Charles L. Byrne
Department of Mathematical Sciences
University of Massachusetts Lowell

Applied and Computational Linear Algebra: A First Course (text for 92.564) (March 7, 2014)

To Eileen,
my wife for the last forty-three years.

My thanks to David Einstein, who read most of
an earlier version of this book
and made many helpful suggestions.

Contents

Preface

Those of us old enough to have first studied linear algebra in the 1960's remember a course devoted largely to proofs, devoid of applications and computation, full of seemingly endless discussion of the representation of linear transformations with respect to various bases, and concerned with matters that would not arise again in our mathematical education. With the growth of computer power and the discovery of powerful algorithms came the *digitization* of many problems previously analyzed solely in terms of functions of continuous variables. As it happened, I began my study of linear algebra in the fall of 1965, just as the two most important new algorithms in computational linear algebra appeared in print; the Cooley-Tukey Fast Fourier Transform (FFT) [103], and the Golub-Kahan method for computing the singular-value decomposition [151] would revolutionize applied linear algebra, but I learned of these more than a decade later. My experience was not at all unique; most of the standard linear algebra texts of the period, such as Cullen [107] and Hoffman and Kunze [170], ignored these advances.

Linear algebra, as we shall see, is largely the study of matrices, at least for the finite-dimensional cases. What connects the theory of matrices to applications are algorithms. Often the particular nature of the applications will prompt us to seek algorithms with particular properties; we then turn to the matrix theory to understand the workings of the algorithms. This book is intended as a text for a graduate course that focuses on applications of linear algebra and on the algorithms used to solve the problems that arise in those applications.

When functions of several continuous variables were approximated by finite-dimensional vectors, partial differential operators on these functions could be approximated by matrix multiplication. Images were represented in terms of grids of pixel values, that is, they became matrices, and then were vectorized into columns of numbers. Image processing then became the manipulation of these column vectors by matrix operations. This digitization meant that very large systems of linear equations now had to be dealt with. The need for fast algorithms to solve these large systems of linear equations turned linear algebra into a branch of applied and computational mathematics. Long forgotten topics in linear algebra, such as singular-value decomposition, were resurrected. Newly discovered algorithms, such as the

simplex method and the fast Fourier transform (FFT), revolutionized the field. As algorithms were increasingly applied to real-world data in real-world situations, the stability of these algorithms in the presence of noise became important. New algorithms emerged to answer the special needs of particular applications, and methods developed in other areas, such as likelihood maximization for statistical parameter estimation, found new application in reconstruction of medical and synthetic-aperture-radar (SAR) images.

The traditional topics of linear algebra, the geometry of Euclidean spaces, solving systems of linear equations and finding eigenvectors and eigenvalues, have not lost their importance, but now have a greater variety of roles to play. Orthogonal projections onto hyperplanes and convex sets form the building blocks for algorithms to design protocols for intensity-modulated radiation therapy. The unitary matrices that arise in discrete Fourier transformation are inverted quickly using the FFT, making essentially real-time magnetic-resonance imaging possible. In high-resolution radar and sonar, eigenvalues of certain matrices can tell us how many objects of interest are out there, while their eigenvectors can tell us where they are. Maximum-likelihood estimation of mixing probabilities lead to systems of linear equations to be solved to provide sub-pixel resolution of SAR images.

Part I

Preliminaries

Chapter 1

Introduction

1.1 Chapter Summary

This chapter introduces some of the topics to be considered in this course.

1.2 Overview of this Course

We shall focus here on applications that require the solution of systems of linear equations, often subject to constraints on the variables. These systems are typically large and sparse, that is, the entries of the matrices are predominantly zero. Transmission and emission tomography provide good examples of such applications. Fourier-based methods, such as filtered back-projection and the Fast Fourier Transform (FFT), are the standard tools for these applications, but statistical methods involving likelihood maximization are also employed. Because of the size of these problems and the nature of the constraints, iterative algorithms are essential.

Because the measured data is typically insufficient to specify a single unique solution, optimization methods, such as least-squares, likelihood maximization, and entropy maximization, are often part of the solution process. In the companion text "A First Course in Optimization", we present the fundamentals of optimization theory, and discuss *problems of optimization*, in which optimizing a function of one or several variables is

the primary goal. Here, in contrast, our focus is on *problems of inference*, optimization is not our primary concern, and optimization is introduced to overcome the non-uniqueness of possible solutions.

1.3 Solving Systems of Linear Equations

Many of the problems we shall consider involve solving, as least approximately, systems of linear equations. When an exact solution is sought and the number of equations and the number of unknowns are small, methods such as Gauss elimination can be used. It is common, in applications such as medical imaging, to encounter problems involving hundreds or even thousands of equations and unknowns. It is also common to prefer inexact solutions to exact ones, when the equations involve noisy, measured data. Even when the number of equations and unknowns is large, there may not be enough data to specify a unique solution, and we need to incorporate prior knowledge about the desired answer. Such is the case with medical tomographic imaging, in which the images are artificially discretized approximations of parts of the interior of the body.

1.4 Imposing Constraints

The iterative algorithms we shall investigate begin with an initial guess x^0 of the solution, and then generate a sequence $\{x^k\}$, converging, in the best cases, to our solution. When we use iterative methods to solve optimization problems, subject to constraints, it is necessary that the limit of the sequence $\{x^k\}$ of iterates obey the constraints, but not that each of the x^k do. An iterative algorithm is said to be an *interior-point method* if each vector x^k obeys the constraints. For example, suppose we wish to minimize $f(x)$ over all x in $\mathbb{R}^J$ having non-negative entries; an interior-point iterative method would have x^k non-negative for each k.

1.5 Operators

Most of the iterative algorithms we shall study involve an *operator*, that is, a function $T : \mathbb{R}^J \to \mathbb{R}^J$. The algorithms begin with an initial guess, x^0, and then proceed from x^k to $x^{k+1} = Tx^k$. Ideally, the sequence $\{x^k\}$ converges to the solution to our optimization problem. To minimize the function $f(x)$ using a gradient descent method with fixed step-length α, for example, the operator is

$$Tx = x - \alpha \nabla f(x).$$

In problems with non-negativity constraints our solution x is required to have non-negative entries x_j. In such problems, the *clipping* operator T, with $(Tx)_j = \max\{x_j, 0\}$, plays an important role.

A subset C of $\mathbb{R}^J$ is *convex* if, for any two points in C, the line segment connecting them is also within C. As we shall see, for any x outside C, there is a point c within C that is closest to x; this point c is called the *orthogonal projection* of x onto C, and we write $c = P_C x$. Operators of the type $T = P_C$ play important roles in iterative algorithms. The clipping operator defined previously is of this type, for C the non-negative orthant of $\mathbb{R}^J$, that is, when C is the set

$$\mathbb{R}^J_+ = \{x \in \mathbb{R}^J | x_j \geq 0, \, j = 1, ..., J\}.$$

1.6 Acceleration

For problems involving many variables, it is important to use algorithms that provide an acceptable approximation of the solution in a reasonable amount of time. For medical tomography image reconstruction in a clinical setting, the algorithm must reconstruct a useful image from scanning data in the time it takes for the next patient to be scanned, which is roughly fifteen minutes. Some of the algorithms we shall encounter work fine on small problems, but require far too much time when the problem is large. Figuring out ways to speed up convergence is an important part of iterative optimization. One approach we shall investigate in some detail is the use of *block-iterative* or *partial gradient* methods.

1.7 Required Homework Problems

The following exercises are the required problems for 92.564: 3.4; 3.9; 3.10; 3.11; 3.13; 3.15; 3.21; 3.40; 4.2; 5.5; 5.6; 5.12; 5.13; 6.5; 6.14; 6.18; 6.19; 6.24; 8.1; 8.5; and 8.6.

Chapter 2

An Overview of Applications

2.1 Chapter Summary

The theory of linear algebra, applications of that theory, and the associated computations are the three threads that weave their way through this course. In this chapter we present an overview of the applications we shall study in more detail later.

2.2 Transmission Tomography

Although transmission tomography (TT) is commonly associated with medical diagnosis, it has scientific uses, such as determining the sound-speed profile in the ocean, industrial uses, such as searching for faults in girders, mapping the interior of active volcanos, and security uses, such as the scanning of cargo containers for nuclear material. Previously, when people spoke of a "CAT scan" they usually meant x-ray transmission tomography, although the term is now used by lay people to describe any of the several scanning modalities in medicine, including single-photon emission computed tomography (SPECT), positron emission tomography (PET), ultrasound, and magnetic resonance imaging (MRI).

2.2.1 Brief Description

Computer-assisted tomography (CAT) scans have revolutionized medical practice. One example of CAT is transmission tomography. The goal here is to image the spatial distribution of various matter within the body, by estimating the distribution of radiation attenuation. At least in theory, the data are line integrals of the function of interest.

In transmission tomography, radiation, usually x-ray, is transmitted through the object being scanned. The object of interest need not be a living human being; King Tut has received a CAT-scan and industrial uses of transmission scanning are common. Recent work [237] has shown the practicality of using cosmic rays to scan cargo for hidden nuclear material; tomographic reconstruction of the scattering ability of the contents can reveal the presence of shielding. Because of their ability to penetrate granite, cosmic rays are being used to obtain transmission-tomographic three-dimensional images of the interior of active volcanos, to measure the size of the magma column and help predict the size and occurrence of eruptions.

In the simplest formulation of transmission tomography, the beams are

assumed to travel along straight lines through the object, the initial intensity of the beams is known and the intensity of the beams, as they exit the object, is measured for each line. The goal is to estimate and image the x-ray attenuation function, which correlates closely with the spatial distribution of attenuating material within the object. Unexpected absence of attenuation can indicate a broken bone, for example.

As the x-ray beam travels along its line through the body, it is weakened by the attenuating material it encounters. The reduced intensity of the exiting beam provides a measure of how much attenuation the x-ray encountered as it traveled along the line, but gives no indication of where along that line it encountered the attenuation; in theory, what we have learned is the integral of the attenuation function along the line. It is only by repeating the process with other beams along other lines that we can begin to localize the attenuation and reconstruct an image of this non-negative attenuation function. In some approaches, the lines are all in the same plane and a reconstruction of a single slice through the object is the goal; in other cases, a fully three-dimensional scanning occurs. The word "tomography" itself comes from the Greek "tomos", meaning part or slice; the word "atom" was coined to describe something supposed to be "without parts".

2.2.2 The Theoretical Problem

In theory, we will have the integral of the attenuation function along every line through the object. The *Radon Transform* is the operator that assigns to each attenuation function its integrals over every line. The mathematical problem is then to invert the Radon Transform, that is, to recapture the attenuation function from its line integrals. Is it always possible to determine the attenuation function from its line integrals? Yes. One way to show this is to use the Fourier transform to prove what is called the *Central Slice Theorem*. The reconstruction is then inversion of the Fourier transform; various methods for such inversion rely on frequency-domain filtering and back-projection.

2.2.3 The Practical Problem

Practise, of course, is never quite the same as theory. The problem, as we have described it, is an over-simplification in several respects, the main one being that we never have all the line integrals. Ultimately, we will construct a discrete image, made up of finitely many pixels. Consequently, it is reasonable to assume, from the start, that the attenuation function to be estimated is well approximated by a function that is constant across small squares (or cubes), called pixels (or voxels), and that the goal is to determine these finitely many pixel values.

2.2.4 The Discretized Problem

When the problem is discretized in this way, different mathematics begins to play a role. The line integrals are replaced by finite sums, and the problem can be viewed as one of solving a large number of linear equations, subject to side constraints, such as the non-negativity of the pixel values. The Fourier transform and the Central Slice Theorem are still relevant, but in discrete form, with the fast Fourier transform (FFT) playing a major role in discrete filtered back-projection methods. This approach provides fast reconstruction, but is limited in other ways. Alternatively, we can turn to iterative algorithms for solving large systems of linear equations, subject to constraints. This approach allows for greater inclusion of the physics into the reconstruction, but can be slow; accelerating these iterative reconstruction algorithms is a major concern, as is controlling sensitivity to noise in the data.

2.2.5 Mathematical Tools

As we just saw, Fourier transformation in one and two dimensions, and frequency-domain filtering are important tools that we need to discuss in some detail. In the discretized formulation of the problem, periodic convolution of finite vectors and its implementation using the fast Fourier transform play major roles. Because actual data is always finite, we consider the issue of under-determined problems that allow for more than one answer, and the need to include prior information to obtain reasonable reconstructions. Under-determined problems are often solved using optimization, such as maximizing the entropy or minimizing the norm of the image, subject to the data as constraints. Constraints are often described mathematically using the notion of convex sets. Finding an image satisfying several sets of constraints can often be viewed as finding a vector in the intersection of convex sets, the so-called *convex feasibility problem* (CFP).

2.3 Emission Tomography

Unlike transmission tomography, emission tomography (ET) is used only with living beings, principally humans and small animals. Although this modality was initially used to uncover pathologies, it is now used to study normal functioning, as well. In emission tomography, including positron emission tomography (PET) and single- photon emission tomography (SPECT), the patient inhales, swallows, or is injected with, chemicals to which radioactive material has been chemically attached [265]. The

chemicals are designed to accumulate in that specific region of the body we wish to image. For example, we may be looking for tumors in the abdomen, weakness in the heart wall, or evidence of brain activity in a selected region. In some cases, the chemicals are designed to accumulate more in healthy regions, and less so, or not at all, in unhealthy ones. The opposite may also be the case; tumors may exhibit greater avidity for certain chemicals. The patient is placed on a table surrounded by detectors that count the number of emitted photons. On the basis of where the various counts were obtained, we wish to determine the concentration of radioactivity at various locations throughout the region of interest within the patient.

Although PET and SPECT share some applications, their uses are generally determined by the nature of the chemicals that have been designed for this purpose, as well as by the half-life of the radionuclides employed. Those radioactive isotopes used in PET generally have half-lives on the order of minutes and must be manufactured on site, adding to the expense of PET. The isotopes used in SPECT have half-lives on the order of many hours, or even days, so can be manufactured off-site and can also be used in scanning procedures that extend over some appreciable period of time.

2.3.1 Coincidence-Detection PET

In a typical PET scan to detect tumors, the patient receives an injection of glucose, to which a radioactive isotope of fluorine, ^{18}F, has been chemically attached. The radionuclide emits individual positrons, which travel, on average, between 4 mm and 2.5 cm (depending on their kinetic energy) before encountering an electron. The resulting annihilation releases two gamma-ray photons that then proceed in essentially opposite directions. Detection in the PET case means the recording of two photons at nearly the same time at two different detectors. The locations of these two detectors then provide the end points of the line segment passing, more or less, through the site of the original positron emission. Therefore, each possible pair of detectors determines a *line of response* (LOR). When a LOR is recorded, it is assumed that a positron was emitted somewhere along that line. The PET data consists of a chronological list of LOR that are recorded. Because the two photons detected at either end of the LOR are not detected at exactly the same time, the time difference can be used in *time-of-flight* PET to further localize the site of the emission to a smaller segment of perhaps 8 cm in length.

2.3.2 Single-Photon Emission Tomography

Single-photon computed emission tomography (SPECT) is similar to PET and has the same objective: to image the distribution of a radionuclide, such as technetium ^{99m}Tc, within the body of the patient. In SPECT

the radionuclide employed emits single gamma-ray photons, which then travel through the body of the patient and, in some fraction of the cases, are detected. Detections in SPECT correspond to individual sensor locations outside the body. The data in SPECT are the photon counts at each of the finitely many detector locations. Unlike PET, in SPECT lead collimators are placed in front of the gamma-camera detectors to eliminate photons arriving at oblique angles. While this helps us narrow down the possible sources of detected photons, it also reduces the number of detected photons and thereby decreases the signal-to-noise ratio.

2.3.3 The Line-Integral Model for PET and SPECT

To solve the reconstruction problem we need a model that relates the count data to the radionuclide density function. A somewhat unsophisticated, but computationally attractive, model is taken from transmission tomography: to view the count at a particular detector as the line integral of the radionuclide density function along the line from the detector that is perpendicular to the camera face. The count data then provide many such line integrals and the reconstruction problem becomes the familiar one of estimating a function from noisy measurements of line integrals. Viewing the data as line integrals allows us to use the Fourier transform in reconstruction. The resulting *filtered back-projection* (FBP) algorithm is a commonly used method for medical imaging in clinical settings.

The line-integral model for PET assumes a fixed set of possible LOR, with most LOR recording many emissions. Another approach is *list-mode* PET, in which detections are recording as they occur by listing the two end points of the associated LOR. The number of potential LOR is much higher in list-mode, with most of the possible LOR being recording only once, or not at all [175, 218, 61].

2.3.4 Problems with the Line-Integral Model

It is not really accurate, however, to view the photon counts at the detectors as line integrals. Consequently, applying filtered back-projection to the counts at each detector can lead to distorted reconstructions. There are at least three degradations that need to be corrected before FBP can be successfully applied [183]: attenuation, scatter, and spatially dependent resolution.

In the SPECT case, as in most such inverse problems, there is a trade-off to be made between careful modeling of the physical situation and computational tractability. The FBP method slights the physics in favor of computational simplicity and speed. In recent years, iterative methods, such as the *algebraic reconstruction technique* (ART), its multiplicative variant, MART, the expectation maximization maximum likelihood (MLEM

or EMML) method, and the rescaled block-iterative EMML (RBI-EMML), that incorporate more of the physics have become competitive.

2.3.5 The Stochastic Model: Discrete Poisson Emitters

In iterative reconstruction we begin by *discretizing* the problem; that is, we imagine the region of interest within the patient to consist of finitely many tiny squares, called *pixels* for two-dimensional processing or cubes, called *voxels* for three-dimensional processing. We imagine that each pixel has its own level of concentration of radioactivity and these concentration levels are what we want to determine. Proportional to these concentration levels are the average rates of emission of photons. To achieve our goal we must construct a model that relates the measured counts to these concentration levels at the pixels. The standard way to do this is to adopt the model of *independent Poisson emitters*. Any Poisson-distributed random variable has a mean equal to its variance. The *signal-to-noise ratio* (SNR) is usually taken to be the ratio of the mean to the standard deviation, which, in the Poisson case, is then the square root of the mean. Consequently, the Poisson SNR increases as the mean value increases, which points to the desirability (at least, statistically speaking) of higher dosages to the patient.

2.3.6 Reconstruction as Parameter Estimation

The goal is to reconstruct the distribution of radionuclide intensity by estimating the pixel concentration levels. The pixel concentration levels can be viewed as parameters and the data are instances of random variables, so the problem looks like a fairly standard parameter estimation problem of the sort studied in beginning statistics. One of the basic tools for statistical parameter estimation is likelihood maximization, which is playing an increasingly important role in medical imaging. There are several problems, however.

One problem is that the number of parameters is quite large, as large as the number of data values, in most cases. Standard statistical parameter estimation usually deals with the estimation of a handful of parameters. Another problem is that we do not quite know the relationship between the pixel concentration levels and the count data. The reason for this is that the probability that a photon emitted from a given pixel will be detected at a given detector will vary from one patient to the next, since whether or not a photon makes it from a given pixel to a given detector depends on the geometric relationship between detector and pixel, as well as what is in the patient's body between these two locations. If there are ribs or skull getting in the way, the probability of making it goes down. If there are just lungs, the probability goes up. These probabilities can change during the

scanning process, when the patient moves. Some motion is unavoidable, such as breathing and the beating of the heart. Determining good values of the probabilities in the absence of motion, and correcting for the effects of motion, are important parts of SPECT image reconstruction.

2.3.7 X-Ray Fluorescence Computed Tomography

X-ray fluorescence computed tomography (XFCT) is a form of emission tomography that seeks to reconstruct the spatial distribution of elements of interest within the body [193]. Unlike SPECT and PET, these elements need not be radioactive. Beams of synchrotron radiation are used to stimulate the emission of fluorescence x-rays from the atoms of the elements of interest. These fluorescence x-rays can then be detected and the distribution of the elements estimated and imaged. As with SPECT, attenuation is a problem; making things worse is the lack of information about the distribution of attenuators at the various fluorescence energies.

2.4 Magnetic Resonance Imaging

Protons have *spin*, which, for our purposes here, can be viewed as a charge distribution in the nucleus revolving around an axis. Associated with the resulting current is a *magnetic dipole moment* collinear with the axis of the spin. In elements with an odd number of protons, such as hydrogen, the nucleus itself will have a net magnetic moment. The objective in *magnetic resonance imaging* (MRI) is to determine the density of such elements in a volume of interest within the body. The basic idea is to use strong magnetic fields to force the individual spinning nuclei to emit signals that, while too weak to be detected alone, are detectable in the aggregate. The signals are generated by the precession that results when the axes of the magnetic dipole moments are first aligned and then perturbed.

In much of MRI, it is the distribution of hydrogen in water molecules that is the object of interest, although the imaging of phosphorus to study energy transfer in biological processing is also important. There is ongoing work using tracers containing fluorine, to target specific areas of the body and avoid background resonance. Because the magnetic properties of blood change when the blood is oxygenated, increased activity in parts of the brain can be imaged through *functional* MRI (fMRI). Non-radioactive isotopes of gadolinium are often injected as contrast agents because of their ability to modify certain parameters called the T1 relaxation times.

2.4.1 Alignment

In the absence of an external magnetic field, the axes of these magnetic dipole moments have random orientation, dictated mainly by thermal effects. When an external magnetic field is introduced, it induces a small fraction, about one in 10^5, of the dipole moments to begin to align their axes with that of the external magnetic field. Only because the number of protons per unit of volume is so large do we get a significant number of moments aligned in this way. A strong external magnetic field, about $20,000$ times that of the earth's, is required to produce enough alignment to generate a detectable signal.

2.4.2 Precession

When the axes of the aligned magnetic dipole moments are perturbed, they begin to precess, like a spinning top, around the axis of the external magnetic field, at the *Larmor frequency*, which is proportional to the intensity of the external magnetic field. If the magnetic field intensity varies spatially, then so does the Larmor frequency. Each precessing magnetic dipole moment generates a signal; taken together, they contain information about the density of the element at the various locations within the body. As we shall see, when the external magnetic field is appropriately chosen, a Fourier relationship can be established between the information extracted from the received signal and this density function.

2.4.3 Slice Isolation

When the external magnetic field is the *static field*, then the Larmor frequency is the same everywhere. If, instead, we impose an external magnetic field that varies spatially, then the Larmor frequency is also spatially varying. This external field is now said to include a *gradient field*.

2.4.4 Tipping

When a magnetic dipole moment is given a component out of its axis of alignment, it begins to precess around its axis of alignment, with frequency equal to its Larmor frequency. To create this off-axis component, we apply a *radio-frequency field* (rf field) for a short time. The effect of imposing this rf field is to tip the aligned magnetic dipole moment axes away from the axis of alignment, initiating precession. The dipoles that have been tipped ninety degrees out of their axis of alignment generate the strongest signal.

2.4.5 Imaging

The information we seek about the proton density function is contained within the received signal. By carefully adding gradient fields to the external field, we can make the Larmor frequency spatially varying, so that each frequency component of the received signal contains a piece of the information we seek. The proton density function is then obtained through Fourier transformations. Fourier-transform estimation and extrapolation techniques play a major role in this rapidly expanding field [159].

2.4.6 The Line-Integral Approach

By appropriately selecting the gradient field and the radio-frequency field, it is possible to create a situation in which the received signal comes primarily from dipoles along a given line in a preselected plane. Performing an FFT of the received signal gives us line integrals of the density function along lines in that plane. In this way, we obtain the three-dimensional Radon transform of the desired density function. The Central Slice Theorem for this case tells us that, in theory, we have the Fourier transform of the density function.

2.4.7 Phase Encoding

In the line-integral approach, the line-integral data is used to obtain values of the Fourier transform of the density function along lines through the origin in Fourier space. It would be more convenient for the FFT if we have Fourier-transform values on the points of a rectangular grid. We can obtain this by selecting the gradient fields to achieve *phase encoding*.

2.4.8 A New Application

A recent article [264] in The Boston Globe describes a new application of MRI, as a guide for the administration of ultra-sound to kill tumors and perform bloodless surgery. In MRI-guided focused ultra-sound, the sound waves are focused to heat up the regions to be destroyed and real-time MRI imaging shows the doctor where this region is located and if the sound waves are having the desired effect. The use of this technique in other areas is also being studied: to open up the blood-brain barrier to permit chemo-therapy for brain cancers; to cure hand tremors, chronic pain, and some effects of stroke, epilepsy, and Parkinson's disease; and to remove uterine fibroids.

2.5 Intensity Modulated Radiation Therapy

A fairly recent addition to the list of applications using linear algebra and the geometry of Euclidean space is *intensity modulated radiation therapy* (IMRT). Although it is not actually an imaging problem, intensity modulated radiation therapy is an emerging field that involves some of the same mathematical techniques used to solve the medical imaging problems discussed previously, particularly methods for solving the convex feasibility problem.

2.5.1 Brief Description

In IMRT beamlets of radiation with different intensities are transmitted into the body of the patient. Each voxel within the patient will then absorb a certain dose of radiation from each beamlet. The goal of IMRT is to direct a sufficient dosage to those regions requiring the radiation, those that are designated *planned target volumes* (PTV), while limiting the dosage received by the other regions, the so-called *organs at risk* (OAR).

2.5.2 The Problem and the Constraints

The intensities and dosages are obviously non-negative quantities. In addition, there are *implementation constraints*; the available treatment machine will impose its own requirements, such as a limit on the difference in intensities between adjacent beamlets. In dosage space, there will be a lower bound on the acceptable dosage delivered to those regions designated as the PTV, and an upper bound on the acceptable dosage delivered to those regions designated as the OAR. The problem is to determine the intensities of the various beamlets to achieve these somewhat conflicting goals.

2.5.3 Convex Feasibility and IMRT

The CQ algorithm [62, 63] is an iterative algorithm for solving the split feasibility problem. Because it is particularly simple to implement in many cases, it has become the focus of recent work in IMRT. In [86] Censor *et al.* extend the CQ algorithm to solve what they call the *multiple-set split feasibility problem* (MSSFP) . In the sequel [84] it is shown that the constraints in IMRT can be modeled as inclusion in convex sets and the extended CQ algorithm is used to determine dose intensities for IMRT that satisfy both dose constraints and radiation-source constraints.

One drawback to the use of x-rays in radiation therapy is that they continue through the body after they have encountered their target. A re-

cent technology, proton-beam therapy, directs a beam of protons at the target. Since the protons are heavy, and have mass and charge, their trajectories can be controlled in ways that x-ray trajectories cannot be. The new proton center at Massachusetts General Hospital in Boston is one of the first to have this latest technology. As with most new and expensive medical procedures, there is some debate going on about just how much of an improvement it provides, relative to other methods.

2.6 Array Processing

Passive sonar is used to estimate the number and direction of distant sources of acoustic energy that have generated sound waves propagating through the ocean. An array, or arrangement, of sensors, called *hydrophones*, is deployed to measure the incoming waveforms over time and space. The data collected at the sensors is then processed to provide estimates of the waveform parameters being sought. In active sonar, the party deploying the array is also the source of the acoustic energy, and what is sensed are the returning waveforms that have been reflected off of distant objects. Active sonar can be used to map the ocean floor, for example. Radar is another active array-processing procedure, using reflected radio waves instead of sound to detect distant objects. Radio astronomy uses array processing and the radio waves emitted by distant sources to map the heavens.

To illustrate how array processing operates, consider Figure 2.1. Imagine a source of acoustic energy sufficiently distant from the line of sensors that the incoming wavefront is essentially planar. As the peaks and troughs of the wavefronts pass over the array of sensors, the measurements at the sensors give the elapsed time between a peak at one sensor and a peak at the next sensor, thereby giving an indication of the angle of arrival.

In practice, of course, there are multiple sources of acoustic energy, so each sensor receives a superposition of all the plane-wave fronts from all directions. because the sensors are spread out in space, what each receives is slightly different from what its neighboring sensors receive, and this slight difference can be exploited to separate the spatially distinct components of the signals. What we seek is the function that describes how much energy came from each direction.

When we describe the situation mathematically, using the wave equation, we find that what is received at each sensor is a value of the Fourier transform of the function we want. Because we have only finitely many sensors, we have only finitely many values of this Fourier transform. So, we

have the problem of estimating a function from finitely many values of its Fourier transform.

2.7 A Word about Prior Information

An important point to keep in mind when applying linear-algebraic methods to measured data is that, while the data is usually limited, the information we seek may not be lost. Although processing the data in a reasonable way may suggest otherwise, other processing methods may reveal that the desired information is still available in the data. Figure 2.2 illustrates this point.

The original image on the upper right of Figure 2.2 is a discrete rectangular array of intensity values simulating a slice of a head. The data was obtained by taking the two-dimensional discrete Fourier transform of the original image, and then discarding, that is, setting to zero, all these spatial frequency values, except for those in a smaller rectangular region around the origin. The problem then is under-determined. A minimum two-norm solution would seem to be a reasonable reconstruction method.

The minimum two-norm solution is shown on the lower right. It is calculated simply by performing an inverse discrete Fourier transform on the array of modified discrete Fourier transform values. The original image has relatively large values where the skull is located, but the minimum two-norm reconstruction does not want such high values; the norm involves the sum of squares of intensities, and high values contribute disproportionately to the norm. Consequently, the minimum two-norm reconstruction chooses instead to conform to the measured data by spreading what should be the skull intensities throughout the interior of the skull. The minimum two-norm reconstruction does tell us something about the original; it tells us about the existence of the skull itself, which, of course, is indeed a prominent feature of the original. However, in all likelihood, we would already know about the skull; it would be the interior that we want to know about.

Using our knowledge of the presence of a skull, which we might have obtained from the minimum two-norm reconstruction itself, we construct the prior estimate shown in the upper left. Now we use the same data as before, and calculate a minimum weighted two-norm solution, using as the weight vector the reciprocals of the values of the prior image. This minimum weighted two-norm reconstruction is shown on the lower left; it is clearly almost the same as the original image. The calculation of the minimum weighted two-norm solution can be done iteratively using the ART algorithm, as discussed in [240].

When we weight the skull area with the inverse of the prior image, we allow the reconstruction to place higher values there without having much of an effect on the overall weighted norm. In addition, the reciprocal weighting in the interior makes spreading intensity into that region costly, so the interior remains relatively clear, allowing us to see what is really present there.

When we try to reconstruct an image from limited data, it is easy to assume that the information we seek has been lost, particularly when a reasonable reconstruction method fails to reveal what we want to know. As this example, and many others, show, the information we seek is often still in the data, but needs to be brought out in a more subtle way.

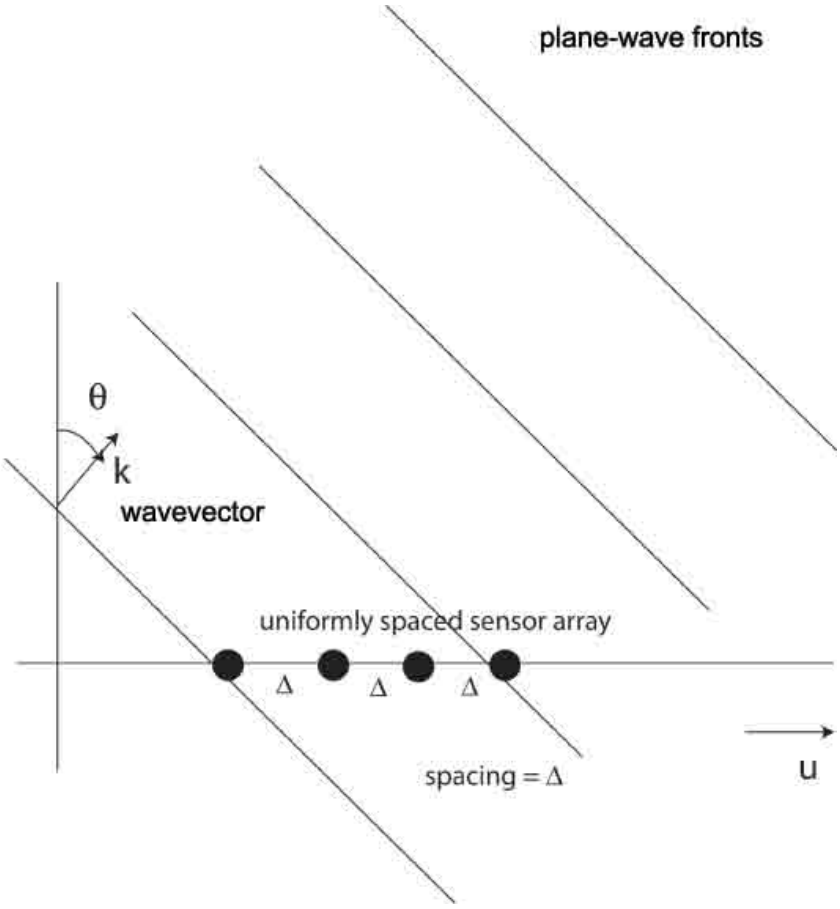

FIGURE 2.1: A uniform line array sensing a plane-wave field.

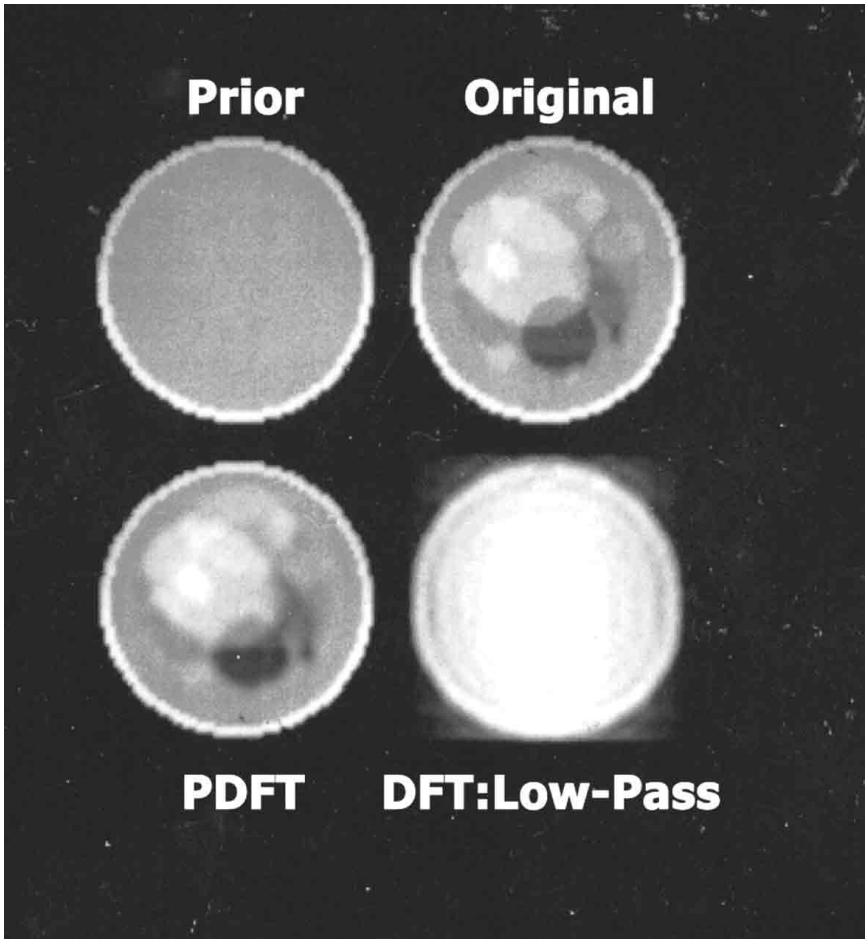

FIGURE 2.2: Extracting information in image reconstruction.

Chapter 3

Matrix Theory

3.1 Chapter Summary

In this chapter we review the fundamentals of matrix algebra.

3.2 Vector Spaces

Linear algebra is the study of *vector spaces* and *linear transformations*. It is not simply the study of matrices, although matrix theory takes up most of linear algebra.

It is common in mathematics to consider abstraction, which is simply a means of talking about more than one thing at the same time. A vector space V is an abstract algebraic structure defined using axioms. There are many examples of vector spaces, such as the sets of real or complex numbers themselves, the set of all polynomials, the set of row or column vectors of a given dimension, the set of all infinite sequences of real or complex numbers, the set of all matrices of a given size, and so on. The beauty of an abstract approach is that we can talk about all of these, and much more, all at once, without being specific about which example we mean.

A vector space is a set whose members are called *vectors*, on which there are two algebraic operations, called *scalar multiplication* and *vector addition*. As in any axiomatic approach, these notions are intentionally abstract. A vector is defined to be a member of a vector space, nothing more. Scalars are a bit more concrete, in that scalars are almost always real or complex numbers, although sometimes, but not in this book, they are members of an unspecified finite field. The operations themselves are not explicitly defined, except to say that they behave according to certain axioms, such as associativity and distributivity.

If v is a member of a vector space V and α is a scalar, then we denote by αv the scalar multiplication of v by α; then αv is another vector in V. If w is also a member of V, then we denote by $v + w$ the vector addition of v and w. The following properties serve to define a vector space, with u, v, and w denoting arbitrary members of V and α and β arbitrary scalars:

- 1. $v + w = w + v$;
- 2. $u + (v + w) = (u + v) + w$;
- 3. there is a unique "zero vector" , denoted 0, such that, for every v, $v + 0 = v$;
- 4. for each v there is a unique vector $-v$ such that $v + (-v) = 0$;
- 5. $1v = v$, for all v;
- 6. $(\alpha\beta)v = \alpha(\beta v)$;
- 7. $\alpha(v + w) = \alpha v + \alpha w$;
- 8. $(\alpha + \beta)v = \alpha v + \beta v$.

Ex. 3.1 *Show that, if $z + z = z$, then z is the zero vector.*

Ex. 3.2 *Prove that $0v = 0$, for all $v \in V$, and use this to prove that $(-1)v = -v$ for all $v \in V$. Hint: Two different "zeros" are being used here. The first is the real number zero and the second is the zero vector in V. Use Exercise 3.1.*

We then write

$$w - v = w + (-v) = w + (-1)v,$$

for all v and w.

If $u^1, ..., u^N$ are members of V and $\alpha_1, ..., \alpha_N$ are scalars, then the vector

$$x = \alpha_1 u^1 + \alpha_2 u^2 + ... + \alpha_N u^N$$

is called a *linear combination* of the vectors $u^1, ..., u^N$, with coefficients $\alpha_1, ..., \alpha_N$.

If W is a subset of a vector space V, then W is called a *subspace* of V if W is also a vector space for the same operations. What this means is simply that when we perform scalar multiplication on a vector in W, or when we add vectors in W, we always get members of W back again. Another way to say this is that W is *closed to linear combinations.*

When we speak of subspaces of V we do not mean to exclude the case of $W = V$. Note that V is itself a subspace, but not a *proper subspace* of V. Every subspace must contain the zero vector, 0; the smallest subspace of V is the subspace containing only the zero vector, $W = \{0\}$.

Ex. 3.3 *Show that, in the vector space $V = \mathbb{R}^2$, the subset of all vectors whose entries sum to zero is a subspace, but the subset of all vectors whose entries sum to one is not a subspace.*

Ex. 3.4 *Let V be a vector space, and W and Y subspaces of V. Show that the union of W and Y, written $W \cup Y$, is also a subspace if and only if either $W \subseteq Y$ or $Y \subseteq W$.*

We often refer to things like $\begin{bmatrix} 1 & 2 & 0 \end{bmatrix}$ as vectors, although they are but one example of a certain type of vector. For clarity, in this book we shall call such an object a *real row vector of dimension three* or a *real row three-vector.* Similarly, we shall call $\begin{bmatrix} 3i \\ -1 \\ 2+i \\ 6 \end{bmatrix}$ a *complex column vector of dimension four* or a *complex column four-vector.* For notational convenience, whenever we refer to something like a real three-vector or a complex four-vector, we shall always mean that they are columns, rather than rows. The space of

real (column) N-vectors will be denoted $\mathbb{R}^N$, while the space of complex (column) N vectors is $\mathbb{C}^N$.

Shortly after beginning a discussion of vector spaces, we arrive at the notion of the size or dimension of the vector space. A vector space can be finite dimensional or infinite dimensional. The spaces $\mathbb{R}^N$ and $\mathbb{C}^N$ have dimension N; not a big surprise. The vector spaces of all infinite sequences of real or complex numbers are infinite dimensional, as is the vector space of all real or complex polynomials. If we choose to go down the path of finite dimensionality, we very quickly find ourselves talking about matrices. If we go down the path of infinite dimensionality, we quickly begin to discuss convergence of infinite sequences and sums, and find that we need to introduce norms, which takes us into functional analysis and the study of Hilbert and Banach spaces. In this course we shall consider only the finite dimensional vector spaces, which means that we shall be talking mainly about matrices.

3.3 Matrix Algebra

A system $Ax = b$ of linear equations is called a *complex system*, or a *real system* if the entries of A, x and b are complex, or real, respectively. Note that when we say that the entries of a matrix or a vector are complex, we do not intend to rule out the possibility that they are real, but just to open up the possibility that they are not real.

3.3.1 Matrix Operations

If A and B are real or complex M by N and N by K matrices, respectively, then the product $C = AB$ is defined as the M by K matrix whose entry C_{mk} is given by

$$C_{mk} = \sum_{n=1}^{N} A_{mn} B_{nk}. \tag{3.1}$$

If x is an N-dimensional column vector, that is, x is an N by 1 matrix, then the product $b = Ax$ is the M-dimensional column vector with entries

$$b_m = \sum_{n=1}^{N} A_{mn} x_n. \tag{3.2}$$

Ex. 3.5 *Show that, for each $k = 1, ..., K$, $\mathrm{Col}_k(C)$, the kth column of the*

matrix $C = AB$, is

$$\mathrm{Col}_k(C) = A\mathrm{Col}_k(B).$$

It follows from this exercise that, for given matrices A and C, every column of C is a linear combination of the columns of A if and only if there is a third matrix B such that $C = AB$.

For any N, we denote by I the N by N identity matrix with entries $I_{n,n} = 1$ and $I_{m,n} = 0$, for $m, n = 1, ..., N$ and $m \neq n$. For every x we have $Ix = x$. We always speak of *the* identity matrix, although there is one for each N. The size of I is always to be inferred from the context.

The matrix $A^\dagger$ is the *conjugate transpose* of the matrix A, that is, the N by M matrix whose entries are

$$(A^\dagger)_{nm} = \overline{A_{mn}} \tag{3.3}$$

When the entries of A are real, $A^\dagger$ is just the *transpose* of A, written A^T.

Definition 3.1 *A square matrix S is* symmetric *if $S^T = S$ and* Hermitian *if $S^\dagger = S$.*

Definition 3.2 *A square matrix S is* normal *if $S^\dagger S = SS^\dagger$.*

Ex. 3.6 *Let $C = AB$. Show that $C^\dagger = B^\dagger A^\dagger$.*

Ex. 3.7 *Let D be a fixed diagonal matrix, that is, a square matrix such that $D_{mn} = 0$ whenever $m \neq n$. Suppose that $D_{mm} \neq D_{nn}$ if $m \neq n$. Show that if, for some matrix B, we have $BD = DB$, then B is a diagonal matrix.*

Ex. 3.8 *Prove that, if $AB = BA$ for every N by N matrix A, then $B = cI$, for some constant c.*

3.3.2 Matrix Inverses

We begin with the definition of invertibility.

Definition 3.3 *A square matrix A is said to be* invertible*, or to be a* non-singular *matrix, if there is a matrix B such that*

$$AB = BA = I$$

where I is the identity matrix of the appropriate size.

Note that, in this definition, the matrices A and B must commute, although, as we shall see, it is enough to require that $AB = I$.

Proposition 3.1 *If $AB = BA = I$ and $AC = CA = I$, then $B = C$.*

Ex. 3.9 *Prove Proposition 3.1.*

As a consequence of Proposition 3.1 we can make the following definition.

Definition 3.4 *Let A be square. If there is a matrix B with $AB = BA = I$, then B is called the* inverse *of A and we write $B = A^{-1}$.*

The following proposition shows that invertibility follows from an apparently weaker condition.

Proposition 3.2 *If A is square and there exist matrices B and C such that $AB = I$ and $CA = I$, then $B = C = A^{-1}$ and A is invertible.*

Ex. 3.10 *Prove Proposition 3.2.*

Later in this chapter, after we have discussed the concept of rank of a matrix, we will improve Proposition 3.2; a square matrix A is invertible if and only if there is a matrix B with $AB = I$, and, for any (possibly non-square) A, if there are matrices B and C with $AB = I$ and $CA = I$ (where the two I may possibly be different in size), then A must be square and invertible.

The 2 by 2 matrix $S = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ has an inverse

$$S^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

whenever the *determinant* of S, $\det(S) = ad - bc$, is not zero. More generally, associated with every complex square matrix is the complex number called its determinant, which is obtained from the entries of the matrix using formulas that can be found in any text on linear algebra. The significance of the determinant is that the matrix is invertible if and only if its determinant is not zero. This is of more theoretical than practical importance, since no computer can tell when a number is precisely zero. A matrix A that is not square cannot have an inverse, but does have a *pseudo-inverse*, which can be found using the singular-value decomposition.

Note that, if A is invertible, then $Ax = 0$ can happen only when $x = 0$. We shall show later, using the notion of the rank of a matrix, that the converse is also true: a square matrix A with the property that $Ax = 0$ only when $x = 0$ must be invertible.

3.3.3 The Sherman-Morrison-Woodbury Identity

In a number of applications, stretching from linear programming to radar tracking, we are faced with the problem of computing the inverse of a slightly modified version of a matrix B, when the inverse of B itself has already been computed. For example, when we use the simplex algorithm in linear programming, the matrix B consists of some, but not all, of the columns of a larger matrix A. At each step of the simplex algorithm, a new B_{new} is formed from $B = B_{\text{old}}$ by removing one column of B and replacing it with another column taken from A.

Then B_{new} differs from B in only one column. Therefore

$$B_{\text{new}} = B_{\text{old}} - uv^T, \tag{3.4}$$

where u is the column vector that equals the old column minus the new one, and v is the column of the identity matrix corresponding to the column of B_{old} being altered. The inverse of B_{new} can be obtained fairly easily from the inverse of B_{old} using the Sherman-Morrison-Woodbury Identity:

The Sherman-Morrison-Woodbury Identity: If $v^T B^{-1} u \neq 1$, then

$$(B - uv^T)^{-1} = B^{-1} + \alpha^{-1}(B^{-1}u)(v^T B^{-1}), \tag{3.5}$$

where

$$\alpha = 1 - v^T B^{-1} u.$$

Ex. 3.11 *Let B be invertible and $v^T B^{-1} u = 1$. Show that $B - uv^T$ is not invertible. Show that Equation (3.5) holds, if $v^T B^{-1} u \neq 1$. Hint: If $v^T B^{-1} u = 1$, then there is a nonzero vector w with $(B - uv^T)w = 0$; therefore, $B - uv^T$ cannot have an inverse. Find w.*

3.4 Bases and Dimension

The related notions of a basis and of linear independence are fundamental in linear algebra.

3.4.1 Linear Independence and Bases

As we shall see shortly, the *dimension* of a *finite-dimensional* vector space will be defined as the number of members of any basis. Obviously, we first need to see what a basis is, and then to convince ourselves that if a vector space V has a basis with N members, then every basis for V has N members.

Definition 3.5 *The* span *of a collection of vectors $\{u^1, ..., u^N\}$ in V is the set of all vectors x that can be written as linear combinations of the u^n; that is, for which there are scalars $\alpha_1, ..., \alpha_N$, such that*

$$x = \alpha_1 u^1 + ... + \alpha_N u^N. \tag{3.6}$$

Definition 3.6 *A collection of vectors $\{w^1, ..., w^N\}$ in V is called a* spanning set *for a subspace W if the set W is their span.*

Definition 3.7 *A subspace W of a vector space V is called* finite dimensional *if it is the span of a finite set of vectors from V. The whole space V is then finite dimensional if it is the span of a finite set of vectors.*

The assertion in the following proposition may seem obvious, but the proof, which the reader is asked to supply as Exercise 3.12, is surprisingly subtle. The point of Exercise 3.12 is to encourage the readers to discover, for themselves, some of the important notions to be defined and discussed shortly. Therefore, it is important that this exercise be attempted before reading further in the text.

Proposition 3.3 *Let V be a finite dimensional vector space and W a subspace of V. Then W is also finite dimensional.*

Ex. 3.12 *Prove Proposition 3.3.*

This definition tells us what it means to be finite dimensional, but does not tell us what *dimension* means, nor what the actual dimension of a finite dimensional subset is; for that we need the notions of *linear independence* and *basis*.

Definition 3.8 *A collection of vectors $\mathcal{U} = \{u^1, ..., u^N\}$ in V is* linearly independent *if there is no choice of scalars $\alpha_1, ..., \alpha_N$, not all zero, such that*

$$0 = \alpha_1 u^1 + ... + \alpha_N u^N. \tag{3.7}$$

Ex. 3.13 *Show that the following are equivalent:*

- *1. the set $\mathcal{U} = \{u^1, ..., u^N\}$ is linearly independent;*
- *2. $u^1 \neq 0$ and no u^n is a linear combination of the members of $\mathcal{U}$ that precede it in the list;*
- *3. no u^n is a linear combination of the other members of $\mathcal{U}$.*

Definition 3.9 *A collection of vectors $\mathcal{U} = \{u^1, ..., u^N\}$ in V is called a* basis *for a subspace W if the collection is linearly independent and W is their span.*

Ex. 3.14 *Show that*

- *1. if* $\mathcal{U} = \{u^1, ..., u^N\}$ *is a spanning set for* W*, then* $\mathcal{U}$ *is a basis for* W *if and only if, after the removal of any one member,* $\mathcal{U}$ *is no longer a spanning set for* W*; and*
- *2. if* $\mathcal{U} = \{u^1, ..., u^N\}$ *is a linearly independent set in* W*, then* $\mathcal{U}$ *is a basis for* W *if and only if, after including in* $\mathcal{U}$ *any new member from* W*,* $\mathcal{U}$ *is no longer linearly independent.*

Ex. 3.15 *Prove that every finite dimensional vector space that is not just the zero vector has a basis.*

3.4.2 Dimension

We turn now to the task of showing that every basis for a finite dimensional vector space has the same number of members. That number will then be used to define the dimension of that space.

Suppose that W is a subspace of V, that $\mathcal{W} = \{w^1, ..., w^N\}$ is a spanning set for W, and $\mathcal{U} = \{u^1, ..., u^M\}$ is a linearly independent subset of W. Beginning with w^1, we augment the set $\{u^1, ..., u^M\}$ with w^j if w^j is not in the span of the u^m and the w^k previously included. At the end of this process, we have a linearly independent spanning set, and therefore, a basis, for W (Why?). Similarly, beginning with w^1, we remove w^j from the set $\{w^1, ..., w^N\}$ if w^j is a linear combination of the w^k, $k = 1, ..., j-1$. In this way we obtain a linearly independent set that spans W, hence another basis for W. The following lemma will allow us to prove that all bases for a subspace W have the same number of elements.

Lemma 3.1 *Let* $\mathcal{W} = \{w^1, ..., w^N\}$ *be a spanning set for a subspace* W *of* V*, and* $\mathcal{U} = \{u^1, ..., u^M\}$ *a linearly independent subset of* W*. Then* $M \leq N$*.*

Proof: Suppose that $M > N$. Let $B_0 = \mathcal{W} = \{w^1, ..., w^N\}$. To obtain the set B_1, form the set $C_1 = \{u^1, w^1, ..., w^N\}$ and remove the first member of C_1 that is a linear combination of members of C_1 that occur to its left in the listing; since u^1 has no members to its left, it is not removed. Since $\mathcal{W}$ is a spanning set, $u^1 \neq 0$ is a linear combination of the members of $\mathcal{W}$, so that some member of $\mathcal{W}$ is a linear combination of u^1 and the members of $\mathcal{W}$ to the left of it in the list; remove the first member of $\mathcal{W}$ for which this is true.

We note that the set B_1 is a spanning set for W and has N members. Having obtained the spanning set B_k, with N members and whose first k members are $u^k, ..., u^1$, we form the set $C_{k+1} = B_k \cup \{u^{k+1}\}$, listing the

members so that the first $k+1$ of them are $\{u^{k+1}, u^k, ..., u^1\}$. To get the set B_{k+1} we remove the first member of C_{k+1} that is a linear combination of the members to its left; there must be one, since B_k is a spanning set, and so u^{k+1} is a linear combination of the members of B_k. Since the set $\mathcal{U}$ is linearly independent, the member removed is from the set $\mathcal{W}$. Continuing in this fashion, we obtain a sequence of spanning sets $B_1, ..., B_N$, each with N members. The set B_N is $B_N = \{u^N, ..., u^1\}$ and u^{N+1} must then be a linear combination of the members of B_N, which contradicts the linear independence of $\mathcal{U}$. ∎

Corollary 3.1 *Every basis for a subspace W has the same number of elements.*

Definition 3.10 *The* dimension of a subspace W, *denoted* $\dim(W)$, *is the number of elements in any basis.*

Ex. 3.16 *Let V be a finite dimensional vector space and W any subspace of V. Show that* $\dim(W)$ *cannot exceed* $\dim(V)$.

3.4.3 Rank of a Matrix

We rely on the following lemma to define the rank of a matrix.

Lemma 3.2 *For any matrix A, the maximum number of linearly independent rows equals the maximum number of linearly independent columns.*

Proof: Suppose that A is an M by N matrix, and that $K \leq N$ is the maximum number of linearly independent columns of A. Select K linearly independent columns of A and use them as the K columns of an M by K matrix U. Since every column of A must be a linear combination of these K selected ones, there is a K by N matrix B such that $A = UB$; see the discussion that follows Exercise 3.5. From $A^\dagger = B^\dagger U^\dagger$ we conclude that every column of $A^\dagger$ is a linear combination of the K columns of the matrix $B^\dagger$. Therefore, there can be at most K linearly independent columns of $A^\dagger$. ∎

Definition 3.11 *The* rank *of A, written rank(A), is the maximum number of linearly independent rows or of linearly independent columns of A.*

Ex. 3.17 *Let u and v be two non-zero N-dimensional complex column vectors. Show that the rank of the N by N matrix $uv^\dagger$ is one.*

Ex. 3.18 *Show that the rank of a matrix $C = AB$ is never greater than the smaller of the rank of A and the rank of B. Can it ever be strictly less than the smaller of these two numbers?*

Ex. 3.19 *Show that rank(A + B) is never greater than the sum of rank(A) and rank(B).*

Definition 3.12 *An M by N matrix A is said to have* full rank *or to be a* full-rank matrix *if the rank of A is the minimum of M and N.*

Proposition 3.4 *A square matrix is invertible if and only if it has full rank.*

Ex. 3.20 *Prove Proposition 3.4.*

Corollary 3.2 *A square matrix A is invertible if and only if there is a matrix B such that AB = I.*

Corollary 3.3 *A square matrix A is invertible if and only if there is a matrix G such that AG is invertible.*

Corollary 3.4 *If A and B are square matrices and C = AB is invertible, then both A and B are invertible.*

Definition 3.13 *An M by N matrix A is said to have* left inverse *B if B is an N by M matrix such that BA = I_N, the N by N identity matrix. Similarly, A is said to have a* right inverse *C if C is an N by M matrix such that AC = I_M, the M by M identity matrix.*

Ex. 3.21 *Let A be an M by N matrix. When does A have a left inverse? When does it have a right inverse? Give your answer in terms of the rank of the matrix A.*

Ex. 3.22 *Let A and B be M by N matrices, P an invertible M by M matrix, and Q an invertible N by N matrix, such that B = PAQ, that is, the matrices A and B are* equivalent*. Show that the rank of B is the same as the rank of A. Hint: show that A and AQ have the same rank.*

3.5 Representing a Linear Transformation

Let V and W be vector spaces. A function $T : V \to W$ is called a *linear transformation* if

$$T(\alpha u + \beta v) = \alpha T(u) + \beta T(v),$$

for all scalars α and β and all u and v in V. For notational convenience we often write simply Tu instead of $T(u)$. When both V and W are finite-dimensional a linear transformation can be represented by a matrix, which

is why we say that there is a close relationship between abstract linear algebra and matrix theory.

Let $\mathcal{A} = \{a^1, a^2, ..., a^N\}$ be a basis for the finite-dimensional complex vector space V. Now that the basis for V is specified, there is a natural association, an *isomorphism*, between V and the vector space $\mathbb{C}^N$ of N-dimensional column vectors with complex entries. Any vector v in V can be written as

$$v = \sum_{n=1}^{N} \gamma_n a^n. \tag{3.8}$$

The column vector $\gamma = (\gamma_1, ..., \gamma_N)^T$ is uniquely determined by v and the basis $\mathcal{A}$ and we denote it by $[v]_{\mathcal{A}}$. Notice that the ordering of the list of members of $\mathcal{A}$ matters, so we shall always assume that the ordering has been fixed.

Let W be a second finite-dimensional vector space, and let T be any linear transformation from V to W. Let $\mathcal{B} = \{b^1, b^2, ..., b^M\}$ be a basis for W. For $n = 1, ..., N$, let

$$Ta^n = A_{1n}b^1 + A_{2n}b^2 + ... + A_{Mn}b^M. \tag{3.9}$$

Then the M by N matrix A having the A_{mn} as entries is said to *represent* T, with respect to the bases $\mathcal{A}$ and $\mathcal{B}$, and we write $A = [T]_{\mathcal{A}}^{\mathcal{B}}$.

Ex. 3.23 *Show that* $[Tv]_{\mathcal{B}} = A[v]_{\mathcal{A}}$.

Ex. 3.24 *Let P_2 and P_3 be the vector spaces of real polynomials of degrees two and three, respectively. Let $T : P_3 \to P_2$ be the differentiation operator. Select bases for P_2 and P_3 and represent T by matrix multiplication.*

Ex. 3.25 *Suppose that V, W and Z are vector spaces, with bases $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, respectively. Suppose also that T is a linear transformation from V to W and U is a linear transformation from W to Z. Let A represent T with respect to the bases $\mathcal{A}$ and $\mathcal{B}$, and let B represent U with respect to the bases $\mathcal{B}$ and $\mathcal{C}$. Show that the matrix BA represents the linear transformation UT with respect to the bases $\mathcal{A}$ and $\mathcal{C}$.*

3.6 The Geometry of Euclidean Space

We denote by $\mathbb{R}^N$ the real Euclidean space consisting of all N-dimensional column vectors $x = (x_1, ..., x_N)^T$ with real entries x_j; here the

superscript T denotes the transpose of the 1 by N matrix (or, row vector) $(x_1, ..., x_N)$. We denote by $\mathbb{C}^N$ the space of all N-dimensional column vectors with complex entries. For x in $\mathbb{C}^N$ we denote by $x^\dagger$ the N-dimensional row vector whose entries are the complex conjugates of the entries of x.

3.6.1 Dot Products

For $x = (x_1, ..., x_N)^T$ and $y = (y_1, ..., y_N)^T$ in $\mathbb{C}^N$, the dot product $x \cdot y$ is defined to be

$$x \cdot y = \sum_{n=1}^{N} x_n \overline{y_n}. \tag{3.10}$$

Note that we can write

$$x \cdot y = y^\dagger x, \tag{3.11}$$

where juxtaposition indicates matrix multiplication. The 2-norm, or *Euclidean norm*, or *Euclidean length*, of x is

$$\|x\|_2 = \sqrt{x \cdot x} = \sqrt{x^\dagger x}. \tag{3.12}$$

The *Euclidean distance* between two vectors x and y in $\mathbb{C}^N$ is $\|x - y\|_2$. These notions also apply to vectors in $\mathbb{R}^N$.

In later chapters we shall consider vector norms other than the two-norm. However, for the remainder of this chapter, the only vector norm we shall consider is the two-norm.

The spaces $\mathbb{R}^N$ and $\mathbb{C}^N$, along with their dot products, are examples of a finite-dimensional Hilbert space.

Definition 3.14 *Let V be a real or complex vector space. The scalar-valued function $\langle u, v\rangle$ is called an* inner product *on V if the following four properties hold, for all u, w, and v in V, and all scalars α:*

$$\langle u + w, v\rangle = \langle u, v\rangle + \langle w, v\rangle; \tag{3.13}$$

$$\langle \alpha u, v\rangle = \alpha \langle u, v\rangle; \tag{3.14}$$

$$\langle v, u\rangle = \overline{\langle u, v\rangle}; \tag{3.15}$$

and

$$\langle u, u\rangle \geq 0, \tag{3.16}$$

with equality in Inequality (3.16) if and only if $u = 0$.

Once we have an inner product on the vector space V we also have a norm, denoted $\|\cdot\|_2$ defined by

$$\|u\|_2^2 = \langle u, u \rangle.$$

The dot products on $\mathbb{R}^N$ and $\mathbb{C}^N$ are examples of inner products. The properties of an inner product are precisely the ones needed to prove Cauchy's Inequality, which then holds for any inner product. We shall favor the dot product notation $u \cdot v$ for the inner product of vectors in $\mathbb{R}^N$ or $\mathbb{C}^N$, although we shall occasionally use the matrix multiplication form, $v^\dagger u$ or the inner product notation $\langle u, v \rangle$.

Ex. 3.26 *Show that, for any real number λ, we have*

$$\|\lambda x + (1-\lambda)y\|_2^2 + \lambda(1-\lambda)\|x-y\|_2^2 = \lambda\|x\|_2^2 + (1-\lambda)\|y\|_2^2. \quad (3.17)$$

We may conclude from Exercise 3.26 that, for any α in the interval $(0,1)$ and x not equal to y, we have

$$\|\alpha x + (1-\alpha)y\|_2^2 < \alpha\|x\|_2^2 + (1-\alpha)\|y\|_2^2, \quad (3.18)$$

so that the square of the two-norm is a strictly convex function.

Definition 3.15 *A collection of vectors $\{u^1, ..., u^N\}$ in an inner product space V is called* orthonormal *if $\|u^n\|_2 = 1$, for all n, and $\langle u^m, u^n \rangle = 0$, for $m \neq n$.*

3.6.2 Cauchy's Inequality

Cauchy's Inequality, also called the Cauchy-Schwarz Inequality, tells us that

$$|\langle x, y \rangle| \leq \|x\|_2 \|y\|_2, \quad (3.19)$$

with equality if and only if $y = \alpha x$, for some scalar α. The Cauchy-Schwarz Inequality holds for any inner product.

A simple application of Cauchy's Inequality gives us

$$\|x + y\|_2 \leq \|x\|_2 + \|y\|_2; \quad (3.20)$$

this is called the *Triangle Inequality*. We say that the vectors x and y are *mutually orthogonal* if $\langle x, y \rangle = 0$.

The *Parallelogram Law* is an easy consequence of the definition of the 2-norm:

$$\|x + y\|_2^2 + \|x - y\|_2^2 = 2\|x\|_2^2 + 2\|y\|_2^2. \quad (3.21)$$

It is important to remember that Cauchy's Inequality and the Parallelogram Law hold only for the 2-norm.

3.6.3 An Alternative Approach to Orthogonality

A more geometric approach to Cauchy's Inequality begins with an alternative approach to orthogonality [79]. Let x and y be nonzero vectors in $\mathbb{R}^N$. Say that x is orthogonal to y if

$$\|x - y\|_2 = \|x + y\|_2. \tag{3.22}$$

To visualize this, draw a triangle with vertices x, y and $-y$.

Ex. 3.27 *Show that x and y are orthogonal if and only if $x \cdot y = 0$ and if and only if Pythagoras' Theorem holds; that is,*

$$\|x - y\|_2^2 = \|x\|_2^2 + \|y\|_2^2.$$

Ex. 3.28 *Let p be the orthogonal projection of x on the line determined by y and the origin. Then $p = \gamma y$ for some constant γ. Find γ using the fact that y and $x - \gamma y$ are orthogonal. Use Pythagoras' Theorem to obtain Cauchy's Inequality.*

Ex. 3.29 *Define the angle between vectors x and y to be α such that*

$$\cos\alpha = \frac{x \cdot y}{\|x\|_2 \|y\|_2}.$$

Use this to prove the Law of Cosines and the Triangle Inequality.

3.7 Vectorization of a Matrix

When the complex M by N matrix A is stored in the computer it is usually *vectorized*; that is, the matrix

$$A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1N} \\ A_{21} & A_{22} & \dots & A_{2N} \\ \vdots & & & \\ A_{M1} & A_{M2} & \dots & A_{MN} \end{bmatrix}$$

becomes

$$\text{vec}(A) = (A_{11}, A_{21}, ..., A_{M1}, A_{12}, A_{22}, ..., A_{M2}, ..., A_{MN})^T.$$

Definition 3.16 *The* trace *of a square matrix A, abbreviated* $\mathrm{tr}(A)$*, is the sum of the entries on its main diagonal; that is, for an N by N matrix A, we have*

$$\mathrm{tr}(A) = \sum_{n=1}^{N} A_{n,n}.$$

It can be shown that

$$\mathrm{tr}(ABC) = \mathrm{tr}(CAB) = \mathrm{tr}(BCA), \tag{3.23}$$

but it is not generally true that $\mathrm{tr}(ABC) = \mathrm{tr}(BAC)$.

Ex. 3.30 *Let A be an M by N matrix and B an N by M matrix. We know that we need not have $AB = BA$, even when $M = N$. However, AB and BA do share some properties. Show that AB and BA have the same trace.*

Ex. 3.31

- **a)** *Show that the complex dot product* $\mathrm{vec}(A)\cdot\mathrm{vec}(B) = \mathrm{vec}(B)^\dagger\mathrm{vec}(A)$ *can be obtained by*

$$\mathrm{vec}(A)\cdot\mathrm{vec}(B) = \mathrm{trace}\,(AB^\dagger) = tr(AB^\dagger).$$

We can therefore use the trace to define an inner product between matrices: $< A, B >= \mathrm{trace}\,(AB^\dagger)$*. This inner product is called the trace inner product.*

- **b)** *Show that* $\mathrm{trace}\,(AA^\dagger) \geq 0$ *for all A, so that we can use the trace to define a norm on matrices:* $\|A\|_F^2 = \mathrm{trace}\,(AA^\dagger)$*. This norm is the Frobenius norm*

3.8 Solving Systems of Linear Equations

In this section we discuss systems of linear equations, Gaussian elimination, and the notions of basic and non-basic variables.

3.8.1 Row-Reduction

One approach to solving systems of linear equations is to use elementary row operations to convert the original system to another system with the same solutions.

Definition 3.17 *There are three types of* elementary row operations*. The first is to multiply a given row by a scalar. The second is to switch two rows. The third is to add to a given row some multiple of another row.*

Definition 3.18 *An M by N matrix B is said to be in* row-reduced echelon form *if the following conditions hold:*

- *1. the first non-zero entry of any row is a one;*
- *2. in any column containing one of these "first non-zero" ones, the remaining entries are zero;*
- *3. all zero rows come at the bottom; and*
- *4. if $j < k$ then the column containing the first non-zero entry of the jth row occurs before the column containing the first non-zero entry of the kth row.*

Lemma 3.3 *Any matrix A can be transformed into a matrix B in row-reduced echelon form using elementary row operations.*

Ex. 3.32 *Prove Lemma 3.3.*

Proposition 3.5 *Let A be an M by N matrix with rank R. Then there are invertible matrices P and Q such that PAQ is a diagonal matrix with the entries of the R by R identity matrix in the upper left corner and all the rest of the entries equal to zero.*

Proof: We know that any matrix A can be transformed to row-reduced echelon form using row operations, or, equivalently, by multiplying A on the left by elementary matrices. The proof follows by applying the same reasoning to $A^\dagger$. ∎

Proposition 3.6 *Let A be an arbitrary M by N matrix and B the matrix in row-reduced echelon form obtained from A. There is a non-zero solution of the system of linear equations $Ax = 0$ if and only if B has fewer than N non-zero rows.*

Ex. 3.33 *Prove Proposition 3.6.*

Corollary 3.5 *If A is M by N and $M < N$, then there is a non-zero x with $Ax = 0$.*

Ex. 3.34 *Prove Corollary 3.5.*

Ex. 3.35 *Let $\mathcal{W} = \{w^1, ..., w^N\}$ be a spanning set for a subspace W in $\mathbb{R}^K$, and $\mathcal{U} = \{u^1, ..., u^M\}$ a linearly independent subset of W. Let A be the K by M matrix whose columns are the vectors u^m and B the K by N matrix whose columns are the w^n. Then there is an N by M matrix D such that $A = BD$ (Why?). Prove Lemma 3.1 for this case by showing that, if $M > N$, then there is a non-zero vector x with $Dx = 0$.*

Definition 3.19 *Let A be an M by N matrix. The* null space *of A, denoted $NS(A)$, is the set of all x such that $Ax = 0$. The* nullity *of A, denoted $n(A)$, is the dimension of its null space.*

Proposition 3.7 *Let A be an N by N matrix with rank $J < N$. Then there are $N - J$ linearly independent solutions of the system $Ax = 0$, and the null space of A has dimension $N - J$.*

Ex. 3.36 *Prove Proposition 3.7.*

3.8.2 Row Operations as Matrix Multiplications

Suppose that we want to apply a row operation to the M by N matrix A. We can first apply that row operation to the M by M identity matrix, to obtain the new matrix E, and then multiply A by E on the left. The matrix EA is exactly what we would have obtained if we had just performed the row operation on A directly. For example, to multiply the first row of A by k we could multiply A by the matrix $E_1(k)$, which is the identity matrix, except that the one in the first row is replaced by k.

If A is square and we are able to row reduce A to the identity matrix I, then there are matrices E_1, E_2,...,E_J such that

$$E_J E_{J-1} \cdots E_2 E_1 A = I.$$

It follows then that

$$E_J E_{J-1} \cdots E_2 E_1 = A^{-1}.$$

We can also use this approach to calculate the determinant of A.

3.8.3 Determinants

Associated with each square matrix A is a number, its determinant, denoted $\det(A)$. Most texts that discuss determinants define the concept by telling us how to compute it. There is a different way that is more interesting (see [107]).

We define the determinant to be a complex-valued function of square complex matrices having the following two properties:

- 1. $\det(AB) = \det(A)\det(B)$ for all compatible square matrices A and B;
- 2. the determinant of the matrix $E_1(k)$ is k, where $E_1(k)$ is as defined in the previous subsection.

Using only these two properties, we can prove the following concerning the effects of row operations on the determinant of A:

- 1. multiplying one row by k multiplies the determinant by k;
- 2. interchanging two rows changes the sign of the determinant;
- 3. adding to one row a multiple of another row has no effect on the determinant.

Ex. 3.37 *Prove these assertions concerning the effects of row operations on the determinant. Hint: Consider writing row operations in more than one way. For example, we can switch rows one and two as follows: add row one to row two; subtract row two from row one; add row one to row two; and finally, multiply row one by* -1.

Note that, if a square matrix B has a zero row, then its determinant must be zero; switching rows if necessary, we may assume that the first row is zero, so that multiplying row one by k doesn't change B.

Of course, it remains to be shown that such a function of square matrices exists. To show the existence of such a function it is sufficient to show how it may be calculated, for any given square matrix A. Once we have converted A to an upper triangular matrix using row operations we can calculate the determinant of A immediately, since the determinant of an upper triangular matrix can easily be shown to be the product of the entries along its main diagonal. If we prefer, we can use more row operations to convert A to row-reduced echelon form. If A is invertible, this reduction produces the identity matrix, with determinant equal to one. If A is not invertible, the row-reduced echelon form will have a zero row, so that the determinant is zero. We have the following proposition.

Proposition 3.8 *A square matrix is invertible if and only if its determinant is not zero.*

Proof: If A is invertible, then $\det(A)\det(A^{-1}) = \det(I) = 1$, so $\det(A)$ is nonzero. Conversely, if $\det(A)$ is not zero, then its row reduced echelon form matrix R must also have $\det(R)$ nonzero, since each row operation is invertible. Therefore, $R = I$, and A^{-1} is the product of the row-reduction matrices used to go from A to R. ∎

3.8.4 Homogeneous Systems of Linear Equations

Consider the system of three linear equations in five unknowns given by

$$\begin{aligned} x_1 + 2x_2 \ + 2x_4 + x_5 &= 0 \\ -x_1 - x_2 + x_3 + x_4 \ &= 0 \\ x_1 + 2x_2 - 3x_3 - x_4 - 2x_5 &= 0. \end{aligned} \tag{3.24}$$

This system can be written in matrix form as $Ax = 0$, with A the coefficient matrix

$$A = \begin{bmatrix} 1 & 2 & 0 & 2 & 1 \\ -1 & -1 & 1 & 1 & 0 \\ 1 & 2 & -3 & -1 & -2 \end{bmatrix}, \tag{3.25}$$

and $x = (x_1, x_2, x_3, x_4, x_5)^T$.

The standard approach to solving a system of M equations in N unknowns is to apply Gaussian elimination, to obtain a second, simpler, system with the same solutions. To avoid potential numerical difficulties, Gauss elimination may involve *row pivoting*, which means that when we are about to eliminate the variable x_k from the equations $k+1$ through M, we switch the kth row with the one below it that has the coefficient of x_k with the largest absolute value. In the example below we do not employ pivoting.

Using Gaussian elimination, we obtain the equivalent system of equations

$$x_1 - 2x_4 + x_5 = 0$$

$$x_2 + 2x_4 = 0$$

$$x_3 + x_4 + x_5 = 0. \tag{3.26}$$

The associated matrix is R, the row reduced echelon form matrix obtained from A:

$$R = \begin{bmatrix} 1 & 0 & 0 & -2 & 5 \\ 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}. \tag{3.27}$$

From this simpler system we see that the variables x_4 and x_5 can be freely chosen, with the other three variables then determined by this system of equations. The variables x_4 and x_5 are then *independent*, the others *dependent*. The variables x_1, x_2 and x_3 are then called *basic variables*; note that this terminology is commonly used in linear programming, but has nothing to do with the notion of a basis. To obtain a basis of solutions we can let $x_4 = 1$ and $x_5 = 0$, obtaining the solution $x = (2, -2, -1, 1, 0)^T$, and then choose $x_4 = 0$ and $x_5 = 1$ to get the solution $x = (-1, 0, -1, 0, 1)^T$. Every solution to $Ax = 0$ is then a linear combination of these two solutions. Notice that which variables are basic and which are non-basic is somewhat arbitrary, in that we could have chosen as the non-basic variables any two whose columns are independent.

Having decided that x_4 and x_5 are the non-basic variables, we can write

the original matrix A in block-matrix form as $A = \begin{bmatrix} B & C \end{bmatrix}$, where B is the square invertible matrix

$$B = \begin{bmatrix} 1 & 2 & 0 \\ -1 & -1 & 1 \\ 1 & 2 & -3 \end{bmatrix}, \tag{3.28}$$

and C is the matrix

$$C = \begin{bmatrix} 2 & 1 \\ 1 & 0 \\ -1 & -2 \end{bmatrix}. \tag{3.29}$$

With $x_B = (x_1, x_2, x_3)^T$ and $x_C = (x_4, x_5)^T$ the vector x can be written in concatenated form as a block matrix, that is,

$$x = \begin{bmatrix} x_B^T & x_C^T \end{bmatrix}^T = \begin{bmatrix} x_B \\ x_C \end{bmatrix}.$$

Now we can write

$$Ax = Bx_B + Cx_C = 0, \tag{3.30}$$

so that

$$x_B = -B^{-1}Cx_C. \tag{3.31}$$

3.8.5 Real and Complex Systems of Linear Equations

Any complex system can be converted to a real system in the following way. A complex matrix A can be written as $A = A_1 + iA_2$, where A_1 and A_2 are real matrices and $i = \sqrt{-1}$. Similarly, $x = x^1 + ix^2$ and $b = b^1 + ib^2$, where x^1, x^2, b^1 and b^2 are real vectors. Denote by $\tilde{A}$ the real matrix

$$\tilde{A} = \begin{bmatrix} A_1 & -A_2 \\ A_2 & A_1 \end{bmatrix}, \tag{3.32}$$

by $\tilde{x}$ the real vector

$$\tilde{x} = \begin{bmatrix} x^1 \\ x^2 \end{bmatrix}, \tag{3.33}$$

and by $\tilde{b}$ the real vector

$$\tilde{b} = \begin{bmatrix} b^1 \\ b^2 \end{bmatrix}. \tag{3.34}$$

Then x satisfies the system $Ax = b$ if and only if $\tilde{x}$ satisfies the system $\tilde{A}\tilde{x} = \tilde{b}$.

3.9 Under-Determined Systems of Linear Equations

Suppose that $Ax = b$ is a linear system of M equations in N unknowns, where $M < N$. Then we say that the system is *under-determined*. Typically, there will be an infinite number of solutions, although there need not be any solutions in this case (give an example). A standard procedure in such cases is to find a solution x having the smallest two-norm

$$\|x\|_2 = \sqrt{\sum_{n=1}^{N} |x_n|^2}.$$

As we shall see shortly, a *minimum two-norm* solution of $Ax = b$ is a vector of the form $x = A^\dagger z$, where $A^\dagger$ denotes the conjugate transpose of the matrix A. Then $Ax = b$ becomes $AA^\dagger z = b$. Typically, $(AA^\dagger)^{-1}$ will exist, and we get $z = (AA^\dagger)^{-1}b$, from which it follows that the minimum two-norm solution is $x = A^\dagger(AA^\dagger)^{-1}b$. When M and N are not too large, forming the matrix $AA^\dagger$ and solving for z is not prohibitively expensive or time-consuming. However, in image processing the vector x is often a vectorization of a two-dimensional (or even three-dimensional) image and M and N can be on the order of tens of thousands or more. The ART algorithm gives us a fast method for finding a minimum two-norm solution without computing $AA^\dagger$.

We begin by describing a minimum two-norm solution of a consistent system $Ax = b$, starting with the fundamental *subspace decomposition* lemma.

Lemma 3.4 *For every x in $\mathbb{C}^N$ there are unique vectors $A^\dagger z$ in the range of $A^\dagger$ and w in the null space of A, such that $x = A^\dagger z + w$. The z need not be unique.*

Proof: Any z that minimizes the function

$$f(z) = \frac{1}{2}\|x - A^\dagger z\|_2^2$$

satisfies the equation

$$0 = \nabla f(z) = A(x - A^\dagger z).$$

Then $w = x - A^\dagger z$ satisfies $Aw = 0$. Expanding $\|x\|^2 = \|A^\dagger z + w\|^2$ and using the fact that $Aw = 0$ we find that

$$\|x\|^2 = \|A^\dagger z\|^2 + \|w\|^2.$$

If we also had

$$x = A^\dagger \hat{z} + \hat{w},$$

with $A\hat{w} = 0$, then, writing

$$0 = (A^\dagger z - A^\dagger \hat{z}) + (\hat{w} - w),$$

we get

$$0 = \|A^\dagger z - A^\dagger \hat{z}\|^2 + \|\hat{w} - w\|^2.$$

It follows then that $\hat{w} = w$ and that $A^\dagger \hat{z} = A^\dagger z$. ∎

Corollary 3.6 *For every M by N matrix A and every b in $\mathbb{C}^M$ there are unique vectors Ax and w in $\mathbb{C}^M$ such that $A^\dagger w = 0$ and $b = Ax + w$. The x is a minimizer of $\|b - Ax\|_2$ and need not be unique.*

Corollary 3.7 *An N by N matrix A is invertible if and only if $Ax = 0$ implies $x = 0$.*

Proof: If A is invertible and $Ax = 0$, then clearly we must have $x = 0$. Conversely, suppose that $Ax = 0$ only when $x = 0$. Then the null space of A is the subspace of $\mathbb{C}^N$ consisting only of the zero vector. Consequently, every vector in $\mathbb{C}^N$ lies in the column space of $A^\dagger$, so that N is the rank of $A^\dagger$, which is also the rank of A. So A has full rank and A must be invertible. ∎

Theorem 3.1 *A minimum two-norm solution of $Ax = b$ has the form $x = A^\dagger z$ for some M-dimensional complex vector z.*

Proof: If $Ax = b$ then $A(x + w) = b$ for all w in the null space of A. If $x = A^\dagger z$ and w is in the null space of A, then

$$\begin{aligned}\|x + w\|_2^2 = \|A^\dagger z + w\|_2^2 &= (A^\dagger z + w)^\dagger (A^\dagger z + w) \\ &= (A^\dagger z)^\dagger (A^\dagger z) + (A^\dagger z)^\dagger w + w^\dagger (A^\dagger z) + w^\dagger w \\ &= \|A^\dagger z\|_2^2 + (A^\dagger z)^\dagger w + w^\dagger (A^\dagger z) + \|w\|_2^2 \\ &= \|A^\dagger z\|_2^2 + \|w\|_2^2,\end{aligned}$$

since

$$w^\dagger (A^\dagger z) = (Aw)^\dagger z = 0^\dagger z = 0$$

and

$$(A^\dagger z)^\dagger w = z^\dagger A w = z^\dagger 0 = 0.$$

Therefore, $\|x + w\|_2 = \|A^\dagger z + w\|_2 > \|A^\dagger z\|_2 = \|x\|_2$ unless $w = 0$. This completes the proof. ∎

In a later chapter we shall consider other approaches to solving underdetermined systems of linear equations.

3.10 Over-Determined Systems of Linear Equations

When there are more equations than there are unknowns in the system $Ax = b$ we say that the system is *over-determined*; it is most likely then that there will be no exact solution, although there may be (give an example). In such cases, it is common to seek a *least squares* solution. A least squares solution is not an exact solution of $Ax = b$ when none exist, but rather an exact solution of the system $A^\dagger Ax = A^\dagger b$. A least squares solution is a minimizer of the function

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2.$$

Ex. 3.38 *Let A be an M by N matrix with complex entries. View A as a linear function with domain $\mathbb{C}^N$, the space of all N-dimensional complex column vectors, and range contained within $\mathbb{C}^M$, via the expression $A(x) = Ax$. Suppose that $M > N$. The range of A, denoted $R(A)$, cannot be all of $\mathbb{C}^M$. Show that every vector z in $\mathbb{C}^M$ can be written uniquely in the form $z = Ax + w$, where $A^\dagger w = 0$. Show that $\|z\|_2^2 = \|Ax\|_2^2 + \|w\|_2^2$, where $\|z\|_2^2$ denotes the square of the two-norm of z. Hint: If $z = Ax + w$ then consider $A^\dagger z$. Assume $A^\dagger A$ is invertible.*

3.11 Eigenvalues and Eigenvectors

Let A be a complex M by N matrix. It is often helpful to know how large the two-norm $\|Ax\|_2$ can be, relative to $\|x\|_2$; that is, we want to find a constant a so that

$$\|Ax\|_2/\|x\|_2 \leq a,$$

for all $x \neq 0$. We can reformulate the problem by asking how large $\|Au\|_2^2$ can be, subject to $\|u\|_2 = 1$. Using Lagrange multipliers, we discover that a unit vector u that maximizes $\|Au\|_2^2$ has the property that

$$A^\dagger Au = \lambda u,$$

for some constant λ. This leads to the more general problem discussed in this section.

Definition 3.20 *Given an N by N complex matrix S, we say that a complex number λ is an* eigenvalue *of S if there is a nonzero vector u with $Su = \lambda u$. The column vector u is then called an* eigenvector *of S associated with eigenvalue λ.*

Clearly, if u is an eigenvector of S, then so is cu, for any constant $c \neq 0$; therefore, it is common to choose eigenvectors to have norm equal to one.

If λ is an eigenvalue of S, then the matrix $S-\lambda I$ fails to have an inverse, since $(S-\lambda I)u = 0$ but $u \neq 0$, and so its determinant must be zero. If we treat λ as a variable and compute the *characteristic polynomial* of S,

$$P(\lambda) = \det(S - \lambda I),$$

we obtain a polynomial of degree N in λ. Its roots $\lambda_1, ..., \lambda_N$ are then the eigenvalues of S. If $\|u\|_2^2 = u^\dagger u = 1$ then $u^\dagger S u = \lambda u^\dagger u = \lambda$. Note that the eigenvalues need not be real, even if S is a real matrix.

We know that a square matrix S is invertible if and only if $Sx = 0$ implies that $x = 0$. We can say this another way now: S is invertible if and only if $\lambda = 0$ is not an eigenvalue of S.

Ex. 3.39 *Compute the eigenvalues for the real square matrix*

$$S = \begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}. \tag{3.35}$$

Note that the eigenvalues are complex, even though the entries of S are real.

The eigenvalues of the Hermitian matrix

$$H = \begin{bmatrix} 1 & 2+i \\ 2-i & 1 \end{bmatrix} \tag{3.36}$$

are $\lambda = 1 + \sqrt{5}$ and $\lambda = 1 - \sqrt{5}$, with corresponding eigenvectors $u = (\sqrt{5}, 2-i)^T$ and $v = (\sqrt{5}, i-2)^T$, respectively. Then $\tilde{H}$, defined as in Equation (3.32), has the same eigenvalues, but both with multiplicity two. Finally, the associated eigenvectors of $\tilde{B}$ are

$$\begin{bmatrix} u^1 \\ u^2 \end{bmatrix}, \tag{3.37}$$

and

$$\begin{bmatrix} -u^2 \\ u^1 \end{bmatrix}, \tag{3.38}$$

for $\lambda = 1 + \sqrt{5}$, and

$$\begin{bmatrix} v^1 \\ v^2 \end{bmatrix}, \tag{3.39}$$

and

$$\begin{bmatrix} -v^2 \\ v^1 \end{bmatrix}, \tag{3.40}$$

for $\lambda = 1 - \sqrt{5}$.

Definition 3.21 *The* spectral radius *of S, denoted $\rho(S)$, is the largest value of $|\lambda|$, where λ denotes any eigenvalue of S.*

Ex. 3.40 *Use the facts that λ is an eigenvalue of S if and only if $det(S - \lambda I) = 0$, and $det(AB) = det(A)det(B)$ to show that λ^2 is an eigenvalue of S^2 if and only if either λ or $-\lambda$ is an eigenvalue of S. Then use this result to show that $\rho(S)^2 = \rho(S^2)$.*

Ex. 3.41 *([79]) We know that the products AB and BA, even when they are the same size, need not be equal. They do have some things in common, though. Let A be M by N and B be N by M, with $M \leq N$. Show that every eigenvalue of A is also an eigenvalue of B and that B has $N - M$ additional eigenvalues that are equal to zero.*

3.12 Sylvester's Nullity Theorem

Recall that the nullity of a matrix A is $n(A)$, the dimension of its null space. The following is taken from [79].

Theorem 3.2 Sylvester's Nullity Theorem *Let A and B be M by N and N by J matrices, respectively. Then*

- *1. $n(AB) \leq n(A) + n(B)$;*
- *2. $n(AB) \geq n(A)$;*
- *3. $n(AB) \geq n(B)$, provided that $M \geq N$.*

Proof: Let R be $r(A)$, the rank of A. Select invertible matrices P and Q so that $PAQ = A^*$ has the entries of the R by R identity matrix in the upper left corner and zeros everywhere else. Set $B^* = Q^{-1}B$. Then A^*, B^*, and $A^*B^* = PAB$ are equivalent to, so have the same ranks and nullities as, A, B and AB, respectively.

The first R rows of A^*B^* are those of B^*, and the remaining $M - R$ ones are zero. The matrix B^* has $r(B^*) = r(B)$ linearly independent rows, of which at most $N - R$ do not appear in A^*B^*. Therefore, there must be at least $r(B) - (N - R) = r(A) + r(B) - N$ linearly independent rows in A^*B^*, and so $r(A^*B^*) \geq r(A) + r(B) - N$.

We know that $r(A) = N - n(A)$, $r(B) = J - n(B)$, and

$$r(AB) = r(A^*B^*) = J - n(A^*B^*) = J - n(AB).$$

Therefore,

$$J - n(AB) \geq N - n(A) + J - n(B) - N,$$

so that $n(AB) \leq n(A) + n(B)$.

The null space of A is a subspace of the null space of AB, so that $n(A) \leq n(AB)$.

Since $r(AB) \leq r(B)$, we have $n(B) \leq M - r(B) \leq n(AB)$, provided that $N \leq M$. ∎

Chapter 4

The ART, MART and EMART

4.1 Chapter Summary

The ART and the MART are two iterative algorithms that were designed to address issues that arose in solving large-scale systems of linear equations for medical imaging [153]. The EMART is a more recently discovered method that combines useful features of both ART and MART [54]. In this chapter we give an overview of these methods; later, we shall revisit them in more detail.

4.2 Overview

In many applications, such as in image processing, we need to solve a system of linear equations that is quite large, often several tens of thousands

of equations in about the same number of unknowns. In these cases, issues such as the costs of storage and retrieval of matrix entries, the computation involved in apparently trivial operations, such as matrix-vector products, and the speed of convergence of iterative methods demand greater attention. At the same time, the systems to be solved are often underdetermined, and solutions satisfying certain additional constraints, such as nonnegativity, are required.

Both the *algebraic reconstruction technique* (ART) and the *multiplicative algebraic reconstruction technique* (MART) were introduced as two iterative methods for discrete image reconstruction in transmission tomography.

Both methods are what are called *row-action* methods, meaning that each step of the iteration uses only a single equation from the system. The MART is limited to nonnegative systems for which nonnegative solutions are sought. In the under-determined case, both algorithms find the solution closest to the starting vector, in the two-norm or weighted two-norm sense for ART, and in the cross-entropy sense for MART, so both algorithms can be viewed as solving optimization problems. For both algorithms, the starting vector can be chosen to incorporate prior information about the desired solution. In addition,the ART can be employed in several ways to obtain a least-squares solution, in the over-determined case.

The *simultaneous* MART (SMART) algorithm is a simultaneous variant of the MART in which all the equations are employed at each step of the iteration. Closely related to the SMART is the *expectation maximization maximum likelihood* (EMML) method, which is also a simultaneous algorithm.

The EMART is a row-action variant of the EMML algorithm. Like MART, it applies to nonnegative systems of equations and produces nonnegative solutions, but, like ART, does not require exponentiation, so is computationally simpler than MART.

4.3 The ART in Tomography

In x-ray transmission tomography, as an x-ray beam passes through the body, it encounters various types of matter, such as soft tissue, bone, ligaments, air, each weakening the beam to a greater or lesser extent. If the intensity of the beam upon entry is I_{in} and I_{out} is its lower intensity after passing through the body, then, at least approximately,

$$I_{out} = I_{in} e^{-\int_L f},$$

where $f = f(x, y) \geq 0$ is the *attenuation function* describing the two-dimensional distribution of matter within the slice of the body being scanned and $\int_L f$ is the integral of the function f over the line L along which the x-ray beam has passed. This is the continuous model. In the discrete model the slice of the body being scanned is viewed as consisting of pixels, which we number $j = 1, 2, ..., J$. The x-rays are sent into the body along I lines, which we number $i = 1, 2, ..., I$. The line integral of f along the ith line is measured, approximately, from the entering and exiting strengths of the x-ray beams; these measurements are denoted b_i.

For $i = 1, ..., I$, let L_i be the set of pixel indices j for which the j-th pixel intersects the i-th line segment, as shown in Figure 4.1, and let $|L_i|$ be the cardinality of the set L_i. Let $A_{ij} = 1$ for j in L_i, and $A_{ij} = 0$ otherwise. With $i = k(\mathrm{mod}\, I) + 1$, the iterative step of the ART algorithm is

$$x_j^{k+1} = x_j^k + \frac{1}{|L_i|}(b_i - (Ax^k)_i), \tag{4.1}$$

for j in L_i, and

$$x_j^{k+1} = x_j^k, \tag{4.2}$$

if j is not in L_i. In each step of ART, we take the error, $b_i - (Ax^k)_i$, associated with the current x^k and the i-th equation, and distribute it equally over each of the pixels that intersects L_i.

This model is too simple; we are assuming that, if the line segment intersects a pixel, then the entire amount of attenuating material within that pixel affects the x-ray strength. A somewhat more sophisticated version of ART allows A_{ij} to include the length of the i-th line segment that lies within the j-th pixel; A_{ij} is taken to be the ratio of this length to the length of the diagonal of the j-pixel.

More generally, ART can be viewed as an iterative method for solving an arbitrary system of linear equations, $Ax = b$.

4.4 The ART in the General Case

Let A be a matrix with complex entries, having I rows and J columns, and let b be a member of $\mathbb{C}^I$. We want to solve the system $Ax = b$. Note that when we say that A is a complex matrix and b a complex vector, we do not exclude the case in which the entries of both A and b are real.

Ex. 4.1 *Find the point in* $\mathbb{R}^2$ *on the line* $y = -3x + 6$ *closest to the point* $(4, 2)$.

Ex. 4.2 *Find the point in* $\mathbb{R}^3$ *on the plane* $x + 2y - 3z = 12$ *closest to the point* $(1, 1, 1)$.

Associated with each equation $(Ax)_i = b_i$ in the system $Ax = b$ there is a hyperplane H_i defined to be the subset of $\mathbb{C}^J$ given by

$$H_i = \{x | (Ax)_i = b_i\}. \tag{4.3}$$

Ex. 4.3 *Show that the* i*th column of* $A^\dagger$ *is normal to the hyperplane* H_i*; that is, it is orthogonal to every vector lying in* H_i.

Ex. 4.4 *Show that, for any vector* z *in* $\mathbb{C}^J$*, the member of* H_i *closest to* z *is* x *having the entries*

$$x_j = z_j + \alpha_i^{-1}\overline{A_{ij}}(b_i - (Az)_i), \tag{4.4}$$

where

$$\alpha_i = \sum_{j=1}^{J} |A_{ij}|^2.$$

Definition 4.1 *The* orthogonal projection operator *onto the hyperplane* H_i *is the function* $P_i : \mathbb{C}^J \to \mathbb{C}^J$ *defined for each* z *in* $\mathbb{C}^J$ *by* $P_i z = x$*, where* x *is the member of* H_i *closest to* z.

The ART algorithm can be expressed in terms of the operators P_i. Let x^0 be arbitrary and, for each nonnegative integer k, let $i(k) = k(\mathrm{mod}\, I) + 1$. The iterative step of the ART is

$$x^{k+1} = P_{i(k)} x^k. \tag{4.5}$$

Using the formula in Equation (4.4), we can write the iterative step of the ART explicitly.

Algorithm 4.1 (ART) *For* $k = 0, 1, ...$ *and* $i = i(k) = k(\mathrm{mod}\, I) + 1$*, the entries of* x^{k+1} *are*

$$x_j^{k+1} = x_j^k + \alpha_i^{-1}\overline{A_{ij}}(b_i - (Ax^k)_i). \tag{4.6}$$

Because the ART uses only a single equation at each step, it has been called a *row-action* method.

4.4.1 Simplifying the Notation

To simplify our notation, we shall assume, throughout this chapter, that the rows of A have been rescaled to have Euclidean length one; that is

$$\alpha_i = \sum_{j=1}^{J} |A_{ij}|^2 = 1, \tag{4.7}$$

for each $i = 1, ..., I$, and that the entries of b have been rescaled accordingly, to preserve the equations $Ax = b$. The ART is then the following: begin with an arbitrary vector x^0; for each nonnegative integer k, having found x^k, the next iterate x^{k+1} has entries

$$x_j^{k+1} = x_j^k + \overline{A_{ij}}(b_i - (Ax^k)_i). \tag{4.8}$$

4.4.2 Consistency

When we are dealing with a general system of linear equations $Ax = b$, we shall say that the system is *consistent* if there are vectors x with $Ax = b$; that is, the system has exact solutions. If not, the system will be called *inconsistent*.

When we are dealing with nonnegative systems $Ax = b$, in which the entries of A are nonnegative, the entries of b are positive, and we seek a nonnegative solution x, we shall say that such a system is consistent if there are nonnegative vectors x with $Ax = b$; otherwise, the system is inconsistent. It will always be clear from the context which category of systems we are discussing. The ART applies to general systems of linear equations, while the MART and EMART apply only to nonnegative systems. Note that a nonnegative system can be inconsistent even when it possesses exact solutions x that happen not to be nonnegative.

4.4.3 When $Ax = b$ Has Solutions

For the consistent case we have the following result concerning the ART.

Theorem 4.1 *Let $A\hat{x} = b$ and let x^0 be arbitrary. Let $\{x^k\}$ be generated by Equation (4.8). Then the sequence of Euclidean distances or two-norms $\{||\hat{x} - x^k||_2\}$ is decreasing and $\{x^k\}$ converges to the solution of $Ax = b$ closest to x^0.*

A proof of Theorem 4.1 is provided in Chapter 14.

So, when the system $Ax = b$ has exact solutions, the ART converges to the solution closest to x^0, in the 2-norm. How fast the algorithm converges will depend on the ordering of the equations and on whether or not we use *relaxation*, which we shall discuss later. In selecting the equation ordering, the important thing is to avoid particularly bad orderings, in which the hyperplanes H_i and H_{i+1} are nearly parallel.

4.4.4 When $Ax = b$ Has No Solutions

When there are no exact solutions, the ART does not converge to a single vector, but, for each fixed i, the subsequence $\{x^{nI+i}, n = 0, 1, ...\}$

converges to a vector z^i and the collection $\{z^i \,|i = 1, ..., I\}$ is called the *limit cycle*. The ART limit cycle will vary with the ordering of the equations, and contains more than one vector unless an exact solution exists; see [55] for details.

Figures 4.2 and 4.3 illustrate the behavior of the ART in the two cases.

4.4.5 The Geometric Least-Squares Solution

When the system $Ax = b$ has no solutions, it is reasonable to seek an approximate solution, such as a *least squares* solution, which minimizes $||Ax - b||_2$. It is important to note that the system $Ax = b$ has solutions if and only if the related system $WAx = Wb$ has solutions, where W denotes an invertible matrix; when solutions of $Ax = b$ exist, they are identical to those of $WAx = Wb$. But, when $Ax = b$ does not have solutions, the least-squares solutions of $Ax = b$, which need not be unique, but usually are, and the least-squares solutions of $WAx = Wb$ need not be identical. In the typical case in which $A^\dagger A$ is invertible, the unique least-squares solution of $Ax = b$ is

$$(A^\dagger A)^{-1} A^\dagger b, \tag{4.9}$$

while the unique least-squares solution of $WAx = Wb$ is

$$(A^\dagger W^\dagger W A)^{-1} A^\dagger W^\dagger b, \tag{4.10}$$

and these need not be the same.

A simple example is the following. Consider the system

$$x = 1$$

$$x = 2, \tag{4.11}$$

which has the unique least-squares solution $x = 1.5$, and the system

$$2x = 2$$

$$x = 2, \tag{4.12}$$

which has the least-squares solution $x = 1.2$.

Definition 4.2 *A* geometric least-squares *solution of $Ax = b$ is a least-squares solution of $WAx = Wb$, for W the diagonal matrix whose entries are the reciprocals of the Euclidean lengths of the rows of A.*

In our example above, the geometric least-squares solution for the first system is found by using $W_{11} = 1 = W_{22}$, so is again $x = 1.5$, while the

geometric least-squares solution of the second system is found by using $W_{11} = 0.5$ and $W_{22} = 1$, so that the geometric least-squares solution is $x = 1.5$, not $x = 1.2$.

Since we are assuming that the rows of A have been rescaled to have Euclidean length one, any least-squares solution for A is now a geometric least-squares solution.

Open Question: If there is a unique geometric least-squares solution, where is it, in relation to the vectors of the limit cycle? Can it be calculated easily, from the vectors of the limit cycle?

There is a partial answer to the first question. It is known that, if the system $Ax = b$ has no exact solution, A has full rank, and $I = J + 1$, then the vectors of the limit cycle lie on a sphere in J-dimensional space having the geometric least-squares solution at its center [55]. This is not generally true for $I \neq J + 1$, however.

4.5 The MART

The *multiplicative* ART (MART) is an iterative algorithm closely related to the ART. It also was devised to obtain tomographic images, but, like ART, applies more generally; MART applies to nonnegative systems of linear equations $Ax = b$ for which the b_i are positive, the A_{ij} are nonnegative, and the solution x we seek is to have nonnegative entries. It is not so easy to see the relation between ART and MART if we look at the most general formulation of MART. For that reason, we begin with a simpler case, transmission tomographic imaging, in which the relation is most clearly apparent.

4.5.1 A Special Case of MART

We begin by considering the application of MART to the transmission tomography problem. For $i = 1, ..., I$, let L_i be the set of pixel indices j for which the j-th pixel intersects the i-th line segment, and let $|L_i|$ be the cardinality of the set L_i. Let $A_{ij} = 1$ for j in L_i, and $A_{ij} = 0$ otherwise. With $i = k(\mathrm{mod}\, I) + 1$, the iterative step of the ART algorithm is

$$x_j^{k+1} = x_j^k + \frac{1}{|L_i|}(b_i - (Ax^k)_i), \tag{4.13}$$

for j in L_i, and

$$x_j^{k+1} = x_j^k, \tag{4.14}$$

if j is not in L_i. In each step of ART, we take the error, $b_i - (Ax^k)_i$, associated with the current x^k and the i-th equation, and distribute it equally over each of the pixels that intersects L_i.

Suppose, now, that each b_i is positive, and we know in advance that the desired image we wish to reconstruct must be nonnegative. We can begin with $x^0 > 0$, but as we compute the ART steps, we may lose nonnegativity. One way to avoid this loss is to correct the current x^k multiplicatively, rather than additively, as in ART. This leads to the *multiplicative* ART (MART).

The MART, in this case, has the iterative step

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i} \Big), \tag{4.15}$$

for those j in L_i, and

$$x_j^{k+1} = x_j^k, \tag{4.16}$$

otherwise. Therefore, we can write the iterative step as

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i} \Big)^{A_{ij}}. \tag{4.17}$$

4.5.2 The MART in the General Case

Taking the entries of the matrix A to be either one or zero, depending on whether or not the j-th pixel is in the set L_i, is too crude. The line L_i may just clip a corner of one pixel, but pass through the center of another. Surely, it makes more sense to let A_{ij} be the length of the intersection of line L_i with the j-th pixel, or, perhaps, this length divided by the length of the diagonal of the pixel. It may also be more realistic to consider a strip, instead of a line. Other modifications to A_{ij} may be made, in order to better describe the physics of the situation. Finally, all we can be sure of is that A_{ij} will be nonnegative, for each i and j. In such cases, what is the proper form for the MART?

The MART, which can be applied only to nonnegative systems, is a sequential, or row-action, method that uses one equation only at each step of the iteration.

Algorithm 4.2 (MART) *Let x^0 be a positive vector. For $k = 0, 1, ...$, and $i = k(\mathrm{mod}\, I) + 1$, having found x^k define x^{k+1} by*

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i} \Big)^{m_i^{-1} A_{ij}}, \tag{4.18}$$

where $m_i = \max \{A_{ij} \,|j = 1, 2, ..., J\}$.

Some treatments of MART leave out the m_i, but require only that the entries of A have been rescaled so that $A_{ij} \leq 1$ for all i and j. The m_i is important, however, in accelerating the convergence of MART.

Notice that we can write x_j^{k+1} as a weighted geometric mean of x_j^k and $x_j^k\left(\frac{b_i}{(Ax^k)_i}\right)$:

$$x_j^{k+1} = \left(x_j^k\right)^{1-m_i^{-1}A_{ij}}\left(x_j^k\left(\frac{b_i}{(Ax^k)_i}\right)\right)^{m_i^{-1}A_{ij}}. \tag{4.19}$$

This will help to motivate the EMART.

4.5.3 Cross-Entropy

For $a > 0$ and $b > 0$, let the cross-entropy or Kullback-Leibler (KL) distance from a to b be

$$KL(a, b) = a \log \frac{a}{b} + b - a, \tag{4.20}$$

with $KL(a, 0) = +\infty$, and $KL(0, b) = b$. Extend to nonnegative vectors coordinate-wise, so that

$$KL(x, z) = \sum_{j=1}^{J} KL(x_j, z_j). \tag{4.21}$$

Unlike the Euclidean distance, the KL distance is not symmetric; $KL(Ax, b)$ and $KL(b, Ax)$ are distinct, and we can obtain different approximate solutions of $Ax = b$ by minimizing these two distances with respect to nonnegative x.

4.5.4 Convergence of MART

In the consistent case, by which we mean that $Ax = b$ has nonnegative solutions, we have the following convergence theorem for MART.

Theorem 4.2 *In the consistent case, the MART converges to the unique nonnegative solution of $b = Ax$ for which the distance $\sum_{j=1}^{J} KL(x_j, x_j^0)$ is minimized.*

If the starting vector x^0 is the vector whose entries are all equal to one, then the MART converges to the solution that maximizes the *Shannon entropy*,

$$SE(x) = \sum_{j=1}^{J} x_j \log x_j - x_j. \tag{4.22}$$

As with ART, the speed of convergence is greatly affected by the ordering of the equations, converging most slowly when consecutive equations correspond to nearly parallel hyperplanes.

Open Question: When there are no nonnegative solutions, MART does not converge to a single vector, but, like ART, is always observed to produce a limit cycle of vectors. Unlike ART, there is no proof of the existence of a limit cycle for MART. Is there such a proof?

4.6 The EMART

The MART enforces positivity of the x_j^k, but at the cost of an exponentiation in each step. The EMART is similar to the MART, guarantees positivity at each step, but does not employ exponentiation.

The EMART is a row-action version of the *expectation maximization maximum likelihood* (EMML) algorithm (see [51, 53]). The EMML algorithm, which was developed as a method for reconstructing tomographic medical images, was found to converge too slowly to be of practical use. Several faster variants of the EMML algorithm were subsequently discovered, one of which is the EMART.

As with MART, we assume that the entries of the matrix A are nonnegative, that the entries of b are positive, and that we seek a nonnegative solution of $Ax = b$.

Algorithm 4.3 (EMART) *Let x^0 be an arbitrary positive vector and $i = k(\mathrm{mod}\, I) + 1$. Then let*

$$x_j^{k+1} = (1 - m_i^{-1} A_{ij}) x_j^k + m_i^{-1} A_{ij} \Big(x_j^k \frac{b_i}{(Ax^k)_i} \Big). \tag{4.23}$$

Notice that x_j^{k+1} is always positive, since it is a weighted arithmetic mean of x_j^k and $x_j^k \Big(\frac{b_i}{(Ax^k)_i} \Big)$.

In the consistent case, in which there are nonnegative solutions of $Ax = b$, the EMART converges to a nonnegative solution. However, no characterization of the solution, in terms of x^0, is known.

The EMART fails to converge in the inconsistent case. What is always observed, but for which no proof is known, is that, for each fixed $i = 1, 2, ..., I$, as $m \to +\infty$, the EMART subsequences $\{x^{mI+i}\}$ converge to separate limit vectors, say $x^{\infty,i}$. For details concerning the MART and EMART see [54] and [56].

Open Questions: We know that, in the consistent case, the MART converges to the nonnegative solution of $Ax = b$ for which $KL(x, x^0)$ is minimized. Is there a similar characterization of the EMART solution, in terms of x^0? When there are no nonnegative solutions, EMART does not converge to a single vector, but, like ART and MART, is always observed to produce a limit cycle of vectors. Unlike ART, no one has found a proof of the existence of a limit cycle for EMART. Is there such a proof?

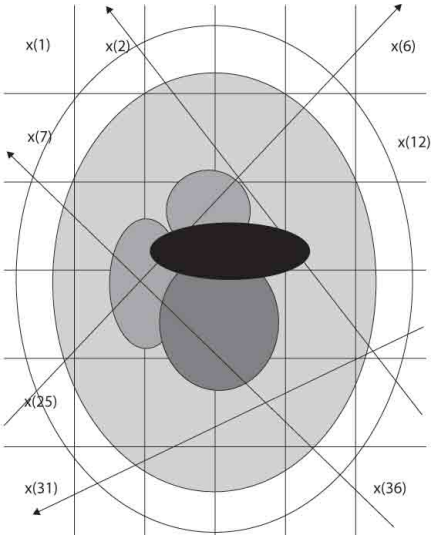

FIGURE 4.1: Line segments through a discretized object.

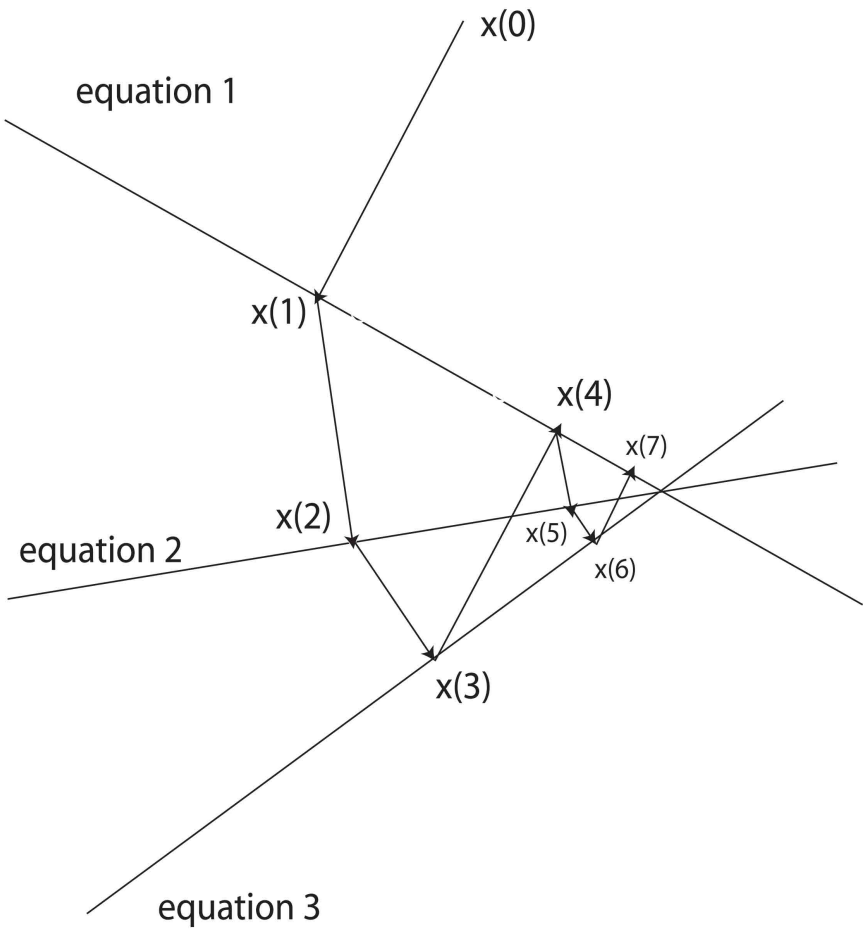

FIGURE 4.2: The ART algorithm in the consistent case.

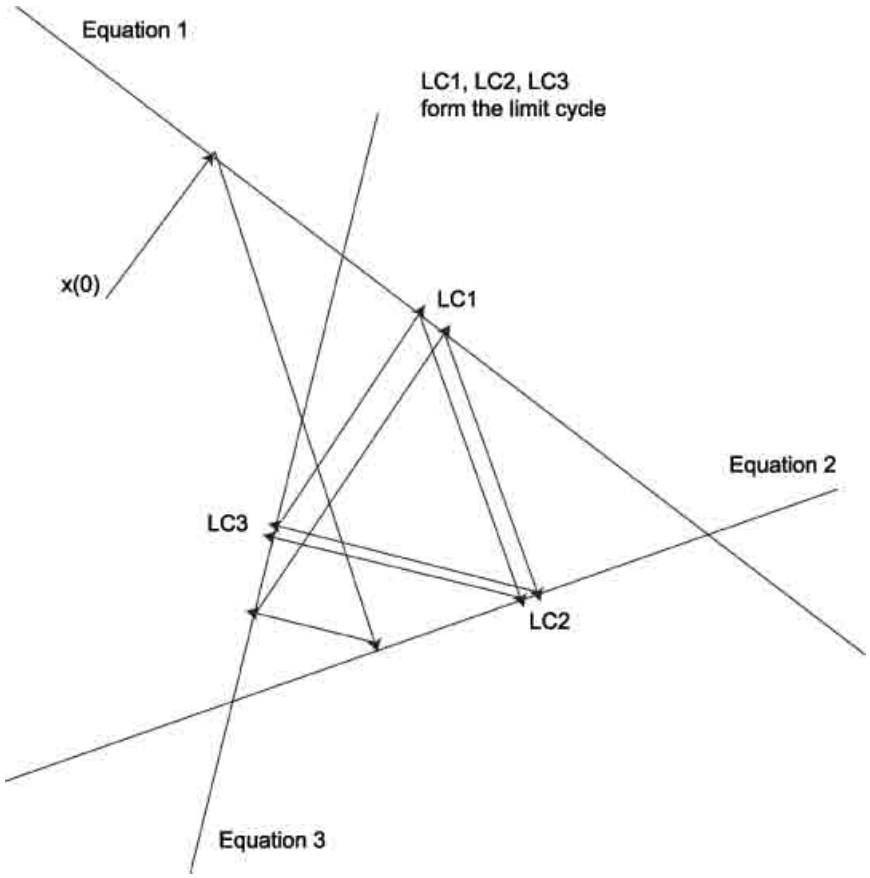

FIGURE 4.3: The ART algorithm in the inconsistent case.

Part II

Algebra

Chapter 5

Matrix Factorization and Decomposition

5.1 Chapter Summary

In this chapter we continue our study of matrix algebra. The emphasis now is on matrix factorization, eigenvector decomposition, and several forms of data compression.

5.2 Orthogonal and Unitary Matrices

The orthogonal and unitary matrices play important roles in matrix factorization and decomposition.

Definition 5.1 *A complex square matrix U is said to be* unitary *if $U^\dagger U = I$. A real square matrix O is* orthogonal *if $O^T O = I$. A square matrix T is* upper triangular *if all the entries of T below the main diagonal are zero.*

For example, the matrix

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

is orthogonal.

Ex. 5.1 *Consider the transformation of points in $\mathbb{R}^2$ that takes the point (x, y) into the point (x_1, y_1) by rotating the vector (x, y) around the origin counter-clockwise through an angle θ. Find the matrix A with the property that $A(x, y)^T = (x_1, y_1)^T$.*

Ex. 5.2 *In this exercise we extend the previous exercise to three-dimensional space. Let p be the nonzero vector connecting the origin with the point p in $\mathbb{R}^3$. Consider the transformation that takes a point (x, y, z) to (x_1, y_1, z_1) by rotating through an angle θ around the axis determined by the origin and the point p. Find the matrix A with the property that $A(x, y, z)^T = (x_1, y_1, z_1)^T$.*

5.3 Proof By Induction

Proof by induction is a tool used in a wide variety of proofs; we shall use it shortly to prove Schur's Lemma. In this section we present the basic idea and an example to illustrate its use.

All proofs by induction have the same basic form. There is some property, say Property P, that a positive integer n may or may not have. The assertion, which we must prove, is that all n have Property P. The proof is by contradiction; we assume the assertion is false and that not all n have Property P. Therefore, there must be a first n that does not have Property P. We begin by checking to see if $n = 1$ has Property P. Having established that $n = 1$ has Property P, we focus on the first n that does not have Property P; we know that this n is not one, so $n - 1$ is also a positive integer, and $n - 1$ does have Property P, since n is the first one without Property P. The rest of the proof involves showing that, because $n - 1$ has Property P, so must n. This will give us our contradiction and allow us to conclude that there is no such first n without Property P.

For example, let Property P be the following: n is a positive integer such that the sum of the first n positive integers is $\frac{1}{2}n(n+1)$. This clearly holds for $n = 1$, so $n = 1$ has Property P. Assume that not all n do have Property P, and let n be the first that does not have Property P. Then $n - 1$ is a positive integer and

$$1 + 2 + ... + n - 1 = \frac{1}{2}(n-1)n.$$

Then

$$1+2+...+n = 1+2+...+n-1+n = \frac{1}{2}(n-1)n+n = \frac{1}{2}n(n-1+2) = \frac{1}{2}n(n+1).$$

Therefore, n must also have Property P. This contradicts our assumption that not all positive integers have Property P. Therefore, Property P holds for all positive integers.

Note that there are other ways to prove this theorem. We have used induction here because we are trying to illustrate the use of induction. In most cases in which induction is used, induction is the best, and maybe the only, way to prove the theorem.

Ex. 5.3 *Prove that*

$$\frac{1}{2!} + \frac{2}{3!} + ... + \frac{n}{(n+1)!} = 1 - \frac{1}{(n+1)!}.$$

5.4 Schur's Lemma

Schur's Lemma is a useful tool for proving the diagonalization theorems for Hermitian and normal matrices.

Theorem 5.1 (Schur's Lemma) *For any square matrix S there is a unitary matrix U such that $U^\dagger SU = T$ is an upper triangular matrix.*

Proof: We proceed by induction. The theorem is obviously true for any 1 by 1 matrix. Assume that the theorem is true for any $n-1$ by $n-1$ matrix. We show that it is true also for any n by n matrix.

Because every polynomial has at least one (possibly complex) root, S has at least one eigenvector. Therefore, let $Su^1 = \lambda u^1$, with $\|u^1\|_2 = 1$. Let $\{u^1, u^2, ..., u^n\}$ be an orthonormal basis for $\mathbb{C}^n$. Then

$$U = \begin{bmatrix} u^1 & u^2 & ... & u^n \end{bmatrix} \tag{5.1}$$

is unitary and

$$U^\dagger SU = \begin{bmatrix} (u^1)^\dagger \\ (u^2)^\dagger \\ . \\ . \\ . \\ (u^n)^\dagger \end{bmatrix} \begin{bmatrix} Su^1 & Su^2 & ... & Su^n \end{bmatrix} = \begin{bmatrix} \lambda_1 & c_{12} & ... & c_{1n} \\ 0 & & & \\ 0 & & & \\ . & & S_1 & \\ . & & & \\ . & & & \\ 0 & & & \end{bmatrix},$$

where S_1 is of order $n-1$.

Now let U_1 be an $n-1$ by $n-1$ unitary matrix such that $U_1^\dagger S_1 U_1$ is upper triangular; such a U_1 exists by the induction hypothesis. Let

$$U_2 = \begin{bmatrix} 1 & 0 & ... & 0 \\ 0 & & & \\ 0 & & & \\ . & & U_1 & \\ . & & & \\ . & & & \\ 0 & & & \end{bmatrix}.$$

Then U_2 and UU_2 are unitary and

$$(UU_2)^\dagger S(UU_2) = U_2^\dagger (U^\dagger SU) U_2$$

$$= \begin{bmatrix} \lambda_1 & b_{12} & ... & b_{1n} \\ 0 & & & \\ 0 & & & \\ . & & U_1^\dagger S U_1 & \\ . & & & \\ . & & & \\ 0 & & & \end{bmatrix},$$

which is upper triangular. ∎

Using essentially the same proof, we can establish the following version of Schur's Lemma:

Theorem 5.2 *Let S be a real square matrix with only real eigenvalues. Then there is a real orthogonal matrix O such that $O^T SO$ is upper triangular.*

Corollary 5.1

- *(a) If $S^\dagger = S$ then there is a unitary matrix U such that $U^\dagger SU$ is a real diagonal matrix.*
- *(b) If S is real and $S^T = S$ then there is an orthogonal matrix O such that $O^T SO$ is a real diagonal matrix.*

Ex. 5.4 *Use Schur's Lemma to prove Corollary 5.1.*

Theorem 5.3 *For a given complex square matrix S there is a unitary matrix U such that $U^\dagger SU = D$ is a diagonal matrix if and only if S is normal.*

Ex. 5.5 *Use Schur's Lemma to prove Theorem 5.3. Hint: Show that T is also normal and then compute $(TT^\dagger)_{nn}$ two ways.*

Lemma 5.1 *Let T be upper triangular. Then* $\det(T)$ *is the product of the entries on its main diagonal.*

Proof: Assume first that the entries of the main diagonal are all one. Then T can be row reduced to I using only row operations that do not change the determinant. Then, for more general T, the diagonal elements can be changed to ones by dividing by their current values, provided that none is zero. If T has a zero on its main diagonal, then either that row is a zero row, or can be row reduced to a zero row by row operations that do not alter the determinant. ∎

Using Schur's Lemma we obtain a different proof of Proposition 3.8, which we restate now.

Proposition 5.1 *A square matrix A is invertible if and only if its determinant is not zero.*

Proof: From Schur's Lemma we know that there is a unitary matrix U such that $U^\dagger AU = T$ is upper triangular. From Lemma 5.1, the determinant of T is the product of the entries on its main diagonal. Clearly, T is invertible if and only if none of these entries is zero; this is true because $Tx = 0$ implies $x = 0$ if and only if no diagonal entry is zero. Therefore, T is invertible if and only if the determinant of T is not zero. But, the determinant of A is the same as that of T and A is invertible precisely when T is invertible. ∎

Ex. 5.6 *Prove that the eigenvalues of an upper triangular matrix T are the entries of its main diagonal, so that the trace of T is the sum of its eigenvalues.*

Ex. 5.7 *Prove that, if S is square, U is unitary, and $U^\dagger SU = T$ is upper triangular, then the eigenvalues of S and T are the same and S and T have the same trace. Hint: use the facts that* $\det(AB) = \det(A)\det(B)$ *and Equation (3.23).*

Ex. 5.8 *Use the two previous exercises to prove that, for any square matrix S, the trace of S is the sum of its eigenvalues.*

5.5 The Hermitian Case

Let H be an N by N Hermitian matrix. As we just saw, there is a unitary matrix U such that $U^\dagger HU = D$ is real and diagonal. Then $HU = UD$, so that the columns of U are eigenvectors of H with two-norms equal to one, and the diagonal entries of D are the eigenvalues of H. Since U is invertible, its columns form a set of N mutually orthogonal norm-one eigenvectors of the Hermitian matrix H; call them $\{u^1, ..., u^N\}$. We denote by λ_n, $n = 1, 2, ..., N$, the N eigenvalues, so that $Hu^n = \lambda_n u^n$. This is the well known *eigenvalue-eigenvector decomposition* of the matrix H. Not every square matrix has such a decomposition, which is why we focus on Hermitian H. The singular-value decomposition, which we discuss shortly, provides a similar decomposition for an arbitrary, possibly non-square, matrix.

The matrix H can also be written as

$$H = \sum_{n=1}^{N} \lambda_n u^n (u^n)^\dagger,$$

a linear superposition of the *dyad* matrices $u^n(u^n)^\dagger$. The Hermitian matrix H is invertible if and only if none of the λ are zero and its inverse is

$$H^{-1} = \sum_{n=1}^{N} \lambda_n^{-1} u^n (u^n)^\dagger.$$

We also have $H^{-1} = UL^{-1}U^\dagger$.

Ex. 5.9 *Show that if $z = (z_1, ..., z_N)^T$ is a column vector with complex entries and $H = H^\dagger$ is an N by N Hermitian matrix with complex entries*

then the quadratic form $z^\dagger Hz$ is a real number. Show that the quadratic form $z^\dagger Hz$ can be calculated using only real numbers. Let $z = x + iy$, with x and y real vectors and let $H = A + iB$, where A and B are real matrices. Then show that $A^T = A$, $B^T = -B$, $x^T Bx = 0$ and finally,

$$z^\dagger Hz = \begin{bmatrix} x^T & y^T \end{bmatrix} \begin{bmatrix} A & -B \\ B & A \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

Use the fact that $z^\dagger Hz$ is real for every vector z to conclude that the eigenvalues of H are real.

Ex. 5.10 *Show that the eigenvalues of a Hermitian matrix H are real by computing the conjugate transpose of the* 1 *by* 1 *matrix $z^\dagger Hz$.*

Definition 5.2 *A Hermitian matrix Q is said to be* nonnegative-definite *if all the eigenvalues of Q are nonnegative, and* positive-definite *if all the eigenvalues are positive.*

Proposition 5.2 *A Hermitian matrix Q is a nonnegative-definite matrix if and only if there is another matrix C, not necessarily square, such that $Q = C^\dagger C$.*

Proof: Assume that Q is nonnegative-definite and let $Q = ULU^\dagger$ be the eigenvalue/eigenvector decomposition of Q. Since the eigenvalues of Q are nonnegative, each diagonal entry of the matrix L has a nonnegative square root; the matrix with these square roots as entries is called $\sqrt{L}$. Using the fact that $U^\dagger U = I$, we have

$$Q = ULU^\dagger = U\sqrt{L}U^\dagger U\sqrt{L}U^\dagger;$$

we then take $C = U\sqrt{L}U^\dagger$, so $C^\dagger = C$. This choice of C is called the *Hermitian square root* of Q.

Conversely, assume now that $Q = C^\dagger C$, for some arbitrary, possibly not square, matrix C. Let $Qu = \lambda u$, for some non-zero eigenvector u, so that λ is an eigenvalue of Q. Then

$$\lambda \|u\|_2^2 = \lambda u^\dagger u = u^\dagger Qu = u^\dagger C^\dagger Cu = \|Cu\|_2^2,$$

so that

$$\lambda = \|Cu\|_2^2 / \|u\|_2^2 \geq 0.$$

∎

If N is a square complex matrix with $N = UDU^\dagger$, where, as above, $U^\dagger U = I$ and D is diagonal, but not necessarily real, then we do have $N^\dagger N = NN^\dagger$; then N is *normal*, which means that $N^T N = NN^T$. The matrix N will be Hermitian if and only if D is real. It follows then that a

real normal matrix N will be symmetric if and only if its eigenvalues are real, since it is then Hermitian and real.

The normal matrices are precisely those for which such an eigenvector-eigenvalue decomposition holds, as we saw above. In Chapter 17, the appendix on Hermitian and Normal Linear Operators, we prove this result again, as a statement about operators on a finite-dimensional vector space.

The following exercise gives an example of a matrix N that is real, normal, not symmetric, and has non-real eigenvalues. The matrix $N^T N$ has repeated eigenvalues. As we shall see in Theorem 5.4, if a real, normal matrix is such that $N^T N$ does not have repeated eigenvalues, then N is symmetric and so the eigenvalues of N are real.

Ex. 5.11 *Show that the* 2 *by* 2 *matrix* $N = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ *is real, normal, and has eigenvalues* $\pm i$*. Show that the eigenvalues of* $N^T N$ *are both* 1*.*

5.6 Diagonalizable Matrices

For an arbitrary square matrix S, the eigenvectors need not be mutually orthogonal. Nevertheless, we will still be able to get an eigenvector/eigenvalue decomposition of S, if S is a *diagonalizable* matrix.

Definition 5.3 *A* J *by* J *matrix* S *is* diagonalizable *if* $\mathbb{C}^J$ *has a basis of eigenvectors of* S*.*

As the following lemma tells us, most square matrices are diagonalizable.

Lemma 5.2 *A square matrix* S *is diagonalizable if all its eigenvalues are distinct.*

Proof: We need to show that the eigenvectors associated with different eigenvalues are linearly independent. Let S be J by J. Let λ_j be the eigenvalues of S, $Su^j = \lambda_j u^j$, and $u^j \neq 0$, for $j = 1, ..., J$. Let u^m be the first eigenvector that is in the span of $\{u_j | j = 1, ..., m-1\}$. Then

$$u^m = a_1 u^1 + ... + a_{m-1} u^{m-1}, \tag{5.2}$$

for some constants a_j that are not all zero. Multiply both sides by λ_m to get

$$\lambda_m u^m = a_1 \lambda_m u^1 + ... + a_{m-1} \lambda_m u^{m-1}. \tag{5.3}$$

From

$$\lambda_m u^m = Au^m = a_1\lambda_1 u^1 + ... + a_{m-1}\lambda_{m-1}u^{m-1}, \tag{5.4}$$

it follows that

$$a_1(\lambda_m - \lambda_1)u^1 + ... + a_{m-1}(\lambda_m - \lambda_{m-1})u^{m-1} = 0, \tag{5.5}$$

from which we can conclude that some u^n in $\{u^1, ..., u^{m-1}\}$ is in the span of the others. This is a contradiction. ∎

When S is diagonalizable, we let U be a square matrix whose columns are J linearly independent eigenvectors of S and L the diagonal matrix having the eigenvalues of S along its main diagonal; then we have $SU = UL$, or $U^{-1}SU = L$. Note that U^{-1} is not equal to $U^\dagger$, unless the columns of U form an orthonormal set.

5.7 The Singular Value Decomposition (SVD)

The year 1965 was a good one for the discovery of important algorithms. In that year, Cooley and Tukey [103] introduced the *fast Fourier transform* (FFT) algorithm and Golub and Kahan [151] their method for calculating the *singular-value decomposition* (SVD).

We have just seen that an N by N Hermitian matrix H can be written in terms of its eigenvalues and eigenvectors as $H = ULU^\dagger$ or as

$$H = \sum_{n=1}^{N} \lambda_n u^n (u^n)^\dagger.$$

The *singular value decomposition* (SVD) is a similar result that applies to any rectangular matrix A. It is an important tool in image compression and pseudo-inversion.

5.7.1 Defining the SVD

Let A be any M by N complex matrix. In presenting the SVD of A we shall assume that $N \geq M$; the SVD of $A^\dagger$ will come from that of A. Let $Q = A^\dagger A$ and $P = AA^\dagger$; we assume, reasonably, that P, the smaller of the two matrices, is invertible, so all the eigenvalues $\lambda_1, ..., \lambda_M$ of P are positive. We let the eigenvalue/eigenvector decomposition of P be $P = ULU^\dagger$, where $\{u^1, ..., u^M\}$ are orthonormal eigenvectors of P and $Pu^m = \lambda_m u^m$.

From $PU = UL$ or $AA^\dagger U = UL$ it follows that $A^\dagger AA^\dagger U = A^\dagger UL$.

Therefore, the M columns of $W = A^\dagger U$ are eigenvectors of Q corresponding to the eigenvalues λ_m; since $Pu^m = AA^\dagger u^m$ is not the zero vector, $A^\dagger u^m$ cannot be the zero vector either. But the columns of W do not have norm one. To normalize these columns we replace them with the M columns of $A^\dagger U L^{-1/2}$, which are orthonormal eigenvectors of Q.

Ex. 5.12 *Show that the nonzero eigenvalues of $Q = A^\dagger A$ and $P = AA^\dagger$ are the same.*

Let Z be the N by N matrix whose first M columns are those of the matrix $A^\dagger U L^{-1/2}$ and whose remaining $N - M$ columns are any mutually orthogonal norm-one vectors that are all orthogonal to each of the first M columns; note that this gives us $Z^\dagger Z = I$.

Let Σ be the M by N matrix with diagonal entries $\Sigma_{mm} = \sqrt{\lambda_m}$, for $m = 1, ..., M$, and whose remaining entries are zero. The nonzero entries of Σ, the $\sqrt{\lambda_m}$, are called the *singular values* of A. The *singular value decomposition* (SVD) of A is $A = U\Sigma Z^\dagger$. The SVD of $A^\dagger$ is $A^\dagger = Z\Sigma^T U^\dagger$.

Ex. 5.13 *Show that $U\Sigma Z^\dagger$ equals A.*

We have assumed, for convenience, that none of the eigenvalues λ_m, $m = 1, ..., M$ are zero. If this is not true, we can obtain the SVD of A simply by modifying the definition of $L^{-1/2}$ to have $1/\sqrt{\lambda_m}$ on the main diagonal if λ_m is not zero, and zero if it is. To show that $U\Sigma Z^\dagger = A$ now we need to use the fact that $Pu^m = 0$ implies that $A^\dagger u^m = 0$. To see this, note that

$$0 = Pu^m = AA^\dagger u^m$$

implies that

$$0 = (u^m)^\dagger Pu^m = (u^m)^\dagger AA^\dagger u^m = \|A^\dagger u^m\|^2.$$

As an example of the singular-value decomposition, consider the matrix A, whose SVD is given by

$$A = \begin{bmatrix} 4 & 8 & 8 \\ 3 & 6 & 6 \end{bmatrix} = \begin{bmatrix} 4/5 & 3/5 \\ 3/5 & -4/5 \end{bmatrix} \begin{bmatrix} 15 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1/3 & 2/3 & 2/3 \\ 2/3 & -2/3 & 1/3 \\ 2/3 & 1/3 & -2/3 \end{bmatrix},$$

which can also be written in dyad form as

$$A = 15 \begin{bmatrix} 4/5 \\ 3/5 \end{bmatrix} \begin{bmatrix} 1/3 & 2/3 & 2/3 \end{bmatrix}.$$

It is just a coincidence that, in this example, the matrices U and Z are symmetric.

The SVD of A^T is then

$$A^T = \begin{bmatrix} 4 & 3 \\ 8 & 6 \\ 8 & 6 \end{bmatrix} = \begin{bmatrix} 1/3 & 2/3 & 2/3 \\ 2/3 & -2/3 & 1/3 \\ 2/3 & 1/3 & -2/3 \end{bmatrix} \begin{bmatrix} 15 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 4/5 & 3/5 \\ 3/5 & -4/5 \end{bmatrix}.$$

Ex. 5.14 *If H is a Hermitian matrix, its eigenvalue/eigenvector decomposition $H = ULU^\dagger$ need not be its SVD. Illustrate this point for the real symmetric matrix* $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$.

Using the SVD of A we can write A as a sum of dyads:

$$A = \sum_{m=1}^{M} \sqrt{\lambda_m} u^m (z^m)^\dagger, \tag{5.6}$$

where z^m denotes the mth column of the matrix Z.

In image processing, matrices such as A are used to represent discrete two-dimensional images, with the entries of A corresponding to the grey level or color at each pixel. It is common to find that most of the M singular values of A are nearly zero, so that A can be written approximately as a sum of far fewer than M dyads; this leads to SVD image compression. Such compression is helpful when many images are being transmitted, as, for example, when pictures of the surface of Mars are sent back to Earth.

Figures 5.1 and 5.2 illustrate what can be achieved with SVD compression. In both Figures the original is in the upper left. It is a 128 by 128 digitized image, so $M = 128$. In the images that follow, the number of terms retained in the sum in Equation (5.6) is, first, 2, then 4, 6, 8, 10, 20 and finally 30. The full sum has 128 terms, remember. In Figure 5.1 the text is nearly readable using only 10 terms, and certainly could be made perfectly readable with suitable software, so storing just this compressed image would be acceptable. In Figure 5.2, an image of a satellite, we get a fairly good idea of the general shape of the object from the beginning, with only two terms.

Ex. 5.15 *Suppose that $M = N$ and A is invertible. Show that we can write*

$$A^{-1} = \sum_{m=1}^{M} (\sqrt{\lambda_m})^{-1} z^m (u^m)^\dagger.$$

5.7.2 An Application in Space Exploration

The *Galileo* was deployed from the space shuttle *Atlantis* on October 18, 1989. After a detour around Venus and back past Earth to pick up gravity-assisted speed, *Galileo* headed for Jupiter. Its mission included a study of Jupiter's moon Europa, and the plan was to send back one high-resolution photo per minute, at a rate of $134KB$ per second, via a huge high-gain antenna, that is, one capable of transmitting most of its energy in a narrow beam. When the time came to open the antenna, it stuck. Without the pictures, the mission would be a failure.

There was a much smaller *low-gain* antenna on board, one with far less directionality, but the best transmission rate was going to be ten bits per second. All that could be done from earth was to reprogram an old on-board computer to compress the pictures prior to transmission. The problem was that pictures could be taken much faster than they could be transmitted to earth; some way to store them prior to transmission was key. The original designers of the software had long since retired, but the engineers figured out a way to introduce state-of-the art image compression algorithms into the computer. It happened that there was an ancient reel-to-reel storage device on board that was there only to serve as a backup for storing atmospheric data. Using this device and the compression methods, the engineers saved the mission [16].

5.7.3 A Theorem on Real Normal Matrices

Consider the real square matrix

$$S = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}.$$

Since

$$S^T S = S S^T = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix},$$

S is a real normal matrix. The eigenvalues of S are complex, S is not symmetric, and the eigenvalues of $S^T S$ are not distinct. In contrast, we have the following theorem.

Let N be a real square matrix that is normal; that is $N^T N = N N^T$. Now we use the SVD of N to prove the following theorem.

Theorem 5.4 *If N is a real normal matrix and all the eigenvalues of $N^T N$ are distinct, then N is symmetric.*

Proof: Let $Q = N^T N$. Since Q is real, symmetric, and nonnegative definite, there is an orthogonal matrix O such that $QO = NN^T O = OD^2$, with $D \geq 0$ and D^2 the diagonal matrix whose diagonal entries are the

eigenvalues of $Q = N^T N$. We shall want to be able to assume that the entries of D are all positive, which requires a bit of explanation.

We replace the matrix N with the new matrix $N + \alpha I$, where $\alpha > 0$ is selected so that the matrix $(N+\alpha I)(N+\alpha I)^T$ has only positive eigenvalues. We can do this because

$$(N + \alpha I)(N + \alpha I)^T = NN^T + \alpha(N + N^T) + \alpha^2 I;$$

the first and third matrices have only nonnegative eigenvalues and the second one has only real ones, so a large enough α can be found. Now we can prove the theorem for the new matrix $N + \alpha I$, showing that it is symmetric. But it then follows that the matrix N must also be symmetric.

Now we continue with the proof, assuming that $D > 0$. The columns of $Z = N^T O D^{-1}$ are then orthonormal eigenvectors of $N^T N$ and the SVD of N is $N = ODZ^T$.

Since N is normal, we have $N^T NO = OD^2$, and

$$ZD^2 = N^T NZ = OD^2 O^T Z,$$

so that

$$O^T ZD^2 = D^2 O^T Z.$$

It follows from Exercise 3.7 that $O^T Z = B$ is diagonal. From $Z = OB$ and

$$N = ODZ^T = ODB^T O^T = ODBO^T = OCO^T,$$

where $C = DB$ is diagonal, it follows that $N^T = N$. ∎

This proof illustrates a use of the SVD of N, but the theorem can be proved using the eigenvector diagonalization of the normal matrix N itself. Note that the characteristic polynomial of N has real coefficients, so its roots occur in conjugate pairs. If N has a complex root λ, then both λ and $\overline{\lambda}$ are eigenvalues of N. It follows that $|\lambda|^2$ is an eigenvalue of $N^T N$ with multiplicity at least two. Consequently, if $N^T N$ has no repeated eigenvalues, then every eigenvalue of N is real. Using $U^\dagger NU = D$, with D real and diagonal, we get $N = UDU^\dagger$, so that $N^\dagger = UDU^\dagger = N$. Therefore N is real and Hermitian, and so is symmetric.

5.7.4 The Golub-Kahan Algorithm

We have obtained the SVD of A using the eigenvectors and eigenvalues of the Hermitian matrices $Q = A^\dagger A$ and $P = AA^\dagger$; for large matrices, this is not an efficient way to get the SVD. The Golub-Kahan algorithm [151] calculates the SVD of A without forming the matrices P and Q.

A matrix A is *bi-diagonal* if the only non-zero entries occur on the main diagonal and the first diagonal above the main one. Any matrix can be reduced to bi-diagonal form by multiplying the matrix first on the left by a

succession of Householder matrices, and then on the right by another succession of Householder matrices. The QR factorization is easier to calculate when the matrix involved is bi-diagonal.

The Golub-Kahan algorithm for calculating the SVD of A involves first reducing A to a matrix B in bi-diagonal form and then applying a variant of the QR factorization.

Using Householder matrices, we get unitary matrices U_0 and Z_0 such that $A = U_0 B Z_0^\dagger$, where B is bi-diagonal. Then we find the SVD of B,

$$B = \tilde{U}\Sigma\tilde{Z}^\dagger,$$

using QR factorization. Finally, the SVD for A itself is

$$A = U_0\tilde{U}\Sigma\tilde{Z}^\dagger Z_0^\dagger.$$

Ever since the publication of the Golub-Kahan algorithm, there have been efforts to improve both the accuracy and the speed of the method. The improvements announced in [120] and [121] won for their authors the 2009 SIAM Activity Group on Linear Algebra Prize.

5.8 Generalized Inverses

Even if A does not have an inverse, as, for example, when A is not square, it does have *generalized inverses* or *pseudo-inverses*.

Definition 5.4 *A matrix G is called a* generalized inverse *or* pseudo-inverse *for a matrix A if $x = Gb$ is a solution of $Ax = b$, whenever there are solutions.*

It is not obvious that generalized inverses exist for an arbitrary matrix A, but they do. In fact, we can use the SVD to obtain a pseudo-inverse for any A.

5.8.1 The Moore-Penrose Pseudo-Inverse

The *Moore-Penrose pseudo-inverse* is the matrix

$$A^\sharp = Z\Sigma^\sharp U^\dagger,$$

where $\Sigma^\sharp$ is the transpose of the matrix obtained from the matrix Σ in the SVD by taking the inverse of each of its nonzero entries and leaving unchanged the zero entries. The Moore-Penrose (MP) pseudo-inverse of $A^\dagger$ is

$$(A^\dagger)^\sharp = (A^\sharp)^\dagger = U(\Sigma^\sharp)^T Z^\dagger = U(\Sigma^\dagger)^\sharp Z^\dagger.$$

Ex. 5.16 *Show that $A^\sharp$ is a generalized inverse for A.*

Some important properties of the MP pseudo-inverse are the following:

- 1. $AA^\sharp A = A$,
- 2. $A^\sharp AA^\sharp = A^\sharp$,
- 3. $(A^\sharp A)^\dagger = A^\sharp A$,
- 4. $(AA^\sharp)^\dagger = AA^\sharp$.

The MP pseudo-inverse of an arbitrary M by N matrix A can be used in much the same way as the inverse of nonsingular matrices to find approximate or exact solutions of systems of equations $Ax = b$. The examples in the following exercises illustrate this point.

Ex. 5.17 *If $M > N$ the system $Ax = b$ probably has no exact solution. Show that whenever $A^\dagger A$ is invertible the pseudo-inverse of A is $A^\sharp = (A^\dagger A)^{-1}A^\dagger$ so that the vector $x = A^\sharp b$ is the least squares approximate solution.*

Ex. 5.18 *If $M < N$ the system $Ax = b$ probably has infinitely many solutions. Show that whenever the matrix $AA^\dagger$ is invertible the pseudo-inverse of A is $A^\sharp = A^\dagger(AA^\dagger)^{-1}$, so that the vector $x = A^\sharp b$ is the exact solution of $Ax = b$ closest to the origin; that is, it is the minimum norm solution.*

In general, the vector $A^\sharp b$ is the vector of smallest norm for which $\|Ax - b\|_2$ is minimized; that is, $A^\sharp b$ is the *minimum-norm least-squares* solution for the system $Ax = b$.

5.8.2 An Example of the MP Pseudo-Inverse

The matrix

$$A = \begin{bmatrix} 4 & 8 & 8 \\ 3 & 6 & 6 \end{bmatrix}$$

has MP pseudo-inverse

$$A^\sharp = \begin{bmatrix} 1/3 & 2/3 & 2/3 \\ 2/3 & -2/3 & 1/3 \\ 2/3 & 1/3 & -2/3 \end{bmatrix} \begin{bmatrix} 1/15 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 4/5 & 3/5 \\ 3/5 & -4/5 \end{bmatrix}.$$

5.8.3 Characterizing the MP Pseudo-Inverse

The MP pseudo-inverse is characterized by the four properties listed above. In other words, an N by M matrix X is the MP pseudo-inverse of A if and only if it satisfies the properties

- 1. $AXA = A$,
- 2. $XAX = X$,
- 3. $(XA)^\dagger = XA$,
- 4. $(AX)^\dagger = AX$.

5.8.4 Calculating the MP Pseudo-Inverse

The properties in the previous subsection that characterize the MP pseudo-inverse suggest algorithms for calculating $X = A^\sharp$ without first calculating the SVD. Let $X = A^\sharp$.

Lemma 5.3 *Let $C = XX^\dagger$. Then $CA^\dagger = X$.*

Proof: We have

$$CA^\dagger = XX^\dagger A^\dagger = X(AX)^\dagger = X(AX) = X.$$

∎

Lemma 5.4 *Let $B = A^\dagger AA^\dagger$. Then $B^\dagger C = A$.*

Proof: We have

$$B^\dagger C = AA^\dagger AXX^\dagger = AA^\dagger(AX)X^\dagger = AA^\dagger(AX)^\dagger X^\dagger = AA^\dagger(X^\dagger A^\dagger)X^\dagger$$

$$= AA^\dagger(XAX)^\dagger = AA^\dagger X^\dagger = A(XA)^\dagger = AXA = A.$$

∎

We know, therefore, that there is at least one Hermitian matrix W, namely $W = C$, having the property that $B^\dagger W = A$. We show now that if we have any Hermitian W with $B^\dagger W = A$, then $WA^\dagger = X = A^\sharp$.

Proposition 5.3 *If $B^\dagger W = A$ and $W^\dagger = W$, then $X = A^\sharp = WA^\dagger$.*

Proof: Let $Y = WA^\dagger$. We show first that $(YA)^\dagger = YA$, or, equivalently, $WA^\dagger A = A^\dagger AW$. From $WB = A^\dagger$ we have

$$A^\dagger(AW) = WB(AW) = WA^\dagger AA^\dagger(AW) = WA^\dagger(B^\dagger W) = WA^\dagger A.$$

Therefore, $(YA)^\dagger = YA$. Next, we show that $(AY)^\dagger = AY$. This is trivial, since we have

$$(AY)^\dagger = (AWA^\dagger)^\dagger = AWA^\dagger = AY.$$

Then we show $YAY = Y$. We have

$$Y = WA^\dagger = W(WB) = W(WA^\dagger A)A^\dagger = W(A^\dagger AW)A^\dagger = YAY.$$

Finally, we show that $AYA = A$. Again, this is easy, since

$$AYA = A(WA^\dagger A) = AA^\dagger AW = B^\dagger W = A.$$

This completes the proof of the proposition. ∎

This proposition suggests that we may be able to calculate the MP pseudo-inverse without first finding the SVD. Suppose that we solve the matrix equations $B^\dagger W = A$ and $W^\dagger = W$. Having found W, we form $Y = WA^\dagger = X$. One approach may be to solve iteratively the combined system $B^\dagger W = A$ and $W = \frac{1}{2}(W + W^\dagger)$. We leave it to the interested reader to investigate the feasibility of this idea.

5.9 Principal-Component Analysis and the SVD

The singular-value decomposition has many uses. One of the most important is as a tool for revealing information hidden in large amounts of data. A good illustration of this is *principal-component analysis* (PCA).

5.9.1 An Example

Suppose, for example, that D is an M by N matrix, that each row of D corresponds to particular applicant to the university, and that each column of D corresponds to a particular measurement of a student's ability or aptitude. One column of D could be SAT mathematics score, another could be IQ, and so on. To permit cross-measurement correlation, the actual scores are not stored, but only the difference between the actual score and the group average; if the average IQ for the group is 110 and John has an IQ of 103, then -7 is entered in the IQ column for John's row. We shall assume that M is greater than N.

The matrix $\frac{1}{M}D^\dagger D$ is the *covariance matrix*, each entry describing how one measurement category is related to a second. We shall focus on the matrix $D^\dagger D$, although proper statistical correlation would require that we normalize to remove the distortions coming from the use of scores that are not all on the same scale. How do we compare twenty points of difference

in IQ with one hundred points of difference in SAT score? Once we have calculated $D^\dagger D$, we may find that this N by N matrix is not diagonal, meaning that there is correlation between different measurement categories.

Although the column space of D, denoted $CS(D)$, the span of the columns of D in the space $\mathbb{C}^M$, is probably of dimension N, it may well be the case that the columns of D are nearly spanned by a much smaller set of its members; that is, there is a smaller subset of the column space $CS(D)$ such that each column of D is nearly equal to a linear combination of the members of this smaller set. That would suggest that knowing some of the columns of D, we could predict fairly well what the other columns would be. Statistically speaking, this would say that some scores are highly correlated with others. The goal of principal-component analysis is to find such a smaller set in $CS(D)$.

5.9.2 Decomposing $D^\dagger D$

The matrix $Q = D^\dagger D$ is Hermitian and nonnegative definite; almost certainly, all of its eigenvalues are positive. We list these eigenvalues as follows:

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N > 0,$$

and assume that λ_{J+k} is nearly zero, for $k = 1, 2, ..., N - J$. With u^j, $j = 1, ..., J$ denoting the orthonormal eigenvectors of $D^\dagger D$ corresponding to the first J eigenvalues, we see that the matrix $D^\dagger D$ is nearly equal to the sum of J dyads:

$$D^\dagger D \approx \sum_{j=1}^{J} \lambda_j u^j (u^j)^\dagger. \tag{5.7}$$

5.9.3 Decomposing D Itself

Let E be the N by J matrix whose J columns are the vectors u^j and R be the J by J diagonal matrix whose entries are $\lambda_j^{-1/2}$, for $j = 1, ..., J$. Let W be the M by J matrix $W = DER$. The matrix D is then approximately equal to the sum of J dyads:

$$D \approx \sum_{j=1}^{J} \sqrt{\lambda_j} w^j (u^j)^\dagger, \tag{5.8}$$

where w^j denotes the jth column of the matrix W. The approximation is with respect to the Frobenius norm. The columns of W lie in $CS(D)$ and each column of D is nearly in the span of the w^j. The w^j are the *principal-component vectors*.

5.9.4 Using the SVD in PCA

In the previous subsection, we obtained a decomposition of the matrix D using the eigenvectors and eigenvalues of the Hermitian matrix $D^\dagger D$. This is not an efficient way to proceed. Instead, we can use the SVD.

Let $A = D^\dagger$. As we saw previously, the singular-value decomposition of A is

$$A = U\Sigma Z^\dagger,$$

so that the SVD of the matrix D is

$$D = Z\Sigma^\dagger U^\dagger = \sum_{j=1}^{N} \sqrt{\lambda_j} z^j (u^j)^\dagger.$$

The first J columns of the matrix Z are the w^j defined above, so the Golub-Kahan SVD algorithm [151] can then be used to obtain the principal-component vectors of the data matrix D.

5.10 PCA and Factor Analysis

Principal-component analysis has as one of its goals the approximation of a covariance matrix $D^\dagger D$ by nonnegative-definite matrices of lower rank. A related area is *factor analysis*, which attempts to describe an arbitrary N by N Hermitian positive-definite matrix Q as $Q = G^\dagger G + K$, where G is some N by J matrix, for some $J < N$, and K is diagonal. Factor analysis views Q as a covariance matrix, $Q = E(vv^\dagger)$, where v is a random column vector with mean zero, and attempts to account for the off-diagonal correlated components of Q using the lower-rank matrix $G^\dagger G$. Underlying this is the following model for the random vector v:

$$v = Gx + w,$$

where both x and w are uncorrelated. The entries of the random vector x are the *common factors* that affect each entry of v while those of w are the *special factors*, each associated with a single entry of v. Factor analysis plays an increasingly prominent role in signal and image processing [36] as well as in the social sciences.

In [248] Gil Strang points out that, from a linear algebra standpoint, factor analysis raises some questions. As his example shows, the representation of Q as $Q = G^\dagger G + K$ is not unique. The matrix Q does not uniquely determine the size of the matrix G:

$$Q = \begin{bmatrix} 1 & .74 & .24 & .24 \\ .74 & 1 & .24 & .24 \\ .24 & .24 & 1 & .74 \\ .24 & .24 & .74 & 1 \end{bmatrix} = \begin{bmatrix} .7 & .5 \\ .7 & .5 \\ .7 & -.5 \\ .7 & -.5 \end{bmatrix} \begin{bmatrix} .7 & .7 & .7 & .7 \\ .5 & .5 & -.5 & -.5 \end{bmatrix} + .26I$$

and

$$Q = \begin{bmatrix} .6 & \sqrt{.38} & 0 \\ .6 & \sqrt{.38} & 0 \\ .4 & 0 & \sqrt{.58} \\ .4 & 0 & \sqrt{.58} \end{bmatrix} \begin{bmatrix} .6 & .6 & .4 & .4 \\ \sqrt{.38} & \sqrt{.38} & 0 & 0 \\ 0 & 0 & \sqrt{.58} & \sqrt{.58} \end{bmatrix} + .26I.$$

It is also possible to represent Q with different diagonal components K.

5.11 Schmidt's MUSIC Method

The "multiple signal identification and classification" (MUSIC) method, originally due to Schmidt [236], is similar to PCA in some respects.

The basic problem now is the following. We have a positive-definite N by N matrix R that we believe has the form

$$R = \sum_{j=1}^{J} \alpha_j e^j (e^j)^\dagger + \sigma^2 I = S + \sigma^2 I, \tag{5.9}$$

where $J < N$ is not known, and the scalars σ and $\alpha_j > 0$, and the column vectors e^j are not known, but are assumed to be linearly independent. The problem is to determine these unknown scalars and vectors. In applications we usually do have a model for the vectors e^j: it is assumed that each e^j has the form $e^j = e(\theta_j)$, where θ_j is an unknown member of a known family of parameters denoted by θ.

We can say that $R = G^\dagger G + K$, where now $K = \sigma^2 I$, so the MUSIC problem fits into the formulation of factor analysis also. But the MUSIC does more than find a G; it uses the model of parameterized vectors $e(\theta)$ to determine the individual e^j.

The MUSIC method proceeds as follows. First, we calculate the eigenvector/eigenvalue decomposition of R. Let $\lambda_1 \geq ... \geq \lambda_N > 0$ be the ordered eigenvalues, with associated orthonormal eigenvectors u^j. Since $J < N$, we know that the rank of S is J, so that the system $Sx = 0$ has $N - J$ linearly independent solutions. Each of these is an eigenvector of S corresponding

to the eigenvalue 0. Therefore, they are also eigenvectors of R corresponding to the eigenvalue $\lambda = \sigma^2$. Since, for $j = 1, 2, ..., J$, $Su^j \neq 0$, for these j we have $\lambda_j > \sigma^2$. So we can tell what J is from the list of eigenvalues of R. Now we find the θ_j. Note that the e^j are in the span of the $u^1, ..., u^J$, but they are not the u^j themselves, generally, since the e^j are probably not mutually orthogonal.

For each $m = 1, ..., N - J$ and each $j = 1, ..., J$, the eigenvector u^{J+m} is orthogonal to e^j. Therefore, the function of θ given by

$$F(\theta) = \sum_{m=1}^{N-J} |(u^{J+m})^\dagger e(\theta)|^2 \tag{5.10}$$

is such that $F(\theta_j) = 0$, for $j = 1, ..., J$. In most situations $F(\theta)$ will have precisely J zeros in the parameter family, so the zeros of $F(\theta)$ will identify the parameter values θ_j. Finding these parameter values then amounts to determining approximately the zeros of $F(\theta)$. Once J and the θ_j have been found, determining the coefficients α_j becomes a linear problem.

5.12 Singular Values of Sparse Matrices

In image reconstruction from projections the M by N matrix A is usually quite large and often ϵ-sparse; that is, most of its elements do not exceed ϵ in absolute value, where ϵ denotes a small positive quantity.

In transmission tomography each column of A corresponds to a single pixel in the digitized image, while each row of A corresponds to a line segment through the object, along which an x-ray beam has traveled. The entries of a given row of A are nonzero only for those columns whose associated pixel lies on that line segment; clearly, most of the entries of any given row of A will then be zero.

In emission tomography the I by J nonnegative matrix P has entries $P_{ij} \geq 0$; for each detector i and pixel j, P_{ij} is the probability that an emission at the jth pixel will be detected at the ith detector. When a detection is recorded at the ith detector, we want the likely source of the emission to be one of only a small number of pixels. For single-photon emission tomography (SPECT), a lead collimator is used to permit detection of only those photons approaching the detector straight on. In positron emission tomography (PET), coincidence detection serves much the same purpose. In both cases the probabilities P_{ij} will be zero (or nearly zero) for most combinations of i and j. Such matrices are called *sparse* (or *almost sparse*).

We discuss now a convenient estimate for the largest singular value of an almost sparse matrix A, which, for notational convenience only, we take

to be real. Related estimates of the largest singular value will be presented later, in Chapter 15.

In [62] it was shown that if A is normalized so that each row has length one, then the spectral radius of A^TA, which is the square of the largest singular value of A itself, does not exceed the maximum number of nonzero elements in any column of A. A similar upper bound on $\rho(A^TA)$ can be obtained for non-normalized, ϵ-sparse A.

Let A be an M by N matrix. For each $n = 1, ..., N$, let $s_n > 0$ be the number of nonzero entries in the nth column of A, and let s be the maximum of the s_n. Let G be the M by N matrix with entries

$$G_{mn} = A_{mn}/(\sum_{l=1}^{N} s_l A_{ml}^2)^{1/2}.$$

Lent has shown that the eigenvalues of the matrix G^TG do not exceed one [197]. This result suggested the following proposition, whose proof was given in [62].

Proposition 5.4 *Let A be an M by N matrix. For each $m = 1, ..., M$ let $\nu_m = \sum_{n=1}^{N} A_{mn}^2 > 0$. For each $n = 1, ..., N$ let $\sigma_n = \sum_{m=1}^{M} e_{mn}\nu_m$, where $e_{mn} = 1$ if $A_{mn} \neq 0$ and $e_{mn} = 0$ otherwise. Let σ denote the maximum of the σ_n. Then the eigenvalues of the matrix A^TA do not exceed σ. If A is normalized so that the Euclidean length of each of its rows is one, then the eigenvalues of A^TA do not exceed s, the maximum number of nonzero elements in any column of A.*

Proof: For simplicity, we consider only the normalized case; the proof for the more general case is similar.

Let $A^TAv = cv$ for some nonzero vector v. We show that $c \leq s$. We have $AA^TAv = cAv$ and so $w^TAA^Tw = v^TA^TAA^TAv = cv^TA^TAv = cw^Tw$, for $w = Av$. Then, with $e_{mn} = 1$ if $A_{mn} \neq 0$ and $e_{mn} = 0$ otherwise, we have

$$(\sum_{m=1}^{M} A_{mn}w_m)^2 = (\sum_{m=1}^{M} A_{mn}e_{mn}w_m)^2$$

$$\leq (\sum_{m=1}^{M} A_{mn}^2 w_m^2)(\sum_{m=1}^{M} e_{mn}^2) =$$

$$(\sum_{m=1}^{M} A_{mn}^2 w_m^2)s_j \leq (\sum_{m=1}^{M} A_{mn}^2 w_m^2)s.$$

Therefore,

$$w^TAA^Tw = \sum_{n=1}^{N}(\sum_{m=1}^{M} A_{mn}w_m)^2 \leq \sum_{n=1}^{N}(\sum_{m=1}^{M} A_{mn}^2 w_m^2)s,$$

and

$$w^T AA^T w = c \sum_{m=1}^{M} w_m^2 = c \sum_{m=1}^{M} w_m^2 (\sum_{n=1}^{N} A_{mn}^2)$$

$$= c \sum_{m=1}^{M} \sum_{n=1}^{N} w_m^2 A_{mn}^2.$$

This completes the proof. ∎

If we normalize A so that its rows have length one, then the trace of the matrix AA^T is $\mathrm{tr}(AA^T) = M$, which is also the sum of the eigenvalues of $A^T A$. Consequently, the maximum eigenvalue of $A^T A$ does not exceed M; this result improves that upper bound considerably, if A is sparse and so $s << M$. A more general theorem along the same lines is Theorem 15.5.

In image reconstruction from projection data that includes scattering we often encounter matrices A most of whose entries are small, if not exactly zero. A slight modification of the proof provides us with a useful upper bound for L, the largest eigenvalue of $A^T A$, in such cases. Assume that the rows of A have length one. For $\epsilon > 0$ let s be the largest number of entries in any column of A whose magnitudes exceed ϵ. Then we have

$$L \leq s + MN\epsilon^2 + 2\epsilon(MNs)^{1/2}.$$

The proof of this result is similar to that for Proposition 5.4.

5.13 The "Matrix Inversion Theorem"

In this section we bring together several of the conditions equivalent to saying that an N by N matrix A is invertible. Taken together, these conditions are sometimes called the "Matrix Inversion Theorem". The equivalences on the list are roughly in increasing order of difficulty of proof. The reader is invited to supply proofs. We begin with the definition of invertibility.

- 1. According to the definition of invertibility, we say A is invertible if there is a matrix B such that $AB = BA = I$. Then $B = A^{-1}$, the inverse of A.
- 2. A is invertible if and only if there are matrices B and C such that $AB = CA = I$. Then $B = C = A^{-1}$.
- 3. A is invertible if and only if the rank of A is N.

- 4. A is invertible if and only if there is a matrix B with $AB = I$. Then $B = A^{-1}$.
- 5. A is invertible if and only if the columns of A are linearly independent.
- 6. A is invertible if and only if $Ax = 0$ implies $x = 0$.
- 7. A is invertible if and only if A can be transformed by elementary row operations into an upper triangular matrix having no zero entries on its main diagonal.
- 8. A is invertible if and only if the upper triangular matrix $T = U^\dagger AU$ given by Schur's Lemma is invertible, and if and only if there are no zeros on the main diagonal of T.
- 9. A is invertible if and only if its determinant is not zero.
- 10. A is invertible if and only if A has no zero eigenvalues.

5.14 Matrix Diagonalization and Systems of Linear ODE's

We know that the ordinary linear differential equation

$$x'(t) = ax(t)$$

has the solution

$$x(t) = x(0)e^{at}.$$

In this section we use matrix diagonalization to generalize this solution to systems of linear ordinary differential equations.

Consider the system of linear ordinary differential equations

$$x'(t) = 4x(t) - y(t) \tag{5.11}$$
$$y'(t) = 2x(t) + y(t), \tag{5.12}$$

which we write as $z'(t) = Az(t)$, with

$$A = \begin{bmatrix} 4 & -1 \\ 2 & 1 \end{bmatrix},$$

$$z(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix},$$

and

$$z'(t) = \begin{bmatrix} x'(t) \\ y'(t) \end{bmatrix}.$$

We then have

$$\det(A - \lambda I) = (4 - \lambda)(1 - \lambda) + 2 = (\lambda - 2)(\lambda - 3),$$

so the eigenvalues of A are $\lambda = 2$ and $\lambda = 3$.

The vector u given by

$$u = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

solves the system $Au = 2u$ and the vector v given by

$$v = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

solves the system $Av = 3v$. Therefore, u and v are linearly independent eigenvectors of A. With

$$B = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix},$$

$$B^{-1} = \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix},$$

and

$$D = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix},$$

we have $A = BDB^{-1}$ and $B^{-1}AB = D$; this is a diagonalization of A using its eigenvalues and eigenvectors. In this example A is not symmetric, the eigenvectors of A are not mutually orthogonal, and B^{-1} is not B^T.

Note that not every N by N matrix A will have such a diagonalization; we need N linearly independent eigenvectors of A, which need not exist. They do exist if the eigenvalues of A are all different, as in the example here, and also if the matrix A is Hermitian or normal. The reader should prove that matrix

$$M = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

has no such diagonalization.

Continuing with our example, we let $w(t) = B^{-1}z(t)$ so that $w'(t) = Dw(t)$. Because D is diagonal, this new system is uncoupled;

$$w_1'(t) = 2w_1(t),$$

and

$$w_2'(t) = 3w_2(t).$$

The solutions are then

$$w_1(t) = w_1(0)e^{2t},$$

and

$$w_2(t) = w_2(0)e^{3t}.$$

It follows from $z(t) = Bw(t)$ that

$$x(t) = w_1(0)e^{2t} + w_2(0)e^{3t},$$

and

$$y(t) = 2w_1(0)e^{2t} + w_2(0)e^{3t}.$$

We want to express $x(t)$ and $y(t)$ in terms of $x(0)$ and $y(0)$. To do this we use $z(0) = Bw(0)$, which tells us that

$$x(t) = (-x(0) + y(0))e^{2t} + (2x(0) - y(0))e^{3t},$$

and

$$y(t) = (-2x(0) + 2y(0))e^{2t} + (2x(0) - y(0))e^{3t}.$$

We can rewrite this as

$$z(t) = E(t)z(0),$$

where

$$E(t) = \begin{bmatrix} -e^{2t} + 2e^{3t} & e^{2t} - e^{3t} \\ -2e^{2t} + 2e^{3t} & 2e^{2t} - e^{3t} \end{bmatrix}.$$

What is the matrix $E(t)$?

To mimic the solution $x(t) = x(0)e^{at}$ of the problem $x'(t) = ax(t)$, we try

$$z(t) = e^{tA}z(0),$$

with the matrix exponential defined by

$$e^{tA} = \sum_{n=0}^{\infty} \frac{1}{n!} t^n A^n.$$

Since $A = BDB^{-1}$, it follows that $A^n = BD^nB^{-1}$, so that

$$e^{tA} = Be^{tD}B^{-1}.$$

Since D is diagonal, we have

$$e^{tD} = \begin{bmatrix} e^{2t} & 0 \\ 0 & e^{3t} \end{bmatrix}.$$

A simple calculation shows that

$$e^{tA} = B \begin{bmatrix} e^{2t} & 0 \\ 0 & e^{3t} \end{bmatrix} B^{-1} = \begin{bmatrix} -e^{2t} + 2e^{3t} & e^{2t} - e^{3t} \\ -2e^{2t} + 2e^{3t} & 2e^{2t} - e^{3t} \end{bmatrix} = E(t).$$

Therefore, the solution of the original system is

$$z(t) = e^{tA}z(0).$$

5.15 Classical Lie Algebras

Any additive group of square matrices that is closed under the *commutation operation* $[A, B] = AB - BA$ is a matrix Lie (pronounced "Lee") algebra. Here are some examples. Unless otherwise noted, the entries can be real or complex.

- **1.** The collection M_N of all N by N matrices.
- **2.** The collection of matrices in M_N with zero trace.
- **3.** The collection of all real skew-symmetric matrices in M_N.
- **4.** The collection of all A in M_N with $A + A^\dagger = 0$.

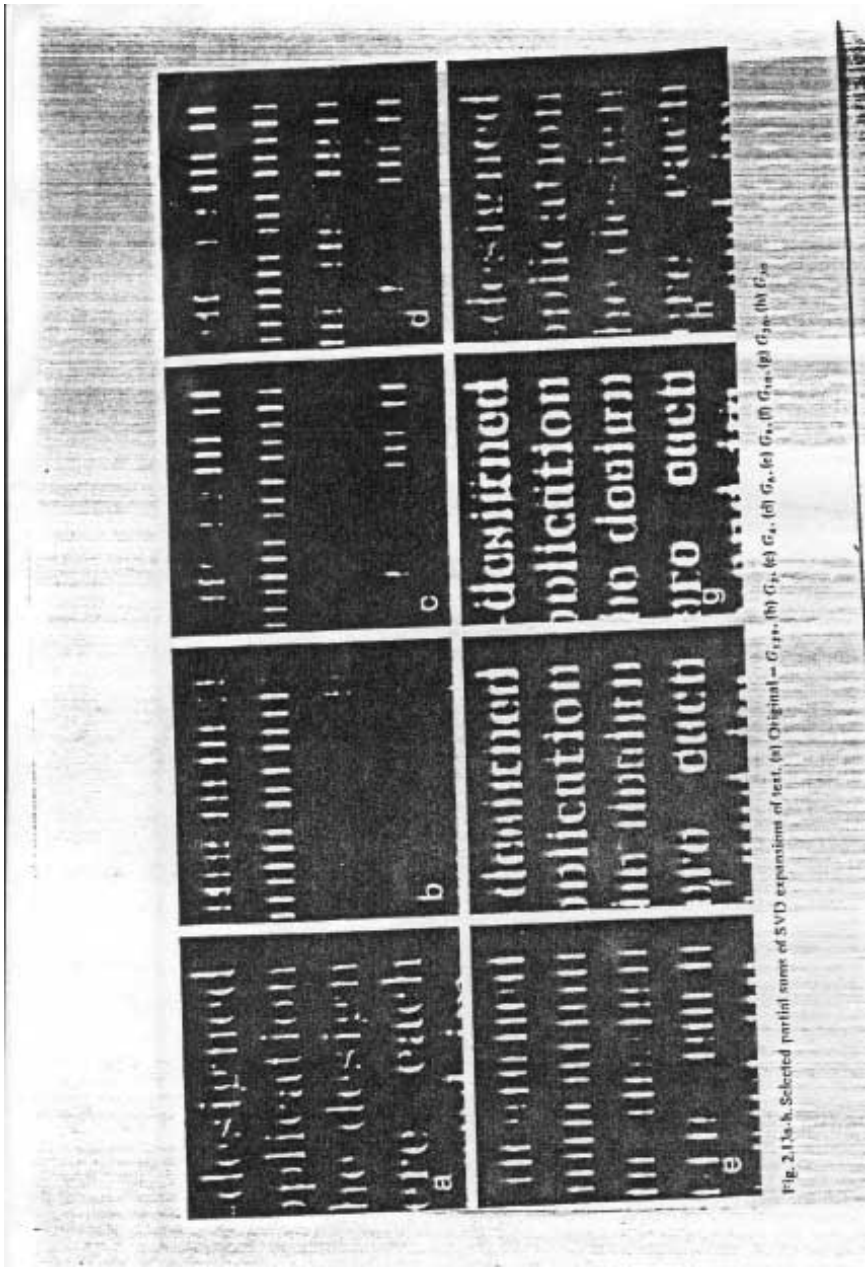

FIGURE 5.1: Compressing text with the SVD.

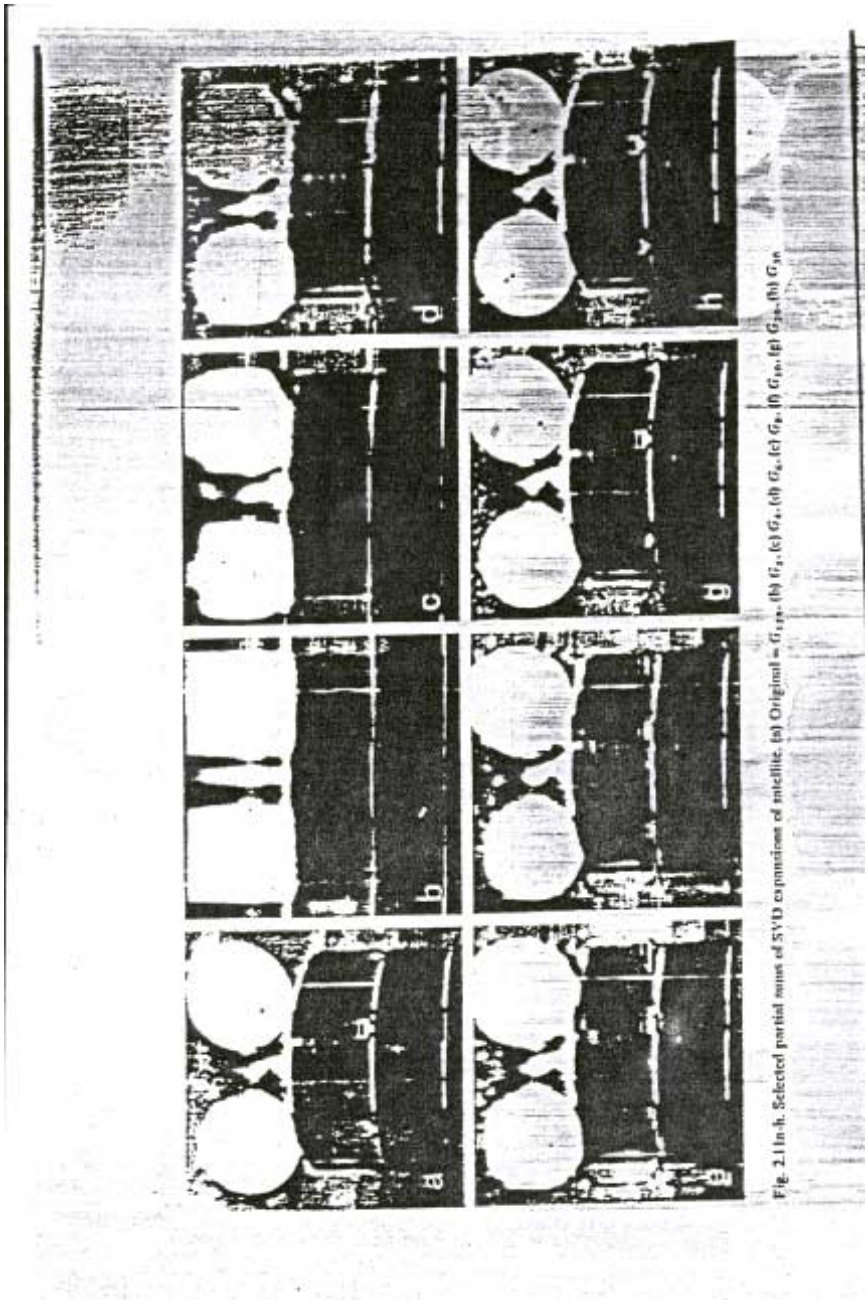

FIGURE 5.2: Compressing an image with the SVD.

Chapter 6

Metric Spaces and Norms

6.1 Chapter Summary

In many applications in which we seek a solution of a linear system of equations $Ax = b$ the entries of the vector b are measurements. If small changes in b result in large changes in the solution x, then we have an unstable situation. In order to measure such changes we need a notion of size of a vector. This leads us to study metrics and norms.

The usual dot product is an inner product on $\mathbb{R}^J$ or $\mathbb{C}^J$ and can be used to define the Euclidean norm $\|x\|_2$ of a vector x, which, in turn, provides a *metric*, or a measure of distance between two vectors, $d(x, y) = \|x - y\|_2$. The notions of metric and norm are actually more general notions, with no necessary connection to the inner product.

6.2 Metric Space Topology

To prepare for our discussion of norms on vectors and matrices we take a quick look at metric space topology.

6.2.1 General Topology

Let S be a non-empty set and $\mathcal{T}$ a non-empty collection of subsets of S. The collection $\mathcal{T}$ is called a *topology* for S if the following conditions hold:

- 1. the empty set and the set S are in $\mathcal{T}$;
- 2. for any finite or infinite sub-collection of members of $\mathcal{T}$, their union is again in $\mathcal{T}$;
- 3. for any positive integer N and sets U_n, $n = 1, 2, ..., N$ in $\mathcal{T}$, their intersection, the set $\cap_{n=1}^{N} U_n$, is in $\mathcal{T}$.

The members of $\mathcal{T}$ are then called the *open sets* for the topology. Notice that we are not given any property that a subset of S may or may not have such that having it would qualify the subset to be called open; a subset of S is open precisely when it is a member of the topology, that is, when it is a member of the collection of subsets called the open subsets. The empty set and S itself are always open, but there need not be any other open subsets. On the other hand, it could be the case that every subset of S is open. It all depends on the collection $\mathcal{T}$ we are given. The *interior* of a subset C of S is the largest open subset of S that is contained within C.

A subset C of S is called a *closed* subset if its complement, the set of all members of S that are not in C, is an open set. The *closure* of a subset C is the smallest closed subset of S that contains C. Once again, we do not describe what it means to be a closed set in terms of some property that C may or may not have, except that its complement is open.

Although the terminology sounds familiar and is borrowed from geometry, these definitions are quite abstract and it is remarkable that a deep theory of topological spaces and continuous functions can be built on such definitions.

6.2.2 Metric Spaces

Metric spaces are the most important and most familiar examples of topological spaces. In contrast to what happens in general topology, now the fundamental notion is that of a *metric* and sets are called open or closed depending on how they behave with respect to the metric. Unlike the general case, now the topology is built up by defining what it means for an individual subset to be open and then including all such subsets in the topology $\mathcal{T}$. We begin with the basic definitions.

Definition 6.1 *Let S be a non-empty set. We say that the function $d : S \times S \to [0, +\infty)$ is a* metric *if the following hold:*

$$d(s,t) \geq 0, \tag{6.1}$$

for all s and t in S;

$$d(s,t) = 0 \tag{6.2}$$

if and only if $s = t$;

$$d(s,t) = d(t,s), \tag{6.3}$$

for all s and t in S; and, for all s, t, and u in S,

$$d(s,t) \leq d(s,u) + d(u,t). \tag{6.4}$$

The pair $\{S, d\}$ is a metric space.

The last inequality is the *Triangle Inequality* for this metric.

6.3 Analysis in Metric Space

Analysis is concerned with issues of convergence and limits.

Definition 6.2 *A sequence* $\{s^k\}$, $k = 1, 2, ...$, *in the metric space* (S, d) *is said to have* limit s^* *if*

$$\lim_{k \to +\infty} d(s^k, s^*) = 0. \tag{6.5}$$

Any sequence with a limit is said to be convergent.

Ex. 6.1 *Show that a sequence can have at most one limit.*

Definition 6.3 *The sequence* $\{s^k\}$ *is said to be a* Cauchy sequence *if, for any* $\epsilon > 0$, *there is positive integer* m, *such that, for any nonnegative integer* n,

$$d(s^m, s^{m+n}) \leq \epsilon. \tag{6.6}$$

Ex. 6.2 *Show that every convergent sequence is a Cauchy sequence.*

Definition 6.4 *The metric space* (S, d) *is said to be* complete *if every Cauchy sequence is a convergent sequence.*

Completeness is part of the axiomatic approach to the definition of the real numbers. From that, it follows that the finite-dimensional spaces $\mathbb{R}^J$and $\mathbb{C}^J$ are complete metric spaces, with respect to the usual Euclidean distance.

Ex. 6.3 *Let* S *be the set of rational numbers, with* $d(s, t) = |s - t|$. *Show that* (S, d) *is a metric space, but not a complete metric space.*

Definition 6.5 *A sequence* $\{s^k\}$ *in* S *is said to be* bounded *if there is a positive constant* $b > 0$ *such that* $d(s^1, s^k) \leq b$, *for all* k.

Ex. 6.4 *Show that any convergent sequence in a metric space is bounded. Find a bounded sequence of real numbers that is not convergent.*

Ex. 6.5 *Show that, if* $\{s^k\}$ *is bounded, then, for any element* c *in the metric space, there is a constant* $r > 0$, *with* $d(c, s^k) \leq r$, *for all* k.

Definition 6.6 *A point* s *in* S *is a* limit point *of a subset* C *of* S *if there are members* c^k *of* C *such that the sequence* $\{c^k\}$ *converges to* s. *Denote by* C^* *the set of all limit points of the set* C.

For any c in C the constant sequence formed by taking $c^k = c$ for each k converges to c. Therefore, every point of C is a limit point of C and $C \subseteq C^*$.

Definition 6.7 *A subset* C *of the metric space is said to be* closed *if every limit point of* C *is in* C; *that is,* $C = C^*$. *The* closure *of a subset* C, *denoted* $cl(C)$, *is the smallest closed set containing* C.

For example, in $\mathbb{R}^J = \mathbb{R}$, the set $C = (0, 1]$ is not closed, because it does not contain the point $s = 0$, which is the limit of the sequence $\{s^k = \frac{1}{k}\}$; the set $C = [0, 1]$ is closed and is the *closure* of the set $(0, 1]$, that is, it is the smallest closed set containing $(0, 1]$.

It is not obvious that there is always a smallest closed set containing C, so it is not clear that the closure of C is well defined. The following proposition gives an explicit description of the closure of C.

Proposition 6.1 *For any subset C of S the closure of C is the set C^*.*

This proposition tells us that we obtain the closure of C by including all its limit points.

Ex. 6.6 *Prove Proposition 6.1. Hint: you need to show that the set C^* is a closed set, which is not immediately obvious. If you think it is obvious, think again.*

Definition 6.8 *For any bounded sequence $\{x^k\}$ in $\mathbb{R}^J$, there is at least one subsequence, often denoted $\{x^{k_n}\}$, that is convergent; the notation implies that the positive integers k_n are ordered, so that $k_1 < k_2 < ....$ The limit of such a subsequence is then said to be a* cluster point *of the original sequence.*

Ex. 6.7 *Show that your bounded, but not convergent, sequence found in Exercise 6.4 has a cluster point.*

Ex. 6.8 *Show that, if x is a cluster point of the sequence $\{x^k\}$, and if $d(x, x^k) \geq d(x, x^{k+1})$, for all k, then x is the limit of the sequence.*

6.4 Motivating Norms

We turn now to metrics that come from norms. Our interest in norms for vectors and matrices stems from their usefulness in analyzing iterative algorithms. Most of the algorithms we shall study involve generating a sequence of vectors $\{x^k\}, k = 0, 1, 2, ...$ in $\mathbb{R}^J$ or $\mathbb{C}^J$, where x^{k+1} comes from x^k according to the formula $x^{k+1} = T(x^k)$, where T is a (possibly nonlinear) operator on the space of vectors. When we investigate iterative algorithms, we will want to know if the sequence $\{x^k\}$ generated by the algorithm converges. As a first step, we will usually ask if the sequence is bounded? If it is bounded, then it will have at least one cluster point. We then try to discover if that cluster point is really the limit of the sequence.

It would help if we know that the vector $T(x) - T(y)$ is smaller, in some sense, than the vector $x - y$.

Affine operators T have the form $T(x) = Bx + d$, where B is a matrix and d is a fixed vector. Such affine operators arise, for example, in the Landweber algorithm for solving $Ax = b$; the iterative step is

$$x^{k+1} = x^k + \gamma A^\dagger(b - (Ax^k)),$$

which we can write as

$$x^{k+1} = (I - \gamma A^\dagger A)x^k + \gamma A^\dagger b.$$

Then $x^{k+1} = T(x^k)$, where T is the affine operator

$$T(x) = (I - \gamma A^\dagger A)x + \gamma A^\dagger b.$$

For affine operators $T(x) - T(y) = Bx - By = B(x - y)$, so we are interested in the size of Bz, relative to the size of z, for all vectors z. Vector and matrix norms will help us here.

6.5 Norms

The metric spaces that interest us most are vector spaces V for which the metric comes from a norm, which is a measure of the length of a vector.

Definition 6.9 *We say that $\|\cdot\|$ is a* norm *on V if*

$$\|x\| \geq 0, \tag{6.7}$$

for all x,

$$\|x\| = 0 \tag{6.8}$$

if and only if $x = 0$,

$$\|\gamma x\| = |\gamma| \, \|x\|, \tag{6.9}$$

for all x and scalars γ, and

$$\|x + y\| \leq \|x\| + \|y\|, \tag{6.10}$$

for all vectors x and y.

Lemma 6.1 *The function $d(x, y) = \|x - y\|$ defines a metric on V.*

It can be shown that $\mathbb{R}^J$ and $\mathbb{C}^J$ are complete for any metric arising from a norm.

6.5.1 Some Common Norms on $\mathbb{C}^J$

We consider now the most common norms on the space $\mathbb{C}^J$. These notions apply equally to $\mathbb{R}^J$.

6.5.1.1 The 1-norm

The 1-norm on $\mathbb{C}^J$ is defined by

$$\|x\|_1 = \sum_{j=1}^{J} |x_j|. \tag{6.11}$$

6.5.1.2 The ∞-norm

The ∞-norm on $\mathbb{C}^J$ is defined by

$$\|x\|_\infty = \max\{|x_j| \, | j = 1, ..., J\}. \tag{6.12}$$

6.5.1.3 The p-norm

For any $p \geq 1$, the p-norm is defined by

$$\|x\|_p = \Big(\sum_{j=1}^{J} |x_j|^p \Big)^{1/p}. \tag{6.13}$$

6.5.1.4 The 2-norm

The 2-norm, also called the Euclidean norm, is the most commonly used norm on $\mathbb{C}^J$. It is the p-norm for $p = 2$ and is the one that comes from the inner product:

$$\|x\|_2 = \sqrt{\sum_{j=1}^{J} |x_j|^2} = \sqrt{\langle x, x \rangle} = \sqrt{x^\dagger x}. \tag{6.14}$$

6.5.1.5 Weighted 2-norms

Let A be an invertible matrix and $Q = A^\dagger A$. Define

$$\|x\|_Q = \|Ax\|_2 = \sqrt{x^\dagger Q x}, \tag{6.15}$$

for all vectors x. This is the *Q-weighted* 2-norm of x. If Q is the diagonal matrix with diagonal entries $Q_{jj} > 0$, then

$$\|x\|_Q = \sqrt{\sum_{j=1}^{J} Q_{jj} |x_j|^2}. \tag{6.16}$$

Ex. 6.9 *Show that the* 1*-norm is a norm.*

Ex. 6.10 *Show that the* ∞*-norm is a norm.*

Ex. 6.11 *Show that the* 2*-norm is a norm. Hint: for the triangle inequality, use the Cauchy Inequality.*

Ex. 6.12 *Show that the* Q*-weighted* 2*-norm is a norm.*

6.6 The Generalized Arithmetic-Geometric Mean Inequality

Suppose that $x_1, ..., x_N$ are positive numbers. Let $a_1, ..., a_N$ be positive numbers that sum to one. Then the *Generalized AGM Inequality* (GAGM Inequality) is

$$x_1^{a_1} x_2^{a_2} \cdots x_N^{a_N} \leq a_1 x_1 + a_2 x_2 + ... + a_N x_N, \tag{6.17}$$

with equality if and only if $x_1 = x_2 = ... = x_N$. We can prove this using the convexity of the function $-\log x$.

6.7 The Hölder and Minkowski Inequalities

To show that the p-norm is a norm we need Minkowski's Inequality, which follows from Hölder's Inequality.

Let $c = (c_1, ..., c_N)$ and $d = (d_1, ..., d_N)$ be vectors with complex entries and let p and q be positive real numbers such that

$$\frac{1}{p} + \frac{1}{q} = 1.$$

The p-norm of c is defined to be

$$\|c\|_p = \Big(\sum_{n=1}^{N} |c_n|^p \Big)^{1/p},$$

with the q-norm of d, denoted $\|d\|_q$, defined similarly.

6.7.1 Hölder's Inequality

Hölder's Inequality is the following:

$$\sum_{n=1}^{N} |c_n d_n| \leq \|c\|_p \|d\|_q,$$

with equality if and only if

$$\Big(\frac{|c_n|}{\|c\|_p}\Big)^p = \Big(\frac{|d_n|}{\|d\|_q}\Big)^q,$$

for each n.

Hölder's Inequality follows from the GAGM Inequality. To see this, we fix n and apply Inequality (6.17), with

$$x_1 = \Big(\frac{|c_n|}{\|c\|_p}\Big)^p,$$

$$a_1 = \frac{1}{p},$$

$$x_2 = \Big(\frac{|d_n|}{\|d\|_q}\Big)^q,$$

and

$$a_2 = \frac{1}{q}.$$

From (6.17) we then have

$$\Big(\frac{|c_n|}{\|c\|_p}\Big)\Big(\frac{|d_n|}{\|d\|_q}\Big) \leq \frac{1}{p}\Big(\frac{|c_n|}{\|c\|_p}\Big)^p + \frac{1}{q}\Big(\frac{|d_n|}{\|d\|_q}\Big)^q.$$

Now sum both sides over the index n.

It will be helpful later to note here that

$$\sum_{n=1}^{N} \overline{c_n} d_n = \sum_{n=1}^{N} |c_n||d_n|$$

if each $\overline{c_n} d_n$ is non-negative, which means that the complex numbers c_n and d_n have the same phase angles.

6.7.2 Minkowski's Inequality

Minkowski's Inequality, which is a consequence of Hölder's Inequality, states that

$$\|c + d\|_p \leq \|c\|_p + \|d\|_p\,;$$

it is the triangle inequality for the metric induced by the p-norm.

To prove Minkowski's Inequality, we write

$$\sum_{n=1}^{N} |c_n + d_n|^p \leq \sum_{n=1}^{N} |c_n|(|c_n + d_n|)^{p-1} + \sum_{n=1}^{N} |d_n|(|c_n + d_n|)^{p-1}.$$

Then we apply Hölder's Inequality to both of the sums on the right side of the equation.

For the choices $p = q = 2$, Hölder's Inequality becomes the famous Cauchy Inequality.

Ex. 6.13 *Show that the p-norm is a norm.*

6.8 Matrix Norms

Any I by J matrix A can be turned into a column vector in $\mathbb{C}^{I+J}$ by vectorization; that is, by writing each column of A below the previous one. Therefore, we can define a norm for any matrix by simply vectorizing the matrix and taking a norm of the resulting vector; the 2-norm of the vectorized matrix is the *Frobenius norm* of the matrix itself. Such norms for matrices may not be compatible with the role of a matrix as representing a linear transformation. For that reason, we consider norms on matrices that are induced by the norms of the vectors on which the matrices operate.

Definition 6.10 *Let A be an I by J complex matrix. A norm on A, denoted $\|A\|$, is said to be* compatible *with given norms on $\mathbb{C}^J$ and $\mathbb{C}^I$ if $\|Ax\| \leq \|A\|\|x\|$, for every x in $\mathbb{C}^J$.*

6.8.1 Induced Matrix Norms

One way to obtain a compatible norm for matrices is through the use of an induced matrix norm.

Definition 6.11 *Let $\|x\|$ be any norm on $\mathbb{C}^J$, not necessarily the Euclidean norm, $\|b\|$ any norm on $\mathbb{C}^I$, and A a rectangular I by J matrix. The* induced matrix norm *of A, simply denoted $\|A\|$, derived from these two vector norms, is the smallest positive constant c such that*

$$\|Ax\| \leq c\|x\|, \tag{6.18}$$

for all x in $\mathbb{C}^J$. This induced norm can be written as

$$\|A\| = \max_{x \neq 0}\{\|Ax\|/\|x\|\}. \tag{6.19}$$

When A is square we always assume that it is the same norm being used on x and Ax.

We study induced matrix norms in order to measure the distance from Ax to Az, $\|Ax - Az\|$, relative to $\|x - z\|$, the distance from x to z:

$$\|Ax - Az\| \leq \|A\| \, \|x - z\|, \tag{6.20}$$

for all vectors x and z and $\|A\|$ is the smallest number for which this statement is valid.

Ex. 6.14 *Show that $\rho(S) \leq \|S\|$ for any square matrix S.*

Ex. 6.15 *Let the matrices A be I by J, and B be J by K. Show that, for any norms on the spaces $\mathbb{R}^I$, $\mathbb{R}^J$ and $\mathbb{R}^K$, we have the inequality*

$$\|AB\| \leq \|A\| \, \|B\|,$$

for the induced matrix norms.

Using the next two lemmas, we can show that there are induced matrix norms for S that are as close to $\rho(S)$ as we wish.

Lemma 6.2 *Let M be an invertible matrix and $\|x\|$ any vector norm. Define*

$$\|x\|_M = \|Mx\|. \tag{6.21}$$

Then, for any square matrix S, the matrix norm

$$\|S\|_M = \max_{x \neq 0} \{\|Sx\|_M / \|x\|_M\} \tag{6.22}$$

is

$$\|S\|_M = \|MSM^{-1}\|. \tag{6.23}$$

In [7] this result is used to prove the following lemma:

Lemma 6.3 *Let S be any square matrix and let $\epsilon > 0$ be given. Then there is an invertible matrix M such that*

$$\|S\|_M \leq \rho(S) + \epsilon. \tag{6.24}$$

Later, we shall show that if a J by J matrix S is diagonalizable, that is, if there is a basis for $\mathbb{C}^J$ consisting of eigenvectors of S, then there is an invertible matrix M such that $\|S\|_M = \rho(S)$.

Ex. 6.16 *Show that, if $\rho(S) < 1$, then there is a vector norm on $\mathbb{C}^J$ for which the induced matrix norm of S is less than one.*

Ex. 6.17 *Show that $\rho(S) < 1$ if and only if $\lim_{k\to\infty} S^k = 0$.*

Definition 6.12 *Let A be an arbitrary matrix. Denote by $|A|$ the matrix whose entries are the absolute values of those of A, that is, $|A|_{ij} = |A_{ij}|$.*

Proposition 6.2 *Let A and B be J by J real matrices. If $|A|_{ij} \leq B_{ij}$ for all i and j, then $\rho(A) \leq \rho(B)$.*

Proof: Let $\sigma = \rho(B)$ and $\epsilon > 0$ be arbitrary. Let $B_1 = (\sigma+\epsilon)^{-1}B$ and $A_1 = (\sigma+\epsilon)^{-1}A$. Then $\rho(B_1) < 1$, so that $B_1^k \to 0$, as $k \to \infty$. Therefore, $A_1^k \to 0$ also. From Exercise 6.17 we can conclude that $\rho(A_1) < 1$. Therefore, $\rho(A) < \sigma + \epsilon$. Since ϵ is arbitrary, it follows that $\rho(A) \leq \sigma = \rho(B)$. ∎

Corollary 6.1 *For any square matrix A we have $\rho(A) \leq \rho(|A|)$.*

6.8.2 Some Examples of Induced Matrix Norms

If we choose the two vector norms carefully, then we can get an explicit description of $\|A\|$, but, in general, we cannot.

For example, let $\|x\| = \|x\|_1$ and $\|Ax\| = \|Ax\|_1$ be the 1-norms of the vectors x and Ax, where

$$\|x\|_1 = \sum_{j=1}^{J} |x_j|. \tag{6.25}$$

Lemma 6.4 *The 1-norm of A, induced by the 1-norms of vectors in $\mathbb{C}^J$ and $\mathbb{C}^I$, is*

$$\|A\|_1 = \max\{\sum_{i=1}^{I} |A_{ij}|, j = 1, 2, ..., J\}. \tag{6.26}$$

Proof: Use basic properties of the absolute value to show that

$$\|Ax\|_1 \leq \sum_{j=1}^{J}(\sum_{i=1}^{I} |A_{ij}|)|x_j|. \tag{6.27}$$

Then let $j = m$ be the index for which the maximum column sum is reached and select $x_j = 0$, for $j \neq m$, and $x_m = 1$. ∎

The *infinity norm* of the vector x is

$$\|x\|_\infty = \max\{|x_j|, j = 1, 2, ..., J\}. \tag{6.28}$$

Lemma 6.5 *The infinity norm of the matrix A, induced by the infinity norms of vectors in $\mathbb{R}^J$ and $\mathbb{C}^I$, is*

$$\|A\|_\infty = \max\{\sum_{j=1}^{J} |A_{ij}|, i = 1, 2, ..., I\}. \tag{6.29}$$

The proof is similar to that of the previous lemma.

From these two lemmas we learn that

$$\|A^\dagger\|_1 = \|A\|_\infty,$$

and

$$\|A^\dagger\|_\infty = \|A\|_1.$$

6.8.3 The Two-Norm of a Matrix

We shall be particularly interested in the two-norm (or 2-norm) of a matrix A, denoted by $\|A\|_2$, which is the induced matrix norm derived from the Euclidean vector norms.

From the definition of the two-norm of A, we know that

$$\|A\|_2 = \max\{\|Ax\|_2/\|x\|_2\}, \tag{6.30}$$

with the maximum over all nonzero vectors x. Since

$$\|Ax\|_2^2 = x^\dagger A^\dagger A x, \tag{6.31}$$

we have

$$\|A\|_2 = \sqrt{\max\{\frac{x^\dagger A^\dagger A x}{x^\dagger x}\}}, \tag{6.32}$$

over all nonzero vectors x.

Proposition 6.3 *The two-norm of a matrix A is*

$$\|A\|_2 = \sqrt{\rho(A^\dagger A)}; \tag{6.33}$$

that is, the term inside the square-root in Equation (6.32) is the largest eigenvalue of the matrix $A^\dagger A$.

Proof: Let

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_J \geq 0 \tag{6.34}$$

be the eigenvalues of $A^\dagger A$, and let $\{u^j, j = 1, ..., J\}$ be the associated

mutually orthogonal eigenvectors of $A^\dagger A$ with $\|u^j\|_2 = 1$. Then, for any x, we have

$$x = \sum_{j=1}^{J} [(u^j)^\dagger x] u^j, \tag{6.35}$$

while

$$A^\dagger A x = \sum_{j=1}^{J} [(u^j)^\dagger x] A^\dagger A u^j = \sum_{j=1}^{J} \lambda_j [(u^j)^\dagger x] u^j. \tag{6.36}$$

It follows that

$$\|x\|_2^2 = x^\dagger x = \sum_{j=1}^{J} |(u^j)^\dagger x|^2, \tag{6.37}$$

and

$$\|Ax\|_2^2 = x^\dagger A^\dagger A x = \sum_{j=1}^{J} \lambda_j |(u^j)^\dagger x|^2. \tag{6.38}$$

Maximizing $\|Ax\|_2^2 / \|x\|_2^2$ over $x \neq 0$ is equivalent to maximizing $\|Ax\|_2^2$, subject to $\|x\|_2^2 = 1$. The right side of Equation (6.38) is then a convex combination of the λ_j, which will have its maximum when only the coefficient of λ_1 is non-zero. ∎

Ex. 6.18 *Show that $\|A\|_2 = \|A^\dagger\|_2$ for any matrix A. Hints: use Exercise 5.12 and Proposition 6.3.*

Note that it can be shown ([7], p. 164) that for any square matrix S and any matrix norm we have

$$\rho(S) = \lim_{n \to \infty} (\|S^n\|)^{1/n}.$$

6.8.4 The Two-Norm of an Hermitian Matrix

Let H be an Hermitian matrix. We then have the following result:

Proposition 6.4 *The two-norm of H is $\|H\|_2 = \rho(H)$.*

Ex. 6.19 *Prove Proposition 6.4. Hint: use $H^\dagger H = H^2$ and Exercise 3.40.*

Using Proposition 6.4, we can prove the following theorem.

Theorem 6.1 *For any matrix A we have the inequality*

$$\|A\|_2^2 \leq \|A\|_1 \|A\|_\infty. \tag{6.39}$$

Proof: Let $H = A^\dagger A$. We know that $\|A\|_2^2 = \|H\|_2$ and that

$$\|H\|_2 \leq \|H\|_1 = \|A^\dagger A\|_1 \leq \|A^\dagger\|_1 \|A\|_1 = \|A\|_\infty \|A\|_1.$$

∎

The inequality (6.39) also follows, as a particular case, from the more general Theorem 15.5 concerning upper bounds for the singular values of a matrix A.

Ex. 6.20 *Show that if the rows of the matrix A are rescaled so that, for each i, we have $\sum_{j=1}^J |A_{ij}| \leq 1$, then no eigenvalue of $A^\dagger A$ is larger than the maximum number of non-zero entries in any column of A. In Corollary 15.2 we shall see that the same conclusion holds if the rows of A are rescaled to have Euclidean length not greater than one.*

Ex. 6.21 *Show that $\sum_{j=1}^J |A_{i,j}| \leq 1$ implies that $\sum_{j=1}^J |A_{i,j}|^2 \leq 1$. On the other hand, if $\sum_{j=1}^J |A_{i,j}|^2 \leq 1$, it does not follow that $\sum_{j=1}^J |A_{i,j}| \leq 1$. How large can $\sum_{j=1}^J |A_{i,j}|$ be, if $\sum_{j=1}^J |A_{i,j}|^2 \leq 1$?*

If S is not Hermitian, then the two-norm of S cannot be calculated directly from the eigenvalues of S. Take, for example, the square, non-Hermitian matrix

$$S = \begin{bmatrix} i & 2 \\ 0 & i \end{bmatrix}, \tag{6.40}$$

having eigenvalues $\lambda = i$ and $\lambda = i$. The eigenvalues of the Hermitian matrix

$$S^\dagger S = \begin{bmatrix} 1 & -2i \\ 2i & 5 \end{bmatrix} \tag{6.41}$$

are $\lambda = 3 + 2\sqrt{2}$ and $\lambda = 3 - 2\sqrt{2}$. Therefore, the two-norm of S is

$$\|S\|_2 = \sqrt{3 + 2\sqrt{2}}. \tag{6.42}$$

6.8.5 The p-norm of a Matrix

The p-norm of an I by J complex matrix A is the norm induced by the p-norms on the vectors in $\mathbb{C}^I$ and $\mathbb{C}^J$; we can say that $\|A\|_p$ is the maximum of $\|Ax\|_p$, over all x with $\|x\|_p = 1$.

Previously, we were able to use the explicit descriptions of $\|A\|_1$ and $\|A\|_\infty$ to show that $\|A^\dagger\|_1 = \|A\|_\infty$. A similar result holds for the p-norm.

Theorem 6.2 *Let $\frac{1}{p} + \frac{1}{q} = 1$. Then*

$$\|A^\dagger\|_p = \|A\|_q.$$

Proof: We select a vector x with $\|x\|_p = 1$. We then construct the vector v with

$$|v_i|^q = |(Ax)_i|^p / \|Ax\|_p^p,$$

and such that v_i and $(Ax)_i$ have the same phase angles. Then $\|v\|_q = 1$. It follows that

$$\sum_{i=1}^{I} \overline{(Ax)_i} v_i = \|Ax\|_p.$$

We also have

$$\sum_{i=1}^{I} \overline{(Ax)_i} v_i = \sum_{j=1}^{J} \overline{x_j} (A^\dagger v)_j,$$

so that

$$\|Ax\|_p = \sum_{j=1}^{J} \overline{x_j} (A^\dagger v)_j \leq \|x\|_p \|A^\dagger v\|_q.$$

It then follows that the maximum of $\|Ax\|_p$, over all x with $\|x\|_p = 1$, is not greater than the maximum of $\|A^\dagger v\|_q$, over all v with $\|v\|_q = 1$. Since this is true for all A, the theorem follows. ∎

We can use Theorem 6.2 to prove *Young's Inequality*.

Theorem 6.3 (Young's Inequality) *For any complex matrix A we have*

$$\|A\|_2^2 \leq \|A\|_p \|A\|_q. \tag{6.43}$$

Proof: We know that $\rho(S) \leq \|S\|$, for all square matrices S and all induced matrix norms. Also, for $S = H$ Hermitian, we have $\rho(H) = \|H\|_2$, from which we conclude that $\|H\|_2 \leq \|H\|$, for all induced matrix norms. Now we let $H = A^\dagger A$.

From $\|A\|_2^2 = \|H\|_2$, we have

$$\|A\|_2^2 = \sqrt{\|H\|_2^2} = \sqrt{\|H\|_2 \|H\|_2} \leq \sqrt{\|H\|_p \|H\|_q}.$$

Since

$$\|H\|_p = \|A^\dagger A\|_p \leq \|A^\dagger\|_p \|A\|_p = \|A\|_q \|A\|_p,$$

it follows that

$$\|A\|_2^2 \leq \|A\|_p \|A\|_q.$$

∎

6.8.6 Using Diagonalizable Matrices

When S is diagonalizable, we let U be a square matrix whose columns are J linearly independent eigenvectors of S and L the diagonal matrix having the eigenvalues of S along its main diagonal; then we have $SU = UL$, or $U^{-1}SU = L$.

Ex. 6.22 *Let $M = U^{-1}$ and define $\|x\|_M = \|Mx\|_2$, the Euclidean norm of Mx. Show that the induced matrix norm of S is $\|S\|_M = \rho(S)$.*

We see from this exercise that, for any diagonalizable matrix S, in particular, for any Hermitian matrix, there is a vector norm such that the induced matrix norm of S is $\rho(S)$.

In the Hermitian case $S = H$, we know that we can select the eigenvector columns of U to be mutually orthogonal and scaled to have length one, so that $U^{-1} = U^\dagger$ and $\|Mx\|_2 = \|U^\dagger x\|_2 = \|x\|_2$, so that the required vector norm is just the Euclidean norm, and $\|H\|_M$ is just $\|H\|_2$, which we know to be $\rho(H)$.

Ex. 6.23 *The* Cayley-Hamilton Theorem *asserts that if S is any square matrix and $P(\lambda)$ its characteristic polynomial, then $P(S) = 0$. Prove this for the case of diagonalizable S.*

6.9 Estimating Eigenvalues

Calculating the eigenvalues of a square matrix amounts to solving for the roots of a polynomial. In general, this requires an iterative procedure, since there are no algebraic formulas for finding the roots of arbitrary polynomials. In this section we give two simply methods for obtaining somewhat crude estimates of the eigenvalues. Later, we shall present better estimation methods.

6.9.1 Using the Trace

The trace of a square matrix S, written trace(S) or tr(S), is the sum of the entries on the main diagonal of S. If S is diagonalizable, then we can write $S = ULU^{-1}$, where L is the diagonal matrix whose diagonal entries are the eigenvalues of S. For any square matrices A, B, and C we have

$$\text{tr}(ABC) = \text{tr}(CAB) = \text{tr}(BCA),$$

but these are not necessarily equal to tr(BAC). Therefore,

$$\text{tr}(S) = \text{tr}(ULU^{-1}) = \text{tr}(U^{-1}UL) = \text{tr}(L),$$

so that the trace of S is the sum of its eigenvalues. The same result holds for non-diagonalizable matrices, but the proof is a bit harder; try to prove this using Schur's Lemma 5.1.

6.9.2 Gerschgorin's Theorem

Gerschgorin's theorem gives us a way to estimate the eigenvalues of an arbitrary square matrix S.

Theorem 6.4 *Let S be J by J. For $j = 1, ..., J$, let C_j be the circle in the complex plane with center S_{jj} and radius $r_j = \sum_{m \neq j} |S_{jm}|$. Then every eigenvalue of S lies within one of the C_j.*

Proof: Let λ be an eigenvalue of S, with associated eigenvector u. Let u_j be the entry of the vector u having the largest absolute value. From $Su = \lambda u$, we have

$$(\lambda - S_{jj})u_j = \sum_{m \neq j} S_{jm}u_m, \tag{6.44}$$

so that

$$|\lambda - S_{jj}| \leq \sum_{m \neq j} |S_{jm}||u_m|/|u_j| \leq r_j. \tag{6.45}$$

This completes the proof. ∎

6.9.3 Strictly Diagonally Dominant Matrices

Definition 6.13 *A square J by J matrix S is said to be* strictly diagonally dominant *if, for each $j = 1, ..., J$,*

$$|S_{jj}| > r_j = \sum_{m \neq j} |S_{jm}|. \tag{6.46}$$

When the matrix S is strictly diagonally dominant, all the eigenvalues of S lie within the union of the spheres with centers S_{jj} and radii S_{jj}. With D the diagonal component of S, the matrix $D^{-1}S$ then has all its eigenvalues within the circle of radius one, centered at $(1, 0)$. Then $\rho(I - D^{-1}S) < 1$. This result is used when we discuss splitting methods in Chapter 10 (see also [66]).

6.10 Conditioning

Let S be a square, invertible matrix and z the solution to $Sz = h$. We are concerned with the extent to which the solution changes as the right side, h, changes. Denote by δ_h a small perturbation of h, and by δ_z the solution of $S\delta_z = \delta_h$. Then $S(z + \delta_z) = h + \delta_h$. Applying the compatibility condition $\|Ax\| \leq \|A\|\|x\|$, we get

$$\|\delta_z\| \leq \|S^{-1}\|\|\delta_h\|, \tag{6.47}$$

and

$$\|z\| \geq \|h\|/\|S\|. \tag{6.48}$$

Therefore

$$\frac{\|\delta_z\|}{\|z\|} \leq \|S\|\,\|S^{-1}\|\frac{\|\delta_h\|}{\|h\|}. \tag{6.49}$$

Definition 6.14 *The quantity $c = \|S\|\|S^{-1}\|$ is the* condition number *of S, with respect to the given matrix norm.*

Note that $c \geq 1$: for any non-zero z, we have

$$1 = \|I\| = \|SS^{-1}\| \leq \|S\|\|S^{-1}\|. \tag{6.50}$$

Ex. 6.24 *Show that when Q is Hermitian and positive-definite, the condition number of Q, with respect to the matrix norm induced by the Euclidean vector norm, is*

$$c = \lambda_{max}(Q)/\lambda_{min}(Q), \tag{6.51}$$

the ratio of the largest to the smallest eigenvalues of Q.

Chapter 7

Under-Determined Systems of Linear Equations

7.1 Chapter Summary

When a system of M linear equations in N unknowns, denoted $Ax = b$, has multiple solutions, we say that the system is *under-determined*. Then it has infinitely many solutions; if $Ax = b$ and $Az = b$ and $x \neq z$, then $x + \alpha(z - x)$ is also a solution, for any scalar α. In such cases, we usually select one solution out of the infinitely many possibilities by requiring that the solution also satisfy some additional constraints. For example, we can select that solution x for which $\|x\|_2$ is minimized, which we denote by $\hat{x}$. This *minimum two-norm* solution is given by

$$\hat{x} = A^\dagger (AA^\dagger)^{-1} b,$$

provided that the matrix $AA^\dagger$ has an inverse. In this chapter we survey several of the constraints that are commonly used and the algorithms that are employed to calculate these constrained solutions.

7.2 Minimum Two-Norm Solutions

When the system $Ax = b$ is under-determined, it is reasonable to ask for that solution $x = \hat{x}$ having the smallest two-norm

$$\|x\|_2 = \sqrt{\sum_{n=1}^{N} |x_n|^2}.$$

As we showed previously, the *minimum two-norm* solution of $Ax = b$ is a vector of the form $\hat{x} = A^\dagger z$. Then $A\hat{x} = b$ becomes $AA^\dagger z = b$. Typically, $(AA^\dagger)^{-1}$ will exist, and we get $z = (AA^\dagger)^{-1}b$, from which it follows that the minimum two-norm solution is $\hat{x} = A^\dagger(AA^\dagger)^{-1}b$. When M and N are not too large, forming the matrix $AA^\dagger$ and solving for z is not prohibitively expensive or time-consuming.

When M and N are large, we turn to iterative algorithms to find the minimum two-norm solution. Both the ART and the Landweber algorithm converge to that solution closest to the starting vector x^0, in the two-norm sense. Therefore, when we begin with $x^0 = 0$, these algorithms give us the minimum two-norm solution.

If C is a closed convex set in $\mathbb{R}^N$, the *projected Landweber algorithm* converges to that solution x in C closest to x^0, in the two-norm sense. Again, if we take $x^0 = 0$, the projected Landweber algorithm converges to that solution x in C having the smallest two-norm.

7.3 Minimum Weighted Two-Norm Solutions

The *minimum weighted two-norm solution* is the $x = \tilde{x}$ satisfying $Ax = b$ for which the weighted two-norm

$$\|x\|_w = \sqrt{\sum_{n=1}^{N} |x_n|^2 w_n}$$

is minimized. This solution can be found easily by changing variables, letting $u_n = x_n\sqrt{w_n}$, to convert the problem into a minimum two-norm

problem, and then applying any of the methods discussed in the previous chapter. The minimum weighted two-norm approach is a discrete version of a method, called the PDFT, for estimating a function from values of its Fourier transform [43].

Figure 2.2 illustrates the potential advantages to be obtained through the use of weights. In that example, we have a prior estimate of the magnitudes of the x_n, which we called $p_n > 0$. Then we chose for the weights $w_n = p_n^{-1}$.

7.4 Minimum One-Norm Solutions

Instead of the minimum two-norm solution, we can seek a *minimum one-norm* solution, that is, minimize

$$||x||_1 = \sum_{n=1}^{N} |x_n|,$$

subject to $Ax = b$; we denote by x^* the minimum one-norm solution. As we shall see, this problem can be formulated as a linear programming problem, so is easily solved.

The entries of x need not be nonnegative, so the problem is not yet a linear programming problem. Let

$$B = \begin{bmatrix} A & -A \end{bmatrix},$$

and consider the linear programming problem of minimizing the function

$$c^T z = \sum_{n=1}^{2N} z_n,$$

subject to the constraints $z \geq 0$, and $Bz = b$. Let z^* be the solution. We write

$$z^* = \begin{bmatrix} u^* \\ v^* \end{bmatrix}.$$

Then, as we shall see, $x^* = u^* - v^*$ minimizes the one-norm, subject to $Ax = b$.

First, we show that $u_n^* v_n^* = 0$, for each n. If this were not the case and there is an n such that $0 < v_n^* < u_n^*$, then we can create a new vector z by replacing the old u_n^* with $u_n^* - v_n^*$ and the old v_n^* with zero, while maintaining $Bz = b$. But then, since $u_n^* - v_n^* < u_n^* + v_n^*$, it follows that $c^T z < c^T z^*$, which is a contradiction. Consequently, we have $\|x^*\|_1 = c^T z^*$.

Now we select any x with $Ax = b$. Write $u_n = x_n$, if $x_n \geq 0$, and $u_n = 0$, otherwise. Let $v_n = u_n - x_n$, so that $x = u - v$. Then let

$$z = \begin{bmatrix} u \\ v \end{bmatrix}.$$

Then $b = Ax = Bz$, and $c^T z = \|x\|_1$. And so,

$$\|x^*\|_1 = c^T z^* \leq c^T z = \|x\|_1,$$

and x^* must be a minimum one-norm solution.

Ex. 7.1 *Find a system of linear equations $Ax = b$ for which there are multiple minimum one-norm solutions.*

7.5 Sparse Solutions

For any vector x, we define the *support* of x to be the subset S of $\{1, 2, ..., N\}$ consisting of those n for which the entries $x_n \neq 0$. For any under-determined system $Ax = b$, there will, of course, be at least one solution, call it x', of minimum support, that is, for which $|S|$, the size of the support set S, is minimum. However, finding such a maximally sparse solution requires combinatorial optimization, and is known to be computationally difficult. It is important, therefore, to have a computationally tractable method for finding maximally sparse solutions.

7.5.1 Maximally Sparse Solutions

Consider the following problem: among all solutions x of the consistent system $Ax = b$, find one, x', that is maximally sparse, that is, has the minimum number of non-zero entries. Obviously, there will be at least one such solution having minimal support, but finding one, however, is a combinatorial optimization problem and is generally NP-hard.

7.5.2 Why the One-Norm?

When a system of linear equations $Ax = b$ is under-determined, we can find the *minimum two-norm solution*. One drawback to this approach is that the two-norm penalizes relatively large values of x_n much more than the smaller ones, so tends to provide non-sparse solutions. Alternatively, we may seek the minimum one-norm solution. The one-norm still penalizes

relatively large entries x_n more than the smaller ones, but much less so than the two-norm does. As a result, it often happens that the minimum one-norm solution actually is a maximally sparse solution, as well.

7.5.3 Comparison with the Weighted Two-Norm Solution

Our intention is to select weights w_n so that w_n^{-1} is reasonably close to the absolute value of the corresponding entry of the minimum one-norm solution $|x_n^*|$; consider, therefore, what happens when $w_n^{-1} = |x_n^*|$. We claim that $\tilde{x}$ is also a minimum-one-norm solution.

To see why this is true, note that, for any x, we have

$$\sum_{n=1}^{N} |x_n| = \sum_{n=1}^{N} \frac{|x_n|}{\sqrt{|x_n^*|}} \sqrt{|x_n^*|}$$

$$\leq \sqrt{\sum_{n=1}^{N} \frac{|x_n|^2}{|x_n^*|}} \sqrt{\sum_{n=1}^{N} |x_n^*|}.$$

Therefore,

$$\sum_{n=1}^{N} |\tilde{x}_n| \leq \sqrt{\sum_{n=1}^{N} \frac{|\tilde{x}_n|^2}{|x_n^*|}} \sqrt{\sum_{n=1}^{N} |x_n^*|}$$

$$\leq \sqrt{\sum_{n=1}^{N} \frac{|x_n^*|^2}{|x_n^*|}} \sqrt{\sum_{n=1}^{N} |x_n^*|} = \sum_{n=1}^{N} |x_n^*|.$$

Therefore, $\tilde{x}$ is also a solution that minimizes the one-norm. If x^* is unique, then $\tilde{x} = x^*$.

7.5.4 Iterative Reweighting

Let x be the truth. generally, we want each weight w_n to be a good prior estimate of the reciprocal of $|x_n|$. Because we do not yet know x, we may take a sequential-optimization approach, beginning with weights $w_n^0 > 0$, finding the minimum weighted two-norm solution using these weights, then using this solution to get a (we hope!) better choice for the weights, and so on. This sequential approach was successfully implemented in the early 1980's by Michael Fiddy and his students [137].

In [76], the same approach is taken, but with respect to the one-norm. Since the one-norm still penalizes larger values disproportionately, balance can be achieved by minimizing a weighted one-norm, with weights close to the reciprocals of the $|x_n|$. Again, not yet knowing x, they employ a sequential approach, using the previous minimum weighted one-norm solution to

obtain the new set of weights for the next minimization. At each step of the sequential procedure, the previous reconstruction is used to estimate the true support of the desired solution.

It is interesting to note that an on-going debate among users of the minimum weighted two-norm approach concerns the nature of the prior weighting. With x denoting the truth, does w_n approximate $|x_n|$ or $|x_n|^2$? This is close to the issue treated in [76], the use of a weight in the minimum one-norm approach.

It should be noted again that finding a sparse solution is not usually the goal in the use of the minimum weighted two-norm approach, but the use of the weights has much the same effect as using the one-norm to find sparse solutions: to the extent that the weights approximate the entries of x^*, their use reduces the penalty associated with the larger entries of an estimated solution.

7.6 Why Sparseness?

One obvious reason for wanting sparse solutions of $Ax = b$ is that we have prior knowledge that the desired solution is sparse. Such a problem arises in signal analysis from Fourier-transform data. In other cases, such as in the reconstruction of locally constant signals, it is not the signal itself, but its discrete derivative, that is sparse.

7.6.1 Signal Analysis

Suppose that our signal $f(t)$ is known to consist of a small number of complex exponentials, so that $f(t)$ has the form

$$f(t) = \sum_{j=1}^{J} a_j e^{i\omega_j t},$$

for some small number of frequencies ω_j in the interval $[0, 2\pi)$. For $n = 0, 1, ..., N-1$, let $f_n = f(n)$, and let f be the N-vector with entries f_n; we assume that J is much smaller than N. The discrete (vector) Fourier transform of f is the vector $\hat{f}$ having the entries

$$\hat{f}_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f_n e^{2\pi ikn/N},$$

for $k = 0, 1, ..., N-1$; we write $\hat{f} = Ef$, where E is the N by N matrix with entries $E_{kn} = \frac{1}{\sqrt{N}} e^{2\pi ikn/N}$. If N is large enough, we may safely assume that

each of the ω_j is equal to one of the frequencies $2\pi ik$ and that the vector $\hat{f}$ is J-sparse. The question now is: How many values of $f(n)$ do we need to calculate in order to be sure that we can recapture $f(t)$ exactly? We have the following theorem [75]:

Theorem 7.1 *Let N be prime. Let S be any subset of $\{0, 1, ..., N-1\}$ with $|S| \geq 2J$. Then the vector $\hat{f}$ can be uniquely determined from the measurements f_n for n in S.*

We know that

$$f = E^\dagger \hat{f},$$

where $E^\dagger$ is the conjugate transpose of the matrix E. The point here is that, for any matrix R obtained from the identity matrix I by deleting $N - |S|$ rows, we can recover the vector $\hat{f}$ from the measurements Rf.

If N is not prime, then the assertion of the theorem may not hold, since we can have $n = 0 \bmod N$, without $n = 0$. However, the assertion remains valid for most sets of J frequencies and most subsets S of indices; therefore, with high probability, we can recover the vector $\hat{f}$ from Rf.

Note that the matrix E is *unitary*, that is, $E^\dagger E = I$, and, equivalently, the columns of E form an orthonormal basis for $\mathbb{C}^N$. The data vector is

$$b = Rf = RE^\dagger \hat{f}.$$

In this example, the vector f is not sparse, but can be represented sparsely in a particular orthonormal basis, namely as $f = E^\dagger \hat{f}$, using a sparse vector $\hat{f}$ of coefficients. The *representing basis* then consists of the columns of the matrix $E^\dagger$. The measurements pertaining to the vector f are the values f_n, for n in S. Since f_n can be viewed as the inner product of f with δ^n, the nth column of the identity matrix I, that is,

$$f_n = \langle \delta^n, f \rangle,$$

the columns of I provide the so-called *sampling basis*. With $A = RE^\dagger$ and $x = \hat{f}$, we then have

$$Ax = b,$$

with the vector x sparse. It is important for what follows to note that the matrix A is random, in the sense that we choose which rows of I to use to form R.

7.6.2 Locally Constant Signals

Suppose now that the function $f(t)$ is locally constant, consisting of some number of horizontal lines. We discretize the function $f(t)$ to get

the vector $f = (f(0), f(1), ..., f(N))^T$. The discrete derivative vector is $g = (g_1, g_2, ..., g_N)^T$, with

$$g_n = f(n) - f(n-1).$$

Since $f(t)$ is locally constant, the vector g is sparse. The data we will have will not typically be values $f(n)$. The goal will be to recover f from M linear functional values pertaining to f, where M is much smaller than N. We shall assume, from now on, that we have measured, or can estimate, the value $f(0)$.

Our M by 1 data vector d consists of measurements pertaining to the vector f:

$$d_m = \sum_{n=0}^{N} H_{mn} f_n,$$

for $m = 1, ..., M$, where the H_{mn} are known. We can then write

$$d_m = f(0)\Big(\sum_{n=0}^{N} H_{mn}\Big) + \sum_{k=1}^{N}\Big(\sum_{j=k}^{N} H_{mj}\Big) g_k.$$

Since $f(0)$ is known, we can write

$$b_m = d_m - f(0)\Big(\sum_{n=0}^{N} H_{mn}\Big) = \sum_{k=1}^{N} A_{mk} g_k,$$

where

$$A_{mk} = \sum_{j=k}^{N} H_{mj}.$$

The problem is then to find a sparse solution of $Ax = g$. As in the previous example, we often have the freedom to select the linear functions, that is, the values H_{mn}, so the matrix A can be viewed as random.

7.6.3 Tomographic Imaging

The reconstruction of tomographic images is an important aspect of medical diagnosis, and one that combines aspects of both of the previous examples. The data one obtains from the scanning process can often be interpreted as values of the Fourier transform of the desired image; this is precisely the case in magnetic-resonance imaging, and approximately true for x-ray transmission tomography, positron-emission tomography (PET) and single-photon emission tomography (SPECT). The images one encounters in medical diagnosis are often approximately locally constant, so the associated array of discrete partial derivatives will be sparse. If this sparse derivative array can be recovered from relatively few Fourier-transform values, then the scanning time can be reduced.

7.7 Positive Linear Systems

When the entries of the matrix A are nonnegative, the entries of the vector b are positive, and we require that the entries of x be nonnegative, we say that we have a *positive system*. We call the system *under-determined* when there are multiple nonnegative solutions. It is appropriate now to use the cross-entropy, or Kullback-Leibler (KL), distance between nonnegative vectors, rather than the two-norm or the one-norm.

In the under-determined case, the MART and its block-iterative versions, the RBI-SMART algorithms, all converge to that nonnegative solution x for which $KL(x, x^0)$ is minimized. The EMML algorithm and its block-iterative variants also converge to nonnegative solutions, but they may not all be the same solution, and no explicit characterization of these solutions is known; that is, they depend on x^0, but precisely how is not known. When we wish to impose further constraints on the entries of x, we can use the ABMART or the ABEMML algorithms. See [51], [56] and [57] for details.

7.8 Feasible-Point Methods

In previous sections we considered the minimum two-norm and minimum one-norm solutions for under-determined systems $Ax = b$. A more general approach is to minimize some function $f(x)$, subject to $Ax = b$, which is the subject of this section.

We consider now the problem of minimizing the function $f(x) : \mathbb{R}^N \rightarrow \mathbb{R}$, subject to the equality constraints $Ax = b$, where A is an M by N real matrix, with rank M and $M < N$. The two methods we consider here are *feasible-point methods*, also called *interior-point methods*.

7.8.1 The Reduced Newton-Raphson Method

The first method we consider is a modification of the Newton-Raphson method, in which we begin with a feasible point and each NR step is projected into the null space of the matrix A, to maintain the condition $Ax = b$. The discussion here is taken from [211].

Let $\hat{x}$ be a *feasible point*, that is, $A\hat{x} = b$. Then $x = \hat{x} + p$ is also feasible if p is in the null space of A, that is, $Ap = 0$. Let Z be an N by $N - M$ matrix whose columns form a basis for the null space of A. We want $p = Zv$

for some v. The best v will be the one for which the function

$$\phi(v) = f(\hat{x} + Zv)$$

is minimized. We can apply to the function $\phi(v)$ the steepest descent method, or Newton-Raphson or any other minimization technique. The steepest descent method, applied to $\phi(v)$, is called the *reduced steepest descent method*; the Newton-Raphson method, applied to $\phi(v)$, is called the *reduced Newton-Raphson method*. The gradient of $\phi(v)$, also called the *reduced gradient*, is

$$\nabla\phi(v) = Z^T \nabla f(x),$$

and the Hessian matrix of $\phi(v)$, also called the *reduced Hessian matrix*, is

$$\nabla^2\phi(v) = Z^T \nabla^2 f(x) Z,$$

where $x = \hat{x} + Zv$, so algorithms to minimize $\phi(v)$ can be written in terms of the gradient and Hessian of f itself.

The reduced NR algorithm can then be viewed in terms of the vectors $\{v^k\}$, with $v^0 = 0$ and

$$v^{k+1} = v^k - [\nabla^2\phi(v^k)]^{-1}\nabla\phi(v^k); \tag{7.1}$$

the corresponding x^k is

$$x^k = \hat{x} + Zv^k.$$

7.8.1.1 An Example

Consider the problem of minimizing the function

$$f(x) = \frac{1}{2}x_1^2 - \frac{1}{2}x_3^2 + 4x_1x_2 + 3x_1x_3 - 2x_2x_3,$$

subject to

$$x_1 - x_2 - x_3 = -1.$$

Let $\hat{x} = [1, 1, 1]^T$. Then the matrix A is $A = [1, -1, -1]$ and the vector b is $b = [-1]$. Let the matrix Z be

$$Z = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{7.2}$$

The reduced gradient at $\hat{x}$ is then

$$Z^T \nabla f(\hat{x}) = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 8 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 10 \\ 8 \end{bmatrix}, \tag{7.3}$$

and the reduced Hessian matrix at $\hat{x}$ is

$$Z^T \nabla^2 f(\hat{x}) Z = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 4 & 3 \\ 4 & 0 & -2 \\ 3 & -2 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 9 & 6 \\ 6 & 6 \end{bmatrix}. \tag{7.4}$$

Then the reduced Newton-Raphson equation yields

$$v = \begin{bmatrix} -2/3 \\ -2/3 \end{bmatrix}, \tag{7.5}$$

and the reduced Newton-Raphson direction is

$$p = Zv = \begin{bmatrix} -4/3 \\ -2/3 \\ -2/3 \end{bmatrix}. \tag{7.6}$$

Since the function $\phi(v)$ is quadratic, one reduced Newton-Raphson step suffices to obtain the solution, $x^* = [-1/3, 1/3, 1/3]^T$.

7.8.2 A Primal-Dual Approach

Once again, the objective is to minimize the function $f(x) : \mathbb{R}^N \to \mathbb{R}$, subject to the equality constraints $Ax = b$. According to the Karush-Kuhn-Tucker Theorem [71], $\nabla L(x, \lambda) = 0$ at the optimal values of x and λ, where the Lagrangian $L(x, \lambda)$ is

$$L(x, \lambda) = f(x) + \lambda^T (b - Ax).$$

Finding a zero of the gradient of $L(x, \lambda)$ means that we have to solve the equations

$$\nabla f(x) - A^T \lambda = 0$$

and

$$Ax = b.$$

We define the function $G(x, \lambda)$ taking values in $\mathbb{R}^N \times \mathbb{R}^M$ to be

$$G(x, \lambda) = (\nabla f(x) - A^T \lambda, Ax - b)^T.$$

We then apply the NR method to find a zero of the function G. The Jacobian matrix for G is

$$J_G(x, \lambda) = \begin{bmatrix} \nabla^2 f(x) & -A^T \\ A & 0 \end{bmatrix},$$

so one step of the NR method is

$$(x^{k+1}, \lambda^{k+1})^T = (x^k, \lambda^k)^T - J_G(x^k, \lambda^k)^{-1} G(x^k, \lambda^k). \tag{7.7}$$

We can rewrite this as

$$\nabla^2 f(x^k)(x^{k+1} - x^k) - A^T(\lambda^{k+1} - \lambda^k) = A^T\lambda^k - \nabla f(x^k), \tag{7.8}$$

and

$$A(x^{k+1} - x^k) = b - Ax^k. \tag{7.9}$$

It follows from Equation (7.9) that $Ax^{k+1} = b$, for $k = 0, 1, \ldots$, so that this primal-dual algorithm is a feasible-point algorithm.

Chapter 8

The LU and QR Factorizations

8.1 Chapter Summary

Two important methods for solving the system $Sx = b$, the LU factorization and the QR factorization, involve factoring the square matrix S and thereby reducing the problem to finding the solutions of simpler systems.

In the LU factorization, we seek a lower triangular matrix L and an upper triangular matrix U so that $S = LU$. We then solve $Sx = b$ by solving $Lz = b$ and $Ux = z$.

In the QR factorization, we seek an orthogonal matrix Q, that is, $Q^T = Q^{-1}$, and an upper triangular matrix R so that $S = QR$. Then we solve $Sx = b$ by solving the upper triangular system $Rx = Q^T b$.

In this chapter we investigate these two methods.

8.2 The LU Factorization

The matrix

$$S = \begin{bmatrix} 2 & 1 & 1 \\ 4 & 1 & 0 \\ -2 & 2 & 1 \end{bmatrix}$$

can be reduced to the upper triangular matrix

$$U = \begin{bmatrix} 2 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & -4 \end{bmatrix}$$

through three elementary row operations: first, add -2 times the first row to the second row; second, add the first row to the third row; finally, add three times the new second row to the third row. Each of these row operations can be viewed as the result of multiplying on the left by the matrix obtained by applying the same row operation to the identity matrix. For example, adding -2 times the first row to the second row can be achieved by multiplying A on the left by the matrix

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix};$$

note that the inverse of L_1 is

$$L_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

We can write

$$L_3 L_2 L_1 S = U,$$

where L_1, L_2, and L_3 are the matrix representatives of the three elementary row operations. Therefore, we have

$$S = L_1^{-1} L_2^{-1} L_3^{-1} U = LU.$$

This is the *LU factorization* of S. As we just saw, the LU factorization can be obtained along with the Gauss elimination.

8.2.1 A Shortcut

There is a shortcut we can take in calculating the LU factorization. We begin with the identity matrix I, and then, as we perform a row operation,

for example, adding -2 times the first row to the second row, we put the number 2, the multiplier just used, but with a sign change, in the second row, first column, the position of the entry of S that was just converted to zero. Continuing in this fashion, we build up the matrix L as

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -3 & 1 \end{bmatrix},$$

so that

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

The entries of the main diagonal of L will be all ones. If we want the same to be true of U, we can rescale the rows of U and obtain the factorization $S = LDU$, where D is a diagonal matrix.

8.2.2 A Warning!

We have to be careful when we use the shortcut, as we illustrate now. For the purpose of this discussion let's use the terminology $R_i + aR_j$ to mean the row operation that adds a times the jth row to the ith row, and aR_i to mean the operation that multiplies the ith row by a. Now we transform S to an upper triangular matrix U using the row operations

- 1. $\frac{1}{2}R_1$;
- 2. $R_2 + (-4)R_1$;
- 3. $R_3 + 2R_1$;
- 4. $R_3 + 3R_2$;
- 5. $(-1)R_2$; and finally,
- 6. $(\frac{-1}{4})R_3$.

We end up with

$$U = \begin{bmatrix} 1 & 1/2 & 1/2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}.$$

If we use the shortcut to form the lower triangular matrix L, we find that

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

Let's go through how we formed L from the row operations listed above. We get $L_{11} = 2$ from the first row operation, $L_{21} = 4$ from the second, $L_{31} = -2$ from the third, $L_{32} = -3$ from the fourth, $L_{22} = -1$ from the fifth, and $L_{33} = \frac{-1}{4}$ from the sixth. But, if we multiple LU we do not get back S! The problem is that we performed the fourth operation, adding to the third row three times the second row, before the $(2, 2)$ entry was rescaled to one. Suppose, instead, we do the row operations in this order:

- 1. $\frac{1}{2}R_1$;
- 2. $R_2 + (-4)R_1$;
- 3. $R_3 + 2R_1$;
- 4. $(-1)R_2$;
- 5. $R_3 - 3R_2$; and finally,
- 6. $(\frac{-1}{4})R_3$.

Then the entry L_{32} becomes 3, instead of -3, and now $LU = S$. The message is that if we want to use the shortcut and we plan to rescale the diagonal entries of U to be one, we should rescale a given row prior to adding any multiple of that row to another row; otherwise, we can get the wrong L. The problem is that certain elementary matrices associated with row operations do not commute.

We just saw that

$$L = L_1^{-1}L_2^{-1}L_3^{-1}.$$

However, when we form the matrix L simultaneously with performing the row operations, we are, in effect, calculating

$$L_3^{-1}L_2^{-1}L_1^{-1}.$$

Most of the time the order doesn't matter, and we get the correct L anyway. But this is not always the case. For example, if we perform the operation $\frac{1}{2}R_1$, followed by $R_2 + (-4)R_1$, this is not the same as doing $R_2 + (-4)R_1$, followed by $\frac{1}{2}R_1$.

With the matrix L_1 representing the operation $\frac{1}{2}R_1$ and the matrix L_2 representing the operation $R_2 + (-4)R_1$, we find that storing a 2 in the $(1, 1)$ position, and then a $+4$ in the $(1, 2)$ position as we build L is not equivalent to multiplying the identity matrix by $L_2^{-1}L_1^{-1}$ but rather multiplying the identity matrix by

$$(L_1^{-1}L_2^{-1}L_1)L_1^{-1} = L_1^{-1}L_2^{-1},$$

which is the correct order.

To illustrate this point, consider the matrix S given by

$$S = \begin{bmatrix} 2 & 1 & 1 \\ 4 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In the first instance, we perform the row operations $R_2 + (-2)R_1$, followed by $\frac{1}{2}R_1$ to get

$$U = \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0 & -1 & -2 \\ 0 & 0 & 1 \end{bmatrix}.$$

Using the shortcut, the matrix L becomes

$$L = \begin{bmatrix} 2 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

but we do not get $S = LU$. We do have $U = L_2L_1S$, where

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

and

$$L_2 = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

so that $S = L_1^{-1}L_2^{-1}U$ and the correct L is

$$L = L_1^{-1}L_2^{-1} = \begin{bmatrix} 2 & 0 & 0 \\ 4 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

But when we use the shortcut to generate L, we effectively multiply the identity matrix first by L_1^{-1} and then by L_2^{-1}, giving the matrix $L_2^{-1}L_1^{-1}$ as our candidate for L. But $L_1^{-1}L_2^{-1}$ and $L_2^{-1}L_1^{-1}$ are not the same. But why does reversing the order of the row operations work?

When we perform $\frac{1}{2}R_1$ first, and then $R_2 + (-4)R_1$ to get U, we are multiplying S first by L_2 and then by the matrix

$$E = \begin{bmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The correct L is then $L = L_2^{-1}E^{-1}$.

When we use the shortcut, we are first multiplying the identity by the matrix L_2^{-1} and then by a second matrix that we shall call J; the correct L must then be $L = JL_2^{-1}$. The matrix J is not E^{-1}, but

$$J = L_2^{-1}E^{-1}L_2,$$

so that

$$L = J - +L_2^{-1} = L_2^{-1}E^{-1}L_2L_2^{-1} = L_2^{-1}E^{-1},$$

which is correct.

8.2.3 Using the LU decomposition

Suppose that we have to solve the system of linear equations $Sx = b$. Once we have the LU factorization, it is a simple matter to find x: first, we solve the system $Lz = b$, and then solve $Ux = z$. Because both L and U are triangular, solving these systems is a simple matter. Obtaining the LU factorization is often better than finding S^{-1}; when S is banded, that is, has non-zero values only for the main diagonal and a few diagonals on either side, the L and U retain that banded property, while S^{-1} does not.

If H is real and symmetric, and if $H = LDU$, then $U = L^T$, so we have $H = LDL^T$. If, in addition, the non-zero entries of D are positive, then we can write

$$H = (L\sqrt{D})(L\sqrt{D})^T,$$

which is the Cholesky Decomposition of H.

Ex. 8.1 *Prove that, if L is invertible and lower triangular, then so is L^{-1}.*

Ex. 8.2 *Show that the symmetric matrix*

$$H = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

cannot be written as $H = LDL^T$.

Ex. 8.3 *Show that the symmetric matrix*

$$H = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

cannot be written as $H = LU$, where L is lower triangular, U is upper triangular, and both are invertible.

8.2.4 The Non-Square Case

If A is an M by N matrix, the same elimination procedure leads to a factoring $PA = LU$, where now the matrix L is square and lower-triangular and the matrix U is in *upper echelon form*, meaning that

- 1. the non-zero rows of U come at the top of U and the first non-zero entries are called the *pivots*;
- 2. below each pivot is a column of zeros;
- 3. each pivot lies to the right of the pivot in the row above it.

8.2.5 The LU Factorization in Linear Programming

Each step of the simplex algorithm involves solving systems of equations of the form $Bx = b$ and $B^T z = c$. As we proceed from one step to the next, the matrix B is updated by having one of its columns changed. This can be performed by multiplying B on the right by a matrix F that is the identity matrix, except for one column. The matrix $E = F^{-1}$ is then also the identity matrix, except for one column, so the updated inverse is

$$(B^{\text{new}})^{-1} = EB^{-1}.$$

As the calculations proceed, the next inverse can be represented in product form as

$$(B^{\text{new}})^{-1} = E_k E_{k-1} \cdots E_1 (B_0)^{-1},$$

where B_0 is the original choice for the matrix B. This product approach suggests a role for LU factorization, in which the individual factors L and U are updated in a stable manner as the iteration proceeds [268].

Ex. 8.4

- *a. Show that the matrix $B = A + x\delta_n^T$ differs from A only in the nth column, where x is an arbitrary column vector and δ_n is the nth column of the identity matrix.*
- *b. Let F be a matrix that is the identity matrix, except for one column. Show that the matrix $E = F^{-1}$, when it exists, is then also the identity matrix, except for one column, and compute E explicitly, in terms of the entries of F.*

Hint: use the identity in Equation 3.5.

8.3 When is $S = LU$?

Note that it may not be possible to obtain $S = LU$ without first permuting the rows of S; in such cases we obtain $PS = LU$, where P is obtained from the identity matrix by permuting rows.

We know from Exercise 8.3 that the invertible symmetric matrix

$$H = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

cannot be written as $H = LU$, where both L and U are invertible. In [271] Mark Yin gave a necessary and sufficient condition for a square matrix S to have the form $S = LU$, where both L and U are invertible.

Definition 8.1 *An n by n real matrix S is called a* T-matrix *if, for every partition*

$$S = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}$$

such that S_{11} is square, S_{11} is invertible.

Yin's theorem is the following:

Theorem 8.1 *An n by n matrix S has the form $S = LU$, where L is lower triangular, U is upper triangular, and both are invertible, if and only if S is a T-matrix.*

Proof: Suppose that $S = LU$ as in the statement of the theorem. Let S be partitioned arbitrarily, as

$$S = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix},$$

where S_{11} is square. Let

$$P = L^{-1} = \begin{bmatrix} P_{11} & 0 \\ P_{21} & P_{22} \end{bmatrix}$$

be an invertible lower triangular matrix, partitioned to be compatible with the partitioning of S, such that the matrix

$$PS = \begin{bmatrix} P_{11}S_{11} & P_{11}S_{12} \\ 0 & * \end{bmatrix}$$

is invertible and upper triangular. Since $P_{11}S_{11}$ must then be invertible, so is S_{11}. Since the partition is arbitrary, S is a T-matrix.

Now suppose that S is a T-matrix. We show that $S = LU$ as above. First of all, notice that, if P is invertible and lower triangular, then PS is also a T-matrix, since the upper left corner square sub-matrix of PS is $P_{11}S_{11}$.

The proof uses induction on the size n. The case of $n = 1$ is trivial, so assume that $n > 1$ and that the theorem holds for square matrices of size $n-1$ by $n-1$. Let

$$P_1 = \begin{bmatrix} 1 & 0^T \\ b & I \end{bmatrix},$$

where I is the identity matrix, s_{mn} are the entries of the matrix S, and

$$b^T = -\frac{1}{s_{11}}(s_{21}, ..., s_{n1}).$$

Then

$$P_1 S = \begin{bmatrix} s_{11} & 0^T \\ S_{12} & A_{22} \end{bmatrix},$$

where A_{22} is square and has size $n-1$. Since P_1S is a T-matrix, so is A_{22}. By the induction hypothesis, there is an invertible lower triangular matrix P_2 such that P_2A_{22} is invertible and upper triangular. It follows that RP_1S is invertible and upper triangular, where

$$R = \begin{bmatrix} 1 & 0^T \\ 0 & P_2 \end{bmatrix}.$$

Since RP_1 is invertible and lower triangular, the proof is completed. ∎

8.4 Householder Matrices

A real *Householder matrix* has the form

$$H = I - 2ww^T,$$

where w is a column vector in $\mathbb{R}^N$ with $\|w\|_2 = 1$.

Lemma 8.1 *For any Householder matrix we have $H^T = H$ and $H^{-1} = H$.*

Ex. 8.5 *Prove Lemma 8.1.*

Proposition 8.1 *Let x and y be any distinct members of $\mathbb{R}^N$ with $\|x\|_2 = \|y\|_2$, and let $w = \frac{1}{\|x-y\|_2}(x-y)$. Then $Hx = y$.*

Ex. 8.6 *Prove Proposition 8.1.*

We can use Householder matrices to turn certain non-zero entries of a vector to zero.

Given any vector x in $\mathbb{R}^N$, let $y_n = x_n$, for $n = 1, ..., k-1$, $y_n = 0$, for $n = k+1, ..., N$, and

$$|y_k| = \sqrt{x_k^2 + x_{k+1}^2 + ... + x_N^2},$$

where the sign of y_k is chosen to be opposite that of x_k. Then $\|x\|_2 = \|y\|_2$, the first $k-1$ entries of x and y agree, and the final $N-k$ entries of y are zero. If we then build the Householder matrix H using these x and y to create w, we find that $Hx = y$, so that the final $N-k$ entries are zero.

8.5 The QR Factorization

Given an invertible N by N real matrix S, we can multiply S on the left by a succession of Householder matrices $H_1, H_2, ..., H_{k-1}$ so that

$$H_{k-1} \cdots H_1 S = R$$

is upper triangular. Since $H_n^T = H_n = H_n^{-1}$, it follows that

$$Q^T = H_{k-1} \cdots H_1$$

is orthogonal, and that $S = QR$. This is the QR factorization of S. Once we have $S = QR$, we can solve $Sx = b$ easily, by solving $Rx = Q^T b$.

8.5.1 The Non-Square Case

Using the same approach, any real rectangular matrix A with linearly independent columns can be factored as $A = QR$, where R is square, upper triangular, and invertible, and the columns of Q are orthonormal, so that $Q^T Q = I$.

8.5.2 The QR Factorization and Least Squares

The least-squares solution of $Ax = b$ is the solution of $A^T Ax = A^T b$. Once we have $A = QR$, we have $A^T A = R^T Q^T QR = R^T R$, so we find the least squares solution easily, by solving $R^T z = A^T b$, and then $Rx = z$. Note that $A^T A = R^T R$ is the Cholesky decomposition of $A^T A$.

8.5.3 Upper Hessenberg Matrices

The time required to calculate the QR factorization of a general N by N matrix is proportional to N^3; the time is proportional to N^2 if the matrix has the *upper Hessenberg* form.

We say that a real N by N matrix has upper Hessenberg form if its non-zero entries occur on or above the main diagonal (as with an upper triangular matrix), or on the first sub-diagonal below the main diagonal. Note that any real N by N matrix S can be converted to upper Hessenberg form by multiplying on the left by a succession of Householder matrices; we can find Householder matrices H_1, H_2, ..., H_{k-2} so that

$$H_{k-2} \cdots H_1 A = B,$$

with B in upper Hessenberg form. The matrix

$$C = BH_1 \cdots H_{k-2} = H_{k-2} \cdots H_1 A H_1 \cdots H_{k-2}$$

is also in upper Hessenberg form. Since $C = P^{-1}AP$ for an invertible matrix P, the matrix C is similar to A, and so has the same eigenvalues. This will be helpful later.

8.5.4 The QR Method for Finding Eigenvalues

The QR factorization can be used to calculate the eigenvalues of a real N by N matrix S. The method proceeds as follows: begin with $S = S_0 = Q_0 R_0$, then define $S_1 = R_0 Q_0$. Next, perform the QR factorization on S_1 to get $S_1 = Q_1 R_1$, and define $S_2 = R_1 Q_1$, and so on. If S has only real eigenvalues, this procedure usually converges to an upper triangular matrix, whose eigenvalues are displayed along its main diagonal. Since $S_k = Q_k R_k$ and Q_k is orthogonal, we have $R_k = (Q_k)^T S_k$, so that

$$S_{k+1} = R_k Q_k = (Q_k)^T S_k Q_k = (Q_k)^{-1} S_k Q_k.$$

Therefore, each S_k is similar to S and so they have the same eigenvalues.

Part III

Algorithms

Chapter 9

The Split-Feasibility Problem

9.1 Chapter Summary

The general *feasibility problem* is to find a vector in the intersection of a finite number of given subsets of $\mathbb{R}^J$. Various iterative algorithms, such as the ART, and the Cimmino, Landweber, and projected Landweber algorithms, can be viewed as methods for solving a feasibility problem. In this chapter we present the split-feasibility problem, the CQ algorithm for solving the SFP, as well as two well known particular cases, the Landweber and projected Landweber algorithms. We also discuss recent extensions of the CQ algorithm and applications of the CQ algorithm to radiation therapy.

9.2 Some Examples

9.2.1 The ART

The ART can be viewed as a method for solving a feasibility problem. Let A be an I by J matrix, and b an I by 1 vector; we want to solve $Ax = b$. For each i we define the hyperplane H_i to be

$$H_i = \{x|(Ax)_i = b_i\}.$$

The problem is then to find x in the intersection of the subsets H_i. For any z in $\mathbb{C}^J$ there is a unique member of H_i closest to z; this vector is denoted $P_{H_i}z$, where the operator P_{H_i} is called the orthogonal projection onto H_i. We begin ART by choosing a starting vector, x^0. Having found x^k, we define x^{k+1} to be the orthogonal projection of x^k onto the set H_i, where $i = k(\mathrm{mod}\, I) + 1$; that is, we proceed through each i in order and then repeat as needed. The ART is a *sequential* method, in that we use only one set H_i at each step. If there are solutions of $Ax = b$, then the ART sequence $\{x^k\}$ converges to the solution closest to x^0 in the sense of the two norm. When there are no solutions of $Ax = b$ ART does not converge, but instead, produces a limit cycle of (typically) I distinct vectors, around which the iterates cycle.

9.2.2 Cimmino's Algorithm

Cimmino's algorithm is a *simultaneous* method that uses all the H_i at each step. Having found x^k, we calculate $P_{H_i}x^k$ for each i and then compute the arithmetic mean of these orthogonal projections. Algebraically, we have

$$x^{k+1} = \frac{1}{I}\sum_{i=1}^{I} P_{H_i}x^k. \tag{9.1}$$

This can also be written as

$$x^{k+1} = x^k + \frac{1}{I}A^\dagger(b - Ax^k). \tag{9.2}$$

When $Ax = b$ has solutions Cimmino's method produces the solution closest to x^0 again. When $Ax = b$ has no solutions, Cimmino's algorithm converges to the least-squares solution closest to x^0; that is, it converges to the minimizer of the function

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2 \tag{9.3}$$

closest to x^0.

9.2.3 Landweber's Algorithm

Landweber's algorithm [189] is more general than Cimmino's algorithm; it has the iterative step

$$x^{k+1} = x^k + \gamma A^\dagger (b - Ax^k), \tag{9.4}$$

where we must choose the parameter γ in the interval $(0, \frac{2}{\rho(A^\dagger A)})$. The Landweber algorithm behaves just like Cimmino's algorithm, but offers more flexibility and typically converges faster.

9.2.4 The Projected-Landweber Algorithm

The algorithms just discussed all use orthogonal projection onto the hyperplanes H_i; these hyperplanes are examples of convex sets. The projected Landweber algorithm [18] uses convex sets that may not be hyperplanes.

Definition 9.1 *A nonempty subset C of $\mathbb{C}^J$ is called* convex *if, for any two distinct x and y in C and any α in the interval $[0, 1]$, the* convex combination $(1 - \alpha)x + \alpha y$ *is a member of C as well.*

If C is a nonempty closed convex set and z is any vector, there will be a member of C closest to z in the two norm; we denote this closest vector $P_C z$, and the operator P_C is called the orthogonal projection onto C.

Suppose that we want to find a solution of $Ax = b$ that lies in the closed convex set C; then we can use the projected Landweber algorithm, whose iterative step is

$$x^{k+1} = P_C(x^k + \gamma A^\dagger (b - Ax^k)). \tag{9.5}$$

When there are solutions of $Ax = b$ that lie in C this algorithm, this algorithm provides such a solution. When there are no such solutions, the projected Landweber algorithm minimizes the function in Equation (9.3) over the set C.

9.3 The Split-Feasibility Problem

The *split-feasibility problem* (SFP) [83] is a special type of feasibility problem. The SFP is to find $c \in C$ with $Ac \in Q$, if such points exist, where A is an I by J matrix and C and Q are nonempty, closed convex sets in $\mathbb{C}^J$ and $\mathbb{C}^I$, respectively. If $C = \mathbb{C}^J$ and $Q = \{b\}$, the problem is simply to find x with $Ax = b$; for other sets C, the problem is to find an x in C with $Ax = b$. Recall that, for any nonempty, closed, convex subset C of $\mathbb{C}^J$, and

any x in $\mathbb{C}^J$, the vector $P_C x$ is the unique member of C closest to x in the sense of the two norm.

9.4 The CQ Algorithm

The CQ algorithm for solving the SFP was presented in [62], and developed further in [63]. It has the iterative step

$$x^{k+1} = P_C(x^k - \gamma A^\dagger(I - P_Q)Ax^k), \tag{9.6}$$

where I is the identity operator and $\gamma \in (0, 2/\rho(A^T A))$, for $\rho(A^\dagger A)$ the spectral radius of the matrix $A^\dagger A$, which is also its largest eigenvalue. The CQ algorithm converges to a solution of the SFP, for any starting vector x^0, whenever the SFP has solutions. When the SFP has no solutions, the CQ algorithm converges to a minimizer of the function

$$f(x) = \frac{1}{2}||P_Q Ax - Ax||_2^2$$

over the set C, provided such constrained minimizers exist [63].

Choosing the best relaxation parameter in any algorithm is a nontrivial procedure. Generally speaking, we want to select γ near to $1/L$. If A is normalized so that each row has length one, then the spectral radius of $A^\dagger A$ does not exceed the maximum number of nonzero elements in any column of A. A similar upper bound on $\rho(A^\dagger A)$ can be obtained for nonnormalized, ϵ-sparse A.

9.5 Particular Cases of the CQ Algorithm

It is easy to find important examples of the SFP: if $C \subseteq \mathbb{C}^J$ and $Q = \{b\}$ then solving the SFP amounts to solving the linear system of equations $Ax = b$; if C is a proper subset of $\mathbb{C}^J$, such as the nonnegative cone, then we seek solutions of $Ax = b$ that lie within C, if there are any. Generally, we cannot solve the SFP in closed form and iterative methods are needed.

A number of well known iterative algorithms, such as the Landweber and projected Landweber methods, are particular cases of the CQ algorithm.

9.5.1 Convergence of the Landweber Algorithms

From the convergence theorem for the CQ algorithm it follows that the Landweber algorithm converges to a solution of $Ax = b$ and the projected Landweber algorithm converges to a solution of $Ax = b$ in C, whenever such solutions exist. When there are no solutions of the desired type, the Landweber algorithm converges to a least squares approximate solution of $Ax = b$, while the projected Landweber algorithm will converge to a minimizer, over the set C, of the function $||b - Ax||_2$, whenever such a minimizer exists.

9.5.2 The Simultaneous ART (SART)

Another example of the CQ algorithm is the *simultaneous algebraic reconstruction technique* (SART) [5] for solving $Ax = b$, for nonnegative matrix A. Let A be an I by J matrix with nonnegative entries. Let $A_{i+} > 0$ be the sum of the entries in the ith row of A and $A_{+j} > 0$ be the sum of the entries in the jth column of A. Consider the (possibly inconsistent) system $Ax = b$. The SART algorithm has the following iterative step:

$$x_j^{k+1} = x_j^k + \frac{1}{A_{+j}} \sum\nolimits_{i=1}^{I} A_{ij}(b_i - (Ax^k)_i)/A_{i+}. \tag{9.7}$$

We make the following changes of variables:

$$B_{ij} = A_{ij}/(A_{i+})^{1/2}(A_{+j})^{1/2},$$

$$z_j = x_j(A_{+j})^{1/2},$$

and

$$c_i = b_i/(A_{i+})^{1/2}.$$

Then the SART iterative step can be written as

$$z^{k+1} = z^k + B^T(c - Bz^k). \tag{9.8}$$

This is a particular case of the Landweber algorithm, with $\gamma = 1$. The convergence of SART follows from that of the CQ algorithm, once we know that the largest eigenvalue of B^TB is less than two; in fact, we show that it is one [62].

If B^TB had an eigenvalue greater than one and some of the entries of A are zero, then, replacing these zero entries with very small positive entries, we could obtain a new A whose associated B^TB also had an eigenvalue greater than one. Therefore, we assume, without loss of generality, that A has all positive entries. Since the new B^TB also has only positive entries, this matrix is irreducible and the Perron-Frobenius Theorem applies. We shall use this to complete the proof.

Let $u = (u_1, ..., u_J)^T$ with $u_j = (A_{+j})^{1/2}$ and $v = (v_1, ..., v_I)^T$, with $v_i = (A_{i+})^{1/2}$. Then we have $Bu = v$ and $B^T v = u$; that is, u is an eigenvector of $B^T B$ with associated eigenvalue equal to one, and all the entries of u are positive, by assumption. The Perron-Frobenius theorem applies and tells us that the eigenvector associated with the largest eigenvalue has all positive entries. Since the matrix $B^T B$ is symmetric its eigenvectors are orthogonal; therefore u itself must be an eigenvector associated with the largest eigenvalue of $B^T B$. The convergence of SART follows.

9.5.3 Application of the CQ Algorithm in Dynamic ET

To illustrate how an image reconstruction problem can be formulated as a SFP, we consider briefly *emission computed tomography* (ET) image reconstruction. The objective in ET is to reconstruct the internal spatial distribution of intensity of a radionuclide from counts of photons detected outside the patient. In static ET the intensity distribution is assumed constant over the scanning time. Our data are photon counts at the detectors, forming the positive vector b and we have a real matrix A of detection probabilities; our model is $Ax = b$, for x a nonnegative vector. We could then take $Q = \{b\}$ and $C = \mathbb{R}^J_+$, the nonnegative cone in $\mathbb{R}^J$.

In *dynamic* ET [131] the intensity levels at each voxel may vary with time. The observation time is subdivided into, say, T intervals and one static image, call it x^t, is associated with the time interval denoted by t, for $t = 1, ..., T$. The vector x is the concatenation of these T image vectors x^t. The discrete time interval at which each data value is collected is also recorded and the problem is to reconstruct this succession of images.

Because the data associated with a single time interval is insufficient, by itself, to generate a useful image, one often uses prior information concerning the time history at each fixed voxel to devise a model of the behavior of the intensity levels at each voxel, as functions of time. One may, for example, assume that the radionuclide intensities at a fixed voxel are increasing with time, or are concave (or convex) with time. The problem then is to find $x \geq 0$ with $Ax = b$ and $Dx \geq 0$, where D is a matrix chosen to describe this additional prior information. For example, we may wish to require that, for each fixed voxel, the intensity is an increasing function of (discrete) time; then we want

$$x_j^{t+1} - x_j^t \geq 0,$$

for each t and each voxel index j. Or, we may wish to require that the intensity at each voxel describes a concave function of time, in which case nonnegative second differences would be imposed:

$$(x_j^{t+1} - x_j^t) - (x_j^{t+2} - x_j^{t+1}) \geq 0.$$

In either case, the matrix D can be selected to include the left sides of these inequalities, while the set Q can include the nonnegative cone as one factor.

9.5.4 More on the CQ Algorithm

One of the obvious drawbacks to the use of the CQ algorithm is that we would need the projections P_C and P_Q to be easily calculated. Several authors have offered remedies for that problem, using approximations of the convex sets by the intersection of hyperplanes and orthogonal projections onto those hyperplanes [270].

9.5.5 Convex Feasibility and IMRT

Because the CQ algorithm is simpler than previously published algorithms for solving the SFP, it has become the focus of recent work in intensity modulated radiation therapy (IMRT). In [86] Censor *et al.* extend the CQ algorithm to solve what they call the *multiple-set split-feasibility problem* (MSSFP) . In the sequel [84] it is shown that the constraints in IMRT can be modeled as inclusion in convex sets and the extended CQ algorithm is used to determine dose intensities for IMRT that satisfy both dose constraints and radiation-source constraints.

9.6 Applications of the PLW Algorithm

Suppose that G is an arbitrary I by J matrix, and that $D \subseteq \mathbb{C}^J$ is a closed, nonempty convex set. We can use the PLW algorithm to minimize $\|Gw\|_2$ over $w \in D$: the iterative step is

$$w^{k+1} = P_D(w^k - \gamma G^\dagger G w^k), \tag{9.9}$$

for $0 < \gamma < \frac{2}{\rho(G^\dagger G)}$. The sequence $\{w^k\}$ converges to a minimizer, over $w \in D$, of $\|Gw\|_2$, whenever such minimizers exist.

Suppose now that A is an M by N matrix, and B an M by K matrix. Suppose also that $C \subseteq \mathbb{C}^N$, and $Q \subseteq \mathbb{C}^M$ are closed, nonempty convex sets. We want to find $x \in C$ and $y \in Q$ with $Ax = By$. Failing that, we want to minimize $\|Ax - By\|_2$ over $x \in C$ and $y \in Q$.

Let $G = \begin{bmatrix} A & -B \end{bmatrix}$ and $w = \begin{bmatrix} x \\ y \end{bmatrix}$ in $\mathbb{C}^{N+K}$. Then $Gw = Ax - By$. We apply the iteration in Equation (9.9) to minimize $\|Gw\|_2$ over $w \in D = C \times Q$, or, equivalently, to minimize $\|Ax - By\|_2$ over $x \in C$ and $y \in Q$.

We have

$$G^\dagger G = \begin{bmatrix} A^\dagger A & -A^\dagger B \\ -B^\dagger A & B^\dagger B \end{bmatrix},$$

so that the iteration in Equation (9.9) becomes

$$x^{k+1} = P_C(x^k - \gamma A^\dagger(Ax^k - By^k)), \tag{9.10}$$

and

$$y^{k+1} = P_Q(y^k + \gamma B^\dagger(Ax^k - By^k)). \tag{9.11}$$

Chapter 10

Jacobi and Gauss-Seidel Methods

10.1 Chapter Summary

In this chapter we consider two well known iterative algorithms for solving square systems of linear equations, the Jacobi method and the Gauss-Seidel method. Both these algorithms are easy to describe and to motivate. They both require not only that the system be square, that is, have the same number of unknowns as equations, but satisfy additional constraints needed for convergence.

Linear systems $Ax = b$ need not be square but can be associated with two square systems, $A^\dagger Ax = A^\dagger b$, the so-called *normal equations*, and $AA^\dagger z = b$, sometimes called the *Björck-Elfving equations* [110]. Both the Jacobi and the Gauss-Seidel algorithms can be modified to apply to any square system of linear equations, $Sz = h$. The resulting algorithms, the Jacobi overrelaxation (JOR) and successive overrelaxation (SOR) methods, involve the choice of a parameter. The JOR and SOR will converge for more general classes of matrices, provided that the parameter is appropriately chosen.

When we say that an iterative method is convergent, or converges, under certain conditions, we mean that it converges for any consistent system of the appropriate type, and for any starting vector; any iterative method will converge if we begin at the right answer. We assume throughout this chapter that A is an I by J matrix.

10.2 The Jacobi and Gauss-Seidel Methods: An Example

Suppose we wish to solve the 3 by 3 system

$$S_{11}z_1 + S_{12}z_2 + S_{13}z_3 = h_1$$

$$S_{21}z_1 + S_{22}z_2 + S_{23}z_3 = h_2$$

$$S_{31}z_1 + S_{32}z_2 + S_{33}z_3 = h_3, \tag{10.1}$$

which we can rewrite as

$$z_1 = S_{11}^{-1}[h_1 - S_{12}z_2 - S_{13}z_3]$$

$$z_2 = S_{22}^{-1}[h_2 - S_{21}z_1 - S_{23}z_3]$$

$$z_3 = S_{33}^{-1}[h_3 - S_{31}z_1 - S_{32}z_2], \tag{10.2}$$

assuming that the diagonal terms S_{mm} are not zero. Let $z^0 = (z_1^0, z_2^0, z_3^0)^T$ be an initial guess for the solution. We then insert the entries of z^0 on the right sides and use the left sides to define the entries of the next guess z^1. This is one full cycle of *Jacobi's method*.

The Gauss-Seidel method is similar. Let $z^0 = (z_1^0, z_2^0, z_3^0)^T$ be an initial guess for the solution. We then insert z_2^0 and z_3^0 on the right side of the first equation, obtaining a new value z_1^1 on the left side. We then insert z_3^0 and z_1^1 on the right side of the second equation, obtaining a new value z_2^1 on the left. Finally, we insert z_1^1 and z_2^1 into the right side of the third equation, obtaining a new z_3^1 on the left side. This is one full cycle of the *Gauss-Seidel* (GS) method.

10.3 Splitting Methods

The Jacobi and the Gauss-Seidel methods are particular cases of a more general approach known as *splitting methods*. Splitting methods apply to square systems of linear equations. Let S be an arbitrary N by N square matrix, written as $S = M - K$. Then the linear system of equations $Sz = h$ is equivalent to $Mz = Kz + h$. If M is invertible, then we can also write $z = M^{-1}Kz + M^{-1}h$. This last equation suggests a class of iterative methods

for solving $Sz = h$ known as *splitting methods*. The idea is to select a matrix M so that the equation

$$Mz^{k+1} = Kz^k + h \tag{10.3}$$

can be easily solved to get z^{k+1}; in the Jacobi method M is diagonal, and in the Gauss-Seidel method, M is triangular. Then we write

$$z^{k+1} = M^{-1}Kz^k + M^{-1}h. \tag{10.4}$$

From $K = M - S$, we can write Equation (10.4) as

$$z^{k+1} = z^k + M^{-1}(h - Sz^k). \tag{10.5}$$

Suppose that S is invertible and $\hat{z}$ is the unique solution of $Sz = h$. The error we make at the k-th step is $e^k = \hat{z} - z^k$, so that

$$e^{k+1} = M^{-1}Ke^k.$$

We want the error to decrease with each step, which means that we should seek M and K so that $||M^{-1}K|| < 1$. If S is not invertible and there are multiple solutions of $Sz = h$, then we do not want $M^{-1}K$ to be a strict contraction. The operator T defined by

$$Tz = M^{-1}Kz + M^{-1}h = Bz + d \tag{10.6}$$

is an affine linear operator.

In what follows we shall write an arbitrary square matrix S as

$$S = L + D + U, \tag{10.7}$$

where L is the strictly lower triangular part of S, D the diagonal part, and U the strictly upper triangular part. When $S = H$ is Hermitian, we have

$$H = L + D + L^\dagger. \tag{10.8}$$

We list now several examples of iterative algorithms obtained by the splitting method. In the remainder of the chapter we discuss these methods in more detail.

10.4 Some Examples of Splitting Methods

As we shall now see, the Jacobi and Gauss-Seidel methods, as well as their overrelaxed versions, JOR and SOR, are splitting methods.

Jacobi's Method: Jacobi's method uses $M = D$ and $K = -L - U$, under the assumption that D is invertible. The matrix B is

$$B = M^{-1}K = -D^{-1}(L + U). \tag{10.9}$$

The Gauss-Seidel Method: The Gauss-Seidel (GS) method uses the splitting $M = D + L$, so that the matrix B is

$$B = I - (D + L)^{-1}S. \tag{10.10}$$

The Jacobi Overrelaxation Method (JOR): The JOR uses the splitting

$$M = \frac{1}{\omega}D \tag{10.11}$$

and

$$K = M - S = (\frac{1}{\omega} - 1)D - L - U. \tag{10.12}$$

The matrix B is

$$B = M^{-1}K = (I - \omega D^{-1}S). \tag{10.13}$$

The Successive Overrelaxation Method (SOR): The SOR uses the splitting $M = (\frac{1}{\omega}D + L)$, so that

$$B = M^{-1}K = (D + \omega L)^{-1}[(1 - \omega)D - \omega U] \tag{10.14}$$

or

$$B = I - \omega(D + \omega L)^{-1}S, \tag{10.15}$$

or

$$B = (I + \omega D^{-1}L)^{-1}[(1 - \omega)I - \omega D^{-1}U]. \tag{10.16}$$

10.5 Jacobi's Algorithm and JOR

The Jacobi iterative scheme will not converge, in general. Additional conditions need to be imposed on S in order to guarantee convergence. One such condition is that S be strictly diagonally dominant. In that case, all the eigenvalues of $B = M^{-1}K$ can be shown to lie inside the unit circle

of the complex plane, so that $\rho(B) < 1$. Alternatively, one has the *Jacobi overrelaxation* (JOR) method, which is essentially a special case of the Landweber algorithm and involves an arbitrary parameter.

For S an N by N matrix, Jacobi's method can be written as

$$z_m^{\text{new}} = S_{mm}^{-1}[h_m - \sum_{j \neq m} S_{mj} z_j^{\text{old}}], \tag{10.17}$$

for $m = 1, ..., N$. With D the invertible diagonal matrix with entries $D_{mm} = S_{mm}$ we can write one cycle of Jacobi's method as

$$z^{\text{new}} = z^{\text{old}} + D^{-1}(h - Sz^{\text{old}}). \tag{10.18}$$

The *Jacobi overrelaxation* (JOR) method has the following full-cycle iterative step:

$$z^{\text{new}} = z^{\text{old}} + \omega D^{-1}(h - Sz^{\text{old}}); \tag{10.19}$$

choosing $\omega = 1$ we get the Jacobi method. Convergence of the JOR iteration will depend, of course, on properties of S and on the choice of ω. When $S = Q$, where Q is Hermitian and nonnegative-definite, for example, $S = A^\dagger A$ or $S = AA^\dagger$, we can say more. Note that such Q can always be written in the form $Q = AA^\dagger$ or $Q = A^\dagger A$, for appropriately chosen A.

The JOR method, as applied to $Qz = AA^\dagger z = b$, is equivalent to the Landweber iterative method for $Ax = b$.

Ex. 10.1 *Show that the system $AA^\dagger z = b$ has solutions whenever the system $Ax = b$ has solutions.*

Lemma 10.1 *If $\{z^k\}$ is the sequence obtained from the JOR, then the sequence $\{A^\dagger z^k\}$ is the sequence obtained by applying the Landweber algorithm to the system $D^{-1/2}Ax = D^{-1/2}b$, where D is the diagonal part of the matrix $Q = AA^\dagger$.*

If we select $\omega = 1/I$ we obtain Cimmino's algorithm. Since the trace of the matrix $D^{-1/2}QD^{-1/2}$ equals I, which then is the sum of its eigenvalues, all of which are non-negative, we know that $\omega = 1/I$ is less than two over the largest eigenvalue of the matrix $D^{-1/2}QD^{-1/2}$ and so this choice of ω is acceptable and the Cimmino algorithm converges whenever there are solutions of $Ax = b$. In fact, it can be shown that Cimmino's method converges to a least squares approximate solution generally.

Similarly, the JOR method applied to the system $A^\dagger Ax = A^\dagger b$ is equivalent to the Landweber algorithm, applied to the system $Ax = b$.

Ex. 10.2 *Show that, if $\{z^k\}$ is the sequence obtained from the JOR, then the sequence $\{D^{1/2}z^k\}$ is the sequence obtained by applying the Landweber algorithm to the system $AD^{-1/2}x = b$, where D is the diagonal part of the matrix $S = A^\dagger A$.*

10.6 The Gauss-Seidel Algorithm and SOR

In general, the full-cycle iterative step of the Gauss-Seidel method is the following:

$$z^{\text{new}} = z^{\text{old}} + (D + L)^{-1}(h - Sz^{\text{old}}), \tag{10.20}$$

where $S = D + L + U$ is the decomposition of the square matrix S into its diagonal, lower triangular and upper triangular diagonal parts. The GS method does not converge without restrictions on the matrix S. As with the Jacobi method, strict diagonal dominance is a sufficient condition.

10.6.1 The Nonnegative-Definite Case

Now we consider the square system $Qz = h$, assuming that $Q = L + D + L^\dagger$ is Hermitian and nonnegative-definite, so that $x^\dagger Qx \geq 0$, for all x. It is easily shown that all the entries of D are nonnegative. We assume that all the diagonal entries of D are positive, so that $D + L$ is invertible. The Gauss-Seidel iterative step is $z^{k+1} = Tz^k$, where T is the affine linear operator given by $Tz = Bz+d$, for $B = -(D+L)^{-1}L^\dagger$ and $d = (D+L)^{-1}h$.

Proposition 10.1 *Let λ be an eigenvalue of B that is not equal to one. Then $|\lambda| < 1$.*

Proof: Let $Bv = \lambda v$, for v nonzero. Then $-Bv = (D + L)^{-1}L^\dagger v = -\lambda v$, so that

$$L^\dagger v = -\lambda(D + L)v. \tag{10.21}$$

Therefore,

$$v^\dagger L^\dagger v = -\lambda v^\dagger (D + L)v. \tag{10.22}$$

Adding $v^\dagger(D + L)v$ to both sides, we get

$$v^\dagger Qv = (1 - \lambda)v^\dagger(D + L)v. \tag{10.23}$$

Since the left side of the equation is real, so is the right side. Therefore

$$\begin{aligned}(1-\lambda)v^\dagger(D+L)v &= (1-\overline{\lambda})v^\dagger(D+L)^\dagger v \\ &= (1-\overline{\lambda})v^\dagger Dv + (1-\overline{\lambda})v^\dagger L^\dagger v \\ &= (1-\overline{\lambda})v^\dagger Dv - (1-\overline{\lambda})\lambda v^\dagger(D+L)v. \end{aligned} \tag{10.24}$$

So we have

$$[(1-\lambda)+(1-\overline{\lambda})\lambda]v^\dagger(D+L)v = (1-\overline{\lambda})v^\dagger Dv, \tag{10.25}$$

or

$$(1-|\lambda|^2)v^\dagger(D+L)v = (1-\overline{\lambda})v^\dagger Dv. \tag{10.26}$$

Multiplying by $(1-\lambda)$ on both sides, we get, on the left side,

$$(1-|\lambda|^2)v^\dagger(D+L)v - (1-|\lambda|^2)\lambda v^\dagger(D+L)v, \tag{10.27}$$

which is equal to

$$(1-|\lambda|^2)v^\dagger(D+L)v + (1-|\lambda|^2)v^\dagger L^\dagger v, \tag{10.28}$$

and, on the right side, we get

$$|1-\lambda|^2 v^\dagger Dv. \tag{10.29}$$

Consequently, we have

$$(1-|\lambda|^2)v^\dagger Qv = |1-\lambda|^2 v^\dagger Dv. \tag{10.30}$$

Since $v^\dagger Qv \geq 0$ and $v^\dagger Dv > 0$, it follows that $1-|\lambda|^2 \geq 0$. If $|\lambda| = 1$, then $|1-\lambda|^2 = 0$, so that $\lambda = 1$. This completes the proof. ∎

Note that $\lambda = 1$ if and only if $Qv = 0$. Therefore, if Q is invertible, the affine linear operator T is a strict contraction, and the GS iteration converges to the unique solution of $Qz = h$. If B is diagonalizable, then the GS iteration converges to a solution of $Qz = h$, whenever solutions exist.

10.6.2 The GS Algorithm as ART

We show now that the GS algorithm, when applied to the system $Qz = AA^\dagger z = b$, is equivalent to the ART algorithm, applied to $Ax = b$. Let $AA^\dagger = Q = L + D + L^\dagger$.

It is convenient now to consider separately each sub-iteration step of the GS algorithm. For $m = 0, 1, ...$ and $i = m(\mathrm{mod}\, I) + 1$, we denote by z^{m+1} the vector whose entries are

$$z_i^{m+1} = D_{ii}^{-1}\Big(b_i - (Qz^m)_i + Q_{ii}z_i^m\Big),$$

and $z_n^{m+1} = z_n^m$, for $n \neq i$. Therefore, we can write

$$z_i^{m+1} - z_i^m = D_{ii}^{-1}(b_i - (AA^\dagger z^m)_i).$$

Now let $x^m = A^\dagger z^m$ for each m. Then we have

$$x_j^{m+1} = (A^\dagger z^{m+1})_j = (A^\dagger z^m)_j + \overline{A_{ij}} D_{ii}^{-1}(b_i - (Ax^m)_i),$$

which is one step of the ART algorithm, applied to the system $Ax = b$. Note that

$$D_{ii} = \sum_{j=1}^{J} |A_{ij}|^2.$$

From this, we can conclude that if $\{z^k\}$ is the sequence produced by one step of the GS algorithm, applied to the system $AA^\dagger z = b$, then $\{x^k = A^\dagger z^k\}$ is the sequence produced by one full cycle of the ART algorithm, applied to the system $Ax = b$. Since we know that the ART algorithm converges whenever $Ax = b$ is consistent, we know now that the GS algorithm, applied to the system $AA^\dagger z = b$, converges whenever $Ax = b$ is consistent. So once again we have shown that when $S = Q$ is Hermitian and non-negative definite, the GS method converges whenever there are solutions of $Qz = h$.

10.6.3 Successive Overrelaxation

The *successive overrelaxation* (SOR) method has the following full-cycle iterative step:

$$z^{\text{new}} = z^{\text{old}} + (\omega^{-1}D + L)^{-1}(h - Sz^{\text{old}}); \tag{10.31}$$

the choice of $\omega = 1$ gives the GS method. Convergence of the SOR iteration will depend, of course, on properties of S and on the choice of ω.

Using the form

$$B = (D + \omega L)^{-1}[(1 - \omega)D - \omega U] \tag{10.32}$$

we can show that

$$|\det(B)| = |1 - \omega|^N. \tag{10.33}$$

From this and the fact that the determinant of B is the product of its eigenvalues, we conclude that $\rho(B) > 1$ if $\omega < 0$ or $\omega > 2$. When $S = Q$ is Hermitian and nonnegative-definite, we can say more.

10.6.4 The SOR for Nonnegative-Definite Q

When Q is nonnegative-definite and the system $Qz = h$ is consistent the SOR converges to a solution for any $\omega \in (0, 2)$. This follows from the convergence of the ART algorithm, since, for such Q, the SOR is equivalent to the ART, as we now show.

Now we write $Q = AA^\dagger$ and consider the SOR method applied to the Björck-Elfving equations $AA^\dagger z = b$. Rather than count a full cycle as one iteration, we now count as a single step the calculation of a single new entry. Therefore, for $k = 0, 1, ...$ the $k+1$-st step replaces the value z_i^k only, where $i = k(\mathrm{mod}\, I) + 1$. We have

$$z_i^{k+1} = (1 - \omega)z_i^k + \omega D_{ii}^{-1}(b_i - \sum_{n=1}^{i-1} Q_{in}z_n^k - \sum_{n=i+1}^{I} Q_{in}z_n^k) \tag{10.34}$$

and $z_n^{k+1} = z_n^k$ for $n \neq i$. Now we calculate $x^{k+1} = A^\dagger z^{k+1}$:

$$x_j^{k+1} = x_j^k + \omega D_{ii}^{-1}\overline{A_{ij}}(b_i - (Ax^k)_i). \tag{10.35}$$

This is one step of the relaxed *algebraic reconstruction technique* (ART) applied to the original system of equations $Ax = b$. The relaxed ART converges to a solution, when solutions exist, for any $\omega \in (0, 2)$.

When $Ax = b$ is consistent, so is $AA^\dagger z = b$. We consider now the case in which $Q = AA^\dagger$ is invertible. Since the relaxed ART sequence $\{x^k = A^\dagger z^k\}$ converges to a solution x^∞, for any $\omega \in (0, 2)$, the sequence $\{AA^\dagger z^k\}$ converges to b. Since $Q = AA^\dagger$ is invertible, the SOR sequence $\{z^k\}$ then converges to $Q^{-1}b$.

Chapter 11

Conjugate-Direction Methods

11.1 Chapter Summary

Finding the least-squares solution of a possibly inconsistent system of linear equations $Ax = b$ is equivalent to minimizing the quadratic function $f(x) = \frac{1}{2}||Ax - b||_2^2$ and so can be viewed within the framework of optimization. Iterative optimization methods can then be used to provide, or at least suggest, algorithms for obtaining the least-squares solution. The *conjugate gradient method* is one such method.

11.2 Iterative Minimization

Iterative methods for minimizing a real-valued function $f(x)$ over the vector variable x usually take the following form: having obtained x^{k-1}, a new direction vector d^k is selected, an appropriate scalar $\alpha_k > 0$ is determined and the next member of the iterative sequence is given by

$$x^k = x^{k-1} + \alpha_k d^k. \tag{11.1}$$

Ideally, one would choose the α_k to be the value of α for which the function $f(x^{k-1}+\alpha d^k)$ is minimized. It is assumed that the direction d^k is a *descent direction*; that is, for small positive α the function $f(x^{k-1}+\alpha d^k)$ is strictly decreasing. Finding the optimal value of α at each step of the iteration is difficult, if not impossible, in most cases, and approximate methods, using line searches, are commonly used.

Ex. 11.1 *Differentiate the function $f(x^{k-1}+\alpha d^k)$ with respect to the variable α to show that, when $\alpha = \alpha_k$ is optimal, then*

$$\nabla f(x^k) \cdot d^k = 0. \tag{11.2}$$

Since the gradient $\nabla f(x^k)$ is orthogonal to the previous direction vector d^k and also because $-\nabla f(x)$ is the direction of greatest decrease of $f(x)$, the choice of $d^{k+1} = -\nabla f(x^k)$ as the next direction vector is a reasonable one. With this choice we obtain Cauchy's *steepest descent method* [200]:

$$x^{k+1} = x^k - \alpha_{k+1}\nabla f(x^k).$$

The steepest descent method need not converge in general and even when it does, it can do so slowly, suggesting that there may be better choices for the direction vectors. For example, the Newton-Raphson method [211] employs the following iteration:

$$x^{k+1} = x^k - \nabla^2 f(x^k)^{-1}\nabla f(x^k),$$

where $\nabla^2 f(x)$ is the Hessian matrix for $f(x)$ at x. To investigate further the issues associated with the selection of the direction vectors, we consider the more tractable special case of quadratic optimization.

11.3 Quadratic Optimization

Let A be an arbitrary real I by J matrix. The linear system of equations $Ax = b$ need not have any solutions, and we may wish to find a least-squares solution $x = \hat{x}$ that minimizes

$$f(x) = \frac{1}{2}||b - Ax||_2^2. \tag{11.3}$$

The vector b can be written

$$b = A\hat{x} + \hat{w},$$

where $A^T\hat{w} = 0$ and a least squares solution is an exact solution of the linear system $Qx = c$, with $Q = A^TA$ and $c = A^Tb$. We shall assume that Q is invertible and there is a unique least squares solution; this is the typical case.

We consider now the iterative scheme described by Equation (11.1) for $f(x)$ as in Equation (11.3). For this $f(x)$ the gradient becomes

$$\nabla f(x) = Qx - c.$$

The optimal α_k for the iteration can be obtained in closed form.

Ex. 11.2 *Show that the optimal α_k is*

$$\alpha_k = \frac{r^k \cdot d^k}{d^k \cdot Qd^k}, \tag{11.4}$$

where $r^k = c - Qx^{k-1}$.

Ex. 11.3 *Let $||x||_Q^2 = x \cdot Qx$ denote the square of the Q-norm of x. Show that*

$$||\hat{x} - x^{k-1}||_Q^2 - ||\hat{x} - x^k||_Q^2 = (r^k \cdot d^k)^2/d^k \cdot Qd^k \geq 0$$

for any direction vectors d^k.

If the sequence of direction vectors $\{d^k\}$ is completely general, the iterative sequence need not converge. However, if the set of direction vectors is finite and spans $\mathbb{R}^J$ and we employ them cyclically, convergence follows.

Theorem 11.1 *Let $\{d^1, ..., d^J\}$ be any finite set whose span is all of $\mathbb{R}^J$. Let α_k be chosen according to Equation (11.4). Then, for $k = 1, 2, ...$, $j = k(\mathrm{mod}\, J)$, and any x^0, the sequence defined by*

$$x^k = x^{k-1} + \alpha_k d^j$$

converges to the least squares solution.

Proof: The sequence $\{||\hat{x} - x^k||_Q^2\}$ is decreasing and, therefore, the sequence $\{(r^k \cdot d^k)^2/d^k \cdot Qd^k$ must converge to zero. Therefore, the vectors x^k are bounded, and for each $j = 1, ..., J$, the subsequences $\{x^{mJ+j}, m = 0, 1, ...\}$ have cluster points, say $x^{*,j}$ with

$$x^{*,j} = x^{*,j-1} + \frac{(c - Qx^{*,j-1}) \cdot d^j}{d^j \cdot Qd^j} d^j.$$

Since

$$r^{mJ+j} \cdot d^j \to 0,$$

it follows that, for each $j = 1, ..., J$,

$$(c - Qx^{*,j}) \cdot d^j = 0.$$

Therefore,

$$x^{*,1} = ... = x^{*,J} = x^*$$

with $Qx^* = c$. Consequently, x^* is the least squares solution and the sequence $\{||x^* - x^k||_Q\}$ is decreasing. But a subsequence converges to zero; therefore, $\{||x^* - x^k||_Q\} \to 0$. This completes the proof. ∎

In the quadratic case the steepest descent iteration has the form

$$x^k = x^{k-1} + \frac{r^k \cdot r^k}{r^k \cdot Qr^k} r^k.$$

We have the following result.

Theorem 11.2 *The steepest descent method converges to the least-squares solution.*

Proof: As in the proof of the previous theorem, we have

$$||\hat{x} - x^{k-1}||_Q^2 - ||\hat{x} - x^k||_Q^2 = (r^k \cdot d^k)^2 / d^k \cdot Qd^k \geq 0,$$

where now the direction vectors are $d^k = r^k$. So, the sequence $\{||\hat{x} - x^k||_Q^2\}$ is decreasing, and therefore the sequence $\{(r^k \cdot r^k)^2 / r^k \cdot Qr^k\}$ must converge to zero. The sequence $\{x^k\}$ is bounded; let x^* be a cluster point. It follows that $c - Qx^* = 0$, so that x^* is the least-squares solution $\hat{x}$. The rest of the proof follows as in the proof of the previous theorem. ∎

There is an interesting corollary to the theorem that pertains to a modified version of the ART algorithm. For $k = 1, 2, ...$ and $i = k(\mathrm{mod}\, M)$ and with the rows of A normalized to have length one, the ART iterative step is

$$x^k = x^{k-1} + (b_i - (Ax^{k-1})_i)a^i,$$

where a^i is the ith column of A^T. When $Ax = b$ has no solutions, the ART algorithm does not converge to the least-squares solution; rather, it exhibits subsequential convergence to a limit cycle. However, using the previous theorem, we can show that the following modification of the ART, which we shall call the *least squares ART* (LS-ART), converges to the least-squares solution for every x^0:

$$x^k = x^{k-1} + \frac{r^k \cdot a^i}{a^i \cdot Qa^i} a^i.$$

11.4 Conjugate Bases for $\mathbb{R}^J$

If the set $\{v^1, ..., v^J\}$ is a basis for $\mathbb{R}^J$, then any vector x in $\mathbb{R}^J$ can be expressed as a linear combination of the basis vectors; that is, there are real numbers $a_1, ..., a_J$ for which

$$x = a_1 v^1 + a_2 v^2 + ... + a_J v^J.$$

For each x the coefficients a_j are unique. To determine the a_j we write

$$x \cdot v^m = a_1 v^1 \cdot v^m + a_2 v^2 \cdot v^m + ... + a_J v^J \cdot v^m,$$

for $m = 1, ..., J$. Having calculated the quantities $x \cdot v^m$ and $v^j \cdot v^m$, we solve the resulting system of linear equations for the a_j.

If, instead of an arbitrary basis $\{v^1, ..., v^J\}$, we use an orthogonal basis $\{u^1, ..., u^J\}$, that is, $u^j \cdot u^m = 0$, unless $j = m$, then the system of linear equations is trivial to solve. The solution is $a_j = x \cdot u^j / u^j \cdot u^j$, for each j. Of course, we still need to compute the quantities $x \cdot u^j$.

The least-squares solution of the linear system of equations $Ax = b$ is

$$\hat{x} = (A^T A)^{-1} A^T b = Q^{-1} c.$$

To express $\hat{x}$ as a linear combination of the members of an orthogonal basis $\{u^1, ..., u^J\}$ we need the quantities $\hat{x} \cdot u^j$, which usually means that we need to know $\hat{x}$ first. For a special kind of basis, a *Q-conjugate basis*, knowing $\hat{x}$ ahead of time is not necessary; we need only know Q and c. Therefore, we can use such a basis to find $\hat{x}$. This is the essence of the *conjugate gradient method* (CGM), in which we calculate a conjugate basis and, in the process, determine $\hat{x}$.

11.4.1 Conjugate Directions

From Equation (11.2) we have

$$(c - Qx^k) \cdot d^k = 0,$$

which can be expressed as

$$(\hat{x} - x^k) \cdot Qd^k = (\hat{x} - x^k)^T Qd^k = 0.$$

Two vectors x and y are said to be *Q-orthogonal* (or *Q-conjugate*, or just *conjugate*) if $x \cdot Qy = 0$. So, the least-squares solution that we seek lies in a direction from x^k that is Q-orthogonal to d^k. This suggests that we can do better than steepest descent if we take the next direction to be Q-orthogonal to the previous one, rather than just orthogonal. This leads us to *conjugate direction methods*.

Ex. 11.4 *Say that the set* $\{p^1, ..., p^n\}$ *is a conjugate set for* $\mathbb{R}^J$ *if* $p^i \cdot Qp^j = 0$ *for* $i \neq j$. *Prove that a conjugate set that does not contain zero is linearly independent. Show that if* $p^n \neq 0$ *for* $n = 1, ..., J$, *then the least-squares vector* $\hat{x}$ *can be written as*

$$\hat{x} = a_1 p^1 + ... + a_J p^J,$$

with $a_j = c \cdot p^j / p^j \cdot Qp^j$ *for each* j. *Hint: use the* Q*-inner product* $\langle x, y \rangle_Q = x \cdot Qy$.

Therefore, once we have a conjugate basis, computing the least squares solution is trivial. Generating a conjugate basis can obviously be done using the standard Gram-Schmidt approach.

11.4.2 The Gram-Schmidt Method

Let $\mathcal{V} = \{v^1, ..., v^J\}$ be a basis for the space $\mathbb{R}^J$. The Gram-Schmidt method uses the v^j to create an orthogonal basis $\{u^1, ..., u^J\}$ for $\mathbb{R}^J$. The reader may well wonder why we are working hard to get an orthogonal basis for $\mathbb{R}^J$ when the usual basis is available. In our discussion here we assume, simply for notational convenience, that $\mathcal{V}$ is a basis for $\mathbb{R}^J$, but in practice it will be a basis for some subspace, the span of $\mathcal{V}$, not for the entire vector space. Then we want an orthogonal basis for this span, so we must make sure that the elements of the new basis have the same span.

Begin by taking $u^1 = v^1$. For $j = 2, ..., J$, let

$$u^j = v^j - \frac{u^1 \cdot v^j}{u^1 \cdot u^1} u^1 - ... - \frac{u^{j-1} \cdot v^j}{u^{j-1} \cdot u^{j-1}} u^{j-1}.$$

To apply this approach to obtain a conjugate basis, we would simply replace the dot products $u^k \cdot v^j$ and $u^k \cdot u^k$ with the Q-inner products, that is,

$$p^j = v^j - \frac{p^1 \cdot Qv^j}{p^1 \cdot Qp^1} p^1 - ... - \frac{p^{j-1} \cdot Qv^j}{p^{j-1} \cdot Qp^{j-1}} p^{j-1}. \tag{11.5}$$

Even though the Q-inner products can always be written as $x \cdot Qy = Ax \cdot Ay$, so that we need not compute the matrix Q, calculating a conjugate basis using Gram-Schmidt is not practical for large J. There is a way out, fortunately.

11.4.3 Avoiding the Gram-Schmidt Method

If we take $p^1 = v^1$ and $v^j = Qp^{j-1}$, we have a much more efficient mechanism for generating a conjugate basis, namely a three-term recursion formula [200]. The set $\{p^1, Qp^1, ..., Qp^{J-1}\}$ need not be a linearly independent set, in general, but, if our goal is to find $\hat{x}$, and not really to calculate a full conjugate basis, this does not matter, as we shall see.

Theorem 11.3 *Let $p^1 \neq 0$ be arbitrary. Let p^2 be given by*

$$p^2 = Qp^1 - \frac{Qp^1 \cdot Qp^1}{p^1 \cdot Qp^1} p^1,$$

so that $p^2 \cdot Qp^1 = 0$. Then, for $n \geq 2$, let p^{n+1} be given by

$$p^{n+1} = Qp^n - \frac{Qp^n \cdot Qp^n}{p^n \cdot Qp^n} p^n - \frac{Qp^{n-1} \cdot Qp^n}{p^{n-1} \cdot Qp^{n-1}} p^{n-1}. \tag{11.6}$$

Then, the set $\{p^1, ..., p^J\}$ is a conjugate set for $\mathbb{R}^J$. If $p^n \neq 0$ for each n, then the set is a conjugate basis for $\mathbb{R}^J$.

Proof: We consider the induction step of the proof. Assume that $\{p^1, ..., p^n\}$ is a Q-orthogonal set of vectors; we then show that $\{p^1, ..., p^{n+1}\}$ is also, provided that $n \leq J - 1$. It is clear from Equation (11.6) that

$$p^{n+1} \cdot Qp^n = p^{n+1} \cdot Qp^{n-1} = 0.$$

For $j \leq n - 2$, we have

$$p^{n+1} \cdot Qp^j = p^j \cdot Qp^{n+1} = p^j \cdot Q^2 p^n - ap^j \cdot Qp^n - bp^j \cdot Qp^{n-1},$$

for constants a and b. The second and third terms on the right side are then zero because of the induction hypothesis. The first term is also zero since

$$p^j \cdot Q^2 p^n = (Qp^j) \cdot Qp^n = 0$$

because Qp^j is in the span of $\{p^1, ..., p^{j+1}\}$, and so is Q-orthogonal to p^n. ∎

The calculations in the three-term recursion formula Equation (11.6) also occur in the Gram-Schmidt approach in Equation (11.5); the point is that Equation (11.6) uses only the first three terms, in every case.

11.5 The Conjugate Gradient Method

The main idea in the *conjugate gradient method* (CGM) is to build the conjugate set as we calculate the least squares solution using the iterative algorithm

$$x^n = x^{n-1} + \alpha_n p^n. \tag{11.7}$$

The α_n is chosen so as to minimize $f(x^{n-1} + \alpha p^n)$ as a function of α, and so we have

$$\alpha_n = \frac{r^n \cdot p^n}{p^n \cdot Qp^n}, \tag{11.8}$$

where $r^n = c - Qx^{n-1}$.

Ex. 11.5 *Show that*

$$r^{n+1} = r^n - \alpha_n Q p^n, \tag{11.9}$$

so Qp^n is in the span of r^{n+1} and r^n.

Since the function $f(x) = \frac{1}{2}||Ax - b||_2^2$ has for its gradient $\nabla f(x) = A^T(Ax - b) = Qx - c$, the residual vector $r^n = c - Qx^{n-1}$ is the direction of steepest descent from the point $x = x^{n-1}$. The CGM combines the use of the negative gradient directions from the steepest descent method with the use of a conjugate basis of directions, by using the r^n to construct the next direction p^n in such a way as to form a conjugate set $\{p^1, ..., p^J\}$.

As before, there is an efficient recursive formula that provides the next direction: let $p^1 = r^1 = (c - Qx^0)$ and for $j = 2, 3, ...$

$$p^j = r^j - \beta_{j-1} p^{j-1}, \tag{11.10}$$

with

$$\beta_{j-1} = \frac{r^j \cdot Qp^{j-1}}{p^{j-1} \cdot Qp^{j-1}}. \tag{11.11}$$

It follows from the definition of β_{j-1} that

$$p^j Q p^{j-1} = 0. \tag{11.12}$$

Since the α_n is the optimal choice and

$$r^{n+1} = -\nabla f(x^n),$$

we have, according to Equation (11.2),

$$r^{n+1} \cdot p^n = 0. \tag{11.13}$$

Ex. 11.6 *Prove that $r^n = 0$ whenever $p^n = 0$, in which case we have $c = Qx^{n-1}$, so that x^{n-1} is the least-squares solution.*

Ex. 11.7 *Show that $r^n \cdot p^n = r^n \cdot r^n$, so that*

$$\alpha_n = \frac{r^n \cdot r^n}{p^n \cdot Qp^n}. \tag{11.14}$$

In theory, the CGM converges to the least squares solution in finitely many steps, since we either reach $p^{n+1} = 0$ or $n + 1 = J$. With $x^0 = 0$ and

$$x^n = x^{n-1} + \alpha_n p^n, \tag{11.15}$$

for $n = 1, 2, ..., J$, we have $x^J = \hat{x}$, the least squares solution. In practice, the CGM can be employed as a fully iterative method by cycling back through the previously used directions.

An induction proof similar to the one used to prove Theorem 11.3 establishes that the set $\{p^1, ..., p^J\}$ is a conjugate set [200, 211]. In fact, we can say more.

Theorem 11.4 *For $n = 1, 2, ..., J$ and $j = 1, ..., n-1$ we have*

- **a)** $r^n \cdot r^j = 0$*;*
- **b)** $r^n \cdot p^j = 0$*; and*
- **c)** $p^n \cdot Qp^j = 0$.

The proof presented here through a series of exercises is based on that given in [211].

The proof uses induction on the number n. Throughout the following exercises assume that the statements in the theorem hold for some fixed n with $2 \leq n < J$ and for $j = 1, 2, ..., n-1$. We prove that they hold also for $n+1$ and $j = 1, 2, ..., n$.

Ex. 11.8 *Show that $p^n \cdot Qp^n = r^n \cdot Qp^n$, so that*

$$\alpha_n = \frac{r^n \cdot r^n}{r^n \cdot Qp^n}. \tag{11.16}$$

Hints: use Equation (11.10) and the induction assumption concerning **c)** *of the Theorem.*

Ex. 11.9 *Show that $r^{n+1} \cdot r^n = 0$. Hint: use Equations (11.16) and (11.9).*

Ex. 11.10 *Show that $r^{n+1} \cdot r^j = 0$, for $j = 1, ..., n-1$. Hints: write out r^{n+1} using Equation (11.9) and r^j using Equation (11.10), and use the induction hypotheses.*

Ex. 11.11 *Show that $r^{n+1} \cdot p^j = 0$, for $j = 1, ..., n$. Hints: use Equations (11.9) and (11.10) and induction assumptions* **b**) *and* **c**).

Ex. 11.12 *Show that $p^{n+1} \cdot Qp^j = 0$, for $j = 1, ..., n-1$. Hints: use Equation (11.9), the previous exercise, and the induction assumptions.*

The final step in the proof is to show that $p^{n+1} \cdot Qp^n = 0$. But this follows immediately from Equation (11.12).

11.6 Krylov Subspaces

Another approach to deriving the conjugate gradient method is to use Krylov subspaces. If we select $x^0 = 0$ as our starting vector for the CGM, then $p^1 = r^1 = c$, and each p^{n+1} and x^{n+1} lie in the *Krylov subspace* $\mathcal{K}_n(Q, c)$, defined to be the span of the vectors $\{c, Qc, Q^2c, ..., Q^nc\}$.

For any x in $\mathbb{R}^J$, we have

$$\|x - \hat{x}\|_Q^2 = (x - \hat{x})^T Q(x - \hat{x}).$$

Minimizing $\|x - \hat{x}\|_Q^2$ over all x in $\mathcal{K}_n(Q, c)$ is equivalent to minimizing the same function over all x of the form $x = x^n + \alpha p^{n+1}$. This, in turn, is equivalent to minimizing

$$-2\alpha p^{n+1} \cdot r^{n+1} + \alpha^2 p^{n+1} \cdot Qp^{n+1},$$

over all α, which has for its solution the value $\alpha = \alpha_{n+1}$ used to calculate x^{n+1} in the CGM.

11.7 Convergence Issues

The convergence rate of the CGM depends on the condition number of the matrix Q, which is the ratio of its largest to its smallest eigenvalues. When the condition number is much greater than one convergence can be accelerated by *preconditioning* the matrix Q; this means replacing Q with $P^{-1/2}QP^{-1/2}$, for some positive-definite approximation P of Q (see [7]).

11.8 Extending the CGM

There are versions of the CGM for the minimization of nonquadratic functions. In the quadratic case the next conjugate direction p^{n+1} is built from the residual r^{n+1} and p^n. Since, in that case, $r^{n+1} = -\nabla f(x^n)$, this suggests that in the nonquadratic case we build p^{n+1} from $-\nabla f(x^n)$ and p^n. This leads to the Fletcher-Reeves method. Other similar algorithms, such as the Polak-Ribiere and the Hestenes-Stiefel methods, perform better on certain problems [211].

Chapter 12

Regularization

12.1 Chapter Summary

When we use an iterative algorithm, we want it to solve our problem. We also want the solution in a reasonable amount of time, and we want slight errors in the measurements to cause only slight perturbations in the calculated answer. We have already discussed the use of block-iterative methods to accelerate convergence. Now we turn to regularization as a means of reducing sensitivity to noise. Because a number of regularization methods can be derived using a Bayesian *maximum a posteriori* approach, regularization is sometimes treated under the heading of MAP methods; see, for example, [209, 227] and the discussion in [65]. Penalty functions are also used for regularization [134, 2, 3].

12.2 Where Does Sensitivity Come From?

We illustrate the sensitivity problem that can arise when the inconsistent system $Ax = b$ has more equations than unknowns. We take A to be I by J and we calculate the least-squares solution,

$$x_{LS} = (A^\dagger A)^{-1} A^\dagger b, \tag{12.1}$$

assuming that the J by J Hermitian, nonnegative-definite matrix $Q = (A^\dagger A)$ is invertible, and therefore positive-definite.

The matrix Q has the eigenvalue/eigenvector decomposition

$$Q = \lambda_1 u_1 u_1^\dagger + \cdots + \lambda_J u_J u_J^\dagger, \tag{12.2}$$

where the (necessarily positive) eigenvalues of Q are

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_J > 0, \tag{12.3}$$

and the vectors u_j are the corresponding orthonormal eigenvectors.

12.2.1 The Singular-Value Decomposition of A

The square roots $\sqrt{\lambda_j}$ are called the *singular values* of A. The *singular-value decomposition* (SVD) of A is similar to the eigenvalue/eigenvector decomposition of Q: we have

$$A = \sqrt{\lambda_1} u_1 v_1^\dagger + \cdots + \sqrt{\lambda_I} u_J v_J^\dagger, \tag{12.4}$$

where the v_j are particular eigenvectors of $AA^\dagger$. We see from the SVD that the quantities $\sqrt{\lambda_j}$ determine the relative importance of each term $u_j v_j^\dagger$.

The SVD is commonly used for compressing transmitted or stored images. In such cases, the rectangular matrix A is a discretized image. It is not uncommon for many of the lowest singular values of A to be nearly zero, and to be essentially insignificant in the reconstruction of A. Only those terms in the SVD for which the singular values are significant need to be transmitted or stored. The resulting images may be slightly blurred, but can be restored later, as needed.

When the matrix A is a finite model of a linear imaging system, there will necessarily be model error in the selection of A. Getting the dominant terms in the SVD nearly correct is much more important (and usually much easier) than getting the smaller ones correct. The problems arise when we try to invert the system, to solve $Ax = b$ for x.

12.2.2 The Inverse of $Q = A^\dagger A$

The inverse of Q can then be written

$$Q^{-1} = \lambda_1^{-1} u_1 u_1^\dagger + \cdots + \lambda_J^{-1} u_J u_J^\dagger, \tag{12.5}$$

so that, with $A^\dagger b = c$, we have

$$x_{LS} = \lambda_1^{-1}(u_1^\dagger c) u_1 + \cdots + \lambda_J^{-1}(u_J^\dagger c) u_J. \tag{12.6}$$

Because the eigenvectors are orthonormal, we can express $||A^\dagger b||_2^2 = ||c||_2^2$ as

$$||c||_2^2 = |u_1^\dagger c|^2 + \cdots + |u_J^\dagger c|^2, \tag{12.7}$$

and $||x_{LS}||_2^2$ as

$$||x_{LS}||_2^2 = \lambda_1^{-1}|u_1^\dagger c|^2 + \cdots + \lambda_J^{-1}|u_J^\dagger c|^2. \tag{12.8}$$

It is not uncommon for the eigenvalues of Q to be quite distinct, with some of them much larger than the others. When this is the case, we see that $||x_{LS}||_2$ can be much larger than $||c||_2$, because of the presence of the terms involving the reciprocals of the small eigenvalues. When the measurements b are essentially noise-free, we may have $|u_j^\dagger c|$ relatively small, for the indices near J, keeping the product $\lambda_j^{-1}|u_j^\dagger c|^2$ reasonable in size, but when the b becomes noisy, this may no longer be the case. The result is that those terms corresponding to the reciprocals of the smallest eigenvalues dominate the sum for x_{LS} and the norm of x_{LS} becomes quite large. The least-squares solution we have computed is essentially all noise and useless.

In our discussion of the ART, we saw that when we impose a non-negativity constraint on the solution, noise in the data can manifest itself in a different way. When A has more columns than rows, but $Ax = b$ has no non-negative solution, then, at least for those A having the *full-rank property*, the non-negatively constrained least-squares solution has at most $I - 1$ non-zero entries. This happens also with the EMML and SMART solutions (see [70]). As with the ART, regularization can eliminate the problem.

12.2.3 Reducing the Sensitivity to Noise

As we just saw, the presence of small eigenvalues for Q and noise in b can cause $||x_{LS}||_2$ to be much larger than $||A^\dagger b||_2$, with the result that x_{LS} is useless. In this case, even though x_{LS} minimizes $||Ax - b||_2$, it does so by overfitting to the noisy b. To reduce the sensitivity to noise and thereby obtain a more useful approximate solution, we can *regularize* the problem.

It often happens in applications that, even when there is an exact solution of $Ax = b$, noise in the vector b makes such as exact solution undesirable; in such cases a *regularized solution* is usually used instead. Select $\epsilon > 0$ and a vector p that is a prior estimate of the desired solution. Define

$$F_\epsilon(x) = (1 - \epsilon)\|Ax - b\|_2^2 + \epsilon\|x - p\|_2^2. \tag{12.9}$$

Lemma 12.1 *The function F_ϵ always has a unique minimizer $\hat{x}_\epsilon$, given by*

$$\hat{x}_\epsilon = ((1-\epsilon)A^\dagger A + \epsilon I)^{-1}((1-\epsilon)A^\dagger b + \epsilon p); \tag{12.10}$$

this is a regularized solution of $Ax = b$. Here, p is a prior estimate of the desired solution. Note that the inverse above always exists.

Note that, if $p = 0$, then

$$\hat{x}_\epsilon = (A^\dagger A + \gamma^2 I)^{-1} A^\dagger b, \tag{12.11}$$

for $\gamma^2 = \frac{\epsilon}{1-\epsilon}$. The regularized solution has been obtained by modifying the formula for x_{LS}, replacing the inverse of the matrix $Q = A^\dagger A$ with the inverse of $Q + \gamma^2 I$. When ϵ is near zero, so is γ^2, and the matrices Q and $Q + \gamma^2 I$ are nearly equal. What is different is that the eigenvalues of $Q + \gamma^2 I$ are $\lambda_i + \gamma^2$, so that, when the eigenvalues are inverted, the reciprocal eigenvalues are no larger than $1/\gamma^2$, which prevents the norm of x_ϵ from being too large, and decreases the sensitivity to noise.

Lemma 12.2 *Let ϵ be in $(0, 1)$, and let I be the identity matrix whose dimensions are understood from the context. Then*

$$((1-\epsilon)AA^\dagger + \epsilon I)^{-1} A = A((1-\epsilon)A^\dagger A + \epsilon I)^{-1}, \tag{12.12}$$

and, taking conjugate transposes,

$$A^\dagger((1-\epsilon)AA^\dagger + \epsilon I)^{-1} = ((1-\epsilon)A^\dagger A + \epsilon I)^{-1} A^\dagger. \tag{12.13}$$

Proof: Use the identity

$$A((1-\epsilon)A^\dagger A + \epsilon I) = ((1-\epsilon)AA^\dagger + \epsilon I)A. \tag{12.14}$$

∎

Lemma 12.3 *Any vector p in $\mathbb{R}^J$ can be written as $p = A^\dagger q + r$, where $Ar = 0$.*

What happens to $\hat{x}_\epsilon$ as ϵ goes to zero? This will depend on which case we are in:

Case 1: $J \leq I$, and we assume that $A^\dagger A$ is invertible; or

Case 2: $J > I$, and we assume that $AA^\dagger$ is invertible.

Lemma 12.4 *In Case 1, taking limits as $\epsilon \to 0$ on both sides of the expression for $\hat{x}_\epsilon$ gives $\hat{x}_\epsilon \to (A^\dagger A)^{-1}A^\dagger b$, the least squares solution of $Ax = b$.*

We consider Case 2 now. Write $p = A^\dagger q + r$, with $Ar = 0$. Then

$$\hat{x}_\epsilon = A^\dagger((1-\epsilon)AA^\dagger + \epsilon I)^{-1}((1-\epsilon)b + \epsilon q) +$$

$$((1-\epsilon)A^\dagger A + \epsilon I)^{-1}(\epsilon r). \tag{12.15}$$

Lemma 12.5 (a) *We have*

$$((1-\epsilon)A^\dagger A + \epsilon I)^{-1}(\epsilon r) = r, \tag{12.16}$$

for all $\epsilon \in (0,1)$. **(b)** *Taking the limit of $\hat{x}_\epsilon$, as $\epsilon \to 0$, we get $\hat{x}_\epsilon \to A^\dagger(AA^\dagger)^{-1}b + r$. This is the solution of $Ax = b$ closest to p.*

Proof: For part (a) let

$$t_\epsilon = ((1-\epsilon)A^\dagger A + \epsilon I)^{-1}(\epsilon r). \tag{12.17}$$

Then, multiplying by A gives

$$At_\epsilon = A((1-\epsilon)A^\dagger A + \epsilon I)^{-1}(\epsilon r). \tag{12.18}$$

Now show that $At_\epsilon = 0$. For part (b) draw a diagram for the case of one equation in two unknowns. ∎

12.3 Iterative Regularization

It is often the case that the entries of the vector b in the system $Ax = b$ come from measurements, so are usually noisy. If the entries of b are noisy but the system $Ax = b$ remains consistent (which can easily happen in the under-determined case, with $J > I$), the ART begun at $x^0 = 0$ converges to the solution having minimum norm, but this norm can be quite large. The resulting solution is probably useless. Instead of solving $Ax = b$, we *regularize* by minimizing, for example, the function $F_\epsilon(x)$ given in Equation (12.9). For the case of $p = 0$, the solution to this problem is the vector $\hat{x}_\epsilon$ in Equation (12.11). However, we do not want to calculate $A^\dagger A + \gamma^2 I$, in order to solve

$$(A^\dagger A + \gamma^2 I)x = A^\dagger b, \tag{12.19}$$

when the matrix A is large. Fortunately, there are ways to find $\hat{x}_\epsilon$, using only the matrix A. We later we shall see how this might be accomplished using the ART; now we show how the Landweber algorithm can be used to calculate this regularized solution.

12.3.1 Regularizing Landweber's Algorithm

Our goal is to minimize the function in Equation (12.9), with $p = 0$. Notice that this is equivalent to minimizing the function

$$F(x) = ||Bx - c||_2^2, \tag{12.20}$$

for

$$B = \begin{bmatrix} A \\ \gamma I \end{bmatrix}, \tag{12.21}$$

and

$$c = \begin{bmatrix} b \\ 0 \end{bmatrix}, \tag{12.22}$$

where 0 denotes a column vector with all entries equal to zero and $\gamma = \frac{\epsilon}{1-\epsilon}$. The Landweber iteration for the problem $Bx = c$ is

$$x^{k+1} = x^k + \alpha B^T(c - Bx^k), \tag{12.23}$$

for $0 < \alpha < 2/\rho(B^TB)$, where $\rho(B^TB)$ is the spectral radius of B^TB. Equation (12.23) can be written as

$$x^{k+1} = (1 - \alpha\gamma^2)x^k + \alpha A^T(b - Ax^k). \tag{12.24}$$

Part IV

Appendices

Chapter 13

Appendix: Linear Algebra

13.1 Chapter Summary

Linear algebra is the study of linear transformations between vector spaces. Although the subject is not simply matrix theory, there is a close connection, stemming from the role of matrices in representing linear transformations. Throughout this section we shall limit discussion to finite-dimensional vector spaces.

13.2 Representing a Linear Transformation

Let $\mathcal{A} = \{a^1, a^2, ..., a^N\}$ be a basis for the finite-dimensional complex vector space V. As we saw previously, once a basis for V is specified, there is a natural association, an *isomorphism*, between V and the vector space

$\mathbb{C}^N$ of N-dimensional column vectors with complex entries. Any vector v in V can be written as

$$v = \sum_{n=1}^{N} \gamma_n a^n. \tag{13.1}$$

The column vector $\gamma = (\gamma_1, ..., \gamma_N)^T$ is uniquely determined by v and the basis $\mathcal{A}$ and we denote it by $[v]_{\mathcal{A}}$. Notice that the ordering of the list of members of $\mathcal{A}$ matters, so we shall always assume that the ordering has been fixed.

Let W be a second finite-dimensional vector space, and let T be any linear transformation from V to W. Let $\mathcal{B} = \{b^1, b^2, ..., b^M\}$ be a basis for W. For $n = 1, ..., N$, let

$$Ta^n = A_{1n}b^1 + A_{2n}b^2 + ... + A_{Mn}b^M. \tag{13.2}$$

Then the M by N matrix A having the A_{mn} as entries is said to *represent* T, with respect to the bases $\mathcal{A}$ and $\mathcal{B}$, and we write $A = [T]^{\mathcal{B}}_{\mathcal{A}}$. We then have

$$[Tv]_{\mathcal{B}} = A[v]_{\mathcal{A}}.$$

Suppose that V, W and Z are vector spaces, with bases $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, respectively. Suppose also that T is a linear transformation from V to W and U is a linear transformation from W to Z. Let A represent T with respect to the bases $\mathcal{A}$ and $\mathcal{B}$, and let B represent U with respect to the bases $\mathcal{B}$ and $\mathcal{C}$. Then the matrix BA represents the linear transformation UT with respect to the bases $\mathcal{A}$ and $\mathcal{C}$.

13.3 Linear Operators on V

When $W = V$, we say that the linear transformation T is a *linear operator* on V. In this case, we can also take the basis $\mathcal{B}$ to be $\mathcal{A}$, and say that the matrix A represents the linear operator T, with respect to the basis $\mathcal{A}$. We then write $A = [T]_{\mathcal{A}}$.

Ex. 13.1 *Suppose that $\tilde{\mathcal{A}}$ is a second basis for V. Let T be any linear operator on V and $\tilde{A} = [T]_{\tilde{\mathcal{A}}}$. Show that there is a unique invertible N by N matrix Q having the property that, for all T, the matrix $\tilde{A} = QAQ^{-1}$, so we can write*

$$[T]_{\tilde{\mathcal{A}}} = Q[T]_{\mathcal{A}}Q^{-1}.$$

Hint: the matrix Q is the change-of-basis matrix*, which means that Q represents the identity operator I, with respect to the bases $\mathcal{A}$ and $\tilde{\mathcal{A}}$; that is, $Q = [I]^{\tilde{\mathcal{A}}}_{\mathcal{A}}$.*

Ex. 13.2 *Let T be a linear operator on the finite-dimensional vector space V with basis $\mathcal{A} = \{a^1, a^2, ..., a^N\}$. Let W be the subspace of V spanned by the elements $\{a^1, ..., a^M\}$, where $M < N$. Suppose that W is T-invariant, that is, $Tw \in W$ for every $w \in W$. What can then be said about the representing matrix $A = [T]_{\mathcal{A}}$?*

13.4 Linear Operators on $\mathbb{C}^N$

Let $\mathcal{A}$ be the usual basis for the vector space $V = \mathbb{C}^N$. In practice, we make no distinction between a member x of $\mathbb{C}^N$ and $[x]_{\mathcal{A}}$; that is, we use the equation

$$x = [x]_{\mathcal{A}}$$

without comment. If T is a linear operator on $\mathbb{C}^N$ and $A = [T]_{\mathcal{A}}$, then from

$$[Tx]_{\mathcal{A}} = A[x]_{\mathcal{A}}$$

we write

$$Tx = Ax;$$

in other words, we make no distinction between T and A and say that every linear operator on $\mathbb{C}^N$ is multiplication by a matrix. Of course, all of this presupposes that $\mathcal{A}$ is the usual basis for $\mathbb{C}^N$; if we change the basis, then the distinctions again become necessary.

13.5 Similarity and Equivalence of Matrices

Let $\mathcal{A}$ and $\tilde{\mathcal{A}} = \{\tilde{a}^1, ..., \tilde{a}^N\}$ be bases for V, and $\mathcal{B}$ and $\tilde{\mathcal{B}} = \{\tilde{b}^1, ..., \tilde{b}^M\}$ be bases for W. Let $Q = [I]_{\mathcal{A}}^{\tilde{\mathcal{A}}}$ and $R = [I]_{\mathcal{B}}^{\tilde{\mathcal{B}}}$ be the change-of-bases matrices in V and W, respectively. As we just saw, for any linear operator T on V, the matrices $\tilde{A} = [T]_{\tilde{\mathcal{A}}}$ and $A = [T]_{\mathcal{A}}$ are related according to

$$A = Q^{-1}\tilde{A}Q. \tag{13.3}$$

We describe the relationship in Equation (13.3) by saying that the matrices A and $\tilde{A}$ are *similar*.

Definition 13.1 *Two N by N matrices A and B are said to be* similar *if there is an invertible matrix Q such that $A = Q^{-1}BQ$.*

Ex. 13.3 *Show that similar matrices have the same eigenvalues.*

Let S be a linear transformation from V to W. Then we have

$$[S]_{\mathcal{A}}^{\mathcal{B}} = R^{-1}[S]_{\tilde{\mathcal{A}}}^{\tilde{\mathcal{B}}}Q. \tag{13.4}$$

With $G = [S]_{\mathcal{A}}^{\mathcal{B}}$ and $\tilde{G} = [S]_{\tilde{\mathcal{A}}}^{\tilde{\mathcal{B}}}$, we have

$$G = R^{-1}\tilde{G}Q. \tag{13.5}$$

Definition 13.2 *Two M by N matrices A and B are said to be* equivalent *if there are invertible matrices P and Q such that $B = PAQ$.*

We can therefore describe the relationship in Equation (13.5) by saying that the matrices G and $\tilde{G}$ are equivalent.

Ex. 13.4 *Show that A and B are equivalent if B can be obtained from A by means of elementary row and column operations.*

Ex. 13.5 *Prove that two equivalent matrices A and B must have the same rank, and so two similar matrices must also have the same rank. Hint: use the fact that Q is invertible to show that A and AQ have the same rank.*

Ex. 13.6 *Prove that any two M by N matrices with the same rank r are equivalent. Hints: Let A be an M by N matrix, which we can also view as inducing, by multiplication, a linear transformation T from $V = \mathbb{C}^N$ to $W = \mathbb{C}^M$. Therefore, A represents T in the usual bases of $\mathbb{C}^N$ and $\mathbb{C}^M$. Now construct a basis $\mathcal{A}$ for $\mathbb{C}^N$, such that*

$$\mathcal{A} = \{a^1, ..., a^N\},$$

with $\{a^{r+1}, ..., a^N\}$ forming a basis for the null space of A. Show that the set $\{Aa^1, ..., Aa^r\}$ is linearly independent and can therefore be extended to a basis $\mathcal{B}$ for $\mathbb{C}^M$. Show that the matrix D that represents T with respect to the bases $\mathcal{A}$ and $\mathcal{B}$ is the M by N matrix with the r by r identity matrix in the upper left corner, and all the other entries are zero. Since A is then equivalent to this matrix D, so is the matrix B; therefore A and B are equivalent to each other. Another way to say this is that both A and B can be reduced to D using elementary row and column operations.

13.6 Linear Functionals and Duality

We turn now to the particular case in which the second vector space W is just the space $\mathbb{C}$ of complex numbers. Any linear transformation f

from V to $\mathbb{C}$ is called a *linear functional*. The space of all linear functionals on V is denoted V^* and called the *dual space* of V. The set V^* is itself a finite-dimensional vector space, so it too has a dual space, $(V^*)^* = V^{**}$, the second dual space, which is the set of all linear transformations F from V^* to $\mathbb{C}$.

Ex. 13.7 *Show that the dimension of V^* is the same as that of V. Hint: let $\mathcal{A} = \{a^1, ..., a^N\}$ be a basis for V, and for each $m = 1, ..., N$, let $f^m(a^n) = 0$, if $m \neq n$, and $f^m(a^m) = 1$. Show that the collection $\{f^1, ..., f^N\}$ is a basis for V^*.*

Proposition 13.1 *Let V be a vector space of dimension N and S a subspace of V. Then the dimension of S is $N - 1$ if and only if there is a non-zero member f of V^* such that $S = \{v | f(v) = 0\}$.*

Proof: Let S have dimension $M < N$ and let $\{u^1, u^2, ..., u^M\}$ be a basis for S. Extend this basis for S to a basis for V, denoted

$$\{u^1, u^2, ..., u^M, v^1, v^2, ..., v^{N-M}\}.$$

Now suppose that the dimension of S is $M = N - 1$, and that the enlarged basis has only one new member, v^1. Every vector v in V can be written uniquely as

$$v = a_1 u^1 + a_2 u^2 + ... + a_{N-1} u^{N-1} + a_N v^1.$$

Let $f(v) = a_N$; then f is a member of V^* and $S = \{v | f(v) = 0\}$.

Conversely, suppose now that $S = \{v | f(v) = 0\}$, and its dimension is $M < N - 1$. Then the enlarged basis has at least two new members, v^1 and v^2, neither of them in S. Therefore $\alpha_1 = f(v^1)$ and $\alpha_2 = f(v^2)$ are not zero. We then have $f(v) = 0$ for the vector $v = \alpha_2 v^1 - \alpha_1 v^2$, which means that v is in S. But v is a linear combination of v^1 and v^2, and therefore, because of the linear independence of the members of the enlarged basis, cannot also be a linear combination of the u^m, for $m = 1, 2, ..., M$. ∎

There is a natural identification of V^{**} with V itself. For each v in V, define $J_v(f) = f(v)$ for each f in V^*. Then it is easy to establish that J_v is in V^{**} for each v in V. The set J_V of all members of V^{**} of the form J_v for some v is a subspace of V^{**}.

Ex. 13.8 *Show that the subspace J_V has the same dimension as V^{**} itself, so that it must be all of V^{**}.*

In the previous exercise we established that $J_V = V^{**}$ by showing that these spaces have the same dimension. We can also prove this result in a more direct way. Let F be any member of V^{**}. We show that there is a v in V such that $F(f) = f(v)$ for all f in V^* by displaying v explicitly. Let

$\gamma_n = F(f^n)$, for $n = 1, 2, ..., N$, where f^n are as defined in Exercise 13.7. Then let $v = \gamma_1 a^1 + \gamma_2 a^2 + ... + \gamma_N a^N$. Let f be arbitrary in V^*, written in terms of the basis as

$$f = \alpha_1 f^1 + \alpha_2 f^2 + ... + \alpha_N f^N,$$

so that

$$f(v) = \alpha_1 f^1(v) + \alpha_2 f^2(v) + ... + \alpha_N f^N(v) = \alpha_1\gamma_1 + \alpha_2\gamma_2 + ... + \alpha_N\gamma_N.$$

Then

$$F(f) = \alpha_1 F(f^1) + \alpha_2 F(f^2) + ... + \alpha_N F(f^N) = \alpha_1\gamma_1 + \alpha_2\gamma_2 + ... + \alpha_N\gamma_N = f(v).$$

We shall see later that once V has been endowed with an inner product, there is a simple way to describe every linear functional on V: for each f in V^* there is a unique vector v_f in V with $f(v) = \langle v, v_f \rangle$, for each v in V. As a result, we have an identification of V^* with V itself.

13.7 Diagonalization

Let $T : V \to V$ be a linear operator, $\mathcal{A}$ a basis for V, and $A = [T]_{\mathcal{A}}$. As we change the basis, the matrix representing T also changes. We wonder if it is possible to find some basis $\mathcal{B}$ such that $B = [T]_{\mathcal{B}}$ is a diagonal matrix L. Let $P = [I]_{\mathcal{B}}^{\mathcal{A}}$ be the change-of basis matrix from $\mathcal{B}$ to $\mathcal{A}$. We would then have $P^{-1}AP = L$, or $A = PLP^{-1}$. When this happens, we say that A has been *diagonalized* by P. According to Lemma 5.2, A is diagonalizable if all its eigenvalues are distinct.

Suppose that the basis $\mathcal{B} = \{b^1, ..., b^N\}$ is such that $B = [T]_{\mathcal{B}} = L$, where L is the diagonal matrix $L = \text{diag}\,\{\lambda_1, ..., \lambda_N\}$. Then we have $AP = PL$, which tells us that p^n, the n-th column of P, is an eigenvector of the matrix A, with λ_n as its eigenvalue. Since $p^n = [b^n]_{\mathcal{A}}$, we have

$$0 = (A - \lambda_n I)p^n = (A - \lambda_n I)[b^n]_{\mathcal{A}} = [(T - \lambda_n I)b^n]_{\mathcal{A}},$$

from which we conclude that

$$(T - \lambda_n I)b^n = 0,$$

or

$$Tb^n = \lambda_n b^n;$$

therefore, b^n is an eigenvector of the linear operator T.

13.8 Using Matrix Representations

The matrix A has eigenvalues λ_n, $n = 1, ..., N$, precisely when these λ_n are the roots of the *characteristic polynomial*

$$P(\lambda) = \det(A - \lambda I).$$

We would like to be able to define the characteristic polynomial of T itself to be $P(\lambda)$; the problem is that we do not yet know that different matrix representations of T have the same characteristic polynomial, although we do know that, since they are similar matrices, they have the same eigenvalues.

Ex. 13.9 *Use the fact that det(GH)=det(G)det(H) for any square matrices G and H to show that*

$$\det([T]_{\mathcal{B}} - \lambda I) = \det([T]_{\mathcal{C}} - \lambda I),$$

for any bases $\mathcal{B}$ and $\mathcal{C}$ for V.

13.9 An Inner Product on V

For any two column vectors $x = (x_1, ..., x_N)^T$ and $y = (y_1, ..., y_n)^T$ in $\mathbb{C}^N$, their *complex dot product* is defined by

$$x \cdot y = \sum_{n=1}^{N} x_n \overline{y_n} = y^\dagger x,$$

where $y^\dagger$ is the *conjugate transpose* of the vector y, that is, $y^\dagger$ is the row vector with entries $\overline{y_n}$.

The association of the elements v in V with the complex column vector $[v]_{\mathcal{A}}$ can be used to obtain an *inner product* on V. For any v and w in V, define

$$\langle v, w \rangle = [v]_{\mathcal{A}} \cdot [w]_{\mathcal{A}}, \tag{13.6}$$

where the right side is the ordinary complex dot product in $\mathbb{C}^N$. Note that, with respect to this inner product, the basis $\mathcal{A}$ becomes an orthonormal basis.

For particular vector spaces V we may want to define an inner product that conforms well to the special nature of the elements of V. For example, suppose that V is the vector space of all N by N complex matrices. This space has dimension N^2. A basis for this space is the collection of all N by N matrices that have a one in a single entry and zero everywhere else. The induced inner product that we get using this basis can be described in another way: it is $\langle A, B\rangle = \text{trace}(B^\dagger A)$. The resulting norm of A is the *Frobenius norm*.

13.10 Orthogonality

Two vectors v and w in the inner-product space V are said to be *orthogonal* if $\langle v, w\rangle = 0$. A basis $\mathcal{U} = \{u^1, u^2, ..., u^N\}$ is called an *orthogonal basis* if every two vectors in $\mathcal{U}$ are orthogonal, and *orthonormal* if, in addition, $\|u^n\| = 1$, for each n.

Ex. 13.10 *Let $\mathcal{U}$ and $\mathcal{V}$ be orthonormal bases for the inner-product space V, and let Q be the change-of-basis matrix satisfying*

$$[v]_{\mathcal{U}} = Q[v]_{\mathcal{V}}.$$

Show that $Q^{-1} = Q^\dagger$, so that Q is a unitary matrix.

13.11 Representing Linear Functionals

Let $f : V \to \mathbb{C}$ be a linear functional on the inner-product space V and let $\mathcal{A} = \{a^1, ..., a^N\}$ be an orthonormal basis for V. Let v_f be the member of V defined by

$$v_f = \sum_{m=1}^{N} \overline{f(a^m)} a^m.$$

Then for each

$$v = \sum_{n=1}^{N} \alpha_n a^n,$$

in V, we have

$$\langle v, v_f\rangle = \sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n f(a^m)\langle a^n, a^m\rangle$$

$$= \sum_{n=1}^{N} \alpha_n f(a^n) = f(\sum_{n=1}^{N} \alpha_n a^n) = f(v).$$

So we see that once V has been given an inner product, each linear functional f on V can be thought of as corresponding to a vector v_f in V, so that

$$f(v) = \langle v, v_f \rangle.$$

Ex. 13.11 *Show that the vector v_f associated with the linear functional f is unique by showing that*

$$\langle v, y \rangle = \langle v, w \rangle,$$

for every v in V implies that $y = w$.

13.12 Adjoint of a Linear Transformation

If T is a linear operator on an inner product space V, we say that T is *self-adjoint* if $\langle Tu, v \rangle = \langle u, Tv \rangle$, for all u and v in V. This definition allows us to speak of self-adjoint linear operators before we have introduced the adjoint of a linear operator, the topic of this section.

Let $T : V \to W$ be a linear transformation from a vector space V to a vector space W. The *adjoint* of T is the linear operator $T^* : W^* \to V^*$ defined by

$$(T^* g)(v) = g(Tv), \tag{13.7}$$

for each $g \in W^*$ and $v \in V$.

Once V and W have been given inner products, and V^* and W^* have been identified with V and W, respectively, the operator T^* can be defined as a linear operator from W to V as follows. Let $T : V \to W$ be a linear transformation from an inner-product space V to an inner-product space W. For each fixed w in W, define a linear functional f on V by

$$f(v) = \langle Tv, w \rangle.$$

By our earlier discussion, f has an associated vector v_f in V such that

$$f(v) = \langle v, v_f \rangle.$$

Therefore,

$$\langle Tv, w \rangle = \langle v, v_f \rangle,$$

for each v in V. The *adjoint* of T is the linear transformation T^* from W to V defined by $T^*w = v_f$.

When $W = V$, and T is a linear operator on V, then so is T^*. In this case, we can ask whether or not $T^*T = TT^*$, that is, whether or not T is *normal*, and whether or not $T = T^*$, that is, whether or not T is *self-adjoint*.

Ex. 13.12 *Let $\mathcal{U}$ be an orthonormal basis for the inner-product space V and T a linear operator on V. Show that*

$$[T^*]_{\mathcal{U}} = ([T]_{\mathcal{U}})^\dagger. \tag{13.8}$$

13.13 Normal and Self-Adjoint Operators

Let T be a linear operator on an inner-product space V. We say that T is *normal* if $T^*T = TT^*$, and *self-adjoint* if $T^* = T$. A square matrix A is said to be *normal* if $A^\dagger A = AA^\dagger$, and *Hermitian* if $A^\dagger = A$.

Ex. 13.13 *Let $\mathcal{U}$ be an orthonormal basis for the inner-product space V. Show that T is normal if and only if $[T]_{\mathcal{U}}$ is a normal matrix, and T is self-adjoint if and only if $[T]_{\mathcal{U}}$ is Hermitian. Hint: use Exercise (**??**).*

Ex. 13.14 *Compute the eigenvalues for the real square matrix*

$$A = \begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}. \tag{13.9}$$

Note that the eigenvalues are complex, even though the entries of A are real. The matrix A is not Hermitian.

Ex. 13.15 *Show that the eigenvalues of the complex matrix*

$$B = \begin{bmatrix} 1 & 2+i \\ 2-i & 1 \end{bmatrix} \tag{13.10}$$

are the real numbers $\lambda = 1 + \sqrt{5}$ and $\lambda = 1 - \sqrt{5}$, with corresponding eigenvectors $u = (\sqrt{5}, 2-i)^T$ and $v = (\sqrt{5}, i-2)^T$, respectively.

Ex. 13.16 *Show that the eigenvalues of the real matrix*

$$C = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \tag{13.11}$$

are both equal to one, and that the only eigenvectors are non-zero multiples of the vector $(1, 0)^T$. Compute C^TC and CC^T. Are they equal?

13.14 It is Good to be "Normal"

For a given linear operator on V, when does there exist an orthonormal basis for V consisting of eigenvectors of T? The answer is: When T is normal.

Consider an N by N matrix A. We use A to define a linear operator T on the space of column vectors $V = \mathbb{C}^N$ by $Tv = Av$, that is, the operator T works by multiplying each column vector v in $\mathbb{C}^N$ by the matrix A. Then A represents T with respect to the usual orthonormal basis $\mathcal{A}$ for $\mathbb{C}^N$. Suppose now that there is an orthonormal basis $\mathcal{U} = \{u^1, ..., u^N\}$ for $\mathbb{C}^N$ such that

$$Tu^n = Au^n = \lambda_n u^n,$$

for each n. The matrix representing T in the basis $\mathcal{U}$ is the matrix $B = Q^{-1}AQ$, where Q is the change-of-basis matrix with

$$Q[v]_{\mathcal{U}} = [v]_{\mathcal{A}}.$$

But we also know that B is the diagonal matrix $B = L = \text{diag}(\lambda_1, ..., \lambda_N)$. Therefore, $L = Q^{-1}AQ$, or $A = QLQ^{-1}$.

As we saw in Exercise (13.10), the matrix Q is unitary, that is, $Q^{-1} = Q^\dagger$. Therefore, $A = QLQ^\dagger$. Then we have

$$A^\dagger A = QL^\dagger Q^\dagger QLQ^\dagger = QL^\dagger LQ^\dagger$$

$$= QLL^\dagger Q^\dagger = QLQ^\dagger QL^\dagger Q^\dagger = AA^\dagger,$$

so that

$$A^\dagger A = AA^\dagger,$$

and A is normal.

Two fundamental results in linear algebra are the following, which we discuss in more detail in the chapter "Hermitian and Normal Linear Operators".

Theorem 13.1 *For a linear operator T on a finite-dimensional complex inner-product space V there is an orthonormal basis of eigenvectors if and only if T is normal.*

Corollary 13.1 *A self-adjoint linear operator T on a finite-dimensional complex inner-product space V has an orthonormal basis of eigenvectors.*

Ex. 13.17 *Show that the eigenvalues of a self-adjoint linear operator T on a finite-dimensional complex inner-product space are real numbers. Hint: consider $Tu = \lambda u$, and begin with $\lambda\langle u, u\rangle = \langle Tu, u\rangle$.*

Combining the various results obtained so far, we can conclude the following.

Corollary 13.2 *Let T be a linear operator on a finite-dimensional real inner-product space V. Then V has an orthonormal basis consisting of eigenvectors of T if and only if T is self-adjoint.*

13.15 Bases and Inner Products

Throughout this section V will denote a finite-dimensional real or complex vector space. We know that it is always possible to find a basis for V; we simply build up a set of linearly independent vectors until including any additional vector will render the set linearly dependent. As we have seen, once we have a basis for V it is a simple matter to use that basis to induce an inner product on V. In this section we make several assertions without proof; the proofs are left as exercises for the reader.

Let $\mathcal{A} = \{a^1, ..., a^N\}$ be a basis for V. Each vector x in V can then be written uniquely as a linear combination of the members of $\mathcal{A}$:

$$x = \alpha_1 a^1 + ... + \alpha_N a^N.$$

The column vector $\alpha = (\alpha_1, ..., \alpha_N)^T$ is then denoted $[x]_{\mathcal{A}}$. We denote by $F_{\mathcal{A}}$ the linear transformation $F_{\mathcal{A}} : V \to \mathbb{C}^N$ that associates with each x in V the column vector $[x]_{\mathcal{A}}$, and by $E_{\mathcal{A}}$ the linear transformation $E_{\mathcal{A}} : \mathbb{C}^N \to V$ that associates with each vector α in $\mathbb{C}^N$ the member of V given by

$$x = \alpha_1 a^1 + ... + \alpha_N a^N.$$

Note that $E_{\mathcal{A}}$ is the inverse of $F_{\mathcal{A}}$.

The inner product on V induced by the basis $\mathcal{A}$ is

$$\langle x, y \rangle_{\mathcal{A}} = [x]_{\mathcal{A}} \cdot [y]_{\mathcal{A}},$$

which can also be written as

$$\langle x, y \rangle_{\mathcal{A}} = F_{\mathcal{A}} x \cdot F_{\mathcal{A}} y.$$

The basis $\mathcal{A}$ is orthonormal with respect to this inner product. We denote by $V_{\mathcal{A}}$ the vector space V with the inner product $\langle x, y \rangle_{\mathcal{A}}$.

The adjoint of $F_{\mathcal{A}}$ is the linear transformation $F^*_{\mathcal{A}} : \mathbb{C}^N \to V_{\mathcal{A}}$ for which

$$\langle F^*_{\mathcal{A}} \alpha, y \rangle_{\mathcal{A}} = \alpha \cdot F_{\mathcal{A}} y,$$

for all α in $\mathbb{C}^N$ and y in V. But we also have

$$\langle F_{\mathcal{A}}^* \alpha, y\rangle_{\mathcal{A}} = F_{\mathcal{A}} F_{\mathcal{A}}^* \alpha \cdot F_{\mathcal{A}} y.$$

It follows that

$$F_{\mathcal{A}} F_{\mathcal{A}}^* = I.$$

Therefore,

$$F_{\mathcal{A}}^* = E_{\mathcal{A}}.$$

Let $\mathcal{B} = \{b^1, ..., b^N\}$ be a second basis for V. The change-of-basis matrix $Q = [I]_{\mathcal{A}}^{\mathcal{B}}$ has the property

$$[x]_{\mathcal{B}} = Q[x]_{\mathcal{A}},$$

or

$$F_{\mathcal{B}} x = Q F_{\mathcal{A}} x,$$

for all x in V. Therefore we can write

$$F_{\mathcal{B}} = Q F_{\mathcal{A}},$$

so that

$$Q = F_{\mathcal{B}} E_{\mathcal{A}}.$$

Ex. 13.18 *Viewing $F_{\mathcal{B}}$ as a linear transformation from the inner product space $V_{\mathcal{A}}$ to $\mathbb{C}^N$, show that the adjoint of $F_{\mathcal{B}}$ is the linear transformation $F'_{\mathcal{B}}$ given by $F'_{\mathcal{B}} = E_{\mathcal{A}} Q^\dagger$.*

Then we have

$$\langle x, y\rangle_{\mathcal{B}} = F_{\mathcal{B}} x \cdot F_{\mathcal{B}} y = Q F_{\mathcal{A}} x \cdot Q F_{\mathcal{A}} y = Q^\dagger Q F_{\mathcal{A}} x \cdot F_{\mathcal{A}} y.$$

Writing

$$H = Q^\dagger Q = F_{\mathcal{A}} F'_{\mathcal{B}} F_{\mathcal{B}} E_{\mathcal{A}},$$

where $F'_{\mathcal{B}} = E_{\mathcal{A}} Q^\dagger$ is the adjoint of the linear transformation $F_{\mathcal{B}}$, with respect to the vector space $V_{\mathcal{A}}$, we have

$$\langle x, y\rangle_{\mathcal{B}} = H F_{\mathcal{A}} x \cdot F_{\mathcal{A}} y.$$

The matrix H is hermitian and positive-definite.

Now let S be the linear transformation on V for which $H = [S]_{\mathcal{A}}$. This means that

$$H F_{\mathcal{A}} x = F_{\mathcal{A}} S x,$$

for all x in V. Then we can get an explicit description of S;

$$S = E_{\mathcal{A}} H F_{\mathcal{A}} = E_{\mathcal{A}} Q^\dagger Q F_{\mathcal{A}}.$$

This tells us that for any other basis $\mathcal{B}$ the associated inner product can be expressed in terms of the inner product from $\mathcal{A}$ by

$$\langle x, y\rangle_{\mathcal{B}} = \langle Sx, y\rangle_{\mathcal{A}}.$$

The linear operator S is self-adjoint and positive-definite on the inner product space $V_{\mathcal{A}}$.

If T is any self-adjoint, positive-definite linear operator on $V_{\mathcal{A}}$ then T induces another inner product, denoted $\langle x, y\rangle_T$, by

$$\langle x, y\rangle_T = \langle Tx, y\rangle_{\mathcal{A}}.$$

We also know that $V_{\mathcal{A}}$ has an orthonormal basis $\{u^1, ..., u^N\}$ of eigenvectors of T, with $Tu^n = \lambda_n u^n$. Let $b^n = \frac{1}{\sqrt{\lambda_n}} u^n$. Then the family $\mathcal{B} = \{b^1, ..., b^N\}$ is another basis for V and

$$\langle x, y\rangle_T = \langle x, y\rangle_{\mathcal{B}}.$$

If we begin with a vector space V that already has an inner product $\langle x, y\rangle$, then

$$\langle x, y\rangle = \langle x, y\rangle_{\mathcal{A}},$$

for any orthonormal basis $\mathcal{A}$.

We can summarize our findings as follows:

- 1. Any inner product $\langle x, y\rangle$ on V is $\langle x, y\rangle_{\mathcal{A}}$, for any orthonormal basis $\mathcal{A}$;
- 2. Any basis $\mathcal{A}$ induces an inner product, $\langle x, y\rangle_{\mathcal{A}}$;
- 3. If $\mathcal{A}$ and $\mathcal{B}$ are any two bases for V, then

  $$\langle x, y\rangle_{\mathcal{B}} = \langle Sx, y\rangle_{\mathcal{A}},$$

  for some self-adjoint, positive definite linear operator S on $V_{\mathcal{A}}$;
- 4. If T is any self-adjoint positive-definite linear operator on $V_{\mathcal{A}}$, then T induces an inner product

  $$\langle x, y\rangle_T = \langle Tx, y\rangle,$$

  and there is a basis $\mathcal{B}$ such that

  $$\langle x, y\rangle_T = \langle x, y\rangle_{\mathcal{B}}.$$

Chapter 14

Appendix: More ART and MART

14.1 Chapter Summary

Although the ART and the MART were developed to compute tomographic images, they can be viewed more generally as iterative methods for solving systems of linear equations.

14.2 The ART in the General Case

Let A be a complex matrix with I rows and J columns, and let b be a member of $\mathbb{C}^I$. We want to solve the system $Ax = b$. For each index value i, let H_i be the hyperplane of J-dimensional vectors given by

$$H_i = \{x | (Ax)_i = b_i\}, \tag{14.1}$$

and P_i the orthogonal projection operator onto H_i. Let x^0 be arbitrary and, for each nonnegative integer k, let $i(k) = k(\mathrm{mod}\, I) + 1$. The iterative step of the ART is

$$x^{k+1} = P_{i(k)} x^k. \tag{14.2}$$

Because the ART uses only a single equation at each step, it has been called a *row-action* method.

14.2.1 Calculating the ART

Given any vector z the vector in H_i closest to z, in the sense of the Euclidean distance, has the entries

$$x_j = z_j + \overline{A_{ij}}(b_i - (Az)_i) / \sum_{m=1}^{J} |A_{im}|^2. \tag{14.3}$$

To simplify our calculations, we shall assume, throughout this chapter, that the rows of A have been rescaled to have Euclidean length one; that is

$$\sum_{j=1}^{J} |A_{ij}|^2 = 1, \tag{14.4}$$

for each $i = 1, ..., I$, and that the entries of b have been rescaled accordingly, to preserve the equations $Ax = b$. The ART is then the following: begin with an arbitrary vector x^0; for each nonnegative integer k, having found x^k, the next iterate x^{k+1} has entries

$$x_j^{k+1} = x_j^k + \overline{A_{ij}}(b_i - (Ax^k)_i). \tag{14.5}$$

As we shall show shortly, when the system $Ax = b$ has exact solutions the ART converges to the solution closest to x^0, in the 2-norm. How fast the algorithm converges will depend on the ordering of the equations and on whether or not we use relaxation. In selecting the equation ordering, the important thing is to avoid particularly bad orderings, in which the hyperplanes H_i and H_{i+1} are nearly parallel.

14.2.2 Full-cycle ART

We again consider the *full-cycle* ART, with iterative step $z^{m+1} = Tz^m$, for

$$T = P_I P_{I-1} \cdots P_2 P_1. \tag{14.6}$$

When the system $Ax = b$ has solutions, the fixed points of T are solutions. When there are no solutions of $Ax = b$, the operator T will still have fixed points, but they will no longer be exact solutions.

14.2.3 Relaxed ART

The ART employs orthogonal projections onto the individual hyperplanes. If we permit the next iterate to fall short of the hyperplane, or somewhat beyond it, we get a relaxed version of ART.The relaxed ART algorithm is as follows:

Algorithm 14.1 (Relaxed ART) *With $\omega \in (0, 2)$, x^0 arbitrary, and $i = k(\mathrm{mod}\, I) + 1$, let*

$$x_j^{k+1} = x_j^k + \omega \overline{A_{ij}}(b_i - (Ax^k)_i)). \tag{14.7}$$

The relaxed ART converges to the solution closest to x^0, in the consistent case. In the inconsistent case, it does not converge, but subsequences associated with the same i converge to distinct vectors, forming a limit cycle.

14.2.4 Constrained ART

Let C be a closed, nonempty convex subset of $\mathbb{C}^J$ and $P_C x$ the orthogonal projection of x onto C. If there are solutions of $Ax = b$ that lie within C, we can find them using the constrained ART algorithm:

Algorithm 14.2 (Constrained ART) *With x^0 arbitrary and $i = k(\mathrm{mod}\, I) + 1$, let*

$$z_j^{k+1} = x_j^k + \overline{A_{ij}}(b_i - (Ax^k)_i), \tag{14.8}$$

and $x^{k+1} = P_C z^{k+1}$.

For example, if A and b are real and we seek a nonnegative solution to $Ax = b$, we can use

Algorithm 14.3 (Nonnegative ART) *With $i = k(\mathrm{mod}\, I) + 1$, and x^0 arbitrary, let*

$$x_j^{k+1} = (x_j^k + A_{ij}(b_i - (Ax^k)_i))_+, \tag{14.9}$$

where, for any real number a, $a_+ = \max\{a, 0\}$.

The constrained ART converges to a solution of $Ax = b$ within C, whenever such solutions exist.

Noise in the data vector b can manifest itself in a variety of ways. Suppose that the system $Ax = b$ ought to have nonnegative solutions, but because the entries of b are noisy measured data, it does not. Theorem 14.1 tells us that when $J > I$, but $Ax = b$ has no nonnegative solutions, the nonnegatively constrained least-squares solution can have at most $I-1$ nonzero entries, regardless of how large J is [51]. This phenomenon also occurs

with several other approximate methods, such as those that minimize the cross-entropy distance. This gives us a sense of what can happen when we impose positivity on the calculated least-squares solution, that is, when we minimize $||Ax - b||_2$ over all nonnegative vectors x.

Definition 14.1 *The matrix A has the* full-rank property *if A and every matrix Q obtained from A by deleting columns have full rank.*

Theorem 14.1 *Let A have the full-rank property. Suppose there is no nonnegative solution to the system of equations $Ax = b$. Then there is a subset S of the set $\{j = 1, 2, ..., J\}$, with cardinality at most $I - 1$, such that, if $\hat{x}$ is any minimizer of $||Ax - b||_2$ subject to $x \geq 0$, then $\hat{x}_j = 0$ for j not in S. Therefore, $\hat{x}$ is unique.*

14.2.5 When $Ax = b$ Has Solutions

For the consistent case, in which the system $Ax = b$ has exact solutions, we have the following result.

Theorem 14.2 *Let $A\hat{x} = b$ and let x^0 be arbitrary. Let $\{x^k\}$ be generated by Equation (14.5). Then the sequence $\{||\hat{x} - x^k||_2\}$ is decreasing and $\{x^k\}$ converges to the solution of $Ax = b$ closest to x^0.*

The proof of the next lemma follows from the definition of the ART iteration, with a little algebraic manipulation.

Lemma 14.1 *Let x^0 and y^0 be arbitrary and $\{x^k\}$ and $\{y^k\}$ be the sequences generated by applying the ART algorithm, beginning with x^0 and y^0, respectively; that is, $y^{k+1} = P_{i(k)}y^k$. Then*

$$||x^0 - y^0||_2^2 - ||x^I - y^I||_2^2 = \sum_{i=1}^{I} |(Ax^{i-1})_i - (Ay^{i-1})_i|^2. \tag{14.10}$$

Ex. 14.1 *Prove Lemma 14.1.*

Proof of Theorem 14.2: Let $A\hat{x} = b$. Let $v_i^r = (Ax^{rI+i-1})_i$ and $v^r = (v_1^r, ..., v_I^r)^T$, for $r = 0, 1, ...$ It follows from Equation (14.10) that the sequence $\{||\hat{x} - x^{rI}||_2\}$ is decreasing and the sequence $\{v^r - b\} \to 0$. So $\{x^{rI}\}$ is bounded; let $x^{*,0}$ be a cluster point. Then, for $i = 1, 2, ..., I$, let $x^{*,i}$ be the successor of $x^{*,i-1}$ using the ART algorithm. It follows that $(Ax^{*,i-1})_i = b_i$ for each i, from which we conclude that $x^{*,0} = x^{*,i}$ for all i and that $Ax^{*,0} = b$. Using $x^{*,0}$ in place of the arbitrary solution $\hat{x}$, we have that the sequence $\{||x^{*,0} - x^k||_2\}$ is decreasing. But a subsequence

converges to zero, so $\{x^k\}$ converges to $x^{*,0}$. By Equation (14.10), the difference $||\hat{x} - x^k||_2^2 - ||\hat{x} - x^{k+1}||_2^2$ is independent of which solution $\hat{x}$ we pick; consequently, so is $||\hat{x} - x^0||_2^2 - ||\hat{x} - x^{*,0}||_2^2$. It follows that $x^{*,0}$ is the solution closest to x^0. This completes the proof. ∎

14.2.6 When $Ax = b$ Has No Solutions

When there are no exact solutions, the ART does not converge to a single vector, but, for each fixed i, the subsequence $\{x^{nI+i}, n = 0, 1, ...\}$ converges to a vector z^i and the collection $\{z^i \,|i = 1, ..., I\}$ is called the *limit cycle*. This was shown by Tanabe [249] and also follows from the results of De Pierro and Iusem [113]. Proofs of subsequential convergence are given in [65, 66]. The ART limit cycle will vary with the ordering of the equations, and contains more than one vector unless an exact solution exists.

Open Question: If there is a unique geometric least-squares solution, where is it, in relation to the vectors of the limit cycle? Can it be calculated easily, from the vectors of the limit cycle?

There is a partial answer to the second question. In [55] (see also [65]) it was shown that if the system $Ax = b$ has no exact solution, and if $I = J+1$, then the vectors of the limit cycle lie on a sphere in J-dimensional space having the least-squares solution at its center. This is not true more generally, however.

Open Question: In both the consistent and inconsistent cases, the sequence $\{x^k\}$ of ART iterates is bounded, as Tanabe [249], and De Pierro and Iusem [113] have shown. The proof is easy in the consistent case. Is there an easy proof for the inconsistent case?

14.3 Regularized ART

If the entries of b are noisy but the system $Ax = b$ remains consistent (which can easily happen in the under-determined case, with $J > I$), the ART begun at $x^0 = 0$ converges to the solution having minimum Euclidean norm, but this norm can be quite large. The resulting solution is probably useless.

We know from a previous exercise that the system $AA^\dagger z = b$ has a solution if and only if the system $Ax = b$ has solutions.

Ex. 14.2 *Show that the matrix $AA^\dagger + \epsilon I$ is always invertible, for any $\epsilon > 0$.*

Then show that

$$(AA^\dagger + \epsilon I)^{-1}A = A(A^\dagger A + \epsilon I)^{-1}.$$

Instead of solving $Ax = b$, we *regularize* by minimizing, for example, the function

$$F_\epsilon(x) = ||Ax - b||_2^2 + \epsilon^2||x||_2^2. \tag{14.11}$$

The solution to this problem is the vector

$$\hat{x}_\epsilon = (A^\dagger A + \epsilon^2 I)^{-1}A^\dagger b, \tag{14.12}$$

which always exists, even when the system $Ax = b$ has no solutions.

However, we do not want to calculate $A^\dagger A + \epsilon^2 I$ when the matrix A is large. Fortunately, there are ways to find $\hat{x}_\epsilon$, using only the matrix A and the ART algorithm.

We discuss two methods for using ART to obtain regularized solutions of $Ax = b$. The first one is presented in [65], while the second one is due to Eggermont, Herman, and Lent [126].

Both methods rely on the fact that when the ART is applied to a consistent system $Ax = b$ it converges to the solution of that system closest to where we began the iteration. We know from Theorem 3.1 that the solution of $Ax = b$ closest to the origin has the form $x = A^\dagger z$, so that $b = AA^\dagger z$. Assuming $AA^\dagger$ is invertible, we have $z = (AA^\dagger)^{-1}b$ and and

$$x = A^\dagger(AA^\dagger)^{-1}b.$$

If we want to find the solution closest to a given vector p, we write $t = x - p$, so that $At = Ax - Ap = b - Ap$ and then find the solution of $At = b - Ap$ closest to the origin. Then

$$t = A^\dagger(AA^\dagger)^{-1}(b - Ap),$$

and

$$x = t + p = A^\dagger(AA^\dagger)^{-1}(b - Ap) + p.$$

In our first method we use ART to solve the system of equations given in matrix form by

$$\begin{bmatrix} A^\dagger & \epsilon I \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = 0. \tag{14.13}$$

We begin with $u^0 = b$ and $v^0 = 0$. Then, the lower component of the limit vector is $v^\infty = -\epsilon\hat{x}_\epsilon$.

The method of Eggermont *et al.* is similar. In their method we use ART to solve the system of equations given in matrix form by

$$\begin{bmatrix} A & \epsilon I \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} = b. \tag{14.14}$$

We begin at $x^0 = 0$ and $v^0 = 0$. Then, the limit vector has for its upper component $x^\infty = \hat{x}_\epsilon$ as before, and that $\epsilon v^\infty = b - A\hat{x}_\epsilon$.

Ex. 14.3 *Prove that the two iterative methods for regularized ART perform as indicated.*

14.4 Avoiding the Limit Cycle

Generally, the greater the minimum value of $||Ax - b||_2^2$ the more the vectors of the LC are distinct from one another. There are several ways to avoid the LC in ART and to obtain a least-squares solution. One way is the *double ART* (DART) [59]:

14.4.1 Double ART (DART)

We know that any b can be written as $b = A\hat{x} + \hat{w}$, where $A^T\hat{w} = 0$ and $\hat{x}$ is a minimizer of $||Ax - b||_2^2$. The vector $\hat{w}$ is the orthogonal projection of b onto the null space of the matrix transformation $A^\dagger$. Therefore, in Step 1 of DART we apply the ART algorithm to the consistent system of linear equations $A^\dagger w = 0$, beginning with $w^0 = b$. The limit is $w^\infty = \hat{w}$, the member of the null space of $A^\dagger$ closest to b. In Step 2, apply ART to the consistent system of linear equations $Ax = b - w^\infty = A\hat{x}$. The limit is then the minimizer of $||Ax - b||_2$ closest to x^0. Notice that we could also obtain the least-squares solution by applying ART to the system $A^\dagger y = A^\dagger b$, starting with $y^0 = 0$, to obtain the minimum-norm solution, which is $y = A\hat{x}$, and then applying ART to the system $Ax = y$.

14.4.2 Strongly Under-relaxed ART

Another method for avoiding the LC is *strong under-relaxation*, due to Censor, Eggermont and Gordon [82]. Let $t > 0$. Replace the iterative step in ART with

$$x_j^{k+1} = x_j^k + t\overline{A_{ij}}(b_i - (Ax^k)_i). \tag{14.15}$$

In [82] it is shown that, as $t \to 0$, the vectors of the LC approach the geometric least squares solution closest to x^0; a short proof is in [55]. Bertsekas [20] uses strong under-relaxation to obtain convergence of more general incremental methods.

14.5 The MART

The *multiplicative* ART (MART) [153] is an iterative algorithm closely related to the ART. It also was devised to obtain tomographic images, but, like ART, applies more generally; MART applies to systems of linear equations $Ax = b$ for which the b_i are positive, the A_{ij} are nonnegative, and the solution x we seek is to have nonnegative entries. It is not so easy to see the relation between ART and MART if we look at the most general formulation of MART. For that reason, we began with a simpler case, transmission tomographic imaging, in which the relation is most clearly visible.

14.5.1 The MART in the General Case

The MART, which can be applied only to nonnegative systems, is a sequential, or row-action, method that uses one equation only at each step of the iteration.

Algorithm 14.4 (MART) *Let x^0 be any positive vector, and $i = k(\mathrm{mod}\, I) + 1$. Having found x^k for positive integer k, define x^{k+1} by*

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i} \Big)^{m_i^{-1} A_{ij}}, \tag{14.16}$$

where $m_i = \max \{A_{ij} \,|j = 1, 2, ..., J\}$.

Some treatments of MART leave out the m_i, but require only that the entries of A have been rescaled so that $A_{ij} \leq 1$ for all i and j. The m_i is important, however, in accelerating the convergence of MART. There is another way to do the rescaling for MART, which we discuss in the appendix on Geometric Programming and the MART.

The MART can be accelerated by relaxation, as well.

Algorithm 14.5 (Relaxed MART) *Let x^0 be any positive vector, and $i = k(\mathrm{mod}\, I) + 1$. Having found x^k for positive integer k, define x^{k+1} by*

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i} \Big)^{\gamma_i m_i^{-1} A_{ij}}, \tag{14.17}$$

where γ_i is in the interval $(0, 1)$.

As with ART, finding the best relaxation parameters is a bit of an art.

14.5.2 Cross-Entropy

For $a > 0$ and $b > 0$, let the cross-entropy or Kullback-Leibler distance from a to b be

$$KL(a, b) = a \log \frac{a}{b} + b - a, \tag{14.18}$$

with $KL(a, 0) = +\infty$, and $KL(0, b) = b$. Extend to nonnegative vectors coordinate-wise, so that

$$KL(x, z) = \sum_{j=1}^{J} KL(x_j, z_j). \tag{14.19}$$

Unlike the Euclidean distance, the KL distance is not symmetric; $KL(Ax, b)$ and $KL(b, Ax)$ are distinct, and we can obtain different approximate solutions of $Ax = b$ by minimizing these two distances with respect to nonnegative x.

14.5.3 Convergence of MART

In the consistent case, by which we mean that $Ax = b$ has nonnegative solutions, we have the following convergence theorem for MART.

Theorem 14.3 *In the consistent case, the MART converges to the unique nonnegative solution of $b = Ax$ for which the distance $\sum_{j=1}^{J} KL(x_j, x_j^0)$ is minimized.*

If the starting vector x^0 is the vector whose entries are all one, then the MART converges to the solution that maximizes the Shannon entropy,

$$SE(x) = \sum_{j=1}^{J} x_j \log x_j - x_j. \tag{14.20}$$

As with ART, the speed of convergence is greatly affected by the ordering of the equations, converging most slowly when consecutive equations correspond to nearly parallel hyperplanes.

Open Question: When there are no nonnegative solutions, MART does not converge to a single vector, but, like ART, is always observed to produce a limit cycle of vectors. Unlike ART, there is no proof of the existence of a limit cycle for MART.

Chapter 15

Appendix: Eigenvalue Bounds

15.1 Chapter Summary

The ART is a sequential algorithm, using only a single equation from the system $Ax = b$ at each step of the iteration. In this chapter we consider iterative procedures for solving $Ax = b$ in which several or all of the equations are used at each step. Such methods are called *block-iterative* and *simultaneous* algorithms, respectively. We survey a number of these

block-iterative methods. We obtain upper bounds on the spectral radius of positive-definite matrices and use these bounds in the selection of parameters in the iterative methods.

15.2 Introduction and Notation

We are concerned here with iterative methods for solving, at least approximately, the system of I linear equations in J unknowns symbolized by $Ax = b$. In the applications of interest to us, such as medical imaging, both I and J are quite large, making the use of iterative methods the only feasible approach. It is also typical of such applications that the matrix A is sparse, that is, has relatively few non-zero entries. Therefore, iterative methods that exploit this sparseness to accelerate convergence are of special interest to us.

The *algebraic reconstruction technique* (ART) of Gordon, et al. [153] is a *sequential* method; at each step only one equation is used. The current vector x^{k-1} is projected orthogonally onto the hyperplane corresponding to that single equation, to obtain the next iterate x^k. The iterative step of the ART is

$$x_j^k = x_j^{k-1} + \overline{A_{ij}}\left(\frac{b_i - (Ax^{k-1})_i}{\sum_{t=1}^J |A_{it}|^2}\right), \tag{15.1}$$

where $i = k(\mathrm{mod}\, I)$. The sequence $\{x^k\}$ converges to the solution closest to x^0 in the consistent case, but only converges subsequentially to a limit cycle in the inconsistent case.

Cimmino's method [96] is a *simultaneous* method, in which all the equations are used at each step. The current vector x^{k-1} is projected orthogonally onto each of the hyperplanes and these projections are averaged to obtain the next iterate x^k. The iterative step of Cimmino's method is

$$x_j^k = \frac{1}{I}\sum_{i=1}^I \left(x_j^{k-1} + \overline{A_{ij}}\left(\frac{b_i - (Ax^{k-1})_i}{\sum_{t=1}^J |A_{it}|^2}\right)\right),$$

which can also be written as

$$x_j^k = x_j^{k-1} + \sum_{i=1}^I \overline{A_{ij}}\left(\frac{b_i - (Ax^{k-1})_i}{I\sum_{t=1}^J |A_{it}|^2}\right). \tag{15.2}$$

Landweber's iterative scheme [189] with

$$x^k = x^{k-1} + B^\dagger(d - Bx^{k-1}), \tag{15.3}$$

converges to the least-squares solution of $Bx = d$ closest to x^0, provided that the largest singular value of B does not exceed one. If we let B be the matrix with entries

$$B_{ij} = A_{ij} / \sqrt{I \sum_{t=1}^{J} |A_{it}|^2},$$

and define

$$d_i = b_i / \sqrt{I \sum_{t=1}^{J} |A_{it}|^2},$$

then, since the trace of the matrix $BB^\dagger$ is one, convergence of Cimmino's method follows. However, using the trace in this way to estimate the largest singular value of a matrix usually results in an estimate that is far too large, particularly when A is large and sparse, and therefore in an iterative algorithm with unnecessarily small step sizes.

The appearance of the term

$$I \sum_{t=1}^{J} |A_{it}|^2$$

in the denominator of Cimmino's method suggested to Censor et al. [87] that, when A is sparse, this denominator might be replaced with

$$\sum_{t=1}^{J} s_t |A_{it}|^2,$$

where s_t denotes the number of non-zero entries in the tth column of A. The resulting iterative method is the *component-averaging* (CAV) iteration. Convergence of the CAV method was established by showing that no singular value of the matrix B exceeds one, where B has the entries

$$B_{ij} = A_{ij} / \sqrt{\sum_{t=1}^{J} s_t |A_{it}|^2}.$$

In [69] we extended this result, to show that no eigenvalue of $A^\dagger A$ exceeds the maximum of the numbers

$$p_i = \sum_{t=1}^{J} s_t |A_{it}|^2.$$

Convergence of CAV then follows, as does convergence of several other methods, including the ART, Landweber's method, the SART [5], the block-iterative CAV (BICAV) [88], the CARP1 method of Gordon and

Gordon [154], a block-iterative variant of CARP1 obtained from the DROP method of Censor et al. [85], and the SIRT method [259].

For a positive integer N with $1 \leq N \leq I$, we let $B_1, ..., B_N$ be not necessarily disjoint subsets of the set $\{i = 1, ..., I\}$; the subsets B_n are called *blocks*. We then let A_n be the matrix and b^n the vector obtained from A and b, respectively, by removing all the rows except for those whose index i is in the set B_n. For each n, we let s_{nt} be the number of non-zero entries in the tth column of the matrix A_n, s_n the maximum of the s_{nt}, s the maximum of the s_t, and $L_n = \rho(A_n^\dagger A_n)$ be the spectral radius, or largest eigenvalue, of the matrix $A_n^\dagger A_n$, with $L = \rho(A^\dagger A)$. We denote by A_i the ith row of the matrix A, and by ν_i the length of A_i, so that

$$\nu_i^2 = \sum_{j=1}^{J} |A_{ij}|^2.$$

15.3 Block-Iterative Algorithms

An iterative algorithm for solving the system of equations $Ax = b$ is called a *block-iterative* algorithm if only a single data vector b^n is used at each step of the iteration. The ART is an extreme case of block-iteration, in that only a single entry of b is used at each step. A method is called *simultaneous* if the entire vector b is used at each step. We consider simultaneous methods in the next few sections, and then discuss block-iterative methods in more detail.

15.4 Cimmino's Algorithm

The ART seeks a solution of $Ax = b$ by projecting the current vector x^{k-1} orthogonally onto the next hyperplane $H(a^{i(k)}, b_{i(k)})$ to get x^k; here $i(k) = k(\text{mod}\,)I$. In Cimmino's algorithm, we project the current vector x^{k-1} onto each of the hyperplanes and then average the result to get x^k. The algorithm begins at $k = 1$, with an arbitrary x^0; the iterative step is then

$$x^k = \frac{1}{I} \sum_{i=1}^{I} P_i x^{k-1}, \tag{15.4}$$

where P_i is the orthogonal projection onto $H(a^i, b_i)$. The iterative step can then be written as

$$x_j^k = x_j^{k-1} + \frac{1}{I}\sum_{i=1}^{I}\left(\frac{\overline{A_{ij}}(b_i - (Ax^{k-1})_i)}{\nu_i^2}\right). \tag{15.5}$$

As we saw in our discussion of the ART, when the system $Ax = b$ has no solutions, the ART does not converge to a single vector, but to a limit cycle. One advantage of many simultaneous algorithms, such as Cimmino's, is that they do converge to the least squares solution in the inconsistent case.

When $\nu_i = 1$ for all i, Cimmino's algorithm has the form $x^{k+1} = Tx^k$, for the operator T given by

$$Tx = (I - \frac{1}{I}A^\dagger A)x + \frac{1}{I}A^\dagger b.$$

Experience with Cimmino's algorithm shows that it is slow to converge. In the next section we consider how we might accelerate the algorithm.

15.5 The Landweber Algorithms

For simplicity, we assume, in this section, that $\nu_i = 1$ for all i. The Landweber algorithm [189, 18], with the iterative step

$$x^k = x^{k-1} + \gamma A^\dagger(b - Ax^{k-1}), \tag{15.6}$$

converges to the least squares solution closest to the starting vector x^0, provided that $0 < \gamma < 2/\lambda_{max}$, where λ_{max} is the largest eigenvalue of the nonnegative-definite matrix $A^\dagger A$. Loosely speaking, the larger γ is, the faster the convergence. However, precisely because A is large, calculating the matrix $A^\dagger A$, not to mention finding its largest eigenvalue, can be prohibitively expensive. The matrix A is said to be sparse if most of its entries are zero. Useful upper bounds for λ_{max} are then given by Theorems 15.1 and 15.6.

15.5.1 Finding the Optimum γ

The operator

$$Tx = x + \gamma A^\dagger(b - Ax) = (I - \gamma A^\dagger A)x + \gamma A^\dagger b$$

is an affine linear operator. To guarantee convergence we need $0 \leq \gamma < 2/\lambda_{max}$. Should we always try to take γ near its upper bound, or is there

an optimum value of γ? To answer this question we consider the eigenvalues of B for various values of γ.

Lemma 15.1 *If $\gamma < 0$, then none of the eigenvalues of B is less than one.*

Lemma 15.2 *For*

$$0 \leq \gamma \leq \frac{2}{\lambda_{max} + \lambda_{min}}, \tag{15.7}$$

we have

$$\rho(B) = 1 - \gamma\lambda_{min}; \tag{15.8}$$

the smallest value of $\rho(B)$ occurs when

$$\gamma = \frac{2}{\lambda_{max} + \lambda_{min}}, \tag{15.9}$$

and equals

$$\frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}}. \tag{15.10}$$

Similarly, for

$$\gamma \geq \frac{2}{\lambda_{max} + \lambda_{min}}, \tag{15.11}$$

we have

$$\rho(B) = \gamma\lambda_{max} - 1; \tag{15.12}$$

the smallest value of $\rho(B)$ occurs when

$$\gamma = \frac{2}{\lambda_{max} + \lambda_{min}}, \tag{15.13}$$

and equals

$$\frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}}. \tag{15.14}$$

We see from this lemma that, if $0 \leq \gamma < 2/\lambda_{max}$, and $\lambda_{min} > 0$, then $\|B\|_2 = \rho(B) < 1$, so that B is a strict contraction. We minimize $\|B\|_2$ by taking

$$\gamma = \frac{2}{\lambda_{max} + \lambda_{min}}, \tag{15.15}$$

in which case we have

$$\|B\|_2 = \frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}} = \frac{c-1}{c+1}, \tag{15.16}$$

for $c = \lambda_{max}/\lambda_{min}$, the *condition number* of the positive-definite matrix $A^\dagger A$. The closer c is to one, the smaller the norm $\|B\|_2$, and the faster the convergence.

On the other hand, if $\lambda_{min} = 0$, then $\rho(B) = 1$ for all γ in the interval $(0, 2/\lambda_{max})$. For example, consider the orthogonal projection P_0 onto the hyperplane $H_0 = H(a, 0)$, where $\|a\|_2 = 1$. This operator can be written

$$P_0 = I - aa^\dagger. \tag{15.17}$$

The largest eigenvalue of $aa^\dagger$ is $\lambda_{max} = 1$; the remaining ones are zero. The relaxed projection operator

$$B = I - \gamma aa^\dagger \tag{15.18}$$

has $\rho(B) = 1 - \gamma > 1$, if $\gamma < 0$, and for $\gamma \geq 0$, we have $\rho(B) = 1$.

15.5.2 The Projected Landweber Algorithm

When we require a nonnegative approximate solution x for the real system $Ax = b$ we can use a modified version of the Landweber algorithm, called the projected Landweber algorithm [18], in this case having the iterative step

$$x^{k+1} = (x^k + \gamma A^\dagger(b - Ax^k))_+, \tag{15.19}$$

where, for any real vector a, we denote by $(a)_+$ the nonnegative vector whose entries are those of a, for those that are nonnegative, and are zero otherwise. The projected Landweber algorithm converges to a vector that minimizes $\|Ax - b\|_2$ over all nonnegative vectors x, for the same values of γ.

The projected Landweber algorithm is actually more general. For any closed, nonempty convex set C in $\mathbb{R}^J$, define the iterative sequence

$$x^{k+1} = P_C(x^k + \gamma A^\dagger(b - Ax^k)). \tag{15.20}$$

This sequence converges to a minimizer of the function $\|Ax - b\|_2$ over all x in C, whenever such minimizers exist. As we saw previously, both the Landweber and projected Landweber algorithms are special cases of the CQ algorithm [62]

15.6 Some Upper Bounds for L

For the iterative algorithms we shall consider here, having a good upper bound for the largest eigenvalue of the matrix $A^\dagger A$ is important. In the applications of interest, principally medical image processing, the matrix A is large; even calculating $A^\dagger A$, not to mention computing eigenvalues, is prohibitively expensive. In addition, the matrix A is typically sparse, but $A^\dagger A$ will not be, in general. In this section we present upper bounds for L that are particularly useful when A is sparse and do not require the calculation of $A^\dagger A$.

15.6.1 Earlier Work

Many of the concepts we study in computational linear algebra were added to the mathematical toolbox relatively recently, as this area blossomed with the growth of electronic computers. Based on my brief investigations into the history of matrix theory, I believe that the concept of a norm of a matrix was not widely used prior to about 1945. This was recently confirmed when I read the paper [155]; as pointed out there, the use of matrix norms became an important part of numerical linear algebra only after the publication of [262]. Prior to the late 1940's a number of papers were published that established upper bounds on $\rho(A)$, for general square matrix A. As we now can see, several of these results are immediate consequences of the fact that $\rho(A) \leq \|A\|$, for any induced matrix norm. We give two examples.

For a given N by N matrix A, let

$$C_n = \sum_{m=1}^{N} |A_{mn}|,$$

$$R_m = \sum_{n=1}^{N} |A_{mn}|,$$

and C and R the maxima of C_n and R_m, respectively. We now know that $C = \|A\|_1$, and $R = \|A\|_\infty$, but the earlier authors did not.

In 1930 Browne [32] proved the following theorem.

Theorem 15.1 (Browne) *Let λ be any eigenvalue of A. Then*

$$|\lambda| \leq \frac{1}{2}(C + R).$$

In 1944 Farnell [132] published the following theorems.

Theorem 15.2 (Farnell I) *For any eigenvalue λ of A we have*

$$|\lambda| \leq \sqrt{CR}.$$

Theorem 15.3 (Farnell II) *Let*

$$r_m = \sum_{n=1}^{N} |A_{mn}|^2,$$

and

$$c_m = \sum_{n=1}^{N} |A_{nm}|^2.$$

Then, for any eigenvalue λ of A, we have

$$|\lambda| \leq \sqrt{\sum_{m=1}^{N} \sqrt{r_m c_m}}.$$

In 1946 Brauer [28] proved the following theorem.

Theorem 15.4 (Brauer) *For any eigenvalue λ of A, we have*

$$|\lambda| \leq \min\{C, R\}.$$

Ex. 15.1 *Prove Theorems 15.1, 15.2, and 15.4 using properties of matrix norms. Can you also prove Theorem 15.3 this way?*

Let A be an arbitrary rectangular complex matrix. Since the largest singular value of A is the square root of the maximum eigenvalue of the square matrix $S = A^\dagger A$, we could use the inequality

$$\rho(A^\dagger A) = \|A^\dagger A\|_2 \leq \|A^\dagger A\|,$$

for any induced matrix norm, to establish an upper bound for the singular values of A. However, that bound would be in terms of the entries of $A^\dagger A$, not of A itself. In what follows we obtain upper bounds on the singular values of A in terms of the entries of A itself.

Ex. 15.2 *Let A be an arbitrary rectangular matrix. Prove that no singular value of A exceeds $\sqrt{\|A\|_1 \|A\|_\infty}$.*

We see from this exercise that Farnell (I) does generalize to arbitrary rectangular matrices and singular values. Brauer's Theorem 15.4 may suggest that no singular value of a rectangular matrix A exceeds the minimum

of $\|A\|_1$ and $\|A\|_\infty$, but this is not true. Consider the matrix A whose SVD is given by

$$A = \begin{bmatrix} 4 & 3 \\ 8 & 6 \\ 8 & 6 \end{bmatrix} = \begin{bmatrix} 1/3 & 2/3 & 2/3 \\ 2/3 & -2/3 & 1/3 \\ 2/3 & 1/3 & -2/3 \end{bmatrix} \begin{bmatrix} 15 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 4/5 & 3/5 \\ 3/5 & -4/5 \end{bmatrix}.$$

The largest singular value of A is 15, $\|A\|_1 = 20$, $\|A\|_\infty = 14$, and we do have

$$15 \leq \sqrt{(20)(14)},$$

but we do not have

$$15 \leq \min\{20, 14\} = 14.$$

15.6.2 Our Basic Eigenvalue Inequality

In [259] van der Sluis and van der Vorst show that certain rescaling of the matrix A results in none of the eigenvalues of $A^\dagger A$ exceeding one. A modification of their proof leads to upper bounds on the eigenvalues of the original $A^\dagger A$ ([69]). For any a in the interval $[0, 2]$ let

$$c_{aj} = c_{aj}(A) = \sum_{i=1}^{I} |A_{ij}|^a,$$

$$r_{ai} = r_{ai}(A) = \sum_{j=1}^{J} |A_{ij}|^{2-a},$$

and c_a and r_a the maxima of the c_{aj} and r_{ai}, respectively. We prove the following theorem.

Theorem 15.5 *For any a in the interval $[0, 2]$, no eigenvalue of the matrix $A^\dagger A$ exceeds the maximum of*

$$\sum_{j=1}^{J} c_{aj} |A_{ij}|^{2-a},$$

over all i, nor the maximum of

$$\sum_{i=1}^{I} r_{ai} |A_{ij}|^a,$$

over all j. Therefore, no eigenvalue of $A^\dagger A$ exceeds $c_a r_a$.

Proof: Let $A^\dagger Av = \lambda v$, and let $w = Av$. Then we have

$$\|A^\dagger w\|_2^2 = \lambda \|w\|_2^2.$$

Applying Cauchy's Inequality, we obtain

$$\Big|\sum_{i=1}^{I} \overline{A_{ij}} w_i\Big|^2 \leq \Big(\sum_{i=1}^{I} |A_{ij}|^{a/2} |A_{ij}|^{1-a/2} |w_i|\Big)^2$$

$$\leq \Big(\sum_{i=1}^{I} |A_{ij}|^a\Big)\Big(\sum_{i=1}^{I} |A_{ij}|^{2-a} |w_i|^2\Big).$$

Therefore,

$$\|A^\dagger w\|_2^2 \leq \sum_{j=1}^{J} \Big(c_{aj}(\sum_{i=1}^{I} |A_{ij}|^{2-a} |w_i|^2)\Big) = \sum_{i=1}^{I} \Big(\sum_{j=1}^{J} c_{aj} |A_{ij}|^{2-a}\Big) |w_i|^2$$

$$\leq \max_i \Big(\sum_{j=1}^{J} c_{aj} |A_{ij}|^{2-a}\Big) \|w\|^2.$$

The remaining two assertions follow in similar fashion. ∎

As a corollary, we obtain the following eigenvalue inequality, which is central to our discussion.

Corollary 15.1 *For each $i = 1, 2, ..., I$, let*

$$p_i = \sum_{j=1}^{J} s_j |A_{ij}|^2,$$

and let p be the maximum of the p_i. Then $L \leq p$.

Proof: Take $a = 0$. Then, using the convention that $0^0 = 0$, we have $c_{0j} = s_j$. ∎

Corollary 15.2 *([62]; [258], Th. 4.2) If $\sum_{j=1}^{J} |A_{ij}|^2 \leq 1$ for each i, then $L \leq s$.*

Proof: For all i we have

$$p_i = \sum_{j=1}^{J} s_j |A_{ij}|^2 \leq s \sum_{j=1}^{J} |A_{ij}|^2 \leq s.$$

Therefore,

$$L \leq p \leq s.$$

∎

The next corollary gives Inequality (6.39) that we saw earlier.

Corollary 15.3 *Selecting $a = 1$, we have*

$$L = \|A\|_2^2 \leq \|A\|_1 \|A\|_\infty = c_1 r_1.$$

Therefore, the largest singular value of A does not exceed $\sqrt{\|A\|_1 \|A\|_\infty}$.

Corollary 15.4 *Selecting $a = 2$, we have*

$$L = \|A\|_2^2 \leq \|A\|_F^2,$$

where $\|A\|_F$ denotes the Frobenius norm of A.

Corollary 15.5 *Let G be the matrix with entries*

$$G_{ij} = A_{ij} \sqrt{\alpha_i} \sqrt{\beta_j},$$

where

$$\alpha_i \leq \Big(\sum_{j=1}^{J} s_j \beta_j |A_{ij}|^2 \Big)^{-1},$$

for all i. Then $\rho(G^\dagger G) \leq 1$.

Proof: We have

$$\sum_{j=1}^{J} s_j |G_{ij}|^2 = \alpha_i \sum_{j=1}^{J} s_j \beta_j |A_{ij}|^2 \leq 1,$$

for all i. The result follows from Corollary 15.1. ∎

Corollary 15.6 *If $\sum_{j=1}^{J} s_j |A_{ij}|^2 \leq 1$ for all i, then $L \leq 1$.*

Corollary 15.7 *If $0 < \gamma_i \leq p_i^{-1}$ for all i, then the matrix B with entries $B_{ij} = \sqrt{\gamma_i} A_{ij}$ has $\rho(B^\dagger B) \leq 1$.*

Proof: We have

$$\sum_{j=1}^{J} s_j |B_{ij}|^2 = \gamma_i \sum_{j=1}^{J} s_j |A_{ij}|^2 = \gamma_i p_i \leq 1.$$

Therefore, $\rho(B^\dagger B) \leq 1$, according to the theorem. ∎

Corollary 15.8 *If, for some a in the interval $[0, 2]$, we have*

$$\alpha_i \leq r_{ai}^{-1}, \tag{15.21}$$

for each i, and

$$\beta_j \leq c_{aj}^{-1}, \tag{15.22}$$

for each j, then, for the matrix G with entries

$$G_{ij} = A_{ij}\sqrt{\alpha_i}\sqrt{\beta_j},$$

no eigenvalue of $G^\dagger G$ exceeds one.

Proof: We calculate $c_{aj}(G)$ and $r_{ai}(G)$ and find that

$$c_{aj}(G) \leq \Big(\max_i \alpha_i^{a/2}\Big)\beta_j^{a/2}\sum_{i=1}^{I}|A_{ij}|^a = \Big(\max_i \alpha_i^{a/2}\Big)\beta_j^{a/2}c_{aj}(A),$$

and

$$r_{ai}(G) \leq \Big(\max_j \beta_j^{1-a/2}\Big)\alpha_i^{1-a/2}r_{ai}(A).$$

Therefore, applying the inequalities (15.21) and (15.22), we have

$$c_{aj}(G)r_{ai}(G) \leq 1,$$

for all i and j. Consequently, $\rho(G^\dagger G) \leq 1$. ∎

15.6.3 Another Upper Bound for L

The next theorem ([62]) provides another upper bound for L that is useful when A is sparse. As previously, for each i and j, we let $e_{ij} = 1$, if A_{ij} is not zero, and $e_{ij} = 0$, if $A_{ij} = 0$. Let $0 < \nu_i = \sqrt{\sum_{j=1}^{J}|A_{ij}|^2}$, $\sigma_j = \sum_{i=1}^{I} e_{ij}\nu_i^2$, and σ be the maximum of the σ_j.

Theorem 15.6 *([62]) No eigenvalue of $A^\dagger A$ exceeds σ.*

Proof: Let $A^\dagger Av = cv$, for some non-zero vector v and scalar c. With $w = Av$, we have

$$w^\dagger AA^\dagger w = cw^\dagger w.$$

Then

$$\Big|\sum_{i=1}^{I}\overline{A_{ij}}w_i\Big|^2 = \Big|\sum_{i=1}^{I}\overline{A_{ij}}e_{ij}\nu_i\frac{w_i}{\nu_i}\Big|^2 \leq \Big(\sum_{i=1}^{I}|A_{ij}|^2\frac{|w_i|^2}{\nu_i^2}\Big)\Big(\sum_{i=1}^{I}\nu_i^2 e_{ij}\Big)$$

$$= \Big(\sum_{i=1}^{I}|A_{ij}|^2\frac{|w_i|^2}{\nu_i^2}\Big)\sigma_j \leq \sigma\Big(\sum_{i=1}^{I}|A_{ij}|^2\frac{|w_i|^2}{\nu_i^2}\Big).$$

Therefore, we have

$$cw^\dagger w = w^\dagger AA^\dagger w = \sum_{j=1}^{J} \Big| \sum_{i=1}^{I} \overline{A_{ij}} w_i \Big|^2$$

$$\leq \sigma \sum_{j=1}^{J} \Big(\sum_{i=1}^{I} |A_{ij}|^2 \frac{|w_i|^2}{\nu_i^2} \Big) = \sigma \sum_{i=1}^{I} |w_i|^2 = \sigma w^\dagger w.$$

We conclude that $c \leq \sigma$. ∎

Corollary 15.9 *Let the rows of A have Euclidean length one. Then no eigenvalue of $A^\dagger A$ exceeds the maximum number of non-zero entries in any column of A.*

Proof: We have $\nu_i^2 = \sum_{j=1}^{J} |A_{ij}|^2 = 1$, for each i, so that $\sigma_j = s_j$ is the number of non-zero entries in the jth column of A, and $\sigma = s$ is the maximum of the σ_j. ∎

Corollary 15.10 *Let ν be the maximum Euclidean length of any row of A and s the maximum number of non-zero entries in any column of A. Then $L \leq \nu^2 s$.*

When the rows of A have length one, it is easy to see that $L \leq I$, so the choice of $\gamma = \frac{1}{I}$ in the Landweber algorithm, which gives Cimmino's algorithm [96], is acceptable, although perhaps much too small.

The proof of Theorem 15.6 is based on results presented by Arnold Lent in informal discussions with Gabor Herman, Yair Censor, Rob Lewitt and me at MIPG in Philadelphia in the late 1990's.

15.7 Eigenvalues and Norms: A Summary

It is helpful, at this point, to summarize the main facts concerning eigenvalues and norms. Throughout this section A will denote an arbitrary matrix, S an arbitrary square matrix, and H an arbitrary Hermitian matrix. We denote by $\|A\|$ an arbitrary induced matrix norm of A.

Here are some of the things we now know:

- 1. $\rho(S^2) = \rho(S)^2$;
- 2. $\rho(S) \leq \|S\|$, for any matrix norm;
- 3. $\rho(H) = \|H\|_2 \leq \|H\|$, for any matrix norm;

- 4. $\|A\|_2^2 = \rho(A^\dagger A) = \|A^\dagger A\|_2 \leq \|A^\dagger A\|$;
- 5. $\|A^\dagger A\|_1 \leq \|A^\dagger\|_1 \|A\|_1 = \|A\|_\infty \|A\|_1$;
- 6. $\|A\|_2^2 \leq \|A\|_1 \|A\|_\infty$;
- 7. $\rho(S) \leq \min\{\|S\|_1, \|S\|_\infty\}$;
- 8. if $\sum_{j=1}^J |A_{ij}|^2 \leq 1$, for all i, then $\|A\|_2^2 \leq s$, where s is the largest number of non-zero entries in any column of A.

15.8 Convergence of Block-Iterative Algorithms

The following theorem is a basic convergence result concerning block-iterative algorithms.

Theorem 15.7 *Let $L_n \leq 1$, for $n = 1, 2, ..., N$. If the system $Ax = b$ is consistent, then, for any starting vector x^0, and with $n = n(k) = k(\mathrm{mod}\, N)$ and $\lambda_k \in [\epsilon, 2 - \epsilon]$ for all k, the sequence $\{x^k\}$ with iterative step*

$$x^k = x^{k-1} + \lambda_k A_n^\dagger (b^n - A_n x^{k-1}) \tag{15.23}$$

converges to the solution of $Ax = b$ for which $\|x - x^0\|_2$ is minimized.

We begin with the following lemma.

Lemma 15.3 *Let T be any (not necessarily linear) operator on $\mathbb{R}^J$, and $S = I - T$, where I denotes the identity operator. Then, for any x and y, we have*

$$\|x - y\|_2^2 - \|Tx - Ty\|_2^2 = 2\langle Sx - Sy, x - y\rangle - \|Sx - Sy\|^2. \tag{15.24}$$

The proof is a simple calculation and we omit it here.

Proof of Theorem 15.7: Let $Az = b$. Applying Equation (15.24) to the operator

$$Tx = x + \lambda_k A_n^\dagger (b^n - A_n x),$$

we obtain

$$\|z - x^{k-1}\|_2^2 - \|z - x^k\|_2^2 =$$

$$2\lambda_k \|b^n - A_n x^{k-1}\|_2^2 - \lambda_k^2 \|A_n^\dagger b^n - A_n^\dagger A_n x^{k-1}\|_2^2. \tag{15.25}$$

Since $L_n \leq 1$, it follows that

$$\|A_n^\dagger b^n - A_n^\dagger A_n x^{k-1}\|_2^2 \leq \|b^n - A_n x^{k-1}\|_2^2.$$

Therefore,

$$\|z - x^{k-1}\|_2^2 - \|z - x^k\|_2^2 \geq (2\lambda_k - \lambda_k^2)\|b^n - A_n x^{k-1}\|_2^2,$$

from which we draw several conclusions:

- the sequence $\{\|z - x^k\|_2\}$ is decreasing;
- the sequence $\{\|b^n - A_n x^{k-1}\|_2\}$ converges to zero.

In addition, for fixed $n = 1, ..., N$ and $m \to \infty$,

- the sequence $\{\|b^n - A_n x^{mN+n-1}\|_2\}$ converges to zero;
- the sequence $\{x^{mN+n}\}$ is bounded.

Let $x^{*,1}$ be a cluster point of the sequence $\{x^{mN+1}\}$; then there is subsequence $\{x^{m_r N+1}\}$ converging to $x^{*,1}$. The sequence $\{x^{m_r N+2}\}$ is also bounded, and we select a cluster point $x^{*,2}$. Continuing in this fashion, we obtain cluster points $x^{*,n}$, for $n = 1, ..., N$. From the conclusions reached previously, we can show that $x^{*,n} = x^{*,n+1} = x^*$, for $n = 1, 2, ..., N-1$, and $Ax^* = b$. Replacing the generic solution $\hat{x}$ with the solution x^*, we see that the sequence $\{\|x^* - x^k\|_2\}$ is decreasing. But, subsequences of this sequence converge to zero, so the entire sequence converges to zero, and so $x^k \to x^*$.

Now we show that x^* is the solution of $Ax = b$ that minimizes $\|x - x^0\|_2$. Since $x^k - x^{k-1}$ is in the range of $A^\dagger$ for all k, so is $x^* - x^0$, from which it follows that x^* is the solution minimizing $\|x - x^0\|_2$. Another way to get this result is to use Equation (15.25). Since the right side of Equation (15.25) is independent of the choice of solution, so is the left side. Summing both sides over the index k reveals that the difference

$$\|x - x^0\|_2^2 - \|x - x^*\|_2^2$$

is independent of the choice of solution. Consequently, minimizing $\|x - x^0\|_2$ over all solutions x is equivalent to minimizing $\|x - x^*\|_2$ over all solutions x; the solution to the latter problem is clearly $x = x^*$. ∎

15.9 Simultaneous Iterative Algorithms

In this section we apply the previous theorems to obtain convergence of several simultaneous iterative algorithms for linear systems.

15.9.1 The General Simultaneous Iterative Scheme

In this section we are concerned with simultaneous iterative algorithms having the following iterative step:

$$x_j^k = x_j^{k-1} + \lambda_k \sum_{i=1}^{I} \gamma_{ij} \overline{A_{ij}} (b_i - (Ax^{k-1})_i), \tag{15.26}$$

with $\lambda_k \in [\epsilon, 1]$ and the choices of the parameters γ_{ij} that guarantee convergence. Although we cannot prove convergence for this most general iterative scheme, we are able to prove the following theorems for the separable case of $\gamma_{ij} = \alpha_i \beta_j$.

Theorem 15.8 *If, for some a in the interval $[0, 2]$, we have*

$$\alpha_i \leq r_{ai}^{-1}, \tag{15.27}$$

for each i, and

$$\beta_j \leq c_{aj}^{-1}, \tag{15.28}$$

for each j, then the sequence $\{x^k\}$ given by Equation (15.26) converges to the minimizer of the proximity function

$$\sum_{i=1}^{I} \alpha_i |b_i - (Ax)_i|^2$$

for which

$$\sum_{j=1}^{J} \beta_j^{-1} |x_j - x_j^0|^2$$

is minimized.

Proof: For each i and j, let

$$G_{ij} = \sqrt{\alpha_i} \sqrt{\beta_j} A_{ij},$$

$$z_j = x_j / \sqrt{\beta_j},$$

and

$$d_i = \sqrt{\alpha_i} b_i.$$

Then $Ax = b$ if and only if $Gz = d$. From Corollary 15.8 we have that $\rho(G^\dagger G) \leq 1$. Convergence then follows from Theorem 15.7. ∎

Corollary 15.11 *Let $\gamma_{ij} = \alpha_i \beta_j$, for positive α_i and β_j. If*

$$\alpha_i \leq \Big(\sum_{j=1}^{J} s_j \beta_j |A_{ij}|^2 \Big)^{-1}, \tag{15.29}$$

for each i, then the sequence $\{x^k\}$ in (15.26) converges to the minimizer of the proximity function

$$\sum_{i=1}^{I} \alpha_i |b_i - (Ax)_i|^2$$

for which

$$\sum_{j=1}^{J} \beta_j^{-1} |x_j - x_j^0|^2$$

is minimized.

Proof: We know from Corollary 15.5 that $\rho(G^\dagger G) \leq 1$. ∎

We now obtain convergence for several known algorithms as corollaries to the previous theorems.

15.9.2 The SIRT Algorithm

Corollary 15.12 *([259]) For some a in the interval $[0, 2]$ let $\alpha_i = r_{ai}^{-1}$ and $\beta_j = c_{aj}^{-1}$. Then the sequence $\{x^k\}$ in (15.26) converges to the minimizer of the proximity function*

$$\sum_{i=1}^{I} \alpha_i |b_i - (Ax)_i|^2$$

for which

$$\sum_{j=1}^{J} \beta_j^{-1} |x_j - x_j^0|^2$$

is minimized.

For the case of $a = 1$, the iterative step becomes

$$x_j^k = x_j^{k-1} + \sum_{i=1}^{I} \left(\frac{\overline{A_{ij}}(b_i - (Ax^{k-1})_i)}{(\sum_{t=1}^{J} |A_{it}|)(\sum_{m=1}^{I} |A_{mj}|)} \right),$$

which was considered in [161]. The SART algorithm [5] is a special case, in which it is assumed that $A_{ij} \geq 0$, for all i and j.

15.9.3 The CAV Algorithm

Corollary 15.13 *If $\beta_j = 1$ and α_i satisfies*

$$0 < \alpha_i \leq \Big(\sum_{j=1}^{J} s_j |A_{ij}|^2 \Big)^{-1},$$

for each i, then the algorithm with the iterative step

$$x^k = x^{k-1} + \lambda_k \sum_{i=1}^{I} \alpha_i (b_i - (Ax^{k-1})_i) A_i^\dagger \tag{15.30}$$

converges to the minimizer of

$$\sum_{i=1}^{I} \alpha_i |b_i - (Ax^{k-1})_i|^2$$

for which $\|x - x^0\|$ is minimized.

When

$$\alpha_i = \Big(\sum_{j=1}^{J} s_j |A_{ij}|^2 \Big)^{-1},$$

for each i, this is the relaxed *component-averaging* (CAV) method of Censor et al. [87].

15.9.4 The Landweber Algorithm

When $\beta_j = 1$ and $\alpha_i = \alpha$ for all i and j, we have the relaxed Landweber algorithm. The convergence condition in Equation (15.21) becomes

$$\alpha \leq \Big(\sum_{j=1}^{J} s_j |A_{ij}|^2 \Big)^{-1} = p_i^{-1}$$

for all i, so $\alpha \leq p^{-1}$ suffices for convergence. Actually, the sequence $\{x^k\}$ converges to the minimizer of $\|Ax - b\|_2$ for which the distance $\|x - x^0\|_2$ is minimized, for any starting vector x^0, when $0 < \alpha < 1/L$. Easily obtained estimates of L are usually over-estimates, resulting in overly conservative choices of α. For example, if A is first normalized so that $\sum_{j=1}^{J} |A_{ij}|^2 = 1$ for each i, then the trace of $A^\dagger A$ equals I, which tells us that $L \leq I$. But this estimate, which is the one used in Cimmino's method [96], is far too large when A is sparse.

15.9.5 The Simultaneous DROP Algorithm

Corollary 15.14 *Let* $0 < w_i \leq 1$,

$$\alpha_i = w_i \nu_i^{-2} = w_i \Big(\sum_{j=1}^{J} |A_{ij}|^2 \Big)^{-1}$$

and $\beta_j = s_j^{-1}$, *for each* i *and* j. *Then the simultaneous algorithm with the iterative step*

$$x_j^k = x_j^{k-1} + \lambda_k \sum_{i=1}^{I} \left(\frac{w_i \overline{A_{ij}} (b_i - (Ax^{k-1})_i)}{s_j \nu_i^2} \right), \tag{15.31}$$

converges to the minimizer of the function

$$\sum_{i=1}^{I} \left| \frac{w_i (b_i - (Ax)_i)}{\nu_i} \right|^2$$

for which the function

$$\sum_{j=1}^{J} s_j |x_j - x_j^0|^2$$

is minimized.

For $w_i = 1$, this is the CARP1 algorithm of [154] (see also [117, 87, 88]). The simultaneous DROP algorithm of [85] requires only that the weights w_i be positive, but dividing each w_i by their maximum, $\max_i\{w_i\}$, while multiplying each λ_k by the same maximum, gives weights in the interval $(0, 1]$. For convergence of their algorithm, we need to replace the condition $\lambda_k \leq 2 - \epsilon$ with $\lambda_k \leq \frac{2-\epsilon}{\max_i\{w_i\}}$.

The denominator in CAV is

$$\sum_{t=1}^{J} s_t |A_{it}|^2,$$

while that in CARP1 is

$$s_j \sum_{t=1}^{J} |A_{it}|^2.$$

It was reported in [154] that the two methods differed only slightly in the simulated cases studied.

15.10 Block-iterative Algorithms

The methods discussed in the previous section are *simultaneous*, that is, all the equations are employed at each step of the iteration. We turn now to *block-iterative methods*, which employ only some of the equations at each step. When the parameters are appropriately chosen, block-iterative methods can be significantly faster than simultaneous ones.

15.10.1 The Block-Iterative Landweber Algorithm

For a given set of blocks, the block-iterative Landweber algorithm has the following iterative step: with $n = k(\mathrm{mod}\, N)$,

$$x^k = x^{k-1} + \gamma_n A_n^\dagger (b^n - A_n x^{k-1}). \tag{15.32}$$

The sequence $\{x^k\}$ converges to the solution of $Ax = b$ that minimizes $\|x - x^0\|_2$, whenever the system $Ax = b$ has solutions, provided that the parameters γ_n satisfy the inequalities $0 < \gamma_n < 1/L_n$. This follows from Theorem 15.7 by replacing the matrices A_n with $\sqrt{\gamma_n} A_n$ and the vectors b^n with $\sqrt{\gamma_n} b^n$.

If the rows of the matrices A_n are normalized to have length one, then we know that $L_n \leq s_n$. Therefore, we can use parameters γ_n that satisfy

$$0 < \gamma_n \leq \Big(s_n \sum_{j=1}^J |A_{ij}|^2 \Big)^{-1}, \tag{15.33}$$

for each $i \in B_n$.

15.10.2 The BICAV Algorithm

We can extend the block-iterative Landweber algorithm as follows: let $n = k(\mathrm{mod}\, N)$ and

$$x^k = x^{k-1} + \lambda_k \sum_{i \in B_n} \gamma_i (b_i - (Ax^{k-1})_i) A_i^\dagger. \tag{15.34}$$

It follows from Theorem 15.1 that, in the consistent case, the sequence $\{x^k\}$ converges to the solution of $Ax = b$ that minimizes $\|x - x^0\|$, provided that, for each n and each $i \in B_n$, we have

$$\gamma_i \leq \Big(\sum_{j=1}^J s_{nj} |A_{ij}|^2 \Big)^{-1}.$$

The BICAV algorithm [88] uses

$$\gamma_i = \Big(\sum_{j=1}^{J} s_{nj} |A_{ij}|^2 \Big)^{-1}.$$

The iterative step of BICAV is

$$x^k = x^{k-1} + \lambda_k \sum_{i \in B_n} \left(\frac{b_i - (Ax^{k-1})_i}{\sum_{t=1}^{J} s_{nt} |A_{it}|^2} \right) A_i^\dagger. \tag{15.35}$$

15.10.3 A Block-Iterative CARP1

The obvious way to obtain a block-iterative version of CARP1 would be to replace the denominator term

$$s_j \sum_{t=1}^{J} |A_{it}|^2$$

with

$$s_{nj} \sum_{t=1}^{J} |A_{it}|^2.$$

However, this is problematic, since we cannot redefine the vector of unknowns using $z_j = x_j \sqrt{s_{nj}}$, since this varies with n. In [85], this issue is resolved by taking τ_j to be not less than the maximum of the s_{nj}, and using the denominator

$$\tau_j \sum_{t=1}^{J} |A_{it}|^2 = \tau_j \nu_i^2.$$

A similar device is used in [177] to obtain a convergent block-iterative version of SART. The iterative step of DROP is

$$x_j^k = x_j^{k-1} + \lambda_k \sum_{i \in B_n} \left(\overline{A_{ij}} \frac{(b_i - (Ax^{k-1})_i)}{\tau_j \nu_i^2} \right). \tag{15.36}$$

Convergence of the DROP (*diagonally-relaxed orthogonal projection*) iteration follows from their Theorem 11. We obtain convergence as a corollary of our previous results.

The change of variables is $z_j = x_j \sqrt{\tau_j}$, for each j. Using our eigenvalue bounds, it is easy to show that the matrices C_n with entries

$$(C_n)_{ij} = \left(\frac{A_{ij}}{\sqrt{\tau_j} \nu_i} \right),$$

for all $i \in B_n$ and all j, have $\rho(C_n^\dagger C_n) \leq 1$. The resulting iterative scheme, which is equivalent to Equation (15.36), then converges, whenever $Ax = b$ is consistent, to the solution minimizing the proximity function

$$\sum_{i=1}^{I} \left| \frac{b_i - (Ax)_i}{\nu_i} \right|^2$$

for which the function

$$\sum_{j=1}^{J} \tau_j |x_j - x_j^0|^2$$

is minimized.

15.10.4 Using Sparseness

Suppose, for the sake of illustration, that each column of A has s nonzero elements, for some $s < I$, and we let $r = s/I$. Suppose also that the number of members of B_n is $I_n = I/N$ for each n, and that N is not too large. Then s_n is approximately equal to $rI_n = s/N$. On the other hand, unless A_n has only zero entries, we know that $s_n \geq 1$. Therefore, it is no help to select N for which $s/N < 1$. For a given degree of sparseness s we need not select N greater than s. The more sparse the matrix A, the fewer blocks we need to gain the maximum advantage from the rescaling, and the more we can benefit from parallelization in the calculations at each step of the algorithm in Equation (15.23).

15.11 Exercises

Ex. 15.3 *Prove Lemma 15.1.*

Ex. 15.4 (Computer Problem) *Compare the speed of convergence of the ART and Cimmino algorithms.*

Ex. 15.5 (Computer Problem) *By generating sparse matrices of various sizes, test the accuracy of the estimates of the largest singular-value given above.*

Chapter 16

Appendix: Fourier Transforms and the FFT

16.1 Chapter Summary

The *Fourier transform* of a complex-valued function $f(x)$ of the real variable x is defined as

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{ix\omega} dx. \tag{16.1}$$

If we have $F(\omega)$, we can obtain $f(x)$ again via the *Fourier Inversion Formula*,

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{-ix\omega} d\omega. \tag{16.2}$$

In many applications, particularly in remote sensing, what we are able to measure are values of $f(x)$, and what we really want is the function $F(\omega)$. This is the case in medical tomography, magnetic-resonance imaging, sonar, radar, optical and radio astronomy, and many other areas. Because our measurements are finite in number, the problem becomes how to estimate $F(\omega)$ from finitely many values of $f(x)$. The *fast Fourier transform* (FFT) is a fast algorithm for calculating one such estimate, the *discrete Fourier transform*. Discovered in 1965 by Cooley and Tukey, the FFT has revolutionized signal and image processing. The man in Figure 16.1 is John Tukey.

16.2 Non-periodic Convolution

Recall the algebra problem of multiplying one polynomial by another. Suppose

$$A(x) = a_0 + a_1 x + ... + a_M x^M$$

and

$$B(x) = b_0 + b_1 x + ... + b_N x^N.$$

Let $C(x) = A(x)B(x)$. With

$$C(x) = c_0 + c_1 x + ... + c_{M+N} x^{M+N},$$

each of the coefficients c_j, $j = 0, ..., M+N$, can be expressed in terms of the a_m and b_n (an easy exercise!). The vector $c = (c_0, ..., c_{M+N})$ is called the *nonperiodic convolution* of the vectors $a = (a_0, ..., a_M)$ and $b = (b_0, ..., b_N)$. Non-periodic convolution can be viewed as a particular case of periodic convolution, as we shall see.

16.3 The DFT as a Polynomial

Given the complex numbers $f_0, f_1, ..., f_{N-1}$, which may or may not be measured values of $f(x)$, we form the vector $\mathbf{f} = (f_0, f_1, ..., f_{N-1})^T$. The DFT of the vector $\mathbf{f}$ is the function

$$DFT_{\mathbf{f}}(\omega) = \sum_{n=0}^{N-1} f_n e^{in\omega},$$

defined for ω in the interval $[0, 2\pi)$. Because $e^{in\omega} = (e^{i\omega})^n$, we can write the DFT as a polynomial

$$DFT_{\mathbf{f}}(\omega) = \sum_{n=0}^{N-1} f_n (e^{i\omega})^n.$$

If we have a second vector, say $\mathbf{d} = (d_0, d_1, ..., d_{N-1})^T$, then we define $DFT_{\mathbf{d}}(\omega)$ similarly. When we multiply $DFT_{\mathbf{f}}(\omega)$ by $DFT_{\mathbf{d}}(\omega)$, we are multiplying two polynomials together, so the result is a sum of powers of the form

$$c_0 + c_1 e^{i\omega} + c_2 (e^{i\omega})^2 + ... + c_{2N-2} (e^{i\omega})^{2N-2}, \tag{16.3}$$

for

$$c_j = f_0 d_j + f_1 d_{j-1} + ... + f_j d_0.$$

This is *nonperiodic convolution* again. In the next section, we consider what happens when, instead of using arbitrary values of ω, we consider only the N special values $\omega_k = \frac{2\pi}{N}k$, $k = 0, 1, ..., N-1$. Because of the periodicity of the complex exponential function, we have

$$(e^{i\omega_k})^{N+j} = (e^{i\omega_k})^j,$$

for each k. As a result, all the powers higher than $N-1$ that showed up in the previous multiplication in Equation (16.3) now become equal to lower powers, and the product now only has N terms, instead of the $2N-1$ terms we got previously. When we calculate the coefficients of these powers, we find that we get more than we got when we did the nonperiodic convolution. Now what we get is called *periodic convolution*.

16.4 The Vector DFT and Periodic Convolution

As we just discussed, nonperiodic convolution is another way of looking at the multiplication of two polynomials. This relationship between convolution on the one hand and multiplication on the other is a fundamental aspect of convolution. Whenever we have a convolution we should ask what related mathematical objects are being multiplied. We ask this question now with regard to periodic convolution; the answer turns out to be the *vector discrete Fourier transform* (vDFT).

16.4.1 The Vector DFT

Let $\mathbf{f} = (f_0, f_1, ..., f_{N-1})^T$ be a column vector whose entries are N arbitrary complex numbers. For $k = 0, 1, ..., N-1$, we let

$$F_k = \sum_{n=0}^{N-1} f_n e^{2\pi i k n/N} = DFT_{\mathbf{f}}(\omega_k). \tag{16.4}$$

Then we let $\mathbf{F} = (F_0, F_1, ..., F_{N-1})^T$ be the column vector with the N complex entries F_k. The vector $\mathbf{F}$ is called the *vector discrete Fourier transform* of the vector $\mathbf{f}$, and we denote it by $\mathbf{F} = vDFT_{\mathbf{f}}$.

As we can see from Equation (16.4), there are N multiplications involved in the calculation of each F_k, and there are N values of k, so it would seem that, in order to calculate the vector DFT of $\mathbf{f}$, we need N^2 multiplications.

In many applications, N is quite large and calculating the vector $\mathbf{F}$ using the definition would be unrealistically time-consuming. The *fast Fourier transform* algorithm (FFT), to be discussed later, gives a quick way to calculate the vector $\mathbf{F}$ from the vector $\mathbf{f}$. The FFT, usually credited to Cooley and Tukey, was discovered in the mid-1960's and revolutionized signal and image processing.

16.4.2 Periodic Convolution

Given the N by 1 vectors $\mathbf{f}$ and $\mathbf{d}$ with complex entries f_n and d_n, respectively, we define a third N by 1 vector $\mathbf{f} * \mathbf{d}$, the *periodic convolution* of $\mathbf{f}$ and $\mathbf{d}$, to have the entries

$$(\mathbf{f} * \mathbf{d})_n = f_0 d_n + f_1 d_{n-1} + ... + f_n d_0 + f_{n+1} d_{N-1} + ... + f_{N-1} d_{n+1}, \quad (16.5)$$

for $n = 0, 1, ..., N-1$.

Notice that the term on the right side of Equation (16.5) is the sum of all products of entries, one from $\mathbf{f}$ and one from $\mathbf{d}$, where the sum of their respective indices is either n or $n+N$.

In the exercises that follow we investigate properties of the vector DFT and relate it to periodic convolution. It is not an exaggeration to say that these two exercises are the most important ones in signal processing.

Ex. 16.1 *Let* $\mathbf{F} = vDFT_{\mathbf{f}}$ *and* $\mathbf{D} = vDFT_{\mathbf{d}}$. *Define a third vector* $\mathbf{E}$ *having for its kth entry* $E_k = F_k D_k$, *for* $k = 0, ..., N-1$. *Show that* $\mathbf{E}$ *is the vDFT of the vector* $\mathbf{f} * \mathbf{d}$.

The vector $vDFT_{\mathbf{f}}$ can be obtained from the vector $\mathbf{f}$ by means of matrix multiplication by a certain matrix G, called the *DFT matrix*. The matrix G has an inverse that is easily computed and can be used to go from $\mathbf{F} = vDFT_{\mathbf{f}}$ back to the original $\mathbf{f}$. The details are in Exercise 16.2.

Ex. 16.2 *Let* G *be the* N *by* N *matrix whose entries are* $G_{jk} = e^{i(j-1)(k-1)2\pi/N}$. *The matrix* G *is sometimes called the DFT matrix. Show that the inverse of* G *is* $G^{-1} = \frac{1}{N} G^{\dagger}$, *where* $G^{\dagger}$ *is the conjugate transpose of the matrix* G. *Then* $\mathbf{f} * \mathbf{d} = G^{-1}\mathbf{E} = \frac{1}{N} G^{\dagger}\mathbf{E}$.

As mentioned previously, nonperiodic convolution is really a special case of periodic convolution. Extend the $M+1$ by 1 vector a to an $M+N+1$ by 1 vector by appending N zero entries; similarly, extend the vector b to an $M+N+1$ by 1 vector by appending zeros. The vector c is now the periodic convolution of these extended vectors. Therefore, since we have an efficient algorithm for performing periodic convolution, namely the Fast Fourier Transform algorithm (FFT), we have a fast way to do the periodic (and thereby nonperiodic) convolution and polynomial multiplication.

16.5 The Fast Fourier Transform (FFT)

A fundamental problem in signal processing is to estimate the function $F(\omega)$ from finitely many values of its (inverse) Fourier transform, $f(x)$. As we have seen, the DFT is one such estimate. The *fast Fourier transform* (FFT), discovered in 1965 by Cooley and Tukey, is an important and efficient algorithm for calculating the vector DFT [103]. John Tukey has been quoted as saying that his main contribution to this discovery was the firm and often voiced belief that such an algorithm must exist.

To illustrate the main idea underlying the FFT, consider the problem of evaluating a real polynomial $P(x)$ at a point, say $x = c$. Let the polynomial be

$$P(x) = a_0 + a_1 x + a_2 x^2 + ... + a_{2K} x^{2K},$$

where a_{2K} might be zero. Performing the evaluation efficiently by Horner's method,

$$P(c) = (((a_{2K} c + a_{2K-1})c + a_{2K-2})c + a_{2K-3})c + ...,$$

requires $2K$ multiplications, so the complexity is on the order of the degree of the polynomial being evaluated. But suppose we also want $P(-c)$. We can write

$$P(x) = (a_0 + a_2 x^2 + ... + a_{2K} x^{2K}) + x(a_1 + a_3 x^2 + ... + a_{2K-1} x^{2K-2})$$

or

$$P(x) = Q(x^2) + xR(x^2).$$

Therefore, we have $P(c) = Q(c^2) + cR(c^2)$ and $P(-c) = Q(c^2) - cR(c^2)$. If we evaluate $P(c)$ by evaluating $Q(c^2)$ and $R(c^2)$ separately, one more multiplication gives us $P(-c)$ as well. The FFT is based on repeated use of this idea, which turns out to be more powerful when we are using complex exponentials, because of their periodicity.

Say the data are $\{f_n = f(n), n = 0, ..., N-1\}$. The DFT estimate of $F(\omega)$ is the function $DFT_{\mathbf{f}}(\omega)$, defined for ω in $[0, 2\pi]$, and given by

$$DFT_{\mathbf{f}}(\omega) = \sum_{n=0}^{N-1} f(n) e^{in\omega}.$$

The DFT estimate $DFT(\omega)$ is data consistent; its inverse Fourier-transform value at $x = n$ is $f(n)$ for $n = 0, ..., N-1$. The DFT is also used in a more general context in which the f_n are not necessarily values of a function $f(x)$.

Given any complex N-dimensional column vector $\mathbf{f} = (f_0, f_1, ..., f_{N-1})^T$,

define the DFT of the vector $\mathbf{f}$ to be the function $DFT_{\mathbf{f}}(\omega)$, defined for ω in $[0, 2\pi)$, given by

$$DFT_{\mathbf{f}}(\omega) = \sum_{n=0}^{N-1} f_n e^{in\omega}.$$

Let $\mathbf{F}$ be the complex N-dimensional vector $\mathbf{F} = (F_0, F_1, ..., F_{N-1})^T$, where $F_k = DFT_{\mathbf{f}}(2\pi k/N), k = 0, 1, ..., N-1$. So the vector $\mathbf{F}$ consists of N values of the function $DFT_{\mathbf{f}}$, taken at N equispaced points $2\pi/N$ apart in $[0, 2\pi)$.

From the formula for $DFT_{\mathbf{f}}$ we have, for $k = 0, 1, ..., N-1$,

$$F_k = F(2\pi k/N) = \sum_{n=0}^{N-1} f_n e^{2\pi ink/N}. \tag{16.6}$$

To calculate a single F_k requires N multiplications; it would seem that to calculate all N of them would require N^2 multiplications. However, using the FFT algorithm, we can calculate vector $\mathbf{F}$ in approximately $N \log_2(N)$ multiplications.

Suppose that $N = 2M$ is even. We can rewrite Equation (16.6) as follows:

$$F_k = \sum_{m=0}^{M-1} f_{2m} e^{2\pi i(2m)k/N} + \sum_{m=0}^{M-1} f_{2m+1} e^{2\pi i(2m+1)k/N},$$

or, equivalently,

$$F_k = \sum_{m=0}^{M-1} f_{2m} e^{2\pi imk/M} + e^{2\pi ik/N} \sum_{m=0}^{M-1} f_{2m+1} e^{2\pi imk/M}. \tag{16.7}$$

Note that if $0 \leq k \leq M-1$ then

$$F_{k+M} = \sum_{m=0}^{M-1} f_{2m} e^{2\pi imk/M} - e^{2\pi ik/N} \sum_{m=0}^{M-1} f_{2m+1} e^{2\pi imk/M}, \tag{16.8}$$

so there is no additional computational cost in calculating the second half of the entries of $\mathbf{F}$, once we have calculated the first half. The FFT is the algorithm that results when we take full advantage of the savings obtainable by splitting a DFT calculating into two similar calculations of half the size.

We assume now that $N = 2^L$. Notice that if we use Equations (16.7) and (16.8) to calculate vector $\mathbf{F}$, the problem reduces to the calculation of two similar DFT evaluations, both involving half as many entries, followed by one multiplication for each of the k between 0 and $M-1$. We can split these in half as well. The FFT algorithm involves repeated splitting of the calculations of DFTs at each step into two similar DFTs, but with half the

number of entries, followed by as many multiplications as there are entries in either one of these smaller DFTs. We use recursion to calculate the cost $C(N)$ of computing $\mathbf{F}$ using this FFT method. From Equation (16.7) we see that $C(N) = 2C(N/2) + (N/2)$. Applying the same reasoning to get $C(N/2) = 2C(N/4) + (N/4)$, we obtain

$$C(N) = 2C(N/2) + (N/2) = 4C(N/4) + 2(N/2) = ...$$

$$= 2^L C(N/2^L) + L(N/2) = N + L(N/2).$$

Therefore, the cost required to calculate $\mathbf{F}$ is approximately $N \log_2 N$.

From our earlier discussion of discrete linear filters and convolution, we see that the FFT can be used to calculate the periodic convolution (or even the nonperiodic convolution) of finite length vectors.

Finally, let's return to the original context of estimating the Fourier transform $F(\omega)$ of function $f(x)$ from finitely many samples of $f(x)$. If we have N equispaced samples, we can use them to form the vector $\mathbf{f}$ and perform the FFT algorithm to get vector $\mathbf{F}$ consisting of N values of the DFT estimate of $F(\omega)$. It may happen that we wish to calculate more than N values of the DFT estimate, perhaps to produce a smooth looking graph. We can still use the FFT, but we must trick it into thinking we have more data that the N samples we really have. We do this by *zero-padding*. Instead of creating the N-dimensional vector $\mathbf{f}$, we make a longer vector by appending, say, J zeros to the data, to make a vector that has dimension $N + J$. The DFT estimate is still the same function of ω, since we have only included new zero coefficients as fake data; but, the FFT thinks we have $N + J$ data values, so it returns $N + J$ values of the DFT, at $N + J$ equispaced values of ω in $[0, 2\pi)$.

FIGURE 16.1: John Tukey: co-inventor of the FFT.

Chapter 17

Appendix: Self-Adjoint and Normal Linear Operators

17.1 Chapter Summary

We saw previously that if the finite-dimensional vector space V has an orthonormal basis of eigenvectors of the linear operator T, then T is a normal operator. We need to prove the converse: if T is normal, then V has an orthonormal basis consisting of eigenvectors of T. Earlier, we proved this result using matrix representations of linear operators and Schur's Lemma. Now we give a proof within the context of linear operators themselves. Throughout this chapter T will denote an arbitrary linear operator on a finite-dimensional inner-product space V, with adjoint operator T^*.

17.2 The Diagonalization Theorem

In this chapter we present a proof of the following theorem.

Theorem 17.1 *For a linear operator T on a finite-dimensional complex inner-product space V there is an orthonormal basis of eigenvectors if and only if T is normal.*

17.3 Invariant Subspaces

A subspace W of V is said to be *T-invariant* if Tw is in W whenever w is in W. For any T-invariant subspace W, the restriction of T to W, denoted T_W, is a linear operator on W.

For any subspace W, the *orthogonal complement* of W is the space $W^\perp = \{v | \langle w, v\rangle = 0, \text{for all } w \in W\}$.

Proposition 17.1 *Let W be a T-invariant subspace of V. Then*

- *(a) if T is self-adjoint, so is T_W;*
- *(b) $W^\perp$ is T^*-invariant;*
- *(c) if W is both T- and T^*-invariant, then $(T_W)^* = (T^*)_W$;*
- *(d) if W is both T- and T^*-invariant, and T is normal, then T_W is normal.*
- *(e) if T is normal and $Tx = \lambda x$, then $T^*x = \overline{\lambda}x$.*

Ex. 17.1 *Prove Proposition (17.1).*

Proposition 17.2 *If T is normal, $Tu^1 = \lambda_1 u^1$, $Tu^2 = \lambda_2 u^2$, and $\lambda_1 \neq \lambda_2$, then $\langle u^1, u^2\rangle = 0$.*

Ex. 17.2 *Prove Proposition 17.2. Hint: use (e) of Proposition 17.1.*

17.4 Proof of the Diagonalization Theorem

We turn now to the proof of the theorem.

Proof of Theorem 17.1 The proof is by induction on the dimension of the inner-product space V. To begin with, let $N = 1$, so that V is simply the span of some unit vector x. Then any linear operator T on V has $Tx = \lambda x$, for some λ, and the set $\{x\}$ is an orthonormal basis for V.

Now suppose that the theorem is true for every inner-product space of dimension $N - 1$. We know that every linear operator T on V has at least one eigenvector, say x^1, since its characteristic polynomial has at least one distinct root λ_1 in C. Take x^1 to be a unit vector. Let W be the span of the vector x^1, and $W^\perp$ the orthogonal complement of W. Since $Tx^1 = \lambda_1 x^1$ and T is normal, we know that $T^*x^1 = \overline{\lambda_1}x^1$. Therefore, both W and $W^\perp$ are T- and T^*-invariant. Therefore, $T_{W^\perp}$ is normal on $W^\perp$. By the induction hypothesis, we know that $W^\perp$ has an orthonormal basis consisting of $N - 1$ eigenvectors of $T_{W^\perp}$, and, therefore, of T. Augmenting this set with the original x^1, we get an orthonormal basis for all of V. ∎

17.5 Corollaries

The theorem has several important corollaries.

Corollary 17.1 *A self-adjoint linear operator T on a finite-dimensional complex inner-product space V has an orthonormal basis of eigenvectors.*

Corollary 17.2 *Let T be a linear operator on a finite-dimensional real inner-product space V. Then V has an orthonormal basis consisting of eigenvectors of T if and only if T is self-adjoint.*

Corollary 17.3 *Let A be a normal matrix. Then there is a unitary matrix U and diagonal matrix L such that $A = ULU^\dagger$.*

Proving the existence of the orthonormal basis uses essentially the same argument as the induction proof given earlier. The eigenvalues of a self-adjoint linear operator T on a finite-dimensional complex inner-product space are real numbers. If T be a linear operator on a finite-dimensional real inner-product space V and V has an orthonormal basis $\mathcal{U} = \{u^1, ..., u^N\}$ consisting of eigenvectors of T, then we have

$$Tu^n = \lambda_n u^n = \overline{\lambda_n}u^n = T^*u^n,$$

so, since $T = T^*$ on each member of the basis, these operators are the same everywhere, so $T = T^*$ and T is self-adjoint.

Definition 17.1 *A linear operator P on a finite-dimensional inner-product space is a* perpendicular projection *if*

$$P^2 = P = P^*.$$

Corollary 17.4 (The Spectral Theorem) *Let T be a normal operator on a finite-dimensional inner-product space. Then T can be written as*

$$T = \sum_{m=1}^{M} \lambda_m P_m, \tag{17.1}$$

where λ_m, $m = 1, ..., M$ are the distinct eigenvalues of T, P_m is the perpendicular projection

$$P_m = \sum_{n \in I_m} u^n (u^n)^\dagger, \tag{17.2}$$

and

$$I_m = \{n | \lambda_n = \lambda_m\}.$$

Corollary 17.5 *Let T be a normal operator on a finite-dimensional inner-product space. Then there is a complex polynomial $f(z)$ such that*

$$T^* = f(T).$$

Proof: Let $f(z)$ be any polynomial such that $f(\lambda_m) = \overline{\lambda_m}$, for each $m = 1, ..., M$. The assertion then follows, since

$$T^* = \sum_{m=1}^{M} \overline{\lambda_m} P_m,$$

and $P_m P_k = 0$, for $m \neq k$. ∎

17.6 A Counter-Example

We present now an example of a real 2 by 2 matrix A with $A^T A = A A^T$, but with no eigenvectors in R^2. Take $0 < \theta < \pi$ and A to be the matrix

$$A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{17.3}$$

This matrix represents rotation through an angle of θ in R^2. Its transpose represents rotation through the angle $-\theta$. These operations obviously can be done in either order, so the matrix A is normal. But there is no non-zero vector in R^2 that is an eigenvector. Clearly, A is not symmetric.

17.7 Simultaneous Diagonalization

Any linear operator T on a finite-dimensional inner-product space can be written as $T = R + iS$, where both R and S are Hermitian linear operators; simply take $R = \frac{1}{2}(T + T^*)$ and $S = \frac{1}{2i}(T - T^*)$.

Ex. 17.3 *Show that T is a normal operator if and only if $RS = SR$.*

Theorem 17.2 *Let T and U be commuting normal linear operators on a finite-dimensional inner-product space V. There there is an orthonormal basis for V consisting of vectors that are simultaneously eigenvectors for T and for U.*

Proof: For each m let W_m be the range of the perpendicular projection P_m in the spectral theorem expansion for T; that is,

$$W_m = \{x \in V | Tx = \lambda_m x\}.$$

It is easy to see that, for each x in W_m, the vector Ux is in W_m; therefore, the sets W_m are T- and U-invariant. It follows along the lines of our proof of the spectral theorem that the restriction of U to each of the subspaces W_m is a normal operator. Therefore, each W_m has an orthonormal basis consisting of eigenvectors of U. Combining these bases for the W_m gives the desired basis for V. ∎

When T is normal, we have $RS = SR$, so there is an orthonormal basis for V consisting of simultaneous eigenvectors for R and S. It follows that these basis vectors are eigenvectors for T as well. This shows that the spectral theorem for normal operators can be derived from the spectral theorem for Hermitian operators, once we have the simultaneous-diagonalization theorem for commuting Hermitian operators.

It can be shown that, for any family of commuting normal operators on V, there is an orthonormal basis of simultaneous eigenvectors. The recent article by Bouten, van Handel and James [25] describes the use of this result in quantum filtering.

17.8 Quadratic Forms and Congruent Operators

If Q is a Hermitian positive-definite N by N matrix, then the function

$$\phi(x, y) = y^\dagger Q x = \langle x, y \rangle_Q$$

is an inner product on $\mathbb{C}^N$, and the quadratic form

$$\hat{\phi}(x) = x^\dagger Q x = \langle x, x \rangle_Q = \|x\|_Q^2,$$

is the square of the Q-norm. If S is an arbitrary N by N matrix, then the function $\hat{\phi}(x) = x^\dagger S x$ will not be a norm, generally, and $\phi(x, y) = y^\dagger S x$ will not be an inner product, unless S is Hermitian and positive-definite. However, the function $\phi(x, y) = y^\dagger S x$ will still possess some of the properties of an inner product. Such functions are called *sesquilinear forms* or *sesquilinear functionals*.

17.8.1 Sesquilinear Forms

Let V be any complex vector space. A *sesquilinear functional* $\phi(x, y)$ of two variables in V is linear in the first variable and conjugate-linear in the second; that is,

$$\phi(x, \alpha_1 y^1 + \alpha_2 y^2) = \overline{\alpha_1}\phi(x, y^1) + \overline{\alpha_2}\phi(x, y^2);$$

the term *sesquilinear* means *one and one-half linear*. An inner product on V is a special kind of sesquilinear functional.

17.8.2 Quadratic Forms

Any sesquilinear functional has an associated *quadratic form* given by

$$\hat{\phi}(x) = \phi(x, x).$$

If P is any invertible linear operator on V, we can define a new quadratic form by

$$\hat{\phi}_P(x) = \phi(Px, Px).$$

17.8.3 Congruent Linear Operators

Let T be a linear operator on an inner product space V. Then T can be used to define a sesquilinear functional $\phi_T(x, y)$ according to

$$\phi_T(x, y) = \langle Tx, y \rangle. \tag{17.4}$$

For this sesquilinear functional $\phi_T(x, y)$, we have

$$(\hat{\phi}_T)_P(x) = \phi_T(Px, Px) = \langle TPx, Px \rangle = \langle P^*TPx, x \rangle.$$

We say that a linear operator U on V is *congruent* to T if there is an invertible linear operator P with $U = P^*TP$.

In order for the sesquilinear functional $\phi_T(x, y) = \langle Tx, y\rangle$ to be an inner product, it is necessary and sufficient that T be positive-definite; that is, for all x in V,

$$\phi_T(x, x) = \langle Tx, x\rangle \geq 0,$$

with equality if and only if $x = 0$.

17.8.4 Congruent Matrices

Now let $V = \mathbb{C}^N$, with the usual basis and inner product. Linear operators T, U and P are identified with their corresponding matrix representations. We then say that the matrix B is *congruent* to matrix A if there is an invertible matrix P for which $B = P^\dagger AP$.

17.8.5 Does ϕ_T Determine T?

Let T and U be linear operators on an inner product space V. Is it possible for

$$\langle Tx, x\rangle = \langle Ux, x\rangle,$$

for all x in the inner product space V, and yet have $T \neq U$? As we shall see, the answer is "No". First, we answer a simpler question. Is it possible for

$$\langle Tx, y\rangle = \langle Ux, y\rangle,$$

for all x and y, with $T \neq U$? The answer again is "No".

Ex. 17.4 *Show that*

$$\langle Tx, y\rangle = \langle Ux, y\rangle,$$

for all x and y, implies that $T = U$.

We can use the result of the exercise to answer our first question, but first, we need the *polarization identity*.

Ex. 17.5 *Establish the* polarization identity*:*

$$\langle Tx, y\rangle = \frac{1}{4}\langle T(x + y), x + y\rangle - \frac{1}{4}\langle T(x - y), x - y\rangle$$

$$+\frac{i}{4}\langle T(x + iy), x + iy\rangle - \frac{i}{4}\langle T(x - iy), x - iy\rangle.$$

Ex. 17.6 *Show that the answer to our first question is "No"; the quadratic form determines the operator.*

17.8.6 A New Sesquilinear Functional

Given any sesquilinear functional $\phi(x, y)$ and two linear operators P and Q on V, we can define a second sesquilinear functional

$$\psi(x, y) = \phi(Px, Qy).$$

For the sesquilinear functional ϕ_T, we have

$$\psi(x, y) = \phi_T(Px, Qy) = \langle TPx, Qy\rangle = \langle Q^*TPx, y\rangle.$$

Chapter 18

Appendix: Sturm-Liouville Problems

18.1 Chapter Summary

Previously, we discussed self-adjoint linear operators on an inner product space. An important application of this theory is the analysis of linear ordinary differential equations in Sturm-Liouville form. Now the linear operators involved are differential operators, the members of the inner product space are twice differentiable functions of a single variable, and the inner product is defined in terms of an integration. The eigenvectors of the differential operators are *eigenfunctions*. The expansion of members of the inner product space in terms of bases of eigenvectors becomes the famous expansion of functions as sums of Bessel functions, Legendre polynomials and so on.

18.2 Second-Order Linear ODE

The most general form of the second-order linear homogeneous ordinary differential equation with variable coefficients is

$$R(x)y''(x) + P(x)y'(x) + Q(x)y(x) = 0. \tag{18.1}$$

Many differential equations of this type arise when we employ the technique of separating the variables to solve a partial differential equation.

18.2.1 The Standard Form

Of course, dividing through by the function $R(x)$ and renaming the coefficient functions, we can also write Equation (18.1) in the *standard* form as

$$y''(x) + P(x)y'(x) + Q(x)y(x) = 0. \tag{18.2}$$

There are other equivalent forms of Equation (18.1).

18.2.2 The Sturm-Liouville Form

Let $S(x) = \exp(-F(x))$, where $F'(x) = (R'(x) - P(x))/R(x)$. Then we have

$$\frac{d}{dx}(S(x)R(x)) = S(x)P(x).$$

From Equation (18.1) we obtain

$$S(x)R(x)y''(x) + S(x)P(x)y'(x) + S(x)Q(x)y(x) = 0,$$

so that

$$\frac{d}{dx}(S(x)R(x)y'(x)) + S(x)Q(x)y(x) = 0,$$

which then has the form

$$\frac{d}{dx}(p(x)y'(x)) - w(x)q(x)y(x) + \lambda w(x)y(x) = 0, \tag{18.3}$$

where $w(x) > 0$ and λ is a constant. Rewriting Equation (18.3) as

$$-\frac{1}{w(x)}\Big(\frac{d}{dx}(p(x)y'(x))\Big) + q(x)y(x) = \lambda y(x), \tag{18.4}$$

suggests an analogy with the linear algebra eigenvalue problem

$$Ax = \lambda x, \tag{18.5}$$

where A is a square matrix, λ is an eigenvalue of A, and $x \neq 0$ is an associated eigenvector. It also suggests that we study the linear differential operator

$$(Ly)(x) = -\frac{1}{w(x)}\Big(\frac{d}{dx}(p(x)y'(x))\Big) + q(x)y(x) \tag{18.6}$$

to see if we can carry the analogy with linear algebra further.

18.3 Inner Products and Self-Adjoint Differential Operators

For the moment, let V_0 be the vector space of complex-valued integrable functions $f(x)$, defined for $a \leq x \leq b$, for which

$$\int_a^b |f(x)|^2 dx < \infty.$$

For any f and g in V_0 the inner product of f and g is then

$$\langle f, g\rangle = \int_a^b f(x)\overline{g(x)}dx.$$

The linear differential operator

$$Ty = y''$$

is defined for the subspace V_1 of functions $y(x)$ in V_0 that are twice continuously differentiable. Now let V be the subspace of V_1 consisting of all $y(x)$ with $y(a) = y(b) = 0$.

Proposition 18.1 *The operator $Ty = y''$ is self-adjoint on V.*

Proof: We need to show that

$$\langle Ty, z\rangle = \int_a^b y''(x)z(x)dx = \int_a^b y(x)z''(x)dx = \langle y, Tz\rangle,$$

for all $y(x)$ and $z(x)$ in V. This follows immediately from two applications of integration by parts and the restrictions $y(a) = z(a) = y(b) = z(b) = 0$. ∎

It is useful to note that

$$\langle Ty, y\rangle = -\int_a^b |y'(x)|^2 dx \leq 0,$$

for all $y(x)$ in V, which prompts us to say that the differential operator $(-T)y = -y''$ is *non-negative definite*. We then expect all eigenvalues of $-T$ to be non-negative. We know, in particular, that solutions of

$$-y''(x) = \lambda y(x),$$

with $y(0) = y(1) = 0$ are $y_m(x) = \sin(m\pi x)$, and the eigenvalues are $\lambda_m = m^2\pi^2$.

We turn now to the differential operator L given by Equation (18.6). We take V_0 to be all complex-valued integrable functions $f(x)$ with

$$\int_a^b |f(x)|^2 w(x)dx < \infty.$$

We let the inner product of any $f(x)$ and $g(x)$ in V_0 be

$$\langle f, g\rangle = \int_a^b f(x)\overline{g(x)}w(x)dx. \tag{18.7}$$

Let V_1 be all functions in V_0 that are twice continuously differentiable, and V all the functions $y(x)$ in V_1 with $y(a) = y(b) = 0$. We then have the following result.

Theorem 18.1 *The operator L given by Equation (18.6) is self-adjoint on the inner product space V.*

Proof: It is easily seen that

$$(Ly)z - y(Lz) = \frac{1}{w(x)}\frac{d}{dx}(pyz' - py'z).$$

Therefore,

$$\int_a^b \Big((Ly)z - y(Lz)\Big)w(x)dx = (pyz' - py'z)|_a^b = 0.$$

Therefore, $L^* = L$ on V. ∎

It is interesting to note that

$$\langle Ly, y\rangle = \int_a^b p(y')^2 dx + \int_a^b qy^2 dx,$$

so that, if we have $p(x) \geq 0$ and $q(x) \geq 0$, then the operator L is non-negative-definite and we expect all its eigenvalues to be non-negative.

18.4 Orthogonality

Once again, let V be the space of all twice continuously differentiable functions $y(x)$ on $[a, b]$ with $y(a) = y(b) = 0$. Let λ_m and λ_n be distinct eigenvalues of the linear differential operator L given by Equation (18.6), with associated eigenfunctions $u_m(x)$ and $u_n(x)$, respectively. Let the inner product on V be given by Equation (18.7).

Theorem 18.2 *The eigenfunctions $u_m(x)$ and $u_n(x)$ are orthogonal.*

Proof: We have

$$\frac{d}{dx}(p(x)u'_m(x)) - w(x)q(x)u_m(x) = -\lambda_m u_m(x)w(x),$$

and

$$\frac{d}{dx}(p(x)u'_n(x)) - w(x)q(x)u_n(x) = -\lambda_m u_n(x)w(x),$$

so that

$$u_n(x)\frac{d}{dx}(p(x)u'_m(x)) - w(x)q(x)u_m(x)u_n(x) = -\lambda_m u_m(x)u_n(x)w(x)$$

and

$$u_m(x)\frac{d}{dx}(p(x)u'_n(x)) - w(x)q(x)u_m(x)u_n(x) = -\lambda_n u_m(x)u_n(x)w(x).$$

Subtracting one equation from the other, we get

$$u_n(x)\frac{d}{dx}(p(x)u'_m(x)) - u_m(x)\frac{d}{dx}(p(x)u'_n(x)) = (\lambda_n - \lambda_m)u_m(x)u_n(x)w(x).$$

The left side of the previous equation can be written as

$$u_n(x)\frac{d}{dx}(p(x)u'_m(x)) - u_m(x)\frac{d}{dx}(p(x)u'_n(x))$$

$$= \frac{d}{dx}\Big(p(x)u_n(x)u'_m(x) - p(x)u_m(x)u'_n(x)\Big).$$

Therefore,

$$(\lambda_n - \lambda_m)\int_a^b u_m(x)u_n(x)w(x)dx =$$

$$\Big(p(x)u_n(x)u'_m(x) - p(x)u_m(x)u'_n(x)\Big)\Big|_a^b = 0. \tag{18.8}$$

Since $\lambda_m \neq \lambda_n$, it follows that

$$\int_a^b u_m(x)u_n(x)w(x)dx = 0.$$

∎

Note that it is not necessary to have $u_m(a) = u_m(b) = 0$ for all m in order for the right side of Equation (18.8) to be zero; it is enough to have

$$p(a)u_m(a) = p(b)u_m(b) = 0.$$

We shall make use of this fact in our discussion of Bessel's and Legendre's equations.

18.5 Normal Form of Sturm-Liouville Equations

We can put an equation in the Sturm-Liouville form into normal form by first writing it in standard form. There is a better way, though. With the change of variable from x to μ, where

$$\mu(x) = \int_a^x \frac{1}{p(t)}dt,$$

and

$$\mu'(x) = 1/p(x),$$

we can show that

$$\frac{dy}{dx} = \frac{1}{p(x)}\frac{dy}{d\mu}$$

and

$$\frac{d^2y}{dx^2} = \frac{1}{p^2}\frac{d^2y}{d\mu^2} - \frac{p'(x)}{p(x)}\frac{dy}{d\mu}.$$

It follows that

$$\frac{d^2y}{d\mu^2} + q_1(\mu)y = 0. \tag{18.9}$$

For that reason, we study equations of the form

$$y'' + q(x)y = 0. \tag{18.10}$$

18.6 Examples

In this section we present several examples. We shall study these in more detail later in these notes.

18.6.1 Wave Equations

Separating the variables to solve wave equations leads to important ordinary differential equations.

18.6.1.1 The Homogeneous Vibrating String

The wave equation for the homogeneous vibrating string is

$$T\frac{\partial^2 u}{\partial x^2} = m\frac{\partial^2 u}{\partial t^2}, \tag{18.11}$$

where T is the constant tension and m the constant mass density. Separating the variables leads to the differential equation

$$-y''(x) = \lambda y(x). \tag{18.12}$$

18.6.1.2 The Non-homogeneous Vibrating String

When the mass density $m(x)$ varies with x, the resulting wave equation becomes

$$T\frac{\partial^2 u}{\partial x^2} = m(x)\frac{\partial^2 u}{\partial t^2}. \tag{18.13}$$

Separating the variables leads to the differential equation

$$-\frac{T}{m(x)}y''(x) = \lambda y(x). \tag{18.14}$$

18.6.1.3 The Vibrating Hanging Chain

In the hanging chain problem, considered in more detail later, the tension is not constant along the chain, since at each point it depends on the weight of the part of the chain below. The wave equation becomes

$$\frac{\partial^2 u}{\partial t^2} = g\frac{\partial}{\partial x}\Big(x\frac{\partial u}{\partial x}\Big). \tag{18.15}$$

Separating the variables leads to the differential equation

$$-g\frac{d}{dx}\Big(x\frac{dy}{dx}\Big) = \lambda y(x). \tag{18.16}$$

Note that all three of these differential equations have the form

$$Ly = \lambda y,$$

for L given by Equation (18.6).

If we make the change of variable

$$z = 2\sqrt{\frac{\lambda x}{g}},$$

the differential equation in (18.16) becomes

$$z^2\frac{d^2y}{dz^2} + z\frac{dy}{dz} + (z^2 - 0^2)y = 0. \tag{18.17}$$

As we shall see shortly, this is a special case of Bessel's Equation, with $\nu = 0$.

18.6.2 Bessel's Equations

For each non-negative constant ν the associated Bessel's Equation is

$$x^2y''(x) + xy'(x) + (x^2 - \nu^2)y(x) = 0. \tag{18.18}$$

Note that the differential equation in Equation (18.16) has the form $Ly = \lambda y$, but Equation (18.17) was obtained by a change of variable that absorbed the λ into the z, so we do not expect this form of the equation to be in eigenvalue form. However, we can rewrite Equation (18.18) as

$$-\frac{1}{x}\frac{d}{dx}\Big(xy'(x)\Big) + \frac{\nu^2}{x^2}y(x) = y(x), \tag{18.19}$$

which is in the form of a Sturm-Liouville eigenvalue problem, with $w(x) = x = p(x)$, $q(x) = \frac{\nu^2}{x^2}$, and $\lambda = 1$. As we shall discuss again in the chapter on Bessel's Equations, we can use this fact to obtain a family of orthogonal eigenfunctions.

Let us fix ν and denote by $J_\nu(x)$ a solution of Equation (18.18). Then $J_\nu(x)$ solves the eigenvalue problem in Equation (18.19), for $\lambda = 1$. A little calculation shows that for any a the function $u(x) = J_\nu(ax)$ satisfies the eigenvalue problem

$$-\frac{1}{x}\frac{d}{dx}\Big(xy'(x)\Big) + \frac{\nu^2}{x^2}y(x) = a^2y(x). \tag{18.20}$$

Let $\gamma_m > 0$ be the positive roots of $J_\nu(x)$ and define $y_m(x) = J_\nu(\gamma_m x)$ for each m. Then we have

$$-\frac{1}{x}\frac{d}{dx}\Big(xy_m'(x)\Big) + \frac{\nu^2}{x^2}y_m(x) = \gamma_m^2y_m(x), \tag{18.21}$$

and $y_m(1) = 0$ for each m. We have the following result.

Theorem 18.3 *Let γ_m and γ_n be distinct positive zeros of $J_\nu(x)$. Then*

$$\int_0^1 y_m(x)y_n(x)xdx = 0.$$

Proof: The proof is quite similar to the proof of Theorem 18.2. The main point is that now

$$\Big(xy_n(x)y'_m(x) - xy_m(x)y'_n(x)\Big)|_0^1 = 0$$

because $y_m(1) = 0$ for all m and the function $w(x) = x$ is zero when $x = 0$.

18.6.3 Legendre's Equations

Legendre's equations have the form

$$(1 - x^2)y''(x) - 2xy'(x) + p(p + 1)y(x) = 0, \tag{18.22}$$

where p is a constant. When $p = n$ is a non-negative integer, there is a solution $P_n(x)$ that is a polynomial of degree n, containing only even or odd powers, as n is either even or odd; $P_n(x)$ is called the nth Legendre polynomial. Since the differential equation in (18.22) can be written as

$$-\frac{d}{dx}\Big((1 - x^2)y'(x)\Big) = p(p + 1)y(x), \tag{18.23}$$

it is a Sturm-Liouville eigenvalue problem with $w(x) = 1$, $p(x) = (1 - x^2)$ and $q(x) = 0$. The polynomials $P_n(x)$ are eigenfunctions of the Legendre differential operator T given by

$$(Ty)(x) = -\frac{d}{dx}\Big((1 - x^2)y'(x)\Big), \tag{18.24}$$

but we have not imposed any explicit boundary conditions. Nevertheless, we have the following orthogonality theorem.

Theorem 18.4 *For $m \neq n$ we have*

$$\int_{-1}^1 P_m(x)P_n(x)dx = 0.$$

Proof: In this case, Equation (18.8) becomes

$$(\lambda_n - \lambda_m)\int_{-1}^1 P_m(x)P_n(x)dx =$$

$$\Big((1 - x^2)[P_n(x)P'_m(x) - P_m(x)P'_n(x)]\Big)|_{-1}^1 = 0, \tag{18.25}$$

which holds not because we have imposed end-point conditions on the $P_n(x)$, but because $p(x) = 1 - x^2$ is zero at both ends. ∎

18.6.4 Other Famous Examples

Well known examples of Sturm-Liouville problems also include

- **Chebyshev:**

$$\frac{d}{dx}\left(\sqrt{1-x^2}\frac{dy}{dx}\right)+\lambda(1-x^2)^{-1/2}y=0;$$

- **Hermite:**

$$\frac{d}{dx}\left(e^{-x^2}\frac{dy}{dx}\right)+\lambda e^{-x^2}y=0;$$

  and

- **Laguerre:**

$$\frac{d}{dx}\left(xe^{-x}\frac{dy}{dx}\right)+\lambda e^{-x}y=0.$$

Ex. 18.1 *For each of the three differential equations just listed, see if you can determine the interval over which their eigenfunctions will be orthogonal.*

In the next appendix we consider Hermite's Equation and its connection to quantum mechanics.

Chapter 19

Appendix: Matrix and Vector Differentiation

19.1 Chapter Summary

The notation associated with matrix and vector algebra is designed to reduce the number of things we have to think about as we perform our calculations. This notation can be extended to multi-variable calculus, as we show in this chapter.

19.2 Functions of Vectors and Matrices

As we saw in the previous chapter, the least squares approximate solution of $A\mathbf{x} = \mathbf{b}$ is a vector $\hat{\mathbf{x}}$ that minimizes the function $||A\mathbf{x} - \mathbf{b}||$. In our discussion of band-limited extrapolation we showed that, for any nonnegative definite matrix Q, the vector having norm one that maximizes the quadratic form $\mathbf{x}^\dagger Q\mathbf{x}$ is an eigenvector of Q associated with the largest eigenvalue. In the chapter on best linear unbiased optimization we seek a matrix that minimizes a certain function. All of these examples involve what we can call *matrix-vector differentiation*, that is, the differentiation of a function with respect to a matrix or a vector. The gradient of a function of several variables is a well-known example and we begin there. Since there is some possibility of confusion, we adopt the notational convention that boldfaced symbols, such as $\mathbf{x}$, indicate a column vector, while x denotes a scalar.

19.3 Differentiation with Respect to a Vector

Let $\mathbf{x} = (x_1, ..., x_N)^T$ be an N-dimensional real column vector. Let $z = f(\mathbf{x})$ be a real-valued function of the entries of $\mathbf{x}$. The derivative of z with respect to $\mathbf{x}$, also called the *gradient* of z, is the column vector

$$\frac{\partial z}{\partial \mathbf{x}} = \mathbf{a} = (a_1, ..., a_N)^T$$

with entries

$$a_n = \frac{\partial z}{\partial x_n}.$$

Ex. 19.1 *Let* $\mathbf{y}$ *be a fixed real column vector and* $z = f(\mathbf{x}) = \mathbf{y}^T\mathbf{x}$*. Show that*

$$\frac{\partial z}{\partial \mathbf{x}} = \mathbf{y}.$$

Ex. 19.2 *Let* Q *be a real symmetric nonnegative definite matrix, and let* $z = f(\mathbf{x}) = \mathbf{x}^T Q\mathbf{x}$*. Show that the gradient of this quadratic form is*

$$\frac{\partial z}{\partial \mathbf{x}} = 2Q\mathbf{x}.$$

Hint: Write Q as a linear combination of dyads involving the eigenvectors.

Ex. 19.3 *Let* $z = ||A\mathbf{x} - \mathbf{b}||^2$*. Show that*

$$\frac{\partial z}{\partial \mathbf{x}} = 2A^T A\mathbf{x} - 2A^T\mathbf{b}.$$

Hint: Use $z = (A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b})$.

We can also consider the second derivative of $z = f(\mathbf{x})$, which is the *Hessian matrix* of z

$$H = \frac{\partial^2 z}{\partial \mathbf{x}^2} = \nabla^2 f(\mathbf{x})$$

with entries

$$H_{mn} = \frac{\partial^2 z}{\partial x_m \partial x_n}.$$

If the entries of the vector $\mathbf{z} = (z_1, ..., z_M)^T$ are real-valued functions of the vector $\mathbf{x}$, the derivative of $\mathbf{z}$ is the matrix whose mth column is the derivative of the real-valued function z_m. This matrix is usually called the *Jacobian matrix* of $\mathbf{z}$. If $M = N$ the determinant of the Jacobian matrix is the *Jacobian*.

Ex. 19.4 *Suppose* $(u, v) = (u(x, y), v(x, y))$ *is a change of variables from the Cartesian* (x, y) *coordinate system to some other* (u, v) *coordinate system. Let* $\mathbf{x} = (x, y)^T$ *and* $\mathbf{z} = (u(\mathbf{x}), v(\mathbf{x}))^T$.

- **(a)** *Calculate the Jacobian for the rectangular coordinate system obtained by rotating the* (x, y) *system through an angle of* θ.
- **(b)** *Calculate the Jacobian for the transformation from the* (x, y) *system to polar coordinates.*

19.4 Differentiation with Respect to a Matrix

Now we consider real-valued functions $z = f(A)$ of a real matrix A. As an example, for square matrices A we have

$$z = f(A) = \text{trace}\,(A) = \sum_{n=1}^{N} A_{nn},$$

the sum of the entries along the main diagonal of A.

The derivative of $z = f(A)$ is the matrix

$$\frac{\partial z}{\partial A} = B$$

whose entries are

$$B_{mn} = \frac{\partial z}{\partial A_{mn}}.$$

Ex. 19.5 *Show that the derivative of* $\text{trace}\,(A)$ *is* $B = I$*, the identity matrix.*

Ex. 19.6 *Show that the derivative of* $z = \text{trace}\,(DAC)$ *with respect to* A *is*

$$\frac{\partial z}{\partial A} = D^T C^T. \tag{19.1}$$

Consider the function f defined for all J by J positive-definite symmetric matrices by

$$f(Q) = -\log\det(Q). \tag{19.2}$$

Proposition 19.1 *The gradient of $f(Q)$ is $g(Q) = -Q^{-1}$.*

Proof: Let ΔQ be symmetric. Let γ_j, for $j = 1, 2, ..., J$, be the eigenvalues of the symmetric matrix $Q^{-1/2}(\Delta Q)Q^{-1/2}$. These γ_j are then real and are also the eigenvalues of the matrix $Q^{-1}(\Delta Q)$. We shall consider $\|\Delta Q\|$ small, so we may safely assume that $1 + \gamma_j > 0$.

Note that

$$\langle Q^{-1}, \Delta Q\rangle = \sum_{j=1}^{J} \gamma_j,$$

where the inner product on matrices is the trace inner product,

$$\langle A, B\rangle = \text{trace} B^{\dagger} A,$$

since the trace of any square matrix is the sum of its eigenvalues. Then we have

$$f(Q + \Delta Q) - f(Q) = -\log\det(Q + \Delta Q) + \log\det(Q)$$

$$= -\log\det(I + Q^{-1}(\Delta Q)) = -\sum_{j=1}^{J} \log(1 + \gamma_j).$$

From the submultiplicativity of the Frobenius norm we have

$$\|Q^{-1}(\Delta Q)\|/\|Q^{-1}\| \leq \|\Delta Q\| \leq \|Q^{-1}(\Delta Q)\|\|Q\|.$$

Therefore, taking the limit as $\|\Delta Q\|$ goes to zero is equivalent to taking the limit as $\|\gamma\|$ goes to zero, where γ is the vector whose entries are the γ_j.

To show that $g(Q) = -Q^{-1}$ note that

$$\limsup_{\|\Delta Q\|\to 0} \frac{f(Q + \Delta Q) - f(Q) - \langle -Q^{-1}, \Delta Q\rangle}{\|\Delta Q\|}$$

$$= \limsup_{\|\Delta Q\|\to 0} \frac{|-\log\det(Q + \Delta Q) + \log\det(Q) + \langle Q^{-1}, \Delta Q\rangle|}{\|\Delta Q\|}$$

$$\leq \limsup_{\|\gamma\|\to 0} \frac{\sum_{j=1}^{J} |\log(1 + \gamma_j) - \gamma_j|}{\|\gamma\|/\|Q^{-1}\|}$$

$$\leq \|Q^{-1}\| \sum_{j=1}^{J} \lim_{\gamma_j\to 0} \frac{\gamma_j - \log(1 + \gamma_j)}{|\gamma_j|} = 0.$$

∎

We note in passing that the derivative of $\det(DAC)$ with respect to A is the matrix $\det(DAC)(A^{-1})^T$.

Although the trace is not independent of the order of the matrices in a product, it is independent of cyclic permutation of the factors:

$$\text{trace}\,(ABC) = \text{trace}\,(CAB) = \text{trace}\,(BCA).$$

Therefore, the trace is independent of the order for the product of two matrices:

$$\text{trace}\,(AB) = \text{trace}\,(BA).$$

From this fact we conclude that

$$\mathbf{x}^T\mathbf{x} = \text{trace}\,(\mathbf{x}^T\mathbf{x}) = \text{trace}\,(\mathbf{x}\mathbf{x}^T).$$

If $\mathbf{x}$ is a random vector with correlation matrix

$$R = E(\mathbf{x}\mathbf{x}^T),$$

then

$$E(\mathbf{x}^T\mathbf{x}) = E(\text{trace}\,(\mathbf{x}\mathbf{x}^T)) = \text{trace}\,(E(\mathbf{x}\mathbf{x}^T)) = \text{trace}\,(R).$$

We shall use this trick in the chapter on detection.

Ex. 19.7 *Let* $z = \text{trace}\,(A^TCA)$. *Show that the derivative of* z *with respect to the matrix* A *is*

$$\frac{\partial z}{\partial A} = CA + C^TA. \tag{19.3}$$

Therefore, if $C = Q$ *is symmetric, then the derivative is* $2QA$.

We have restricted the discussion here to real matrices and vectors. It often happens that we want to optimize a real quantity with respect to a complex vector. We can rewrite such quantities in terms of the real and imaginary parts of the complex values involved, to reduce everything to the real case just considered. For example, let Q be a hermitian matrix; then the quadratic form $\mathbf{k}^\dagger Q\mathbf{k}$ is real, for any complex vector $\mathbf{k}$. As we saw in Exercise 5.9, we can write the quadratic form entirely in terms of real matrices and vectors.

If $w = u + iv$ is a complex number with real part u and imaginary part v, the function $z = f(w) = |w|^2$ is real-valued. The derivative of $z = f(w)$ with respect to the complex variable w does not exist. When we write $z = u^2 + v^2$, we consider z as a function of the real vector $\mathbf{x} = (u, v)^T$. The derivative of z with respect to $\mathbf{x}$ is the vector $(2u, 2v)^T$.

Similarly, when we consider the real quadratic form $\mathbf{k}^\dagger Q\mathbf{k}$, we view each of the complex entries of the N by 1 vector $\mathbf{k}$ as two real numbers forming a two-dimensional real vector. We then differentiate the quadratic

form with respect to the $2N$ by 1 real vector formed from these real and imaginary parts. If we turn the resulting $2N$ by 1 real vector back into an N by 1 complex vector, we get $2Q\mathbf{k}$ as the derivative; so, it appears as if the formula for differentiating in the real case carries over to the complex case.

19.5 Eigenvectors and Optimization

We can use these results concerning differentiation with respect to a vector to show that eigenvectors solve certain optimization problems.

Consider the problem of maximizing the quadratic form $\mathbf{x}^\dagger Q\mathbf{x}$, subject to $\mathbf{x}^\dagger\mathbf{x} = 1$; here the matrix Q is Hermitian, positive-definite, so that all of its eigenvalues are positive. We use the Lagrange-multiplier approach, with the Lagrangian

$$L(\mathbf{x}, \lambda) = \mathbf{x}^\dagger Q\mathbf{x} - \lambda\mathbf{x}^\dagger\mathbf{x},$$

where the scalar variable λ is the Lagrange multiplier. We differentiate $L(\mathbf{x}, \lambda)$ with respect to $\mathbf{x}$ and set the result equal to zero, obtaining

$$2Q\mathbf{x} - 2\lambda\mathbf{x} = 0,$$

or

$$Q\mathbf{x} = \lambda\mathbf{x}.$$

Therefore, $\mathbf{x}$ is an eigenvector of Q and λ is its eigenvalue. Since

$$\mathbf{x}^\dagger Q\mathbf{x} = \lambda\mathbf{x}^\dagger\mathbf{x} = \lambda,$$

we conclude that $\lambda = \lambda_1$, the largest eigenvalue of Q, and $\mathbf{x} = \mathbf{u}^1$, a norm-one eigenvector associated with λ_1.

Now consider the problem of maximizing $\mathbf{x}^\dagger Q\mathbf{x}$, subject to $\mathbf{x}^\dagger\mathbf{x} = 1$, and $\mathbf{x}^\dagger\mathbf{u}^1 = 0$. The Lagrangian is now

$$L(\mathbf{x}, \lambda, \alpha) = \mathbf{x}^\dagger Q\mathbf{x} - \lambda\mathbf{x}^\dagger\mathbf{x} - \alpha\mathbf{x}^\dagger\mathbf{u}^1.$$

Differentiating with respect to the vector $\mathbf{x}$ and setting the result equal to zero, we find that

$$2Q\mathbf{x} - 2\lambda\mathbf{x} - \alpha\mathbf{u}^1 = 0,$$

or

$$Q\mathbf{x} = \lambda\mathbf{x} + \beta\mathbf{u}^1,$$

for $\beta = \alpha/2$. But, we know that

$$(\mathbf{u}^1)^\dagger Q\mathbf{x} = \lambda(\mathbf{u}^1)^\dagger\mathbf{x} + \beta(\mathbf{u}^1)^\dagger\mathbf{u}^1 = \beta,$$

and

$$(\mathbf{u}^1)^\dagger Q\mathbf{x} = (Q\mathbf{u}^1)^\dagger \mathbf{x} = \lambda_1 (\mathbf{u}^1)^\dagger \mathbf{x} = 0,$$

so $\beta = 0$ and we have

$$Q\mathbf{x} = \lambda \mathbf{x}.$$

Since

$$\mathbf{x}^\dagger Q\mathbf{x} = \lambda,$$

we conclude that $\mathbf{x}$ is a norm-one eigenvector of Q associated with the second-largest eigenvalue, $\lambda = \lambda_2$.

Continuing in this fashion, we can show that the norm-one eigenvector of Q associated with the nth largest eigenvalue λ_n maximizes the quadratic form $\mathbf{x}^\dagger Q\mathbf{x}$, subject to the constraints $\mathbf{x}^\dagger \mathbf{x} = 1$ and $\mathbf{x}^\dagger \mathbf{u}^m = 0$, for $m = 1, 2, ..., n-1$.

Bibliography

[1] Agmon, S. (1954) "The relaxation method for linear inequalities." *Canadian Journal of Mathematics* **6**, pp. 382–392.

[2] Ahn, S., and Fessler, J. (2003) "Globally convergent image reconstruction for emission tomography using relaxed ordered subset algorithms." *IEEE Transactions on Medical Imaging*, **22(5)**, pp. 613–626.

[3] Ahn, S., Fessler, J., Blatt, D., and Hero, A. (2006) "Convergent incremental optimization transfer algorithms: application to tomography." *IEEE Transactions on Medical Imaging*, **25(3)**, pp. 283–296.

[4] Anderson, T. (1972) "Efficient estimation of regression coefficients in time series." *Proc. of Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: The Theory of Statistics* University of California Press, Berkeley, CA, pp. 471–482.

[5] Anderson, A. and Kak, A. (1984) "Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm." *Ultrasonic Imaging* **6**, pp. 81–94.

[6] Ash, R. and Gardner, M. (1975) *Topics in Stochastic Processes* Boston: Academic Press.

[7] Axelsson, O. (1994) *Iterative Solution Methods.* Cambridge, UK: Cambridge University Press.

[8] Baillet, S., Mosher, J., and Leahy, R. (2001) "Electromagnetic Brain Mapping" , *IEEE Signal Processing Magazine*, **18 (6)**, pp. 14–30.

[9] Baillon, J.-B., Bruck, R.E., and Reich, S. (1978) "On the asymptotic behavior of nonexpansive mappings and semigroups in Banach spaces" , *Houston Journal of Mathematics*, **4**, pp. 1–9.

[10] Barrett, H., White, T., and Parra, L. (1997) "List-mode likelihood." *J. Opt. Soc. Am. A* **14**, pp. 2914–2923.

[11] Bauschke, H. (1996) "The approximation of fixed points of compositions of nonexpansive mappings in Hilbert space," *Journal of Mathematical Analysis and Applications*, **202**, pp. 150–159.

[12] Bauschke, H. (2001) "Projection algorithms: results and open problems."in *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications*, Butnariu, D., Censor, Y., and Reich, S., editors, Amsterdam: Elsevier Science. pp. 11–22.

[13] Bauschke, H. and Borwein, J. (1996) "On projection algorithms for solving convex feasibility problems."*SIAM Review* **38 (3)**, pp. 367–426.

[14] Bauschke, H., Borwein, J., and Lewis, A. (1997) "The method of cyclic projections for closed convex sets in Hilbert space."*Contemporary Mathematics: Recent Developments in Optimization Theory and Nonlinear Analysis* **204**, American Mathematical Society, pp. 1–38.

[15] Bauschke, H., and Lewis, A. (2000) "Dykstra's algorithm with Bregman projections: a convergence proof." *Optimization*, **48**, pp. 409–427.

[16] Benson, M. (2003) "What *Galileo* Saw." in *The New Yorker*; reprinted in [97].

[17] Bertero, M. (1992) "Sampling theory, resolution limits and inversion methods."in [19], pp. 71–94.

[18] Bertero, M., and Boccacci, P. (1998) *Introduction to Inverse Problems in Imaging* Bristol, UK: Institute of Physics Publishing.

[19] Bertero, M. and Pike, E.R., editors (1992) *Inverse Problems in Scattering and Imaging* Malvern Physics Series, Adam Hilger, IOP Publishing, London.

[20] Bertsekas, D.P. (1997) "A new class of incremental gradient methods for least squares problems."*SIAM J. Optim.* **7**, pp. 913–926.

[21] Blackman, R. and Tukey, J. (1959) *The Measurement of Power Spectra*. New York: Dover Publications.

[22] Boas, D., Brooks, D., Miller, E., DiMarzio, C., Kilmer, M., Gaudette, R., and Zhang, Q. (2001) "Imaging the body with diffuse optical tomography." *IEEE Signal Processing Magazine*, **18 (6)**, pp. 57–75.

[23] Bochner, S. and Chandrasekharan, K. (1949) *Fourier Transforms*, Annals of Mathematical Studies, No. 19. Princeton, NJ: Princeton University Press.

[24] Born, M. and Wolf, E. (1999) *Principles of Optics: 7th edition*. Cambridge, UK: Cambridge University Press.

[25] Bouten, L., van Handel, R., and James, M. ((2009) "A discrete invitation to quantum filtering and feedback control." *SIAM Review*, **51(2)**, pp. 239–316.

[26] Borwein, J. and Lewis, A. (2000) *Convex Analysis and Nonlinear Optimization*. Canadian Mathematical Society Books in Mathematics, New York: Springer-Verlag.

[27] Bracewell, R.C. (1979) "Image reconstruction in radio astronomy." in [164], pp. 81–104.

[28] Brauer, A. (1946) "Characteristic roots of a matrix." *Duke Mathematics Journal*, **13**, pp. 387–395.

[29] Bregman, L.M. (1967) "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming." *USSR Computational Mathematics and Mathematical Physics* **7**, pp. 200–217.

[30] Bregman, L., Censor, Y., and Reich, S. (1999) "Dykstra's algorithm as the nonlinear extension of Bregman's optimization method." *Journal of Convex Analysis*, **6 (2)**, pp. 319–333.

[31] Brooks, D., and MacLeod, R. (1997) "Electrical imaging of the heart." *IEEE Signal Processing Magazine*, **14 (1)**, pp. 24–42.

[32] Browne, E. (1930) "The characteristic roots of a matrix." *Bulletin of the American Mathematical Society*, **36**, pp. 705–710.

[33] Browne, J. and A. DePierro, A. (1996) "A row-action alternative to the EM algorithm for maximizing likelihoods in emission tomography." *IEEE Trans. Med. Imag.* **15**, pp. 687–699.

[34] Bruck, R.E., and Reich, S. (1977) "Nonexpansive projections and resolvents of accretive operators in Banach spaces." *Houston Journal of Mathematics*, **3**, pp. 459–470.

[35] Bruckstein, A., Donoho, D., and Elad, M. (2009) "From sparse solutions of systems of equations to sparse modeling of signals and images." *SIAM Review*, **51(1)**, pp. 34–81.

[36] Bruyant, P., Sau, J., and Mallet, J.J. (1999) "Noise removal using factor analysis of dynamic structures: application to cardiac gated studies." *Journal of Nuclear Medicine* **40 (10)**, pp. 1676–1682.

[37] Budinger, T., Gullberg, G., and Huesman, R. (1979) "Emission computed tomography." in [164], pp. 147–246.

[38] Burg, J. (1967) "Maximum entropy spectral analysis." *paper presented at the 37th Annual SEG meeting, Oklahoma City, OK.*

[39] Burg, J. (1972) "The relationship between maximum entropy spectra and maximum likelihood spectra." *Geophysics* **37**, pp. 375–376.

[40] Burg, J. (1975) *Maximum Entropy Spectral Analysis*, Ph.D. dissertation, Stanford University.

[41] Byrne, C. and Fitzgerald, R. (1979) "A unifying model for spectrum estimation." in *Proceedings of the RADC Workshop on Spectrum Estimation- October 1979*, Griffiss AFB, Rome, NY.

[42] Byrne, C. and Fitzgerald, R. (1982) "Reconstruction from partial information, with applications to tomography." *SIAM J. Applied Math.* **42(4)**, pp. 933–940.

[43] Byrne, C., Fitzgerald, R., Fiddy, M., Hall, T. and Darling, A. (1983) "Image restoration and resolution enhancement." *J. Opt. Soc. Amer.* **73**, pp. 1481–1487.

[44] Byrne, C., and Wells, D. (1983) "Limit of continuous and discrete finite-band Gerchberg iterative spectrum extrapolation." *Optics Letters* **8 (10)**, pp. 526–527.

[45] Byrne, C. and Fitzgerald, R. (1984) "Spectral estimators that extend the maximum entropy and maximum likelihood methods." *SIAM J. Applied Math.* **44(2)**, pp. 425–442.

[46] Byrne, C., Levine, B.M., and Dainty, J.C. (1984) "Stable estimation of the probability density function of intensity from photon frequency counts." *JOSA Communications* **1(11)**, pp. 1132–1135.

[47] Byrne, C., and Wells, D. (1985) "Optimality of certain iterative and non-iterative data extrapolation procedures." *Journal of Mathematical Analysis and Applications* **111 (1)**, pp. 26–34.

[48] Byrne, C. and Fiddy, M. (1987) "Estimation of continuous object distributions from Fourier magnitude measurements." *JOSA A* **4**, pp. 412–417.

[49] Byrne, C. and Fiddy, M. (1988) "Images as power spectra; reconstruction as Wiener filter approximation." *Inverse Problems* **4**, pp. 399–409.

[50] Byrne, C., Haughton, D., and Jiang, T. (1993) "High-resolution inversion of the discrete Poisson and binomial transformations." *Inverse Problems* **9**, pp. 39–56.

[51] Byrne, C. (1993) "Iterative image reconstruction algorithms based on cross-entropy minimization." *IEEE Transactions on Image Processing* **IP-2**, pp. 96–103.

[52] Byrne, C. (1995) "Erratum and addendum to 'Iterative image reconstruction algorithms based on cross-entropy minimization'." *IEEE Transactions on Image Processing* **IP-4**, pp. 225–226.

[53] Byrne, C. (1996) "Iterative reconstruction algorithms based on cross-entropy minimization."in *Image Models (and their Speech Model Cousins)*, S.E. Levinson and L. Shepp, editors, IMA Volumes in Mathematics and its Applications, Volume 80, pp. 1–11. New York: Springer-Verlag.

[54] Byrne, C. (1996) "Block-iterative methods for image reconstruction from projections." *IEEE Transactions on Image Processing* **IP-5**, pp. 792–794.

[55] Byrne, C. (1997) "Convergent block-iterative algorithms for image reconstruction from inconsistent data." *IEEE Transactions on Image Processing* **IP-6**, pp. 1296–1304.

[56] Byrne, C. (1998) "Accelerating the EMML algorithm and related iterative algorithms by rescaled block-iterative (RBI) methods." *IEEE Transactions on Image Processing* **IP-7**, pp. 100–109.

[57] Byrne, C. (1998) "Iterative deconvolution and deblurring with constraints." *Inverse Problems*, **14**, pp. 1455-1467.

[58] Byrne, C. (1999) "Iterative projection onto convex sets using multiple Bregman distances." *Inverse Problems* **15**, pp. 1295–1313.

[59] Byrne, C. (2000) "Block-iterative interior point optimization methods for image reconstruction from limited data." *Inverse Problems* **16**, pp. 1405–1419.

[60] Byrne, C. (2001) "Bregman-Legendre multidistance projection algorithms for convex feasibility and optimization."in *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications*, Butnariu, D., Censor, Y., and Reich, S., editors, pp. 87–100. Amsterdam: Elsevier Publ.,

[61] Byrne, C. (2001) "Likelihood maximization for list-mode emission tomographic image reconstruction." *IEEE Transactions on Medical Imaging* **20(10)**, pp. 1084–1092.

[62] Byrne, C. (2002) "Iterative oblique projection onto convex sets and the split feasibility problem." *Inverse Problems* **18**, pp. 441–453.

[63] Byrne, C. (2004) "A unified treatment of some iterative algorithms in signal processing and image reconstruction." *Inverse Problems* **20**, pp. 103–120.

[64] Byrne, C. (2005) "Choosing parameters in block-iterative or ordered-subset reconstruction algorithms." *IEEE Transactions on Image Processing*, **14 (3)**, pp. 321–327.

[65] Byrne, C. (2005) *Signal Processing: A Mathematical Approach*, AK Peters, Publ., Wellesley, MA.

[66] Byrne, C. (2007) *Applied Iterative Methods*, AK Peters, Publ., Wellesley, MA.

[67] Byrne, C. (2008) "Sequential unconstrained minimization algorithms for constrained optimization." *Inverse Problems*, **24(1)**, article no. 015013.

[68] Byrne, C. (2009) "Block-iterative algorithms." *International Transactions in Operations Research*, **16(4)**.

[69] Byrne, C. (2009) "Bounds on the largest singular value of a matrix and the convergence of simultaneous and block-iterative algorithms for sparse linear systems." *International Transactions in Operations Research*, **16(4)**.

[70] Byrne,C. (2014) *Iterative Optimization in Inverse Problems*, Taylor and Francis, Publ.

[71] Byrne, C. (2014) *A First Course in Optimization*, Taylor and Francis, Publ.

[72] Byrne, C. (2014) *Signal Processing: A Mathematical Approach*, 2nd edition, Taylor and Francis, Publ.

[73] Byrne, C. and Censor, Y. (2001) "Proximity function minimization using multiple Bregman projections, with applications to split feasibility and Kullback-Leibler distance minimization." *Annals of Operations Research* **105**, pp. 77–98.

[74] Candès, E., and Romberg, J. (2007) "Sparsity and incoherence in compressive sampling." *Inverse Problems*, **23(3)**, pp. 969–985.

[75] Candès, E., Romberg, J., and Tao, T. (2006) "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information." *IEEE Transactions on Information Theory*, **52(2)**, pp. 489–509.

[76] Candès, E., Wakin, M., and Boyd, S. (2007) "Enhancing sparsity by reweighted l_1 minimization." preprint available at http://www.acm.caltech.edu/ emmanuel/publications.html .

[77] Candy, J. (1988) *Signal Processing: The Modern Approach* New York: McGraw-Hill Publ.

[78] Capon, J. (1969) "High-resolution frequency-wavenumber spectrum analysis."*Proc. of the IEEE* **57**, pp. 1408–1418.

[79] Carlson, D., Johnson, C., Lay, D., and Porter, A.D. (2002) *Linear Algebra Gems: Assets for Undergraduates*, The Mathematical Society of America, MAA Notes **59**.

[80] Cederquist, J., Fienup, J., Wackerman, C., Robinson, S., and Kryskowski, D. (1989) "Wave-front phase estimation from Fourier intensity measurements." *Journal of the Optical Society of America A* **6(7)**, pp. 1020–1026.

[81] Censor, Y. (1981) "Row-action methods for huge and sparse systems and their applications." *SIAM Review*, **23**: 444–464.

[82] Censor, Y., Eggermont, P.P.B., and Gordon, D. (1983) "Strong underrelaxation in Kaczmarz's method for inconsistent systems." *Numerische Mathematik* **41**, pp. 83–92.

[83] Censor, Y. and Elfving, T. (1994) "A multi-projection algorithm using Bregman projections in a product space." *Numerical Algorithms*, **8**, pp. 221–239.

[84] Censor, Y., Bortfeld, T., Martin, B., and Trofimov, A. "A unified approach for inversion problems in intensity-modulated radiation therapy." *Physics in Medicine and Biology* 51 (2006), pp. 2353-2365.

[85] Censor, Y., Elfving, T., Herman, G.T., and Nikazad, T. (2008) "On diagonally-relaxed orthogonal projection methods." *SIAM Journal on Scientific Computation*, **30(1)**, pp. 473–504.

[86] Censor, Y., Elfving, T., Kopf, N., and Bortfeld, T. "The multiple-sets split feasibility problem and its application for inverse problems." *Inverse Problems* 21 (2005), pp. 2071-2084.

[87] Censor, Y., Gordon, D., and Gordon, R. (2001) "Component averaging: an efficient iterative parallel algorithm for large and sparse unstructured problems." *Parallel Computing*, **27**, pp. 777–808.

[88] Censor, Y., Gordon, D., and Gordon, R. (2001) "BICAV: A block-iterative, parallel algorithm for sparse systems with pixel-related weighting." *IEEE Transactions on Medical Imaging*, **20**, pp. 1050–1060.

[89] Censor, Y., and Reich, S. (1996) "Iterations of paracontractions and firmly nonexpansive operators with applications to feasibility and optimization" , *Optimization*, **37**, pp. 323–339.

[90] Censor, Y., and Reich, S. (1998) "The Dykstra algorithm for Bregman projections." *Communications in Applied Analysis*, **2**, pp. 323–339.

[91] Censor, Y. and Segman, J. (1987) "On block-iterative maximization."*J. of Information and Optimization Sciences* **8**, pp. 275–291.

[92] Censor, Y. and Zenios, S.A. (1997) *Parallel Optimization: Theory, Algorithms and Applications*. New York: Oxford University Press.

[93] Chang, J.-H., Anderson, J.M.M., and Votaw, J.R. (2004) "Regularized image reconstruction algorithms for positron emission tomography." *IEEE Transactions on Medical Imaging* **23(9)**, pp. 1165–1175.

[94] Childers, D., editor (1978) *Modern Spectral Analysis*. New York:IEEE Press.

[95] Chui, C. and Chen, G. (1991) *Kalman Filtering*, second edition. Berlin: Springer-Verlag.

[96] Cimmino, G. (1938) "Calcolo approssimato per soluzioni dei sistemi di equazioni lineari." *La Ricerca Scientifica XVI, Series II, Anno IX* **1**, pp. 326–333.

[97] Cohen, J. (2010) (editor) *The Best of The Best American Science Writing*, Harper-Collins Publ.

[98] Combettes, P. (1993) "The foundations of set theoretic estimation." *Proceedings of the IEEE* **81 (2)**, pp. 182–208.

[99] Combettes, P. (1996) "The convex feasibility problem in image recovery." *Advances in Imaging and Electron Physics* **95**, pp. 155–270.

[100] Combettes, P. (2000) "Fejér monotonicity in convex optimization." in *Encyclopedia of Optimization*, C.A. Floudas and P. M. Pardalos, editors, Boston: Kluwer Publ.

[101] Combettes, P., and Trussell, J. (1990) "Method of successive projections for finding a common point of sets in a metric space." *Journal of Optimization Theory and Applications* **67 (3)**, pp. 487–507.

[102] Combettes, P., and Wajs, V. (2005) "Signal recovery by proximal forward-backward splitting." *Multi-scale Modeling and Simulation*, **4(4)**, pp. 1168–1200.

[103] Cooley, J. and Tukey, J. (1965) "An algorithm for the machine calculation of complex Fourier series." *Math. Comp.*, **19**, pp. 297–301.

[104] Csiszár, I. (1989) "A geometric interpretation of Darroch and Ratcliff's generalized iterative scaling." *The Annals of Statistics* **17 (3)**, pp. 1409–1413.

[105] Csiszár, I. (1991) "Why least squares and maximum entropy? An axiomatic approach to inference for linear inverse problems." *The Annals of Statistics* **19 (4)**, pp. 2032–2066.

[106] Csiszár, I. and Tusnády, G. (1984) "Information geometry and alternating minimization procedures." *Statistics and Decisions* **Supp. 1**, pp. 205–237.

[107] Cullen, C. (1966) *Matrices and Linear Transformations*. Reading, MA: Addison-Wesley.

[108] Dainty, J. C. and Fiddy, M. (1984) "The essential role of prior knowledge in phase retrieval." *Optica Acta* **31**, pp. 325–330.

[109] Darroch, J. and Ratcliff, D. (1972) "Generalized iterative scaling for log-linear models." *Annals of Mathematical Statistics* **43**, pp. 1470–1480.

[110] Dax, A. (1990) "The convergence of linear stationary iterative processes for solving singular unstructured systems of linear equations." *SIAM Review*, **32**, pp. 611–635.

[111] Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977) "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society, Series B* **37**, pp. 1–38.

[112] De Pierro, A. (1995) "A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography." *IEEE Transactions on Medical Imaging* **14**, pp. 132–137.

[113] De Pierro, A. and Iusem, A. (1990) "On the asymptotic behavior of some alternate smoothing series expansion iterative methods." *Linear Algebra and its Applications* **130**, pp. 3–24.

[114] De Pierro, A., and Yamaguchi, M. (2001) "Fast EM-like methods for maximum 'a posteriori' estimates in emission tomography." *Transactions on Medical Imaging*, **20 (4)**.

[115] Deutsch, F., and Yamada, I. (1998) "Minimizing certain convex functions over the intersection of the fixed point sets of nonexpansive mappings." *Numerical Functional Analysis and Optimization*, **19**, pp. 33–56.

[116] Dhanantwari, A., Stergiopoulos, S., and Iakovidis, I. (2001) "Correcting organ motion artifacts in x-ray CT medical imaging systems by adaptive processing. I. Theory." *Med. Phys.* **28(8)**, pp. 1562–1576.

[117] Dines, K., and Lyttle, R. (1979) "Computerized geophysical tomography." *Proc. IEEE*, **67**, pp. 1065–1073.

[118] Donoho, D. (2006) "Compressed sampling." *IEEE Transactions on Information Theory*, **52 (4)**. (download preprints at http://www.stat.stanford.edu/ donoho/Reports).

[119] Driscoll, P., and Fox, W. (1996) "Presenting the Kuhn-Tucker conditions using a geometric method." *The College Mathematics Journal*, **38 (1)**, pp. 101–108.

[120] Drmač, Z., and Veselić, K. (2008) "New fast and accurate Jacobi SVD algorithms: Part I. " *SIAM J. Matrix Anal. Appl.*, **29**, pp. 1322–1342.

[121] Drmač, Z., and Veselić, K. (2008) "New fast and accurate Jacobi SVD algorithms: Part II. " *SIAM J. Matrix Anal. Appl.*, **29**, pp. 1343–1362.

[122] Duda, R., Hart, P., and Stork, D. (2001) *Pattern Classification*, Wiley.

[123] Duffin, R., Peterson, E., and Zener, C. (1967) *Geometric Programming: Theory and Applications*. New York: Wiley.

[124] Dugundji, J. (1970) *Topology* Boston: Allyn and Bacon, Inc.

[125] Dykstra, R. (1983) "An algorithm for restricted least squares regression." *J. Amer. Statist. Assoc.*, **78 (384)**, pp. 837–842.

[126] Eggermont, P.P.B., Herman, G.T., and Lent, A. (1981) "Iterative algorithms for large partitioned linear systems, with applications to image reconstruction." *Linear Algebra and its Applications* **40**, pp. 37–67.

[127] Elsner, L., Koltracht, L., and Neumann, M. (1992) "Convergence of sequential and asynchronous nonlinear paracontractions." *Numerische Mathematik*, **62**, pp. 305–319.

[128] Erdogan, H., and Fessler, J. (1999) "Fast monotonic algorithms for transmission tomography." *IEEE Transactions on Medical Imaging*, **18(9)**, pp. 801–814.

[129] Everitt, B. and Hand, D. (1981) *Finite Mixture Distributions* London: Chapman and Hall.

[130] Farkas, J. (1902) "Über die Theorie der einfachen Ungleichungen." *J. Reine Angew. Math.*, **124**, pp. 1–24.

[131] Farncombe, T. (2000) "Functional dynamic SPECT imaging using a single slow camera rotation." *Ph.D. thesis, Dept. of Physics, University of British Columbia.*

[132] Farnell, A.B. (1944) "Limits for the characteristic roots of a matrix." *Bulletin of the American Mathematical Society*, **50**, pp. 789–794.

[133] Fernandez, J., Sorzano, C., Marabini, R., and Carazo, J-M. (2006) "Image processing and 3-D reconstruction in electron microscopy." *IEEE Signal Processing Magazine*, **23 (3)**, pp. 84–94.

[134] Fessler, J., Ficaro, E., Clinthorne, N., and Lange, K. (1997) "Grouped-coordinate ascent algorithms for penalized-likelihood transmission image reconstruction." *IEEE Transactions on Medical Imaging*, **16 (2)**, pp. 166–175.

[135] Feynman, R., Leighton, R., and Sands, M. (1963) *The Feynman Lectures on Physics, Vol. 1.* Boston: Addison-Wesley.

[136] Fiddy, M. (1983) "The phase retrieval problem." in *Inverse Optics*, SPIE Proceedings 413 (A.J. Devaney, editor), pp. 176–181.

[137] Fiddy, M. (2008) *private communication.*

[138] Fiacco, A., and McCormick, G. (1990) *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* Philadelphia, PA: SIAM Classics in Mathematics (reissue).

[139] Fienup, J. (1979) "Space object imaging through the turbulent atmosphere." *Optical Engineering* **18**, pp. 529–534.

[140] Fienup, J. (1987) "Reconstruction of a complex-valued object from the modulus of its Fourier transform using a support constraint." *Journal of the Optical Society of America A* **4(1)**, pp. 118–123.

[141] Fleming, W. (1965) *Functions of Several Variables*, Addison-Wesley Publ., Reading, MA.

[142] Frieden, B. R. (1982) *Probability, Statistical Optics and Data Testing.* Berlin: Springer-Verlag.

[143] Gale, D. (1960) *The Theory of Linear Economic Models.* New York: McGraw-Hill.

[144] Gasquet, C. and Witomski, F. (1998) *Fourier Analysis and Applications*. Berlin: Springer-Verlag.

[145] Gelb, A., editor, (1974) *Applied Optimal Estimation*, written by the technical staff of The Analytic Sciences Corporation, MIT Press, Cambridge, MA.

[146] Geman, S., and Geman, D. (1984) "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images." *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-6**, pp. 721–741.

[147] Gerchberg, R. W. (1974) "Super-restoration through error energy reduction." *Optica Acta* **21**, pp. 709–720.

[148] Gifford, H., King, M., de Vries, D., and Soares, E. (2000) "Channelized Hotelling and human observer correlation for lesion detection in hepatic SPECT imaging." *Journal of Nuclear Medicine* **41(3)**, pp. 514–521.

[149] Goebel, K., and Reich, S. (1984) *Uniform Convexity, Hyperbolic Geometry, and Nonexpansive Mappings*, New York: Dekker.

[150] Golshtein, E., and Tretyakov, N. (1996) *Modified Lagrangians and Monotone Maps in Optimization*. New York: John Wiley and Sons, Inc.

[151] Golub, G., and Kahan, W. (1965) "Calculating the singular values and pseudo-inverse of a matrix." *SIAM J. Numer. Anal.*, Ser. B, **2**, pp. 205–224.

[152] Gordan, P. (1873) "Über die Auflösungen linearer Gleichungen mit reelen Coefficienten." *Math. Ann.*, **6**, pp. 23–28.

[153] Gordon, R., Bender, R., and Herman, G.T. (1970) "Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography." *J. Theoret. Biol.* **29**, pp. 471–481.

[154] Gordon, D., and Gordon, R.(2005) "Component-averaged row projections: A robust block-parallel scheme for sparse linear systems." *SIAM Journal on Scientific Computing*, **27**, pp. 1092–1117.

[155] Grcar, J. (2011) "John von Neumann's analysis of Gaussian elimination and the origins of modern numerical analysis." *SIAM Review*, **53 (4)**, pp. 607–682.

[156] Green, P. (1990) "Bayesian reconstructions from emission tomography data using a modified EM algorithm." *IEEE Transactions on Medical Imaging* **9**, pp. 84–93.

[157] Gubin, L.G., Polyak, B.T. and Raik, E.V. (1967) "The method of projections for finding the common point of convex sets." *USSR Computational Mathematics and Mathematical Physics* **7**, pp. 1–24.

[158] Gullberg, G., Huesman, R., Malko, J., Pelc, N., and Budinger, T. (1986) "An attenuated projector-backprojector for iterative SPECT reconstruction." *Physics in Medicine and Biology*, **30**, pp. 799–816.

[159] Haacke, E., Brown, R., Thompson, M., and Venkatesan, R. (1999) *Magnetic Resonance Imaging*. New York: Wiley-Liss.

[160] Hager, W. (1988) *Applied Numerical Linear Algebra*, Englewood Cliffs, NJ: Prentice- Hall.

[161] Hager, B., Clayton, R., Richards, M., Comer, R., and Dziewonsky, A. (1985) "Lower mantle heterogeneity, dynamic typography and the geoid." *Nature*, **313**, pp. 541–545.

[162] Haykin, S. (1985) *Array Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall.

[163] Hebert, T. and Leahy, R. (1989) "A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors." *IEEE Transactions on Medical Imaging* **8**, pp. 194–202.

[164] Herman, G.T. (ed.) (1979) *Image Reconstruction from Projections*, Topics in Applied Physics, Vol. 32, Springer-Verlag, Berlin.

[165] Herman, G.T., and Natterer, F. (eds.) (1981) *Mathematical Aspects of Computerized Tomography*, Lecture Notes in Medical Informatics, Vol. 8, Springer-Verlag, Berlin.

[166] Herman, G.T., Censor, Y., Gordon, D., and Lewitt, R. (1985) "Comment." (on the paper [260]), *Journal of the American Statistical Association* **80**, pp. 22–25.

[167] Herman, G. T. (1999) *private communication*.

[168] Herman, G. T. and Meyer, L. (1993) "Algebraic reconstruction techniques can be made computationally efficient." *IEEE Transactions on Medical Imaging* **12**, pp. 600–609.

[169] Hildreth, C. (1957) "A quadratic programming procedure." *Naval Research Logistics Quarterly* **4**, pp. 79–85. Erratum, p. 361.

[170] Hoffman, K., and Kunze, R. (1965) *Linear Algebra*. Prentice-Hall.

[171] Hogg, R. and Craig, A. (1978) *Introduction to Mathematical Statistics*, MacMillan, New York.

[172] Holte, S., Schmidlin, P., Linden, A., Rosenqvist, G. and Eriksson, L. (1990) "Iterative image reconstruction for positron emission tomography: a study of convergence and quantitation problems." *IEEE Transactions on Nuclear Science* **37**, pp. 629–635.

[173] Hudson, M., Hutton, B., and Larkin, R. (1992) "Accelerated EM reconstruction using ordered subsets." *Journal of Nuclear Medicine*, **33**, p.960.

[174] Hudson, H.M. and Larkin, R.S. (1994) "Accelerated image reconstruction using ordered subsets of projection data." *IEEE Transactions on Medical Imaging* **13**, pp. 601–609.

[175] Huesman, R., Klein, G., Moses, W., Qi, J., Ruetter, B., and Virador, P. (2000) "List-mode maximum likelihood reconstruction applied to positron emission mammography (PEM) with irregular sampling." *IEEE Transactions on Medical Imaging* **19 (5)**, pp. 532–537.

[176] Hutton, B., Kyme, A., Lau, Y., Skerrett, D., and Fulton, R. (2002) "A hybrid 3-D reconstruction/registration algorithm for correction of head motion in emission tomography." *IEEE Transactions on Nuclear Science* **49 (1)**, pp. 188–194.

[177] Jiang, M., and Wang, G. (2003) "Convergence studies on iterative algorithms for image reconstruction." *IEEE Transactions on Medical Imaging*, **22(5)**, pp. 569–579.

[178] Kaczmarz, S. (1937) "Angenäherte Auflösung von Systemen linearer Gleichungen." *Bulletin de l'Academie Polonaise des Sciences et Lettres* **A35**, pp. 355–357.

[179] Kak, A., and Slaney, M. (2001) *Principles of Computerized Tomographic Imaging*. SIAM, Philadelphia, PA.

[180] Kalman, R. (1960) "A new approach to linear filtering and prediction problems." *Trans. ASME, J. Basic Eng.* **82**, pp. 35–45.

[181] Katznelson, Y. (1983) *An Introduction to Harmonic Analysis*. New York: John Wiley and Sons, Inc.

[182] Kheifets, A. (2004) *private communication*.

[183] King, M., Glick, S., Pretorius, H., Wells, G., Gifford, H., Narayanan, M., and Farncombe, T. (2004) "Attenuation, scatter, and spatial resolution compensation in SPECT." in [265], pp. 473–498.

[184] Koltracht, L., and Lancaster, P. (1990) "Constraining strategies for linear iterative processes." *IMA J. Numer. Anal.*, **10**, pp. 555–567.

[185] Körner, T. (1988) *Fourier Analysis*. Cambridge, UK: Cambridge University Press.

[186] Körner, T. (1996) *The Pleasures of Counting*. Cambridge, UK: Cambridge University Press.

[187] Kuhn, H., and Tucker, A. (eds.) (1956) *Linear Inequalities and Related Systems*. Annals of Mathematical Studies, No. 38. New Jersey: Princeton University Press.

[188] Kullback, S. and Leibler, R. (1951) "On information and sufficiency." *Annals of Mathematical Statistics* **22**, pp. 79–86.

[189] Landweber, L. (1951) "An iterative formula for Fredholm integral equations of the first kind." *Amer. J. of Math.* **73**, pp. 615–624.

[190] Lane, R. (1987) "Recovery of complex images from Fourier magnitude." *Optics Communications* **63(1)**, pp. 6–10.

[191] Lange, K. and Carson, R. (1984) "EM reconstruction algorithms for emission and transmission tomography." *Journal of Computer Assisted Tomography* **8**, pp. 306–316.

[192] Lange, K., Bahn, M. and Little, R. (1987) "A theoretical study of some maximum likelihood algorithms for emission and transmission tomography." *IEEE Trans. Med. Imag.* **MI-6(2)**, pp. 106–114.

[193] La Rivière, P., and Vargas, P. (2006) "Monotonic penalized-likelihood image reconstruction for x-ray fluorescence computed tomography." *IEEE Transactions on Medical Imaging* **25(9)**, pp. 1117–1129.

[194] Leahy, R., Hebert, T., and Lee, R. (1989) "Applications of Markov random field models in medical imaging." in *Proceedings of the Conference on Information Processing in Medical Imaging* Lawrence-Berkeley Laboratory, Berkeley, CA.

[195] Leahy, R. and Byrne, C. (2000) "Guest editorial: Recent development in iterative image reconstruction for PET and SPECT." *IEEE Trans. Med. Imag.* **19**, pp. 257–260.

[196] Leis, A., Beck, M., Gruska, M., Best, C., Hegerl, R., Baumeister, W., and Leis, J. (2006) "Cryo-electron tomography of biological specimens." *IEEE Signal Processing Magazine*, **23 (3)**, pp. 95–103.

[197] Lent, A. (1998) *private communication*.

[198] Levitan, E. and Herman, G. (1987) "A maximum *a posteriori* probability expectation maximization algorithm for image reconstruction in emission tomography." *IEEE Transactions on Medical Imaging* **6**, pp. 185–192.

[199] Liao, C.-W., Fiddy, M., and Byrne, C. (1997) "Imaging from the zero locations of far-field intensity data." *Journal of the Optical Society of America -A* **14 (12)**, pp. 3155–3161.

[200] Luenberger, D. (1969) *Optimization by Vector Space Methods*. New York: John Wiley and Sons, Inc.

[201] Lustig, M., Donoho, D., and Pauly, J. (2008) *Magnetic Resonance in Medicine*, to appear.

[202] Magness, T., and McQuire, J. (1962) "Comparison of least squares and minimum variance estimates of regression parameters." *Annals of Mathematical Statistics* **33**, pp. 462–470.

[203] Mann, W. (1953) "Mean value methods in iteration." *Proc. Amer. Math. Soc.* **4**, pp. 506–510.

[204] McLachlan, G.J. and Krishnan, T. (1997) *The EM Algorithm and Extensions*. New York: John Wiley and Sons, Inc.

[205] McVeigh, E., and Ozturk, C. (2001) "Imaging myocardial strain." *IEEE Signal Processing Magazine*, **18 (6)**, pp. 44–56.

[206] Meidunas, E. (2001) "Re-scaled block iterative expectation maximization maximum likelihood (RBI-EMML) abundance estimation and sub-pixel material identification in hyperspectral imagery" *MS thesis, Department of Electrical Engineering, University of Massachusetts Lowell.*

[207] Meijering, E., Smal, I., and Danuser, G. (2006) "Tracking in molecular bioimaging." *IEEE Signal Processing Magazine*, **23 (3)**, pp. 46–53.

[208] Motzkin, T. and Schoenberg, I. (1954) "The relaxation method for linear inequalities." *Canadian Journal of Mathematics* **6**, pp. 393–404.

[209] Mumcuoglu, E., Leahy, R., and Cherry, S. (1996) "Bayesian reconstruction of PET images: Methodology and performance analysis." *Phys. Med. Biol.*, **41**, pp. 1777–1807.

[210] Narayanan, M., Byrne, C. and King, M. (2001) "An interior point iterative maximum-likelihood reconstruction algorithm incorporating upper and lower bounds with application to SPECT transmission imaging." *IEEE Transactions on Medical Imaging* **TMI-20 (4)**, pp. 342–353.

[211] Nash, S. and Sofer, A. (1996) *Linear and Nonlinear Programming*. New York: McGraw-Hill.

[212] Natterer, F. (1986) *Mathematics of Computed Tomography*. New York: John Wiley and Sons, Inc.

[213] Natterer, F., and Wübbeling, F. (2001) *Mathematical Methods in Image Reconstruction*. Philadelphia, PA: SIAM Publ.

[214] Ollinger, J., and Fessler, J. (1997) "Positron-emission tomography." *IEEE Signal Processing Magazine*, **14 (1)**, pp. 43–55.

[215] Oppenheim, A. and Schafer, R. (1975) *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall.

[216] Papoulis, A. (1975) "A new algorithm in spectral analysis and band-limited extrapolation." *IEEE Transactions on Circuits and Systems* **22**, pp. 735–742.

[217] Papoulis, A. (1977) *Signal Analysis*. New York: McGraw-Hill.

[218] Parra, L. and Barrett, H. (1998) "List-mode likelihood: EM algorithm and image quality estimation demonstrated on 2-D PET." *IEEE Transactions on Medical Imaging* **17**, pp. 228–235.

[219] Paulraj, A., Roy, R., and Kailath, T. (1986) "A subspace rotation approach to signal parameter estimation." *Proceedings of the IEEE* **74**, pp. 1044–1045.

[220] Peressini, A., Sullivan, F., and Uhl, J. (1988) *The Mathematics of Nonlinear Programming*. Berlin: Springer-Verlag.

[221] Peters, T. (1981) "Resolution improvement to CT systems using aperture-function correction." in [165], pp. 241–251.

[222] Pretorius, H., King, M., Pan, T-S, deVries, D., Glick, S., and Byrne, C. (1998) "Reducing the influence of the partial volume effect on SPECT activity quantitation with 3D modelling of spatial resolution in iterative reconstruction." *Phys.Med. Biol.* **43**, pp. 407–420.

[223] Pižurica, A., Philips, W., Lemahieu, I., and Acheroy, M. (2003) "A versatile wavelet domain noise filtration technique for medical imaging." *IEEE Transactions on Medical Imaging: Special Issue on Wavelets in Medical Imaging* **22**, pp. 323–331.

[224] Poggio, T. and Smale, S. (2003) "The mathematics of learning: dealing with data." *Notices of the American Mathematical Society* **50 (5)**, pp. 537–544.

[225] Priestley, M. B. (1981) *Spectral Analysis and Time Series*. Boston: Academic Press.

[226] Prony, G.R.B. (1795) "Essai expérimental et analytique sur les lois de la dilatabilité de fluides élastiques et sur celles de la force expansion de la vapeur de l'alcool, à différentes températures." *Journal de l'Ecole Polytechnique* (Paris) **1(2)**, pp. 24–76.

[227] Qi, J., Leahy, R., Cherry, S., Chatziioannou, A., and Farquhar, T. (1998) "High resolution 3D Bayesian image reconstruction using the microPET small animal scanner. " *Phys. Med. Biol.*, **43 (4)**, pp. 1001–1013.

[228] Qian, H. (1990) "Inverse Poisson transformation and shot noise filtering." *Rev. Sci. Instrum.* **61**, pp. 2088–2091.

[229] Quistgaard, J. (1997) "Signal acquisition and processing in medical diagnostic ultrasound." *IEEE Signal processing Magazine*, **14 (1)**, pp. 67–74.

[230] Reich, S. (1979) "Weak convergence theorems for nonexpansive mappings in Banach spaces." *Journal of Mathematical Analysis and Applications*, **67**, pp. 274–276.

[231] Rockafellar, R. (1970) *Convex Analysis*. Princeton, NJ: Princeton University Press.

[232] Rockmore, A., and Macovski, A. (1976) "A maximum likelihood approach to emission image reconstruction from projections." *IEEE Transactions on Nuclear Science*, **NS-23**, pp. 1428–1432.

[233] Sarder, P., and Nehorai, A. (2006) "Deconvolution methods for 3-D fluoresence microscopy images." *IEEE Signal Processing Magazine*, **23 (3)**, pp. 32–45.

[234] Saulnier, G., Blue, R., Newell, J., Isaacson, D., and Edic, P. (2001) "Electrical impedance tomography." *IEEE Signal Processing Magazine*, **18 (6)**, pp. 31–43.

[235] Schmidlin, P. (1972) "Iterative separation of sections in tomographic scintigrams." *Nucl. Med.* **15(1)**.

[236] Schmidt, R. (1981) "A signal subspace approach to multiple emitter location and spectral estimation." *PhD thesis, Stanford University*.

[237] Schultz, L., Blanpied, G., Borozdin, K., *et al.* (2007) "Statistical reconstruction for cosmic ray muon tomography." *IEEE Transactions on Image Processing*, **16(8)**, pp. 1985–1993.

[238] Shaw, C. (2010) "Dimensions in medical imaging: the more the better?" *Proceedings of the IEEE*, **98(1)**, pp. 2–5.

[239] Shepp, L., and Vardi, Y. (1982) "Maximum likelihood reconstruction for emission tomography." *IEEE Transactions on Medical Imaging*, **MI-1**, pp. 113–122.

[240] Shieh, M., Byrne, C., Testorf, M., and Fiddy, M. (2006) "Iterative image reconstruction using prior knowledge." *Journal of the Optical Society of America, A*, **23(6)**, pp. 1292–1300.

[241] Shieh, M., Byrne, C., and Fiddy, M. (2006) "Image reconstruction: a unifying model for resolution enhancement and data extrapolation: Tutorial." *Journal of the Optical Society of America, A*, **23(2)**, pp. 258–266.

[242] Shieh, M., and Byrne, C. (2006) "Image reconstruction from limited Fourier data." *Journal of the Optical Society of America, A*, **23(11)**.

[243] Smith, C. Ray and Grandy, W.T., editors (1985) *Maximum-Entropy and Bayesian Methods in Inverse Problems*. Dordrecht: Reidel Publ.

[244] Smith, C. Ray and Erickson, G., editors (1987) *Maximum-Entropy and Bayesian Spectral Analysis and Estimation Problems*. Dordrecht: Reidel Publ.

[245] Soares, E., Byrne, C., Glick, S., Appledorn, R., and King, M. (1993) "Implementation and evaluation of an analytic solution to the photon attenuation and nonstationary resolution reconstruction problem in SPECT." *IEEE Transactions on Nuclear Science*, **40 (4)**, pp. 1231–1237.

[246] Stark, H. and Yang, Y. (1998) *Vector Space Projections: A Numerical Approach to Signal and Image Processing, Neural Nets and Optics*. New York: John Wiley and Sons, Inc.

[247] Stiemke, E. (1915) "Über positive Lösungen homogener linearer Gleichungen." *Math. Ann*, **76**, pp. 340–342.

[248] Strang, G. (1980) *Linear Algebra and its Applications*. New York: Academic Press.

[249] Tanabe, K. (1971) "Projection method for solving a singular system of linear equations and its applications." *Numer. Math.* **17**, pp. 203–214.

[250] Therrien, C. (1992) *Discrete Random Signals and Statistical Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall.

[251] Thévenaz, P., Blu, T., and Unser, M. (2000) "Interpolation revisited." *IEEE Transactions on Medical Imaging*, **19**, pp.739–758.

[252] Tsui, B., Gullberg, G., Edgerton, E., Ballard, J., Perry, J., McCartney, W., and Berg, J. (1989) "Correction of non-uniform attenuation in cardiac SPECT imaging." *Journal of Nuclear Medicine*, **30(4)**, pp. 497–507.

[253] Tucker, A. (1956) "Dual systems of homogeneous linear relations." in [187], pp. 3–18.

[254] Twomey, S. (1996) *Introduction to the Mathematics of Inversion in Remote Sensing and Indirect Measurement*. New York: Dover Publ.

[255] Udpa, L., Ayres, V., Fan, Y., Chen, Q., Kumar, S. (2006) "Deconvolution of atomic force microscopy data for cellular and molecular imaging." *IEEE Signal Processing Magazine*, **23 (3)**, pp. 73–83.

[256] Unser, M. (1999) "Splines: A perfect fit for signal and image processing." *IEEE Signal Processing Magazine*, **16**, pp. 22–38.

[257] Van Trees, H. (1968) *Detection, Estimation and Modulation Theory*. New York: John Wiley and Sons, Inc.

[258] van der Sluis, A. (1969) "Condition numbers and equilibration of matrices." *Numer. Math.*, **14**, pp. 14–23.

[259] van der Sluis, A., and van der Vorst, H.A. (1990) "SIRT- and CG-type methods for the iterative solution of sparse linear least-squares problems." *Linear Algebra and its Applications*, **130**, pp. 257–302.

[260] Vardi, Y., Shepp, L.A. and Kaufman, L. (1985) "A statistical model for positron emission tomography." *Journal of the American Statistical Association* **80**, pp. 8–20.

[261] von Neumann, J., and Morgenstern, O. (1944) *Theory of Games and Economic Behavior*. New Jersey: Princeton University Press.

[262] von Neumann, J., and Goldstine, H. H. (1947) "Numerical inverting of matrices of high order." Bulletin of the American Mathematical Society, **53**, pp. 1021–1099.

[263] Vonesch, C., Aguet, F., Vonesch, J-L, and Unser, M. (2006) "The colored revolution in bio-imaging." *IEEE Signal Processing Magazine*, **23 (3)**, pp. 20–31.

[264] Weintraub, K. (2012) "Bloodless brain surgery." in *The Boston Globe*, April 23, 2012, page B5.

[265] Wernick, M. and Aarsvold, J., editors (2004) *Emission Tomography: The Fundamentals of PET and SPECT*. San Diego: Elsevier Academic Press.

[266] Wiener, N. (1949) *Time Series*. Cambridge, MA: MIT Press.

[267] Wright, G.A. (1997) "Magnetic resonance imaging." *IEEE Signal Processing Magazine*, **14 (1)**, pp. 56–66.

[268] Wright, M. (2009) "The dual flow between linear algebra and optimization." view-graphs of talk given at the History of Numerical Linear Algebra Minisymposium - Part II, SIAM Conference on Applied Linear Algebra, Monterey, CA, October 28, 2009.

[269] Wright, W., Pridham, R., and Kay, S. (1981) "Digital signal processing for sonar." *Proc. IEEE* **69**, pp. 1451–1506.

[270] Yang, Q. (2004) "The relaxed CQ algorithm solving the split feasibility problem." *Inverse Problems*, **20**, pp. 1261–1266.

[271] Yin, M. (2011) "About triangular matrices." , seminar notes.

[272] Yin, W., and Zhang, Y. (2008) "Extracting salient features from less data via l_1-minimization." *SIAG/OPT Views-and-News*, **19(1)**, pp. 11–19.

[273] Youla, D. (1978) "Generalized image restoration by the method of alternating projections." *IEEE Transactions on Circuits and Systems* **CAS-25 (9)**, pp. 694–702.

[274] Youla, D.C. (1987) "Mathematical theory of image restoration by the method of convex projections." in *Image Recovery: Theory and Applications*, pp. 29–78, Stark, H., editor (1987) Orlando FL: Academic Press.

[275] Young, R. (1980) *An Introduction to Nonharmonic Fourier Analysis*. Boston: Academic Press.

[276] Zhou, X., and Wong, S. (2006) "Informatics challenges of high-throughput microscopy." *IEEE Signal Processing Magazine*, **23 (3)**, pp. 63–72.

[277] Zimmer, C., Zhang, B., Dufour, A., Thébaud, A., Berlemont, S., Meas-Yedid, V., and Marin, J-C. (2006) "On the digital trail of mobile cells." *IEEE Signal Processing Magazine*, **23 (3)**, pp. 54–62.

Index