eigenvalue λ.

A few points to note about eigenvalues and eigenvectors:

1. For $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, its eigenvalues do not have to be in $\mathbb{R}$. In other words, just because we have square matrix over the field of reals ($\boldsymbol{A} \in \mathbb{R}^{n \times n}$), we cannot assume that its eigenvalues are real; it may have complex or imaginary eigenvalues.

2. Similarly, not all real matrices ($\boldsymbol{A} \in \mathbb{R}^{n \times n}$) have real eigenvectors.

3. If $\boldsymbol{s}$ is an eigenvector of $\boldsymbol{A}$ with an eigenvalue λ, so is any scaled version of it ($r\boldsymbol{s}$), with the *same* eigenvalue λ. Proof:

$$\boldsymbol{A}(r\boldsymbol{s}) = r\boldsymbol{A}\boldsymbol{s} = r\lambda s = \lambda(r\boldsymbol{s})$$

4. As a special case, λ can be zero. When $\lambda = 0$, we have $\boldsymbol{As} = \boldsymbol{0} \implies \boldsymbol{s} \in \mathcal{N}(\boldsymbol{A})$. Note that $\boldsymbol{s} \neq \boldsymbol{0}$.

5. Although presented here in terms of matrices and vectors, the ideas behind the eigenvectors came from other fields, such as physics. For this reason, eigenvectors are defined as those vectors that do not change their "direction" when $\boldsymbol{A}$ applies on them.

   In this book, however, for our own overly pedantic reasons, we stay away from the notion of "direction" of vectors to the extent possible.

11.2 Examples of Eigenvalues and Eigenvectors

Before writing down a general method for finding eigenvalues and eigenvectors, let's look at a few examples.

11.2.1 Permutation Matrix

A permutation matrix is the one that shuffles the elements of a vector (or the rows of a matrix, as we saw earlier in §4.3.4, page 82, when dealing with elementary matrices). In $\mathbb{R}^2$, the only possible permutation is $r_1 \leftrightarrow r_2$ with the following $\boldsymbol{A}$.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \implies \boldsymbol{A}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{A}\begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} = -1\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

As we can see above, we have two eigenvectors for this matrix, with eigenvalues 1 and -1.

11.2.2 Projection Matrix

We came across projection matrices $\boldsymbol{P} \in \mathbb{R}^{n\times n}$, which take any vector $\boldsymbol{x} \in \mathbb{R}^n$ to its projection onto a subspace $\mathcal{S} \subset \mathbb{R}^n$. If the vector $\boldsymbol{x}$ that $\boldsymbol{P}$ is acting on is already in the subspace $\mathcal{S}$, we know that its projection is itself. Calling these vectors $\boldsymbol{x}_{\parallel}$, we can write:

$$\boldsymbol{P}\boldsymbol{x}_{\parallel} = \boldsymbol{x}_{\parallel} = 1 \times \boldsymbol{x}_{\parallel}$$

So we have a whole bunch of eigenvectors $\boldsymbol{x}_{\parallel}$ in $\mathcal{S}$. Furthermore, the eigenvalues for these eigenvectors would be one, $\lambda = 1$.

We also know that if $\boldsymbol{x}$ is orthogonal to the subspace $\mathcal{S}$, the projection will be the zero vector. Calling such vectors $\boldsymbol{x}_{\perp}$, we write:

$$\boldsymbol{P}\boldsymbol{x}_{\perp} = \boldsymbol{0} = 0 \times \boldsymbol{x}_{\perp}$$

So $\boldsymbol{x}_{\perp}$ is an eigenvector with $\lambda = 0$. Note that for any matrix $\boldsymbol{A}$, we can always write $\boldsymbol{A}\boldsymbol{0} = \boldsymbol{0}$, but $\boldsymbol{0}$ is *not* an eigenvector by our definition above.

11.2.3 Shear Matrix

A shear matrix transforms a square to a parallelogram[1]. Here is an example of a horizontal shear matrix, as shown in Figure 11.1.

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \implies \boldsymbol{A}\boldsymbol{q_1} = \boldsymbol{A}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{A}\boldsymbol{q_2} = \boldsymbol{A}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

When any vector is transformed by $\boldsymbol{A}$, its first component is not affected. A horizontal shear leaves the x axis alone, and therefore $\boldsymbol{q_1}$ is an eigenvector with eigenvalue $\lambda = 1$.

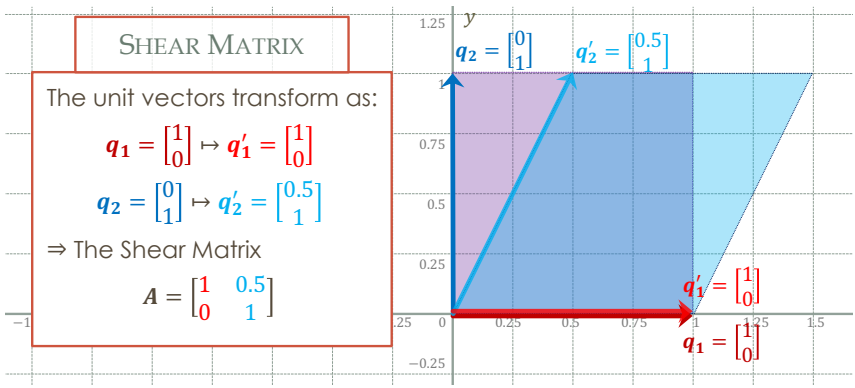


Fig. 11.1 An example shear matrix, showing a square being transformed into a parallelogram.

11.2.4 Rotation Matrix

We looked at rotation matrices earlier, when talking about orthogonal matrices. In $\mathbb{R}^2$, the rotation matrix is

$$\boldsymbol{Q_\theta} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

[1] Although we state it like this, we should note that squares and parallelograms do not exist in a vector space. They live in coordinate spaces, and this statement is an example of the Notational Abuse, about which we complained in a box earlier. What we mean is that the two vectors forming the sides of a unit square get transformed such that they form sides of a parallelogram. We should perhaps eschew our adherence to this pedantic exactitude, now that we are in the advanced section.

For any nontrivial θ (which means $\theta \neq 2k\pi$ for integer k), we can see that $\boldsymbol{Q_\theta}$ changes every single vector in $\mathbb{R}^2$. We have no eigenvectors for this matrix in $\mathbb{R}^2$.

11.2.5 Differentiation

We can think of the set of all functions (of one variable, for instance) as a vector space. It satisfies all the requisite properties. The calculus operation of differentiation is a linear transformation in this space; it satisfies both the homogeneity and additivity properties of linearity.

$$\frac{d}{dx}\, e^{ax} = ae^{ax} \implies e^{ax} \text{ is an eigenvector with eigenvalue } a$$

$$\frac{d^2}{dx^2}\, \sin x = -\sin x \implies \sin x \text{ is an eigenvector with eigenvalue } -1$$

11.3 Computing Eigenvalues and Finding Eigenvectors

In order to find the eigenvalues and then eigenvectors, we start from their definitions.

$$\boldsymbol{A}\boldsymbol{s} = \lambda \boldsymbol{s} \implies (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{s} = \boldsymbol{0}$$

Remembering that $\boldsymbol{s} \neq \boldsymbol{0}$, we can see that $\boldsymbol{A} - \lambda \boldsymbol{I}$ has a nontrivial null space (to which the eigenvector $\boldsymbol{s}$ belongs). Since $\mathcal{N}(\boldsymbol{A} - \lambda \boldsymbol{I})$ has nonzero vectors in it, $\boldsymbol{A} - \lambda \boldsymbol{I}$ is singular and its determinant has to be zero, which gives us an equation for eigenvalues.

$$|\boldsymbol{A} - \lambda \boldsymbol{I}| = 0$$

Thus, we get rid of $\boldsymbol{s}$ and end up with a polynomial in λ (when we expand the determinant) equalling zero from which we can solve for the possible values of λ. This polynomial is called the *characteristic polynomial* of the matrix.

For a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, when we expand the determinant $|\boldsymbol{A} - \lambda \boldsymbol{I}|$ using the Laplace formula in Eqn (3.5), we get a polynomial of order n in λ, which should give us n roots, but not all of them may be real. Once we have the eigenvalues, we can put them back in $(\boldsymbol{A} - \lambda \boldsymbol{I}) = \boldsymbol{0}$ to find the eigenvectors $\boldsymbol{s}$, which is the same as finding the null space of $(\boldsymbol{A} - \lambda \boldsymbol{I})$.

Let's look at the examples from the previous section again to see how we get the eigenvalues and eigenvectors.

Permutation Matrix

Starting from the permutation matrix in $\mathbb{R}^2$, here are the steps.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad |\boldsymbol{A} - \lambda \boldsymbol{I}| = 0 \implies \begin{vmatrix} -\lambda & 1 \\ 1 & -\lambda \end{vmatrix} = 0$$

$$\text{Expanding the determinant, } \lambda^2 - 1 = 0 \implies \lambda = \pm 1$$

As we saw earlier, we have two eigenvalues, 1 and -1. To find the corresponding eigenvectors, we substitute the λ values in either $\boldsymbol{A}\boldsymbol{s} = \lambda \boldsymbol{s}$ and solve, or, equivalently, find the null space of $\boldsymbol{A} - \lambda \boldsymbol{I}$.

$$\text{With } \lambda = 1, \quad (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{s} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \boldsymbol{s} = \boldsymbol{0} \implies \boldsymbol{s} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\text{With } \lambda = -1, \quad (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{s} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \boldsymbol{s} = \boldsymbol{0} \implies \boldsymbol{s} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Here, to find $\boldsymbol{s}$, we are still using the column-picture of matrix multiplication and figuring out what linear combinations of the columns of $\boldsymbol{A} - \lambda \boldsymbol{I}$ give the zero vector. As we can see, we get two distinct eigenvectors. We can also see that the eigenvectors are actually orthogonal to each other. Note that any scaled versions of the eigenvectors are still eigenvectors with the same λ. For this reason, we typically normalize them.

The fact that we got real eigenvalues and distinct and orthogonal eigenvectors is not an accident. Real symmetric matrices (as our $\boldsymbol{A}$ was, in the case of this permutation matrix) always have real eigenvalues and a full set of orthogonal eigenvectors. They are the best matrices to work with.

Projection Matrix

As a concrete example, let's consider the projection matrix in $\mathbb{R}^3$ that projects vectors to the xy plane. Still calling it $\boldsymbol{A}$, what $\boldsymbol{A}$ needs to do is to leave the first two components alone and make the last component zero.

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad |\boldsymbol{A} - \lambda \boldsymbol{I}| = 0 \implies \begin{vmatrix} 1-\lambda & 0 & 0 \\ 0 & 1-\lambda & 0 \\ 0 & 0 & -\lambda \end{vmatrix} = 0$$

$$\text{Expanding the determinant, } -\lambda(1-\lambda)^2 = 0 \implies \lambda = 0, 1, 1$$

Again, as we saw earlier, we have two eigenvalues, 0 and 1. But note that $\lambda = 1$ is repeated. We state this fact more fancifully, that the *algebraic multiplicity* of the eigenvalue ($\lambda = 1$) is two. Let's go ahead and try to find the corresponding eigenvectors.

$$\text{With } \lambda = 0, \quad (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{s} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{s} = \boldsymbol{0} \implies \boldsymbol{s} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\text{With } \lambda = 1, \quad (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{s} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \boldsymbol{s} = \boldsymbol{0} \implies \boldsymbol{s} = \begin{bmatrix} t_1 \\ t_2 \\ 0 \end{bmatrix}$$

Where t_1 and t_2 are any real numbers. We see that we have a small issue with the second eigenvalue with algebraic multiplicity two: The eigenvectors corresponding to it span a subspace, which is called the *eigenspace* associated with the second eigenvalue ($\lambda = 1$). The dimension of this eigenspace is two, which is called its *geometric multiplicity*. All eigenvalues have eigenspaces associated with them, with geometric multiplicity of at least one.

What we need to do when we have an eigenvalue with a geometric multiplicity greater than one is to select any full set of linearly independent vectors that span its eigenspace. In other words, we take its basis as the eigenvectors. In this particular case of projecting to the xy plane, the perfect basis would be the unit vectors along x and y directions. Putting it all together, here is the full solution:

$$\lambda_1 = 0, \boldsymbol{s_1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \lambda_2 = 1, \boldsymbol{s_2} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{s_3} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Shear Matrix

Moving on to our next example,

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \quad |\boldsymbol{A} - \lambda \boldsymbol{I}| = 0 \implies \begin{vmatrix} 1 - \lambda & 0.5 \\ 0 & 1 - \lambda \end{vmatrix} = 0$$

$$\text{Expanding the determinant, } (1 - \lambda)^2 = 0 \implies \lambda = 1$$

As we saw earlier, we have a single eigenvalue of 1, but with an algebraic multiplicity of two. The corresponding eigenvector is.

$$\text{With } \lambda = 1, \quad (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{s} = \begin{bmatrix} 0 & 0.5 \\ 0 & 0 \end{bmatrix} \boldsymbol{s} = \boldsymbol{0} \implies \boldsymbol{s} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

The eigenvalue has an algebraic multiplicity of two, and a geometric multiplicity of one. We cannot find a full set of (real) eigenvectors, and we are in trouble because for $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, we would like to have n eigenvectors.

Rotation Matrix

We saw that there were no eigenvectors for a rotation matrix in $\mathbb{R}^2$. Let's consider a $\frac{\pi}{2}$-rotation, call the matrix $\boldsymbol{A}$ and attempt to find its eigenvalues and eigenvectors.

$$\boldsymbol{Q_\theta} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} \cos\frac{\pi}{2} & -\sin\frac{\pi}{2} \\ \sin\frac{\pi}{2} & \cos\frac{\pi}{2} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \boldsymbol{A}$$

$$|\boldsymbol{A} - \lambda \boldsymbol{I}| = 0 \implies \begin{vmatrix} -\lambda & -1 \\ 1 & -\lambda \end{vmatrix} = 0$$

$$\text{Expanding the determinant, } \lambda^2 + 1 = 0 \implies \lambda = \pm i$$

We have no real eigenvalues. For the sake of completeness, we can try to find the eigenvectors, although we do not expect to find any in $\mathbb{R}^2$.

$$\text{With } \lambda = i, \quad (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{s} = \begin{bmatrix} -i & -1 \\ 1 & -i \end{bmatrix} \boldsymbol{s} = \boldsymbol{0} \implies \boldsymbol{s} = \begin{bmatrix} 1 \\ -i \end{bmatrix}$$

$$\text{With } \lambda = -i, \quad (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{s} = \begin{bmatrix} i & -1 \\ 1 & i \end{bmatrix} \boldsymbol{s} = \boldsymbol{0} \implies \boldsymbol{s} = \begin{bmatrix} 1 \\ i \end{bmatrix}$$

Thus, if we allow ourselves to step into the field of complex numbers, we can find eigenvectors of the rotation matrix $\boldsymbol{s} \in \mathbb{C}^2$. Physically, the vector that is conserved during rotation is perpendicular to the plane of rotation, which is why gyroscopes work the way they do. How that fact corresponds to the actual eigenvectors of the matrix we computed above, however, is a fairly tortured explanation.

11.4 Properties

The eigenvalues and eigenvectors provide deep insights into the structure of the matrix, and have properties related to the properties of the matrix itself. Here are some of them with proofs, where possible. It is worth our time to verify these properties on the examples we worked out above.

11.4.1 Eigenvalues

1. The sum of eigenvalues equals the trace of the matrix. For an $n \times n$ matrix $\boldsymbol{A} = \big[a_{ij}\big]$,

$$\sum_{i=1}^{n} \lambda_i = \text{trace}(\boldsymbol{A}) = \sum_{i=1}^{n} a_{ii}$$

Proof: Since the characteristic polynomial has roots λ_i, we can write:

$$|\boldsymbol{A} - \lambda \boldsymbol{I}| = (-1)^n(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n)$$

where we constructed the RHS to have roots λ_i and to match the coefficient of λ^n in the determinant on the LHS, which expands to give a polynomial in λ with the coefficient of $\lambda^n = (-1)^n$. On the LHS, the coefficient of λ^{n-1} is

$$(-1)^n \sum_{i=1}^{n} a_{ii}$$

On the RHS, the coefficient of λ^{n-1} is

$$(-1)^n \sum_{i=1}^{n} \lambda_i$$

Since the LHS and RHS coefficients have to match, we see that

$$\sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} a_{ii} = \text{trace}(\boldsymbol{A})$$

which proves the property.

Although we proved it by comparing the coefficient of λ^{n-1}, it is a lot easier to prove once we learn matrix similarity in the next chapter. Note that we have to include λ_i in the summation as many times as its algebraic multiplicity. Note also that for $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, this property means that if $\boldsymbol{A}$ has any complex eigenvalues, they should come in pairs of complex conjugates.

2. The product of eigenvalues equals the determinant of the matrix. For an $n \times n$ matrix $\boldsymbol{A}$,

$$\prod_{i=1}^{n} \lambda_i = |\boldsymbol{A}|$$

Proof: Again we start with the equality:

$$|\boldsymbol{A} - \lambda \boldsymbol{I}| = (-1)^n(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n)$$

Set $\lambda = 0$ to get

$$|\boldsymbol{A}| = (-1)^n(-1)^n\lambda_1\lambda_2\cdots\lambda_n = \lambda_1\lambda_2\cdots\lambda_n = \prod_{i=1}^{n}\lambda_i$$

A corollary of this property is that singular matrices have at least one zero eigenvalue.

3. The eigenvalues of a real, symmetric matrix are real.

$$\boldsymbol{A} \in \mathbb{R}^{n\times n}, \boldsymbol{A}^{\mathsf{T}} = \boldsymbol{A} \implies \lambda_i \in \mathbb{R}$$

Proof: To prove this property, we have to step into the scary field of complex numbers again. The strategy for proving something is real is to assume that it is complex, and then show that its conjugate (where we replace all i with $-i$) is the same as itself, which shows that it has to be real. Following this strategy, let's say that $\lambda \in \mathbb{C}$ is a possibly complex eigenvalue of $\boldsymbol{A}$, and $\boldsymbol{s} \in \mathbb{C}^{n\times n}$ is the corresponding eigenvector. (Note that in step (2) below, we take the complex conjugate, which is defined as: $(a + ib)^* = a - ib$.)

(1) By the definition of eigenvalues, we have:	$\boldsymbol{A}\boldsymbol{s} = \lambda\boldsymbol{s}$
(2) Taking the complex conjugate, we get:	$\boldsymbol{A}^*\boldsymbol{s}^* = \lambda^*\boldsymbol{s}^*$
(3) Since $\boldsymbol{A}$ is real $\boldsymbol{A}^* = \boldsymbol{A}$. Therefore:	$\boldsymbol{A}\boldsymbol{s}^* = \lambda^*\boldsymbol{s}^*$
(4) Multiplying (1) on the left with $\boldsymbol{s}^{*\mathsf{T}}$:	$\boldsymbol{s}^{*\mathsf{T}}\boldsymbol{A}\boldsymbol{s} = \boldsymbol{s}^{*\mathsf{T}}\lambda\boldsymbol{s}$
(5) Flipping it around and reordering:	$\lambda\boldsymbol{s}^{*\mathsf{T}}\boldsymbol{s} = \boldsymbol{s}^{*\mathsf{T}}\boldsymbol{A}\boldsymbol{s}$
(6) Using the product rule of transposes:	$\lambda\boldsymbol{s}^{*\mathsf{T}}\boldsymbol{s} = \left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{s}^*\right)^{\mathsf{T}}\boldsymbol{s}$
(7) Since $\boldsymbol{A}$ is symmetric:	$= \left(\boldsymbol{A}\boldsymbol{s}^*\right)^{\mathsf{T}}\boldsymbol{s}$
(8) Using step (3) above:	$= \lambda^*\boldsymbol{s}^{*\mathsf{T}}\boldsymbol{s}$

Finally, from steps (6) and (8), we get:

$$\lambda\boldsymbol{s}^{*\mathsf{T}}\boldsymbol{s} = \lambda^*\boldsymbol{s}^{*\mathsf{T}}\boldsymbol{s} \implies (\lambda - \lambda^*)\boldsymbol{s}^{*\mathsf{T}}\boldsymbol{s} = 0 \implies \lambda = \lambda^*$$

since $\boldsymbol{s}^{*\mathsf{T}}\boldsymbol{s}$ is the square of the norm $\|\boldsymbol{s}\|^2$ (it is, for $\boldsymbol{s} \in \mathbb{C}^n$, as we shall see in the next chapter) of an eigenvector, which cannot be the zero vector $\boldsymbol{0}$. $\lambda = \lambda^*$ means $\lambda \in \mathbb{R}$.

4. The "opposite" of the previous property: The eigenvalues of a real, antisymmetric (AKA skew symmetric) matrix are imaginary.

$$\boldsymbol{A} \in \mathbb{R}^{n\times n}, \boldsymbol{A}^{\mathsf{T}} = -\boldsymbol{A} \implies \lambda_i = i\lambda_i', \text{ where } \lambda_i' \in \mathbb{R}$$

Proof: Identical to the previous proof, but with a negative sign in step (7), showing $\lambda = -\lambda^*$.

5. If we multiply a matrix ($\boldsymbol{A}$) by a scalar (α), then all its eigenvalues (λ_i) are multiplied by the same scalar.
Proof: $\boldsymbol{A}\boldsymbol{s} = \lambda\boldsymbol{s} \implies (\alpha\boldsymbol{A})\boldsymbol{s} = (\alpha\lambda)\boldsymbol{s}$. That is it.

6. The eigenvalues of $\boldsymbol{A} + \alpha\boldsymbol{I}$ are $\lambda_i + \alpha$, the eigenvalues of $\boldsymbol{A}$ "shifted" by α.
Proof: If $\boldsymbol{s}$ is an eigenvector of $\boldsymbol{A}$ with the eigenvalue λ, we have:

$$(\boldsymbol{A} + \alpha\boldsymbol{I})\boldsymbol{s} = \boldsymbol{A}\boldsymbol{s} + \alpha\boldsymbol{s} = \lambda\boldsymbol{s} + \alpha\boldsymbol{s} = (\lambda + \alpha)\boldsymbol{s}$$

which means $\boldsymbol{s}$ is an eigenvector of $\boldsymbol{A} + \alpha\boldsymbol{I}$ with the eigenvalue $\lambda + \alpha$, which goes for every eigenvector/eigenvalue pair.

7. The eigenvalues of real, symmetric matrices are related to the pivots in the row echelon form (REF, or $\boldsymbol{U}$ in the $\boldsymbol{LU}$ or $\boldsymbol{PLU}$ decomposition): The number of positive eigenvalues is the same as the number of positive pivots. Same goes for negative ones too. We will leave this property, known as Sylvester's Law of Inertia, without proof.

11.4.2 Eigenvectors

1. The eigenvectors of a matrix[2] corresponding to distinct eigenvalues are linearly independent.

$$\begin{gathered}\boldsymbol{A} \in \mathbb{R}^{n\times n} \text{ with } \boldsymbol{A}\boldsymbol{s_i} = \lambda_i\boldsymbol{s_i}, \boldsymbol{s_i} \in \mathbb{R}^n, \text{ for } 0 < i \leq n \\ \lambda_i \neq \lambda_j \text{ and } a_i\boldsymbol{s_i} + a_j\boldsymbol{s_j} = \boldsymbol{0} \\ \implies a_i = a_j = 0 \text{ for } 0 < i, j \leq n, i \neq j\end{gathered}$$

[2]We use $\mathbb{R}$ (the real field, $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{s} \in \mathbb{R}^n$) in the mathematical statement and proof of this property for convenience and because of its relevance to computer science, but the property applies to $\mathbb{C}$ as well.

Sylvester's Law of Inertia

The connection between the signs of the eigenvalues of a real, symmetric matrix and its pivots is called Sylvester's Law of Inertia. Inertia, in this context, is just the triplet Inertia$(\boldsymbol{A}) = (p, n, z)$, with $p-$ the number of positive eigenvalues, n, that of negative ones and z the zeros. What the rule states is that these numbers are the same as numbers of positive and negative pivots and the zero rows/columns (which, of course, is the nullity). Note that this applies only if $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{A}$.

The interesting corollary to this law is that the number of zero eigenvalues is the same as the nullity of the matrix. Remembering that $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\mathbb{R}^{n\times n}$ and is symmetric for any $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, we can extend the law to state that the rank of $\boldsymbol{A}$ (which is the same as the rank of $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}$) is the number of nonzero eigenvalues of $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}$.

The nullity of $\boldsymbol{A}$ is the number of pivotless columns in $\boldsymbol{A}$, which has to be the same as the number of pivotless columns in $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}$, Why? Because $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}$ and $\boldsymbol{A}$ have the same rank, and the same number of columns n. Therefore, the nullity of $\boldsymbol{A}$ is the number of zero eigenvalues of $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}$.

Proof: We need to prove that if $\boldsymbol{s_i}$ and $\boldsymbol{s_j}$ are eigenvectors of $\boldsymbol{A}$ with distinct eigenvalues λ_i and λ_j, we will not be able to find nonzero a_i and a_j such that $a_i\boldsymbol{s_i} + a_j\boldsymbol{s_j} = \boldsymbol{0}$.

(1) Starting from:	$a_i\boldsymbol{s_i} + a_j\boldsymbol{s_j}$	$= \boldsymbol{0}$
(2) Multiplying on the left by $\boldsymbol{A}$:	$a_i\boldsymbol{A}\boldsymbol{s_i} + a_j\boldsymbol{A}\boldsymbol{s_j}$	$= \boldsymbol{0}$
(3) Since $\boldsymbol{s_i}$ and $\boldsymbol{s_j}$ are eigenvectors:	$a_i\lambda_i\boldsymbol{s_i} + a_j\lambda_j\boldsymbol{s_j}$	$= \boldsymbol{0}$
(4) Multiplying (1) with λ_i, we get:	$a_i\lambda_i\boldsymbol{s_i} + a_j\lambda_i\boldsymbol{s_j}$	$= \boldsymbol{0}$
(5) Subtracting (3) from (4):	$a_j(\lambda_i - \lambda_j)\boldsymbol{s_j}$	$= \boldsymbol{0}$
(6) Since $\lambda_i \neq \lambda_j$ and $\boldsymbol{s_j} \neq \boldsymbol{0}$:	a_j	$= 0$
(7) Putting a_j in (1), we can show:	a_i	$= 0$

The converse of this statement is *not* true: If the eigenvalues are *not* distinct, the eigenvectors may still be linearly independent, as in the case of the projection-matrix. Or the identity matrix $\boldsymbol{I}_n$, which has the eigenvalue one repeated n times, but is already diagonalized.

2. Real symmetric matrices have a full set of orthogonal eigenvectors.

$$\boldsymbol{A} \in \mathbb{R}^{n\times n} \text{ with } \boldsymbol{A}\boldsymbol{s_i} = \lambda_i\boldsymbol{s_i}, \boldsymbol{s_i} \in \mathbb{R}^n, \text{ for } 0 < i \leq n$$
$$\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{A} \implies \boldsymbol{s_i} \perp \boldsymbol{s_j} \text{ for } 0 < i, j \leq n, i \neq j$$

Proof:

(1) By the definition of eigenvalues:	$\boldsymbol{A}\boldsymbol{s_i}$	$= \lambda_i \boldsymbol{s_i}$
(2) Taking the dot product with $\boldsymbol{s_j}$:	$(\boldsymbol{A}\boldsymbol{s_i})^\mathsf{T}\boldsymbol{s_j}$	$= \lambda_i \boldsymbol{s_i}^\mathsf{T}\boldsymbol{s_j}$
(3) Using the product rule of transposes:	$\boldsymbol{s_i}^\mathsf{T}\boldsymbol{A}^\mathsf{T}\boldsymbol{s_j}$	$= \lambda_i \boldsymbol{s_i}^\mathsf{T}\boldsymbol{s_j}$
(4) Since $\boldsymbol{A}$ is symmetric:	$\boldsymbol{s_i}^\mathsf{T}\boldsymbol{A}\boldsymbol{s_j}$	$= \lambda_i \boldsymbol{s_i}^\mathsf{T}\boldsymbol{s_j}$
(5) Since $\boldsymbol{s_j}$ is an eigenvector of $\boldsymbol{A}$:	$\boldsymbol{s_i}^\mathsf{T}\lambda_j\boldsymbol{s_j}$	$= \lambda_i \boldsymbol{s_i}^\mathsf{T}\boldsymbol{s_j}$
(6) Reordering:	$\lambda_j \boldsymbol{s_i}^\mathsf{T}\boldsymbol{s_j}$	$= \lambda_i \boldsymbol{s_i}^\mathsf{T}\boldsymbol{s_j}$
(7) Gathering terms:	$(\lambda_j - \lambda_i)\boldsymbol{s_i}^\mathsf{T}\boldsymbol{s_j}$	$= 0$
(8) Since $\lambda_j \neq \lambda_i \implies$	$\boldsymbol{s_i}^\mathsf{T}\boldsymbol{s_j}$	$= 0$
(9) $\boldsymbol{s_i}^\mathsf{T}\boldsymbol{s_j} = 0 \implies$	$\boldsymbol{s_i} \perp \boldsymbol{s_j}$	

It may happen that the eigenvalues are repeated. For instance, the identity matrix $\boldsymbol{I} \in \mathbb{R}^{n\times n}$ has n repeated eigenvalues of 1. Every vector in $\mathbb{R}^{n\times n}$ is an eigenvector of $\boldsymbol{I}$. In this case also, we can *choose* an orthogonal set of vectors as the full set of eigenvectors for the matrix.

In some cases, the eigenvectors may span a subspace (called eigenspace, of course), as in the case of the projection matrix. Here again, we can choose an orthogonal eigenbasis for the eigenspace associated with the repeated eigenvalue.

3. The eigenvectors of $\boldsymbol{A} + \alpha\boldsymbol{I}$ are the same as those of $\boldsymbol{A}$.
Proof: As we already proved, if $\boldsymbol{s}$ is an eigenvector of $\boldsymbol{A}$ with the eigenvalue λ,

$$(\boldsymbol{A} + \alpha\boldsymbol{I})\boldsymbol{s} = \boldsymbol{A}\boldsymbol{s} + \alpha\boldsymbol{s} = \lambda\boldsymbol{s} + \alpha\boldsymbol{s} = (\lambda + \alpha)\boldsymbol{s}$$

which should be proof enough. If not, it means $\boldsymbol{s}$ is an eigenvector of $\boldsymbol{A} + \alpha\boldsymbol{I}$ with the eigenvalue $\lambda + \alpha$.

11.5 Unit Circles and Ellipses

One fair question we may have at this point is why we are doing all this. It is all an academic exercise in intellectual acrobatics? We may not be able to answer this question completely yet, but we can look at a linear transformation and see what the eigenvalue analysis tells

us about it. In the last chapter, we will see how these insights are harnessed in statistical analyses.

Let's start with an example $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$, find its eigenvalues and eigenvectors, and look at them in the coordinate space $\mathbb{R}^2$.

$$\boldsymbol{A} = \frac{1}{4}\begin{bmatrix} 5 & -\sqrt{3} \\ -\sqrt{3} & 3 \end{bmatrix} \qquad \lambda_1 = \frac{3}{2};\ \boldsymbol{s_1} = \frac{1}{2}\begin{bmatrix} \sqrt{3} \\ -1 \end{bmatrix} \qquad \lambda_2 = \frac{1}{2};\ \boldsymbol{s_2} = \frac{1}{2}\begin{bmatrix} 1 \\ \sqrt{3} \end{bmatrix}$$

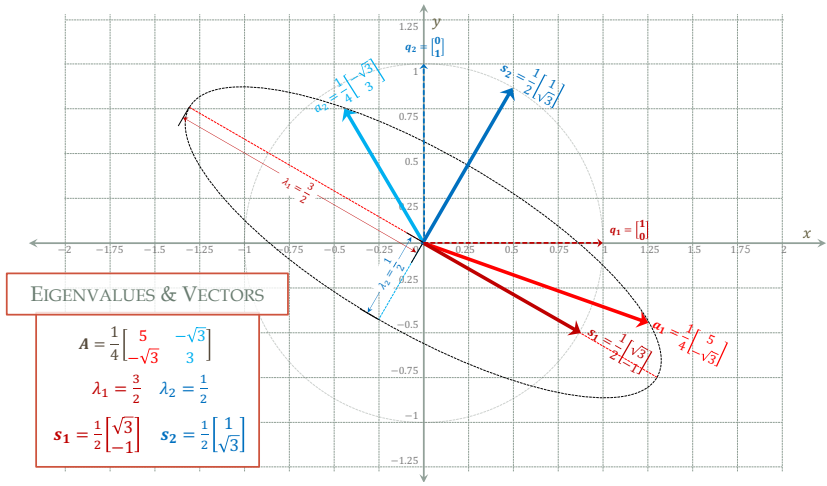


Fig. 11.2 Visualization of eigenvalues and eigenvectors: $\boldsymbol{A}$ transforms the unit circle into a rotate ellipse. The eigenvalues specify the lengths of its major and minor axes. And the eigenvectors specify the orientation of the axes.

As we can see from Figure 11.2, $\boldsymbol{A}$ takes the first basis vector ($\boldsymbol{q_1}$, shown in red, dashed arrow) to its first column vector (a_1 shown bright red arrow): $\boldsymbol{q_1} \mapsto \boldsymbol{a_1}$. Similarly for the second one as well, $\boldsymbol{q_2} \mapsto \boldsymbol{a_2}$, shown in various shades of blue. What happens to the basis vectors happens to all vectors, and therefore, the unit circle in the figure gets mapped to the ellipse, as shown.

Although we know about this unit-circle-to-ellipse business, from the matrix $\boldsymbol{A}$ itself, we know very little else. Note that the vectors to which the unit vectors transform (in $\boldsymbol{q_i} \mapsto \boldsymbol{a_i}$) are nothing special; they are on the ellipse somewhere. What we would like to know are the details of the ellipse, like its size and orientation, which is exactly what the eigenvalues and eigenvectors tell us. The eigenvalues λ_i are the lengths of the major and minor axes of the ellipse and the eigenvectors are the unit vectors along the axes. In Figure 11.2, the

eigenvectors ($\boldsymbol{s_1}$ and $\boldsymbol{s_2}$) are shown in darker shades of red and blue, while the corresponding eigenvalues are marked as the lengths of the axes.

When we move on to higher dimensions, ellipses become ellipsoids or hyper-ellipsoids, and the axes are their principal axes. The mathematics of eigenanalysis still stays the same: We get the directions and lengths of the principal axes. And, if the matrix on which we are performing the eigenanalysis happens to contain the covariance of the variables in a dataset, then what the eigenanalysis gives us are insights about the directions along which we can decompose the covariance. If we sort the directions by the eigenvalues, we can extract the direction for the highest variance, second highest variance and so on. We will revisit this idea in more detail in one of our last topics, the Principal Component Analysis, which is the mainstay of dimensionality reduction in data science.

11.6 Diagonalization

We learned quite a bit about eigenvalues and eigenvectors by now. We might still wonder why at this point. What is the point in learning all this trivia about them? We hinted at its significance in data analytics for dimensionality reduction. We have one more good reason; there is a method to this madness. Once we have the eigenvectors of a matrix, we can *diagonalize* it. And once we diagonalize, we can immediately see how it can help in computing the powers of the matrix. Why would we want to take powers of matrices? Because it is the basis of modeling time-varying systems mathematically.

11.6.1 S and Λ

Suppose $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ has its eigenvalues λ_i and the corresponding eigenvectors $\boldsymbol{s_i}$. Let's construct two matrices, arranging the eigenvectors as columns, and the eigenvalues as diagonal elements:

$$\boldsymbol{S} = \begin{bmatrix} | & | & \cdots & | \\ \boldsymbol{s_1} & \boldsymbol{s_2} & \cdots & \boldsymbol{s_n} \\ | & | & \cdots & | \end{bmatrix} \quad \boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

For simplicity, we may write $\boldsymbol{S} = [\boldsymbol{s}]$ and $\boldsymbol{\Lambda} = [\lambda]$. With these new matrices, we arrive at the most important result from this chapter:

$$\boldsymbol{AS} = \boldsymbol{S\Lambda}$$

The LHS is a matrix multiplication, where the product matrix has columns that are product of $\boldsymbol{A}$ and the corresponding column in $\boldsymbol{S}$. In other words, $\boldsymbol{AS} = \boldsymbol{A}[\boldsymbol{s_i}] = [\boldsymbol{As_i}]$. We think of the RHS using the column picture of matrix multiplication again: The columns of $\boldsymbol{S\Lambda} = [\boldsymbol{s_i}]\boldsymbol{\Lambda}$ are the linear combinations of $\boldsymbol{s_i}$ taken with the scaling factors in the columns of $\boldsymbol{\Lambda}$. But the scaling factors are merely λ_i in the i^{th} place. Therefore, the i^{th} column in $\boldsymbol{AS} = \boldsymbol{S\Lambda}$ is the same as $\boldsymbol{As_i} = \boldsymbol{s_i}\lambda_i$, which is the now-familiar the definition of eigenvalues and eigenvectors.

11.6.2 The Decomposition

If we know that $\boldsymbol{S}$ is invertible, we can go one step further and write:

$$\text{If } \boldsymbol{S^{-1}} \text{ exists, } \boldsymbol{AS} = \boldsymbol{S\Lambda} \implies \boldsymbol{A} = \boldsymbol{S\Lambda S^{-1}}$$

This is the famous eigenvalue decomposition of a real, square, *diagonalizable* matrix. Most matrices are diagonalizable over the field of complex numbers.

11.6.3 Powers of $\boldsymbol{A}$

As previously advertised, the reason for this decomposition is that it gives us a way to express the powers of the matrix. For a square, diagonalizable matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, we can write,

$$\begin{aligned}\boldsymbol{A} = \boldsymbol{S\Lambda S^{-1}} \implies \boldsymbol{A}^2 &= \boldsymbol{S\Lambda S^{-1}S\Lambda S^{-1}} = \boldsymbol{S\Lambda}(\boldsymbol{S^{-1}S})\boldsymbol{\Lambda S^{-1}} \\ &= \boldsymbol{S\Lambda I\Lambda S^{-1}} = \boldsymbol{S\Lambda\Lambda S^{-1}} \\ \boldsymbol{A}^2 &= \boldsymbol{S\Lambda}^2\boldsymbol{S^{-1}}\end{aligned}$$

Similarly, we can easily see that $\boldsymbol{A}^k = \boldsymbol{S\Lambda}^k\boldsymbol{S^{-1}}$. While this result may look fairly mundane, let's think about a medium-sized matrix, say 100×100, and $k = 60$. Imagine the number of operations required to compute $\boldsymbol{A}^{60}$, which is probably of the order of $60\times100^3 = 6\times10^7$ (for the 60 matrix multiplications). But if we have the decomposition, taking the 60^{th} power of $\boldsymbol{\Lambda}$ is trivial, merely 100 exponentiations. The two matrix multiplications cost about $2 \times 100^3 = 2 \times 10^6$ operations

Invertibility vs. Diagonalizability

Not all invertible matrices can be diganolized. We saw an example earlier. The shear matrix has an inverse that is easily written down.:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{A^{-1}} = \begin{bmatrix} 1 & -0.5 \\ 0 & 1 \end{bmatrix}$$

But $\boldsymbol{A}$ has only one eigenvector, and is not diagonalizable.

Not all diagonalizable matrices are invertible. We already know that the projection matrix is not invertible because it destroys information; it is a many-to-one (or injective) mapping. But it can be diagonalized.

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{S} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{S^{-1}} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \boldsymbol{S^{T}}$$

$$\boldsymbol{A} = \boldsymbol{S\Lambda S^{-1}}$$

In fact, $\boldsymbol{A}$ is already a diagonal matrix: It has nonzero elements only along its diagonal.

We know the condition for $\boldsymbol{A}$ to be invertible, or for $\boldsymbol{A^{-1}}$ to exist. Let's state it several different ways, as a means to remind ourselves. $\boldsymbol{A}$ is invertible if:

- $|\boldsymbol{A}| \neq 0$. Otherwise, as Eqn (5.1) clearly shows, we cannot compute $\boldsymbol{A^{-1}}$ because $|\boldsymbol{A}|$ appears in the denominator.
- $\mathcal{N}(\boldsymbol{A}) = \boldsymbol{0}$, its null space contains only the zero vector. Otherwise, for some $\boldsymbol{x}$, we have $\boldsymbol{Ax} = \boldsymbol{0}$, and there is no way we can invert it to go from $\boldsymbol{0}$ to $\boldsymbol{x}$.
- $\lambda_i \neq 0$, all its eigenvalues are nonzero. Otherwise, $|\boldsymbol{A}|$, being the product of eigenvalues, would be zero.
- $\lambda_i \neq 0$, all its eigenvalues are nonzero. Another reason, otherwise, for the zero λ, we have $\boldsymbol{Ax} = \boldsymbol{0}$, which implies the existence of a nontrivial null space.

The diagonalizability of $\boldsymbol{A}$ is tested using the invertibility of its eigenvector matrix $\boldsymbol{S}$. Although this point is probably not critical for our view of Linear Algebra as it applies to computer science, we might as well state it here. For a matrix to be non-diagonalizable, the algebraic multiplicity of one of its eigenvalues (the number of times it is repeated) has to be greater than its geometric multiplicity (the number of associated eigenvectors), which means the characteristic polynomial needs to have repeated roots to begin with. The roots are repeated if the discriminant of the polynomial (similar to $b^2 - 4ac$ in the quadratic case) is zero. The discriminant being a continuous function of the coefficients of the polynomial, which are the elements of the matrix, it being zero happens with a frequency of the order of zero. But the roots being complex happens half the time because the discriminant is less than zero half the time.

and the exponentiations of the 100 diagonal elements, a negligible amount. The overall saving in computational time, therefore, is roughly 30.

The best algorithms for matrix multiplications take about $n^{2.3}$ operations. If we are computing the k^{th} power of $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, therefore,

it would cost us $kn^{2.3}$ operations. But with diagonalization, it will cost us n exponentiations of the diagonal elements of $\mathbf{\Lambda}$ and two multiplications, or a total of $n + 2n^{2.3}$ operations, which implies a reduction of $\frac{kn^{2.3}}{n+2n^{2.3}} \approx \frac{k}{2}$ for large n.

Since we have any power of $\boldsymbol{A}$ being expressed as, essentially, powers of $\mathbf{\Lambda}$, we can make the same statement about polynomials of $\boldsymbol{A}$.

11.6.4 Inverse of $\boldsymbol{A}$

Since we have a product for $\boldsymbol{A}$, we can take its inverse using the product rule of inverses.

$$\boldsymbol{A^{-1}} = \left(\boldsymbol{S\Lambda S^{-1}}\right)^{-1} = \left(\boldsymbol{S^{-1}}\right)^{-1} \boldsymbol{\Lambda^{-1} S^{-1}} = \boldsymbol{S\Lambda^{-1} S^{-1}}$$

Since $\mathbf{\Lambda}$ is a diagonal matrix with λ_i along the diagonal, its inverse is another diagonal matrix with the elements equal to the reciprocal of λ_1. Therefore, $\boldsymbol{A^{-1}}$ has the same eigenvectors $\boldsymbol{s_i}$ as $\boldsymbol{A}$, with eigenvalues equal to the reciprocals of the eigenvalues of $\boldsymbol{A}$: $\frac{1}{\lambda_i}$. It can be seen even more directly as follows:

$$\boldsymbol{As_i} = \lambda_i \boldsymbol{s_i} \implies \boldsymbol{s_i} = \lambda_i \boldsymbol{A^{-1} s_i} \implies \boldsymbol{A^{-1} s_i} = \frac{1}{\lambda_i} \boldsymbol{s_i}$$

Enough said.

Since $\boldsymbol{A^{-1}} = \boldsymbol{S\Lambda^{-1} S^{-1}}$, we can see that $\boldsymbol{A}^k = \boldsymbol{S\Lambda}^k \boldsymbol{S^{-1}}$ holds for $k < 0$ as well. Extrapolating even further, through the Taylor series expansion, we can compute entities like $e^{\boldsymbol{A}}$ (matrix exponentiation), which are essential in solving differential equations—a topic we consider beyond the scope of this book.

11.6.5 Difference Equations

In building mathematical models of systems that evolve in time, we may come across the situation where the state of the system at any time step depends on the state at the previous step. If all the parameters specifying the state can be written as a vector, we may be able to write the time evolution as $\boldsymbol{x_{k+1}} = \boldsymbol{Ax_k}$.

If we know the initial conditions at time step zero, we can write:

$$\boldsymbol{x_k} = \boldsymbol{Ax_{k-1}} = \boldsymbol{A^2 x_{k-2}} = \cdots = \boldsymbol{A^k x_0}$$

And, if we have the eigenvalue decomposition of $\boldsymbol{A} = \boldsymbol{S\Lambda S^{-1}}$, we know that we can compute the matrix raised to the power k without worrying too much about the computational cost.

11.6.6 Eigenbasis

If we have a full set of real eigenvectors for $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, we can use them as a basis for $\mathbb{R}^n$, which we will call the eigenbasis. We can then express any vector $\boldsymbol{x_0} \in \mathbb{R}^n$ as a linear combination of the eigenbasis vectors, remembering what we learned about changing the bases of vectors earlier in §7.2 (page 129).

$$\boldsymbol{x_0} = \boldsymbol{s_1}c_1 + \boldsymbol{s_2}c_2 + \cdots + \boldsymbol{s_n}c_n = \sum_{i=1}^{n} \boldsymbol{s_i}c_i = \boldsymbol{Sc}$$

where c_i are the coordinates of $\boldsymbol{x}$ in the eigenbasis. (We wrote the scalar after the vector in the summation so that the matrix product is easier to spot.)

Once we have the vector $\boldsymbol{x_0}$ in the eigenbasis of $\boldsymbol{A}$, we can do simplify the multiplications of the powers of $\boldsymbol{A}$ with $\boldsymbol{x}$ as in the following:

$$\boldsymbol{A}\boldsymbol{x_0} = \boldsymbol{A}\sum_{i=1}^{n} \boldsymbol{s_i}c_i = \sum_{i=1}^{n} \boldsymbol{A}\boldsymbol{s_i}c_i = \sum_{i=1}^{n} \lambda_i \boldsymbol{s_i}c_i = \boldsymbol{S\Lambda c}$$

$$\boldsymbol{A}^k\boldsymbol{x_0} = \sum_{i=1}^{n} \lambda_i^k \boldsymbol{s_i}c_i = \boldsymbol{S\Lambda^k c}$$

Why does this matter? Why use the eigenbasis? Let's think of $\boldsymbol{A}$ as the transformation encoding the time evolution of a system with $\boldsymbol{x_k}$ its state at a given step (or iteration, or a point in time). Given the state of the system at one step, we evolve it to the next step by multiplying with $\boldsymbol{A}$ to get $\boldsymbol{x_{k+1}} = \boldsymbol{A}\boldsymbol{x_k}$.

Knowing the initial state $\boldsymbol{x_0}$ and, more importantly, the transition matrix $\boldsymbol{A}$, what can we say about the stability of the system? We can say the following:

$$\lim_{k\to\infty} \boldsymbol{x_k} = \lim_{k\to\infty} \boldsymbol{A}^k\boldsymbol{x_0} = \sum_{i=1,|\lambda_i|>1}^{n} \lambda_i^k \boldsymbol{s_i}c_i$$

In other words, in the sum that makes up $\boldsymbol{x_k}$, only those eigenvalues (whether they are real or complex) whose absolute value is greater than 1 matter; they are the only ones that will survive when we take the limit $k \to \infty$.

11.7 Fibonacci Numbers

As an example of how this idea of the powers of a matrix applies to a real-world problem, let's look at the Fibonacci numbers. This problem may be academic in nature, but it does show the kind of thinking that goes into transforming a problem to bring it into the domain of eigenvalues.

The famous Fibonacci sequence appears in nature in unexpected ways, and is heavily used in mathematics and, closer to home, in computer science. The Wikipedia entry on it has a comprehensive listing of its properties and interesting facts.

To see how we connect the eigenvalue computation with Fibonacci numbers, let's start by writing them down. The sequence of numbers is: $0, 1, 1, 2, 3, 5, 8, 13, \cdots$: Each number in the sequence (after the first two) is the sum of the previous two. Calling them $f_0, f_1, f_2, \cdots, f_k, \cdots$, we can say that $f_{k+2} = f_{k+1} + f_k$. This is what we might call a second-order difference equation because each number depends on the previous two. We do not see any vector or matrix here, do we? In order to reveal them, let's make a vector out of two Fibonacci numbers:

$$\boldsymbol{x_k} = \begin{bmatrix} f_{k+1} \\ f_k \end{bmatrix} \implies \boldsymbol{x_{k+1}} = \begin{bmatrix} f_{k+2} \\ f_{k+1} \end{bmatrix} = \begin{bmatrix} f_{k+1} + f_k \\ f_{k+1} \end{bmatrix}$$

which gives us a chance to write it as matrix equation, and connect $\boldsymbol{x_k}$ to $\boldsymbol{x_0}$:

$$\boldsymbol{x_{k+1}} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} f_{k+1} \\ f_k \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \boldsymbol{x_k} = \boldsymbol{A x_k} = \boldsymbol{A}^{k+1} \boldsymbol{x_0}$$

From the Fibonacci sequence, we can see that the numbers are growing, and we may want to find out how fast they are growing. Or maybe we want to have an approximation for the k^{th} Fibonacci number. The first question about the growth rate is directly answered by the eigenvalues of $\boldsymbol{A}$, and the second one by the eigenbasis representation of $\boldsymbol{x_k}$.

Let's first compute the eigenvalues.

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \quad \boldsymbol{A} - \lambda \boldsymbol{I} = \begin{bmatrix} 1-\lambda & 1 \\ 1 & -\lambda \end{bmatrix}$$

$$|\boldsymbol{A} - \lambda \boldsymbol{I}| = 0 \implies -(1-\lambda)\lambda - 1 = 0 \text{ or } \lambda^2 - \lambda - 1 = 0$$

$$\lambda_1 = \frac{1+\sqrt{5}}{2} \approx 1.618 \quad \lambda_2 = \frac{1-\sqrt{5}}{2} \approx -0.618$$

Since $|\lambda_1| > 1$ and $|\lambda_2| < 1$, it is the first eigenvalue that will dominate large Fibonacci numbers. In particular, f_{k+1} is going to be about 1.618 times bigger than f_k as $k \to \infty$.

Let's now try to find an equation, a closed-form formula, for f_k. We will start with the eigenvectors, express $\boldsymbol{x_0}$ in the eigenbasis and evolve it to $\boldsymbol{x_k}$. The eigenvectors of $\boldsymbol{A}$ are the solutions to $(\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{s} = 0$.

$$(\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{s} = 0 \implies \begin{bmatrix} 1-\lambda & 1 \\ 1 & -\lambda \end{bmatrix} \boldsymbol{s} = 0$$

$$\implies \boldsymbol{s} = \begin{bmatrix} \lambda \\ 1 \end{bmatrix} \text{ for } \lambda = \lambda_1, \lambda_2$$

$\boldsymbol{x_0}$ is a linear combination of $\boldsymbol{s_1}$ and $\boldsymbol{s_2}$.

$$\boldsymbol{x_0} = c_1 \boldsymbol{s_1} + c_2 \boldsymbol{s_2} \implies \begin{bmatrix} 1 \\ 0 \end{bmatrix} = c_1 \begin{bmatrix} \lambda_1 \\ 1 \end{bmatrix} + c_2 \begin{bmatrix} \lambda_2 \\ 1 \end{bmatrix}$$

After a bit of algebra, we will get:

$$c_1 = \frac{1}{\sqrt{5}}, c_1 = -\frac{1}{\sqrt{5}}$$

We know how the evolution of $\boldsymbol{x}$:

$$\boldsymbol{x_k} = \boldsymbol{A}^k \boldsymbol{x_0} = \boldsymbol{A}^k (c_1 \boldsymbol{s_1} + c_2 \boldsymbol{s_2}) = c_1 \lambda_1^k \boldsymbol{s_1} + c_2 \lambda_2^k \boldsymbol{s_2} = \begin{bmatrix} f_{k+1} \\ f_k \end{bmatrix}$$

We can now read the second element in $\boldsymbol{x_k}$ as f_k:

$$\begin{bmatrix} f_{k+1} \\ f_k \end{bmatrix} = c_1 \lambda_1^k \begin{bmatrix} \lambda_1 \\ 1 \end{bmatrix} + c_2 \lambda_2^k \begin{bmatrix} \lambda_2 \\ 1 \end{bmatrix} \implies f_k = c_1 \lambda_1^k + c_2 \lambda_2^k$$

Knowing that the second term vanishes for large k (because $|\lambda_2| = 0.618 < 1$) , we finally get an expression for f_k:

$$f_k \approx c_1 \lambda_1^k = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^k = f_k^{(\text{approx})}$$

How good this approximation is is shown in Table 11.1, which shows that the approximation is stunningly accurate. By the time we reach $k = 11$, the error is about 25 parts in a million.

Table 11.1 Fibonacci numbers (f_k) vs. its approximation ($f_k^{(\text{approx})}$)

k	f_k	$f_k^{(\text{approx})}$	k	f_k	$f_k^{(\text{approx})}$	k	f_k	$f_k^{(\text{approx})}$
0	0	0.45	4	3	3.07	8	21	21.01
1	1	0.72	5	5	4.96	9	34	33.99
2	1	1.17	6	8	8.02	10	55	55.00
3	2	1.89	7	13	12.98	11	89	89.00

11.8 Applications of Eigenvalues and Eigenvectors

The ideas behind eigenvalue decomposition has a multitude of applications, especially in physics and other physical sciences. In our domain of computer science, the Google Page Rank algorithm, described in its own box in the next chapter, is a brilliant success story of this line of thinking. Since this chapter is in the advanced part of this book, we do not list the applications here, but rather note that a good starting point to explore would be the Wikipedia page.

Recap: The Story So Far

Let's think the unthinkable, let's do the undoable. Let us prepare to grapple with the ineffable itself, and see if we may not eff it after all.

—Douglas Adams

As we saw multiple times, everything in Linear Algebra is connected to everything else. It is a big and beautiful jigsaw puzzle. Although we chose to learn it in the particular sequence that we did, we could have started our exploration from any one of its corner pieces. Let's look at what we have learned, this time from the perspective of the interplay between the ranks and shapes of matrices and the four fundamental spaces. As we shall see, this summary will also reveal more about the process of solving equations and projecting onto column (or even row) spaces.

Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, we have a mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$, as shown in Figure R.1, where the input space is shown in green and the output space in red. The equation $\boldsymbol{Ax} = \boldsymbol{b}$ expresses that $\boldsymbol{b}$ is a linear combination of the columns of $\boldsymbol{A}$. All such linear combinations lie in $\mathcal{C}(\boldsymbol{A})$, which is a subspace of $\mathbb{R}^m$, the output space shown in red.

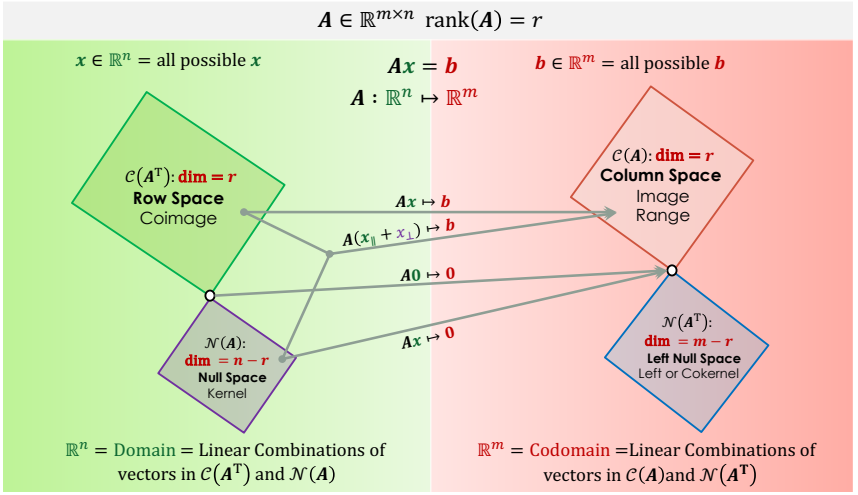


Fig. R.1 Recap of the four fundamental spaced defined by a matrix.

It is true that multiple vectors $\boldsymbol{x} \in \mathbb{R}^n$ can map to the same vector $\boldsymbol{b} \in \mathcal{C}(\boldsymbol{A})$. However, the mapping from the row space $\mathcal{C}(\boldsymbol{A}^\mathbf{T})$ to the column space $\mathcal{C}(\boldsymbol{A})$ is one-to-one. In other words, for any vector $\boldsymbol{b} \in \mathcal{C}(\boldsymbol{A})$, there is one and only one vector $\boldsymbol{x} \in \mathcal{C}(\boldsymbol{A}^\mathbf{T})$ such that $\boldsymbol{Ax} = \boldsymbol{b}$. Why is that? Let's prove it, somewhat formally.

Proof: To establish this result, we assume the contrary and derive a contradiction. Suppose there exist two distinct nonzero vectors $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{C}(\boldsymbol{A}^\mathbf{T})$, with $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$, such that $\boldsymbol{Ax}_1 = \boldsymbol{b}$ and $\boldsymbol{Ax}_2 = \boldsymbol{b}$.

(1) Assumption:	$\boldsymbol{Ax}_1 = \boldsymbol{b}$,	$\boldsymbol{Ax}_1 = \boldsymbol{b}$
(2) Subtracting, we get:	$\boldsymbol{A}(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \mathbf{0}$	
(3) $\boldsymbol{A}(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \mathbf{0} \implies$	$\boldsymbol{x}_1 - \boldsymbol{x}_2$	$\in \mathcal{N}(\boldsymbol{A})$
(4) $\boldsymbol{x}_1 \neq \boldsymbol{x}_2 \implies$	$\boldsymbol{x}_1 - \boldsymbol{x}_2$	$\neq \mathbf{0}$
(5) Closure property of $\mathcal{C}(\boldsymbol{A}^\mathbf{T})$:		
$\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{C}(\boldsymbol{A}^\mathbf{T}) \implies$	$\boldsymbol{x}_1 - \boldsymbol{x}_2$	$\in \mathcal{C}(\boldsymbol{A}^\mathbf{T})$
(6) $\boldsymbol{x}_1 - \boldsymbol{x}_2 \in \mathcal{C}(\boldsymbol{A}^\mathbf{T})$ and		
$\boldsymbol{x}_1 - \boldsymbol{x}_2 \in \mathcal{N}(\boldsymbol{A}) \implies$	$\boldsymbol{x}_1 - \boldsymbol{x}_2$	$\in \mathcal{C}(\boldsymbol{A}^\mathbf{T}) \cap \mathcal{N}(\boldsymbol{A})$
(7) $\mathcal{C}(\boldsymbol{A}^\mathbf{T}) \perp \mathcal{N}(\boldsymbol{A}) \implies$	$\boldsymbol{x}_1 - \boldsymbol{x}_2$	$= \mathbf{0}$

Statement (7) above tells us that since $\mathcal{C}(\boldsymbol{A}^\mathbf{T})$ and $\mathcal{N}(\boldsymbol{A})$ are orthogonal to each other, the only vector they have in common is the

zero vector. But from statement (4), we assumed the existence of two distinct vectors in $\mathcal{C}(\boldsymbol{A}^{\mathbf{T}})$ that map to the same $\boldsymbol{b}$, implying their difference lies in both $\mathcal{C}(\boldsymbol{A}^{\mathbf{T}})$ and $\mathcal{N}(\boldsymbol{A})$. These two statements cannot both be true. Therefore, for every $\boldsymbol{b} \in \mathcal{C}(\boldsymbol{A})$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, there is one and only one vector $\boldsymbol{x} \in \mathcal{C}(\boldsymbol{A}^{\mathbf{T}})$ that satisfies the equation, which is another way of saying that the mapping $\mathcal{C}(\boldsymbol{A}^{\mathbf{T}}) \mapsto \mathcal{C}(\boldsymbol{A})$ is one-to-one.

Intuitively, since we have r linearly independent columns (the pivot columns) that span $\mathcal{C}(\boldsymbol{A})$, any one nonzero linear combination

$$\boldsymbol{b} = \sum_{i=1}^{r} x_i \boldsymbol{a_i}$$

should have a unique set of coefficients x_i. If we place these coefficients to form a vector $\boldsymbol{x}$, this vector has to be in the row space: $\boldsymbol{x} \in \mathcal{C}(\boldsymbol{A}^{\mathbf{T}})$. If not, where in the input space could the vector $\boldsymbol{x}$ be?

R.1 Full-Rank Square Matrices

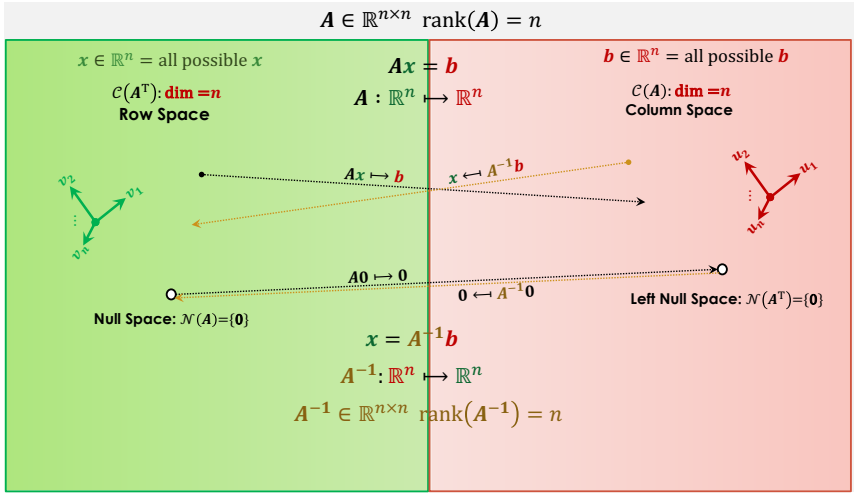


Fig. R.2 Recap of the four fundamental spaced defined by a full-rank, square matrix. Notice that the row and column spaces are, in fact, all of the input and output spaces respectively: $\mathcal{C}(\boldsymbol{A}^{\mathbf{T}}) \equiv \mathbb{R}^n$ and $\mathcal{C}(\boldsymbol{A}^{\mathbf{T}}) \equiv \mathbb{R}^n$. Also, $\boldsymbol{A}^{-1}$ is a mapping from $\mathcal{C}(\boldsymbol{A})$ back to $\mathcal{C}(\boldsymbol{A}^{\mathbf{T}})$.

When we have a full-rank square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, $\text{rank}(\boldsymbol{A}) = n$, the row space is all of $\mathbb{R}^n$. So is the column space. The mapping $\boldsymbol{A} : \mathbb{R}^n \mapsto \mathbb{R}^n$, which is the same as $\boldsymbol{A} : \mathcal{C}(\boldsymbol{A}^\mathsf{T}) \mapsto \mathcal{C}(\boldsymbol{A})$, is one-to-one. If we are given any $\boldsymbol{b}$ in $\boldsymbol{Ax} = \boldsymbol{b}$, we can always find the corresponding $\boldsymbol{x}$ by the inverse mapping $\boldsymbol{A^{-1}}$, which is guaranteed to exist. In terms of the four fundamental spaces, the picture looks like what is depicted in Figure R.2. Note that both $\mathcal{N}(\boldsymbol{A})$ and $\mathcal{N}(\boldsymbol{A}^\mathsf{T})$ contain only the zero vector.

Solving $\boldsymbol{Ax} = \boldsymbol{b}$ is the same as finding the inverse $\boldsymbol{A}^{-1}$, either of which can be done using Gauss-Jordan elimination. This is what people typically have in mind when they say, "n equations in n unknowns means you can solve." As we shall see again soon, they should really be saying is, "n independent and consistent equations in n unknowns means we can find a unique solution."

Orthonormal Bases Also shown in Figure R.2 are the basis vectors for the input and output spaces, $\boldsymbol{v_i}$ and $\boldsymbol{u_i}$. Since $\boldsymbol{A}$ is full rank, the column vectors will suffice as the basis for $\mathcal{C}(\boldsymbol{A})$ (and the columns of $\boldsymbol{A}^\mathsf{T}$ for $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$). If we would like them to be normalized, we know how to do it. And, if we would like them to be orthogonal as well, we could perform the Gram-Schmidt process. Let's say the set $\{\boldsymbol{v_i}\}$ is, in fact, such an orthonormal basis. By multiplying them with $\boldsymbol{A}$, we can get some vectors in the column space; but there is no guarantee that we would get $\{\boldsymbol{u_i}\}$. Is there a special basis $\{\boldsymbol{v_i}\}$ such that $\{\boldsymbol{Av_i}\} = \{\sigma_i \boldsymbol{u_i}\}$ (where σ_i is some scaling factor)?

Since the row and column spaces are both $\mathbb{R}^n$, we might as well use the columns of $\boldsymbol{I}_n$ as their basis. If we take $\boldsymbol{v_i} = \boldsymbol{i_i}$, what do we get upon transforming them using $\boldsymbol{A}$? We get the columns of $\boldsymbol{A}$, which are pretty far from an orthonormal basis. But we do know that if $\boldsymbol{A}$ is diagonalizable, and if its eigenvectors are real and orthogonal, we could use them as the basis, giving us $\boldsymbol{Av_i} = \lambda_i \boldsymbol{v_i}$, which are scaled versions of $\boldsymbol{v_i}$. And, when can we guarantee that the eigenvectors are real and orthogonal? When $\boldsymbol{A}$ is real and symmetric. Using a different symbol (σ_i instead of λ_i and with $\boldsymbol{u_i} = \boldsymbol{v_i}$), we can write:

$$\boldsymbol{Av_i} = \sigma_i \boldsymbol{v_i} \implies \{\boldsymbol{Av_i}\} = \{\sigma_i \boldsymbol{u_i}\} \text{ or } \boldsymbol{AV} = \boldsymbol{\Sigma U}$$

where $\boldsymbol{V}$ and $\boldsymbol{U}$ are orthonormal basis matrices with $\boldsymbol{v_i}$ and $\boldsymbol{u_i}$ as their columns.

R.2 Full-Column-Rank, Tall Matrices

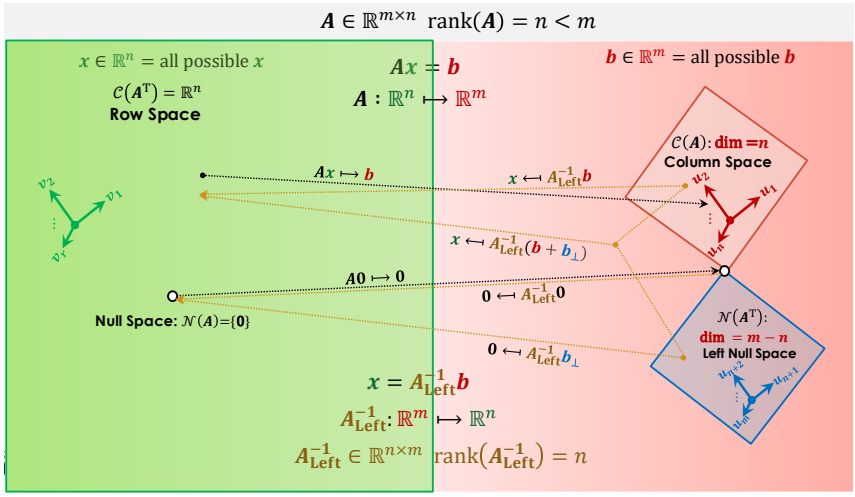


Fig. R.3 Recap of the four fundamental spaced defined by a full-column-rank, tall matrix. Here, the row space is all of the input space: $\mathcal{C}(\boldsymbol{A}^\mathsf{T}) \equiv \mathbb{R}^n$, while the column space is a proper subspace of the output space: $\mathcal{C}(\boldsymbol{A}) \subset \mathbb{R}^m$. $\boldsymbol{A}_{\text{Left}}^{-1}$ is the one-to-one mapping $\mathcal{C}(\boldsymbol{A}) \mapsto \mathcal{C}(\boldsymbol{A}^\mathsf{T})$.

Figure R.3 shows the four fundamental subspaces of a full-column-rank, tall matrix. Notice that the null space $\mathcal{N}(\boldsymbol{A})$ is trivial (meaning it contains only the zero vector), and the row space is all of the input space: $\mathcal{C}(\boldsymbol{A}^\mathsf{T}) = \mathbb{R}^n$. What this means is that any vector in the input space will be mapped to a unique vector in $\mathcal{C}(\boldsymbol{A})$. Why unique? Because in $\boldsymbol{Ax} = \boldsymbol{b}$, $\boldsymbol{b} \in \mathbb{R}^m$ is a linear combination of the columns of $\boldsymbol{A}$, and, as we saw earlier, any linear combination of linearly independent vectors (with a given set of coefficients) is unique. In other words, given a set of n linearly independent vectors $\boldsymbol{a_i}$ and scalars x_i, $\sum x_i \boldsymbol{a_i}$ is a unique vector. And, of course, this sum is exactly what $\boldsymbol{Ax}$ is.

The column space $\mathcal{C}(\boldsymbol{A})$, however, does not cover all of the output space $\mathbb{R}^m$. Therefore, if we take a vector $\boldsymbol{b} \in \mathbb{R}^m$ such that it is not in the column space ($\boldsymbol{b} \notin \mathcal{C}(\boldsymbol{A})$), it is not a linear combination of the columns of $\boldsymbol{A}$, and there is no solution for $\boldsymbol{x}$ such that $\boldsymbol{Ax} = \boldsymbol{b}$. The system of linear equations is inconsistent. If we perform Gaussian elimination on the augmented matrix $\left[\boldsymbol{A} \mid \boldsymbol{b}\right]$, we will get at least one

row of the form zero equals nonzero, indicating this fact. In other words, for $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, with a general $\boldsymbol{b} \in \mathbb{R}^m$: No solution if $\boldsymbol{b} \notin \mathcal{C}(\boldsymbol{A})$.

We also saw that we can get to the best approximation to the solution, $\hat{\boldsymbol{x}}$, by projecting $\boldsymbol{b}$ onto $\mathcal{C}(\boldsymbol{A})$ as $\hat{\boldsymbol{b}}$, in which case $\boldsymbol{A}\hat{\boldsymbol{x}} = \hat{\boldsymbol{b}}$ has a unique solution because $\hat{\boldsymbol{b}} \in \mathcal{C}(\boldsymbol{A})$ by construction (projection). This process of projection is considered least squares minimization because $\hat{\boldsymbol{b}}$ is the vector in $\mathcal{C}(\boldsymbol{A})$ "closest" to $\boldsymbol{b}$. In other words, the error vector $\boldsymbol{b} - \hat{\boldsymbol{b}}$ has the smallest Euclidean norm.

We now have a recipe for finding the best approximation: Project $\boldsymbol{b}$ onto $\mathcal{C}(\boldsymbol{A})$ as $\hat{\boldsymbol{b}} = \boldsymbol{A}\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{b}$ and solve $\boldsymbol{A}\hat{\boldsymbol{x}} = \hat{\boldsymbol{b}}$, and we get

$$\boldsymbol{A}\hat{\boldsymbol{x}} = \hat{\boldsymbol{b}} \implies \boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{A}\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{b}.$$

Notice that this last equation states that a linear combination of the columns of $\boldsymbol{A}$ (weighted by the elements of $\hat{\boldsymbol{x}}$) is identical to another linear combination (weighted by the elements of $\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{b}$) of the same set of vectors (*i.e.*, columns of $\boldsymbol{A}$). Since we know that the linear combinations of linearly independent vectors are unique (which is to say, if a vector is the linear combination of some linearly independent vectors, we cannot have two different sets of weights in the linear combination), we conclude:

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{b}$$

We also saw in our earlier discussion that even when $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is not solvable (with a full-column-rank, tall $\boldsymbol{A}$), the equation $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{A}^{\mathsf{T}}\boldsymbol{b}$ has solutions, which also gives

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{b}$$

Finally, remember the left inverse, which is defined for a full-column-rank, tall matrix? $\boldsymbol{A}_{\text{Left}}^{-1} = (\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathsf{T}}$ with $\boldsymbol{A}_{\text{Left}}^{-1}\boldsymbol{A} = \boldsymbol{I}$. Therefore, when we have $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ and $\boldsymbol{A}$ is a full-column-rank, tall matrix, we can write

$$\boldsymbol{A}_{\text{Left}}^{-1}\boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{A}_{\text{Left}}^{-1}\boldsymbol{b} \implies \hat{\boldsymbol{x}} = \boldsymbol{A}_{\text{Left}}^{-1}\boldsymbol{b} = \hat{\boldsymbol{x}} = \left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{b}$$

Why do we put a hat on $\boldsymbol{x}$ in the equation above? Because we know that the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ does not have solutions, in general, unless $\boldsymbol{b} \in \mathcal{C}(\boldsymbol{A})$. What we are getting is the best approximation to the

solution, which is indeed the same as projecting $\boldsymbol{b}$ onto $\mathcal{C}(\boldsymbol{A})$ and solving. It is also the same as the least squares solution.

In summary, $\boldsymbol{A} \in \mathbb{R}^{m \times n}, \text{rank}(\boldsymbol{A}) = n < m \implies |\boldsymbol{A}|$ and $\boldsymbol{A}^{-1}$ are not defined, and we do not have (in general) solutions to $\boldsymbol{Ax} = \boldsymbol{b}$. We do have a least-squares solution, which is the best approximation, and it can be arrived at in a variety of ways, all of which give the same answer. Although unrelated to solving the system of equations, keep in mind that an eigenanalysis is not possible because the matrix $\boldsymbol{A}$ is not a square one.

R.3 Full-Row-Rank, Wide Matrices

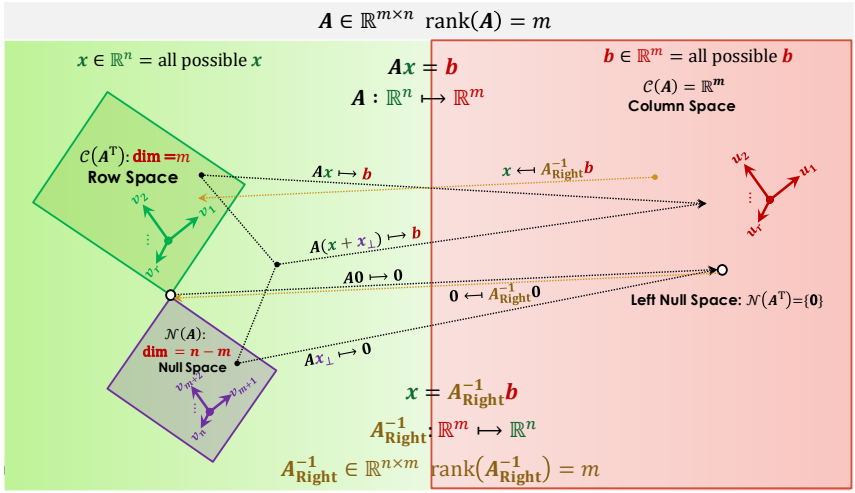


Fig. R.4 Recap of the four fundamental spaced defined by a full-row-rank, wide matrix, showing $\boldsymbol{A}_{\text{Right}}^{-1} : \mathcal{C}(\boldsymbol{A}) \mapsto \mathcal{C}(\boldsymbol{A}^{\mathbf{T}})$. For this matrix, it is the column space that encompasses all of the output space: $\mathcal{C}(\boldsymbol{A}) \equiv \mathbb{R}^m$.

A full-row-rank, wide matrix is indeed the transpose of a full-column-rank, tall matrix. In terms of the four fundamental spaces, as shown in Figure R.4, it is roughly equivalent to swapping the input and output spaces. We might expect to see the right inverse in place of the left inverse, as the mapping from the column space back to the row space. Let's see how it comes about.

For a full-row-rank, wide matrix, its column space is all of the output space: $\mathcal{C}(\boldsymbol{A}) = \mathbb{R}^m$. (See Figure R.4.) The row space is a

subspace of the input space: $\mathcal{C}(\boldsymbol{A}^{\mathsf{T}}) \subset \mathbb{R}^n$ and there is a null space $\mathcal{N}(\boldsymbol{A})$, which contains all vectors that are orthogonal to those in the row space.

The mapping $\boldsymbol{A} : \mathbb{R}^n \mapsto \mathbb{R}^m$ is many-to-one: multiple vectors in the input space map to the same output vector. However, the mapping from the row space to the column space, $\boldsymbol{A} : \mathcal{C}(\boldsymbol{A}^{\mathsf{T}}) \mapsto \mathcal{C}(\boldsymbol{A})$, is still one-to-one, for the same reasons of uniqueness of linear combinations we expounded on in the previous case. In other words, given any vector $\boldsymbol{b} \in \mathcal{C}(\boldsymbol{A})$ (which is the same as saying $\boldsymbol{b} \in \mathbb{R}^m$ because $\mathcal{C}(\boldsymbol{A}) = \mathbb{R}^m$), there is a unique vector $\boldsymbol{x} \in \mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$.

R.3.1 Why Many-to-One?

We know the answer already: We do not have enough equations to constrain the system to a single solution. If we have one equation ($a_{11}x + a_{12}y = b_1$) in two variables, we have a line and any point (x, y) in the line will map to the same b_1.

We can also see this geometrically. Let's take a vector $\boldsymbol{x} \in \mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$, which maps to $\boldsymbol{b} \in \mathcal{C}(\boldsymbol{A})$ through $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. Now take another vector in the input space that is in the null space, $\mathcal{N}(\boldsymbol{A})$. Remembering that the null space is the orthogonal complement of the row space, we call this vector $\boldsymbol{x}_\perp \in \mathcal{N}(\boldsymbol{A})$. We know, by the definition of $\mathcal{N}(\boldsymbol{A})$ as the solution set of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$, $\boldsymbol{A}\boldsymbol{x}_\perp = \boldsymbol{0}$. Therefore, $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{A}\boldsymbol{x}_\perp = \boldsymbol{b} + \boldsymbol{0} = \boldsymbol{b}$ or, $\boldsymbol{A}(\boldsymbol{x} + \boldsymbol{x}_\perp) = \boldsymbol{b}$. In other words, the moment we have one vector $\boldsymbol{x} \in \mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, we can add any vector $\boldsymbol{x}_\perp \in \mathcal{N}(\boldsymbol{A})$ to it and the sum $\boldsymbol{x} + \boldsymbol{x}_\perp$ will still map to the same $\boldsymbol{b}$.

One point to note is that the sum $\boldsymbol{x} + \boldsymbol{x}_\perp$ will always have a norm greater than or equal to that of $\boldsymbol{x}$: $\|\boldsymbol{x} + \boldsymbol{x}_\perp\| \geq \|\boldsymbol{x}\|$ because $\boldsymbol{x}$ and $\boldsymbol{x}_\perp$ for two sides of a right-angled triangle and the sum $\boldsymbol{x} + \boldsymbol{x}_\perp$ is the hypotenuse. In other words, the unique solution $\boldsymbol{x}$ is the *minimum-norm solution* of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$.

R.3.2 How to Find the Unique Part?

We actually know the answer to this one as well. When we found the complete solution of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where we have free variables, we wrote it as $\boldsymbol{x}_p + t_1\boldsymbol{x}_{s1} + t_2\boldsymbol{x}_{s2} + \cdots$, where $\boldsymbol{x}_p$ was the particular solution and $\boldsymbol{x}_{si}$ were the special solutions (of which we had as many as the number of free variables). Now we can see that $\boldsymbol{x}_p$ is, in

fact, the part of the solution that lies in the row space. Any linear combination of the $n - r$ special solutions (which form a basis for the null space) would indeed be in the null space, and that is what is denoted as $\boldsymbol{x}_\perp$ in Figure R.4. We can see how all these different views are coming together beautifully, can't we?

One way of finding the particular solution is to perform Gaussian Elimination on the augmented matrix to locate the pivotless columns, which point to the free variables. We then solve the system after setting the free variables to zero. Now we have r equations and r unknowns, because we have set the $n - r$ free variables to zero. The unique solution to this system is the particular solution $\boldsymbol{x}_p$.

Thinking geometrically, we first notice that the complete solution is the sum $\boldsymbol{x} + \boldsymbol{x}_\perp$. If we project this sum onto the row space $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$, it becomes just $\boldsymbol{x}$, because $\boldsymbol{x}_\perp$ is orthogonal to $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$. When we did the projection, we wrote the matrix that projects onto $\mathcal{C}(\boldsymbol{A})$ as $\boldsymbol{P} = \boldsymbol{A}\left(\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^\mathsf{T}$. The row space $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$ is, as the symbol indicates, the column space of $\boldsymbol{A}^\mathsf{T}$ and the matrix that would project onto it, $\boldsymbol{P}_{\text{RS}}$ would be just $\boldsymbol{P}$, but with $\boldsymbol{A}$ replaced by $\boldsymbol{A}^\mathsf{T}$. Thus, $\boldsymbol{P}_{\text{RS}} = \boldsymbol{A}^\mathsf{T}\left(\boldsymbol{A}\boldsymbol{A}^\mathsf{T}\right)^{-1}\boldsymbol{A}$ (where we also made use of the fact that the transpose of $\boldsymbol{A}^\mathsf{T}$ is $\boldsymbol{A}$).

Let's say we found a solution $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{x}_\perp$ such that $\boldsymbol{A}\boldsymbol{x}' = \boldsymbol{b}$. With $\boldsymbol{x} \in \mathcal{C}(\boldsymbol{A}^\mathsf{T})$ and $\boldsymbol{x}_\perp \in \mathcal{N}(\boldsymbol{A})$, we can see that $\boldsymbol{P}_{\text{RS}}\boldsymbol{x} = \boldsymbol{x}$ and $\boldsymbol{P}_{\text{RS}}\boldsymbol{x}_\perp = \boldsymbol{0}$. Then $\boldsymbol{x} = \boldsymbol{P}_{\text{RS}}\boldsymbol{x}' = \boldsymbol{A}^\mathsf{T}\left(\boldsymbol{A}\boldsymbol{A}^\mathsf{T}\right)^{-1}\boldsymbol{A}\boldsymbol{x}'$. But $\boldsymbol{A}\boldsymbol{x}' = \boldsymbol{b}$, which means we get the minimum-norm solution, the part of $\boldsymbol{x}'$ in $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$, $\boldsymbol{x} = \boldsymbol{A}^\mathsf{T}\left(\boldsymbol{A}\boldsymbol{A}^\mathsf{T}\right)^{-1}\boldsymbol{b}$.

Let's take a step back and remind ourselves what the right inverse is. $\boldsymbol{A}$ is a full-row-rank, wide matrix. Therefore, $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$ is full rank and invertible, which means $\boldsymbol{A}\boldsymbol{A}^\mathsf{T}\left(\boldsymbol{A}\boldsymbol{A}^\mathsf{T}\right)^{-1} = \boldsymbol{I} \implies \boldsymbol{A}_{\text{Right}}^{-1} = \boldsymbol{A}^\mathsf{T}\left(\boldsymbol{A}\boldsymbol{A}^\mathsf{T}\right)^{-1}$. Now we see that $\boldsymbol{A}_{\text{Right}}^{-1}$ is the mapping that takes any vector $\boldsymbol{b} \in \mathcal{C}(\boldsymbol{A})$ and gives us the unique vector $\boldsymbol{x} \in \mathcal{C}(\boldsymbol{A}^\mathsf{T})$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. This is shown in Figure R.4.

Notice another interesting fact: The projection matrix, the one that projects to $\mathcal{C}(\boldsymbol{A})$ is:

$$\boldsymbol{P} = \boldsymbol{A}\left(\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^\mathsf{T} = \boldsymbol{A}\boldsymbol{A}_{\text{Left}}^{-1}$$

and the one that projects to $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$ is:

$$\boldsymbol{P}_{\text{RS}} = \boldsymbol{A}^\mathsf{T}\left(\boldsymbol{A}\boldsymbol{A}^\mathsf{T}\right)^{-1}\boldsymbol{A} = \boldsymbol{A}_{\text{Right}}^{-1}\boldsymbol{A}$$

Beautifully symmetric, isn't it?

As we went through these cases of full-rank matrices, square, tall or wide, we saw how solving the system of linear equations using Gaussian Elimination is related to the picture of the four fundamental subspaces, and how the mapping between the row space and the column space was one-to-one and could be inverted. We further saw that the left and right inverses were in fact these mappings from $\mathcal{C}(\boldsymbol{A})$ to $\mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$. In the case of a full-rank, square matrix, they indeed reduce to the double-sided inverse, $\boldsymbol{A}^{-1}$.

In summary, here's what we've seen in the wide matrix case:

- A full-row-rank, wide matrix does not have a standard inverse or admit eigenvalue decomposition. The rank is equal to the number of rows, which is less than the number of columns.
- For any right-hand side vector, the system is always consistent because the column space spans the entire output space.
- The minimum-norm solution is the best choice and can be found by projecting any solution onto the row space.
- Any solution to the system can be written as the sum of a unique particular solution (lying in the row space) and a null-space component (which accounts for the degrees of freedom).
- The particular solution is obtained by setting all free variables to zero and lies entirely in the row space.
- The projection matrix that gives the row-space component is the familiar form, but with the original matrix replaced by its transpose.
- This projection uses the right inverse, and it annihilates the null-space component, since the row space and null space are orthogonal.

R.4 Any General Matrix

We have not yet dealt with rank-deficient matrices. Even if the matrix is rank deficient, the mapping between the row space and the column space is one-to-one and invertible: If $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, then the mapping

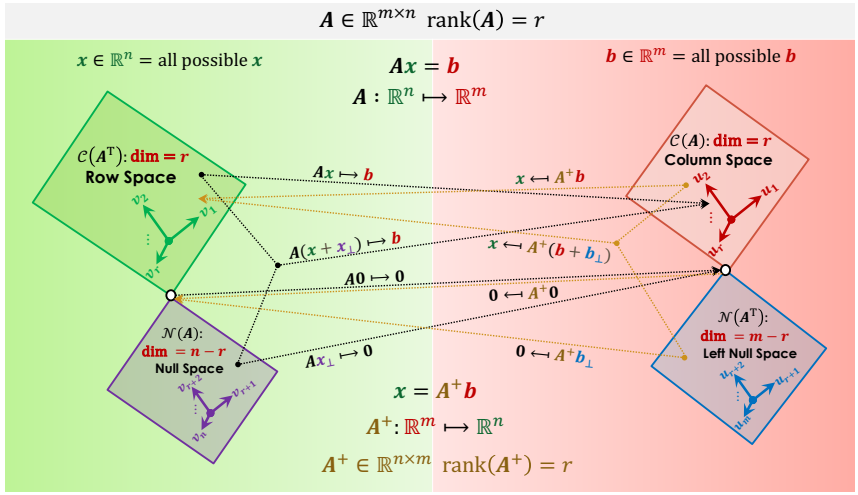


Fig. R.5 The four fundamental spaced defined by a general rank-deficient matrix. Here, the newly defined pseudo-inverse maps $\mathcal{C}(\boldsymbol{A})$ back to $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$.

$\boldsymbol{A} : \mathcal{C}(\boldsymbol{A}^\mathsf{T}) \mapsto \mathcal{C}(\boldsymbol{A})$ is invertible such that there is some pseudo-inverse $\boldsymbol{A}^+ : \mathcal{C}(\boldsymbol{A}) \mapsto \mathcal{C}(\boldsymbol{A}^\mathsf{T})$. But we do not yet have all the tools necessary to unearth this elusive inverse yet. We need to learn Singular Value Decomposition, which we will go through in the very last chapter of this book. For now, let's do a quick preview, which may be useful.

In the previous chapter, we learned that eigenanalysis is possible only for square matrices. It works best for symmetric matrices that are diagonalizable. The "thin" version of Singular Value Decomposition, which is closely related to the eigenanalysis of $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$ or $\boldsymbol{A}\boldsymbol{A}^\mathsf{T}$, works for all matrices because these products are symmetric. What it gives is, in fact, a decomposition of the form:

$$\boldsymbol{A} \in \mathbb{R}^{m\times n} \xrightarrow{\text{SVD}} \boldsymbol{A} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^\mathsf{T}$$
$$\hat{\boldsymbol{U}} \in \mathbb{R}^{m\times r},\ \hat{\boldsymbol{\Sigma}} \in \mathbb{R}^{r\times r},\ \hat{\boldsymbol{V}} \in \mathbb{R}^{r\times n}$$

$\hat{\boldsymbol{\Sigma}}$ is a diagonal matrix, with positive values along the main diagonal, starting from σ_1 all the way to σ_r, where $r = \text{rank}(\boldsymbol{A})$, with the rest of the elements all zero.

In terms of the four fundamental spaces, what SVD is telling us is that there is a special orthonormal basis in the row space, $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_r\}$, that maps to an orthonormal basis in the column

space $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_r\}$. The very fact that there are such bases is already remarkable. The fact that these bases can be unearthed is even more so. We will later see how the basis matrices $\hat{\boldsymbol{U}}$ and $\hat{\boldsymbol{V}}$, as well as the matrix of singular values $\hat{\boldsymbol{\Sigma}}$, come into play. For our purposes here, let's note that following:

$$\hat{\boldsymbol{\Sigma}} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \end{bmatrix} \implies \hat{\boldsymbol{\Sigma}}^{-1} = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma_2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sigma_r} \end{bmatrix}$$

With this, we now define the pseudo-inverse $\boldsymbol{A}^+ = \hat{\boldsymbol{V}}\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{U}}^{\mathsf{T}}$. Note that $\boldsymbol{A}^+ \in \mathbb{R}^{n \times m}$.

If we knew that $\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A} = \boldsymbol{A}$, we could do the following:

$$\begin{aligned} \boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A}\boldsymbol{x} &= \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ \boldsymbol{A}\boldsymbol{A}^+(\boldsymbol{A}\boldsymbol{x}) &= \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ \boldsymbol{A}\boldsymbol{A}^+\boldsymbol{b} &= \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \end{aligned}$$

Since we already have $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, we can say (using the arguments about the uniqueness of linear combinations of linearly independent vectors again) that $\boldsymbol{A}^+\boldsymbol{b} = \boldsymbol{x}$, which is the inverse transformation from $\mathcal{C}(\boldsymbol{A})$ to $\mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$. We have to verify that $\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A} = \boldsymbol{A}$, which is a property of the pseudo-inverse. Furthermore, $\boldsymbol{A}^+$ reduces to $\boldsymbol{A}^{-1}$, $\boldsymbol{A}_{\textbf{Left}}^{-1}$ or $\boldsymbol{A}_{\textbf{Right}}^{-1}$ for full-rank square, full-column-rank and full-row-rank matrices respectively. As we defined it here, we do not have enough knowledge to prove it yet, but we will prove it in the last chapter, considering the full SVD (rather than the "thin" version we used in our definition of the pseudo-inverse $\boldsymbol{A}^+$ above).

The only thing left to do now is to look at the picture in Figure R.5 and marvel at its beauty, elegance, and completeness. In some sense, we have come full circle. We can now appreciate how the algebraic notion of solving equations (using Gauss-Jordan elimination, for instance) is related to the geometric view of four fundamental spaces, and the more abstract ideas of mappings and their inversions.

12

Special Matrices, Similarity, and Algorithms

The surest way to corrupt a youth is to instruct him to hold in higher esteem those who think alike than those who think differently.

—Friedrich Nietzsche

In the last chapter, we saw how the properties of eigenvalues and eigenvectors are related to the characteristics of the matrices. We looked at real, symmetric matrices and studied their eigenanalysis to some extent. In this chapter, we will look at some other special matrices and their eigenvalues and vectors. Let's start with another look at symmetric matrices.

12.1 Real, Symmetric Matrices

Real, symmetric matrices ($\boldsymbol{A} \in \mathbb{R}^{n\times n}, \boldsymbol{A} = \boldsymbol{A}^{\mathsf{T}}$) have real eigenvalues and orthogonal eigenvectors, as we proved in some painstaking detail in the previous chapter (Item §3, page 199). Here are two theorems on such matrices.

12.1.1 Spectral Theorem

As we (almost) saw, a real symmetric matrix has real, orthonormal (which people may call orthogonal at times) eigenvectors (restricting ourselves to $\mathbb{R}$ again). Using the symbol $\boldsymbol{Q}$ for orthonormal matrices, and knowing that they are invertible, with the inverse being the transpose, we can state our eigenvalue decomposition as:

$$\boldsymbol{A} = \boldsymbol{S\Lambda S^{-1}} \iff \boldsymbol{A} = \boldsymbol{Q\Lambda Q^{-1}} = \boldsymbol{Q\Lambda Q^{\mathsf{T}}}$$

Knowing that $\boldsymbol{Q}$ is a matrix with columns $\boldsymbol{q_i}$ (which we may write as $\boldsymbol{Q} = [\boldsymbol{q_i}]$), we can expand the product to read:

$$\boldsymbol{A} = \sum_{1}^{n} \lambda_i \boldsymbol{q_i q_i^{\mathsf{T}}} \tag{12.1}$$

This expansion is possible because $\boldsymbol{\Lambda}$ has only diagonal entries λ_i and $\boldsymbol{q_i} \perp \boldsymbol{q_j} \implies \boldsymbol{q_i^{\mathsf{T}} q_j} = 0$ if $i \neq j$. Looking at each term in the expansion, we can make the following remarks:

- Each term has a projection matrix $\boldsymbol{q_i q_i^{\mathsf{T}}}$.
- It is a projection to the eigenspace of the i^{th} eigenvector.
- Each eigenspace is one-dimensional, being the span of just one vector $\boldsymbol{q_i}$.
- Consequently, each projection matrix is a rank-one matrix.

Eqn (12.1) is the spectral theorem, stating that any real, symmetric matrix can be decomposed as a sum of projection matrices of rank one, scaled by the eigenvalue. The eigenvalue is considered the spectrum. Each component in the term is akin to a pure component of the matrix, much like white light has a spectrum of pure primary colors.

12.1.2 Sylvester's Law of Inertia

Now that we are listing theorems, we have another one that goes by the physics-inspired name, the Law of Inertia (attributed to James Joseph Sylvester, not the brawny movie star). We stated it earlier: For real, symmetric matrices, the number of positive eigenvalues is the same as the number of positive pivots. Similarly, the negative

pivots and negative eigenvalues are equal in number. The proof of this theorem (which we will not attempt) involves the concept of similarity of matrices, which is our topic later in this chapter. We will, however, state the theorem more formally.

We have a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ so that $\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{A}$, with $\boldsymbol{A}\boldsymbol{s_i} = \lambda_i \boldsymbol{s_i}$. The Reduced Echelon Form (REF) of $\boldsymbol{A}$ is $\boldsymbol{R}$, with the pivot in the i^{th} row is r_i. Then, Sylvester's Law of Inertia states that:

$$\begin{aligned}
\text{Count}(\lambda_i > 0) &= \text{Count}(r_i > 0) \\
\text{Count}(\lambda_i = 0) &= \text{Count}(r_i = 0) \\
\text{Count}(\lambda_i < 0) &= \text{Count}(r_i < 0)
\end{aligned}$$

Note that the REF in this law is the result of Gaussian elimination, done with no scaling of the rows. It is not the RREF from Gauss-Jordan elimination, which makes all pivots one by scaling rows.

12.2 Hermitian Matrices

Some of the properties of real, symmetric matrices we stated so far apply also to complex matrices, with one important caveat: We need to redefine what "transpose" means for $\boldsymbol{A} \in \mathbb{C}^{n \times n}$. Let's expand our field to complex numbers and see how the properties and their proofs holds up. But before doing that, we have to state what "transpose" means in $\mathbb{C}$.

12.2.1 Conjugate Transpose and General Symmetry

Earlier, we defined the (Euclidean) norm of $\boldsymbol{x} \in \mathbb{R}^n$ as $\|\boldsymbol{x}\|^2 = \boldsymbol{x}^{\mathsf{T}}\boldsymbol{x}$. The norm stands for the length or the size of the vector, and we would like it to be real (and positive). In other words, we would like $\|\boldsymbol{x}\| \in \mathbb{R}$ even if $\boldsymbol{x} \in \mathbb{C}^n$.

When x_i is complex (in the form $a + ib$ with $b \neq 0$), x_i^2 is not real, and $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{x}$ is not real. What is always real and positive is $(a + ib)(a - ib) = a^2 + b^2$. $a - ib$ is the complex conjugate of $a + ib$, which we write as $(a + ib)^* = a - ib$. So we would like to have $x_i^* x_i$ in $\|\boldsymbol{x}\|^2$ rather than x_i^2, which means the right definition of the norm of $\boldsymbol{x} \in \mathbb{C}^n$ is $\|\boldsymbol{x}\|^2 = \boldsymbol{x}^{*\mathsf{T}}\boldsymbol{x}$.

$\boldsymbol{x}^{*\mathbf{T}}$ is the *conjugate transpose*, also known as *Hermitian transpose*, of $\boldsymbol{x}$, which we will write[1] as $\boldsymbol{x}^{\dagger} = \boldsymbol{x}^{*\mathbf{T}}$. Note that it is a generalization of $\boldsymbol{x}^{\mathbf{T}}$: For $\boldsymbol{x} \in \mathbb{R}$, $\boldsymbol{x}^{\dagger} = \boldsymbol{x}^{\mathbf{T}}$. We may come across yet another notation in some texts for the Hermitian transpose as $\boldsymbol{x}^{\mathrm{H}}$.

Similarly, we can define the complex conjugate transpose (AKA Hermitian transpose) of a matrix $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ as its generalized transpose, $\boldsymbol{A}^{\dagger}$. When $\boldsymbol{A}^{\dagger} = \boldsymbol{A}$, we have the generalized version of symmetry, and we call such matrices Hermitian.

Note that the product rule of transposes applies to the Hermitian transposes as well: $(\boldsymbol{AB})^{\dagger} = \boldsymbol{B}^{\dagger}\boldsymbol{A}^{\dagger}$.

Earlier, while discussing orthogonality, we stated that the inverse of an orthonormal matrix (in $\mathbb{R}^{n \times n}$) is its transpose. For $\boldsymbol{A} \in \mathbb{C}^{n \times n}$, the condition would be $\boldsymbol{A}^{\dagger} = \boldsymbol{A}^{-\mathbf{1}}$, and we call such matrices **unitary**. When we look up information we may come across "unitary" used as a synonym for "orthonormal," which is technically correct because the set of orthonormal matrices is a subset of the set of unitary matrices, just as $\mathbb{R} \subset \mathbb{C}$.

12.3 Eigen Properties of Hermitian Matrices

With this definition of generalized symmetry of matrices, we are ready to restate some of the properties of eigenvalues and eigenvectors we listed for real, symmetric matrices, and extend them to Hermitian matrices. Note, however, that the field of complex numbers, $\mathbb{C}$, is not critical for our use in computer science, except when we look for information on the internet, for instance, we may come across terminology and explanations stated in terms of Hermitian, unitary, complex conjugates etc. Such usage is common in the research literature as well.

1. The eigenvalues of Hermitian matrices are real:

$$\boldsymbol{A} \in \mathbb{C}^{n \times n},\ \boldsymbol{A}^{\dagger} = \boldsymbol{A},\ \boldsymbol{As} = \lambda \boldsymbol{s} \implies \lambda \in \mathbb{R}$$

Proof:

[1] Some people write $\boldsymbol{x}^{*}$ to mean both conjugation on top of transposition. For this reason, a less confusing notation for conjugate by itself may be an overline $\overline{a + ib} = a - ib$, but it may lead to another contextual confusion: Are we underlining the line above or overlining the variable below? Good or bad, we are going to stick with $*$ for complex conjugate and $\dagger$ for conjugate transpose.

(1) By the definition of eigenvalues:	$\boldsymbol{As}$	$= \lambda \boldsymbol{s}$
(2) Multiplying on the left with $\boldsymbol{s}^\dagger$:	$\boldsymbol{s}^\dagger \boldsymbol{As}$	$= \lambda \boldsymbol{s}^\dagger \boldsymbol{s}$
(3) Product rule of Hermitian transposes:	$(\boldsymbol{A}^\dagger \boldsymbol{s})^\dagger \boldsymbol{s}$	$= \lambda \boldsymbol{s}^\dagger \boldsymbol{s}$
(4) Since $\boldsymbol{A}$ is Hermitian:	$(\boldsymbol{As})^\dagger \boldsymbol{s}$	$= \lambda \boldsymbol{s}^\dagger \boldsymbol{s}$
(5) Using (1):	$(\lambda \boldsymbol{s})^\dagger \boldsymbol{s}$	$= \lambda \boldsymbol{s}^\dagger \boldsymbol{s}$
(6) Since λ is a scalar:	$\lambda^* \boldsymbol{s}^\dagger \boldsymbol{s}$	$= \lambda \boldsymbol{s}^\dagger \boldsymbol{s}$
(7) Since $\boldsymbol{s}^\dagger \boldsymbol{s}$ is never zero:	λ^*	$= \lambda$

Step (7) says λ is real. Since all eigenvalues are real, the $\boldsymbol{\Lambda}$ matrix (with eigenvalues in the diagonal) is Hermitian as well.

In fact, the proof we gave in §3 (page 199) holds for Hermitian matrices as well, with minor changes. We , however, provided a brand-new proof, now that we are in the happy position of being able to do the same thing in multiple ways.

2. The eigenvectors of Hermitian matrices are orthogonal:

$$\boldsymbol{A} \in \mathbb{C}^{n \times n},\ \boldsymbol{A}^\dagger = \boldsymbol{A},\ \boldsymbol{As_i} = \lambda_i \boldsymbol{s_i} \quad i \neq j \implies \boldsymbol{s_i} \perp \boldsymbol{s_j}$$

Note that the eigenvectors of complex matrix are, in general, complex: $\boldsymbol{s_i}, \boldsymbol{s_j} \in \mathbb{C}^n$. The proof is pretty much identical to the one given in the case of real, symmetric matrices (§2, page 201), but with minor changes to call transposes Hermitians.

Here is another, higher level, matrix-algebra way of looking at it:

(1) Eigenvector matrix:	$\boldsymbol{A}$	$= \boldsymbol{S\Lambda S^{-1}}$
(2) Taking the Hermitian transpose:	$\boldsymbol{A}^\dagger$	$= \boldsymbol{S^{-1}}^\dagger \boldsymbol{\Lambda}^\dagger \boldsymbol{S}^\dagger$
(3) Since $\boldsymbol{A}$ and $\boldsymbol{\Lambda}$ are Hermitian:	$\boldsymbol{A}$	$= \boldsymbol{S^{-1}}^\dagger \boldsymbol{\Lambda S}^\dagger$
(4) Equating the RHS of (1) and (3):	$\boldsymbol{S\Lambda S^{-1}}$	$= \boldsymbol{S^{-1}}^\dagger \boldsymbol{\Lambda S}^\dagger$

One way this can be true is if $\boldsymbol{S}^\dagger = \boldsymbol{S^{-1}}$ for $\boldsymbol{S} \in \mathbb{C}^{n \times n}$, which is the same as saying $\boldsymbol{S}^\mathsf{T} = \boldsymbol{S^{-1}}$ for $\boldsymbol{S} \in \mathbb{R}^{n \times n}$. If the transpose of a matrix is its inverse, then the matrix is orthonormal. We have not actually proven it because we are not sure at this point

whether the *only way* step (4) can be true is if $\boldsymbol{S}^{\dagger} = \boldsymbol{S}^{-1}$. Of course it is, but we cannot yet see why.

12.4 Markov Matrices

A square matrix is called a Markov matrix if all its entries are non-negative and the sum of each column vector is equal to one. It is also known as a left stochastic matrix[2]. "Stochastic" by the way is a fancy word meaning probabilistic. Markov matrices are also known as probability/transition/stochastic matrices.

Markov Matrix

Definition: $\boldsymbol{A} = \left[a_{ij}\right] \in \mathbb{R}^{n\times n}$ is a Markov matrix if

$$0 \leq a_{ij} \leq 1 \text{ and } \sum_{i=1}^{n} a_{ij} = 1$$

Markov matrices usually describe the transition probabilities between states in a stochastic mathematical model.

12.4.1 Properties of Markov Matrices

All the following properties of Markov matrices follow from the fact that the columns add up to one. In other words, if we were to add up all the *rows* of a Markov matrix, we would get a row of ones because each *column* adds up to one.

1. Markov matrices have one eigenvalue equal to one.
2. The product of two Markov Matrices is another Markov matrix.
3. All eigenvalues of a Markov matrix are less than or equal to one, in absolute value: $|\lambda_i| \leq 1$.

Let's try proving these properties one by one. Since its columns add up to one, a Markov matrix always has one eigenvalue equal to one.

[2]The right stochastic matrix, on the other hand, would be one in which the rows add up to one. Since our vectors are all column vectors, it is the left stochastic matrix that we will focus on. But we should keep in mind that the rows of a matrix are, at times, considered "row vectors." For such a row vector, a matrix would multiply it on the right and we can think of the so-called right-eigenvectors.

Proof:

(1) Since $\boldsymbol{A}$ is a Markov matrix: $\sum_{i=1}^{n} a_{ij} = 1$

(2) Therefore: $a_{jj} = 1 - \sum_{i=1, i\neq j}^{n} a_{ij}$

(3) The diagonal element in $\boldsymbol{A} - \boldsymbol{I}$: $(\boldsymbol{A} - \boldsymbol{I})_{jj} = 1 - \sum_{i=1, i\neq j}^{n} a_{ij}$

(4) Sum of columns in $\boldsymbol{A} - \boldsymbol{I}$: $(\boldsymbol{A} - \boldsymbol{I})_{jj} = 0$

(5) Therefore: $|\boldsymbol{A} - \boldsymbol{I}| = 0$

(5) above says that $|\boldsymbol{A} - \lambda \boldsymbol{I}| = 0$ with $\lambda = 1$, which means one is an eigenvalue of any Markov matrix. All other eigenvalues are less than one in absolute value,

Let's prove it again using a slightly more sophisticated technique. Let's construct a column vector of all ones, and call it $\boldsymbol{u} \in \mathbb{R}^n$. Taking the sum of the rows of the Markov matrix $\boldsymbol{A}$ is the same as multiplying on the left with $\boldsymbol{u}^\mathsf{T}$ (which is a single-row matrix of all ones), and we know that the product $\boldsymbol{u}^\mathsf{T}\boldsymbol{A} = \boldsymbol{u}^\mathsf{T}$ because the columns of $\boldsymbol{A}$ add up to one. Taking the transpose, $\boldsymbol{A}^\mathsf{T}\boldsymbol{u} = \boldsymbol{u}$, which says that $\boldsymbol{A}^\mathsf{T}$ has an eigenvalue of one (with $\boldsymbol{u}$ as the eigenvector, which is not important for us). Now, the eigenvalues of a matrix and its transpose are the same (but the eigenvectors may be different) because the characteristic polynomial, being a determinant, does not change when we take the transpose. So we can see that $\boldsymbol{A}$ has at least one eigenvalue equal to one.

The second property is that the product of two Markov matrices is another Markov matrix. If $\boldsymbol{A}$ and $\boldsymbol{B}$ are Markov, the $\boldsymbol{AB}$ cannot have any negative entries because matrix multiplication, being addition and multiplication of elements, cannot introduce a negative sign. We also know that the each one of the columns of $\boldsymbol{A}$ and $\boldsymbol{B}$ adds up to one, which means $\boldsymbol{u}^\mathsf{T}\boldsymbol{A} = \boldsymbol{u}^\mathsf{T}$ and $\boldsymbol{u}^\mathsf{T}\boldsymbol{B} = \boldsymbol{u}^\mathsf{T}$. Consider $\boldsymbol{u}^\mathsf{T}\boldsymbol{AB} = \boldsymbol{u}^\mathsf{T}\boldsymbol{B} = \boldsymbol{u}^\mathsf{T}$. Therefore the columns of $\boldsymbol{AB}$ also add up to one, or the product of two Markov matrices is another Markov matrix.

The third property is that all the absolute values of the eigenvalues of a Markov matrix are less than one: $|\lambda_i| \leq 1$, which follows from the product property. $\boldsymbol{A}^n$ is a Markov matrix. If one of the eigenvalues

was more than one, then $\boldsymbol{A}^k, k \to \infty$ could not be a Markov matrix because its elements would have to be growing exponentially.

12.4.2 Steady State

In order to make this cryptic proof of the third property a bit more accessible, let's work out an example. Let's say we are studying the human migration patterns across the globe, and know the yearly migration probabilities as in Table 12.1 through some unspecified demographic studies. We know nothing else, except perhaps that the birth and death rate are close enough to each other for us to assume that they add up to zero everywhere. One reasonable question to ask would be about the steady state: If we wait long enough, do the populations stabilize?

Note that the numbers in each column add up to 100% because people either stay or leave. The numbers in each row, on the other hand, do not. Asia-Pacific and Africa lose people to the Americas and Europe.

Once we have probabilities like Table 12.1, the first thing to do would be to put the values in matrices, now that we know enough Linear Algebra.

$$\boldsymbol{A} = \begin{bmatrix} 0.80 & 0.04 & 0.05 & 0.05 \\ 0.10 & 0.90 & 0.07 & 0.08 \\ 0.03 & 0.01 & 0.75 & 0.02 \\ 0.07 & 0.05 & 0.13 & 0.85 \end{bmatrix} \quad \boldsymbol{x_0} = \begin{bmatrix} 4.68 \\ 1.20 \\ 1.34 \\ 0.75 \end{bmatrix} \quad \boldsymbol{x_{k+1}} = \boldsymbol{A x_k}$$

where we put the initial populations in a vector $\boldsymbol{x_0}$. As we can see, $\boldsymbol{A}$ is a Markov matrix. It describes how the population evolves over time. The populations for year k evolve to that of year $k+1$ as $\boldsymbol{A x_k}$, which is identical to what we did in the case of Fibonacci numbers in §11.7 (page 209). As we learned there, the long-term evolution of the populations is fully described by the eigenvalues λ_i of $\boldsymbol{A}$. If $|\lambda_i| > 1$, we will have a growing system, if $|\lambda_i| < 1$, we will have system tending to zero. If we have an eigenvalue $|\lambda_i| = 1$, we will have as steady state. And we know that $\boldsymbol{A}$ does have a eigenvalue equal to one.

Knowing that $\boldsymbol{x} = \boldsymbol{x_k}$ will stabilize and reach an equilibrium value, we can implement an iterative method to compute it: First, initialize it $\boldsymbol{x} \leftarrow \boldsymbol{x_0}$ Then iterate until convergence: $\boldsymbol{x} \leftarrow \boldsymbol{A x}$. Doing all this

Table 12.1 Migration probabilities

Destination ↓ Source →	Asia-Pacific	Americas	Africa	Europe
Asia-Pacific	80%	4%	5%	5%
Americas	10%	90%	7%	8%
Africa	3%	1%	75%	2%
Europe	7%	5%	13%	85%
Population (billions)	4.68	1.20	1.34	0.75

in numeric a program, we get:

$$\boldsymbol{x}^{\mathsf{T}} = \begin{bmatrix} 0.7262 & 1.844 & 0.2549 & 1.175 \end{bmatrix}$$

The eigen way of doing it would be to find the eigenbasis, find their linear combination to form $\boldsymbol{x_0}$:

$$\boldsymbol{x_0} = \sum_{i=1}^{n} c_i \boldsymbol{s_i}$$

and then say that (with $\lambda_1 = 1$ and all other $|\lambda_i| < 1$):

$$\boldsymbol{x_k} = \boldsymbol{A}^k \sum_{i=1}^{n} c_i \boldsymbol{s_i} = \sum_{i=1}^{n} c_i \boldsymbol{A}^k \boldsymbol{s_i} = \sum_{i=1}^{n} c_i \lambda_i^k \boldsymbol{s_i} = c_1 \lambda_1^k \boldsymbol{s_1} = c_1 \boldsymbol{s_1}$$

Therefore, we need to know only c_1. But it still looks like a pain to compute because, in order to find c_1, we have to find all other c_i, which means we have to find all the eigenvectors.

Luckily for us, there is a shortcut. As $\boldsymbol{x}$ evolves from $\boldsymbol{x_0}$ to $\boldsymbol{x_k}$, the sum of its components is the total population, which we are assuming to be a constant. Therefore, in $\boldsymbol{x_k} = c_1 \boldsymbol{s_1}$, the components of $\boldsymbol{x_k}$ and $c_1 \boldsymbol{s_1}$ should add up to the same number, which is the total population. In other words,

$$c_1 = \frac{\sum x_{0i}}{\sum s_{1i}}$$

where the summation is over the components of $\boldsymbol{x_0}$ and $\boldsymbol{s_1}$. So we only need to know one eigenvector, $\boldsymbol{s_1}$ corresponding to the dominant eigenvalue, $\lambda = 1$ in order to compute the limiting value of $\boldsymbol{x} = \boldsymbol{x_k}$ as $k \to \infty$. As expected, this computation will also yield the same answer:

$$\boldsymbol{x}^{\mathsf{T}} = \begin{bmatrix} 0.7262 & 1.844 & 0.2549 & 1.175 \end{bmatrix}$$

12.4.3 Portal to a Big Field

Markov matrices are the starting point of the associated topics in mathematical modeling, some of which are heavily used in machine learning and data mining. For instance, Hidden Markov Models are used in text mining for the so-called part-of-speech tagging. In our short introduction here, we focused only on the basics, ignoring many subtleties. A quick flip through the pages of this book will reveal that permutation matrices are, in fact, Markov matrices. What are their steady states? They have none; they have oscillating solutions. Other, more complicated matrices may have multiple steady states among which the solutions oscillate.

The Google Page Rank algorithm is a multi-billion dollar success story of Markov matrices, which, as we see in the box, does not take much more than our discussion here to fully understand.

12.5 Positive Definite Matrices

We saw that real, symmetric matrices were "good" matrices to work with because they have real eigenvalues and orthogonal eigenvectors. *Positive definite matrices* are even better.

Positive Definite Matrix

Definition: A real, symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is positive definite if all it eigenvalues are positive ($\lambda_i > 0$).

If an eigenvalue is equal to zero (which means $\boldsymbol{A}$ is singular), then we call the matrix positive semidefinite. In other words, if we only have $\lambda_i \geq 0$, then all we can say is that $\boldsymbol{A}$ is positive semidefinite.

Similarly, we can define negative definite and negative semidefinite matrices, although they have dubious mathematical relevance.

We can extend the definition to $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ using our definition of Hermitian transpose, and most of what we learn here will apply to them as well. However, for our purpose in computer science, we can safely restrict ourselves to the real domain.

12.5.1 Test for Positive Definiteness

1. $\lambda_i > 0$? This test follows directly from our definition of positive definite matrices.

2. All the pivots > 0? The second test is essentially the same as the first, by Sylvester's law connecting the signs of pivots and eigenvalues.

3. $\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} > 0$? For a positive definite $\boldsymbol{A}$ and for any nonzero vector $\boldsymbol{x}$, $\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} > 0$. Test (3) is a powerful one. In fact, some textbooks define positive definiteness using this statement. Let's prove that it is equivalent to Test (1), which is our definition of positive definiteness.

   Proof: $x^\mathsf{T}\boldsymbol{A}\boldsymbol{x} > 0 \implies \lambda_i > 0$

(1) Consider an eigenvector of $\boldsymbol{A}$:	$\boldsymbol{A}\boldsymbol{s} = \lambda\boldsymbol{s}$
(2) Left-multiplying with $\boldsymbol{s}^\mathsf{T}$:	$\boldsymbol{s}^\mathsf{T}\boldsymbol{A}\boldsymbol{s} = \lambda\boldsymbol{s}^\mathsf{T}\boldsymbol{s}$
(3) Rearranging:	$\lambda = \dfrac{\boldsymbol{s}^\mathsf{T}\boldsymbol{A}\boldsymbol{s}}{\boldsymbol{s}^\mathsf{T}\boldsymbol{s}}$
(4) Since $\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} > 0$ for any $\boldsymbol{x}$ and $\boldsymbol{s}^\mathsf{T}\boldsymbol{s} > 0$:	$\lambda > 0$

   Proof: $\lambda_i > 0 \implies x^\mathsf{T}\boldsymbol{A}\boldsymbol{x} > 0$

(1) For a real symmetric $\boldsymbol{A}$:	$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\mathsf{T}$
(2) Multiply with $\boldsymbol{x}^\mathsf{T}$ and $\boldsymbol{x}$:	$\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x}^\mathsf{T}\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\mathsf{T}\boldsymbol{x}$
(3) Calling $\boldsymbol{y} = \boldsymbol{Q}\boldsymbol{x}$:	$\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}^\mathsf{T}\boldsymbol{\Lambda}\boldsymbol{y}$
(4) Since $\boldsymbol{\Lambda}$ is diagonal $[\lambda_i]$:	$\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} = \sum \lambda_i y_i^2$
(5) Since $\lambda_i > 0$:	$\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} > 0$

4. Do all the *leading submatrices* have $|\boldsymbol{A_k}| > 0$? For $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, the leading submatrix (AKA upperleft submatrix) of dimension $k < n$, $\boldsymbol{A_k} \in \mathbb{R}^{k\times k}$, is a block matrix of the first k rows and columns of $\boldsymbol{A}$. For positive definite $\boldsymbol{A}$, the determinants of all such $\boldsymbol{A_k}$ are greater than zero.

$$\boldsymbol{A} = \begin{bmatrix} \boxed{\begin{matrix} a_{11} & a_{21} & & \\ a_{21} & a_{22} & & \\ \vdots & \cdots & \ddots & \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{matrix}} & \boldsymbol{A_k} \\ \begin{matrix} \vdots & & \cdots & \end{matrix} & \begin{matrix} \cdots \\ \ddots \end{matrix} \end{bmatrix} \tag{12.2}$$

Eqn (12.2) illustrates the definition of a leading submatrix $\boldsymbol{A_k}$.
Proof: Since $\boldsymbol{A}$ is positive definite, we know that $\boldsymbol{x^T A x} > 0$ for *any* $\boldsymbol{x}$.
Let's consider an $\boldsymbol{x}_{[k,0]}$ which has nonzero entries in the first k elements, the rest being zero. $\boldsymbol{x^T A x} = \boldsymbol{x}_{[k,0]}^{\mathsf{T}} \boldsymbol{A_k} \boldsymbol{x}_{[k,0]}$, has to be greater than zero.
Then, by Test (3), $\boldsymbol{A_k}$ is positive definite, all its eigenvalues are positive, and its determinant, being the product of its eigenvalues, has to be positive.

5. Can we write $\boldsymbol{B} = \boldsymbol{A^T A}$? For a positive definite matrix $\boldsymbol{B}$, we can always find an *invertible* matrix $\boldsymbol{A}$ such that $\boldsymbol{B} = \boldsymbol{A^T A}$. Although not usually used as a test, this property is an important one, and our last topic in the last chapter depends on it. Let's therefore prove it here.

   Proof: $\boldsymbol{B} = \boldsymbol{A^T A} \implies \boldsymbol{B}$ is positive definite.

$$\boldsymbol{x^T B x} = \boldsymbol{x^T A^T A x} = (\boldsymbol{Ax})^{\mathsf{T}} \boldsymbol{Ax} = \|\boldsymbol{Ax}\|^2 > 0$$

   Since $\boldsymbol{A}$ is invertible, $\boldsymbol{Ax} \neq \boldsymbol{0}$ unless $\boldsymbol{x} = \boldsymbol{0}$.
   Proof: $\boldsymbol{B}$ is positive definite $\implies \boldsymbol{B} = \boldsymbol{A^T A}$ for some invertible $\boldsymbol{A}$.

(1) Since $\boldsymbol{B}$ is positive definite:	$\boldsymbol{B} = \boldsymbol{Q \Lambda Q^T}$
(2) Defining $\boldsymbol{\Lambda^{\frac{1}{2}}} = \left[\sqrt{\lambda}\right]$:	$\boldsymbol{B} = \boldsymbol{Q \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}} Q^T}$
(Matrix with $\sqrt{\lambda_i}$ in the diagonal)	$= \boldsymbol{Q \Lambda^{\frac{1}{2}}} \left(\boldsymbol{Q \Lambda^{\frac{1}{2}}}\right)^{\mathsf{T}}$
(3) Defining $\boldsymbol{A^T} = \boldsymbol{Q \Lambda^{\frac{1}{2}}}$:	$\boldsymbol{B} = \boldsymbol{A^T A}$

   Since $\boldsymbol{B}$ is positive definite, its eigenvalues are positive, and $\boldsymbol{\Lambda^{\frac{1}{2}}}$ is invertible. So is $\boldsymbol{Q}$ because $\boldsymbol{Q^{-1}} = \boldsymbol{Q^T}$. So $\boldsymbol{A}$ is invertible.

12.5.2 Applying the Tests

Let's see how we can apply these five tests to various matrices to determine whether they are positive definite.

- If $\boldsymbol{A}$ is positive definite, is $\boldsymbol{A^{-1}}$ positive definite as well?

We can apply Test (1): $\boldsymbol{A}$ has positive λ_i. The eigenvalues of $\boldsymbol{A^{-1}}$ are $\frac{1}{\lambda_i}$ which are positive too. Therefore, $\boldsymbol{A^{-1}}$ is positive definite.

- If $\boldsymbol{A}$ and $\boldsymbol{B}$ are positive definite, how about $\boldsymbol{A+B}$?

  We apply Test (3) here: $\boldsymbol{x^{\mathsf{T}}}(\boldsymbol{A}+\boldsymbol{B})\boldsymbol{x} = \boldsymbol{x^{\mathsf{T}}}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{x^{\mathsf{T}}}\boldsymbol{B}\boldsymbol{x} > 0$ because $\boldsymbol{A}$ and $\boldsymbol{B}$ are both positive definite. So is the sum.

- For any $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ with rank$(\boldsymbol{A}) = n$ (full column rank), is $\boldsymbol{A^{\mathsf{T}}}\boldsymbol{A}$ positive definite?

  Test (3) proves useful again: $\boldsymbol{x^{\mathsf{T}}}\boldsymbol{A^{\mathsf{T}}}\boldsymbol{A}\boldsymbol{x} = (\boldsymbol{A}\boldsymbol{x})^{\mathsf{T}}\boldsymbol{A}\boldsymbol{x} = \|\boldsymbol{A}\boldsymbol{x}\|^2 \geq 0$. It is equal to zero only if $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ has nonzero solutions, which means $\boldsymbol{A}$ has null space and $\boldsymbol{A}$ is rank-deficient. Since we started with the assumption that $\boldsymbol{A}$ is full rank, $\boldsymbol{x^{\mathsf{T}}}\boldsymbol{A^{\mathsf{T}}}\boldsymbol{A}\boldsymbol{x} > 0$ and $\boldsymbol{A^{\mathsf{T}}}\boldsymbol{A}$ is positive definite.

- For a positive definite $\boldsymbol{A}$, is $\boldsymbol{M^{-1}}\boldsymbol{A}\boldsymbol{M}$ positive definite for *any invertible* $\boldsymbol{M}$?

  The answer is yes, and the reason is that $\boldsymbol{A}$ and $\boldsymbol{M^{-1}}\boldsymbol{A}\boldsymbol{M}$ are *similar matrices*, and they have the same eigenvalues. We shall soon define matrix similarity and prove that similar matrices have the same eigenvalues.

12.5.3 Quadratic Forms

One of the tests for positive definiteness, namely $\boldsymbol{x^{\mathsf{T}}}\boldsymbol{A}\boldsymbol{x} > 0$, which we called Test (3), leads to the big topic of quadratic forms. Although not directly relevant to computer science, we will introduce the topic of quadratic forms here, and list some of its general properties, so that we may recognize it if we happen to come across it in some mathematically oriented literature later on.

For a real, symmetric matrix, we can expand $\boldsymbol{x^{\mathsf{T}}}\boldsymbol{A}\boldsymbol{x}$ in $\mathbb{R}^2$as below:

$$\begin{aligned} &\boldsymbol{A} \in \mathbb{R}^{2\times 2}, \boldsymbol{A^{\mathsf{T}}} = \boldsymbol{A} \implies \boldsymbol{A} = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \text{ and } \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &\boldsymbol{x^{\mathsf{T}}}\boldsymbol{A}\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = ax_1^2 + 2bx_1x_2 + cx_2^2 \end{aligned} \tag{12.3}$$

As we can see, the product $\boldsymbol{x^{\mathsf{T}}}\boldsymbol{A}\boldsymbol{x}$ is a pure quadratic form, with no linear or constant terms in it. Furthermore, as in Test (4), if the

determinants of the leading submatrices of $\boldsymbol{A}$ are positive, we have:

$$a > 0 \text{ and } ac - b^2 > 0$$

Let's take an example and see how $a > 0$ and $|\boldsymbol{A}| > 0$ implies that $\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} > 0$ for all $\boldsymbol{x}$.

$$\boldsymbol{A} = \begin{bmatrix} 2 & 6 \\ 6 & c \end{bmatrix} \text{ and } |\boldsymbol{A}| = 2c - 36 > 0 \text{ if } c > 18.$$

Let's set $c = 20 \implies$

$$\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} = 2x_1^2 + 12x_1x_2 + 20x_2^2 = 2(x_1^2 + 6x_1x_2 + 9x_2^2) + 2x_2^2$$
$$\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} = 2(x_1 + 3x_2)^2 + 2x_2^2 > 0$$

As we can see, if $c < 18$, the last term in $\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x}$ above becomes negative, and the whole expression would be negative for *some* value of x_1 and x_2 (both equal to zero, for instance). But if $c > 18$, the expression, being the sum of two squares, can never be negative. And hence, $\boldsymbol{A}$ is positive definite. If $c = 18$, $\boldsymbol{A}$ is positive semidefinite.

Note that in $\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} = 2(x_1+3x_2)^2+2x_2^2$, the 2 multiplying the first term is a in $\boldsymbol{A}$, which is the first pivot. The multiplier of the second term, 2, is the second pivot. Inside the parentheses of the first term, the factor multiplying x_2, namely 3, is the multiplier of the first row in Gaussian elimination to subtract from the second row. The quadratic form, as we can see, is intrinsically connected to pivots and Gaussian elimination. Our discussion with this little $\mathbb{R}^{2\times2}$ generalizes to $\mathbb{R}^{n\times n}$, which has interesting mathematical, if not numeric or algorithmic, applications.

This connection between row operations in Gaussian elimination and completing the squares in the quadratic form becomes even clearer if we work with the general matrix $\boldsymbol{A}$ in Eqn (12.3).

$$\boldsymbol{A} = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \xrightarrow{r_2 \leftarrow -\frac{b}{a}r_1 + r_2} \begin{bmatrix} a & b \\ 0 & c - \frac{b^2}{a} \end{bmatrix}$$

$$a\left(x_1 + \frac{b}{a}x_2\right)^2 = ax_1^2 + 2bx_1x_2 + \frac{b^2}{a}x_2^2 = \boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} - cx_2^2 + \frac{b^2}{a}x_2^2$$

$$\implies \boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} = a\left(x_1 + \frac{b}{a}x_2\right)^2 + \left(c - \frac{b^2}{a}\right)x_2^2$$

We can see the pivots and row multipliers in the expression for (square-completed) $\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x}$ in the last line above, can't we? Furthermore, $\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} > 0$ if:

$$cx_2^2 > \frac{b^2}{a}x_2^2 \quad \text{or when} \quad c > \frac{b^2}{a} \implies ac - b^2 = |\boldsymbol{A}| > 0$$

12.5.4 Positive Definiteness and Symmetry

In our definition of positive definite matrix, we restricted ourselves to symmetric ones. Our basic definition indeed made it abundantly clear, by stating, "A real, symmetric matrix. . . ". We should note that some people relax this restriction, and consider matrices that are not symmetric also positive definite if they pass the tests listed. While we will stick with our definition because it makes sense for our use in computer science, purely mathematical work may consider more general definitions, and indeed more general fields like $\mathbb{C}$, of even unspecified ones.

12.6 Gram Matrix

For $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\text{rank}(\boldsymbol{A}) = r$, its Gram matrix is defined[3] as $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$ and has important applications in machine learning. It also has some interesting properties that we exploit at various points in this book.

Before getting into the properties of Gram matrices, let's revisit some basic properties of ranks. The first one is that the row-rank and the column-rank of a matrix are the same. Earlier, we defined the rank as the number of pivots and also as the number of linearly independent vectors in its row (or column) space. The number of independent vectors is indeed the dimension of the subspace. To prove the equality of row and column ranks, we will use the latter definition.

We know that elementary row operations do not change the row space of a matrix because they involve linear combinations of the rows. Therefore, $\boldsymbol{A}$ has the same row rank as its RREF because they both have the same row space. As we saw earlier, the shape of RREF

[3]The author is not quite sure if $\boldsymbol{A}\boldsymbol{A}^\mathsf{T}$ also considered a Gram matrix, although it should be, by symmetry.

is, in its most general case, as follows:

$$\boldsymbol{A} \xrightarrow{\text{RREF}} \left[\begin{array}{c} \boldsymbol{I}_r \oplus \boldsymbol{F}_{r\times(n-r)} \\ \hline \boldsymbol{0}_{(m-r)\times n} \end{array}\right] = \boldsymbol{R_1}$$

where $\oplus$ indicates that the $\boldsymbol{I}$ and $\boldsymbol{F}$ matrices have their columns mixed in, not cleanly separated. Clearly, the dimension of the row space of $\boldsymbol{A}$ is r, which is the row rank of both $\boldsymbol{A}$ and $\boldsymbol{R_1}$.

We can go further and get rid of the columns of $\boldsymbol{F}$ using elementary *column* operations, which do not change the column space. Starting from $\boldsymbol{R_1}$, the RREF of $\boldsymbol{A}$, we can perform a series of elementary column operations, without affecting its column space, to reduce the matrix to the following form:

$$\text{RREF} = \left[\begin{array}{c} \boldsymbol{I}_r \oplus \boldsymbol{F}_{r\times(n-r)} \\ \hline \boldsymbol{0}_{(m-r)\times n} \end{array}\right] \xrightarrow{\text{Col. Ops}} \left[\begin{array}{c|c} \boldsymbol{I}_r & \boldsymbol{0}_{r\times(n-r)} \\ \hline \boldsymbol{0}_{(m-r)\times n} \end{array}\right] = \boldsymbol{R_2}$$

This form, obviously, has the same row and column rank of r, and since our elementary operations changed neither, we can say that the row rank of $\boldsymbol{R_1}$ is the same as the column rank of $\boldsymbol{R_2}$, which is r. This claim propagates to the original $\boldsymbol{A}$, and it has the same row and column ranks. Therefore, row rank of any matrix is the same as its column rank. It should be immediately obvious from this statement that $\text{rank}(\boldsymbol{A}) = \text{rank}\big(\boldsymbol{A}^\mathsf{T}\big)$.

Getting back to $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$, we can see that every vector $\boldsymbol{x} \in \mathcal{N}(\boldsymbol{A})$ is also in the null space of $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0} \implies \boldsymbol{A}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0} \implies \boldsymbol{x} \in \mathcal{N}(\boldsymbol{A}^\mathsf{T}\boldsymbol{A}) \implies \mathcal{N}(\boldsymbol{A}) \subseteq \mathcal{N}(\boldsymbol{A}^\mathsf{T}\boldsymbol{A})$$

There may be more vectors in $\mathcal{N}(\boldsymbol{A}^\mathsf{T}\boldsymbol{A})$, which is why we only claim $\mathcal{N}(\boldsymbol{A}) \subseteq \mathcal{N}(\boldsymbol{A}^\mathsf{T}\boldsymbol{A})$. Now, coming at it from the other end, if $\boldsymbol{x} \in \mathcal{N}(\boldsymbol{A}^\mathsf{T}\boldsymbol{A})$:

$$\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0} \implies \boldsymbol{x}^\mathsf{T}\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} = 0, (\boldsymbol{A}\boldsymbol{x})^\mathsf{T}(\boldsymbol{A}\boldsymbol{x}) = 0 \implies \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$$

Every vector in $\mathcal{N}(\boldsymbol{A}^\mathsf{T}\boldsymbol{A})$ is also in $\mathcal{N}(\boldsymbol{A})$. Both these conditions can happen only if $\mathcal{N}(\boldsymbol{A}) = \mathcal{N}(\boldsymbol{A}^\mathsf{T}\boldsymbol{A})$, which means $\boldsymbol{A}$ and $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$ have the same nullity.

Since $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, its domain is $\mathbb{R}^n$. And since $\boldsymbol{A}^\mathsf{T}\boldsymbol{A} \in \mathbb{R}^{n\times n}$, its domain also is $\mathbb{R}^n$. Therefore, by the rank-nullity theorem, we have

$\text{rank}(\boldsymbol{A}) = \text{rank}\big(\boldsymbol{A}^\intercal\boldsymbol{A}\big)$ because

$$\text{rank}(\boldsymbol{A}) = \dim(\mathbb{R}^n) - \text{nullity}(\boldsymbol{A})$$
$$\text{rank}\big(\boldsymbol{A}^\intercal\boldsymbol{A}\big) = \dim(\mathbb{R}^n) - \text{nullity}(\boldsymbol{A}^\intercal\boldsymbol{A})$$
$$\text{And nullity}(\boldsymbol{A}) = \text{nullity}(\boldsymbol{A}^\intercal\boldsymbol{A}) \implies \text{rank}(\boldsymbol{A}) = \text{rank}\big(\boldsymbol{A}^\intercal\boldsymbol{A}\big)$$

It is also easy enough to prove that $\boldsymbol{A}$ and $\boldsymbol{A}^\intercal\boldsymbol{A}$ have the same row space, looking at the product $\boldsymbol{A}^\intercal\boldsymbol{A}$ as the linear combinations of the rows of $\boldsymbol{A}$ and as the linear combinations of the columns of $\boldsymbol{A}^\intercal$. At this stage in our Linear Algebra journey, we are in the happy position of being able to prove a lemma in multiple ways, and we can afford to pick and choose.

Summarizing, $\boldsymbol{A}$ and $\boldsymbol{A}^\intercal\boldsymbol{A}$ have the same row and null spaces. $\boldsymbol{A}^\intercal$ and $\boldsymbol{A}\boldsymbol{A}^\intercal$ have the same column space and left-null space. All four of them have the same rank.

For a full-column-rank matrix ($\boldsymbol{A} \in \mathbb{R}^{m\times n}, \text{rank}(\boldsymbol{A}) = n$), the Gram matrix $\boldsymbol{A}^\intercal\boldsymbol{A}$ is a full-rank, square matrix with the same rank. It is also much smaller. We can, therefore test the linear independence of the n column vectors in $\boldsymbol{A}$ by looking at the invertibility (or, equivalently, the determinant) of the Gram matrix. In data science, as we shall see in the last chapter of this book, the Gram matrix is the covariance matrix of a zero-centered data set.

12.7 Matrix Similarity

We consider a matrix $\boldsymbol{A}$ *similar* to another one $\boldsymbol{B}$ if we can write $\boldsymbol{A} = \boldsymbol{M}^{-1}\boldsymbol{B}\boldsymbol{M}$ for some invertible matrix $\boldsymbol{M}$. We denote similarity as $\boldsymbol{A} \sim \boldsymbol{B}$. As we can immediately see, $\boldsymbol{A}$ is similar to $\boldsymbol{\Lambda}$, its eigenvalue matrix, when we have a full set of eigenvectors (and therefore the eigenvector matrix $\boldsymbol{S}$ is invertible) because then $\boldsymbol{A} = \boldsymbol{S}\boldsymbol{\Lambda}\boldsymbol{S}^{-1}$. Note that we are talking about any square matrix $\boldsymbol{A}$ now, not necessarily symmetric matrices.

While this definition of matrix similarity may look strange at first, there are good reasons behind it: Similar matrices share several key properties. Here is a list, with proof wherever necessary:

1. For invertible matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, $\boldsymbol{A}\boldsymbol{B} \sim \boldsymbol{B}\boldsymbol{A}$.

   Proof:

$$\boldsymbol{AB} \sim \boldsymbol{MABM^{-1}} \sim \boldsymbol{BABB^{-1}} \sim \boldsymbol{BA} \text{ with } \boldsymbol{M} = \boldsymbol{B}$$

2. If $\boldsymbol{A} \sim \boldsymbol{B}$, they have the same characteristic polynomials.
   Proof: Let's use $p_{\boldsymbol{A}}(\lambda)$ to denote the Characteristic Polynomial of $\boldsymbol{A}$. Since $\boldsymbol{A} \sim \boldsymbol{B}$, $\boldsymbol{A} = \boldsymbol{M^{-1}BM}$.

   (1) Characteristic Polynomial $\quad p_{\boldsymbol{A}}(\lambda) = |\boldsymbol{A} - \lambda \boldsymbol{I}|$
   (2) Since $\boldsymbol{A} = \boldsymbol{M^{-1}BM}$: $\quad p_{\boldsymbol{A}}(\lambda) = \left|\boldsymbol{M^{-1}BM} - \lambda \boldsymbol{I}\right|$
   (3) Since $\boldsymbol{M^{-1}M} = \boldsymbol{I}$: $\quad p_{\boldsymbol{A}}(\lambda) = \left|\boldsymbol{M^{-1}BM} - \boldsymbol{M^{-1}}\lambda \boldsymbol{IM}\right|$
   (4) Factorizing $\boldsymbol{M}$ and $\boldsymbol{M^{-1}}$: $\quad p_{\boldsymbol{A}}(\lambda) = \left|\boldsymbol{M^{-1}}(\boldsymbol{B} - \lambda \boldsymbol{I})\boldsymbol{M}\right|$
   (5) Since $|\boldsymbol{AB}| = |\boldsymbol{A}|\,|\boldsymbol{B}|$: $\quad p_{\boldsymbol{A}}(\lambda) = \left|\boldsymbol{M^{-1}}\right| |\boldsymbol{B} - \lambda \boldsymbol{I}|\,|\boldsymbol{M}|$
   (6) Since $|\boldsymbol{A}|\left|\boldsymbol{A^{-1}}\right| = |\boldsymbol{I}| = 1$: $p_{\boldsymbol{A}}(\lambda) = |\boldsymbol{B} - \lambda \boldsymbol{I}| = p_{\boldsymbol{B}}(\lambda)$

   Step (6) above shows that both $\boldsymbol{A}$ and the similar matrix $\boldsymbol{B}$ have the same characteristic polynomial. Note that, as a consequence of the characteristic polynomials being the same, the algebraic multiplicities of the eigenvalues (how many times each one is repeated) are also the same for $\boldsymbol{A}$ and $\boldsymbol{B}$.

3. If $\boldsymbol{A} \sim \boldsymbol{B}$, they have the same eigenvalues, but not necessarily the same eigenvectors.
   Proof: We already showed that $\boldsymbol{A}$ and $\boldsymbol{B}$ have the same characteristic polynomial. Therefore they have the same eigenvalues (which are the roots of the said polynomial). But here is another proof.

   (1) Definition of similarity: $\quad \boldsymbol{A} \quad = \boldsymbol{M^{-1}BM}$
   (2) Left-multiplying by $\boldsymbol{M}$: $\quad \boldsymbol{MA} \quad = \boldsymbol{BM}$
   (3) Right-multiplying by $\boldsymbol{s}$: $\quad \boldsymbol{MAs} \quad = \boldsymbol{BM}s$
   (4) Since $\boldsymbol{s}$ is eigenvector of $\boldsymbol{A}$: $\quad \boldsymbol{M}\lambda\boldsymbol{s} \quad = \boldsymbol{BM}s$
   (5) Commuting λ: $\quad \lambda(\boldsymbol{Ms}) \; = \boldsymbol{B}(\boldsymbol{M}s)$
   (6) LHS $\leftrightarrow$ RHS: $\quad \boldsymbol{B}(\boldsymbol{M}s) = \lambda(\boldsymbol{Ms})$

   Step (6) says that $\boldsymbol{Ms}$ is an eigenvector of $\boldsymbol{B}$ with the same eigenvalue λ, by the definition of eigenvalues and eigenvectors.

4. Combining the previous property with the first one, we can see that $\boldsymbol{AB}$ and $\boldsymbol{BA}$ have the same eigenvalues.

5. If $\boldsymbol{A} \sim \boldsymbol{B}$ and $\boldsymbol{A}$ is positive definite, so is $\boldsymbol{B}$
   Proof: Determinant is the product of eigenvalues.

6. $\boldsymbol{A} \sim \boldsymbol{B} \implies |\boldsymbol{A}| = |\boldsymbol{B}|$
   Proof: Determinant is the product of eigenvalues.

7. $\boldsymbol{A} \sim \boldsymbol{B} \implies \text{trace}(\boldsymbol{A}) = \text{trace}(\boldsymbol{B})$
   Proof: Trace is the sum of eigenvalues.

8. Similar matrices have the same rank.
   Proof: The number of (positive and negative) pivots is the same the number of (positive and negative) eigenvalues, through Sylvester's Law of Inertia. And the rank is the number of pivots.

12.7.1 Equivalence Relation

The properties listed above are the ones that are useful for us in Linear Algebra. However, more basic than them, similarity as a relation has some fundamental properties that make it an *equivalence relation*. Here is a formal statement of what it means.

For $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \in \mathbb{R}^{n \times n}$, we can easily show that similarity as a relation is:

Reflexive: $\boldsymbol{A} \sim \boldsymbol{A}$
Proof: $\boldsymbol{A} = \boldsymbol{I}\boldsymbol{A}\boldsymbol{I}^{-1}$

Symmetric: $\boldsymbol{A} \sim \boldsymbol{B} \implies \boldsymbol{B} \sim \boldsymbol{A}$
Proof: $\boldsymbol{A} = \boldsymbol{M}\boldsymbol{B}\boldsymbol{M}^{-1} \implies \boldsymbol{B} = \boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{M} = \boldsymbol{M}'\boldsymbol{A}\boldsymbol{M}'^{-1}$

Transitive: $\boldsymbol{A} \sim \boldsymbol{B}$ and $\boldsymbol{B} \sim \boldsymbol{C}$ then $\boldsymbol{A} \sim \boldsymbol{C}$
Proof: $\boldsymbol{A} = \boldsymbol{M}\boldsymbol{B}\boldsymbol{M}^{-1}, \boldsymbol{B} = \boldsymbol{N}\boldsymbol{C}\boldsymbol{N}^{-1}$
$\implies \boldsymbol{A} = \boldsymbol{M}\boldsymbol{N}\boldsymbol{C}\boldsymbol{N}^{-1}\boldsymbol{M}^{-1} = (\boldsymbol{M}\boldsymbol{N})\,\boldsymbol{C}\,(\boldsymbol{M}\boldsymbol{N})^{-1}$
$\implies \boldsymbol{A} = \boldsymbol{M}'\boldsymbol{C}\boldsymbol{M}'^{-1}$

When a relation has these three fundamental properties, it is called an equivalence relation. And matrix similarity is an equivalence relation.

12.7.2 Diagonalizability

Since $\boldsymbol{A}$ is similar to its eigenvalue matrix $\boldsymbol{\Lambda}$, if $\boldsymbol{A}$ is diagonalizable, so are the matrices similar to it. In fact, this statement would be a good, albeit incomplete, definition of similarity.

Similarity

Definition: A matrix is similar to another matrix if they diagonalize to the same diagonal matrix.

As we can see, the similarity relation puts diagonalizable matrices in families. In $\mathbb{R}^{n\times n}$, we have an infinity of such mutually exclusive families. All matrices with the same set of eigenvalues belong to the same family.

When we said "the same diagonal matrix" in the definition of similarity above, we were being slightly imprecise: We should have specified that shuffling the eigenvalues is okay. We are really looking for the same set of eigenvalues, regardless of the order.

However, this definition leaves something unspecified: What happens if a matrix is not diagonalizable? Does it belong to no family? Is it similar to none? Is it an orphan? It is in this context that the *Jordan Normal Forms* come in to help. Since it is an important topic, we will promote it to a section of its own.

12.8 Jordan Normal Forms

As we remember, in order for a matrix to be diagonalizable, its eigenvector matrix $\boldsymbol{S}$ needs to be invertible, so that we can go from $\boldsymbol{AS} = \boldsymbol{\Lambda S}$ (which is always true) to $\boldsymbol{A} = \boldsymbol{S\Lambda S^{-1}}$. For $\boldsymbol{S}$ to be invertible, we need its columns to be linearly independent, which means we need a full set of eigenvectors. In other words, for $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, we need n linearly independent eigenvectors for it to be diagonalizable.

We also saw (in §11.4.2, page 200) that the eigenvectors corresponding to distinct eigenvalues are linearly independent. We can, therefore, say that a matrix with no repeated (AKA degenerate) eigenvalues is diagonalizable. The converse is not true though: If a matrix does have repeated eigenvalues, it does not mean that it cannot be diagonalized. The identity matrix in $\mathbb{R}^n$, for instance, has the eigenvalue one repeated n times, but is perfectly diagonalizable. In fact, it is already in the diagonal form. As we shall see shortly, the right statement about diagonalizability is that if the algebraic multiplicity of an eigenvalue is greater than its geometric multiplicity, the matrix cannot be diagonalized.

We looked at such an example earlier, namely the shear matrix, which was not diagonalizable. A general form of such a shear matrix

$\boldsymbol{A} \in \mathbb{R}^{2\times 2}$ is shown below. It has a twice-repeated eigenvalue of λ because the product of the eigenvalues is $|\boldsymbol{A}| = \lambda^2$ and their sum is trace$(\boldsymbol{A}) = 2\lambda$, which can be easily verified by solving its characteristic polynomial[4]. But it has only one eigenvector $\boldsymbol{s}$.

$$\boldsymbol{A} = \begin{bmatrix} \lambda & a \\ 0 & \lambda \end{bmatrix} \implies \lambda_1 = \lambda_2 = \lambda, \boldsymbol{s} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{12.4}$$

In fancier language, the eigenvalue λ has an algebraic multiplicity of two, and a geometric multiplicity of one. It is when the geometric multiplicity is smaller than the algebraic multiplicity that we have a matrix that cannot be diagonalized. Note that the geometric multiplicity can never be greater than the algebraic multiplicity.

The closest we can get to a diagonal matrix for this matrix $\boldsymbol{A}$ in Eqn (12.4) is when we have the repeated eigenvalues along the diagonal, and a one in the position of a. This matrix is called the *Jordan normal* (or *canonical*) *form* of $\boldsymbol{A}$. It is defined as a block matrix in terms of what are known as Jordan blocks.

Joran Normal Form

Definition: A square matrix ($\boldsymbol{J}$) made up of *Jordan blocks* is called a Jordan normal form (JNF) of a matrix $\boldsymbol{A}$ if $\boldsymbol{A} \sim \boldsymbol{J}$.

Jordan Block

Definition:A Jordan block of size k and value λ, $J_k(\lambda)$ is a square matrix with the value λ repeated along its main diagonal and ones along the *superdiagonal* with zeros everywhere else. Here are some examples of Jordan blocks:

$$J_1(\lambda) = \begin{bmatrix} \lambda \end{bmatrix} \quad J_2(\lambda_1) = \begin{bmatrix} \lambda_1 & 1 \\ 0 & \lambda_1 \end{bmatrix} \quad J_3(7) = \begin{bmatrix} 7 & 1 & 0 \\ 0 & 7 & 1 \\ 0 & 0 & 7 \end{bmatrix} \tag{12.5}$$

As we can see, superdiagonal means the diagonal above the main diagonal, so to speak. In the matrix $\boldsymbol{J}$, each eigenvector has a Jordan block of its own, and one block has only one eigenvalue. If we have repeated eigenvalues, but linearly independent eigenvectors, we get multiple Jordan blocks. For instance, for the identity matrix in $\mathbb{R}^{n\times n}$,

[4]The characteristic equation is $|A - \lambda' I| = (\lambda - \lambda')^2 = 0$, with λ' as the dummy variable in the polynomial because we already used λ as the diagonal elements of the shear matrix.

Table 12.2 Jordan Normal Forms $\boldsymbol{J}$ of $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$ with $\boldsymbol{A}\boldsymbol{s_i} = \lambda_i \boldsymbol{s_i}$

	Comments on $\boldsymbol{A}$	Multiplicities Algeb.	Geom.	Jordan Normal Form	Examples of Similar Matrices
1	Most matrices $\text{rank}(\boldsymbol{A}) = 2$	1, 1	1, 1	$\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}$	Any $\boldsymbol{A}$ with trace and $\lvert\boldsymbol{A}\rvert$ equal to sum and product of λ_i
2	$\lambda_2 = 0$, $\boldsymbol{s}_2 \in \mathcal{N}(\boldsymbol{A})$ $\text{rank}(\boldsymbol{A}) = 1$	1, 1	1, 1	$\begin{bmatrix}\lambda_1 & 0\\ 0 & 0\end{bmatrix}$	$\begin{bmatrix}\lambda_1 - t_1 & t_2\\ \frac{\lambda_1 t_1 - t_1^2}{t_2} & t_1\end{bmatrix}$
3	Repeated λ and $\boldsymbol{s}$ $\text{rank}(\boldsymbol{A}) = 2$	2	2	$\begin{bmatrix}\lambda & 0\\ 0 & \lambda\end{bmatrix}$	Only $\boldsymbol{J} = \boldsymbol{A}$
4	$\lambda_1 = \lambda_2 = 0$ $\boldsymbol{s}_1, \boldsymbol{s}_2 \in \mathcal{N}(\boldsymbol{A})$ $\text{rank}(\boldsymbol{A}) = 0$	2	2	$\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}$	Only $\boldsymbol{J} = \boldsymbol{A}$
5	Repeated λ, one $\boldsymbol{s}$ $\text{rank}(\boldsymbol{A}) = 2$	2	1	$\begin{bmatrix}\lambda & 1\\ 0 & \lambda\end{bmatrix}$	$\begin{bmatrix}2\lambda - t_1 & t_2\\ \frac{-\lambda^2 - 2\lambda t_1 - t_1^2}{t_2} & t_1\end{bmatrix}$

In the last column, $t_1, t_2 \in \mathbb{R}$ are any numbers that will generate an example of a similar matrix for the corresponding row. They are constructed such that the sum and product of the eigenvalues come out right. Note that the Jordan normal forms in all rows except the fifth one have two Jordan blocks each.

we have n Jordan blocks, $J_1(1)$, each with $\lambda = 1$. In the examples above in Eqn (12.5), the first one corresponds to a good eigenvalue with an associated eigenvector. The second one has the eigenvalue repeated twice, but with only one eigenvector, much like our shear matrix. The third one is for an eigenvalue with algebraic multiplicity of three and geometric multiplicity of two, to use the right terms.

With these definitions of Jordan normal forms and blocks, we can state the Jordan's Theorem.

Jordan's Theorem

Every square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ with $k \leq n$ linearly independent eigenvectors $\boldsymbol{s_i}, 1 \leq i \leq k$ and the associated eigenvalues $\lambda_i, 1 \leq i \leq k$, which are not necessarily linearly independent, is similar to a Jordan matrix $\boldsymbol{J}$ made up of Jordan blocks along its diagonal.

To start with something simple before generalizing and complicating life, let's look at $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$ with eigenvalues λ_1 and λ_2 and the corresponding eigenvectors $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$. Table 12.2 tabulates the various possibilities. Let's go over each of the rows, and generalize it to from $\mathbb{R}^{2\times 2}$ to $\mathbb{R}^{n\times n}$.

1. The first row is the good case, where we have distinct eigenvalues and linearly independent eigenvectors. The Jordan Normal Form (JNF) is the same as $\mathbf{\Lambda}$. Each λ_i is in a Jordan block $J_1(\lambda_i)$ of its own.

2. When one of the two eigenvalues is zero, the matrix is singular, and one of the eigenvectors is the basis of its null space. Qualitatively though, this case is not different from the first row. In $\mathbb{R}^{n\times n}$, we will have JNF $= \mathbf{\Lambda}$.

3. When the eigenvalues are repeated (meaning algebraic multiplicity is two), but we have two linearly independent eigenvectors. Again, JNF $= \mathbf{\Lambda}$. However, we have no other similar matrices: $\boldsymbol{MJM^{-1}} = \lambda \boldsymbol{MIM^{-1}} = \lambda \boldsymbol{I} = \boldsymbol{J}$.

4. This row shows a rank-zero matrix. Although troublesome, it is also qualitatively similar to the previous row.

5. In the last row, we have the geometric multiplicity smaller than the algebraic one, and we have a $J_2(\lambda)$.

In order to show the Jordan form in all its gory detail, we have a general Jordan matrix made up of a large number of Jordan blocks in Eqn (12.6) below:

$$\boldsymbol{J} = \begin{bmatrix} \boxed{\lambda_1}^{J_1(\lambda_1)} & & & & & & & & \\ 0 & \boxed{\lambda_1}^{J_1(\lambda_1)} & & & & & & & \\ 0 & 0 & \boxed{\lambda_1}^{J_1(\lambda_1)} & & & & & & \\ 0 & 0 & 0 & \boxed{\begin{matrix}\lambda_2 & 1 & \\ 0 & \lambda_2 & 1 \\ 0 & 0 & \lambda_2\end{matrix}}^{J_3(\lambda_2)} & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \boxed{\lambda_3}^{J_1(\lambda_3)} & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boxed{\begin{matrix}\lambda_4 & 1 \\ 0 & \lambda_4\end{matrix}}^{J_2(\lambda_4)} & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ddots \end{bmatrix} \tag{12.6}$$

Each Jordan block is highlighted in a colored box with its label $J_k(\lambda)$. Here is what this Jordan matrix $\boldsymbol{J}$ tells us about the original matrix $\boldsymbol{A}$ (of which $\boldsymbol{J}$ is the Jordan Normal Form):

- The eigenvalues of $\boldsymbol{J}$ and $\boldsymbol{A}$ are the same: λ_i.
- The algebraic multiplicity of any eigenvalue is the *size* of the Jordan blocks associated with it.
- Its geometric multiplicity is the *number* of Jordan blocks associated with it.

- $\boldsymbol{A}$ can be diagonalized if and only if each Jordan block in $\boldsymbol{J}$ is of size one. In other words, $\boldsymbol{J}$ needs to be diagonal for $\boldsymbol{A}$ to be diagonalizable.

Table 12.3 lists the multiplicities of the eigenvalues shown in Eqn (12.6). Since $\boldsymbol{J}$ has some elements in the superdiagonal, $\boldsymbol{A}$ is not diagonalizable. The closest $\boldsymbol{A}$ can get to a diagonal matrix is indeed its Jordan normal form $\boldsymbol{J}$.

Table 12.3 Multiplicities of eigenvalues from Joran blocks

Eigenvalue	Multiplicities	
λ_i	Algebraic	Geometric
λ_1	3	3
λ_2	3	1
λ_3	1	1
λ_4	2	1

Much like the other topics in this chapter, Jordan canonical form also is a portal, this time to advanced theoretical explorations in Linear Algebra, perhaps more relevant to mathematicians than computer scientists.

12.9 Algorithms

We already learned three named algorithms, namely,

1. Gaussian Elimination, also known as $\boldsymbol{PLU}$ or just $\boldsymbol{LU}$ decomposition.

2. Gauss-Jordan Elimination, for matrix inversion.

3. Gram-Schmidt Orthonormalization, which gave us $\boldsymbol{QR}$ decomposition.

They are neatly summarized in Table 8.1 and recapped in the associated text. We came across one more algorithm earlier in this chapter (see §12.4.2, page 232), where we computed one eigenvector corresponding to $\lambda = 1$. The method is called the Power Iteration Algorithm, and can, in fact, find the largest eigenvalue/eigenvector pair.

12.9.1 Power Iteration

We saw power iteration in our Markov matrix, migration patterns, example. In general, the power iteration algorithm returns the dominant eigenvalue (which is the one with the largest absolute value). The process is simple: We start with a random initial vector, apply the matrix to it, renormalize it and iterate.

Input: $\boldsymbol{A} \in \mathbb{R}^{n \times n}$
Output: Dominant $\lambda \in \mathbb{C}$
1: *Start with a random* $\boldsymbol{s}$
2: **repeat**
3: *Normalize it:* $\boldsymbol{s} \leftarrow \frac{\boldsymbol{s}}{\|\boldsymbol{s}\|}$
4: $\boldsymbol{s} \leftarrow \boldsymbol{A}\boldsymbol{s}$
5: **until** *Convergence*
6: **return** $\boldsymbol{s}$ *is the eigenvector for the largest* λ

Once we have $\boldsymbol{s}$, we can calculate λ as:

$$\lambda = \frac{\boldsymbol{s}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{s}}{\boldsymbol{s}^{\mathsf{T}}\boldsymbol{s}} \quad \text{(Because } \boldsymbol{A}\boldsymbol{s} = \lambda\boldsymbol{s}\text{)}$$

which, by the way, is called the Rayleigh quotient.

Limitations The power iteration algorithm has a couple of limitations[5].

1. If we start with a bad guess for the initial vector, the power iteration algorithm may not converge to the dominant eigenvalue.
2. For complex eigenvalues, we have to start with a complex initial vector. Otherwise, the algorithm may oscillate between the two conjugate eigenvalue pairs.

12.9.2 $\boldsymbol{QR}$ Algorithm

A general numerical method to compute all eigenvalues and eigenvectors of a matrix is the $\boldsymbol{QR}$ algorithm, based on the Gram-Schmidt process.

Input: $\boldsymbol{A} \in \mathbb{R}^{n \times n}$
Output: $\lambda_i \in \mathbb{C}$
1: **repeat**
2: *Perform the Gram-Schmidt process to get* $\boldsymbol{QR}$

[5] From Wiki University.

3: *We get:* $\boldsymbol{A} = \boldsymbol{QR}$
4: *Consider:* $\boldsymbol{RQ} = \boldsymbol{Q^{-1}QRQ} = \boldsymbol{Q^{-1}AQ}$
5: $\implies$ $\boldsymbol{RQ}$ *and* $\boldsymbol{A}$ *are similar and have the same eigenvalues*
6: *Therefore, set* $\boldsymbol{A} \leftarrow \boldsymbol{RQ}$
7: **until** *Convergence*
8: **return** *The diagonal elements of* $\boldsymbol{R}$ *as the eigenvalues*

Convergence is obtained when $\boldsymbol{A}$ becomes close enough to a triangular matrix. At that point, the eigenvalues are its diagonal elements. Once we have the eigenvalues, we can compute the eigenvectors as the null space of $\boldsymbol{A} - \lambda \boldsymbol{I}$ using elimination algorithms.

12.9.3 Cholesky Decomposition

A matrix factorization technique that has several applications is the Cholesky Decomposition, which says that any positive definite matrix $\boldsymbol{A}$ can be written as the product of a lower triangular matrix and its transpose. We are, once again, dealing with real matrices, but this decomposition applies to complex matrices as well.

$$\boldsymbol{A} = \boldsymbol{LL^{\mathsf{T}}}$$

The algorithm to perform this factorization is written in terms of block matrices. We first partition $\boldsymbol{A}$ and $\boldsymbol{L}$ into blocks as shown in Eqn (12.7). We keep the element in the first row, first column as one block, and partition the rest of each matrix into a column vector ($\boldsymbol{a_{21}}, \boldsymbol{l_{21}} \in \mathbb{R}^{n-1}$), its transpose and a smaller matrix like $\boldsymbol{A_{22}}, \boldsymbol{L_{22}} \in \mathbb{R}^{(n-1)\times(n-1)}$.

$$\boldsymbol{A} = \left[\begin{array}{c|c} a_{11} & \boldsymbol{a_{21}^{\mathsf{T}}} \\ \hline \boldsymbol{a_{21}} & \boldsymbol{A_{22}} \end{array}\right] \qquad \boldsymbol{L} = \left[\begin{array}{c|c} l_{11} & \boldsymbol{0^{\mathsf{T}}} \\ \hline \boldsymbol{l_{21}} & \boldsymbol{L_{22}} \end{array}\right] \tag{12.7}$$

Note that we used the fact that $\boldsymbol{A}$ is symmetric in dividing up the top row of $\boldsymbol{A}$ as $[a_{11}\ \boldsymbol{a_{21}^{\mathsf{T}}}]$. Similarly, we could write the first row of $\boldsymbol{L}$ as $[l_{11}\ \boldsymbol{0^{\mathsf{T}}}]$ because it is lower triangular. And, we do not have to worry about the second factor $\boldsymbol{L^{\mathsf{T}}}$ at all because it is just the transpose of $\boldsymbol{L}$.

$$\boldsymbol{A} = \boldsymbol{LL^{\mathsf{T}}} \implies \left[\begin{array}{c|c} a_{11} & \boldsymbol{a_{21}^{\mathsf{T}}} \\ \hline \boldsymbol{a_{21}} & \boldsymbol{A_{22}} \end{array}\right] = \left[\begin{array}{c|c} l_{11} & \boldsymbol{0^{\mathsf{T}}} \\ \hline \boldsymbol{l_{21}} & \boldsymbol{L_{22}} \end{array}\right] \left[\begin{array}{c|c} l_{11} & \boldsymbol{l_{21}^{\mathsf{T}}} \\ \hline \boldsymbol{0} & \boldsymbol{L_{22}^{\mathsf{T}}} \end{array}\right]$$

Expanding the matrix multiplication $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^\mathsf{T}$ using the block matrices, and comparing the corresponding elements in $\boldsymbol{A}$ and $\boldsymbol{L}\boldsymbol{L}^\mathsf{T}$:

$$\begin{aligned}
\left[\begin{array}{c|c} a_{11} & \boldsymbol{a_{21}^\mathsf{T}} \\ \hline \boldsymbol{a_{21}} & \boldsymbol{A_{22}} \end{array}\right] &= \left[\begin{array}{c|c} l_{11}^2 & l_{11}\boldsymbol{l_{21}} \\ \hline l_{11}\boldsymbol{l_{21}} & \boldsymbol{l_{21}}\boldsymbol{l_{21}^\mathsf{T}} + \boldsymbol{L_{22}}\boldsymbol{L_{22}^\mathsf{T}} \end{array}\right] \\
a_{11} = l_{11}^2 &\implies l_{11} = \sqrt{a_{11}} \\
\boldsymbol{a_{21}} = l_{11}\boldsymbol{l_{21}} &\implies \boldsymbol{l_{21}} = \frac{\boldsymbol{a_{21}}}{l_{11}} = \frac{\boldsymbol{a_{21}}}{\sqrt{a_{11}}} \\
\boldsymbol{A_{22}} = \boldsymbol{l_{21}}\boldsymbol{l_{21}^\mathsf{T}} + \boldsymbol{L_{22}}\boldsymbol{L_{22}^\mathsf{T}} &\implies \boldsymbol{A_{22}} - \boldsymbol{l_{21}}\boldsymbol{l_{21}^\mathsf{T}} = \boldsymbol{L_{22}}\boldsymbol{L_{22}^\mathsf{T}}
\end{aligned} \tag{12.8}$$

The second and third last lines in Eqn (12.8) tell us the elements of $\boldsymbol{L}$. Note that we decide to go with the positive square root for l_{11}. The very last line tells us that once we got l_{11} and $\boldsymbol{l_{21}}$, the problem reduces to computing the Cholesky decomposition of a smaller matrix $\boldsymbol{A'} = \boldsymbol{A_{22}} - \boldsymbol{l_{21}}\boldsymbol{l_{21}^\mathsf{T}}$.

Before we write it down as a formal algorithm, the only thing left to do is to ensure that $\boldsymbol{A'}$ is positive definite. Otherwise, we are not allowed to assume that we can find $\boldsymbol{A'} = \boldsymbol{L'}\boldsymbol{L'^\mathsf{T}}$. In the fourth test for positive definiteness (on 235), we proved that the upper-left submatrices of a positive definite matrix were also positive definite. If we look closely at the proof, we can see that we did not need to confine ourselves to "upper-left": Any submatrix sharing the main diagonal is positive definite. Therefore, if $\boldsymbol{A}$ in our Cholesky factorization is positive definite, so is $\boldsymbol{A_{22}}$. We also saw that sums of positive definite matrices are positive definite. Extending it to nontrivial (meaning, nonzero) differences, we can see that $\boldsymbol{A'} = \boldsymbol{A_{22}} - \boldsymbol{l_{21}}\boldsymbol{l_{21}^\mathsf{T}}$ is positive definite[6].

Looking at Eqn (12.8), we can translate it to an algorithm[7] as below:

Input: $\boldsymbol{A} \in \mathbb{R}^{n\times n}$
Output: $\boldsymbol{L} \in \mathbb{R}^{n\times n}$

1: *Initialize* $\boldsymbol{L}$ *with zeros*
2: **repeat**
3: *Divide* $\boldsymbol{A}$ *into block-matrices as in Eqn* (12.7)
4: $l_{11} \leftarrow \sqrt{a_{11}}$

[6] To be very precise, $\boldsymbol{l_{21}}\boldsymbol{l_{21}^\mathsf{T}}$ is a rank-one matrix, and is only positive semidefinite. But the difference is still positive definite.
[7] This description and the algorithm listed are based on the discussion in this video.

5: $\boldsymbol{l_{21}} \leftarrow \frac{\boldsymbol{a_{21}}}{l_{11}}$
6: $\boldsymbol{A} \leftarrow \boldsymbol{A_{22}} - \boldsymbol{l_{21}}\boldsymbol{l_{21}^{\mathsf{T}}}$
7: **until** $\boldsymbol{A_{22}}$ *becomes a scalar*
8: **return** *The matrix* $\boldsymbol{L}$

Applications of Cholesky Decomposition

1. The $\boldsymbol{LL^{\mathsf{T}}}$ decomposition is faster in solving systems of linear equations than Gauss-Jordan by about a factor of two. The catch is that it applies only when the coefficient matrix $\boldsymbol{A}$ is positive semidefinite.

2. In simulation programs, we usually face the problem of having to work with multivariate normal distribution, where we need to simulate correlated, normally distributed variables, which are specified by their covariance matrix ($\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$) with means ($\boldsymbol{\mu} \in \mathbb{R}^n$). The standard way of accomplishing this task is to draw the required number (n) random variables for a standard normal distribution (with $\sigma = 1, \mu = 0$) and multiply this random vector ($\boldsymbol{x} \in \mathbb{R}^n$) by the lower triangular matrix $\boldsymbol{L}$ and shift it by $\boldsymbol{\mu}$, where $\boldsymbol{L}$ comes from the Cholesky Decomposition of $\boldsymbol{\Sigma} = \boldsymbol{LL^{\mathsf{T}}}$. The required vector of random numbers would then be $\boldsymbol{Lx} + \boldsymbol{\mu}$, which would have the right covariance matrix ($\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$) and means ($\boldsymbol{\mu} \in \mathbb{R}^n$).

13

Singular Value Decomposition

What is best in mathematics deserves not merely to be learnt as a task, but to be assimilated as a part of daily thought, and brought again and again before the mind with ever-renewed encouragement.

—Bertrand Russell

Singular Value Decomposition (SVD) has come into prominence in the last couple of decades because of its direct applicability in algorithms in computer science, especially when dealing with large volumes of data. It finds applications in, for instance, data compression, dimensionality reduction, principal component analysis etc. From a mathematical perspective, SVD is a topic that embodies pure elegance. And it brings a large part of what we discussed in this book (eigenanalysis, vector spaces and bases, orthogonality, the four fundamental subspaces etc.) together in one cohesive unity. In many more ways than one, therefore, Singular Value Decomposition is appropriate as a closing chapter for an introductory book on Linear Algebra.

13.1 What SVD Does

In eigenvalue decomposition (EVD), we took a square matrix $\boldsymbol{A}$, and wrote it as:

$$\boldsymbol{AS} = \boldsymbol{S\Lambda}$$

where $\boldsymbol{\Lambda}$ is a matrix with eigenvalues in the diagonal, $\boldsymbol{S}$ is the matrix where we have the eigenvectors as columns. For real, symmetric matrices, we saw that $\boldsymbol{S}$ was orthogonal, whose transpose is the same as its inverse, and we wrote:

$$\boldsymbol{A} = \boldsymbol{Q\Lambda Q}^{\mathsf{T}}$$

Since an orthogonal matrix is a rotation, we can interpret this decomposition geometrically. In $\boldsymbol{Ax} = \boldsymbol{Q\Lambda Q}^{\mathsf{T}}\boldsymbol{x}$, what $\boldsymbol{A}$ does to $\boldsymbol{x}$ (or, how it transforms $\boldsymbol{x}$) is the same as a rotation ($\boldsymbol{Q}^{\mathsf{T}}$), followed by a scaling ($\boldsymbol{\Lambda}$) and then another rotation ($\boldsymbol{Q}$) by the same angle(s) in the opposite direction.

This interpretation of eigenvalue decomposition (EVD), which we called the Spectral Theorem, works only for real, symmetric matrices. In fact, the computation of eigenvalue and eigenvectors can be done only for a square matrices. Given that the data matrices that we deal with are far from square ones, these restrictions of EVD severely restrict its applicability to computer science. What SVD does is to decompose *any* matrix into a rotation, followed by a scaling, followed by another rotation, viewing $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ as a transformation of $\boldsymbol{x}$, taking it from $\mathbb{R}^n$ to $\mathbb{R}^m$.

To reiterate, when we apply SVD, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is any matrix. It does not have to be symmetric, not even square. Although it does not need to be real either, we will again worry only about real matrices. To assign symbols to these verbose statements:

$$\begin{gathered}\boldsymbol{A} \in \mathbb{R}^{m\times n} \xrightarrow{\text{SVD}} \boldsymbol{A} = \boldsymbol{U\Sigma V}^{\mathsf{T}} \\ \boldsymbol{U} \in \mathbb{R}^{m\times m},\ \boldsymbol{\Sigma} \in \mathbb{R}^{m\times n},\ \boldsymbol{V} \in \mathbb{R}^{n\times n}\end{gathered} \tag{13.1}$$

$\boldsymbol{\Sigma}$ is a diagonal matrix (or as diagonal as a non-square matrix can be), with positive values along the main diagonal, starting from σ_1 all the way to σ_r, where $r = \min(m, n, \text{rank}(\boldsymbol{A}))$, with the rest of the elements all zero. Note also that the action of a diagonal matrix $[\sigma_i]$ on a matrix $\boldsymbol{X}$ multiplying it on the right is to scale the i^{th} row of $\boldsymbol{X}$ by σ_i.

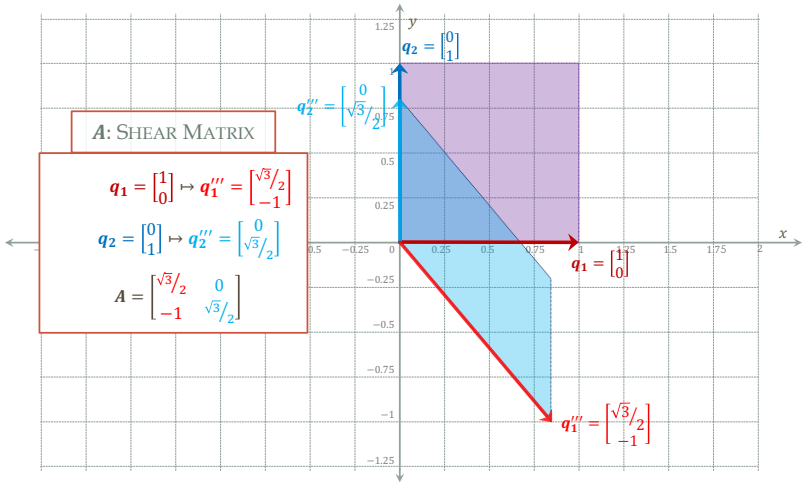


Fig. 13.1 The shear matrix under SVD analysis.

Before learning how SVD does its magic, let's take a look at an example. For ease of visualization, we will work with a square matrix so that we can draw the vectors and their transformations in $\mathbb{R}^2$.

$$\boldsymbol{A} = \begin{bmatrix} \frac{\sqrt{3}}{2} & 0 \\ -1 & \frac{\sqrt{3}}{2} \end{bmatrix} \xrightarrow{\text{SVD}} \boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}}$$

$$\boldsymbol{U} = \begin{bmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \frac{3}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \quad \boldsymbol{V} = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}$$

A few points to note about these matrices:

- We do not yet know how we got the decomposition $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}}$, but we can verify its validity by multiplying. In fact, what we did in coming up with the "decomposition" above was to start from the $\boldsymbol{U}, \boldsymbol{\Sigma}$ and $\boldsymbol{V}$ matrices and take their product to get $\boldsymbol{A}$ as $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}}$.

- $\boldsymbol{U}$ is indeed a rotation, as described in Figure 7.4, with the angle of rotation $\theta = -\frac{\pi}{3}$, a clockwise rotation through 60°.

- Similarly, $\boldsymbol{V}$ is a rotation $\theta = -\frac{\pi}{6}$, or clockwise 30°. Therefore, $\boldsymbol{V}^{\mathsf{T}}$ (being the same as $\boldsymbol{V}^{-1}$) is an anti-clockwise rotation of 30°.

- Most importantly, as shown in Figure 13.1, the matrix $\boldsymbol{A}$ is a vertical shear matrix: It shears the x-component down (proportional to the y component), and leaves the y component alone, but for some scaling. As we learned earlier (in §11.2.3, page 193 and further explanation on page 196), shear matrices do not have a full set of eigenvectors and they cannot be diagonalized. But, they do have a singular value decomposition, as do all matrices.

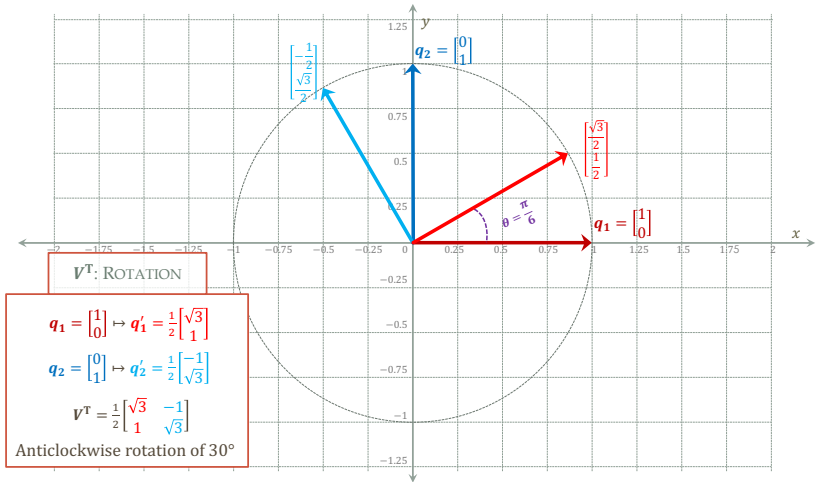


Fig. 13.2 The first rotation by $\boldsymbol{V^T}$, anti-clockwise rotation through 30°.

Figures 13.2 to 13.5 show the actions of these three matrices, and a summary. The transformation a vector $\boldsymbol{x}$ by $\boldsymbol{Ax}$ is broken down into $\boldsymbol{U\Sigma V^T x}$. The first multiplication by $\boldsymbol{V^T}$ is a rotation, shown in Figure 13.2. It does not change the size of any vector $\boldsymbol{x}$, nor of the basis vectors $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ shown in red and blue. In their new, rotated positions, $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ are shown in lighter red and blue. The unit vectors all have their tips on the unit circle, as shown in Figure 13.2. Because of the rotation (of 30°) by $\boldsymbol{V^T}$, all the vectors in the first quadrant are now between the bright red and blue vectors $\boldsymbol{q_1'}$ and $\boldsymbol{q_2'}$.

We then apply the scaling $\boldsymbol{\Sigma}$ on the product $\boldsymbol{V^T x}$, which scales along the *original* (*not* the rotated) unit vectors $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$. In Figure 13.3, we can see how the rotated unit vectors (now shown in translucent red and blue) get transformed into their new versions. What $\boldsymbol{\Sigma}$ does to the unit vectors $\boldsymbol{q_i}$, it does to all vectors. Therefore,

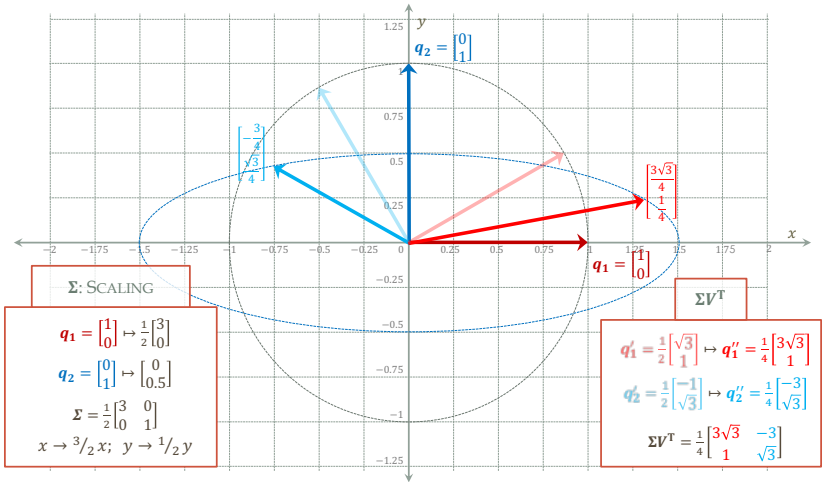


Fig. 13.3 The second scaling by Σ, x-components by 1.5 and y by 0.5.

the unit circle gets elongated along the x direction, and squashed along the y direction. What we mean by this statement is that all vectors whose tips are on the unit circle get transformed such that their tips end up on the said ellipse. As a part of this transformation, the rotated unit vectors, the translucent red and blue vectors $\boldsymbol{q_1'}$ and $\boldsymbol{q_2'}$, get transformed to $\boldsymbol{q_1''}$ and $\boldsymbol{q_2''}$ (in brighter colors) on the ellipse. In other words, the effect of the two transformations, the product $\boldsymbol{\Sigma V^T}$, is to move all the vectors in the first quadrant of the unit circle to the arc of the ellipse between the bright red and blue vectors $\boldsymbol{q_1''}$ and $\boldsymbol{q_2''}$.

Notice that the transformed ellipse in Figure 13.3 has its axes along the x and y directions. The last step, shown in Figure 13.4, is the rotation by $\boldsymbol{U}$. It is a anti-clockwise rotation of 60°. It rotates the unit vectors through that angle. Remembering that the axes of the ellipse after the scaling (in Figure 13.3) were along the directions of x and y unit vectors, we can see that how the ellipse gets rotated. Of course, the rotation happens to all the vectors. The $\boldsymbol{\Sigma V^T}$-transformed versions of the original unit vectors (from Figure 13.2), now shown in translucent red and blue in Figure 13.4 as $\boldsymbol{q_1''}$ and $\boldsymbol{q_2''}$, for instance, get rotated to the bright red vector, with its tip at $(\frac{\sqrt{3}}{2}, -1)$ and the bright blue vector with its tip at $(0, \frac{\sqrt{3}}{2})$. This indeed is exactly what the shear matrix $\boldsymbol{A}$ does, as illustrated in Figure 13.1.

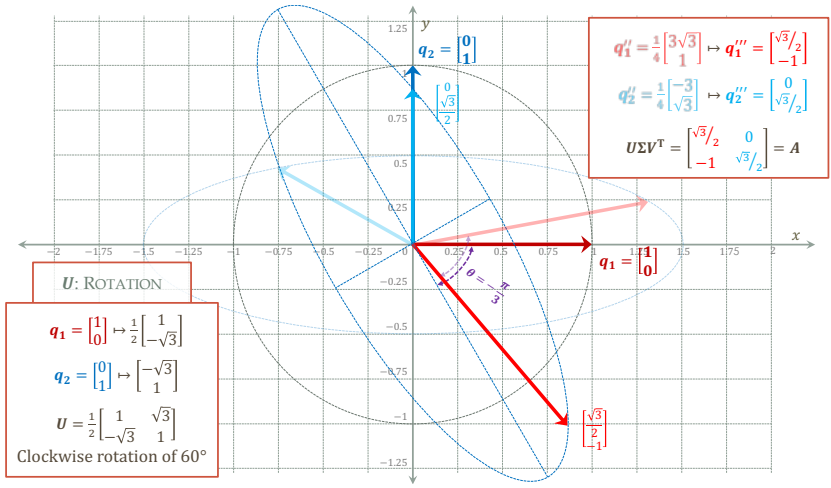


Fig. 13.4 The third rotation by $\boldsymbol{U}$, clockwise rotation through 60°.

Now we are in a position to summarize what $\boldsymbol{A}$ does, either through the direct path $\boldsymbol{Ax}$ or by the three steps $\boldsymbol{U\Sigma V^{\mathsf{T}}x}$, as shown in Figure 13.5.

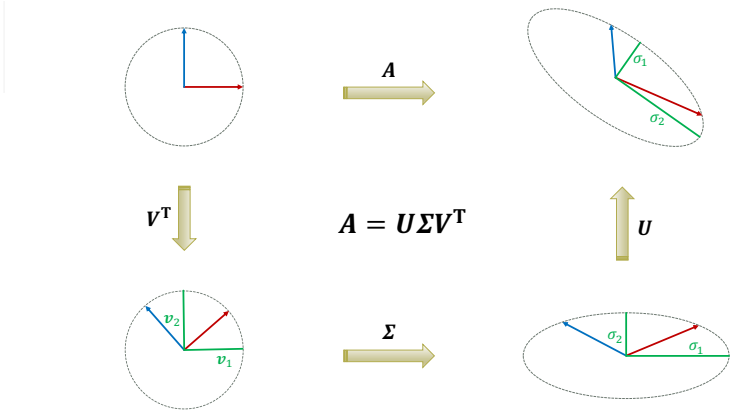


Fig. 13.5 The geometry of SVD, summarized.

We may be tempted to think, from the figures, that one rotation and one scaling might be enough to do everything that $\boldsymbol{A}$ does. We should, however, note that the first quadrant of the unit circle in Figure 13.2 is getting mapped to the arc of the ellipse between the light red and blue vectors in Figure 13.3. One rotation and one scaling could give is this mapping, but the ellipse would have its axes along the unit

vectors. Or, by applying a scaling and then a rotation, we could get to the right ellipse as in Figure 13.4, left panel, but the points in the first quadrant on the unit circle would be mapped to the first quadrant of the ellipse. We really do need two rotations and a scaling.

The story in the general case of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, of course, is much more complicated, but the fundamental idea is still the same. We use the intuitions from $\mathbb{R}^2$ in order to make use of the power of SVD in higher dimensions as well.

13.2 How SVD Works

From our discussions on the four fundamental subspaces defined by a matrix $\boldsymbol{A}$, we know that its row space $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$ gets mapped to its column space $\mathcal{C}(\boldsymbol{A})$. Furthermore, the mapping is one-to-one and onto, which means a vector $\boldsymbol{v} \in \mathcal{C}(\boldsymbol{A}^\mathsf{T})$ gets mapped to a unique vector $\boldsymbol{u} \in \mathcal{C}(\boldsymbol{A})$. Both $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$ and $\mathcal{C}(\boldsymbol{A})$ have the same dimension, which is the rank of $\boldsymbol{A}, r$. In other words, we should be able to find r orthogonal vectors in $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$ to span the row space; we could, for instance, apply the Gram-Schmidt algorithm to do it.

Let's say that these r orthonormal basis vectors in $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$ are $\boldsymbol{v_i}$. When applying the transformation of $\boldsymbol{A}$ to them, we get some vectors in $\mathcal{C}(\boldsymbol{A})$: $\boldsymbol{A v_i} = \boldsymbol{u_i}$. We know that these are r *unique* vectors $\boldsymbol{u_i}$ because $\boldsymbol{A} : \mathcal{C}(\boldsymbol{A}^\mathsf{T}) \mapsto \mathcal{C}(\boldsymbol{A})$ is a one-to-one mapping. However, we have no reason to think that they are orthogonal or normalized. We can insist that they be normalized by factoring out their norms: $\boldsymbol{A v_i} = \sigma_i \boldsymbol{u_i}$ with $\|\boldsymbol{u_i}\| = 1$.

The vectors $\boldsymbol{u_i}$ are unique because each of them is a different linear combination of the columns of $\boldsymbol{A}$ with the coefficients as specified in $\boldsymbol{v_i}$ and the vectors $\boldsymbol{v_i}$ are linearly independent. Taking this as one of the last teachable moments in this textbook, let's harp on this point a bit more. Since $\boldsymbol{v_i} \in \mathcal{C}(\boldsymbol{A}^\mathsf{T})$, we know that $\boldsymbol{A v_i} \neq \boldsymbol{0}$. And since the vectors $\boldsymbol{v_i}$ form a basis for $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$, they are linearly independent, which means, at the very least, the components of one of them cannot all be the same as that of another. By the column picture of matrix multiplication, $\boldsymbol{u_i} = \boldsymbol{A v_i}$ is a linear combination of the columns of $\boldsymbol{A}$ with different coefficients that are components of $\boldsymbol{v_i}$. And, from way back in the first or second chapter, we know that two different

linear combinations $\sum s_i \boldsymbol{v_i}$ and $\sum t_i \boldsymbol{v_i}$ cannot be the same: Linear combinations are unique.

What we are demanding in Singular Value Decomposition (SVD) of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\text{rank}(\boldsymbol{A}) = r \leq \min(m, n)$ is something remarkable: Our goal is to find a *special* orthonormal basis in the row space of $\boldsymbol{A}$ that gets mapped to an orthonormal basis in its column space: We want a basis $\boldsymbol{V}$ for $\mathcal{C}(\boldsymbol{A^\mathsf{T}})$, and a $\boldsymbol{U}$ for $\mathcal{C}(\boldsymbol{A})$, with a *special* requirement each $\boldsymbol{v_i} \in \mathcal{C}(\boldsymbol{A^\mathsf{T}})$ gets mapped to a $\boldsymbol{u_i} \in \mathcal{C}(\boldsymbol{A})$, with a possible scaling factor σ_i. The remarkable fact behind SVD is that we will *always* be able find such bases, and we shall shortly see why and how. We will have a mathematical proof and recipe to find such bases.

Since the rank of the matrix is r, we can find only r linearly independent vectors in $\mathcal{C}(\boldsymbol{A})$ and $\mathcal{C}(\boldsymbol{A^\mathsf{T}})$. We will worry about the rest of the column vectors in $\boldsymbol{U}$ and $\boldsymbol{V}$ later.

Let's write down all we have so far:

$$\hat{\boldsymbol{U}} = \begin{bmatrix} | & | & \cdots & | \\ \boldsymbol{u_1} & \boldsymbol{u_2} & \cdots & \boldsymbol{u_r} \\ | & | & \cdots & | \end{bmatrix} \in \mathbb{R}^{m \times r} = [\boldsymbol{u_i}]$$

$$\hat{\boldsymbol{V}} = \begin{bmatrix} | & | & \cdots & | \\ \boldsymbol{v_1} & \boldsymbol{v_2} & \cdots & \boldsymbol{v_r} \\ | & | & \cdots & | \end{bmatrix} \in \mathbb{R}^{n \times r} = [\boldsymbol{v_i}]$$

We are putting a hat on the $\boldsymbol{U}$ and $\boldsymbol{V}$ matrices because they are smaller, more economical ones for now. They are not the full-sized versions in Eqn (13.1). Each $\boldsymbol{v_i} \in \mathcal{C}(\boldsymbol{A^\mathsf{T}})$ is going to be mapped to a corresponding $\boldsymbol{u_i} \in \mathcal{C}(\boldsymbol{A})$, as per our requirement. We then have the following, where we have arranged $\sigma_i, 1 \leq i \leq r$ as the diagonal elements in $\hat{\boldsymbol{\Sigma}} \in \mathbb{R}^{r \times r}$. Remember that we require the columns of $\hat{\boldsymbol{V}}$ and $\hat{\boldsymbol{U}}$ to be orthonormal.

$$\boldsymbol{A}\boldsymbol{v_i} = \sigma_i \boldsymbol{u_i} \implies \boldsymbol{A}\hat{\boldsymbol{V}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}$$

Remembering that for a matrix with orthonormal columns, we know that the product of its transpose with itself is the identity matrix. For instance, the product $\hat{\boldsymbol{U}}^\mathsf{T}\hat{\boldsymbol{U}}$ for $\hat{\boldsymbol{U}} \in \mathbb{R}^{m \times r}$ is $\boldsymbol{I}_r \in \mathbb{R}^{r \times r}$ because the elements in the product are the dot products of the columns of $\hat{\boldsymbol{U}}$, which are either zero or one. We should not, however, call $\hat{\boldsymbol{U}}$

orthonormal because it is not a square matrix. Same same argument applies to $\hat{\boldsymbol{V}}^\mathsf{T}\hat{\boldsymbol{V}}$ as well.

We can now do a bit of matrix algebra magic to compute $\hat{\boldsymbol{U}}$ and $\hat{\boldsymbol{V}}$

(1) Orthonormal requirement:	$\boldsymbol{A}\hat{\boldsymbol{V}}$	$= \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}$
(2) Taking the transpose of (1):	$\hat{\boldsymbol{V}}^\mathsf{T}\boldsymbol{A}^\mathsf{T}$	$= \hat{\boldsymbol{\Sigma}}^\mathsf{T}\hat{\boldsymbol{U}}^\mathsf{T}$
(3) Multiplying (2) and (1):	$\hat{\boldsymbol{V}}^\mathsf{T}\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\hat{\boldsymbol{V}}$	$= \hat{\boldsymbol{\Sigma}}^\mathsf{T}\hat{\boldsymbol{U}}^\mathsf{T}\hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}$
(4) Since $\hat{\boldsymbol{U}}^\mathsf{T}\hat{\boldsymbol{U}} = \boldsymbol{I}$:	$\hat{\boldsymbol{V}}^\mathsf{T}\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\hat{\boldsymbol{V}}$	$= \hat{\boldsymbol{\Sigma}}^\mathsf{T}\hat{\boldsymbol{\Sigma}}$
(5) Since $\hat{\boldsymbol{\Sigma}}$ is diagonal:	$\hat{\boldsymbol{V}}^\mathsf{T}\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\hat{\boldsymbol{V}}$	$= \hat{\boldsymbol{\Sigma}}^2$
(6) Left and right multiply by $\hat{\boldsymbol{V}}$ and $\hat{\boldsymbol{V}}^\mathsf{T}$:	$\hat{\boldsymbol{V}}\hat{\boldsymbol{V}}^\mathsf{T}\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\hat{\boldsymbol{V}}\hat{\boldsymbol{V}}^\mathsf{T}$	$= \hat{\boldsymbol{V}}\hat{\boldsymbol{\Sigma}}^2\hat{\boldsymbol{V}}^\mathsf{T}$
(7) If $\hat{\boldsymbol{V}}\hat{\boldsymbol{V}}^\mathsf{T} = \boldsymbol{I}$:	$\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$	$= \hat{\boldsymbol{V}}\hat{\boldsymbol{\Sigma}}^2\hat{\boldsymbol{V}}^\mathsf{T}$

In statement (7), we specify a conditional: If $\hat{\boldsymbol{V}}\hat{\boldsymbol{V}}^\mathsf{T} = \boldsymbol{I}$, then we get an equation involving $\hat{\boldsymbol{V}}$ and $\hat{\boldsymbol{\Sigma}}^2$, where we can identify an eigenvalue equation, much like the Spectral theorem, $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\mathsf{T}$. It then follows that the singular value decomposition is the eigenvalue decomposition of the product $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$ and the singular values are the positive square roots of its eigenvalues.

The condition $\hat{\boldsymbol{V}}\hat{\boldsymbol{V}}^\mathsf{T} = \boldsymbol{I}$ is a problem though. We proved that $\hat{\boldsymbol{V}}^\mathsf{T}\hat{\boldsymbol{V}} = \boldsymbol{I}$. Can we also show that $\hat{\boldsymbol{V}}\hat{\boldsymbol{V}}^\mathsf{T} = \boldsymbol{I}$? It turns out that we cannot, for one very good reason: It is not true. One way to get around this difficulty is to consider only full-column-rank matrices $\boldsymbol{A}$, in which case we have $\text{rank}(\boldsymbol{A}) = r = n$ and a square matrix $\boldsymbol{V} \in \mathbb{R}^{n\times n}$. We will drop the hat on $\boldsymbol{V}$ because it is now the basis for the whole of the domain of $\boldsymbol{A}$, namely $\mathbb{R}^n$. What we now have are the matrices as shown in Figure 13.6. Following the same logic, we can show that

$$\boldsymbol{A}\boldsymbol{A}^\mathsf{T} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}^2\hat{\boldsymbol{U}}^\mathsf{T}$$

with the proviso that $\boldsymbol{A}$ be a full-row-rank matrix in this case.

As for $\hat{\boldsymbol{\Sigma}}^2 = [\sigma_i^2]$, it is the square of a diagonal matrix. It is, therefore, simply another matrix with the squared elements along the diagonal.

Comparing these results with our eigenvalue equations in the Spectral Theorem (§12.1.1, page 226), we can see that these are the eigenvalue equations for $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$ and $\boldsymbol{A}\boldsymbol{A}^\mathsf{T}$. Therefore, we conclude that the *left-singular vectors* $\boldsymbol{u_i}$ are the eigenvectors of $\boldsymbol{A}\boldsymbol{A}^\mathsf{T}$ and the *right-*

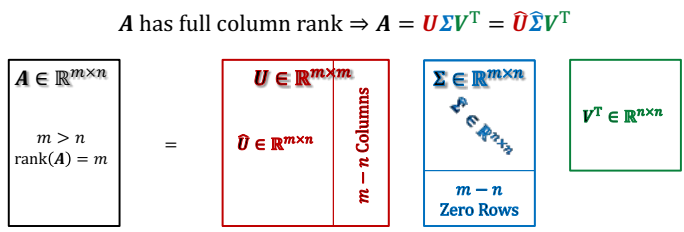


Fig. 13.6 The shapes of $\boldsymbol{U}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{V}$ in SVD when $\boldsymbol{A}$ is a full-column-rank matrix.

singular vectors $\boldsymbol{v_i}$ those of $\boldsymbol{A^\mathsf{T}A}$. And the *singular values* are the square root of the eigenvalues of either $\boldsymbol{A^\mathsf{T}A}$ or $\boldsymbol{AA^\mathsf{T}}$. As we saw in one of the exercises in the previous chapter, the nonzero eigenvalues of $\boldsymbol{A^\mathsf{T}A}$ are the same as those of $\boldsymbol{AA^\mathsf{T}}$.

Although we derived the formulas for the singular vectors using the provisos of the matrix $\boldsymbol{A}$ being full-column or full-row rank, they are indeed valid for all matrices. The right approach would have been to complete the orthonormal basis on the domain side (from $\hat{\boldsymbol{V}}$, which is the basis for the row space) to all of $\mathbb{R}^n$ (by adding the basis for the null space as well). Similarly, we should have completed the basis on the output space ($\mathbb{R}^m$) by including the basis vectors of the left null space so that we can drop the hats on both $\boldsymbol{V}$ and $\boldsymbol{U}$ and get the picture shown in Figure 13.7. It is perhaps wise to repeat these wordy statements with more mathematical precision as in the following list, where we will use the symbol $\mathcal{B}$ to denote basis.

- $\hat{\boldsymbol{V}}$ is a basis for the row space of $\boldsymbol{A}$: $\hat{\boldsymbol{V}} = \mathcal{B}\left\{\mathcal{C}(\boldsymbol{A^\mathsf{T}})\right\}$
- $\boldsymbol{V}$ is a basis for the domain, which is the union of the bases for the row and null spaces of $\boldsymbol{A}$:

$$\hat{\boldsymbol{V}} = \mathcal{B}\left\{\mathbb{R}^n\right\} = \mathcal{B}\left\{\mathcal{C}(\boldsymbol{A^\mathsf{T}})\right\} \cup \mathcal{B}\left\{\mathcal{N}(\boldsymbol{A})\right\}$$

- $\hat{\boldsymbol{U}}$ is a basis for the column space of $\boldsymbol{A}$: $\hat{\boldsymbol{V}} = \mathcal{B}\left\{\mathcal{C}(\boldsymbol{A})\right\}$
- $\boldsymbol{U}$ is a basis for the output space (AKA co-domain), which is the union of the bases for the column and the left null spaces of $\boldsymbol{A}$:

$$\hat{\boldsymbol{U}} = \mathcal{B}\left\{\mathbb{R}^m\right\} = \mathcal{B}\left\{\mathcal{C}(\boldsymbol{A})\right\} \cup \mathcal{B}\left\{\mathcal{N}(\boldsymbol{A^\mathsf{T}})\right\}$$

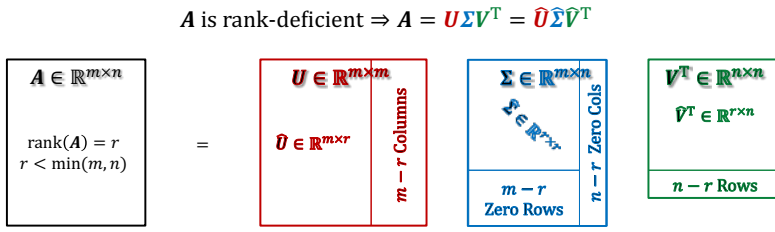


Fig. 13.7 The shapes of $\boldsymbol{U}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{V}$ in SVD when $\boldsymbol{A}$ is a general, rank-deficient matrix.

Now we can confidently state $\boldsymbol{V}^{-1} = \boldsymbol{V}^{\mathsf{T}}$ and $\boldsymbol{U}^{-1} = \boldsymbol{U}^{\mathsf{T}}$ and matrix algebra we performed earlier can be repeated without the hats and everything is perfect.

We have one more way of arriving at this realization, by noting that since these Gram matrices, $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}$ and $\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}}$ have the same rank as $\boldsymbol{A}$, which is r, we have r positive eigenvalues and as many orthogonal eigenvectors. The rest of the eigenvalues ($n-r$ for $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}$ and $m-r$ for $\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}}$) are zeros. Remembering that the eigenvectors corresponding to zero eigenvalues are, in fact, the basis for the null space of the matrix, we can complete the $\boldsymbol{U}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{V}$ (the unhatted ones) matrices. To get the final answer of SVD, we pad $\boldsymbol{U}$ with the basis of the left null space $\mathcal{N}(\boldsymbol{A}^{\mathsf{T}})$ and $\boldsymbol{V}$ with the basis of the null space $\mathcal{N}(\boldsymbol{A})$, which are really the eigenvectors of $\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}}$ and $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}$. They are, indeed, the missing left and right singular vectors we are seeking. And the $\boldsymbol{\Sigma}$ matrix gets the correct zero eigenvalues to pad from the $(r+1)^{\text{th}}$ to the n^{th} diagonal position, resulting in Figure 13.7 and the following

equations.

$$\begin{aligned}
\boldsymbol{U} &= \begin{bmatrix} | & | & \cdots & | \\ \boldsymbol{u_1} & \boldsymbol{u_2} & \cdots & \boldsymbol{u_m} \\ | & | & \cdots & | \end{bmatrix} \in \mathbb{R}^{m\times m} = [\boldsymbol{u_i}] \\
\boldsymbol{V} &= \begin{bmatrix} | & | & \cdots & | \\ \boldsymbol{v_1} & \boldsymbol{v_2} & \cdots & \boldsymbol{v_n} \\ | & | & \cdots & | \end{bmatrix} \in \mathbb{R}^{n\times n} = [\boldsymbol{v_i}] \\
\boldsymbol{\Sigma} &= \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & \vdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \sigma_r & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{m\times n} = [\sigma_i]
\end{aligned} \tag{13.2}$$

The $\boldsymbol{\Sigma}$ matrix is arranged such that $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_r \geq 0$ so that the first singular value is the most important one. The singular vectors are eigenvectors of $\boldsymbol{A^\intercal A}$ and $\boldsymbol{AA^\intercal}$:

$$\boldsymbol{A^\intercal AV} = \boldsymbol{V\Sigma}^2 \quad \text{and} \quad \boldsymbol{AA^\intercal U} = \boldsymbol{U\Sigma}^2 \tag{13.3}$$

We can summarize the singular value decomposition of $\boldsymbol{A} = \boldsymbol{U\Sigma V^\intercal}$ (where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\text{rank}(\boldsymbol{A}) = r$) in a few bullet points:

1. The matrix $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ is the right[1] singular matrix.
   (a) It is an orthonormal basis for $\mathbb{R}^n$, the domain of the transformation $\boldsymbol{A} : \mathbb{R}^n \mapsto \mathbb{R}^m$.
   (b) The columns of $\boldsymbol{V}, \boldsymbol{v_i}$ are the *right* singular vectors.
   (c) $\boldsymbol{v_i}, 0 < i \leq r$ are eigenvectors of $\boldsymbol{A^\intercal A}$.
   (d) The first r of them span the *row* space of $\boldsymbol{A}, \mathcal{C}(\boldsymbol{A^\intercal})$.
   (e) The rest $(n - r)$ span its null space, $\mathcal{N}(\boldsymbol{A})$.
   (f) $\boldsymbol{v_i}, r < i \leq n$ are found (if needed) by computing the null space.
2. $\boldsymbol{U} \in \mathbb{R}^{m\times m}$ is the left singular matrix.

[1] To remember whether $\boldsymbol{U}$ or $\boldsymbol{V}$ is left or right, note that the left singular matrix $\boldsymbol{U}$ appears on the left in $\boldsymbol{U\Sigma V^\intercal}$ and $\boldsymbol{V}$ on the right.

(a) It is an orthonormal basis for $\mathbb{R}^m$, the output space (AKA the co-domain) of the transformation $\boldsymbol{A} : \mathbb{R}^n \mapsto \mathbb{R}^m$.

(b) Its columns, $\boldsymbol{u_i}$, are the *left* singular vectors.

(c) $\boldsymbol{u_i}, 0 < i \leq r$ are the eigenvectors of $\boldsymbol{AA^\mathsf{T}}$.

(d) Again, the first r of them form a basis for the the *column* space $\mathcal{C}(\boldsymbol{A})$.

(e) The rest $(m-r)$ of them span the left null space, $\mathcal{N}(\boldsymbol{A^\mathsf{T}})$.

(f) $\boldsymbol{u_i}, r < i \leq m$ are found (if needed) by computing the left null space.

3. The diagonal matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times n}$ (the same size as $\boldsymbol{A}$) contains the r singular values.

   (a) The first r singular values, $\sigma_i, 0 < i \leq r$ are positive.

   (b) They are the nonnegative square roots of $\boldsymbol{A^\mathsf{T}A}$ (for tall matrices) or $\boldsymbol{AA^\mathsf{T}}$ (for wide ones).

   (c) The rest of the singular values, $\sigma_i, r < i \leq \min(m, n)$, are all zeros.

   (d) By convention, the singular values are arranged in descending order:

$$\sigma_1 \geq \sigma_1 \geq \cdots \geq \sigma_r$$

These bullet points are illustrated in Figure 13.7, the full-blown SVD. Since only the first r singular values are nonzero (and also because the basis vectors $\boldsymbol{v_i}$ of the null space map to $\mathbf{0} \in \mathbb{R}^m$), we can ignore the last $(m - r)$ columns of $\boldsymbol{U}$ and $(n - r)$ columns of $\boldsymbol{V}$. We can also take the leading $r \times r$ part of $\boldsymbol{\Sigma}$ to come up with a "thin" or "economical" version of SVD. Note that this version does not involve any approximations; it is merely a choice of ignoring the zeros that do not matter any way.

We also have the following properties for singular values:

- For square matrices, the product of the singular values is the determinant. For instance, for $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$, $\sigma_1\sigma_2 = \lambda_1\lambda_2 = |\boldsymbol{A}|$.
- For $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$, $\sigma_1 \geq \lambda_1 \geq \lambda_2 \geq \sigma_2$

13.3 Why SVD Is Important

13.3.1 Data Compression

Remember that the Spectral Theorem (§12.1.1, page 226) allowed us to write:

$$\boldsymbol{A} = \sum_{i=1}^{n} \lambda_i \boldsymbol{q_i} \boldsymbol{q_i}^{\mathsf{T}}$$

from the EVD of a real, symmetric $\boldsymbol{A} \in \mathbb{R}^{n \times n} = \boldsymbol{Q\Lambda Q}^{\mathsf{T}}$.

In SVD, we have $\boldsymbol{A} \in \mathbb{R}^{m \times n} = \boldsymbol{U\Sigma V}^{\mathsf{T}}$ (for *any* matrix). This product also can be expanded as:

$$\boldsymbol{A} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u_i} \boldsymbol{v_i}^{\mathsf{T}} \tag{13.4}$$

where we used the fact from Eqn (13.2) that the diagonal matrix $\boldsymbol{\Sigma}$ has only the first $r = \text{rank}(\boldsymbol{A})$ nonzero elements, and therefore, the sum runs from 1 only to r, not to m or n, the matrix dimensions.

It is perhaps important enough to reiterate that Eqn (13.4), by itself, is not an approximation just because we are summing only up to r, which is to say, we are using the "economical" (hatted) SVD matrices. Even if we were to use the full matrices, the multiplication $\boldsymbol{\Sigma V}^{\mathsf{T}}$ would have resulted in a matrix ($\in \mathbb{R}^{m \times n}$) with the last $n - r$ columns zero because only the first r singular values ($\sigma_i, 0 < i \leq r$) are nonzero.

Each term in the summation in Eqn (13.4), $\boldsymbol{A}_i = \boldsymbol{u_i} \boldsymbol{v_i}^{\mathsf{T}}$, is a rank-one matrix because it is a linear combination of one row matrix $\boldsymbol{v_i}^{\mathsf{T}}$, by the row-picture of matrix multiplication. The first of them, $\boldsymbol{A}_1 = \boldsymbol{u_1} \boldsymbol{v_1}^{\mathsf{T}}$, is the most important one because σ_1 is the largest among the singular values. We can therefore see that $\boldsymbol{A}_1$ is the best rank-one approximation of the original matrix $\boldsymbol{A}$.

Let's say $\boldsymbol{A}$ is a megapixel image of size 1000×1000. It takes a million bytes to store it. $\boldsymbol{A}_1$, on the other hand, takes up only 2001 bytes $1(\sigma_1) + 1000(\boldsymbol{u_i}) + 1000(\boldsymbol{u_i})$. If $\boldsymbol{A}_1$ is not good enough, we may include up to k such rank one matrix at a storage cost of $2001k$, which is smaller than a million for $k < 499$. Typically, the first few tens of σ_i would be enough to keep most of the information in $\boldsymbol{A}$.

In general, in order to store up to k rank-one approximations of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, we need $k(m + n + 1)$ units of memory, which could be

significantly smaller than mn. The Singular-Value Decomposition, therefore, can be the basis of a powerful data compression algorithm. Although we have more powerful techniques than the plain old SVD for image compression, it is a technique that still inspires some researchers[2].

13.3.2 Principal Component Analysis

In data science, SVD is commonly used for dimensionality reduction to uncover the low-dimensional patterns in the data. Although it may be called the Principal Component Analysis (PCA), it is essentially SVD using different jargon and with one extra step. We will stick with our current notation to describe PCA.

Let's say $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is a data matrix, such as the Young Adult dataset we used in Figure 10.4 while studying multiple linear regression. Here, we have an observation as a row in the matrix, and variables along which the observations are made in the columns. We have m observations along n variables ($x_j, 1 \leq j \leq n$), and as we saw multiple times, $m \gg n$.

The first step in PCA is to compute one extra row called the mean (as in average) row, and subtract it from all other rows.

$$\mu_j = \frac{1}{m} \sum_{i=0}^{m} a_{ij} = E[x_j]; \quad a_{ij} \leftarrow a_{ij} - \mu_j$$

Thus, we get the so-called zero-centered data in our $\boldsymbol{A}$ matrix (which we will call $\boldsymbol{A}_0$ to avoid confusion), where each column has a mean of 0. Now, the Gram matrix, $\boldsymbol{A}_0^\intercal \boldsymbol{A}_0$ becomes a much smaller matrix in terms of the number of elements because $\boldsymbol{A}_0^\intercal \boldsymbol{A}_0 \in \mathbb{R}^{n \times n}$ and $n \ll m$.

It also becomes proportional to the covariance matrix of the data. To see how this magic happens, we only need to expand the product. Calling $\boldsymbol{A}_0^\intercal \boldsymbol{A}_0$ by a new name, $\boldsymbol{C} \in \mathbb{R}^{n \times n} = [c_{ij}] = \boldsymbol{A}_0^\intercal \boldsymbol{A}_0$, we can write:

$$c_{ij} = \boldsymbol{a}_{\boldsymbol{i}(\mathbf{0})}^\intercal \boldsymbol{a}_{\boldsymbol{j}(\mathbf{0})} = \sum_{k=0}^{m} (a_{ki} - \mu_i)(a_{kj} - \mu_j)$$

[2] A quick search revealed this article from 2017.

which is the covariance between the i^{th} and j^{th} variables. For example, if we set $i = j$ to look at the diagonal elements of $\boldsymbol{C}$, we get

$$\sum_{k=0}^{m}(a_{ki} - \mu_i)(a_{ki} - \mu_i) = m\,(a_{ki} - \mu_i)^2 = m\,\text{Var}(x_i)$$

which is the variance of the i^{th} variable x_i (times the number of observations).

Taking the SVD of $\boldsymbol{A}_0$ is the same as finding the eigenvalues and vectors of $\boldsymbol{C}$, the elements of which are proportional to the covariances of the variables in the data. Therefore, as we saw in §11.5 (page 202), what PCA does is to find directions along which the covariance is the highest, second highest and so on. In other words, it gives us a hierarchical coordinate system, in which the first direction corresponds to the highest variance in the data, the next one the second highest and so on. Note that it comes from the data directly, without assuming any kind of underlying probability distribution for the variables.

Once we have the SVD of $\boldsymbol{A}_0$, we can look at the right singular vectors $\boldsymbol{v_i}$, which are the eigenvectors of $\boldsymbol{A}_0^\intercal\boldsymbol{A}_0$ (as in Eqn (13.3)) and write:

$$\boldsymbol{A}_0^\intercal\boldsymbol{A}_0\boldsymbol{V} = \boldsymbol{V}\boldsymbol{\Sigma}^2 \tag{13.5}$$

We know that the singular values are sorted so that σ_1 is the largest. Using the insights about what eigenanalysis does (from §11.5, page 202), we make the following observations:

1. The eigenvectors of the Gram matrix $\boldsymbol{A}_0^\intercal\boldsymbol{A}_0$ represent the directions in its row space along which the variances in the data are sorted descending. Remembering that $\boldsymbol{A}_0^\intercal\boldsymbol{A}_0$ and $\boldsymbol{A}_0$ have the same row space from §12.6 (page 239), the eigenvectors in the columns of $\boldsymbol{V}$ are the directions in the data space along which we have the sorted variances as well.

2. The matrix $\boldsymbol{T} = \boldsymbol{A}_0\boldsymbol{V}$ holds the *principal components* of the data (the first of which is a linear combination of the columns of $\boldsymbol{A}_0$ with the largest variance, the second the second largest and so on).

3. $\boldsymbol{V}$, whose columns specify the coefficients required in the linear combinations, is called the *loading*.

4. The elements of Σ tell us how much variance each principal component holds. If we want, for instance, to capture 90% of the variance in the data, we start with the first component, and look at σ_1. If it is less than 90%, we keep adding the next component until the cumulative sum of $\sigma_i, 1 \leq i \leq k$ crosses the required threshold of variance, 90%.

5. The k principal components are then used as engineered features in our machine learning algorithm. Typically, $k \ll n$, resulting in significant dimensionality reduction. Although its description looks different, what we are doing here is not unlike taking the first k terms in the summation in Eqn (13.4).

6. Dimensionality reduction has the added advantage of smoothing out the noise in the data. Much like fitting a line or a polynomial on a bunch of (x, y) data points may result in a better fit than the collection of the points themselves, which may be subject to statistical or measurement fluctuations, a lower rank approximation of the data matrix may result in a better model.

We discussed the whole PCA topic as though it was a sub-topic of SVD. For historical reasons, however, PCA may appear with different jargon and notations in other sources, with no reference to SVD. The mathematics behind it is the same as our discourse here. It has to be, for mathematical truth is singular.

A Simulated Example

As a well-established and widely used technique, PCA is available in all modern statistical applications. In order to make its discussion clear, we will use `R` and create a toy example of it using simulation. We are going to simulate a multivariate normal distribution, centered at $\boldsymbol{\mu}$ with a covariance matrix $\boldsymbol{S}$, generating 1000 tuples (x, y).

$$\boldsymbol{\mu} = \begin{bmatrix} 2.5 \\ 2.5 \end{bmatrix} \quad \boldsymbol{S} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

Since we set the parameters for the simulation, we already know what to expect: We started with the covariance matrix $\boldsymbol{S}$, and we expect the bivariate normal distribution to show up as an ellipse, centered at $(1, 3)$ and with the major and minor axis along the eigenvectors $\boldsymbol{s_1}$

and $\boldsymbol{s_2}$ of $\boldsymbol{S}$. The lengths of the axes are going to be the eigenvalues, λ_i, as we established in §11.5 (page 202).

Running the eigenvalue decomposition on $\boldsymbol{S}$ in R, we get:

```
eigen() decomposition
$values
[1] 3 1

$vectors
           [,1]       [,2]
[1,] 0.7071068 -0.7071068
[2,] 0.7071068  0.7071068
```

In this output, `$values` are our eigenvalues λ_i , which says the variances along the major and minor axes of the generated data in Figure 13.8, left panel, are 3 and 1. Therefore the lengths of the axes (σ_i) are the square roots, namely about 1.73 and 1, which is what we see in Figure 13.8, on the left (except that we scaled the standard deviations by a factor of two, so that 95% of the points are within the ellipse).

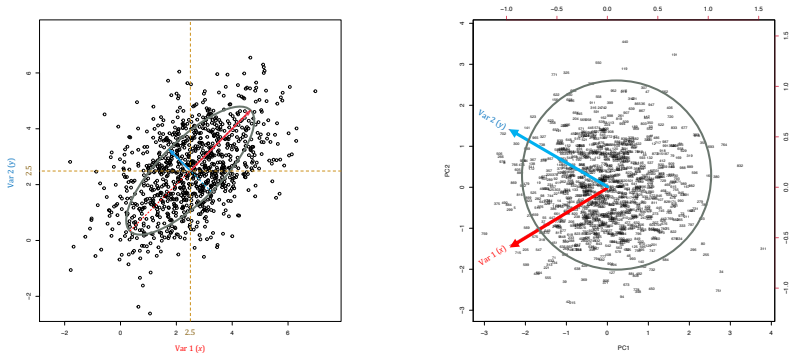


Fig. 13.8 Left: Example of simulated (x, y) pairs, showing the elliptical shape, the directions and sizes of the major and minor axes. Right: The "biplot" from the PCA function in R.

Once we have the 1000 generated rows of simulated data, we run PCA on it, and look at what it can reveal about it.

```
Standard deviations (1, .., p=2):
[1] 1.718043 1.018423
```

```
Rotation (n x k) = (2 x 2):
             PC1        PC2
[1,] -0.7043863 -0.7098168
[2,] -0.7098168  0.7043863
```

From the PCA output, we see the standard deviations σ_1 and σ_2, close to what we specified in our covariance matrix, $\boldsymbol{S}$, as revealed by the eigenanalysis on it. Ideally, the values should have been $\sqrt{3}$ and $\sqrt{1}$, but we got 1.718 and 1.018—close enough. The first principal component is linear combination of x and y with coefficients -0.704 and -0.710. Note that SVD vectors are unique in values only; their signs are not fixed.

The right panel in Figure 13.8 is the so-called "biplot" of the analysis, which shows a scatter plot between the first and second principal components, as well as the loading of the original variables. The first thing to note is that `PC1` and `PC2` are now uncorrelated standard normal distributions, as indicated by the circle in the biplot rather than the ellipse in the data.

The directions shown in red and blue are to be understood as follows: The first variable x "loads" `PC1` with a weight of -0.704 and and `PC2` -0.710 (which form the first right singular vector, $\boldsymbol{v_1}$). It is shown as the red arrow in the biplot, but with some scaling so that its length corresponds to the weight. The second variable x loads `PC1` and `PC2` at -0710 and 0.704 (the second right singular vector, $\boldsymbol{v_2}$), shown in the blue arrow, again with some scaling. Since both the principal components load each variable with similar weights, the lengths of the red and blue arrows are similar.

13.4 Pseudo-Inverse

We talked about the left and right inverses in §5.4 (page 103). Eqn (5.2), for instance, shows how we define the left inverse of a "tall," full-rank matrix. The SVD method gives us another way to define an inverse of any matrix, called the pseudo-inverse, $\boldsymbol{A}^+$.

Let's first define what we are looking for.

PCA on Iris Dataset

All introductory data analytics courses will deal with the Iris dataset at some point. This dataset contains 150 flower measurements along four variables (**Sepal Length**, **Sepal Width**, **Petal Length** and **Petal Width**) from three different iris species (**Setosa**, **Versicolor** and **Virgnica**). There are 50 data points for each species.

Since creating two principal components out of two variables (as we did with the simulated data) does not make much sense, let's try PCA on the Iris dataset. Running PCA on it gives us the following output:

```
Standard deviations (1, .., p=4):
[1] 2.0562689 0.4926162 0.2796596 0.1543862
Rotation (n x k) = (4 x 4):
                    PC1         PC2         PC3         PC4
Sepal.Length  0.36138659 -0.65658877  0.58202985  0.3154872
Sepal.Width  -0.08452251 -0.73016143 -0.59791083 -0.3197231
Petal.Length  0.85667061  0.17337266 -0.07623608 -0.4798390
Petal.Width   0.35828920  0.07548102 -0.54583143  0.7536574
```

Pseudo-Inverse

Definition: A matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ has an associated pseudo-inverse $\boldsymbol{A}^+$ if the following four criteria are met:

1. $\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A} = \boldsymbol{A}$: Note that $\boldsymbol{A}\boldsymbol{A}^+$ does not have to be $\boldsymbol{I}$.
2. $\boldsymbol{A}^+\boldsymbol{A}\boldsymbol{A}^+ = \boldsymbol{A}^+$: The product $\boldsymbol{A}^+\boldsymbol{A}$ does not have to be $\boldsymbol{I}$ either.
3. $(\boldsymbol{A}^+\boldsymbol{A})^\mathsf{T} = \boldsymbol{A}^+\boldsymbol{A}$: Like the Gram matrix, $\boldsymbol{A}\boldsymbol{A}^+$ needs to be symmetric.
4. $(\boldsymbol{A}\boldsymbol{A}^+)^\mathsf{T} = \boldsymbol{A}\boldsymbol{A}^+$: The other product, $\boldsymbol{A}^+\boldsymbol{A}$ should be symmetric too.

With the SVD of $\boldsymbol{A}$, we can come up with $\boldsymbol{A}^+$ that satisfies the four criteria.

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\mathsf{T} \implies \boldsymbol{A}^+ = \boldsymbol{V}\boldsymbol{\Sigma}^+\boldsymbol{U}^\mathsf{T} \tag{13.6}$$

where $\boldsymbol{\Sigma}^+$ is a diagonal matrix with the reciprocals of σ_i when $\sigma_i \neq 0$ and zero when $\sigma_i = 0$. In practice, since floating point zero comparison is always troublesome in computing, we will use a lower bound for σ_i. Note that $\boldsymbol{\Sigma}^+$ has the same size as $\boldsymbol{A}^\mathsf{T}$, while $\boldsymbol{\Sigma}$ has the same size as $\boldsymbol{A}$. In other words, for $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{\Sigma}^+ \in \mathbb{R}^{n \times m}$.

Pesudo-Inverse vs. Other Inverses

The pseudo-inverse reduces to the old familiar double-sided inverse ($\boldsymbol{A^{-1}}$) in the case of full-rank, square matrix. It also reduces to the left inverse ($\boldsymbol{A_{\text{Left}}^{-1}}$) and the right inverse when they are defined, as shown below.

***Pseudo-inverse to Double-Sided Inverse*:** If $\boldsymbol{A}$ is a full-rank, square matrix ($\boldsymbol{A} \in \mathbb{R}^{n\times n}$, rank$(\boldsymbol{A}) = n$), the $\boldsymbol{A}^+$ reduces to $\boldsymbol{A^{-1}}$. Since rank$(\boldsymbol{A}) = n$, rank$\big(\boldsymbol{A^\intercal A}\big) = n \implies \boldsymbol{A^\intercal A}$ has n (positive) eigenvalues $\implies \boldsymbol{\Sigma}$ is full rank. We also have, in $\boldsymbol{A = U\Sigma V^\intercal}, \boldsymbol{U^{-1} = U^\intercal}$, $\boldsymbol{UU^\intercal = I}_n, \boldsymbol{V^{-1} = V^\intercal}$, $\boldsymbol{VV^\intercal = I}_n$ and

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n \end{bmatrix}, \boldsymbol{\Sigma}^+ = \begin{bmatrix} \frac{1}{\sigma_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_n} \end{bmatrix} = \boldsymbol{\Sigma}^{-1}.$$

Now, $\boldsymbol{A^{-1}} = (\boldsymbol{V^\intercal})^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{-1} = (\boldsymbol{V}^{-1})^{-1}\boldsymbol{\Sigma}^+\boldsymbol{U^\intercal} = \boldsymbol{V\Sigma}^+\boldsymbol{U^\intercal} = \boldsymbol{A}^+$.

***Pseudo-inverse to Left Inverse*:** If $\boldsymbol{A}$ is a full-column-rank tall matrix ($\boldsymbol{A} \in \mathbb{R}^{m\times n}$, rank$(\boldsymbol{A}) = n < m$), $\boldsymbol{A}^+$ reduces to: $\boldsymbol{A_{\text{Left}}^{-1}} = (\boldsymbol{A^\intercal A})^{-1}\boldsymbol{A^\intercal}$. Here, we have $\boldsymbol{U}^{-1} = \boldsymbol{U^\intercal}, \quad \boldsymbol{UU^\intercal = I}_m, \boldsymbol{V}^{-1} = \boldsymbol{V^\intercal}, \quad \boldsymbol{VV^\intercal = I}_n$ and

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n \\ 0 & \cdots & 0 \\ \vdots & \cdots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}, \boldsymbol{\Sigma}^+ = \begin{bmatrix} \frac{1}{\sigma_1} & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \cdots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_n} & 0 & \cdots & 0 \end{bmatrix} \implies \boldsymbol{\Sigma^+\Sigma = I}_n.$$

Consider $\boldsymbol{A^+A} = (\boldsymbol{V\Sigma^+U^\intercal})(\boldsymbol{U\Sigma V^\intercal}) = \boldsymbol{V\Sigma^+(U^\intercal U)\Sigma V^\intercal} = \boldsymbol{V(\Sigma^+\Sigma)V^\intercal} = \boldsymbol{VI}_n\boldsymbol{V^\intercal} = \boldsymbol{I}_n$. Thus $\boldsymbol{A}^+$ is the left inverse of $\boldsymbol{A} \implies \boldsymbol{A}^+ = \boldsymbol{A_{\text{Left}}^{-1}} = (\boldsymbol{A^\intercal A})^{-1}\boldsymbol{A^\intercal}$.

***Pseudo-inverse to Right Inverse*:** If $\boldsymbol{A}$ is a full-row-rank wide matrix ($\boldsymbol{A} \in \mathbb{R}^{m\times n}$, rank$(\boldsymbol{A}) = m < n$), $\boldsymbol{A}^+$ reduces to: $\boldsymbol{A_{\text{Right}}^{-1}} = \boldsymbol{A^\intercal}(\boldsymbol{AA^\intercal})^{-1}$. Here again, we have $\boldsymbol{U}^{-1} = \boldsymbol{U^\intercal}, \quad \boldsymbol{UU^\intercal = I}_m, \boldsymbol{V}^{-1} = \boldsymbol{V^\intercal}, \quad \boldsymbol{VV^\intercal = I}_n$ and

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \cdots & \vdots \\ 0 & \cdots & \sigma_m & 0 & \cdots & 0 \end{bmatrix}, \quad \boldsymbol{\Sigma}^+ = \begin{bmatrix} \frac{1}{\sigma_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_m} \\ 0 & \cdots & 0 \\ \vdots & \cdots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \implies \boldsymbol{\Sigma\Sigma^+ = I}_m.$$

In this case, consider $\boldsymbol{AA^+} = (\boldsymbol{U\Sigma V^\intercal})(\boldsymbol{V\Sigma^+U^\intercal}) = \boldsymbol{U\Sigma}(\boldsymbol{V^\intercal V})\boldsymbol{\Sigma^+U^\intercal} = \boldsymbol{U}(\boldsymbol{\Sigma\Sigma^+})\boldsymbol{U^\intercal} = \boldsymbol{UI}_m\boldsymbol{U^\intercal} = \boldsymbol{I}_m$. Thus $\boldsymbol{A}^+$ is the right inverse of $\boldsymbol{A} \implies \boldsymbol{A}^+ = \boldsymbol{A_{\text{Right}}^{-1}} = \boldsymbol{A^\intercal}(\boldsymbol{AA^\intercal})^{-1}$.

At this stage of our Linear Algebra learning, it should be trivial for us to verify that this definition of $\boldsymbol{A}^+$ does satisfy the four criteria.

We only need to note that $\boldsymbol{U}$ and $\boldsymbol{V}$ are square, full rank, orthonormal matrices, with the associated property $\boldsymbol{U}\boldsymbol{U}^\mathsf{T} = \boldsymbol{U}^\mathsf{T}\boldsymbol{U} = \boldsymbol{I}_m$ and $\boldsymbol{V}\boldsymbol{V}^\mathsf{T} = \boldsymbol{V}^\mathsf{T}\boldsymbol{V} = \boldsymbol{I}_n$. Remember that $\boldsymbol{U}$ and $\boldsymbol{V}$ are, in fact, the bases for $\mathbb{R}^m$ and $\mathbb{R}^n$ respectively.

It is possible to rewrite the pseudo-inverse using the "thin" SVD, where we have the hatted matrices. The shapes of these matrices are specified in Eqn (13.2) and shown again pictorially in Figure 13.6, which shows the typical case in data science, where we have a full-column-rank "tall" matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}, m > n, \text{rank}(\boldsymbol{A}) = n$: All the columns in $\boldsymbol{A}$ are linearly independent. $\boldsymbol{U}$ contains the left singular vectors, which form a complete basis for the output space $\mathbb{R}^m$. The first n of these vectors are a basis for the column space $\mathcal{C}(\boldsymbol{A})$. The rest $m - n$ vectors are the basis for the left null space $\mathcal{N}(\boldsymbol{A}^\mathsf{T})$. We have n singular values, in decreasing order, in the leading diagonal of $\boldsymbol{\Sigma}$. Since $\boldsymbol{\Sigma}$ has m rows (same shape as $\boldsymbol{A}$), we have $m-n$ zero rows. $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ holds a full basis for $\mathbb{R}^n$. There are no other dimensions left anywhere to account for.

Figure 13.7 shows the most general case of a rank-deficient matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}, \text{rank}(\boldsymbol{A}) = r < \min(m, n)$. The left and right singular matrices ($\boldsymbol{U}$ and $\boldsymbol{V}$) are both square matrices forming the full basis for $\mathbb{R}^{m\times m}$ and $\mathbb{R}^{n\times n}$ respectively. The first r columns of both contain the bases for the column space and the row space of $\boldsymbol{A}$, namely $\mathcal{C}(\boldsymbol{A}^\mathsf{T})$ and $\mathcal{C}(\boldsymbol{A})$. The rest of the columns are the bases for the null spaces: The last $m - r$ columns in $\boldsymbol{U}$ span the left null space $\mathcal{N}(\boldsymbol{A}^\mathsf{T})$ and the last $n - r$ columns of $\boldsymbol{V}$ span the (right) null space $\mathcal{N}(\boldsymbol{A})$. And, as for the singular values in $\boldsymbol{\Sigma}$, its first r of them are nonzero, the rest are all zeros.

As we can see in Figure 13.7, the columns (and rows) to the right of (and below) the thin lines in $\boldsymbol{U}, \boldsymbol{\Sigma}$ and $\boldsymbol{V}^\mathsf{T}$ are there to account for the null spaces. They do not affect the product at all because the zeros in $\boldsymbol{\Sigma}$ will kill them anyway. We can, therefore, write an economical SVD for $\boldsymbol{A}$ keeping only the columns (and rows) to the left of (and above) the thin lines in $\boldsymbol{U}, \boldsymbol{\Sigma}$ and $\boldsymbol{V}^\mathsf{T}$. We call these smaller matrices $\hat{\boldsymbol{U}}, \hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{V}}^\mathsf{T}$. We can then write:

$$\boldsymbol{A} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^\mathsf{T} \quad \boldsymbol{U} \in \mathbb{R}^{m\times r}, \boldsymbol{\Sigma} \in \mathbb{R}^{r\times r}, \boldsymbol{V} \in \mathbb{R}^{n\times r}$$

Since we defined $\boldsymbol{\Sigma}^+$ appearing in Eqn (13.6) as having nonzero values only in the leading $r \times r$ submatrix, we can see that only the

first r columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ have any bearing on $\boldsymbol{A}^+$. Therefore, we can write it as:

$$\boldsymbol{A}^+ = \hat{\boldsymbol{V}}\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{U}}^{\mathsf{T}} \tag{13.7}$$

Note that $\hat{\boldsymbol{\Sigma}}^+$ is indeed $\hat{\boldsymbol{\Sigma}}^{-1}$ because $\hat{\boldsymbol{\Sigma}}$ is a full-rank, diagonal matrix and its inverse is another matrix with the reciprocals of the diagonal elements along its diagonal:

$$\hat{\boldsymbol{\Sigma}}^{-1} = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma_2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sigma_r} \end{bmatrix} \in \mathbb{R}^{r\times r} = \begin{bmatrix} \frac{1}{\sigma_i} \end{bmatrix}$$

Since we are working with "economical" version $\hat{\boldsymbol{\Sigma}}$, there are no divide-by-zero errors to worry about when computing its inverse: all the singular values σ_i are nonzero.

Note that in Eqn (13.6), the matrices of singular vectors on the right are all square and invertible ones. We are justified in taking their inverses (with $\boldsymbol{U}^{-1} = \boldsymbol{U}^{\mathsf{T}}$ and $\boldsymbol{V}^{-1} = \boldsymbol{V}^{\mathsf{T}}$). But their product is not a square matrix, and we are probably not justified in calling it an inverse, which is one of the reasons why we call it a pseudo-inverse. $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{A}^+ \in \mathbb{R}^{n\times m}$ and both $\boldsymbol{A}\boldsymbol{A}^+$ and $\boldsymbol{A}^+\boldsymbol{A}$ are valid matrix multiplications.

The pseudo-inverse reduces to the left-inverse for full-column-rank matrices, and the right-inverse for the full-row-rank matrices. And of course, for a full-rank, square matrix, it is the plain old inverse. It is indeed worth our time to verify these statements, which is done in a box in this chapter.

The first criterion in the definition of $\boldsymbol{A}^+$ says $\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A} = \boldsymbol{A}$, which means $\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x}$ or $\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{b} = \boldsymbol{b}$ in our favorite equation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. Let's consolidate:

$$\begin{aligned} \boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A}\boldsymbol{x} &= \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ \boldsymbol{A}\boldsymbol{A}^+(\boldsymbol{A}\boldsymbol{x}) &= \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ \boldsymbol{A}\boldsymbol{A}^+\boldsymbol{b} &= \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \end{aligned}$$

What is it telling us? Let's say $\boldsymbol{A}^+\boldsymbol{b} = \boldsymbol{y}$. Then we have the equation $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}$. Earlier, in §R.4, page 224, we argued that this, in conjunction with $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, implied that $\boldsymbol{x} = \boldsymbol{y}$ because of the uniqueness of linear combinations. Here's another argument: Since

we have Eqn (13.7), we can see that $\boldsymbol{A}^{+}\boldsymbol{b}$ is in the row space of $\boldsymbol{A}$ because $\boldsymbol{A}^{+}\boldsymbol{b} = \hat{\boldsymbol{V}}\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{U}}^{\mathsf{T}}\boldsymbol{b}$ is a linear combinations of the columns of $\hat{\boldsymbol{V}}$. Since $\hat{\boldsymbol{V}}$ is a basis for $\mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$, $\boldsymbol{A}^{+}\boldsymbol{b} \in \mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$. In other words, for a nonzero $\boldsymbol{b}$, $\boldsymbol{y}$ is in $\mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$, which is to say, for a $\boldsymbol{b} \in \mathcal{C}(\boldsymbol{A})$, we have $\boldsymbol{A}^{+}\boldsymbol{b} = \boldsymbol{y}$ with $\boldsymbol{y} \in \mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$. Thus, we have the inverse mapping $\boldsymbol{A}^{+} : \mathcal{C}(\boldsymbol{A}) \mapsto \mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$, leading to our closing discussion.

13.5 Fundamental Spaces and SVD

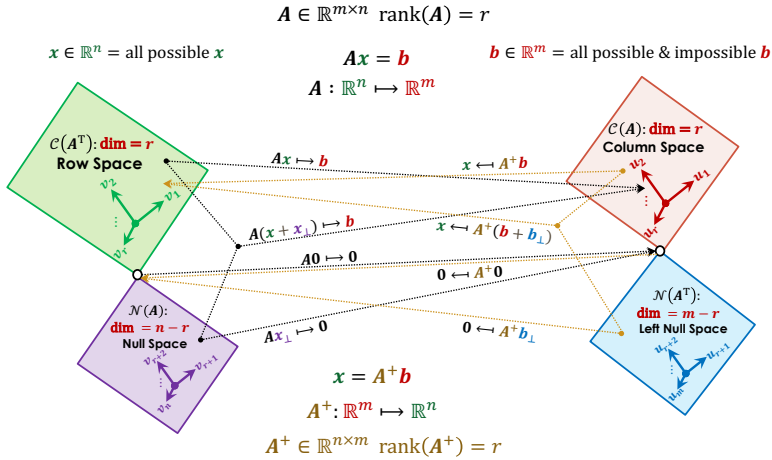


Fig. 13.9 The elegant symmetry of the four fundamental spaces, completed by the pseudo-inverse $\boldsymbol{A}^{+}$.

Singular Value Decomposition and the subsequent definition of the pseudo-inverse complete the beautiful symmetry of the fundamental spaces we started in Chapter 9. $\boldsymbol{A}$ is a mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$: It takes vectors $\boldsymbol{x} \in \mathbb{R}^n$ and gives us vectors $\boldsymbol{b} \in \mathbb{R}^m$. Or, $\boldsymbol{A} : \mathbb{R}^n \mapsto \mathbb{R}^m$. From the row space to the column space of $\boldsymbol{A}$ (from $\mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$ to $\mathcal{C}(\boldsymbol{A})$), the mapping is one-to-one.

$\boldsymbol{A}^{+}$ does just the reverse: $\boldsymbol{A}^{+} : \mathbb{R}^m \mapsto \mathbb{R}^n$. As Figure 13.9 shows, it does it in a completely symmetric fashion. It is a one-to-one mapping from $\mathcal{C}(\boldsymbol{A})$ to $\mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$, and it takes the null spaces to their right counterparts. Instead of describing the picture using our words, it is perhaps wiser to let it speak its own thousand words. Let's stare at it to appreciate its exquisite completeness.

If we can allow ourselves to be mesmerized by the beauty and symmetry of this picture and the elegance of the path that brought us here, perhaps we have earned the right to call ourselves mathematicians or computer scientists. And this picture, perhaps, is the appropriate point to bring this book to its conclusion.

Summing Up. . .

What we want is to see the child in pursuit of knowledge,
and not knowledge in pursuit of the child.
—George Bernard Shaw

We have now come to the end of our journey in the wonderful world of Linear Algebra. Although we have covered a lot of material, we have only scratched the surface of this vast and deep branch of mathematics, for Linear Algebra is a very well-established field of endeavor. Depending on the domain where it is applied, it may appear different. Studied as a pure math course in a graduate level course, Linear Algebra may focus on algebraic structures and generalized fields. When applied to quantum mechanics, it may look like a different beast with alternative notations, working in infinite dimensional spaces and vectors that are functions, where the focus is on the physical interpretation of eigenvalues and vectors.

Even in our own field of computer science, when we go through the literature, we may come across different notations and focus, depending on the background of the authors, and the specific applications

under their investigation. Despite this vastness and diversity, our hope is that we have covered enough ground to dissect the intricacies of any of these applications and styles of discourse. To be sure, the discourses may sound unfamiliar, and the applications may look difficult, even daunting, but their difficulty is one of notations and jargon, not of the underlying mathematics and foundational concepts.

More importantly, we hope that our interest in this particularly beautiful field of mathematics has been kindled, so that we look forward to exploring more, learning further and applying the insights in enriching our professional life as computer scientists.

Glossary

The symbols, terms and abbreviations most commonly used in this book are listed and described below for easy reference.

We can also refer to an Online Glossary of Linear Algebra terms and definitions, courtesy of Robert Campbell of UMBC.

Scalar Scalars are written in lowercase letters, *e.g.*, s, similar to the notation for the elements of vectors and matrices.

Vectors Bold lowercase letters represent vectors, *e.g.*, $\boldsymbol{x}$, with elements x_i. Vectors are always column matrices, and are written with square brackets when needed. Note, however, that **SageMath** writes vectors as $(x_1, x_2, \cdots, x_m)$ (using parentheses with commas between the elements), and we may use that notation as well, albeit rarely.

Matrices Bold capital letters represent matrices, *e.g.*, $\boldsymbol{A}$. For the elements of $\boldsymbol{A}$, the corresponding lowercase letter, a_{ij} is used. When explicitly writing out the elements of the matrix, square brackets are used.

Elements of Matrices: The elements of matrices, being scalars, are written using lowercase letters. When we write $\boldsymbol{A} = \left[a_{ij}\right]$, we mean that $\boldsymbol{A}$ is a matrix with a general element a_{ij} in the i^{th} row and j^{th} column. We also write $\boldsymbol{A} = [\boldsymbol{a_j}]$ to indicate that $\boldsymbol{A}$ is composed of column vectors, with $\boldsymbol{a_i}$ as the j^{th} column. Although used only a couple of times in the book, we also write

$\boldsymbol{A} = [\boldsymbol{a}_i^{\mathsf{T}}]$ to denote the matrix $\boldsymbol{A}$ consisting of row vectors, $\boldsymbol{a}_i^{\mathsf{T}}$ as the i^{th} row.

Determinant: Our favorite symbol for determinant is a vertical line: The determinant of $\boldsymbol{A}$ is $|\boldsymbol{A}|$.

Transpose: The symbol T represents transpose. $\boldsymbol{A} \in \mathbb{R}^{m\times n} \implies \boldsymbol{A}^{\mathsf{T}} \in \mathbb{R}^{n\times m}$.

Hermitian Transpose: The complex conjugate transpose of $\boldsymbol{A}$ is indicated as $\boldsymbol{A}^{\dagger}$, although it is used very sparingly in this book

Fields: Our matrices and vectors are almost always over the field of reals, which is represented as $\mathbb{R}$. We will write $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{x} \in \mathbb{R}^m$. Although we may not use it, it is possible to have matrices and vectors over other fields and rings, such as integers ($\mathbb{Z}$), rationals ($\mathbb{Q}$) or complex ($\mathbb{C}$).

Spaces: We use calligraphic symbols such as $\mathcal{S}$ for spaces. In particular, the column space of $\boldsymbol{A}$ is $\mathcal{C}(\boldsymbol{A})$, row space $\mathcal{C}(\boldsymbol{A}^{\mathsf{T}})$, null space $\mathcal{N}(\boldsymbol{A})$ and its left null space is $\mathcal{N}(\boldsymbol{A}^{\mathsf{T}})$.

Math Symbols: In definitions and equations, we will use common mathematical symbols such as:

- $\forall$: For all or for any. $\forall\, \boldsymbol{x}$ means for any vector $\boldsymbol{x}$.
- $\in$: Is a member of the set. $s \in \mathbb{R}$ says that the scalar s is a member of the set of reals.
- $\implies$: Implies. $i, j \in \mathbb{Z} \implies \frac{i}{j} \in \mathbb{Q}$
- $\subset, \subseteq$: Subset of. $\mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R} \subseteq \mathbb{C}$.

nonstandard Notation: We use the symbol $\oplus$ in a manner not seen elsewhere: We write $\boldsymbol{A} = [\boldsymbol{I} \oplus \boldsymbol{F}]$, for instance, to indicate $\boldsymbol{A}$ is a matrix composed of the columns of the identity matrix and the matrix $\boldsymbol{F}$, but the columns are not necessarily in the order in which they appear in the constituent matrices. They may be "shuffled."

Norm: Double vertical lines indicate the norm (usually the Euclidean norm, usually of a vector). $\|\boldsymbol{x}\|$ is the norm of the vector $\boldsymbol{x}$.

Credits

SageMath and Labs

In order to illustrate the topics in a hands-on manner, we will be using some exercises to be run on **SageMath**. At the time of writing this book, **SageMath** is on version 9.2, and can be downloaded from its website. However, we will be using version 9.1 because of some problems of 9.2 on some platforms. However, **SageMath** is a stable application, and the actual version used is not expected to make much difference in the learning experience. We gratefully acknowledge the efforts of the **SageMath** team in making their excellent tool available freely.

Resources

This book is written explicitly as a textbook to support a corresponding course at Singapore Management University. For each weekly session of our course, which corresponds to a chapter in the book, we will have a curated list of videos either as preparatory, or for review and problem solving.

Linear Algebra is a well-established branch of mathematics, and we have an abundance of online resources to draw from. Here are some more resources:

MIT Open Course Ware 18.06SC: Available on YouTube for free, these excellent lectures by Gilbert Strang and associated recitations are a great resource for our course. Many of the suggested readings (especially the problem solving kind) in this book are drawn from this resource.

3Blue1Brown: Another excellent YouTube channel on mathematics, provided for free by Grant Sanderson, is a must watch for students of applied mathematics and computer science. We have listed several videos from its the playlist Essence of Linear Algebra in our chapters.

Gilbert Strang *"Linear Algebra and its Applications"*: This is the book version of the lectures in 18.06SC, and may be useful for sample problems and as lecture notes. However, lecture notes, problem sets and lecture transcripts from the book are all available online at MIT Open Courseware.

Philip Klein *"Coding the Matrix: Linear Algebra through Computer Science Applications"*: A very comprehensive and well-known work, this book teaches Linear Algebra from a computer science perspective. Some of the labs in our course are inspired by or based on the topics in this book, which has an associated website with a lot of information.

Books

In addition to this textbook, *"Linear Algebra for Computer Science"*, here are some other books that we can freely download and learn from:

Jim Hefferson *"Linear Algebra"*: This book uses `SageMath` and has tutorials and labs that can be downloaded for more practice from the author's website.

Robert Beezer *"A First Course in Linear Algebra"*: Another downloadable book with associated web resources on `SageMath`. In particular, it has an on-line tutorial that can be used as a reference for `SageMath`.

Stephen Boyd *"Introduction to Applied Linear Algebra"*: This well-known book takes a pragmatic approach to teaching Linear Algebra. Commonly referred to by the acronym (*VMLS*) of its subtitle (*"Vectors, Matrices, and Least Squares"*), this book is also freely downloadable and recommended for computer science students.

Copyright

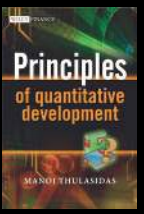
ABOUT LA4CS

An enjoyable and readable textbook on mathematics, **LA4CS** introduces the essential concepts and practice of Linear Algebra to the undergraduate student of computer science.

The focus of this book is on the elegance and beauty of the numerical techniques and algorithms originating from Linear Algebra. As a practical handbook for computer and data scientists, **LA4CS** restricts itself mostly to real fields and tractable discourses, rather than deep and theoretical mathematics.

Its companion website, **LA4CS.com**, features extra information, downloadable material, and links to the video lectures based on this textbook.

ABOUT THE AUTH0R



Manoj Thulasidas, is an **Associate Professor of Computer Science** (Education) who teaches Data Analytics and Linear Algebra to undergraduate students of computer science and information systems at **Singapore Management University**.

His other works include ***The Unreal Universe***, an inquiry into the philosophical underpinnings of physics (from his career as a **physicist** at **CERN** in Geneva) and ***Principles of Quantitative Development***, a practitioner's guide to the lucrative profession of **quantitative finance** (from his experiences as a **quant** in Singapore).

Buy it

Scan or Tap

ASIAN BOOKS
ISBN 978-981-18-2045-8