Linear Algebra for Computer Vision, Robotics, and Machine Learning

Jean Gallier and Jocelyn Quaintance
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA
e-mail: jean@cis.upenn.edu

March 20, 2024

Preface

In recent years, computer vision, robotics, machine learning, and data science have been some of the key areas that have contributed to major advances in technology. Anyone who looks at papers or books in the above areas will be baffled by a strange jargon involving exotic terms such as kernel PCA, ridge regression, lasso regression, support vector machines (SVM), Lagrange multipliers, KKT conditions, *etc.* Do support vector machines chase cattle to catch them with some kind of super lasso? No! But one will quickly discover that behind the jargon which always comes with a new field (perhaps to keep the outsiders out of the club), lies a lot of "classical" linear algebra and techniques from optimization theory. And there comes the main challenge: in order to understand and use tools from machine learning, computer vision, and so on, one needs to have a firm background in linear algebra and optimization theory. To be honest, some probablity theory and statistics should also be included, but we already have enough to contend with.

Many books on machine learning struggle with the above problem. How can one understand what are the dual variables of a ridge regression problem if one doesn't know about the Lagrangian duality framework? Similarly, how is it possible to discuss the dual formulation of SVM without a firm understanding of the Lagrangian framework?

The easy way out is to sweep these difficulties under the rug. If one is just a consumer of the techniques we mentioned above, the cookbook recipe approach is probably adequate. But this approach doesn't work for someone who really wants to do serious research and make significant contributions. To do so, we believe that one must have a solid background in linear algebra and optimization theory.

This is a problem because it means investing a great deal of time and energy studying these fields, but we believe that perseverance will be amply rewarded.

Our main goal is to present fundamentals of linear algebra and optimization theory, keeping in mind applications to machine learning, robotics, and computer vision. This work consists of two volumes, the first one being linear algebra, the second one optimization theory and applications, especially to machine learning.

This first volume covers "classical" linear algebra, up to and including the primary decomposition and the Jordan form. Besides covering the standard topics, we discuss a few topics that are important for applications. These include:

1. Haar bases and the corresponding Haar wavelets.

2. Hadamard matrices.

3. Affine maps (see Section 5.5).

4. Norms and matrix norms (Chapter 8).

5. Convergence of sequences and series in a normed vector space. The matrix exponential e^A and its basic properties (see Section 8.8).

6. The group of unit quaternions, $\mathbf{SU}(2)$, and the representation of rotations in $\mathbf{SO}(3)$ by unit quaternions (Chapter 15).

7. An introduction to algebraic and spectral graph theory.

8. Applications of SVD and pseudo-inverses, in particular, principal component analysis, for short PCA (Chapter 21).

9. Methods for computing eigenvalues and eigenvectors, with a main focus on the QR algorithm (Chapter 17).

Four topics are covered in more detail than usual. These are

1. Duality (Chapter 10).

2. Dual norms (Section 13.7).

3. The geometry of the orthogonal groups $\mathbf{O}(n)$ and $\mathbf{SO}(n)$, and of the unitary groups $\mathbf{U}(n)$ and $\mathbf{SU}(n)$.

4. The spectral theorems (Chapter 16).

Except for a few exceptions we provide complete proofs. We did so to make this book self-contained, but also because we believe that no deep knowledge of this material can be acquired without working out some proofs. However, our advice is to skip some of the proofs upon first reading, especially if they are long and intricate.

The chapters or sections marked with the symbol $\circledast$ contain material that is typically more specialized or more advanced, and they can be omitted upon first (or second) reading.

Acknowledgement: We would like to thank Christine Allen-Blanchette, Kostas Daniilidis, Carlos Esteves, Spyridon Leonardos, Stephen Phillips, João Sedoc, Stephen Shatz, Jianbo Shi, Marcelo Siqueira, and C.J. Taylor for reporting typos and for helpful comments. Mary Pugh and William Yu (at the University of Toronto) taught a course using our book and reported a number of typos and errors. We warmly thank them as well as their students, not only for finding errors, but also for very hepful comments and suggestions for simplifying some proofs. Special thanks to Gilbert Strang. We learned much from his books which have been a major source of inspiration. Thanks to Steven Boyd and James Demmel whose books have been an invaluable source of information. The first author also wishes to express his deepest gratitute to Philippe G. Ciarlet who was his teacher and mentor in 1970-1972 while he was a student at ENPC in Paris. Professor Ciarlet was by far his best teacher. He also

knew how to instill in his students the importance of intellectual rigor, honesty, and modesty. He still has his typewritten notes on measure theory and integration, and on numerical linear algebra. The latter became his wonderful book Ciarlet [14], from which we have borrowed heavily.

Contents

Chapter 1

Introduction

As we explained in the preface, this first volume covers "classical" linear algebra, up to and including the primary decomposition and the Jordan form. Besides covering the standard topics, we discuss a few topics that are important for applications. These include:

1. Haar bases and the corresponding Haar wavelets, a fundamental tool in signal processing and computer graphics.

2. Hadamard matrices which have applications in error correcting codes, signal processing, and low rank approximation.

3. Affine maps (see Section 5.5). These are usually ignored or treated in a somewhat obscure fashion. Yet they play an important role in computer vision and robotics. There is a clean and elegant way to define affine maps. One simply has to define *affine combinations*. Linear maps preserve linear combinations, and similarly affine maps preserve affine combinations.

4. Norms and matrix norms (Chapter 8). These are used extensively in optimization theory.

5. Convergence of sequences and series in a normed vector space. Banach spaces (see Section 8.7). The matrix exponential e^A and its basic properties (see Section 8.8). In particular, we prove the Rodrigues formula for rotations in $\mathbf{SO}(3)$ and discuss the surjectivity of the exponential map $\exp\colon \mathfrak{so}(3) \to \mathbf{SO}(3)$, where $\mathfrak{so}(3)$ is the real vector space of 3×3 skew symmetric matrices (see Section 11.7). We also show that $\det(e^A) = e^{\mathrm{tr}(A)}$ (see Section 14.5).

6. The group of unit quaternions, $\mathbf{SU}(2)$, and the representation of rotations in $\mathbf{SO}(3)$ by unit quaternions (Chapter 15). We define a homomorphism $r\colon \mathbf{SU}(2) \to \mathbf{SO}(3)$ and prove that it is surjective and that its kernel is $\{-I, I\}$. We compute the rotation matrix R_q associated with a unit quaternion q, and give an algorithm to construct a quaternion from a rotation matrix. We also show that the exponential map

exp: $\mathfrak{su}(2) \to \mathbf{SU}(2)$ is surjective, where $\mathfrak{su}(2)$ is the real vector space of skew-Hermitian 2×2 matrices with zero trace. We discuss quaternion interpolation and prove the famous *slerp interpolation formula* due to Ken Shoemake.

7. An introduction to algebraic and spectral graph theory. We define the graph Laplacian and prove some of its basic properties (see Chapter 18). In Chapter 19, we explain how the eigenvectors of the graph Laplacian can be used for graph drawing.

8. Applications of SVD and pseudo-inverses, in particular, principal component analysis, for short PCA (Chapter 21).

9. Methods for computing eigenvalues and eigenvectors are discussed in Chapter 17. We first focus on the QR algorithm due to Rutishauser, Francis, and Kublanovskaya. See Sections 17.1 and 17.3. We then discuss how to use an *Arnoldi iteration*, in combination with the QR algorithm, to approximate eigenvalues for a matrix A of large dimension. See Section 17.4. The special case where A is a symmetric (or Hermitian) tridiagonal matrix, involves a *Lanczos iteration*, and is discussed in Section 17.6. In Section 17.7, we present power iterations and inverse (power) iterations.

Five topics are covered in more detail than usual. These are

1. Matrix factorizations such as LU, $PA = LU$, Cholesky, and reduced row echelon form (rref). Deciding the solvablity of a linear system $Ax = b$, and describing the space of solutions when a solution exists. See Chapter 7.

2. Duality (Chapter 10).

3. Dual norms (Section 13.7).

4. The geometry of the orthogonal groups $\mathbf{O}(n)$ and $\mathbf{SO}(n)$, and of the unitary groups $\mathbf{U}(n)$ and $\mathbf{SU}(n)$.

5. The spectral theorems (Chapter 16).

Most texts omit the proof that the $PA = LU$ factorization can be obtained by a simple modification of Gaussian elimination. We give a complete proof of Theorem 7.5 in Section 7.6. We also prove the uniqueness of the rref of a matrix; see Proposition 7.19.

At the most basic level, duality corresponds to transposition. But duality is really the bijection between subspaces of a vector space E (say finite-dimensional) and subspaces of linear forms (subspaces of the dual space E^*) established by two maps: the first map assigns to a subspace V of E the subspace V^0 of linear forms that vanish on V; the second map assigns to a subspace U of linear forms the subspace U^0 consisting of the vectors in E on which all linear forms in U vanish. The above maps define a bijection such that $\dim(V) + \dim(V^0) = \dim(E)$, $\dim(U) + \dim(U^0) = \dim(E)$, $V^{00} = V$, and $U^{00} = U$.

Another important fact is that if E is a finite-dimensional space with an inner product $u, v \mapsto \langle u, v\rangle$ (or a Hermitian inner product if E is a complex vector space), then there is a canonical isomorphism between E and its dual E^*. This means that every linear form $f \in E^*$ is uniquely represented by some vector $u \in E$, in the sense that $f(v) = \langle v, u\rangle$ for all $v \in E$. As a consequence, every linear map f has an adjoint f^* such that $\langle f(u), v\rangle = \langle u, f^*(v)\rangle$ for all $u, v \in E$.

Dual norms show up in convex optimization; see Boyd and Vandenberghe [11].

Because of their importance in robotics and computer vision, we discuss in some detail the groups of isometries $\mathbf{O}(E)$ and $\mathbf{SO}(E)$ of a vector space with an inner product. The isometries in $\mathbf{O}(E)$ are the linear maps such that $f \circ f^* = f^* \circ f = \mathrm{id}$, and the direct isometries in $\mathbf{SO}(E)$, also called rotations, are the isometries in $\mathbf{O}(E)$ whose determinant is equal to $+1$. We also discuss the hermitian counterparts $\mathbf{U}(E)$ and $\mathbf{SU}(E)$.

We prove the spectral theorems not only for real symmetric matrices, but also for real and complex normal matrices.

We stress the importance of linear maps. Matrices are of course invaluable for computing and one needs to develop skills for manipulating them. But matrices are used to represent a linear map over a basis (or two bases), and the same linear map has different matrix representations. In fact, we can view the various normal forms of a matrix (Schur, SVD, Jordan) as a suitably convenient choice of bases.

We have listed most of the `Matlab` functions relevant to numerical linear algebra and have included `Matlab` programs implementing most of the algorithms discussed in this book.

Chapter 2

Vector Spaces, Bases, Linear Maps

2.1 Motivations: Linear Combinations, Linear Independence and Rank

In linear optimization problems, we often encounter systems of linear equations. For example, consider the problem of solving the following system of three linear equations in the three variables $x_1, x_2, x_3 \in \mathbb{R}$:

$$\begin{aligned} x_1 + 2x_2 - x_3 &= 1 \\ 2x_1 + x_2 + x_3 &= 2 \\ x_1 - 2x_2 - 2x_3 &= 3. \end{aligned}$$

One way to approach this problem is introduce the "vectors" u, v, w, and b, given by

$$u = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \qquad v = \begin{pmatrix} 2 \\ 1 \\ -2 \end{pmatrix} \qquad w = \begin{pmatrix} -1 \\ 1 \\ -2 \end{pmatrix} \qquad b = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

and to write our linear system as

$$x_1 u + x_2 v + x_3 w = b.$$

In the above equation, we used implicitly the fact that a vector z can be multiplied by a scalar $\lambda \in \mathbb{R}$, where

$$\lambda z = \lambda \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} \lambda z_1 \\ \lambda z_2 \\ \lambda z_3 \end{pmatrix},$$

and two vectors y and and z can be added, where

$$y + z = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} + \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} y_1 + z_1 \\ y_2 + z_2 \\ y_3 + z_3 \end{pmatrix}.$$

Also, given a vector

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix},$$

we define the *additive inverse* $-x$ of x (pronounced minus x) as

$$-x = \begin{pmatrix} -x_1 \\ -x_2 \\ -x_3 \end{pmatrix}.$$

Observe that $-x = (-1)x$, the scalar multiplication of x by -1.

The set of all vectors with three components is denoted by $\mathbb{R}^{3\times 1}$. The reason for using the notation $\mathbb{R}^{3\times 1}$ rather than the more conventional notation $\mathbb{R}^3$ is that the elements of $\mathbb{R}^{3\times 1}$ are *column vectors*; they consist of three rows and a single column, which explains the superscript 3×1. On the other hand, $\mathbb{R}^3 = \mathbb{R} \times \mathbb{R} \times \mathbb{R}$ consists of all triples of the form (x_1, x_2, x_3), with $x_1, x_2, x_3 \in \mathbb{R}$, and these are *row vectors*. However, there is an obvious bijection between $\mathbb{R}^{3\times 1}$ and $\mathbb{R}^3$ and they are usually identified. For the sake of clarity, in this introduction, we will denote the set of column vectors with n components by $\mathbb{R}^{n\times 1}$.

An expression such as

$$x_1u + x_2v + x_3w$$

where u, v, w are vectors and the x_is are scalars (in $\mathbb{R}$) is called a *linear combination*. Using this notion, the problem of solving our linear system

$$x_1u + x_2v + x_3w = b.$$

is equivalent to *determining whether b can be expressed as a linear combination of u, v, w.*

Now if the vectors u, v, w are *linearly independent*, which means that there is *no* triple $(x_1, x_2, x_3) \neq (0, 0, 0)$ such that

$$x_1u + x_2v + x_3w = 0_3,$$

it can be shown that *every* vector in $\mathbb{R}^{3\times 1}$ can be written as a linear combination of u, v, w. Here, 0_3 is the *zero vector*

$$0_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

It is customary to abuse notation and to write 0 instead of 0_3. This rarely causes a problem because in most cases, whether 0 denotes the scalar zero or the zero vector can be inferred from the context.

In fact, every vector $z \in \mathbb{R}^{3\times 1}$ can be written *in a unique way* as a linear combination

$$z = x_1u + x_2v + x_3w.$$

This is because if
$$z = x_1u + x_2v + x_3w = y_1u + y_2v + y_3w,$$
then by using our (linear!) operations on vectors, we get
$$(y_1 - x_1)u + (y_2 - x_2)v + (y_3 - x_3)w = 0,$$
which implies that
$$y_1 - x_1 = y_2 - x_2 = y_3 - x_3 = 0,$$
by linear independence. Thus,
$$y_1 = x_1, \quad y_2 = x_2, \quad y_3 = x_3,$$
which shows that z has a unique expression as a linear combination, as claimed. Then our equation
$$x_1u + x_2v + x_3w = b$$
has a *unique solution*, and indeed, we can check that
$$\begin{aligned} x_1 &= 1.4 \\ x_2 &= -0.4 \\ x_3 &= -0.4 \end{aligned}$$
is the solution.

But then, *how do we determine that some vectors are linearly independent?*

One answer is to compute a numerical quantity $\det(u, v, w)$, called the *determinant* of (u, v, w), and to check that it is nonzero. In our case, it turns out that
$$\det(u, v, w) = \begin{vmatrix} 1 & 2 & -1 \\ 2 & 1 & 1 \\ 1 & -2 & -2 \end{vmatrix} = 15,$$
which confirms that u, v, w are linearly independent.

Other methods, which are much better for systems with a large number of variables, consist of computing an LU-decomposition or a QR-decomposition, or an SVD of the *matrix* consisting of the three columns u, v, w,
$$A = \begin{pmatrix} u & v & w \end{pmatrix} = \begin{pmatrix} 1 & 2 & -1 \\ 2 & 1 & 1 \\ 1 & -2 & -2 \end{pmatrix}.$$
If we form the vector of unknowns
$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix},$$

then our linear combination $x_1u + x_2v + x_3w$ can be written in matrix form as

$$x_1u + x_2v + x_3w = \begin{pmatrix} 1 & 2 & -1 \\ 2 & 1 & 1 \\ 1 & -2 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix},$$

so our linear system is expressed by

$$\begin{pmatrix} 1 & 2 & -1 \\ 2 & 1 & 1 \\ 1 & -2 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix},$$

or more concisely as

$$Ax = b.$$

Now what if the vectors u, v, w are *linearly dependent*? For example, if we consider the vectors

$$u = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \qquad v = \begin{pmatrix} 2 \\ 1 \\ -1 \end{pmatrix} \qquad w = \begin{pmatrix} -1 \\ 1 \\ 2 \end{pmatrix},$$

we see that

$$u - v = w,$$

a nontrivial *linear dependence*. It can be verified that u and v are still linearly independent. Now for our problem

$$x_1u + x_2v + x_3w = b$$

it must be the case that b can be expressed as linear combination of u and v. However, it turns out that u, v, b are linearly independent (one way to see this is to compute the determinant $\det(u, v, b) = -6$), so b cannot be expressed as a linear combination of u and v and thus, our system has *no* solution.

If we change the vector b to

$$b = \begin{pmatrix} 3 \\ 3 \\ 0 \end{pmatrix},$$

then

$$b = u + v,$$

and so the system

$$x_1u + x_2v + x_3w = b$$

has the solution

$$x_1 = 1, \quad x_2 = 1, \quad x_3 = 0.$$