Applied and Computational Linear Algebra: A First Course

Charles L. Byrne
Department of Mathematical Sciences
University of Massachusetts Lowell
Lowell, MA 01854

December 10, 2009

(The most recent version is available as a pdf file at
http://faculty.uml.edu/cbyrne/cbyrne.html)

Contents

Part I

Preliminaries

Chapter 1

Introduction

1.1 Background

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. 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, and full of seemingly endless discussion of the representation of linear transformations with respect to various bases. With the growth of computer power came the *digitization* of many problems formally analyzed in terms of functions of continuous variables. Partial differential operators became matrices, pictures became matrices, and the need for fast algorithms to solve large systems of linear equations turned linear algebra into a branch of applied and computational mathematics. Old but forgotten topics in linear algebra, such as singular-value decomposition, were resurrected, and new algorithms, such as the simplex method and the fast Fourier transform (FFT), revolutionized the field. As algorithms came increasingly to be 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.

1.2 New Uses for Old Methods

The traditional topics of linear algebra, the geometry of Euclidean spaces, solving systems of linear equations and finding eigenvectors and eigenval-

ues, 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.

1.3 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.

1.4 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.

1.5 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 R^J having non-negative entries; an interior-point iterative method would have x^k non-negative for each k.

1.6 Operators

Most of the iterative algorithms we shall study involve an *operator*, that is, a function $T : R^J \to R^J$. 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. In gradient descent methods 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 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 R^J, that is, the set

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

1.7 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 *partial gradient* methods.

Chapter 2

An Overview of Applications

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.

2.1 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, 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).

2.1.1 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 [220] 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.

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".

2.1.2 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.

2.1.3 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.

2.1.4 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.

2.1.5 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).

2.2 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, which includes 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 [245]. 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.

2.2.1 Coincidence-Detection PET

In PET 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.

2.2.2 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 within the body of the patient. In SPECT the radionuclide emits single 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.

2.2.3 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 [158, 201, 58].

2.2.4 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 [166]: 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 compu-

tational 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.

2.2.5 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.

2.2.6 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.

2.2.7 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 [176]. 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.

2.3 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).

2.3.1 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.

2.3.2 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.

2.3.3 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*.

2.3.4 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.

2.3.5 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 [143].

2.3.6 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.

2.3.7 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 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 discusses previously, particularly methods for solving the convex feasibility problem.

2.4.1 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).

2.4.2 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.

2.4.3 Convex Feasibility and IMRT

The CQ algorithm [59, 60] is an iterative algorithm for solving the convex feasibility problem. Because it is particularly simple to implement in many cases, it has become the focus of recent work in IMRT. In [77] Censor *et al.* extend the CQ algorithm to solve what they call the *multiple-set split feasibility problem* (MSSFP) . In the sequel [75] 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.

2.5 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 them 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.

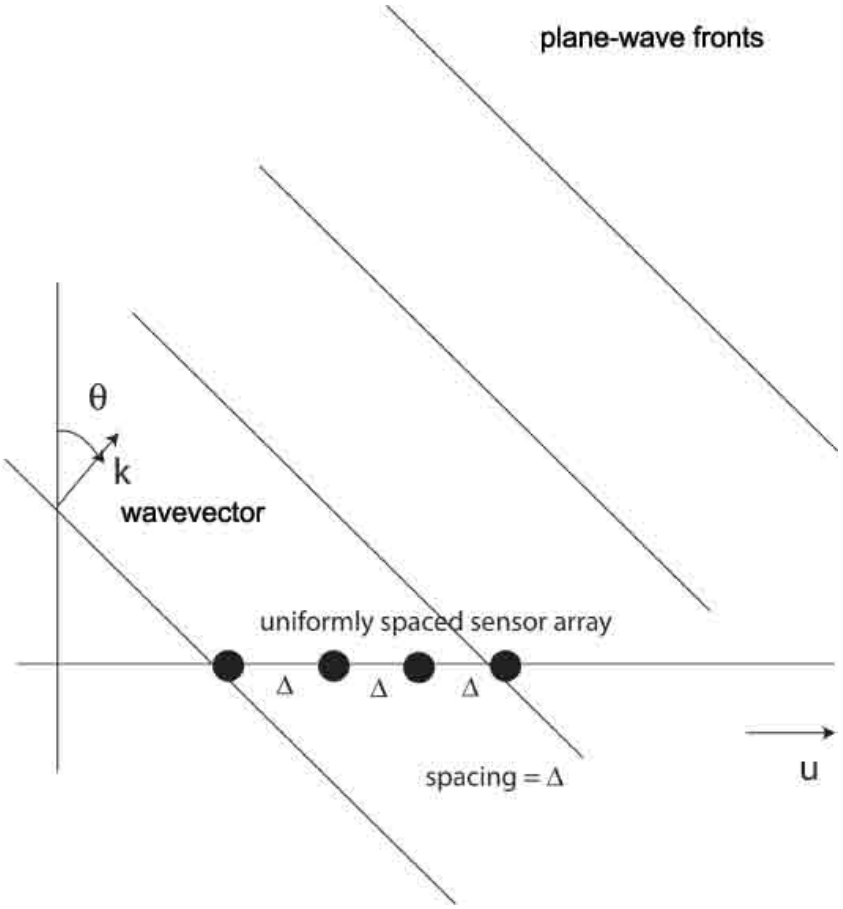

Figure 2.1: A uniform line array sensing a plane-wave field.

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.

2.6 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-norm solution would seem to be a reasonable reconstruction method.

The minimum-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-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-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-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-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-norm reconstruction, using as the weight vector the reciprocals of the values of the prior image. This minimum-weighted-norm reconstruction is shown on the lower left; it is clearly almost the same as the original image. The calculation of the minimum-weighted norm solution can be done iteratively using the ART algorithm [222].

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.

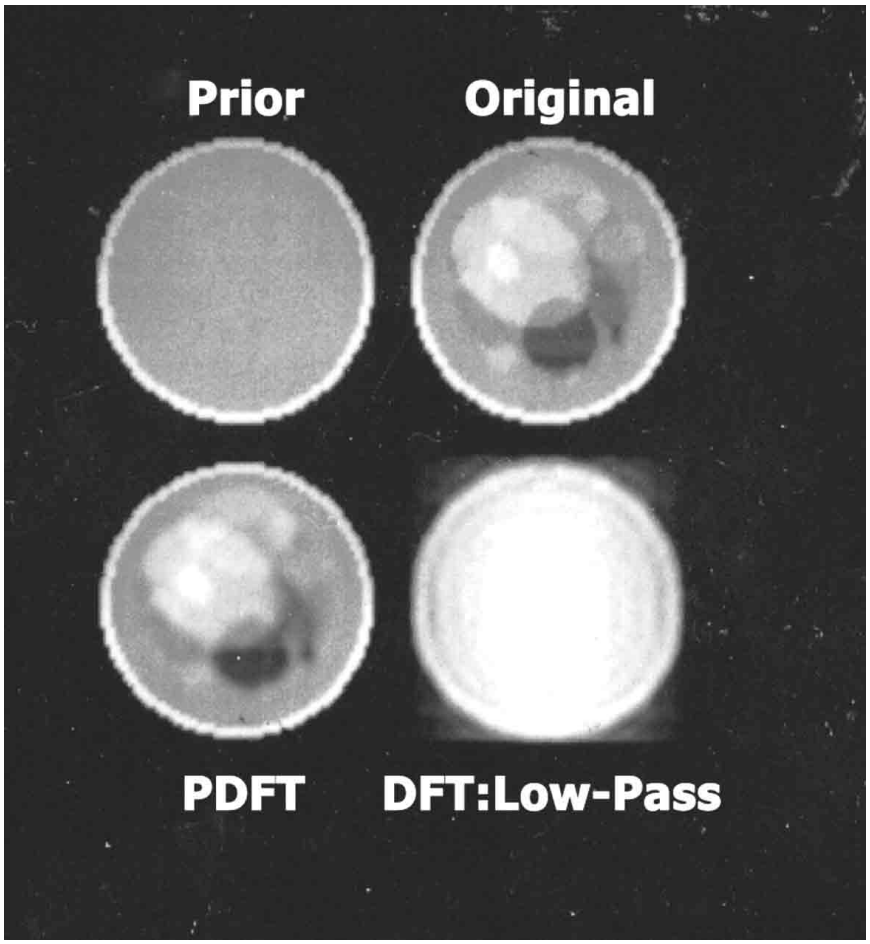

Figure 2.2: Extracting information in image reconstruction.

Chapter 3

Urn Models for Remote Sensing

There seems to be a tradition in physics of using simple models or examples involving urns and marbles to illustrate important principles. In keeping with that tradition, we give an urn model to illustrate various aspects of remote sensing, and apply the model to tomography.

3.1 The Urn Model for Remote Sensing

Suppose that we have J urns numbered $j = 1, ..., J$, each containing marbles of various colors. Suppose that there are I colors, numbered $i = 1, ..., I$. Suppose also that there is a box containing N small pieces of paper, and on each piece is written the number of one of the J urns. Assume that N is much larger than J. Assume that I know the precise contents of each urn. My objective is to determine the precise contents of the box, that is, to estimate the number of pieces of paper corresponding to each of the numbers $j = 1, ..., J$.

Out of my view, my assistant removes one piece of paper from the box, takes one marble from the indicated urn, announces to me the color of the marble, and then replaces both the piece of paper and the marble. This action is repeated many times, at the end of which I have a long list of colors. This list is my data, from which I must determine the contents of the box.

This is a form of remote sensing; what we have access to is not what we are really interested in, but only related to it in some way. Sometimes such data is called "incomplete data" , in contrast to the "complete data" , which would be the list of the actual urn numbers drawn from the box.

If all the marbles of one color are in a single urn, the problem is trivial;

when I hear a color, I know immediately which urn contained that marble. My list of colors is then a list of urn numbers; I have the complete data now. My best estimate of the number of pieces of paper containing the urn number j is then simply N times the proportion of draws that resulted in urn j being selected.

At the other extreme, suppose two urns had identical contents. Then I could not distinguish one urn from the other and would be unable to estimate more than the total number of pieces of paper containing either of the two urn numbers.

Generally, the more the contents of the urns differ, the easier the task of estimating the contents of the box. In remote sensing applications, these issues affect our ability to resolve individual components contributing to the data.

To introduce some mathematics, let us denote by x_j the proportion of the pieces of paper that have the number j written on them. Let P_{ij} be the proportion of the marbles in urn j that have the color i. Let y_i be the proportion of times the color i occurs on the list of colors. The expected proportion of times i occurs on the list is $E(y_i) = \sum_{j=1}^{J} P_{ij}x_j = (Px)_i$, where P is the I by J matrix with entries P_{ij} and x is the J by 1 column vector with entries x_j. A reasonable way to estimate x is to replace $E(y_i)$ with the actual y_i and solve the system of linear equations $y_i = \sum_{j=1}^{J} P_{ij}x_j$, $i = 1, ..., I$. Of course, we require that the x_j be nonnegative and sum to one, so special algorithms may be needed to find such solutions. If there are two urns, j_1 and j_2, such that P_{ij_1} and P_{ij_2} are nearly equal for all i, then we will have a hard time distinguishing x_{j_1} and x_{j_2}.

In a number of applications that fit this model, such as medical tomography, the values x_j are taken to be parameters, the data y_i are statistics, and the x_j are estimated by adopting a probabilistic model and maximizing the likelihood function. iterative algorithms, such as the expectation maximization (EMML) algorithm are often used for such problems.

3.2 The Urn Model in Tomography

Now we apply this simple model to transmission and emission tomography.

3.2.1 The Case of SPECT

In the SPECT case, let there be J pixels or voxels, numbered $j = 1, ..., J$ and I detectors, numbered $i = 1, ..., I$. Let P_{ij} be the probability that a photon emitted at pixel j will be detected at detector i; we assume these probabilities are known to us. Let y_i be the proportion of the total photon count that was recorded at the ith detector. Denote by x_j the (unknown) proportion of the total photon count that was emitted from

pixel j. Selecting an urn randomly is analogous to selecting which pixel will be the next to emit a photon. Learning the color of the marble is analogous to learning where the photon was detected; for simplicity we are assuming that all emitted photons are detected, but this is not essential. The data we have, the counts at each detector, constitute the "incomplete data" ; the "complete data" would be the counts of emissions from each of the J pixels.

If the pixels numbered j_1 and j_2 are neighbors, then we would expect P_{ij_1} and P_{ij_2} to be almost equal, for every i. This makes it difficult to estimate accurately the separate quantities x_{j_1} and x_{j_2}, which is a resolution problem.

We can determine the x_j by finding nonnegative solutions of the system $y_i = \sum_{j=1}^{J} P_{ij}x_j$; this is what the various iterative algorithms, such as MART, EMML and RBI-EMML, seek to do.

3.2.2 The Case of PET

In the PET case, let there be J pixels or voxels, numbered $j = 1, ..., J$ and I lines of response (LOR), numbered $i = 1, ..., I$. Let P_{ij} be the probability that a positron emitted at pixel j will result in a coincidence detection associated with LOR i; we assume these probabilities are known to us. Let y_i be the proportion of the total detections that was associated with the ith LOR. Denote by x_j the (unknown) proportion of the total count that was due to a positron emitted from pixel j. Selecting an urn randomly is analogous to selecting which pixel will be the next to emit a positron. Learning the color of the marble is analogous to learning which LOR was detected; again, for simplicity we are assuming that all emitted positrons are detected, but this is not essential. As in the SPECT case, we can determine the x_j by finding nonnegative solutions of the system $y_i = \sum_{j=1}^{J} P_{ij}x_j$.

3.2.3 The Case of Transmission Tomography

Assume that x-ray beams are sent along I line segments, numbered $i = 1, ..., I$, and that the initial strength of each beam is known. By measuring the final strength, we determine the drop in intensity due to absorption along the ith line segment. Associated with each line segment we then have the proportion of transmitted photons that were absorbed, but we do not know where along the line segment the absorption took place. The proportion of absorbed photons for each line is our data, and corresponds to the proportion of each color in the list. The rate of change of the intensity of the x-ray beam as it passes through the jth pixel is proportional to the intensity itself, to P_{ij}, the length of the ith segment that is within the jth pixel, and to x_j, the amount of attenuating material present in the jth

pixel. Therefore, the intensity of the x-ray beam leaving the jth pixel is the product of the intensity of the beam upon entering the jth pixel and the decay term, $e^{-P_{ij}x_j}$.

The "complete data" is the proportion of photons entering the jth pixel that were absorbed within it; the "incomplete data" is the proportion of photons sent along each line segment that were absorbed. Selecting the jth urn is analogous to having an absorption occurring at the jth pixel. Knowing that an absorption has occurred along the ith line segment does tell us that an absorption occurred at one of the pixels that intersections that line segment, but that is analogous to knowing that there are certain urns that are the only ones that contain the ith color.

The (measured) intensity of the beam at the end of the ith line segment is $e^{-(Px)_i}$ times the (known) intensity of the beam when it began its journey along the ith line segment. Taking logs, we obtain a system of linear equations which we can solve for the x_j.

3.3 Hidden Markov Models

Hidden Markov models (HMM) are increasingly important in speech processing, optical character recognition and DNA sequence analysis. In this section we illustrate HMM using a modification of the urn model.

Suppose, once again, that we have J urns, indexed by $j = 1, ..., J$ and I colors of marbles, indexed by $i = 1, ..., I$. Associated with each of the J urns is a box, containing a large number of pieces of paper, with the number of one urn written on each piece. My assistant selects one box, say the j_0th box, to start the experiment. He draws a piece of paper from that box, reads the number written on it, call it j_1, goes to the urn with the number j_1 and draws out a marble. He then announces the color. He then draws a piece of paper from box number j_1, reads the next number, say j_2, proceeds to urn number j_2, etc. After N marbles have been drawn, the only data I have is a list of colors, $\mathbf{c} = \{c_1, c_2, ..., c_N\}$.

According to the hidden Markov model, the probability that my assistant will proceed from the urn numbered k to the urn numbered j is b_{jk}, with $\sum_{j=1}^{J} b_{jk} = 1$ for all k, and the probability that the color c_i will be drawn from the urn numbered j is a_{ij}, with $\sum_{i=1}^{I} a_{ij} = 1.$ for all j. The colors announced are the *visible states*, while the unannounced urn numbers are the *hidden states*.

There are several distinct objectives one can have, when using HMM. We assume that the data is the list of colors, $\mathbf{c}$.

- **Evaluation:** For given probabilities a_{ij} and b_{jk}, what is the probability that the list $\mathbf{c}$ was generated according to the HMM? Here, the objective is to see if the model is a good description of the data.

- **Decoding:** Given the model, the probabilities and the list **c**, what list $\mathbf{j} = \{j_1, j_2, ..., j_N\}$ of potential visited urns is the most likely? Now, we want to infer the hidden states from the visible ones.

- **Learning:** We are told that there are J urns and I colors, but are not told the probabilities a_{ij} and b_{jk}. We are given several data vectors **c** generated by the HMM; these are the *training sets*. The objective is to learn the probabilities.

Once again, the EMML algorithm can play a role in solving these problems [109].

Chapter 4

The ART and MART

4.1 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 under-determined, and solutions satisfying certain additional constraints, such as non-negativity, are required. The ART and the MART are two iterative algorithms that are designed to address these issues. In this chapter we give an overview of these methods; later, we shall revisit them in more detail.

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 non-negative systems for which non-negative 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. In the appendix "Geometric Programming and the MART" we describe the use of MART to solve the dual geometric programming problem. 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.

4.2 The ART in Tomography

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.

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, $Ax = b$.

4.3 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 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

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

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

$$x^{k+1} = P_{i(k)}x^k. \tag{4.4}$$

Because the ART uses only a single equation at each step, it has been called a *row-action* method. Figures 4.2 and 4.3 illustrate the behavior of the ART.

4.3.1 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

$$x_j = z_j + \overline{A_{ij}}(b_i - (Az)_i)/\sum_{m=1}^{J} |A_{im}|^2. \tag{4.5}$$

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

$$\sum_{j=1}^{J} |A_{ij}|^2 = 1, \tag{4.6}$$

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

$$x_j^{k+1} = x_j^k + \overline{A_{ij}}(b_i - (Ax^k)_i). \tag{4.7}$$

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.

4.3.2 When $Ax = b$ Has Solutions

For the consistent case, in which the system $Ax = b$ has exact solutions, we have the following result.

Theorem 4.1 *Let $A\hat{x} = b$ and let x^0 be arbitrary. Let $\{x^k\}$ be generated by Equation (4.7). Then the sequence $\{||\hat{x} - x^k||_2\}$ is decreasing and $\{x^k\}$ converges to the solution of $Ax = b$ closest to x^0.*

4.3.3 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. There are several open questions about the limit cycle.

Open Question: For a fixed ordering, does the limit cycle depend on the initial vector x^0? If so, how?

4.3.4 The Geometric Least-Squares Solution

When the system $Ax = b$ has no solutions, it is reasonable to seek an approximate solution, such as the *least squares* solution, $x_{LS} = (A^\dagger A)^{-1}A^\dagger b$, 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

$$(A^\dagger A)^{-1}A^\dagger b, \tag{4.8}$$

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

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

and these need not be the same.

A simple example is the following. Consider the system

$$x = 1$$

$$x = 2, \tag{4.10}$$

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

$$2x = 2$$

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

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

Definition 4.1 *The* geometric least-squares *solution of $Ax = b$ is the 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 $x = 1.2$.

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. It is known 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.

4.4 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 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 visible.

4.4.1 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

$$x_j^{k+1} = x_j^k + \frac{1}{|L_i|}(b_i - (Ax^k)_i), \tag{4.12}$$

for j in L_i, and

$$x_j^{k+1} = x_j^k, \tag{4.13}$$

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

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i} \Big), \tag{4.14}$$

for those j in L_i, and

$$x_j^{k+1} = x_j^k, \tag{4.15}$$

otherwise. Therefore, we can write the iterative step as

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i} \Big)^{A_{ij}}. \tag{4.16}$$

4.4.2 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 made 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 4.1 (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*

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

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.

4.4.3 Cross-Entropy

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

$$KL(a, b) = a \log \frac{a}{b} + b - a, \tag{4.18}$$

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

$$KL(x, z) = \sum_{j=1}^{J} KL(x_j, z_j). \tag{4.19}$$

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.

4.4.4 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 one, then the MART converges to the solution that maximizes the Shannon entropy,

$$SE(x) = \sum_{j=1}^{J} x_j \log x_j - x_j. \tag{4.20}$$

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.

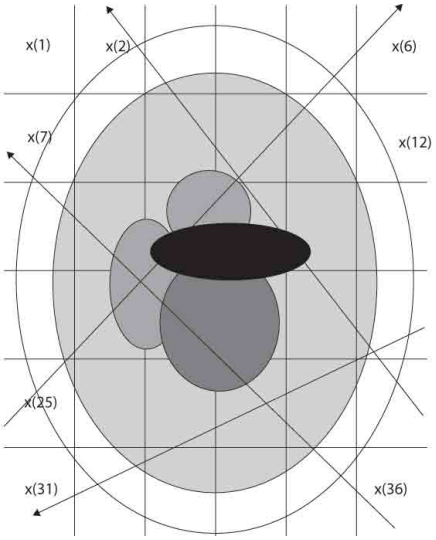

Figure 4.1: Line integrals through a discretized object.

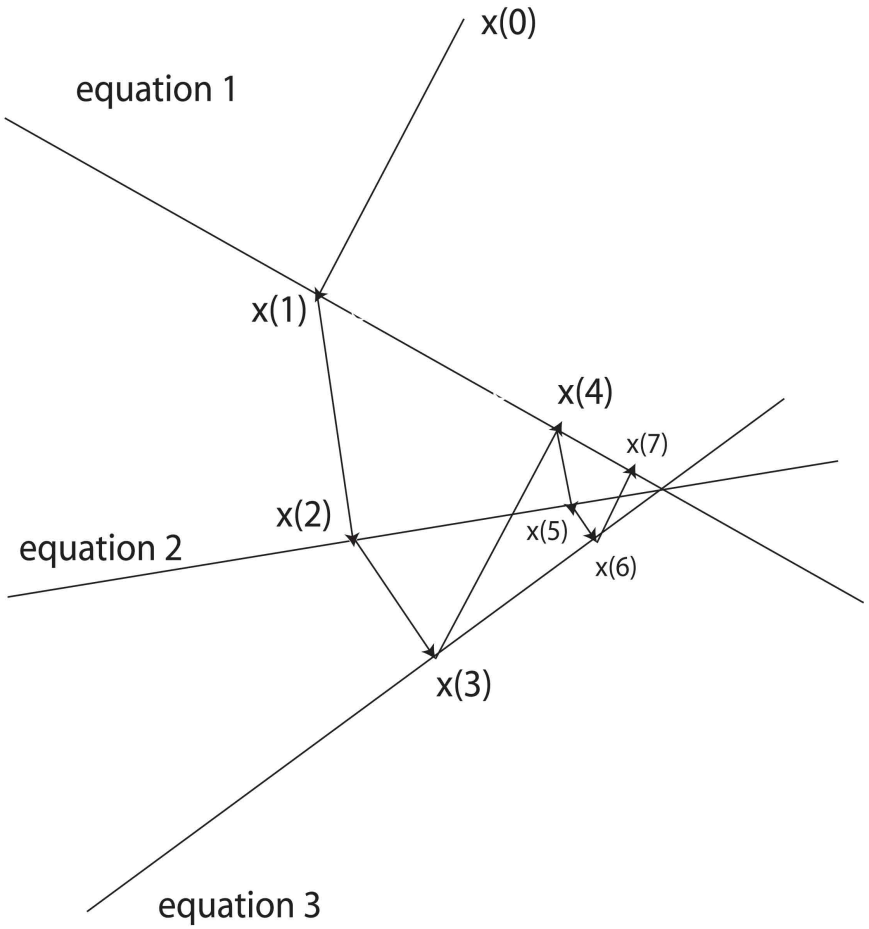

Figure 4.2: The ART algorithm in the consistent case.

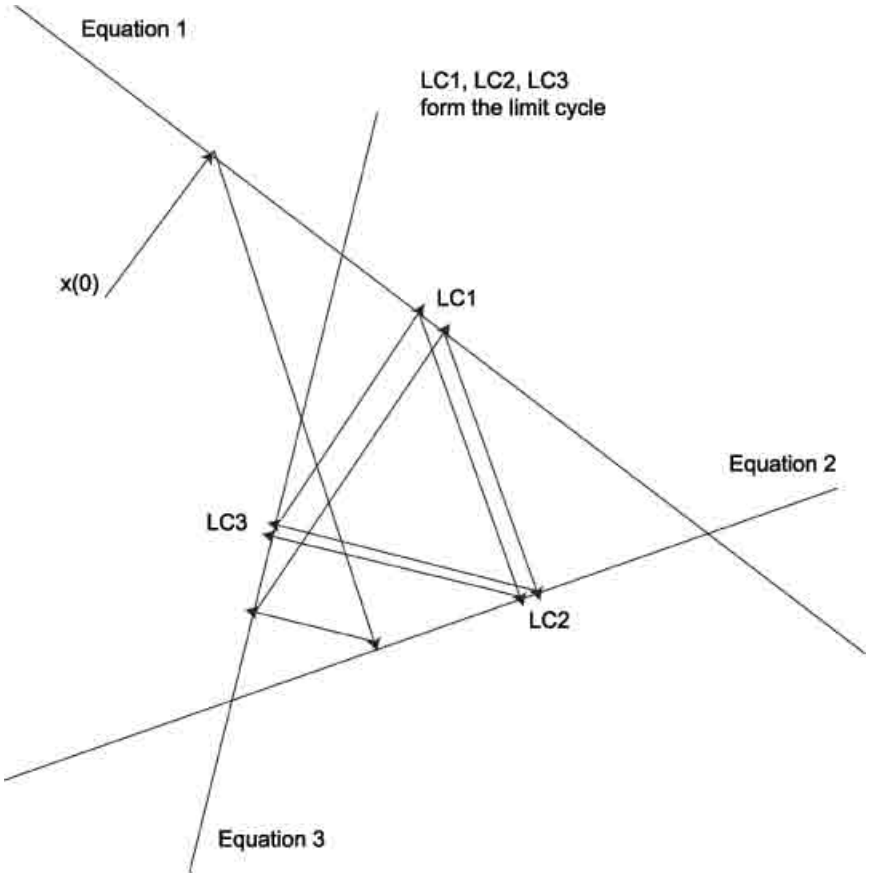

Figure 4.3: The ART algorithm in the inconsistent case.

Part II

Algebra

Chapter 5

A Little Matrix Theory

5.1 Matrix Algebra

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

$$C_{mk} = \sum_{n=1}^{N} A_{mn} B_{nk}. \tag{5.1}$$

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

$$b_m = \sum_{n=1}^{N} A_{mn} x_n. \tag{5.2}$$

Exercise 5.1 *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 $C = AB$.

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

$$(A^\dagger)_{nm} = \overline{A_{mn}} \tag{5.3}$$

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

Exercise 5.2 *Show that $B^\dagger A^\dagger = (AB)^\dagger = C^\dagger$.*

5.2 Bases and Dimension

The notions of a basis and of linear independence are fundamental in linear algebra. Let V be a vector space.

5.2.1 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 5.1 *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 $c_1, ..., c_N$, such that*

$$x = c_1 u^1 + ... + c_N u^N. \tag{5.4}$$

Definition 5.2 *A collection of vectors $\{w^1, ..., w^N\}$ in V is called a* spanning set *for a subspace S if the set S is their span.*

Definition 5.3 *A subset S of a vector space V is called* finite dimensional *if it is contained in the span of a finite set of vectors from V.*

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 5.4 *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*

$$0 = \alpha_1 u^1 + ... + \alpha_N u^N. \tag{5.5}$$

Exercise 5.3 *Show that the following are equivalent:*

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

Definition 5.5 *A collection of vectors $\mathcal{U} = \{u^1, ..., u^N\}$ in V is called a* basis *for a subspace S if the collection is linearly independent and S is their span.*

Exercise 5.4 *Show that*

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

5.2.2 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 subspace.

Suppose that S is a subspace of V, that $\{w^1, ..., w^N\}$ is a spanning set for S, and $\{u^1, ..., u^M\}$ is a linearly independent subset of S. 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 S (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 S, hence another basis for S. The following lemma will allow us to prove that all bases for a subspace S have the same number of elements.

Lemma 5.1 *Let $G = \{w^1, ..., w^N\}$ be a spanning set for a subspace S in R^I, and $H = \{v^1, ..., v^M\}$ a linearly independent subset of S. Then $M \leq N$.*

Proof: Suppose that $M > N$. Let $B_0 = G = \{w^1, ..., w^N\}$. To obtain the set B_1, form the set $C_1 = \{v_1, w_1, ..., w_N\}$ 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 v_1 has no members to its left, it is not removed. Since G is a spanning set, $v_1 \neq 0$ is a linear combination of the members of G, so that some member of G is a linear combination of v_1 and the members of G to the left of it in the list; remove the first member of G for which this is true.

We note that the set B_1 is a spanning set for S and has N members. Having obtained the spanning set B_k, with N members and whose first k members are $v_k, ..., v_1$, we form the set $C_{k+1} = B_k \cup \{v_{k+1}\}$, listing the members so that the first $k+1$ of them are $\{v_{k+1}, v_k, ..., v_1\}$. 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 v_{k+1} is a linear combination of the members of B_k. Since the set H is

linearly independent, the member removed is from the set G. Continuing in this fashion, we obtain a sequence of spanning sets $B_1, ..., B_N$, each with N members. The set B_N is $B_N = \{v_1, ..., v_N\}$ and v_{N+1} must then be a linear combination of the members of B_N, which contradicts the linear independence of H. ∎

Corollary 5.1 *Every basis for a subspace S has the same number of elements.*

Definition 5.6 *The* dimension of a subspace S *is the number of elements in any basis.*

5.3 The Geometry of Real Euclidean Space

We denote by 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 C^N the space of all N-dimensional column vectors with complex entries. For x in C^N we denote by $x^\dagger$ the N-dimensional row vector whose entries are the complex conjugates of the entries of x.

5.3.1 Dot Products

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

$$x \cdot y = \sum_{n=1}^{N} x_n \overline{y_n}. \tag{5.6}$$

Note that we can write

$$x \cdot y = y^\dagger x, \tag{5.7}$$

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

$$||x||_2 = \sqrt{x \cdot x} = \sqrt{x^\dagger x}. \tag{5.8}$$

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

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

Definition 5.7 *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 c:*

$$\langle u + w, v\rangle = \langle u, v\rangle + \langle w, v\rangle; \tag{5.9}$$

$$\langle cu, v\rangle = c\langle u, v\rangle; \tag{5.10}$$

$$\langle v, u\rangle = \overline{\langle u, v\rangle}; \tag{5.11}$$

and

$$\langle u, u\rangle \geq 0, \tag{5.12}$$

with equality in Inequality (5.12) if and only if $u = 0$.

The dot products on R^N and 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, although we shall occasionally use the matrix multiplication form, $v^\dagger u$ or the inner product notation $\langle u, v\rangle$.

Definition 5.8 *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 $m \neq n$.*

5.3.2 Cauchy's Inequality

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

$$|\langle x, y\rangle| \leq ||x||_2 ||y||_2, \tag{5.13}$$

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

$$||x + y||_2 \leq ||x||_2 + ||y||_2; \tag{5.14}$$

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:

$$||x + y||_2^2 + ||x - y||_2^2 = 2||x||_2^2 + 2||y||_2^2. \tag{5.15}$$

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

5.4 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

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

becomes

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

Exercise 5.5 (a) *Show that the complex dot product* $\mathbf{vec}(A)\cdot\mathbf{vec}(B) = \mathbf{vec}(B)^\dagger\mathbf{vec}(A)$ *can be obtained by*

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

where, for a square matrix C, $\text{trace}\,(C)$ *means the sum of the entries along the main diagonal of* C. *We can therefore use the trace to define an inner product between matrices:* $< A, B >= \text{trace}\,(AB^\dagger)$.

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

Exercise 5.6 *Let* $B = ULD^\dagger$ *be an* M *by* N *matrix in diagonalized form; that is,* L *is an* M *by* N *diagonal matrix with entries* $\lambda_1, ..., \lambda_K$ *on its main diagonal, where* $K = \min(M, N)$, *and* U *and* V *are square matrices. Let the* n*-th column of* U *be denoted* $\mathbf{u}^n$ *and similarly for the columns of* V. *Such a diagonal decomposition occurs in the singular value decomposition (SVD). Show that we can write*

$$B = \lambda_1\mathbf{u}^1(\mathbf{v}^1)^\dagger + ... + \lambda_K\mathbf{u}^K(\mathbf{v}^K)^\dagger.$$

If B is an N by N Hermitian matrix, then we can take $U = V$ and $K = M = N$, with the columns of U the eigenvectors of B, normalized to have Euclidean norm equal to one, and the λ_n to be the eigenvalues of B. In this case we may also assume that U is a *unitary* matrix; that is, $UU^\dagger = U^\dagger U = I$, where I denotes the identity matrix.

5.5 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.

5.5.1 Systems of Linear Equations

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

$$\begin{array}{rrrrrl} x_1 & +2x_2 & & +2x_4 & +x_5 & = 0 \\ -x_1 & -x_2 & +x_3 & +x_4 & & = 0\,. \\ x_1 & +2x_2 & -3x_3 & -x_4 & -2x_5 & = 0 \end{array} \tag{5.16}$$

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

$$A = \begin{bmatrix} 1 & 2 & 0 & 2 & 1 \\ -1 & -1 & 1 & 1 & 0 \\ 1 & 2 & -3 & -1 & -2 \end{bmatrix}, \tag{5.17}$$

and $x = (x_1, x_2, x_3, x_4, x_5)^T$. Applying Gaussian elimination to this system, we obtain a second, simpler, system with the same solutions:

$$\begin{array}{rrrrrl} x_1 & & & -2x_4 & +x_5 & = 0 \\ & x_2 & & +2x_4 & & = 0\,. \\ & & x_3 & +x_4 & +x_5 & = 0 \end{array} \tag{5.18}$$

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*. 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 as $A = [\,B \quad N\,]$, where B is the square invertible matrix

$$B = \begin{bmatrix} 1 & 2 & 0 \\ -1 & -1 & 1 \\ 1 & 2 & -3 \end{bmatrix}, \tag{5.19}$$

and N is the matrix

$$N = \begin{bmatrix} 2 & 1 \\ 1 & 0 \\ -1 & -2 \end{bmatrix}. \tag{5.20}$$

With $x_B = (x_1, x_2, x_3)^T$ and $x_N = (x_4, x_5)^T$ we can write

$$Ax = Bx_B + Nx_N = 0, \tag{5.21}$$

so that

$$x_B = -B^{-1}Nx_N. \tag{5.22}$$

Exercise 5.7 *Let $G = \{w^1, ..., w^N\}$ be a spanning set for a subspace S in R^I, and $H = \{v^1, ..., v^M\}$ a linearly independent subset of S. Let A be the I by M matrix whose columns are the vectors v^m and B the I by N matrix whose columns are the w^n. Prove that there is an N by M matrix C such that $A = BC$. Prove Lemma 5.1 by showing that, if $M > N$, then there is a non-zero vector x with $Cx = 0$.*

Definition 5.9 *The* dimension *of a subspace S is the number of elements in any basis.*

5.5.2 Rank of a Matrix

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

Lemma 5.2 *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 I by J matrix, and that $K \leq J$ 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 I by K matrix U. Since every column of A must be a linear combination of these K selected ones, there is a K by J matrix M such that $A = UM$. From $A^T = M^T U^T$ we conclude that every column of A^T is a linear combination of the K columns of the matrix M^T. Therefore, there can be at most K linearly independent columns of A^T. ∎

Definition 5.10 *The* rank *of A is the maximum number of linearly independent rows or of linearly independent columns of A.*

Exercise 5.8 *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.*

5.5.3 Real and Complex Systems of Linear Equations

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. For any matrix A, we denote by A^T and $A^\dagger$ the transpose and conjugate transpose of A, respectively.

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

$$\tilde{A} = \begin{bmatrix} A_1 & -A_2 \\ A_2 & A_1 \end{bmatrix}, \tag{5.23}$$

by $\tilde{x}$ the real vector

$$\tilde{x} = \begin{bmatrix} x^1 \\ x^2 \end{bmatrix}, \tag{5.24}$$

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

$$\tilde{b} = \begin{bmatrix} b^1 \\ b^2 \end{bmatrix}. \tag{5.25}$$

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

Definition 5.11 *A square matrix A is* symmetric *if $A^T = A$ and Hermitian if $A^\dagger = A$.*

Definition 5.12 *A non-zero vector x is said to be an* eigenvector *of the square matrix A if there is a scalar λ such that $Ax = \lambda x$. Then λ is said to be an* eigenvalue *of A.*

If x is an eigenvector of A with eigenvalue λ, then the matrix $A - \lambda I$ has no inverse, so its determinant is zero; here I is the identity matrix with ones on the main diagonal and zeros elsewhere. Solving for the roots of the determinant is one way to calculate the eigenvalues of A. For example, the eigenvalues of the Hermitian matrix

$$B = \begin{bmatrix} 1 & 2+i \\ 2-i & 1 \end{bmatrix} \tag{5.26}$$

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 $\tilde{B}$ has the same eigenvalues, but both with multiplicity two. Finally, the associated eigenvectors of $\tilde{B}$ are

$$\begin{bmatrix} u^1 \\ u^2 \end{bmatrix}, \tag{5.27}$$

and

$$\begin{bmatrix} -u^2 \\ u^1 \end{bmatrix}, \tag{5.28}$$

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

$$\begin{bmatrix} v^1 \\ v^2 \end{bmatrix}, \tag{5.29}$$

and

$$\begin{bmatrix} -v^2 \\ v^1 \end{bmatrix}, \tag{5.30}$$

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

5.6 Solutions of Under-determined Systems of Linear Equations

Suppose that $A\mathbf{x} = \mathbf{b}$ is a consistent linear system of M equations in N unknowns, where $M < N$. Then there are infinitely many solutions. A standard procedure in such cases is to find that solution $\mathbf{x}$ having the smallest norm

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

As we shall see shortly, the *minimum norm* solution of $A\mathbf{x} = \mathbf{b}$ is a vector of the form $\mathbf{x} = A^\dagger \mathbf{z}$, where $A^\dagger$ denotes the conjugate transpose of the matrix A. Then $A\mathbf{x} = \mathbf{b}$ becomes $AA^\dagger \mathbf{z} = \mathbf{b}$. Typically, $(AA^\dagger)^{-1}$ will exist, and we get $\mathbf{z} = (AA^\dagger)^{-1}\mathbf{b}$, from which it follows that the minimum norm solution is $\mathbf{x} = A^\dagger(AA^\dagger)^{-1}\mathbf{b}$. When M and N are not too large, forming the matrix $AA^\dagger$ and solving for $\mathbf{z}$ is not prohibitively expensive and time-consuming. However, in image processing the vector $\mathbf{x}$ 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 the minimum norm solution without computing $AA^\dagger$.

We begin by describing the minimum-norm solution of a consistent system $Ax = b$.

Theorem 5.1 *The minimum norm solution of $A\mathbf{x} = \mathbf{b}$ has the form $\mathbf{x} = A^\dagger \mathbf{z}$ for some M-dimensional complex vector $\mathbf{z}$.*

Proof: Let the *null space* of the matrix A be all N-dimensional complex vectors $\mathbf{w}$ with $A\mathbf{w} = \mathbf{0}$. If $A\mathbf{x} = \mathbf{b}$ then $A(\mathbf{x} + \mathbf{w}) = \mathbf{b}$ for all $\mathbf{w}$ in the null space of A. If $\mathbf{x} = A^\dagger \mathbf{z}$ and $\mathbf{w}$ is in the null space of A, then

$$||\mathbf{x} + \mathbf{w}||^2 = ||A^\dagger \mathbf{z} + \mathbf{w}||^2 = (A^\dagger \mathbf{z} + \mathbf{w})^\dagger (A^\dagger \mathbf{z} + \mathbf{w})$$

$$= (A^\dagger \mathbf{z})^\dagger (A^\dagger \mathbf{z}) + (A^\dagger \mathbf{z})^\dagger \mathbf{w} + \mathbf{w}^\dagger (A^\dagger \mathbf{z}) + \mathbf{w}^\dagger \mathbf{w}$$

$$= ||A^\dagger \mathbf{z}||^2 + (A^\dagger \mathbf{z})^\dagger \mathbf{w} + \mathbf{w}^\dagger (A^\dagger \mathbf{z}) + ||\mathbf{w}||^2$$

$$= ||A^\dagger \mathbf{z}||^2 + ||\mathbf{w}||^2,$$

since

$$\mathbf{w}^\dagger (A^\dagger \mathbf{z}) = (A\mathbf{w})^\dagger \mathbf{z} = \mathbf{0}^\dagger \mathbf{z} = 0$$

and

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

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

Exercise 5.9 *Show that if $\mathbf{z} = (z_1, ..., z_N)^T$ 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 $\mathbf{z}^\dagger H\mathbf{z}$ is a real number. Show that the quadratic form $\mathbf{z}^\dagger H\mathbf{z}$ can be calculated using only real numbers. Let $\mathbf{z} = \mathbf{x} + i\mathbf{y}$, with $\mathbf{x}$ and $\mathbf{y}$ real vectors and let $H = A + iB$, where A and B are real matrices. Then show that $A^T = A$, $B^T = -B$, $\mathbf{x}^T B\mathbf{x} = 0$ and finally,*

$$\mathbf{z}^\dagger H\mathbf{z} = [\,\mathbf{x}^T \quad \mathbf{y}^T\,] \begin{bmatrix} A & -B \\ B & A \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}.$$

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

5.6.1 Matrix Inverses

A square matrix A is said to have inverse A^{-1} provided that

$$AA^{-1} = A^{-1}A = I,$$

where I is the identity matrix. The 2 by 2 matrix $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ has an inverse

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

whenever the *determinant* of A, $\det(A) = ad - bc$ 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 is found using the singular-value decomposition.

5.6.2 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

$$B_{\text{new}} = B_{\text{old}} - uv^T, \tag{5.31}$$

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:

$$(B - uv^T)^{-1} = B^{-1} + \alpha(B^{-1}u)(v^T B^{-1}), \tag{5.32}$$

where

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

5.7 LU Factorization

The matrix

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & 1 & 0 \\ -2 & 2 & 1 \end{bmatrix}$$

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

$$L_3 L_2 L_1 A = U,$$

where L_1, L_2, and L_3 are the matrix representatives of the three elementary row operations. Therefore, we have

$$A = L_1^{-1} L_2^{-1} L_3^{-1} U = LU.$$

This is the *LU factorization* of A. As we just saw, the LU factorization can be obtained along with the Gauss elimination.

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 $A = LDU$, where D ia a diagonal matrix.

Note that it may not be possible to obtain $A = LDU$ without first permuting the rows of A; in such cases we obtain $PA = LDU$, where P is obtained from the identity matrix by permuting rows.

Suppose that we have to solve the system of linear equations $Ax = b$. 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 A^{-1}; when A 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 A^{-1} does not.

If A is real and symmetric, and if $A = LDU$, then $U = L^T$, so we have $A = LDL^T$. If, in addition, the non-zero entries of D are positive, then we can write

$$A = (L\sqrt{D})(L\sqrt{D})^T,$$

which is the Cholesky Decomposition of A.

Exercise 5.10 *Show that the symmetric matrix*

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

cannot be written as $H = LDL^T$.

5.8 Eigenvalues and Eigenvectors

Given N by N complex matrix A, we say that a complex number λ is an *eigenvalue* of A if there is a nonzero vector $\mathbf{u}$ with $A\mathbf{u} = \lambda\mathbf{u}$. The column vector $\mathbf{u}$ is then called an *eigenvector* of A associated with eigenvalue λ; clearly, if $\mathbf{u}$ is an eigenvector of A, then so is $c\mathbf{u}$, for any constant $c \neq 0$. If λ is an eigenvalue of A, then the matrix $A - \lambda I$ fails to have an inverse, since $(A - \lambda I)\mathbf{u} = \mathbf{0}$ but $\mathbf{u} \neq \mathbf{0}$. If we treat λ as a variable and compute the determinant of $A - \lambda I$, we obtain a polynomial of degree N in λ. Its roots $\lambda_1, ..., \lambda_N$ are then the eigenvalues of A. If $||\mathbf{u}||^2 = \mathbf{u}^\dagger\mathbf{u} = 1$ then $\mathbf{u}^\dagger A\mathbf{u} = \lambda\mathbf{u}^\dagger\mathbf{u} = \lambda$.

It can be shown that it is possible to find a set of N mutually orthogonal norm-one eigenvectors of the Hermitian matrix H; call them $\{\mathbf{u}^1, ..., \mathbf{u}^N\}$. The matrix H can then be written as

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

a linear superposition of the *dyad* matrices $\mathbf{u}^n(\mathbf{u}^n)^\dagger$. We can also write $H = ULU^\dagger$, where U is the matrix whose nth column is the column vector $\mathbf{u}^n$ and L is the diagonal matrix with the eigenvalues down the main diagonal and zero elsewhere. 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 is invertible if and only if none of the λ are zero and its inverse is

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

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

A Hermitian matrix Q is said to be nonnegative-definite (positive-definite) if all the eigenvalues of Q are nonnegative (positive). The matrix Q is a nonnegative-definite matrix if and only if there is another matrix C such that $Q = C^\dagger C$. Since the eigenvalues of Q are nonnegative, the

diagonal matrix L has a square root, $\sqrt{L}$. 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 $C^\dagger = C$. Then $\mathbf{z}^\dagger Q\mathbf{z} = \mathbf{z}^\dagger C^\dagger C\mathbf{z} = ||C\mathbf{z}||^2$, so that Q is positive-definite if and only if C is invertible. The matrix C is called the *Hermitian square-root* of Q.

Exercise 5.11 *Let A be an M by N matrix with complex entries. View A as a linear function with domain C^N, the space of all N-dimensional complex column vectors, and range contained within C^M, via the expression $A(\mathbf{x}) = A\mathbf{x}$. Suppose that $M > N$. The range of A, denoted $R(A)$, cannot be all of C^M. Show that every vector $\mathbf{z}$ in C^M can be written uniquely in the form $\mathbf{z} = A\mathbf{x} + \mathbf{w}$, where $A^\dagger\mathbf{w} = \mathbf{0}$. Show that $\|\mathbf{z}\|^2 = \|A\mathbf{x}\|^2 + \|\mathbf{w}\|^2$, where $\|\mathbf{z}\|^2$ denotes the square of the norm of $\mathbf{z}$.*

Hint: If $\mathbf{z} = A\mathbf{x} + \mathbf{w}$ then consider $A^\dagger\mathbf{z}$. Assume $A^\dagger A$ is invertible.

5.9 The Singular Value Decomposition (SVD)

The year 1965 was a good one for the discovery of important algorithms. In that year, Cooley and Tukey [93] introduced the *fast Fourier transform* (FFT) and Golub and Kahan [136] 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

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

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

Let C be any N by K complex matrix. In presenting the SVD of C we shall assume that $K \geq N$; the SVD of $C^\dagger$ will come from that of C. Let $A = C^\dagger C$ and $B = CC^\dagger$; we assume, reasonably, that B, the smaller of the two matrices, is invertible, so all the eigenvalues $\lambda_1, ..., \lambda_N$ of B are positive. Then, write the eigenvalue/eigenvector decomposition of B as $B = ULU^\dagger$.

Exercise 5.12 *Show that the nonzero eigenvalues of $A = C^\dagger C$ and $B = CC^\dagger$ are the same.*

Let V be the K by K matrix whose first N columns are those of the matrix $C^\dagger U L^{-1/2}$ and whose remaining $K-N$ columns are any mutually orthogonal norm-one vectors that are all orthogonal to each of the first N columns. Let M be the N by K matrix with diagonal entries $M_{nn} = \sqrt{\lambda_n}$ for $n = 1, ..., N$ and whose remaining entries are zero. The nonzero entries of M, $\sqrt{\lambda_n}$, are called the *singular values* of C. The *singular value decomposition* (SVD) of C is $C = UMV^\dagger$. The SVD of $C^\dagger$ is $C^\dagger = VM^TU^\dagger$.

Exercise 5.13 *Show that $UMV^\dagger$ equals C.*

Using the SVD of C we can write

$$C = \sum_{n=1}^{N} \sqrt{\lambda_n} \mathbf{u}^n (\mathbf{v}^n)^\dagger,$$

where $\mathbf{v}^n$ denotes the nth column of the matrix V.

In image processing, matrices such as C are used to represent discrete two-dimensional images, with the entries of C corresponding to the grey level or color at each pixel. It is common to find that most of the N singular values of C are nearly zero, so that C can be written approximately as a sum of far fewer than N dyads; this is SVD image compression.

We have obtained the SVD of C using the eigenvectors and eigenvalues of the Hermitian matrices $A = C^\dagger C$ and $B = CC^\dagger$; for large matrices, this is not an efficient way to get the SVD. The Golub-Kahan algorithm [136] calculates the SVD of C without forming the matrices A and B.

If $N \neq K$ then C cannot have an inverse; it does, however, have a *pseudo-inverse*, $C^* = VM^*U^\dagger$, where M^* is the matrix obtained from M by taking the inverse of each of its nonzero entries and leaving the remaining zeros the same. The pseudo-inverse of $C^\dagger$ is

$$(C^\dagger)^* = (C^*)^\dagger = U(M^*)^T V^\dagger = U(M^\dagger)^* V^\dagger.$$

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

1. $CC^*C = C$,
2. $C^*CC^* = C^*$,
3. $(C^*C)^\dagger = C^*C$,
4. $(CC^*)^\dagger = CC^*$.

The pseudo-inverse of an arbitrary I by J matrix G can be used in much the same way as the inverse of nonsingular matrices to find approximate or

exact solutions of systems of equations $G\mathbf{x} = \mathbf{d}$. The following examples illustrate this point.

Exercise 5.14 *If $I > J$ the system $G\mathbf{x} = \mathbf{d}$ probably has no exact solution. Show that whenever $G^\dagger G$ is invertible the pseudo-inverse of G is $G^* = (G^\dagger G)^{-1} G^\dagger$ so that the vector $\mathbf{x} = G^*\mathbf{d}$ is the least squares approximate solution.*

Exercise 5.15 *If $I < J$ the system $G\mathbf{x} = \mathbf{d}$ probably has infinitely many solutions. Show that whenever the matrix $GG^\dagger$ is invertible the pseudo-inverse of G is $G^* = G^\dagger (GG^\dagger)^{-1}$, so that the vector $\mathbf{x} = G^*\mathbf{d}$ is the exact solution of $G\mathbf{x} = \mathbf{d}$ closest to the origin; that is, it is the minimum norm solution.*

5.10 Principle-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 *principle-component analysis* (PCA).

5.10.1 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 $CS(D)$, the column space of D in the space C^M, is probably of

dimension N, it may well be the case that $CS(D)$ is nearly spanned by a much smaller set of its members. The goal of principle-component analysis is to find such a smaller set.

5.10.2 Decomposing $D^\dagger D$

The matrix $B = D^\dagger D$ is Hermitian and non-negative 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:

$$D^\dagger D \approx \sum_{j=1}^{J} \lambda_j u^j (u^j)^\dagger. \tag{5.33}$$

5.10.3 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:

$$D \approx \sum_{j=1}^{J} \sqrt{\lambda_j} w^j (u^j)^\dagger, \tag{5.34}$$

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 the span of the w^j is nearly all of $CS(D)$. The w^j are the *principle-component vectors*.

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

$$C = UMV^\dagger,$$

so that the SVD of the matrix D is

$$D = VM^\dagger U^\dagger = \sum_{j=1}^{N} \sqrt{\lambda_j} v^j (u^j)^\dagger.$$

The first J columns of the matrix V are the w^j defined above, so the Golub-Kahan SVD algorithm [136] can then be used to obtain the principle-component vectors of the data matrix D.

5.11 The 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 [33] as well as in the social sciences.

In [230] 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.

5.12 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.

In [59] 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 [180]. This result suggested the following proposition, whose proof was given in [59].

Proposition 5.1 *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 $A^TA\mathbf{v} = c\mathbf{v}$ for some nonzero vector $\mathbf{v}$. We show that $c \leq s$. We have $AA^TA\mathbf{v} = cA\mathbf{v}$ and so $\mathbf{w}^TAA^T\mathbf{w} = \mathbf{v}^TA^TAA^TA\mathbf{v} = c\mathbf{v}^TA^TA\mathbf{v} = c\mathbf{w}^T\mathbf{w}$, for $\mathbf{w} = A\mathbf{v}$. Then, with $e_{mn} = 1$ if $A_{mn} \neq 0$ and $e_{mn} = 0$ otherwise, we have

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

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

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

Therefore,

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

and

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

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

The result follows immediately. ∎

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^TA. Consequently, the maximum eigenvalue of A^TA 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 15.1.

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^TA, 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 5.1.

Chapter 6

Metric Spaces and Norms

The inner product on R^J or C^J 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. Throughout this chapter the superscript † denotes the conjugate transpose of a matrix or vector.

6.1 Metric Spaces

We begin with the basic definitions.

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

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

for all s and t in $\mathcal{S}$;

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

if and only if $s = t$;

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

for all s and t in $\mathcal{S}$; and, for all s, t, and u in $\mathcal{S}$,

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

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

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

6.2 Analysis in Metric Space

Analysis is concerned with issues of convergence and limits.

Definition 6.2 *A sequence* $\{s^k\}$ *in the metric space* $(\mathcal{S}, d)$ *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.

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*,*

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

Every convergent sequence is a Cauchy sequence.

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

The finite-dimensional spaces $R^J$and C^J are complete metric spaces, with respect to the usual Euclidean distance.

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

Definition 6.6 *A subset* K *of the metric space is said to be* closed *if, for every convergent sequence* $\{s^k\}$ *of elements in* K*, the limit point is again in* K*. The* closure *of a set* K *is the smallest closed set containing* K*.*

For example, in $R^J = R$, the set $K = (0, 1]$ 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 $K = [0, 1]$ is closed and is the *closure* of the set $(0, 1]$, that is, it is the smallest closed set containing $(0, 1]$.

Definition 6.7 *For any bounded sequence* $\{x^k\}$ *in* 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.*

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. We turn now to metrics that come from norms.

6.3 Norms

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

Definition 6.8 *We say that* $\|\cdot\|$ *is a* norm *on* C^J *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* C^J*.*

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

6.3.1 Some Common Norms on C^J

We consider now the most common norms on the space C^J. These notions apply equally to R^J.

The 1-norm

The 1-norm on C^J is defined by

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

The ∞-norm

The ∞-norm on C^J is defined by

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

The 2-norm

The 2-norm, also called the Euclidean norm, is the most commonly used norm on C^J. It is the one that comes from the inner product:

$$\|x\|_2 = \sqrt{\langle x, x\rangle} = \sqrt{x^\dagger x}. \tag{6.13}$$

Weighted 2-norms

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

$$\|x\|_Q = \|Ax\|_2 = \sqrt{x^\dagger Q x}, \tag{6.14}$$

for all vectors x. If Q is the diagonal matrix with diagonal entries $Q_{jj} > 0$, then

$$\|x\|_Q = \sqrt{\sum_{j=1}^{J} Q_{jj}|x_j|^2}; \tag{6.15}$$

for that reason we speak of $\|x\|_Q$ as the Q-*weighted* 2-norm of x.

6.4 Matrix Norms

Any matrix can be turned into a vector by vectorization. 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.

6.4.1 Induced Matrix Norms

One way to obtain a compatible norm for matrices is through the use of an induced matrix norm.

Definition 6.9 *Let $\|x\|$ be any norm on C^J, not necessarily the Euclidean norm, $\|b\|$ any norm on C^I, and A a rectangular I by J matrix. The* induced matrix norm *of A, simply denoted $\|A\|$, derived from these two vectors norms, is the smallest positive constant c such that*

$$\|Ax\| \leq c\|x\|, \tag{6.16}$$

for all x in C^J. This induced norm can be written as

$$\|A\| = \max_{x\neq 0}\{\|Ax\|/\|x\|\}. \tag{6.17}$$

We study induced matrix norms in order to measure the distance $\|Ax - Az\|$, relative to the distance $\|x - z\|$:

$$\|Ax - Az\| \leq \|A\| \, \|x - z\|, \tag{6.18}$$

for all vectors x and z and $\|A\|$ is the smallest number for which this statement can be made.

6.4.2 Condition Number of a Square Matrix

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

$$\|\delta_z\| \leq \|S^{-1}\|\|\delta_h\|, \tag{6.19}$$

and

$$\|z\| \geq \|h\|/\|S\|. \tag{6.20}$$

Therefore

$$\frac{\|\delta_z\|}{\|z\|} \leq \|S\| \, \|S^{-1}\| \frac{\|\delta_h\|}{\|h\|}. \tag{6.21}$$

Definition 6.10 *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

$$\|S^{-1}\| \geq \|S^{-1}z\|/\|z\| = \|S^{-1}z\|/\|SS^{-1}z\| \geq 1/\|S\|. \tag{6.22}$$

When S is Hermitian and positive-definite, the condition number of S, with respect to the matrix norm induced by the Euclidean vector norm, is

$$c = \lambda_{max}(S)/\lambda_{min}(S), \tag{6.23}$$

the ratio of the largest to the smallest eigenvalues of S.

6.4.3 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

$$\|x\|_1 = \sum_{j=1}^{J} |x_j|. \tag{6.24}$$

Lemma 6.2 *The 1-norm of A, induced by the 1-norms of vectors in C^J and C^I, is*

$$\|A\|_1 = \max\{\sum_{i=1}^{I} |A_{ij}|, j = 1, 2, ..., J\}. \tag{6.25}$$

Proof: Use basic properties of the absolute value to show that

$$\|Ax\|_1 \leq \sum_{j=1}^{J}(\sum_{i=1}^{I} |A_{ij}|)|x_j|. \tag{6.26}$$

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

$$\|x\|_\infty = \max\{|x_j|, j = 1, 2, ..., J\}. \tag{6.27}$$

Lemma 6.3 *The infinity norm of the matrix A, induced by the infinity norms of vectors in R^J and C^I, is*

$$\|A\|_\infty = \max\{\sum_{j=1}^{J} |A_{ij}|, i = 1, 2, ..., I\}. \tag{6.28}$$

The proof is similar to that of the previous lemma.

Lemma 6.4 *Let M be an invertible matrix and $\|x\|$ any vector norm. Define*

$$\|x\|_M = \|Mx\|. \tag{6.29}$$

Then, for any square matrix S, the matrix norm

$$\|S\|_M = \max_{x \neq 0}\{\|Sx\|_M / \|x\|_M\} \tag{6.30}$$

is

$$\|S\|_M = \|MSM^{-1}\|. \tag{6.31}$$

In [7] this result is used to prove the following lemma:

Lemma 6.5 *Let S be any square matrix and let $\epsilon > 0$ be given. Then there is an invertible matrix M such that*

$$\|S\|_M \leq \rho(S) + \epsilon. \tag{6.32}$$

6.4.4 The Euclidean Norm of a Square Matrix

We shall be particularly interested in the Euclidean norm (or 2-norm) of the square matrix A, denoted by $\|A\|_2$, which is the induced matrix norm derived from the Euclidean vector norms.

From the definition of the Euclidean norm of A, we know that

$$\|A\|_2 = \max\{\|Ax\|_2/\|x\|_2\}, \tag{6.33}$$

with the maximum over all nonzero vectors x. Since

$$\|Ax\|_2^2 = x^\dagger A^\dagger A x, \tag{6.34}$$

we have

$$\|A\|_2 = \sqrt{\max\{\frac{x^\dagger A^\dagger A x}{x^\dagger x}\}}, \tag{6.35}$$

over all nonzero vectors x.

Proposition 6.1 *The Euclidean norm of a square matrix is*

$$\|A\|_2 = \sqrt{\rho(A^\dagger A)}; \tag{6.36}$$

that is, the term inside the square-root in Equation (6.35) is the largest eigenvalue of the matrix $A^\dagger A$.

Proof: Let

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_J \geq 0 \tag{6.37}$$

and let $\{u^j,\, j = 1, ..., J\}$ be mutually orthogonal eigenvectors of $A^\dagger A$ with $\|u^j\|_2 = 1$. Then, for any x, we have

$$x = \sum_{j=1}^{J} [(u^j)^\dagger x] u^j, \tag{6.38}$$

while

$$A^\dagger A x = \sum_{j=1}^{J} [(u^j)^\dagger x] A^\dagger A u^j = \sum_{j=1}^{J} \lambda_j [(u^j)^\dagger x] u^j. \tag{6.39}$$

It follows that

$$\|x\|_2^2 = x^\dagger x = \sum_{j=1}^{J} |(u^j)^\dagger x|^2, \tag{6.40}$$

and

$$\|Ax\|_2^2 = x^\dagger A^\dagger Ax = \sum_{j=1}^{J} \lambda_j |(u^j)^\dagger x|^2. \tag{6.41}$$

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 (6.41) is then a convex combination of the λ_j, which will have its maximum when only the coefficient of λ_1 is non-zero. ∎

According to Corollary 15.1, we have the inequality

$$\|A\|_2^2 \leq \|A\|_1 \|A\|_\infty = c_1 r_1.$$

If S is not Hermitian, then the Euclidean norm of S cannot be calculated directly from the eigenvalues of S. Take, for example, the square, non-Hermitian matrix

$$S = \begin{bmatrix} i & 2 \\ 0 & i \end{bmatrix}, \tag{6.42}$$

having eigenvalues $\lambda = i$ and $\lambda = i$. The eigenvalues of the Hermitian matrix

$$S^\dagger S = \begin{bmatrix} 1 & -2i \\ 2i & 5 \end{bmatrix} \tag{6.43}$$

are $\lambda = 3 + 2\sqrt{2}$ and $\lambda = 3 - 2\sqrt{2}$. Therefore, the Euclidean norm of S is

$$\|S\|_2 = \sqrt{3 + 2\sqrt{2}}. \tag{6.44}$$

6.4.5 Diagonalizable Matrices

Definition 6.11 *A square matrix S is* diagonalizable *if C^J has a basis of eigenvectors of S.*

In the case in which S is diagonalizable, with V be a square matrix whose columns are linearly independent eigenvectors of S and L the diagonal matrix having the eigenvalues of S along its main diagonal, we have $SV = VL$, or $V^{-1}SV = L$. Let $T = V^{-1}$ and define $\|x\|_T = \|Tx\|_2$, the Euclidean norm of Tx. Then the induced matrix norm of S is $\|S\|_T = \rho(S)$. We see from this 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 we know that, if the eigenvector columns of V are scaled to have length one, then $V^{-1} = V^\dagger$ and $\|Tx\|_2 = \|V^\dagger x\|_2 = \|x\|_2$, so that the required vector norm is just the Euclidean norm, and $\|S\|_T$ is just $\|S\|_2$, which we know to be $\rho(S)$.

6.4.6 Gerschgorin's Theorem

Gerschgorin's theorem gives us a way to estimate the eigenvalues of an arbitrary square matrix A.

Theorem 6.1 *Let A be J by J. For $j = 1, ..., J$, let C_j be the circle in the complex plane with center A_{jj} and radius $r_j = \sum_{m \neq j} |A_{jm}|$. Then every eigenvalue of A lies within one of the C_j.*

Proof: Let λ be an eigenvalue of A, with associated eigenvector u. Let u_j be the entry of the vector u having the largest absolute value. From $Au = \lambda u$, we have

$$(\lambda - A_{jj})u_j = \sum_{m \neq j} A_{jm} u_m, \tag{6.45}$$

so that

$$|\lambda - A_{jj}| \leq \sum_{m \neq j} |A_{jm}||u_m|/|u_j| \leq r_j. \tag{6.46}$$

This completes the proof. ∎

6.4.7 Strictly Diagonally Dominant Matrices

Definition 6.12 *A square I by I matrix S is said to be* strictly diagonally dominant *if, for each $i = 1, ..., I$,*

$$|S_{ii}| > r_i = \sum_{m \neq i} |S_{im}|. \tag{6.47}$$

When the matrix S is strictly diagonally dominant, all the eigenvalues of S lie within the union of the spheres with centers S_{ii} and radii S_{ii}. 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 discussing the Jacobi splitting method [63].

6.5 Exercises

Exercise 6.1 *Show that every convergent sequence is a Cauchy sequence.*

Exercise 6.2 *Let $\mathcal{S}$ be the set of rational numbers, with $d(s, t) = |s - t|$. Show that $(\mathcal{S}, d)$ is a metric space, but not a complete metric space.*

Exercise 6.3 *Show that any convergent sequence in a metric space is bounded. Find a bounded sequence of real numbers that is not convergent.*

Exercise 6.4 *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.*

Exercise 6.5 *Show that your bounded, but not convergent, sequence found in Exercise 6.3 has a cluster point.*

Exercise 6.6 *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.*

Exercise 6.7 *Show that the 1-norm is a norm.*

Exercise 6.8 *Show that the ∞-norm is a norm.*

Exercise 6.9 *Show that the 2-norm is a norm. Hint: for the triangle inequality, use the Cauchy Inequality.*

Exercise 6.10 *Show that the Q-weighted 2-norm is a norm.*

Exercise 6.11 *Show that $\rho(S^2) = \rho(S)^2$.*

Exercise 6.12 *Show that, if S is Hermitian, then every eigenvalue of S is real. Hint: suppose that $Sx = \lambda x$. Then consider $x^\dagger Sx$.*

Exercise 6.13 *Use the SVD of A to obtain the eigenvalue/eigenvector decompositions of B and C:*

$$B = \sum_{i=1}^{N} \lambda_i u^i (u^i)^\dagger, \tag{6.48}$$

and

$$C = \sum_{i=1}^{N} \lambda_i v^i (v^i)^\dagger. \tag{6.49}$$

Exercise 6.14 *Show that, for any square matrix S and any induced matrix norm $\|S\|$, we have $\|S\| \geq \rho(S)$. Consequently, for any induced matrix norm $\|S\|$,*

$$\|S\| \geq |\lambda|, \tag{6.50}$$

for every eigenvalue λ of S. So we know that

$$\rho(S) \leq \|S\|, \tag{6.51}$$

for every induced matrix norm, but, according to Lemma 6.5, we also have

$$\|S\|_M \leq \rho(S) + \epsilon. \tag{6.52}$$

Exercise 6.15 *Show that, if $\rho(S) < 1$, then there is a vector norm on C^J for which the induced matrix norm of S is less than one.*

Exercise 6.16 *Show that, if S is Hermitian, then $\|S\|_2 = \rho(S)$. Hint: use Exercise (6.11).*

Chapter 7

Linear Algebra

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.

7.1 Representing a Linear Transformation

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 C^N of N-dimensional column vectors with complex entries. Any vector v in V can be written as

$$v = \sum_{n=1}^{N} \gamma_n a^n. \tag{7.1}$$

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

$$Ta^n = A_{1n}b^1 + A_{2n}b^2 + ... + A_{Mn}b^M. \tag{7.2}$$

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 $A = [T]^{\mathcal{B}}_{\mathcal{A}}$.

Exercise 7.1 *Show that* $[Tv]_{\mathcal{B}} = A[v]_{\mathcal{A}}$.

Exercise 7.2 *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 $\mathcal{C}$.*

7.2 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 $A = [T]_{\mathcal{A}}$.

Exercise 7.3 *Suppose that $\tilde{\mathcal{A}}$ is a second basis for V and $\tilde{A} = [T]_{\tilde{\mathcal{A}}}$. Show that there is a unique invertible N by N matrix Q having the property that 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, $Q = [I]_{\mathcal{A}}^{\tilde{\mathcal{A}}}$.

7.3 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

$$A = Q^{-1}\tilde{A}Q. \tag{7.3}$$

We describe the relationship in Equation (7.3) by saying that the matrices A and $\tilde{A}$ are *similar*.

Let S be a linear transformation from V to W. Then we have

$$[S]_{\mathcal{A}}^{\mathcal{B}} = R^{-1}[S]_{\tilde{\mathcal{A}}}^{\tilde{\mathcal{B}}}Q. \tag{7.4}$$

With $G = [S]_{\mathcal{A}}^{\mathcal{B}}$ and $\tilde{G} = [S]_{\tilde{\mathcal{A}}}^{\tilde{\mathcal{B}}}$, we have

$$G = R^{-1}\tilde{G}Q. \tag{7.5}$$

Definition 7.1 *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 (7.5) by saying that the matrices G and $\tilde{G}$ are equivalent.

Exercise 7.4 *Show that A and B are equivalent if B can be obtained from A by means of elementary row and column operations.*

Exercise 7.5 *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: show that A and AQ have the same rank.*

Exercise 7.6 *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 $V = C^N$ to $W = C^M$. Therefore, A represents T in the usual bases of C^N and C^M. Now construct a basis $\mathcal{A}$ for 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 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.

7.4 Linear Functionals and Duality

When the second vector space W is just the space C of complex numbers, any linear transformation from V to W 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, $(V^*)^* = V^{**}$.

Exercise 7.7 *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^*.*

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^{**}.

Exercise 7.8 *Show that the subspace J_V has the same dimension as V^{**} itself, so that it must be all of 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.

7.5 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 $P = [I]^{\mathcal{A}}_{\mathcal{B}}$ 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.

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.

7.6 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.

Exercise 7.9 *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.

7.7 An Inner Product on V

For any two column vectors $x = (x_1, ..., x_N)^T$ and $y = (y_1, ..., y_n)^T$ in 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

$$\langle v, w \rangle = [v]_\mathcal{A} \cdot [w]_\mathcal{A}, \tag{7.6}$$

where the right side is the ordinary complex dot product in C^N. Note that, with respect to this inner product, the basis $\mathcal{A}$ becomes an orthonormal basis.

7.8 Representing Linear Functionals

Let $f : V \to C$ be a linear functional on the inner-product space V and let $\mathcal{A} = \{a^1, ..., a^N\}$ be the basis for V used to define the inner product, as in Equation (7.6). The singleton set $\{1\}$ is a basis for the space $W = C$, and the matrix A that represents $T = f$ is a 1 by N matrix, or row vector, $A = A_f$ with entries $f(a^n)$. Therefore, for each

$$v = \sum_{n=1}^{N} \alpha_n a^n,$$

in V, we have

$$f(v) = A_f [v]_\mathcal{A} = \sum_{n=1}^{N} f(a^n) \alpha_n.$$

Consequently, we can write

$$f(v) = \langle v, y_f \rangle,$$

for the vector y_f with $A_f = [y_f]_\mathcal{A}^\dagger$, or

$$y_f = \sum_{n=1}^{N} \overline{f(a^n)} a^n.$$

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 y_f in V, so that

$$f(v) = \langle v, y_f \rangle.$$

Exercise 7.10 *Show that the vector y_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 $y = w$.

7.9 The Adjoint of a Linear Transformation

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

$$(T^* g)(v) = g(Tv), \tag{7.7}$$

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 y_f in V such that

$$f(v) = \langle v, y_f \rangle.$$

Therefore,

$$\langle Tv, w \rangle = \langle v, y_f \rangle,$$

for each v in V. The *adjoint* of T is the linear transformation T^* from W to V defined by $T^* w = y_f$.

When $W = V$, and T is a linear operator on V, then so is T^*. In this case, we can ask whether or not $T^* T = T T^*$, that is, whether or not T is *normal*, and whether or not $T = T^*$, that is, whether or not T is *self-adjoint*.

7.10 Quadratic Forms and Conjugate Matrices

7.10.1 Sesquilinear Forms

A *sesquilinear functional* $\phi(x, y)$ of two vector variables 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*.

7.10.2 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

$$\hat{\phi}_1(x) = \phi(Px, Px).$$

7.10.3 Conjugate Matrices

Let A be a linear operator on an inner product space V. Then A can be used to define a sesquilinear functional $\phi(x, y)$ according to

$$\phi_A(x, y) = \langle Ax, y\rangle. \tag{7.8}$$

Then, for this sesquilinear functional, we have

$$\hat{\phi}_1(x) = \phi_A(Px, Px) = \langle APx, Px\rangle = \langle P^*APx, x\rangle.$$

We say that a square matrix B is *conjugate* to A if there is an invertible P with $B = P^*AP$.

7.10.4 Does ϕ_A Determine A?

Is it possible for

$$\langle Ax, x\rangle = \langle Bx, x\rangle,$$

for all x in the inner product space V, and yet have $A \neq B$? As we shall see, the answer is "No". First, we answer a simpler question. Is it possible for

$$\langle Ax, y\rangle = \langle Bx, y\rangle,$$

for all x and y, with $A \neq B$? The answer is "No".

Exercise 7.11 *Show that*

$$\langle Ax, y\rangle = \langle Bx, y\rangle,$$

for all x and y, implies that $A = B$.

We can use the result of the exercise to answer our first question, but first, we need the *polarization identity*.

Exercise 7.12 *Establish the* polarization identity*:*

$$\langle Ax, y\rangle = \frac{1}{4}\langle A(x+y), x+y\rangle - \frac{1}{4}\langle A(x-y), x-y\rangle$$

$$+\frac{i}{4}\langle A(x+iy), x+iy\rangle - \frac{i}{4}\langle A(x-iy), x-iy\rangle.$$

Exercise 7.13 *Show that the answer to our first question is "No"; the quadratic form determines the matrix.*

7.10.5 A New Sesquilinear Functional

Given a 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 this sesquilinear functional, we have

$$\psi(x, y) = \phi(Px, Qy) = \langle APx, Qy\rangle = \langle Q^*APx, y\rangle.$$

7.11 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.

Exercise 7.14 *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.

Exercise 7.15 *Let $\mathcal{U}$ be an orthonormal basis for the inner-product space V and T a linear operator on V. Show that*

$$[T^*]_{\mathcal{U}} = ([T]_{\mathcal{U}})^\dagger. \tag{7.9}$$

7.12 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 $A^\dagger = A$.

Exercise 7.16 *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 (7.2).*

Exercise 7.17 *Compute the eigenvalues for the real square matrix*

$$A = \begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}. \tag{7.10}$$

Note that the eigenvalues are complex, even though the entries of A are real. The matrix A is not Hermitian.

Exercise 7.18 *Show that the eigenvalues of the complex matrix*

$$B = \begin{bmatrix} 1 & 2+i \\ 2-i & 1 \end{bmatrix} \tag{7.11}$$

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.

Exercise 7.19 *Show that the eigenvalues of the real matrix*

$$C = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \tag{7.12}$$

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?

7.13 It is Good to be "Normal"

For a given linear operator, 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 $V = C^N$ by $Tv = Av$, that is, the operator T works by multiplying each column vector v in C^N by the matrix A. Then A represents T with respect to the usual orthonormal basis $\mathcal{A}$ for C^N. Suppose now that there is an orthonormal basis $\mathcal{U} = \{u^1, ..., u^N\}$ for C^N such that

$$Au^n = \lambda_n u^n,$$

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 (7.14), 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 7.1 *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 7.1 *A self-adjoint linear operator T on a finite-dimensional complex inner-product space V has an orthonormal basis of eigenvectors.*

Exercise 7.20 *Show that the eigenvalues of a self-adjoint linear operator T on a finite-dimensional complex inner-product space are real numbers. Hint: consider $Tu = \lambda_1 u$, 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 7.2 *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.*

Chapter 8

Hermitian and Normal Linear Operators

8.1 The Diagonalization Theorem

In this chapter we present a proof of the following theorem.

Theorem 8.1 *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.*

We saw previously that if V has an orthonormal basis of eigenvectors of 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.

8.2 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 8.1 *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 $T^*x = \overline{\lambda}x$.*

Exercise 8.1 *Prove Proposition (8.1).*

Proposition 8.2 *If T is normal, $Tu^1 = \lambda_1 u^1$, $Tu^2 = \lambda_2 u^2$, and $\lambda_1 \neq \lambda_2$, then $\langle u^1, u^2 \rangle = 0$.*

Exercise 8.2 *Prove Proposition 8.2. Hint: use (e) of Proposition 8.1.*

8.3 Proof of the Diagonalization Theorem

We turn now to the proof of the theorem.

Proof of Theorem 8.1 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 eigenvalue 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, and, therefore, of T. Augmenting this set with the original x^1, we get an orthonormal basis for all of V. ∎

8.4 Corollaries

The theorem has several important corollaries.

Corollary 8.1 *A self-adjoint linear operator T on a finite-dimensional complex inner-product space V has an orthonormal basis of eigenvectors.*

Corollary 8.2 *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.*

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 8.1 *A linear operator P on a finite-dimensional inner-product space is a* perpendicular projection *if*

$$P^2 = P = P^*.$$

Corollary 8.3 (The Spectral Theorem) *Let T be a normal operator on a finite-dimensional inner-product space. Then T can be written as*

$$T = \sum_{m=1}^{M} \lambda_m P_m, \tag{8.1}$$

where λ_m, $m = 1, ..., M$ are the distinct eigenvalues of T, P_m is the perpendicular projection

$$P_m = \sum_{n \in I_m} u^n (u^n)^\dagger, \tag{8.2}$$

and

$$I_m = \{n | \lambda_n = \lambda_m\}.$$

Corollary 8.4 *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 $m \neq k$. ∎

8.5 A Counter-Example

We present now an example of a real 2 by 2 matrix A with $A^T A = AA^T$, but with no eigenvectors in R^2. Take $0 < \theta < \pi$ and A to be the matrix

$$A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{8.3}$$

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.

8.6 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 $S = \frac{1}{2i}(T - T^*)$.

Exercise 8.3 *Show that T is a normal operator if and only if $RS = SR$.*

Theorem 8.2 *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 [24] describes the use of this result in quantum filtering.

Part III

Algorithms

Chapter 9

Fixed-Point Methods

9.1 Chapter Summary

In a broad sense, all iterative algorithms generate a sequence $\{x^k\}$ of vectors. The sequence may converge for any starting vector x^0, or may converge only if the x^0 is sufficiently close to a solution. The limit, when it exists, may depend on x^0, and may, or may not, solve the original problem. Convergence to the limit may be slow and the algorithm may need to be accelerated. The algorithm may involve measured data. The limit may be sensitive to noise in the data and the algorithm may need to be regularized to lessen this sensitivity. The algorithm may be quite general, applying to all problems in a broad class, or it may be tailored to the problem at hand. Each step of the algorithm may be costly, but only a few steps generally needed to produce a suitable approximate answer, or, each step may be easily performed, but many such steps needed. Although convergence of an algorithm is important, theoretically, sometimes in practice only a few iterative steps are used. In this chapter we consider several classes of operators that play important roles in applied linear algebra.

9.2 Operators

A function $T : R^J \to R^J$ is often called an *operator* on R^J. For most of the iterative algorithms we shall consider, the iterative step is

$$x^{k+1} = Tx^k, \tag{9.1}$$

for some operator T. If T is a continuous operator (and it usually is), and the sequence $\{T^k x^0\}$ converges to $\hat{x}$, then $T\hat{x} = \hat{x}$, that is, $\hat{x}$ is a *fixed point* of the operator T. We denote by $\text{Fix}(T)$ the set of fixed points of T. The

convergence of the iterative sequence $\{T^k x^0\}$ will depend on the properties of the operator T.

9.3 Contractions

Contraction operators are perhaps the best known class of operators associated with iterative algorithms.

9.3.1 Lipschitz Continuity

Definition 9.1 *An operator T on R^J is* Lipschitz continuous*, with respect to a vector norm $||\cdot||$, or L-Lipschitz, if there is a positive constant L such that*

$$||Tx - Ty|| \leq L||x - y||, \tag{9.2}$$

for all x and y in R^J.

For example, if $f : R^J \to R$ is differentiable and $\|\nabla f(x)\|_2 \leq L$, for all x, then $T = \nabla f$ is L-Lipschitz, with respect to the 2-norm.

9.3.2 Non-expansive Operators

Definition 9.2 *If $L = 1$, then T is said to be* non-expansive *(ne), with respect to the given norm.*

Lemma 9.1 *Let $T : R^J \to R^J$ be a non-expansive operator, with respect to the 2-norm. Then the set F of fixed points of T is a convex set.*

Proof: Select two distinct points a and b in F, a scalar α in the open interval $(0, 1)$, and let $c = \alpha a + (1 - \alpha)b$. We show that $Tc = c$. Note that

$$a - c = \frac{1 - \alpha}{\alpha}(c - b).$$

We have

$$\|a-b\| = \|a-Tc+Tc-b\| \leq \|a-Tc\|+\|Tc-b\| = \|Ta-Tc\|+\|Tc-Tb\|$$

$$\leq \|a - c\| + \|c - b\| = \|a - b\|;$$

the last equality follows since $a - c$ is a multiple of $(c - b)$. From this, we conclude that

$$\|a - Tc\| = \|a - c\|,$$

$$\|Tc - b\| = \|c - b\|,$$

and that $a - Tc$ and $Tc - b$ are positive multiples of one another, that is, there is $\beta > 0$ such that

$$a - Tc = \beta(Tc - b),$$

or

$$Tc = \frac{1}{1+\beta}a + \frac{\beta}{1+\beta}b = \gamma a + (1-\gamma)b.$$

Then inserting $c = \alpha a + (1-\alpha)b$ and $Tc = \gamma a + (1-\gamma)b$ into

$$\|Tc - b\| = \|c - b\|,$$

we find that $\gamma = \alpha$ and so $Tc = c$. ∎

We want to find properties of an operator T that guarantee that the sequence of iterates $\{T^k x_0\}$ will converge to a fixed point of T, for any x^0, whenever fixed points exist. Being non-expansive is not enough; the non-expansive operator $T = -I$, where $Ix = x$ is the identity operator, has the fixed point $x = 0$, but the sequence $\{T^k x^0\}$ converges only if $x^0 = 0$.

9.3.3 Strict Contractions

One property that guarantees not only that the iterates converge, but that there is a fixed point is the property of being a strict contraction.

Definition 9.3 *An operator T on R^J is a* strict contraction *(sc), with respect to a vector norm $||\cdot||$, if there is $r \in (0,1)$ such that*

$$||Tx - Ty|| \leq r||x - y||, \tag{9.3}$$

for all vectors x and y.

For example, if the operator T is L-Lipschitz for some $L < 1$, then T is a strict contraction. Therefore, if $f : R^J \to R$ is differentiable and $\|f(x)\|_2 \leq L < 1$, for all x, then $T = \nabla f$ is a strict contraction.

For strict contractions, we have the Banach-Picard Theorem [111]:

Theorem 9.1 *Let T be sc. Then, there is a unique fixed point of T and, for any starting vector x^0, the sequence $\{T^k x^0\}$ converges to the fixed point.*

The key step in the proof is to show that $\{x^k\}$ is a Cauchy sequence, therefore, it has a limit.

9.3.4 Eventual Strict Contractions

Consider the problem of finding x such that $x = e^{-x}$. We can see from the graphs of $y = x$ and $y = e^{-x}$ that there is a unique solution, which we shall denote by z. It turns out that $z = 0.56714329040978....$ Let us try

to find z using the iterative sequence $x_{k+1} = e^{-x_k}$, starting with some real x_0. Note that we always have $x_k > 0$ for $k = 1, 2, ...$, even if $x_0 < 0$. The operator here is $Tx = e^{-x}$, which, for simplicity, we view as an operator on the non-negative real numbers.

Since the derivative of the function $f(x) = e^{-x}$ is $f'(x) = -e^{-x}$, we have $|f'(x)| \leq 1$, for all non-negative x, so T is non-expansive. But we do not have $|f'(x)| \leq r < 1$, for all non-negative x; therefore, T is a not a strict contraction, when considered as an operator on the non-negative real numbers.

If we choose $x_0 = 0$, then $x_1 = 1$, $x_2 = 0.368$, approximately, and so on. Continuing this iteration a few more times, we find that after about $k = 14$, the value of x_k settles down to 0.567, which is the answer, to three decimal places. The same thing is seen to happen for any positive starting points x_0. It would seem that T has another property, besides being non-expansive, that is forcing convergence. What is it?

From the fact that $1 - e^{-x} \leq x$, for all real x, with equality if and only if $x = 0$, we can show easily that, for $r = \max\{e^{-x_1}, e^{-x_2}\}$,

$$|z - x_{k+1}| \leq r|z - x_k|,$$

for $k = 3, 4, ...$ Since $r < 1$, it follows, just as in the proof of the Banach-Picard Theorem, that $\{x_k\}$ is a Cauchy sequence and therefore converges. The limit must be a fixed point of T, so the limit must be z.

Although the operator T is not a strict contraction, with respect to the non-negative numbers, once we begin to calculate the sequence of iterates the operator T effectively becomes a strict contraction, with respect to the vectors of the particular sequence being constructed, and so the sequence converges to a fixed point of T. We cannot conclude from this that T has a unique fixed point, as we can in the case of a strict contraction; we must decide that by other means.

9.3.5 Instability

Suppose we rewrite the equation $e^{-x} = x$ as $x = -\log x$, and define $Tx = -\log x$, for $x > 0$. Now our iterative scheme becomes $x_{k+1} = Tx_k = -\log x_k$. A few calculations will convince us that the sequence $\{x_k\}$ is diverging away from the correct answer, not converging to it. The lesson here is that we cannot casually reformulate our problem as a fixed-point problem and expect the iterates to converge to the answer. What matters is the behavior of the operator T.

9.4 Two Useful Identities

The identities in the next two lemmas relate an arbitrary operator T to its complement, $G = I - T$, where I denotes the identity operator. These

identities will allow us to transform properties of T into properties of G that may be easier to work with. A simple calculation is all that is needed to establish the following lemma.

Lemma 9.2 *Let T be an arbitrary operator T on R^J and $G = I - T$. Then*

$$||x - y||_2^2 - ||Tx - Ty||_2^2 = 2(\langle Gx - Gy, x - y\rangle) - ||Gx - Gy||_2^2. \quad (9.4)$$

Lemma 9.3 *Let T be an arbitrary operator T on R^J and $G = I - T$. Then*

$$\langle Tx - Ty, x - y\rangle - ||Tx - Ty||_2^2 =$$

$$\langle Gx - Gy, x - y\rangle - ||Gx - Gy||_2^2. \quad (9.5)$$

Proof: Use the previous lemma. ∎

9.5 Orthogonal Projection Operators

If C is a closed, non-empty convex set in R^J, and x is any vector, then, as we have seen, there is a unique point $P_C x$ in C closest to x, in the sense of the Euclidean distance. This point is called the orthogonal projection of x onto C. If C is a subspace, then we can get an explicit description of $P_C x$ in terms of x; for general convex sets C, however, we will not be able to express $P_C x$ explicitly, and certain approximations will be needed. Orthogonal projection operators are central to our discussion, and, in this overview, we focus on problems involving convex sets, algorithms involving orthogonal projection onto convex sets, and classes of operators derived from properties of orthogonal projection operators.

9.5.1 Properties of the Operator P_C

Although we usually do not have an explicit expression for $P_C x$, we can, however, characterize $P_C x$ as the unique member of C for which

$$\langle P_C x - x, c - P_C x\rangle \geq 0, \quad (9.6)$$

for all c in C; see Proposition 32.4.

P_C is Non-expansive

Recall that an operator T is non-expansive (ne), with respect to a given norm, if, for all x and y, we have

$$||Tx - Ty|| \leq ||x - y||. \quad (9.7)$$

Lemma 9.4 *The orthogonal projection operator $T = P_C$ is non-expansive, with respect to the Euclidean norm, that is,*

$$||P_C x - P_C y||_2 \leq ||x - y||_2, \tag{9.8}$$

for all x and y.

Proof: Use Inequality (9.6) to get

$$\langle P_C y - P_C x, P_C x - x \rangle \geq 0, \tag{9.9}$$

and

$$\langle P_C x - P_C y, P_C y - y \rangle \geq 0. \tag{9.10}$$

Add the two inequalities to obtain

$$\langle P_C x - P_C y, x - y \rangle \geq ||P_C x - P_C y||_2^2, \tag{9.11}$$

and use the Cauchy Inequality. ∎

Because the operator P_C has multiple fixed points, P_C cannot be a strict contraction, unless the set C is a singleton set.

P_C is Firmly Non-expansive

Definition 9.4 *An operator T is said to be* firmly non-expansive *(fne) if*

$$\langle Tx - Ty, x - y \rangle \geq ||Tx - Ty||_2^2, \tag{9.12}$$

for all x and y in R^J.

Lemma 9.5 *An operator T is fne if and only if $G = I - T$ is fne.*

Proof: Use the identity in Equation (9.5). ∎

From Equation (9.11), we see that the operator $T = P_C$ is not simply ne, but fne, as well. A good source for more material on these topics is the book by Goebel and Reich [134].

The Search for Other Properties of P_C

The class of non-expansive operators is too large for our purposes; the operator $Tx = -x$ is non-expansive, but the sequence $\{T^k x^0\}$ does not converge, in general, even though a fixed point, $x = 0$, exists. The class of firmly non-expansive operators is too small for our purposes. Although the convergence of the iterative sequence $\{T^k x^0\}$ to a fixed point does hold for firmly non-expansive T, whenever fixed points exist, the product of two or more fne operators need not be fne; that is, the class of fne

operators is not *closed to finite products*. This poses a problem, since, as we shall see, products of orthogonal projection operators arise in several of the algorithms we wish to consider. We need a class of operators smaller than the ne ones, but larger than the fne ones, closed to finite products, and for which the sequence of iterates $\{T^k x^0\}$ will converge, for any x^0, whenever fixed points exist. The class we shall consider is the class of *averaged* operators.

9.6 Averaged Operators

The term 'averaged operator' appears in the work of Baillon, Bruck and Reich [31, 9]. There are several ways to define averaged operators. One way is in terms of the complement operator.

Definition 9.5 *An operator G on R^J is called* ν-inverse strongly monotone *(ν-ism)[135] (also called co-coercive in [90]) if there is $\nu > 0$ such that*

$$\langle Gx - Gy, x - y\rangle \geq \nu ||Gx - Gy||_2^2. \tag{9.13}$$

Lemma 9.6 *An operator T is ne if and only if its complement $G = I - T$ is $\frac{1}{2}$-ism, and T is fne if and only if G is 1-ism, and if and only if G is fne. Also, T is ne if and only if $F = (I + T)/2$ is fne. If G is ν-ism and $\gamma > 0$ then the operator γG is $\frac{\nu}{\gamma}$-ism.*

Definition 9.6 *An operator T is called averaged (av) if $G = I - T$ is ν-ism for some $\nu > \frac{1}{2}$. If G is $\frac{1}{2\alpha}$-ism, for some $\alpha \in (0, 1)$, then we say that T is α-av.*

It follows that every av operator is ne, with respect to the Euclidean norm, and every fne operator is av.

The averaged operators are sometimes defined in a different, but equivalent, way, using the following characterization of av operators.

Lemma 9.7 *An operator T is av if and only if, for some operator N that is non-expansive in the Euclidean norm, and $\alpha \in (0, 1)$, we have*

$$T = (1 - \alpha)I + \alpha N.$$

Consequently, the operator T is av if and only if, for some α in $(0, 1)$, the operator

$$N = \frac{1}{\alpha}T - \frac{1 - \alpha}{\alpha}I = I - \frac{1}{\alpha}(I - T) = I - \frac{1}{\alpha}G$$

is non-expansive.

Proof: We assume first that there is $\alpha \in (0,1)$ and ne operator N such that $T = (1-\alpha)I + \alpha N$, and so $G = I - T = \alpha(I - N)$. Since N is ne, $I - N$ is $\frac{1}{2}$-ism and $G = \alpha(I-N)$ is $\frac{1}{2\alpha}$-ism. Conversely, assume that G is ν-ism for some $\nu > \frac{1}{2}$. Let $\alpha = \frac{1}{2\nu}$ and write $T = (1-\alpha)I + \alpha N$ for $N = I - \frac{1}{\alpha}G$. Since $I - N = \frac{1}{\alpha}G$, $I - N$ is $\alpha\nu$-ism. Consequently $I - N$ is $\frac{1}{2}$-ism and N is ne. ∎

An averaged operator is easily constructed from a given ne operator N by taking a convex combination of N and the identity I. The beauty of the class of av operators is that it contains many operators, such as P_C, that are not originally defined in this way. As we shall see shortly, finite products of averaged operators are again averaged, so the product of finitely many orthogonal projections is av.

We present now the fundamental properties of averaged operators, in preparation for the proof that the class of averaged operators is closed to finite products.

Note that we can establish that a given operator is av by showing that there is an α in the interval $(0,1)$ such that the operator

$$\frac{1}{\alpha}(A - (1-\alpha)I) \tag{9.14}$$

is ne. Using this approach, we can easily show that if T is sc, then T is av.

Lemma 9.8 *Let $T = (1-\alpha)A + \alpha N$ for some $\alpha \in (0,1)$. If A is averaged and N is non-expansive then T is averaged.*

Proof: Let $A = (1-\beta)I + \beta M$ for some $\beta \in (0,1)$ and ne operator M. Let $1 - \gamma = (1-\alpha)(1-\beta)$. Then we have

$$T = (1-\gamma)I + \gamma[(1-\alpha)\beta\gamma^{-1}M + \alpha\gamma^{-1}N]. \tag{9.15}$$

Since the operator $K = (1-\alpha)\beta\gamma^{-1}M + \alpha\gamma^{-1}N$ is easily shown to be ne and the convex combination of two ne operators is again ne, T is averaged. ∎

Corollary 9.1 *If A and B are av and α is in the interval $[0,1]$, then the operator $T = (1-\alpha)A + \alpha B$ formed by taking the convex combination of A and B is av.*

Corollary 9.2 *Let $T = (1-\alpha)F + \alpha N$ for some $\alpha \in (0,1)$. If F is fne and N is Euclidean-ne then T is averaged.*

The orthogonal projection operators P_H onto hyperplanes $H = H(a,\gamma)$ are sometimes used with *relaxation*, which means that P_H is replaced by the operator

$$T = (1-\omega)I + \omega P_H, \tag{9.16}$$

for some ω in the interval $(0, 2)$. Clearly, if ω is in the interval $(0, 1)$, then T is av, by definition, since P_H is ne. We want to show that, even for ω in the interval $[1, 2)$, T is av. To do this, we consider the operator $R_H = 2P_H - I$, which is reflection through H; that is,

$$P_H x = \frac{1}{2}(x + R_H x), \tag{9.17}$$

for each x.

Lemma 9.9 *The operator $R_H = 2P_H - I$ is an isometry; that is,*

$$||R_H x - R_H y||_2 = ||x - y||_2, \tag{9.18}$$

for all x and y, so that R_H is ne.

Lemma 9.10 *For $\omega = 1 + \gamma$ in the interval $[1, 2)$, we have*

$$(1 - \omega)I + \omega P_H = \alpha I + (1 - \alpha)R_H, \tag{9.19}$$

for $\alpha = \frac{1-\gamma}{2}$; therefore, $T = (1 - \omega)I + \omega P_H$ is av.

The product of finitely many ne operators is again ne, while the product of finitely many fne operators, even orthogonal projections, need not be fne. It is a helpful fact that the product of finitely many av operators is again av.

If $A = (1 - \alpha)I + \alpha N$ is averaged and B is averaged then $T = AB$ has the form $T = (1 - \alpha)B + \alpha NB$. Since B is av and NB is ne, it follows from Lemma 9.8 that T is averaged. Summarizing, we have

Proposition 9.1 *If A and B are averaged, then $T = AB$ is averaged.*

Proposition 9.2 *An operator F is firmly non-expansive if and only if $F = \frac{1}{2}(I + N)$, for some non-expansive operator N.*

9.6.1 Gradient Operators

Another type of operator that is averaged can be derived from gradient operators.

Definition 9.7 *An operator T on R^J is* monotone *if*

$$\langle Tx - Ty, x - y\rangle \geq 0, \tag{9.20}$$

for all x and y.

Firmly non-expansive operators on R^J are monotone operators. Let $g(x) : R^J \to R$ be a differentiable convex function and $f(x) = \nabla g(x)$ its gradient. The operator ∇g is also monotone. If ∇g is non-expansive, then it can be shown that ∇g is fne . If, for some $L > 0$, ∇g is L-Lipschitz, for the 2-norm, that is,

$$||\nabla g(x) - \nabla g(y)||_2 \leq L||x - y||_2, \tag{9.21}$$

for all x and y, then $\frac{1}{L}\nabla g$ is ne, therefore fne, and the operator $T = I - \gamma\nabla g$ is av, for $0 < \gamma < \frac{2}{L}$.

9.6.2 The Krasnoselskii-Mann Theorem

For any operator T that is averaged, convergence of the sequence $\{T^k x^0\}$ to a fixed point of T, whenever fixed points of T exist, is guaranteed by the Krasnoselskii-Mann (KM) Theorem [186]:

Theorem 9.2 *Let T be averaged. Then the sequence $\{T^k x^0\}$ converges to a fixed point of T, whenever Fix(T) is non-empty.*

Proof: Let z be a fixed point of non-expansive operator N and let $\alpha \in (0, 1)$. Let $T = (1 - \alpha)I + \alpha N$, so the iterative step becomes

$$x^{k+1} = Tx^k = (1 - \alpha)x^k + \alpha N x^k. \tag{9.22}$$

The identity in Equation (9.4) is the key to proving Theorem 9.2.

Using $Tz = z$ and $(I - T)z = 0$ and setting $G = I - T$ we have

$$||z - x^k||_2^2 - ||Tz - x^{k+1}||_2^2 = 2\langle Gz - Gx^k, z - x^k\rangle - ||Gz - Gx^k||_2^2. \tag{9.23}$$

Since, by Lemma 9.7, G is $\frac{1}{2\alpha}$-ism, we have

$$||z - x^k||_2^2 - ||z - x^{k+1}||_2^2 \geq (\frac{1}{\alpha} - 1)||x^k - x^{k+1}||_2^2. \tag{9.24}$$

Consequently the sequence $\{x^k\}$ is bounded, the sequence $\{||z - x^k||_2\}$ is decreasing and the sequence $\{||x^k - x^{k+1}||_2\}$ converges to zero. Let x^* be a cluster point of $\{x^k\}$. Then we have $Tx^* = x^*$, so we may use x^* in place of the arbitrary fixed point z. It follows then that the sequence $\{||x^* - x^k||_2\}$ is decreasing; since a subsequence converges to zero, the entire sequence converges to zero. The proof is complete. ∎

A version of the KM Theorem 9.2, with variable coefficients, appears in Reich's paper [213].

9.7 Affine Linear Operators

It may not always be easy to decide if a given operator is averaged. The class of affine linear operators provides an interesting illustration of the problem.

The affine operator $Tx = Bx + d$ will be ne, sc, fne, or av precisely when the linear operator given by multiplication by the matrix B is the same.

9.7.1 The Hermitian Case

When B is Hermitian, we can determine if B belongs to these classes by examining its eigenvalues λ:

- B is non-expansive if and only if $-1 \leq \lambda \leq 1$, for all λ;
- B is averaged if and only if $-1 < \lambda \leq 1$, for all λ;
- B is a strict contraction if and only if $-1 < \lambda < 1$, for all λ;
- B is firmly non-expansive if and only if $0 \leq \lambda \leq 1$, for all λ.

Affine linear operators T that arise, for instance, in splitting methods for solving systems of linear equations, generally have non-Hermitian linear part B. Deciding if such operators belong to these classes is more difficult. Instead, we can ask if the operator is *paracontractive*, with respect to some norm.

9.8 Paracontractive Operators

By examining the properties of the orthogonal projection operators P_C, we were led to the useful class of averaged operators. The orthogonal projections also belong to another useful class, the paracontractions.

Definition 9.8 *An operator T is called* paracontractive *(pc), with respect to a given norm, if, for every fixed point y of T, we have*

$$||Tx - y|| < ||x - y||, \tag{9.25}$$

unless $Tx = x$.

Paracontractive operators are studied by Censor and Reich in [80].

Proposition 9.3 *The operators $T = P_C$ are paracontractive, with respect to the Euclidean norm.*

Proof: It follows from Cauchy's Inequality that

$$||P_C x - P_C y||_2 \leq ||x - y||_2,$$

with equality if and only if

$$P_C x - P_C y = \alpha(x - y),$$

for some scalar α with $|\alpha| = 1$. But, because

$$0 \leq \langle P_C x - P_C y, x - y \rangle = \alpha ||x - y||_2^2,$$

it follows that $\alpha = 1$, and so

$$P_C x - x = P_C y - y.$$

∎

When we ask if a given operator T is pc, we must specify the norm. We often construct the norm specifically for the operator involved. To illustrate, we consider the case of affine operators.

9.8.1 Linear and Affine Paracontractions

Let the matrix B be diagonalizable and let the columns of V be an eigenvector basis. Then we have $V^{-1}BV = D$, where D is the diagonal matrix having the eigenvalues of B along its diagonal.

Lemma 9.11 *A square matrix B is diagonalizable if all its eigenvalues are distinct.*

Proof: Let B be J by J. Let λ_j be the eigenvalues of B, $Bx^j = \lambda_j x^j$, and $x^j \neq 0$, for $j = 1, ..., J$. Let x^m be the first eigenvector that is in the span of $\{x_j | j = 1, ..., m-1\}$. Then

$$x^m = a_1 x^1 + ...a_{m-1} x^{m-1}, \tag{9.26}$$

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

$$\lambda_m x^m = a_1 \lambda_m x^1 + ...a_{m-1} \lambda_m x^{m-1}. \tag{9.27}$$

From

$$\lambda_m x^m = Ax^m = a_1 \lambda_1 x^1 + ...a_{m-1} \lambda_{m-1} x^{m-1}, \tag{9.28}$$

it follows that

$$a_1(\lambda_m - \lambda_1)x^1 + ... + a_{m-1}(\lambda_m - \lambda_{m-1})x^{m-1} = 0, \tag{9.29}$$

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

We see from this Lemma that almost all square matrices B are diagonalizable. Indeed, all Hermitian B are diagonalizable. If B has real entries, but is not symmetric, then the eigenvalues of B need not be real, and the eigenvectors of B can have non-real entries. Consequently, we must consider B as a linear operator on C^J, if we are to talk about diagonalizability. For example, consider the real matrix

$$B = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{9.30}$$

Its eigenvalues are $\lambda = i$ and $\lambda = -i$. The corresponding eigenvectors are $(1, i)^T$ and $(1, -i)^T$. The matrix B is then diagonalizable as an operator on C^2, but not as an operator on R^2.

When B is not Hermitian, it is not as easy to determine if the affine operator T is sc with respect to a given norm. Instead, we often tailor the norm to the operator T. Suppose that B is a diagonalizable matrix, that is, there is a basis for R^J consisting of eigenvectors of B. Let $\{u^1, ..., u^J\}$ be such a basis, and let $Bu^j = \lambda_j u^j$, for each $j = 1, ..., J$. For each x in R^J, there are unique coefficients a_j so that

$$x = \sum_{j=1}^{J} a_j u^j. \tag{9.31}$$

Then let

$$||x|| = \sum_{j=1}^{J} |a_j|. \tag{9.32}$$

Lemma 9.12 *The expression $||\cdot||$ in Equation (9.32) defines a norm on R^J. If $\rho(B) < 1$, then the affine operator T is sc, with respect to this norm.*

It is known that, for any square matrix B and any $\epsilon > 0$, there is a vector norm for which the induced matrix norm satisfies $||B|| \leq \rho(B) + \epsilon$. Therefore, if B is an arbitrary square matrix with $\rho(B) < 1$, there is a vector norm with respect to which B is sc.

Proposition 9.4 *Let T be an affine linear operator whose linear part B is diagonalizable, and $|\lambda| < 1$ for all eigenvalues λ of B that are not equal to one. Then the operator T is pc, with respect to the norm given by Equation (9.32).*

Proof: This is Exercise 9.8. ∎

We see from Proposition 9.4 that, for the case of affine operators T whose linear part is not Hermitian, instead of asking if T is av, we can ask if T is pc; since B will almost certainly be diagonalizable, we can answer this question by examining the eigenvalues of B.

Unlike the class of averaged operators, the class of paracontractive operators is not necessarily closed to finite products, unless those factor operators have a common fixed point.

9.8.2 The Elsner-Koltracht-Neumann Theorem

Our interest in paracontractions is due to the Elsner-Koltracht-Neumann (EKN) Theorem [114]:

Theorem 9.3 *Let T be pc with respect to some vector norm. If T has fixed points, then the sequence $\{T^k x^0\}$ converges to a fixed point of T, for all starting vectors x^0.*

We follow the development in [114].

Theorem 9.4 *Suppose that there is a vector norm on R^J, with respect to which each T_i is a pc operator, for $i = 1, ..., I$, and that $F = \cap_{i=1}^{I} \mathrm{Fix}(T_i)$ is not empty. For $k = 0, 1, ...$, let $i(k) = k(\mathrm{mod}\, I) + 1$, and $x^{k+1} = T_{i(k)} x^k$. The sequence $\{x^k\}$ converges to a member of F, for every starting vector x^0.*

Proof: Let $y \in F$. Then, for $k = 0, 1, ...$,

$$||x^{k+1} - y|| = ||T_{i(k)} x^k - y|| \leq ||x^k - y||, \tag{9.33}$$

so that the sequence $\{||x^k - y||\}$ is decreasing; let $d \geq 0$ be its limit. Since the sequence $\{x^k\}$ is bounded, we select an arbitrary cluster point, x^*. Then $d = ||x^* - y||$, from which we can conclude that

$$||T_i x^* - y|| = ||x^* - y||, \tag{9.34}$$

and $T_i x^* = x^*$, for $i = 1, ..., I$; therefore, $x^* \in F$. Replacing y, an arbitrary member of F, with x^*, we have that $||x^k - x^*||$ is decreasing. But, a subsequence converges to zero, so the whole sequence must converge to zero. This completes the proof. ∎

Corollary 9.3 *If T is pc with respect to some vector norm, and T has fixed points, then the iterative sequence $\{T^k x^0\}$ converges to a fixed point of T, for every starting vector x^0.*

Corollary 9.4 *If $T = T_I T_{I-1} \cdots T_2 T_1$, and $F = \cap_{i=1}^{I} \mathrm{Fix}\,(T_i)$ is not empty, then $F = \mathrm{Fix}\,(T)$.*

Proof: The sequence $x^{k+1} = T_{i(k)}x^k$ converges to a member of $\mathrm{Fix}\,(T)$, for every x^0. Select x^0 in F. ∎

Corollary 9.5 *The product T of two or more pc operators T_i, $i = 1, ..., I$ is again a pc operator, if $F = \cap_{i=1}^{I}\mathrm{Fix}\,(T_i)$ is not empty.*

Proof: Suppose that for $T = T_I T_{I-1} \cdots T_2 T_1$, and $y \in F = \mathrm{Fix}\,(T)$, we have

$$||Tx - y|| = ||x - y||. \tag{9.35}$$

Then, since

$$||T_I(T_{I-1} \cdots T_1)x - y|| \leq ||T_{I-1} \cdots T_1 x - y|| \leq ... \leq ||T_1 x - y|| \leq ||x - y||, \tag{9.36}$$

it follows that

$$||T_i x - y|| = ||x - y||, \tag{9.37}$$

and $T_i x = x$, for each i. Therefore, $Tx = x$. ∎

9.9 Exercises

Exercise 9.1 *Show that a strict contraction can have at most one fixed point.*

Exercise 9.2 *Let T be sc. Show that the sequence $\{T^k x_0\}$ is a Cauchy sequence. Hint: consider*

$$||x^k - x^{k+n}|| \leq ||x^k - x^{k+1}|| + ... + ||x^{k+n-1} - x^{k+n}||, \tag{9.38}$$

and use

$$||x^{k+m} - x^{k+m+1}|| \leq r^m ||x^k - x^{k+1}||. \tag{9.39}$$

Since $\{x^k\}$ is a Cauchy sequence, it has a limit, say $\hat{x}$. Let $e^k = \hat{x} - x^k$. Show that $\{e^k\} \to 0$, as $k \to +\infty$, so that $\{x^k\} \to \hat{x}$. Finally, show that $T\hat{x} = \hat{x}$.

Exercise 9.3 *Suppose that we want to solve the equation*

$$x = \frac{1}{2}e^{-x}.$$

Let $Tx = \frac{1}{2}e^{-x}$ for x in R. Show that T is a strict contraction, when restricted to non-negative values of x, so that, provided we begin with $x^0 > 0$, the sequence $\{x^k = Tx^{k-1}\}$ converges to the unique solution of the equation. Hint: use the mean value theorem from calculus.

Exercise 9.4 *Prove Lemma 9.12.*

Exercise 9.5 *Show that, if the operator T is α-av and $1 > \beta > \alpha$, then T is β-av.*

Exercise 9.6 *Prove Lemma 9.6.*

Exercise 9.7 *Prove Proposition 9.2.*

Exercise 9.8 *Prove Proposition 9.4.*

Exercise 9.9 *Show that, if B is a linear av operator, then $|\lambda| < 1$ for all eigenvalues λ of B that are not equal to one.*

9.10 Course Homework

Do all the exercises in this chapter.

Chapter 10

Jacobi and Gauss-Seidel Methods

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* [99]. 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.

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.

10.1 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.

10.2 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, but only av or pc. The operator T defined by

$$Tz = M^{-1}Kz + M^{-1}h = Bz + d \tag{10.6}$$

is an affine linear operator and will be a sc or av operator whenever $B = M^{-1}K$ is.

It follows from our previous discussion concerning linear av operators that, if $B = B^\dagger$ is Hermitian, then B is av if and only if

$$-1 < \lambda \leq 1, \tag{10.7}$$

for all (necessarily real) eigenvalues λ of B.

In general, though, the matrix $B = M^{-1}K$ will not be Hermitian, and deciding if such a non-Hermitian matrix is av is not a simple matter. We do know that, if B is av, so is $B^\dagger$; consequently, the Hermitian matrix $Q = \frac{1}{2}(B + B^\dagger)$ is also av. Therefore, $I - Q = \frac{1}{2}(M^{-1}S + (M^{-1}S)^\dagger)$ is ism, and so is non-negative definite. We have $-1 < \lambda \leq 1$, for any eigenvalue λ of Q.

Alternatively, we can use Theorem 9.3. According to that theorem, if B has a basis of eigenvectors, and $|\lambda| < 1$ for all eigenvalues λ of B that are not equal to one, then $\{z^k\}$ will converge to a solution of $Sz = h$, whenever solutions exist.

In what follows we shall write an arbitrary square matrix S as

$$S = L + D + U, \tag{10.8}$$

where L is the strictly lower triangular part of S, D the diagonal part, and U the strictly upper triangular part. When S is Hermitian, we have

$$S = L + D + L^\dagger. \tag{10.9}$$

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.

10.3 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

$$B = M^{-1}K = -D^{-1}(L+U). \tag{10.10}$$

The Gauss-Seidel Method: The Gauss-Seidel (GS) method uses the splitting $M = D + L$, so that the matrix B is

$$B = I - (D+L)^{-1}S. \tag{10.11}$$

The Jacobi Overrelaxation Method (JOR): The JOR uses the splitting

$$M = \frac{1}{\omega}D \tag{10.12}$$

and

$$K = M - S = (\frac{1}{\omega} - 1)D - L - U. \tag{10.13}$$

The matrix B is

$$B = M^{-1}K = (I - \omega D^{-1}S). \tag{10.14}$$

The Successive Overrelaxation Method (SOR): The SOR uses the splitting $M = (\frac{1}{\omega}D + L)$, so that

$$B = M^{-1}K = (D + \omega L)^{-1}[(1-\omega)D - \omega U] \tag{10.15}$$

or

$$B = I - \omega(D + \omega L)^{-1}S, \tag{10.16}$$

or

$$B = (I + \omega D^{-1}L)^{-1}[(1-\omega)I - \omega D^{-1}U]. \tag{10.17}$$

10.4 Jacobi's Algorithm and JOR

The matrix B in Equation (10.10) is not generally av and 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 $\rho(B) < 1$. It follows from Lemma 6.5 that B is sc with respect to some vector norm, and the Jacobi iteration converges. If, in addition, S is

Hermitian, the eigenvalues of B are in the interval $(-1, 1)$, and so B is sc with respect to the Euclidean norm.

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

$$z_m^{\text{new}} = S_{mm}^{-1}[h_m - \sum_{j \neq m} S_{mj} z_j^{\text{old}}], \tag{10.18}$$

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

$$z^{\text{new}} = z^{\text{old}} + D^{-1}(h - Sz^{\text{old}}). \tag{10.19}$$

The *Jacobi overrelaxation* (JOR) method has the following full-cycle iterative step:

$$z^{\text{new}} = z^{\text{old}} + \omega D^{-1}(h - Sz^{\text{old}}); \tag{10.20}$$

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 is Hermitian, nonnegative-definite, for example, $S = A^\dagger A$ or $S = AA^\dagger$, we can say more.

10.4.1 The JOR in the Nonnegative-definite Case

When S is nonnegative-definite and the system $Sz = h$ is consistent the JOR converges to a solution for any $\omega \in (0, 2/\rho(D^{-1/2}SD^{-1/2}))$, where $\rho(Q)$ denotes the largest eigenvalue of the nonnegative-definite matrix Q. For nonnegative-definite S, the convergence of the JOR method is implied by the KM Theorem 9.2, since the JOR is equivalent to Landweber's algorithm in these cases.

The JOR method, as applied to $Sz = AA^\dagger z = b$, is equivalent to the Landweber iterative method for $Ax = b$.

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 $S = AA^\dagger$.*

If we select $\omega = 1/I$ we obtain the Cimmino method. Since the trace of the matrix $D^{-1/2}SD^{-1/2}$ equals I we know that $\omega = 1/I$ is not greater than the largest eigenvalue of the matrix $D^{-1/2}SD^{-1/2}$ 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 $Ax = b$.

Lemma 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 $S = A^\dagger A$.*

10.5 The Gauss-Seidel Algorithm and SOR

In general, the full-cycle iterative step of the Gauss-Seidel method is the following:

$$z^{\text{new}} = z^{\text{old}} + (D + L)^{-1}(h - Sz^{\text{old}}), \tag{10.21}$$

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.

10.5.1 The Nonnegative-Definite Case

Now we consider the square system $Sz = h$, assuming that $S = L+D+L^\dagger$ is Hermitian and nonnegative-definite, so that $x^\dagger Sx \geq 0$, 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 $|\lambda| < 1$.*

If B is diagonalizable, then there is a norm with respect to which T is paracontractive, so, by the EKN Theorem 9.3, the GS iteration converges to a solution of $Sz = h$, whenever solutions exist.

Proof of Proposition (10.1): Let $Bv = \lambda v$, for v nonzero. Then $-Bv = (D + L)^{-1}L^\dagger v = -\lambda v$, so that

$$L^\dagger v = -\lambda(D + L)v, \tag{10.22}$$

and

$$Lv = -\overline{\lambda}(D+L)^\dagger v. \tag{10.23}$$

Therefore,

$$v^\dagger L^\dagger v = -\lambda v^\dagger (D+L)v. \tag{10.24}$$

Adding $v^\dagger(D+L)v$ to both sides, we get

$$v^\dagger S v = (1-\lambda)v^\dagger(D+L)v. \tag{10.25}$$

Since the left side of the equation is real, so is the right side. Therefore

$$\begin{aligned}(1-\overline{\lambda})(D+L)^\dagger v &= (1-\lambda)v^\dagger(D+L)v \\ &= (1-\lambda)v^\dagger Dv + (1-\lambda)v^\dagger Lv \\ &= (1-\lambda)v^\dagger Dv - (1-\lambda)\overline{\lambda}v^\dagger(D+L)^\dagger v. \end{aligned} \tag{10.26}$$

So we have

$$[(1-\overline{\lambda}) + (1-\lambda)\overline{\lambda}]v^\dagger(D+L)^\dagger v = (1-\lambda)v^\dagger Dv, \tag{10.27}$$

or

$$(1-|\lambda|^2)v^\dagger(D+L)^\dagger v = (1-\lambda)v^\dagger Dv. \tag{10.28}$$

Multiplying by $(1-\overline{\lambda})$ on both sides, we get, on the left side,

$$(1-|\lambda|^2)v^\dagger(D+L)^\dagger v - (1-|\lambda|^2)\overline{\lambda}v^\dagger(D+L)^\dagger v, \tag{10.29}$$

which is equal to

$$(1-|\lambda|^2)v^\dagger(D+L)^\dagger v + (1-|\lambda|^2)v^\dagger Lv, \tag{10.30}$$

and, on the right side, we get

$$|1-\lambda|^2 v^\dagger Dv. \tag{10.31}$$

Consequently, we have

$$(1-|\lambda|^2)v^\dagger S v = |1-\lambda|^2 v^\dagger Dv. \tag{10.32}$$

Since $v^\dagger Sv \geq 0$ 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 $Sv = 0$. Therefore, if S is invertible, the affine linear operator T is a strict contraction, and the GS iteration converges to the unique solution of $Sz = h$.

10.5.2 Successive Overrelaxation

The *successive overrelaxation* (SOR) method has the following full-cycle iterative step:

$$z^{\text{new}} = z^{\text{old}} + (\omega^{-1}D + L)^{-1}(h - Sz^{\text{old}}); \tag{10.33}$$

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

$$B = (D + \omega L)^{-1}[(1 - \omega)D - \omega U] \tag{10.34}$$

we can show that

$$|\det(B)| = |1 - \omega|^N. \tag{10.35}$$

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 is Hermitian, nonnegative-definite, as, for example, when we take $S = A^\dagger A$ or $S = AA^\dagger$, we can say more.

10.5.3 The SOR for Nonnegative-Definite S

When S is nonnegative-definite and the system $Sz = h$ 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 S, the SOR is equivalent to the ART.

Now we 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 $i = k(\text{mod}\, I) + 1$. We have

$$z_i^{k+1} = (1 - \omega)z_i^k + \omega D_{ii}^{-1}(b_i - \sum_{n=1}^{i-1} S_{in}z_n^k - \sum_{n=i+1}^{I} S_{in}z_n^k) \tag{10.36}$$

and $z_n^{k+1} = z_n^k$ for $n \neq i$. Now we calculate $x^{k+1} = A^\dagger z^{k+1}$:

$$x_j^{k+1} = x_j^k + \omega D_{ii}^{-1}\overline{A_{ij}}(b_i - (Ax^k)_i). \tag{10.37}$$

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 $S = AA^\dagger$ 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 $S = AA^\dagger$ is invertible, the SOR sequence $\{z^k\}$ then converges to $S^{-1}b$.

Chapter 11

The ART and MART Again

11.1 The ART in the General Case

Although the ART was developed to compute tomographic images, it can be viewed more generally as an iterative method for solving an arbitrary system of linear equations, $Ax = b$.

Let A be a complex matrix with I rows and J columns, and let b be a member of 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

$$H_i = \{x|(Ax)_i = b_i\}, \tag{11.1}$$

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

$$x^{k+1} = P_{i(k)}x^k. \tag{11.2}$$

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

11.1.1 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

$$x_j = z_j + \overline{A_{ij}}(b_i - (Az)_i)/\sum_{m=1}^{J} |A_{im}|^2. \tag{11.3}$$

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

$$\sum_{j=1}^{J} |A_{ij}|^2 = 1, \tag{11.4}$$

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

$$x_j^{k+1} = x_j^k + \overline{A_{ij}}(b_i - (Ax^k)_i). \tag{11.5}$$

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.

11.1.2 Full-cycle ART

We also consider the *full-cycle* ART, with iterative step $z^{k+1} = Tz^k$, for

$$T = P_I P_{I-1} \cdots P_2 P_1. \tag{11.6}$$

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.

11.1.3 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 11.1 (Relaxed ART) *With $\omega \in (0, 2)$, x^0 arbitrary, and $i = k(\mathrm{mod}\, I) + 1$, let*

$$x_j^{k+1} = x_j^k + \omega\overline{A_{ij}}(b_i - (Ax^k)_i)). \tag{11.7}$$

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.

11.1.4 Constrained ART

Let C be a closed, nonempty convex subset of 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 11.2 (Constrained ART) *With x^0 arbitrary and $i = k(\mathrm{mod}\, I)+1$, let*

$$x_j^{k+1} = P_C(x_j^k + \overline{A_{ij}}(b_i - (Ax^k)_i)). \tag{11.8}$$

For example, if A and b are real and we seek a nonnegative solution to $Ax = b$, we can use

Algorithm 11.3 (Non-negative ART) *With x^0 arbitrary and $i = k(\mathrm{mod}\, I)+1$, let*

$$x_j^{k+1} = (x_j^k + A_{ij}(b_i - (Ax^k)_i))_+, \tag{11.9}$$

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 can manifest itself in a variety of ways; we have seen what can happen when we impose positivity on the calculated least-squares solution, that is, when we minimize $||Ax-b||_2$ over all non-negative vectors x. Theorem 11.1 tells us that when $J > I$, but $Ax = b$ has no non-negative solutions, the non-negatively constrained least-squares solution can have at most $I-1$ non-zero entries, regardless of how large J is. This phenomenon also occurs with several other approximate methods, such as those that minimize the cross-entropy distance.

Definition 11.1 *The matrix A has the* full-rank property *if A and every matrix Q obtained from A by deleting columns have full rank.*

Theorem 11.1 *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.*

For a proof, see the chapter on optimization.

11.1.5 When $Ax = b$ Has Solutions

For the consistent case, in which the system $Ax = b$ has exact solutions, we have the following result.

Theorem 11.2 *Let $A\hat{x} = b$ and let x^0 be arbitrary. Let $\{x^k\}$ be generated by Equation (11.5). 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 following lemma follows immediately from the definition of the ART iteration.

Lemma 11.1 *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*

$$||x^0 - y^0||_2^2 - ||x^I - y^I||_2^2 = \sum_{i=1}^{I} |(Ax^{i-1})_i - (Ay^{i-1})_i|^2. \tag{11.10}$$

Proof of Theorem 11.2: 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 (11.10) 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 (11.10), 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. ∎

11.1.6 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 [231] and also follows from the results of De Pierro and Iusem [102]. Proofs of subsequential convergence are given in [62, 63].

The ART limit cycle will vary with the ordering of the equations, and contains more than one vector unless an exact solution exists. There are several open questions about the limit cycle.

Open Question: For a fixed ordering, does the limit cycle depend on the initial vector x^0? If so, how?

11.1.7 The Geometric Least-Squares Solution

When the system $Ax = b$ has no solutions, it is reasonable to seek an approximate solution, such as the *least squares* solution, $x_{LS} = (A^\dagger A)^{-1}A^\dagger b$, 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

$$(A^\dagger A)^{-1}A^\dagger b, \tag{11.11}$$

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

$$(A^\dagger W^\dagger WA)^{-1}A^\dagger W^\dagger b, \tag{11.12}$$

and these need not be the same.

A simple example is the following. Consider the system

$$x = 1$$

$$x = 2, \tag{11.13}$$

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

$$2x = 2$$

$$x = 2, \tag{11.14}$$

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

Definition 11.2 *The* geometric least-squares *solution of $Ax = b$ is the 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 $x = 1.2$.

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 [52] (see also [62]) 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.

11.2 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. Instead of solving $Ax = b$, we *regularize* by minimizing, for example, the function

$$F_\epsilon(x) = ||Ax - b||_2^2 + \epsilon^2 ||x||_2^2. \tag{11.15}$$

The solution to this problem is the vector

$$\hat{x}_\epsilon = (A^\dagger A + \epsilon^2 I)^{-1} A^\dagger b. \tag{11.16}$$

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 [62], while the second one is due to Eggermont, Herman, and Lent [113].

In our first method we use ART to solve the system of equations given in matrix form by

$$\begin{bmatrix} A^\dagger & \epsilon I \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = 0. \tag{11.17}$$

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

$$\begin{bmatrix} A & \epsilon I \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} = b. \tag{11.18}$$

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 $\epsilon v^\infty = b - A\hat{x}_\epsilon$.

Open Question: In both the consistent and inconsistent cases, the sequence $\{x^k\}$ of ART iterates is bounded, as Tanabe [231], and De Pierro and Iusem [102] have shown. The proof is easy in the consistent case. Is there an easy proof for the inconsistent case?

11.3 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) [56]:

11.3.1 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 $Ax = y$.

11.3.2 Strongly Under-relaxed ART

Another method for avoiding the LC is *strong under-relaxation*, due to Censor, Eggermont and Gordon [73]. Let $t > 0$. Replace the iterative step in ART with

$$x_j^{k+1} = x_j^k + t\overline{A_{ij}}(b_i - (Ax^k)_i). \tag{11.19}$$

In [73] 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 [52]. Bertsekas [19] uses strong under-relaxation to obtain convergence of more general incremental methods.

Exercise 11.1 *Prove that the two iterative methods for regularized ART perform as indicated.*

11.4 The MART

The *multiplicative* ART (MART) [138] 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 begin with a simpler case, transmission tomographic imaging, in which the relation is most clearly visible.

11.4.1 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 11.4 (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*

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i}\Big)^{m_i^{-1}A_{ij}}, \tag{11.20}$$

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 11.5 (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*

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i}\Big)^{\gamma_i m_i^{-1}A_{ij}}, \tag{11.21}$$

where γ_i is in the interval $(0, 1)$.

As with ART, finding the best relaxation parameters is a bit of an art.

11.4.2 Cross-Entropy

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

$$KL(a, b) = a \log \frac{a}{b} + b - a, \tag{11.22}$$

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

$$KL(x,z) = \sum_{j=1}^{J} KL(x_j, z_j). \tag{11.23}$$

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.

11.4.3 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 11.3 *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,

$$SE(x) = \sum_{j=1}^{J} x_j \log x_j - x_j. \tag{11.24}$$

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 12

A Tale of Two Algorithms

12.1 The Two Algorithms

The algorithms we shall consider are the expectation maximization maximum likelihood method (EMML) and the simultaneous multiplicative algebraic reconstruction technique (SMART). When $\mathbf{y} = P\mathbf{x}$ has nonnegative solutions, both algorithms produce such a solution. In general, the EMML gives a nonnegative minimizer of $KL(\mathbf{y}, P\mathbf{x})$, while the SMART minimizes $KL(P\mathbf{x}, \mathbf{y})$ over nonnegative $\mathbf{x}$.

For both algorithms we begin with an arbitrary positive vector $\mathbf{x}^0$. The iterative step for the EMML method is

$$x_j^{k+1} = (\mathbf{x}^k)'_j = x_j^k \sum_{i=1}^{I} P_{ij} \frac{y_i}{(P\mathbf{x}^k)_i}. \tag{12.1}$$

The iterative step for the SMART is

$$x_j^{m+1} = (\mathbf{x}^m)''_j = x_j^m \exp\Big(\sum_{i=1}^{I} P_{ij} \log \frac{y_i}{(P\mathbf{x}^m)_i}\Big). \tag{12.2}$$

Note that, to avoid confusion, we use k for the iteration number of the EMML and m for the SMART.

12.2 Background

The expectation maximization maximum likelihood method (EMML) has been the subject of much attention in the medical-imaging literature over the past decade. Statisticians like it because it is based on the well-studied principle of likelihood maximization for parameter estimation. Physicists

like it because, unlike its competition, filtered back-projection, it permits the inclusion of sophisticated models of the physical situation. Mathematicians like it because it can be derived from iterative optimization theory. Physicians like it because the images are often better than those produced by other means. No method is perfect, however, and the EMML suffers from sensitivity to noise and slow rate of convergence. Research is ongoing to find faster and less sensitive versions of this algorithm.

Another class of iterative algorithms was introduced into medical imaging by Gordon et al. in [138]. These include the *algebraic reconstruction technique* (ART) and its multiplicative version, MART. These methods were derived by viewing image reconstruction as solving systems of linear equations, possibly subject to constraints, such as positivity. The *simultaneous* MART (SMART) [98, 218] is a variant of MART that uses all the data at each step of the iteration.

Although the EMML and SMART algorithms have quite different histories and are not typically considered together they are closely related [48, 49]. In this chapter we examine these two algorithms in tandem, following [50]. Forging a link between the EMML and SMART led to a better understanding of both of these algorithms and to new results. The proof of convergence of the SMART in the inconsistent case [48] was based on the analogous proof for the EMML [242], while discovery of the faster version of the EMML, the *rescaled block-iterative* EMML (RBI-EMML) [51] came from studying the analogous block-iterative version of SMART [82]. The proofs we give here are elementary and rely mainly on easily established properties of the cross-entropy or Kullback-Leibler distance.

12.3 The Kullback-Leibler Distance

The The Kullback-Leibler distance $KL(\mathbf{x}, \mathbf{z})$ is defined for nonnegative vectors $\mathbf{x}$ and $\mathbf{z}$ by Equations (11.22) and (11.23). Clearly, the KL distance has the property $KL(c\mathbf{x}, c\mathbf{z}) = cKL(\mathbf{x}, \mathbf{z})$ for all positive scalars c.

Exercise 12.1 *Let* $z_+ = \sum_{j=1}^{J} z_j > 0$. *Then*

$$KL(\mathbf{x}, \mathbf{z}) = KL(x_+, z_+) + KL(\mathbf{x}, (x_+/z_+)\mathbf{z}). \tag{12.3}$$

As we shall see, the KL distance mimics the ordinary Euclidean distance in several ways that make it particularly useful in designing optimization algorithms.

12.4 The Alternating Minimization Paradigm

Let P be an I by J matrix with entries $P_{ij} \geq 0$, such that, for each $j = 1, ..., J$, we have $s_j = \sum_{i=1}^{I} P_{ij} > 0$. Let $\mathbf{y} = (y_1, ..., y_I)^T$ with $y_i > 0$ for each i. We shall assume throughout this chapter that $s_j = 1$ for each j. If this is not the case initially, we replace x_j with $x_j s_j$ and P_{ij} with P_{ij}/s_j; the quantities $(P\mathbf{x})_i$ are unchanged.

For each nonnegative vector $\mathbf{x}$ for which $(P\mathbf{x})_i = \sum_{j=1}^{J} P_{ij} x_j > 0$, let $r(\mathbf{x}) = \{r(\mathbf{x})_{ij}\}$ and $q(\mathbf{x}) = \{q(\mathbf{x})_{ij}\}$ be the I by J arrays with entries

$$r(\mathbf{x})_{ij} = x_j P_{ij} \frac{y_i}{(P\mathbf{x})_i}$$

and

$$q(\mathbf{x})_{ij} = x_j P_{ij}.$$

The KL distances

$$KL(r(\mathbf{x}), q(\mathbf{z})) = \sum_{i=1}^{I} \sum_{j=i}^{J} KL(r(\mathbf{x})_{ij}, q(\mathbf{z})_{ij})$$

and

$$KL(q(\mathbf{x}), r(\mathbf{z})) = \sum_{i=1}^{I} \sum_{j=1}^{J} KL(q(\mathbf{x})_{ij}, r(\mathbf{z})_{ij})$$

will play important roles in the discussion that follows. Note that if there is nonnegative $\mathbf{x}$ with $r(\mathbf{x}) = q(\mathbf{x})$ then $\mathbf{y} = P\mathbf{x}$.

12.4.1 Some Pythagorean Identities Involving the KL Distance

The iterative algorithms we discuss in this chapter are derived using the principle of *alternating minimization*, according to which the distances $KL(r(\mathbf{x}), q(\mathbf{z}))$ and $KL(q(\mathbf{x}), r(\mathbf{z}))$ are minimized, first with respect to the variable $\mathbf{x}$ and then with respect to the variable $\mathbf{z}$. Although the KL distance is not Euclidean, and, in particular, not even symmetric, there are analogues of Pythagoras' theorem that play important roles in the convergence proofs.

Exercise 12.2 *Establish the following Pythagorean identities:*

$$KL(r(\mathbf{x}), q(\mathbf{z})) = KL(r(\mathbf{z}), q(\mathbf{z})) + KL(r(\mathbf{x}), r(\mathbf{z})); \tag{12.4}$$

$$KL(r(\mathbf{x}), q(\mathbf{z})) = KL(r(\mathbf{x}), q(\mathbf{x}')) + KL(\mathbf{x}', \mathbf{z}), \tag{12.5}$$

for

$$x'_j = x_j \sum_{i=1}^{I} P_{ij} \frac{y_i}{(P\mathbf{x})_i}; \tag{12.6}$$

$$KL(q(\mathbf{x}), r(\mathbf{z})) = KL(q(\mathbf{x}), r(\mathbf{x})) + KL(\mathbf{x}, \mathbf{z}) - KL(P\mathbf{x}, P\mathbf{z}); \tag{12.7}$$

$$KL(q(\mathbf{x}), r(\mathbf{z})) = KL(q(\mathbf{z}''), r(\mathbf{z})) + KL(\mathbf{x}, \mathbf{z}''), \tag{12.8}$$

for

$$z''_j = z_j \exp(\sum_{i=1}^{I} P_{ij} \log \frac{y_i}{(P\mathbf{z})_i}). \tag{12.9}$$

Note that it follows from Equation (12.3) that $KL(\mathbf{x}, \mathbf{z}) - KL(P\mathbf{x}, P\mathbf{z}) \geq 0$.

12.4.2 Convergence of the SMART and EMML

We shall prove convergence of the SMART and EMML algorithms through a series of exercises.

Exercise 12.3 *Show that, for* $\{\mathbf{x}^k\}$ *given by Equation (12.1),* $\{KL(\mathbf{y}, P\mathbf{x}^k)\}$ *is decreasing and* $\{KL(\mathbf{x}^{k+1}, \mathbf{x}^k)\} \rightarrow 0$. *Show that, for* $\{\mathbf{x}^m\}$ *given by Equation (12.2),* $\{KL(P\mathbf{x}^m, \mathbf{y})\}$ *is decreasing and* $\{KL(\mathbf{x}^m, \mathbf{x}^{m+1})\} \rightarrow 0$.

Hint: Use $KL(r(\mathbf{x}), q(\mathbf{x})) = KL(\mathbf{y}, P\mathbf{x})$, $KL(q(\mathbf{x}), r(\mathbf{x})) = KL(P\mathbf{x}, \mathbf{y})$, and the Pythagorean identities.

Exercise 12.4 *Show that the EMML sequence* $\{\mathbf{x}^k\}$ *is bounded by showing*

$$\sum_{j=1}^{J} x_j^k = \sum_{i=1}^{I} y_i.$$

Show that the SMART sequence $\{\mathbf{x}^m\}$ *is bounded by showing that*

$$\sum_{j=1}^{J} x_j^m \leq \sum_{i=1}^{I} y_i.$$

Exercise 12.5 *Show that* $(\mathbf{x}^*)' = \mathbf{x}^*$ *for any cluster point* $\mathbf{x}^*$ *of the EMML sequence* $\{\mathbf{x}^k\}$ *and that* $(\mathbf{x}^*)'' = \mathbf{x}^*$ *for any cluster point* $\mathbf{x}^*$ *of the SMART sequence* $\{\mathbf{x}^m\}$.

Hint: Use the facts that $\{KL(\mathbf{x}^{k+1}, \mathbf{x}^k)\} \to 0$ and $\{KL(\mathbf{x}^m, \mathbf{x}^{m+1})\} \to 0$.

Exercise 12.6 *Let* $\hat{\mathbf{x}}$ *and* $\tilde{\mathbf{x}}$ *minimize* $KL(\mathbf{y}, P\mathbf{x})$ *and* $KL(P\mathbf{x}, \mathbf{y})$, *respectively, over all* $\mathbf{x} \geq \mathbf{0}$. *Then,* $(\hat{\mathbf{x}})' = \hat{\mathbf{x}}$ *and* $(\tilde{\mathbf{x}})'' = \tilde{\mathbf{x}}$.

Hint: Apply Pythagorean identities to $KL(r(\hat{\mathbf{x}}), q(\hat{\mathbf{x}}))$ and $KL(q(\tilde{\mathbf{x}}), r(\tilde{\mathbf{x}}))$.

Note that, because of convexity properties of the KL distance, even if the minimizers $\hat{\mathbf{x}}$ and $\tilde{\mathbf{x}}$ are not unique, the vectors $P\hat{\mathbf{x}}$ and $P\tilde{\mathbf{x}}$ are unique.

Exercise 12.7 *For the EMML sequence* $\{\mathbf{x}^k\}$ *with cluster point* $\mathbf{x}^*$ *and* $\hat{\mathbf{x}}$ *as defined previously, we have the double inequality*

$$KL(\hat{\mathbf{x}}, \mathbf{x}^k) \geq KL(r(\hat{\mathbf{x}}), r(\mathbf{x}^k)) \geq KL(\hat{\mathbf{x}}, \mathbf{x}^{k+1}), \tag{12.10}$$

from which we conclude that the sequence $\{KL(\hat{\mathbf{x}}, \mathbf{x}^k)\}$ *is decreasing and* $KL(\hat{\mathbf{x}}, \mathbf{x}^*) < +\infty$.

Hint: For the first inequality calculate $KL(r(\hat{\mathbf{x}}), q(\mathbf{x}^k))$ in two ways. For the second one, use $(\mathbf{x})'_j = \sum_{i=1}^I r(\mathbf{x})_{ij}$ and Exercise 12.1.

Exercise 12.8 *Show that, for the SMART sequence* $\{\mathbf{x}^m\}$ *with cluster point* $\mathbf{x}^*$ *and* $\tilde{\mathbf{x}}$ *as defined previously, we have*

$$KL(\tilde{\mathbf{x}}, \mathbf{x}^m) - KL(\tilde{\mathbf{x}}, \mathbf{x}^{m+1}) = KL(P\mathbf{x}^{m+1}, \mathbf{y}) - KL(P\tilde{\mathbf{x}}, \mathbf{y}) +$$

$$KL(P\tilde{\mathbf{x}}, P\mathbf{x}^m) + KL(\mathbf{x}^{m+1}, \mathbf{x}^m) - KL(P\mathbf{x}^{m+1}, P\mathbf{x}^m), \tag{12.11}$$

and so $KL(P\tilde{\mathbf{x}}, P\mathbf{x}^*) = 0$, *the sequence* $\{KL(\tilde{\mathbf{x}}, \mathbf{x}^m)\}$ *is decreasing and* $KL(\tilde{\mathbf{x}}, \mathbf{x}^*) < +\infty$.

Hint: Expand $KL(q(\tilde{\mathbf{x}}), r(\mathbf{x}^m))$ using the Pythagorean identities.

Exercise 12.9 *For $\mathbf{x}^*$ a cluster point of the EMML sequence $\{\mathbf{x}^k\}$ we have $KL(\mathbf{y}, P\mathbf{x}^*) = KL(\mathbf{y}, P\hat{\mathbf{x}})$. Therefore, $\mathbf{x}^*$ is a nonnegative minimizer of $KL(\mathbf{y}, P\mathbf{x})$. Consequently, the sequence $\{KL(\mathbf{x}^*, \mathbf{x}^k)\}$ converges to zero, and so $\{\mathbf{x}^k\} \to \mathbf{x}^*$.*

Hint: Use the double inequality of Equation (12.10) and $KL(r(\hat{\mathbf{x}}), q(\mathbf{x}^*))$.

Exercise 12.10 *For $\mathbf{x}^*$ a cluster point of the SMART sequence $\{\mathbf{x}^m\}$ we have $KL(P\mathbf{x}^*, \mathbf{y}) = KL(P\tilde{\mathbf{x}}, \mathbf{y})$. Therefore, $\mathbf{x}^*$ is a nonnegative minimizer of $KL(P\mathbf{x}, \mathbf{y})$. Consequently, the sequence $\{KL(\mathbf{x}^*, \mathbf{x}^m)\}$ converges to zero, and so $\{\mathbf{x}^m\} \to \mathbf{x}^*$. Moreover,*

$$KL(\tilde{\mathbf{x}}, \mathbf{x}^0) \geq KL(\mathbf{x}^*, \mathbf{x}^0)$$

for all $\tilde{\mathbf{x}}$ as before.

Hints: Use Exercise 12.8. For the final assertion use the fact that the difference $KL(\tilde{\mathbf{x}}, \mathbf{x}^m) - KL(\tilde{\mathbf{x}}, \mathbf{x}^{m+1})$ is independent of the choice of $\tilde{\mathbf{x}}$, since it depends only on $P\mathbf{x}^* = P\tilde{\mathbf{x}}$. Now sum over the index m.

Both the EMML and the SMART algorithms are slow to converge. For that reason attention has shifted, in recent years, to *block-iterative* versions of these algorithms. We take up that topic in a later chapter.

Chapter 13

Block-Iterative Methods

Image reconstruction problems in tomography are often formulated as statistical likelihood maximization problems in which the pixel values of the desired image play the role of parameters. Iterative algorithms based on cross-entropy minimization, such as the *expectation maximization maximum likelihood* (EMML) method and the *simultaneous multiplicative algebraic reconstruction technique* (SMART) can be used to solve such problems. Because the EMML and SMART are slow to converge for large amounts of data typical in imaging problems, acceleration of the algorithms using blocks of data or ordered subsets has become popular. There are a number of different ways to formulate these block-iterative versions of EMML and SMART, involving the choice of certain normalization and regularization parameters. These methods are not faster merely because they are block-iterative; the correct choice of the parameters is crucial. The purpose of this chapter is to discuss these different formulations in detail sufficient to reveal the precise roles played by the parameters and to guide the user in choosing them.

It is not obvious, nor, in fact, is it even true, that using block-iterative methods will accelerate convergence. To better understand the connection between the use of blocks and acceleration, we begin with a discussion of the ART algorithm and its simultaneous versions, the Landweber algorithm and more particularly, Cimmino's method.

13.1 The ART and its Simultaneous Versions

In this section we let $Ax = b$ denote any real system of I linear equations in J unknowns. For each $i = 1, ..., I$ denote by H_i the hyperplane associated with the ith equation, that is,

$$H_i = \{x | (Ax)_i = b_i\},$$

and P_i the orthogonal projection operator onto H_i, that is, for every vector z, $P_i z$ is the vector in H_i closest to z. We can write $P_i z$ explicitly; we have

$$P_i z = z + (b_i - (Az)_i)a^i,$$

where a^i is the ith column of the matrix A^T, which we shall assume has been normalized to have $||a^i|| = 1$.

13.1.1 The ART

For $k = 0, 1, ...$ and $i = i(k) = k(\mathrm{mod}\, I) + 1$, the ART iterative step is

$$x^{k+1} = P_i x^k = x^k + (b_i - (Ax^k)_i)a^i.$$

The ART operates by projecting the current vector onto the next hyperplane and cycling through the hyperplanes repeatedly. The ART uses only one equation at each step of the iteration.

Suppose that $\hat{x}$ is a solution of $Ax = b$, so that $A\hat{x} = b$. Each step of the ART gets us closer to $\hat{x}$, as the following calculations show.

We begin by calculating $||\hat{x} - x^{k+1}||^2$. We use

$$||\hat{x} - x^{k+1}||^2 = \langle \hat{x} - x^{k+1}, \hat{x} - x^{k+1} \rangle$$

and the definition of x^{k+1} to get

$$||\hat{x} - x^{k+1}||^2 = ||\hat{x} - x^k||^2 - 2\langle \hat{x} - x^k, (b_i - (Ax^k)_i)a^i \rangle + \langle (b_i - (Ax^k)_i)a^i, (b_i - (Ax^k)_i)a^i \rangle$$

$$= ||\hat{x} - x^k||^2 - 2(b_i - (Ax^k)_i)\langle \hat{x} - x^k, a^i \rangle + (b_i - (Ax^k)_i)^2$$

$$= ||\hat{x} - x^k||^2 - 2(b_i - (Ax^k)_i)^2 + (b_i - (Ax^k)_i)^2 = ||\hat{x} - x^k||^2 - (b_i - (Ax^k)_i)^2.$$

Therefore, we find that

$$||\hat{x} - x^k||^2 - ||\hat{x} - x^{k+1}||^2 = (b_i - (Ax^k)_i)^2. \tag{13.1}$$

Consequently, we know that

$$||\hat{x} - x^k||^2 \geq ||\hat{x} - x^{k+1}||^2.$$

It will help us later to know that

$$||\hat{x} - x^0||^2 - ||\hat{x} - x^I||^2 = \sum_{i=1}^{I} (b_i - (Ax^{i-1})_i)^2. \tag{13.2}$$

This measures how much closer to $\hat{x}$ we are after we have used all the equations one time.

There is one other consideration concerning the ART. From Equation (13.2) we see that it is helpful to have the quantities $(b_i - (Ax^{i-1})_i)^2$ large;

if the equations are ordered in such a way that these quantities are not large, then the ART will not converge as quickly as it may otherwise do. This can easily happen if the equations correspond to discrete line integrals through the object and the lines are ordered so that each line is close to the previous line. Ordering the lines randomly, or in any way that avoids unfortunate ordering, greatly improves convergence speed [152].

Relaxation also helps to speed up the convergence of ART [222]. A relaxed version of ART has the following iterative step:

$$x^{k+1} = x^k + \beta(b_i - (Ax^k)_i)a^i,$$

where $0 < \beta \leq 1$.

13.1.2 The Landweber Algorithm and Cimmino's Method

As we just saw, the ART uses one equation at a time and, at each step of the iteration, projects orthogonally onto the hyperplane associated with the next equation. A *simultaneous* version of ART uses all the equations at each step, projecting orthogonally onto all the hyperplanes and averaging the result. This is Cimmino's method, and the iterative step is

$$x^{k+1} = x^k + \frac{1}{I}\sum_{i=1}^{I}(b_i - (Ax^k)_i)a^i = x^k + \frac{1}{I}A^T(b - Ax^k),$$

where, as previously, we assume that $||a^i|| = 1$ for all i. A more general iterative algorithm is the Landweber algorithm, with the iterative step

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

for convergence of this algorithm we need $0 \leq \gamma \leq 2/\rho(A^TA)$, where $\rho(A^TA)$ denotes the largest eigenvalue of the matrix A^TA. Since $||a^i|| = 1$ for all i, it follows that the trace of the matrix AA^T is I, which is also the trace of the matrix A^TA; since the trace of A^TA is also the sum of the eigenvalues of A^TA, it follows that the choice of $\gamma = \frac{1}{I}$ in Cimmino's method is acceptable.

Now let us calculate how much closer to $\hat{x}$ we get as we take one step of the Landweber iteration. We have

$$||\hat{x}-x^{k+1}||^2 = ||\hat{x}-x^k||^2 - 2\gamma\langle\hat{x}-x^k, A^T(b-Ax^k)\rangle + \gamma^2\langle A^T(b-Ax^k), A^T(b-Ax^k)\rangle.$$

From the inequality (30.1) in our earlier discussion of eigenvectors and eigenvalues in optimization, we know that, for any matrix B, we have

$$||Bx||^2 \leq \rho(B^TB)||x||^2.$$

Therefore,

$$\langle A^T(b - Ax^k), A^T(b - Ax^k)\rangle = ||A^T(b - Ax^k)||^2 \leq \rho(A^TA)||b - Ax^k||^2.$$

Using

$$\langle \hat{x}-x^k, A^T(b-Ax^k)\rangle = \langle A(\hat{x}-x^k), b-Ax^k\rangle = \langle b-Ax^k, b-Ax^k\rangle = ||b-Ax^k||^2,$$

we find that

$$||\hat{x} - x^k||^2 - ||\hat{x} - x^{k+1}||^2 \geq (2\gamma - \gamma^2\rho(A^TA))||b - Ax^k||^2.$$

For $0 < \gamma < \frac{2}{\rho(A^TA)}$ the sequence $\{||\hat{x} - x^k||^2\}$ is decreasing. If we take $\gamma = \frac{1}{\rho(A^TA)}$ we have

$$||\hat{x} - x^k||^2 - ||\hat{x} - x^{k+1}||^2 \geq \frac{1}{\rho(A^TA)}||b - Ax^k||^2. \tag{13.3}$$

In the case of Cimmino's method, we have $\gamma = \frac{1}{I}$, so that

$$||\hat{x} - x^k||^2 - ||\hat{x} - x^{k+1}||^2 \geq \frac{1}{I}||b - Ax^k||^2. \tag{13.4}$$

Using Equation (13.2) and the inequality in (13.4), we can make a rough comparison between ART and Cimmino's method, with respect to how much closer to $\hat{x}$ we get as we pass through all the equations one time. The two quantities

$$\sum_{i=1}^{I}(b_i - (Ax^{i-1})_i)^2$$

from Equation (13.2) and

$$||b - Ax^k||^2$$

from the inequality in (13.4) are comparable, in that both sums are over $i = 1, ..., I$, even though what is being summed is not the same in both cases. In image reconstruction I is quite large and the most important thing in such comparisons is the range of the summation index, so long as what is being summed is roughly comparable. However, notice that in the inequality in (13.4) the right side also has a factor of $\frac{1}{I}$. This tells us that, roughly speaking, one pass through all the equations using ART improves the squared distance to $\hat{x}$ by a factor of I, compared to using all the equations in one step of Cimmino's method, even though the amount of calculation is about the same.

Because the Landweber algorithm permits other choices for the parameter γ, there is hope that we may obtain better results with $\gamma \neq \frac{1}{I}$. The inequality

$$0 < \gamma < \frac{2}{\rho(A^TA)}$$

suggests using $\gamma = \frac{1}{\rho(A^TA)}$, which means that it would help to have a decent estimate of $\rho(A^TA)$; the estimate used in Cimmino's method is $\rho(A^TA) = I$, which is usually much too large. As a result, the choice of $\gamma = \frac{1}{I}$ means that we are taking unnecessarily small steps at each iteration. A smaller upper bound for $\rho(A^TA)$ would allow us to take bigger steps each time, and therefore, getting close to $\hat{x}$ sooner.

In many image processing applications, such as tomography, the matrix A is *sparse*, which means that most of the entries of A are zero. In the tomography problems for example, the number of non-zero entries of A is usually on the order of $\sqrt{J}$; since I and J are usually roughly comparable, this means that A has about $\sqrt{I}$ non-zero entries. In the appendix on matrix theory we obtain an upper bound estimate for $\rho(A^TA)$ that is particularly useful when A is sparse. Suppose that all the rows of A have length one. Let s be the largest number of non-zero entries in any column of A. Then $\rho(A^TA)$ does not exceed s. Notice that this estimate does not require us to calculate the matrix A^TA and makes use of the sparse nature of A; the matrix A^TA need not be sparse, and would be time-consuming to calculate in practice, anyway.

If, for the sparse cases, we take $\rho(A^TA)$ to be approximately $\sqrt{I}$, and choose $\gamma = \frac{1}{\sqrt{I}}$, we find that we have replaced the factor $\frac{1}{I}$ in the inequality (13.4) with the much larger factor $\frac{1}{\sqrt{I}}$, which then improves the rate of convergence. However, the ART is still faster by, roughly, a factor of $\sqrt{I}$.

13.1.3 Block-Iterative ART

The ART uses only one equation at a time, while the Landweber algorithm uses all the equations at each step of the iteration. It is sometimes convenient to take a middle course, and use some, but not all, equations at each step of the iteration. The collection of equations to be used together constitute a *block*. Such methods are called *block-iterative* or *ordered-subset* methods. Generally speaking, when unfortunate ordering of the blocks and selection of equations within each block are avoided, and the parameters well chosen, these block-iterative methods converge faster than the Cimmino algorithm by roughly a factor of the number of blocks.

We turn now to the iterative algorithms that are based on the KL distance. For these algorithms as well, we find that using block-iterative methods and choosing the parameters carefully, we can improve convergence by roughly the number of blocks, with respect to the simultaneous EMML and SMART methods.

13.2 Overview of KL-based methods

The algorithms we discuss here have interesting histories, which we sketch in this section.

13.2.1 The SMART and its variants

Like the ART, the MART has a simultaneous version, called the SMART. Like MART, SMART applies only to nonnegative systems of equations $Ax = b$. Unlike MART, SMART is a simultaneous algorithm that uses all equations in each step of the iteration. The SMART was discovered in 1972, independently, by Darroch and Ratcliff, working in statistics, [98] and by Schmidlin [218] in medical imaging; neither work makes reference to MART. Darroch and Ratcliff do consider block-iterative versions of their algorithm, in which only some of the equations are used at each step, but their convergence proof involves unnecessary restrictions on the system matrix. Censor and Segman [82] seem to be the first to present the SMART and its block-iterative variants explicitly as generalizations of MART.

13.2.2 The EMML and its variants

The *expectation maximization maximum likelihood* (EMML) method turns out to be closely related to the SMART, although it has quite a different history. The EMML algorithm we discuss here is actually a special case of a more general approach to likelihood maximization, usually called the EM algorithm [100]; the book by McLachnan and Krishnan [187] is a good source for the history of this more general algorithm.

It was noticed by Rockmore and Macovski [215] that the image reconstruction problems posed by medical tomography could be formulated as statistical parameter estimation problems. Following up on this idea, Shepp and Vardi [221] suggested the use of the EM algorithm for solving the reconstruction problem in emission tomography. In [174], Lange and Carson presented an EM-type iterative method for transmission tomographic image reconstruction, and pointed out a gap in the convergence proof given in [221] for the emission case. In [242], Vardi, Shepp and Kaufman repaired the earlier proof, relying on techniques due to Csiszár and Tusnády [96]. In [175] Lange, Bahn and Little improved the transmission and emission algorithms, by including regularization to reduce the effects of noise. The question of uniqueness of the solution in the inconsistent case was resolved in [48].

The MART and SMART were initially designed to apply to consistent systems of equations. Darroch and Ratcliff did not consider what happens in the inconsistent case, in which the system of equations has no nonnegative solutions; this issue was resolved in [48], where it was shown that

the SMART converges to a non-negative minimizer of the Kullback-Leibler distance $KL(Ax, b)$. The EMML, as a statistical parameter estimation technique, was not originally thought to be connected to any system of linear equations. In [48] it was shown that the EMML leads to a non-negative minimizer of the Kullback-Leibler distance $KL(b, Ax)$, thereby exhibiting a close connection between the SMART and the EMML methods. Consequently, when the non-negative system of linear equations $Ax = b$ has a non-negative solution, the EMML converges to such a solution.

13.2.3 Block-iterative Versions of SMART and EMML

As we have seen, Darroch and Ratcliff included what are now called block-iterative versions of SMART in their original paper [98]. Censor and Segman [82] viewed SMART and its block-iterative versions as natural extension of the MART. Consequently, block-iterative variants of SMART have been around for some time. The story with the EMML is quite different.

The paper of Holte, Schmidlin, *et al.* [155] compares the performance of Schmidlin's method of [218] with the EMML algorithm. Almost as an aside, they notice the accelerating effect of what they call *projection interleaving*, that is, the use of blocks. This paper contains no explicit formulas, however, and presents no theory, so one can only make educated guesses as to the precise iterative methods employed. Somewhat later, Hudson, Hutton and Larkin [156, 157] observed that the EMML can be significantly accelerated if, at each step, one employs only some of the data. They referred to this approach as the *ordered subset* EM method (OSEM). They gave a proof of convergence of the OSEM, for the consistent case. The proof relied on a fairly restrictive relationship between the matrix A and the choice of blocks, called *subset balance*. In [51] a revised version of the OSEM, called the *rescaled block-iterative* EMML (RBI-EMML), was shown to converge, in the consistent case, regardless of the choice of blocks.

13.2.4 Basic assumptions

Methods based on cross-entropy, such as the MART, SMART, EMML and all block-iterative versions of these algorithms apply to nonnegative systems that we denote by $Ax = b$, where b is a vector of positive entries, A is a matrix with entries $A_{ij} \geq 0$ such that for each j the sum $s_j = \sum_{i=1}^{I} A_{ij}$ is positive and we seek a solution x with nonnegative entries. If no non-negative x satisfies $b = Ax$ we say the system is *inconsistent*.

Simultaneous iterative algorithms employ all of the equations at each step of the iteration; block-iterative methods do not. For the latter methods we assume that the index set $\{i = 1, ..., I\}$ is the (not necessarily disjoint) union of the N sets or *blocks* B_n, $n = 1, ..., N$. We shall require that $s_{nj} = \sum_{i \in B_n} A_{ij} > 0$ for each n and each j. Block-iterative methods like

ART and MART for which each block consists of precisely one element are called *row-action* or *sequential* methods. We begin our discussion with the SMART and the EMML method.

13.3 The SMART and the EMML method

Both the SMART and the EMML method provide a solution of $b = Ax$ when such exist and (distinct) approximate solutions in the inconsistent case. The SMART algorithm is the following:

Algorithm 13.1 (SMART) *Let x^0 be an arbitrary positive vector. For $k = 0, 1, ...$ let*

$$x_j^{k+1} = x_j^k \exp\Big(s_j^{-1} \sum_{i=1}^{I} A_{ij} \log \frac{b_i}{(Ax^k)_i}\Big). \tag{13.5}$$

The exponential and logarithm in the SMART iterative step are computationally expensive. The EMML method is similar to the SMART, but somewhat less costly to compute.

Algorithm 13.2 (EMML) *Let x^0 be an arbitrary positive vector. For $k = 0, 1, ...$ let*

$$x_j^{k+1} = x_j^k s_j^{-1} \sum_{i=1}^{I} A_{ij} \frac{b_i}{(Ax^k)_i}. \tag{13.6}$$

The main results concerning the SMART are given by the following theorem.

Theorem 13.1 *In the consistent case the SMART converges to the unique nonnegative solution of $b = Ax$ for which the distance $\sum_{j=1}^{J} s_j KL(x_j, x_j^0)$ is minimized. In the inconsistent case it converges to the unique nonnegative minimizer of the distance $KL(Ax, y)$ for which $\sum_{j=1}^{J} s_j KL(x_j, x_j^0)$ is minimized; if A and every matrix derived from A by deleting columns has full rank then there is a unique nonnegative minimizer of $KL(Ax, y)$ and at most $I - 1$ of its entries are nonzero.*

For the EMML method the main results are the following.

Theorem 13.2 *In the consistent case the EMML algorithm converges to nonnegative solution of $b = Ax$. In the inconsistent case it converges to a nonnegative minimizer of the distance $KL(y, Ax)$; if A and every matrix derived from A by deleting columns has full rank then there is a unique nonnegative minimizer of $KL(y, Ax)$ and at most $I - 1$ of its entries are nonzero.*

In the consistent case there may be multiple nonnegative solutions and the one obtained by the EMML algorithm will depend on the starting vector x^0; how it depends on x^0 is an open question.

These theorems are special cases of more general results on block-iterative methods that we shall prove later in this chapter.

Both the EMML and SMART are related to likelihood maximization. Minimizing the function $KL(y, Ax)$ is equivalent to maximizing the likelihood when the b_i are taken to be measurements of independent Poisson random variables having means $(Ax)_i$. The entries of x are the parameters to be determined. This situation arises in emission tomography. So the EMML is a likelihood maximizer, as its name suggests.

The connection between SMART and likelihood maximization is a bit more convoluted. Suppose that $s_j = 1$ for each j. The solution of $b = Ax$ for which $KL(x, x^0)$ is minimized necessarily has the form

$$x_j = x_j^0 \exp\Big(\sum_{i=1}^{I} A_{ij}\lambda_i\Big) \tag{13.7}$$

for some vector λ with entries λ_i. This *log linear* form also arises in transmission tomography, where it is natural to assume that $s_j = 1$ for each j and $\lambda_i \leq 0$ for each i. We have the following lemma that helps to connect the SMART algorithm with the transmission tomography problem:

Lemma 13.1 *Minimizing $KL(d, x)$ over x as in Equation (13.7) is equivalent to minimizing $KL(x, x^0)$, subject to $Ax = Ad$.*

The solution to the latter problem can be obtained using the SMART.

With $x_+ = \sum_{j=1}^{J} x_j$ the vector A with entries $p_j = x_j/x_+$ is a probability vector. Let $d = (d_1, ..., d_J)^T$ be a vector whose entries are nonnegative integers, with $K = \sum_{j=1}^{J} d_j$. Suppose that, for each j, p_j is the probability of index j and d_j is the number of times index j was chosen in K trials. The likelihood function of the parameters λ_i is

$$L(\lambda) = \prod_{j=1}^{J} p_j^{d_j} \tag{13.8}$$

so that the log-likelihood function is

$$LL(\lambda) = \sum_{j=1}^{J} d_j \log p_j. \tag{13.9}$$

Since A is a probability vector, maximizing $L(\lambda)$ is equivalent to minimizing $KL(d, p)$ with respect to λ, which, according to the lemma above, can be solved using SMART. In fact, since all of the block-iterative versions

of SMART have the same limit whenever they have the same starting vector, any of these methods can be used to solve this maximum likelihood problem. In the case of transmission tomography the λ_i must be non-positive, so if SMART is to be used, some modification is needed to obtain such a solution.

Those who have used the SMART or the EMML on sizable problems have certainly noticed that they are both slow to converge. An important issue, therefore, is how to accelerate convergence. One popular method is through the use of *block-iterative* (or *ordered subset*) methods.

13.4 Ordered-Subset Versions

To illustrate block-iterative methods and to motivate our subsequent discussion we consider now the *ordered subset* EM algorithm (OSEM), which is a popular technique in some areas of medical imaging, as well as an analogous version of SMART, which we shall call here the OSSMART. The OSEM is now used quite frequently in tomographic image reconstruction, where it is acknowledged to produce usable images significantly faster then EMML. From a theoretical perspective both OSEM and OSSMART are incorrect. How to correct them is the subject of much that follows here.

The idea behind the OSEM (OSSMART) is simple: the iteration looks very much like the EMML (SMART), but at each step of the iteration the summations are taken only over the current block. The blocks are processed cyclically.

The OSEM iteration is the following: for $k = 0, 1, ...$ and $n = k(\mathrm{mod}\, N) + 1$, having found x^k let

OSEM:

$$x_j^{k+1} = x_j^k s_{nj}^{-1} \sum_{i \in B_n} A_{ij} \frac{b_i}{(Ax^k)_i}. \tag{13.10}$$

The OSSMART has the following iterative step:

OSSMART

$$x_j^{k+1} = x_j^k \exp\Big(s_{nj}^{-1} \sum_{i \in B_n} A_{ij} \log \frac{b_i}{(Ax^k)_i}\Big). \tag{13.11}$$

In general we do not expect block-iterative algorithms to converge in the inconsistent case, but to exhibit *subsequential convergence* to a *limit cycle*, as we shall discuss later. We do, however, want them to converge to a solution in the consistent case; the OSEM and OSSMART fail to do this except when the matrix A and the set of blocks $\{B_n,\, n = 1, ..., N\}$ satisfy

the condition known as *subset balance*, which means that the sums s_{nj} depend only on j and not on n. While this may be approximately valid in some special cases, it is overly restrictive, eliminating, for example, almost every set of blocks whose cardinalities are not all the same. When the OSEM does well in practice in medical imaging it is probably because the N is not large and only a few iterations are carried out.

The experience with the OSEM was encouraging, however, and strongly suggested that an equally fast, but mathematically correct, block-iterative version of EMML was to be had; this is the *rescaled block-iterative* EMML (RBI-EMML). Both RBI-EMML and an analogous corrected version of OSSMART, the RBI-SMART, provide fast convergence to a solution in the consistent case, for any choice of blocks.

13.5 The RBI-SMART

We turn next to the block-iterative versions of the SMART, which we shall denote BI-SMART. These methods were known prior to the discovery of RBI-EMML and played an important role in that discovery; the importance of rescaling for acceleration was apparently not appreciated, however.

We start by considering a formulation of BI-SMART that is general enough to include all of the variants we wish to discuss. As we shall see, this formulation is too general and will need to be restricted in certain ways to obtain convergence. Let the iterative step be

$$x_j^{k+1} = x_j^k \exp\Big(\beta_{nj} \sum_{i \in B_n} \alpha_{ni} A_{ij} \log\Big(\frac{b_i}{(Ax^k)_i}\Big)\Big), \tag{13.12}$$

for $j = 1, 2, ..., J$, $n = k(\mathrm{mod}\, N) + 1$ and β_{nj} and α_{ni} positive. As we shall see, our convergence proof will require that β_{nj} be separable, that is, $b_{nj} = \gamma_j \delta_n$ for each j and n and that

$$\gamma_j \delta_n \sigma_{nj} \leq 1, \tag{13.13}$$

for $\sigma_{nj} = \sum_{i \in B_n} \alpha_{ni} A_{ij}$. With these conditions satisfied we have the following result.

Theorem 13.3 *Let x be a nonnegative solution of $b = Ax$. For any positive vector x^0 and any collection of blocks $\{B_n, n = 1, ..., N\}$ the sequence $\{x^k\}$ given by Equation (13.12) converges to the unique solution of $b = Ax$ for which the weighted cross-entropy $\sum_{j=1}^J \gamma_j^{-1} KL(x_j, x_j^0)$ is minimized.*

The inequality in the following lemma is the basis for the convergence proof.

Lemma 13.2 *Let $b = Ax$ for some nonnegative x. Then for $\{x^k\}$ as in Equation (13.12) we have*

$$\sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^k) - \sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^{k+1}) \geq \tag{13.14}$$

$$\delta_n \sum_{i \in B_n} \alpha_{ni} KL(b_i, (Ax^k)_i). \tag{13.15}$$

Proof: First note that

$$x_j^{k+1} = x_j^k \exp\Big(\gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \log\Big(\frac{b_i}{(Ax^k)_i}\Big)\Big), \tag{13.16}$$

and

$$\exp\Big(\gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \log\Big(\frac{b_i}{(Ax^k)_i}\Big)\Big) \tag{13.17}$$

can be written as

$$\exp\Big((1 - \gamma_j \delta_n \sigma_{nj}) \log 1 + \gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \log\Big(\frac{b_i}{(Ax^k)_i}\Big)\Big), \tag{13.18}$$

which, by the convexity of the exponential function, is not greater than

$$(1 - \gamma_j \delta_n \sigma_{nj}) + \gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \frac{b_i}{(Ax^k)_i}. \tag{13.19}$$

It follows that

$$\sum_{j=1}^{J} \gamma_j^{-1} (x_j^k - x_j^{k+1}) \geq \delta_n \sum_{i \in B_n} \alpha_{ni} ((Ax^k)_i - b_i). \tag{13.20}$$

We also have

$$\log(x_j^{k+1}/x_j^k) = \gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \log \frac{b_i}{(Ax^k)_i}. \tag{13.21}$$

Therefore

$$\sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^k) - \sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^{k+1}) \tag{13.22}$$

$$= \sum_{j=1}^{J} \gamma_j^{-1}(x_j \log(x_j^{k+1}/x_j^k) + x_j^k - x_j^{k+1}) \tag{13.23}$$

$$= \sum_{j=1}^{J} x_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \log \frac{b_i}{(Ax^k)_i} + \sum_{j=1}^{J} \gamma_j^{-1}(x_j^k - x_j^{k+1}) \tag{13.24}$$

$$= \delta_n \sum_{i \in B_n} \alpha_{ni} (\sum_{j=1}^{J} x_j A_{ij}) \log \frac{b_i}{(Ax^k)_i} + \sum_{j=1}^{J} \gamma_j^{-1}(x_j^k - x_j^{k+1}) \tag{13.25}$$

$$\geq \delta_n \Big(\sum_{i \in B_n} \alpha_{ni} (b_i \log \frac{b_i}{(Ax^k)_i} + (Ax^k)_i - b_i) \Big) = \delta_n \sum_{i \in B_n} \alpha_{ni} KL(b_i, (Ax^k)_i). \tag{13.26}$$

This completes the proof of the lemma. ∎

From the inequality (13.15) we conclude that the sequence

$$\{\sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^k)\} \tag{13.27}$$

is decreasing, that $\{x^k\}$ is therefore bounded and the sequence

$$\{\sum_{i \in B_n} \alpha_{ni} KL(b_i, (Ax^k)_i)\} \tag{13.28}$$

is converging to zero. Let x^* be any cluster point of the sequence $\{x^k\}$. Then it is not difficult to show that $b = Ax^*$. Replacing x with x^* we have that the sequence $\{\sum_{j=1}^{J} \gamma_j^{-1} KL(x_j^*, x_j^k)\}$ is decreasing; since a subsequence converges to zero, so does the whole sequence. Therefore x^* is the limit of the sequence $\{x^k\}$. This proves that the algorithm produces a solution of $b = Ax$. To conclude further that the solution is the one for which the quantity $\sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^0)$ is minimized requires further work to replace the inequality (13.15) with an equation in which the right side is independent of the particular solution x chosen; see the final section of this chapter for the details.

We see from the theorem that how we select the γ_j is determined by how we wish to weight the terms in the sum $\sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^0)$. In some cases we want to minimize the cross-entropy $KL(x, x^0)$ subject to $b = Ax$; in this case we would select $\gamma_j = 1$. In other cases we may have some prior knowledge as to the relative sizes of the x_j and wish to

emphasize the smaller values more; then we may choose γ_j proportional to our prior estimate of the size of x_j. Having selected the γ_j, we see from the inequality (13.15) that convergence will be accelerated if we select δ_n as large as permitted by the condition $\gamma_j \delta_n \sigma_{nj} \leq 1$. This suggests that we take

$$\delta_n = 1/\min\{\sigma_{nj}\gamma_j,\, j = 1, ..., J\}. \tag{13.29}$$

The *rescaled* BI-SMART (RBI-SMART) as presented in [50, 52, 53] uses this choice, but with $\alpha_{ni} = 1$ for each n and i. For each $n = 1, ..., N$ let

$$m_n = \max\{s_{nj} s_j^{-1} | j = 1, ..., J\}. \tag{13.30}$$

The original RBI-SMART is as follows:

Algorithm 13.3 (RBI-SMART) *Let x^0 be an arbitrary positive vector. For $k = 0, 1, ...$, let $n = k(\mathrm{mod}\, N) + 1$. Then let*

$$x_j^{k+1} = x_j^k \exp\Big(m_n^{-1} s_j^{-1} \sum_{i \in B_n} A_{ij} \log\Big(\frac{b_i}{(Ax^k)_i}\Big)\Big). \tag{13.31}$$

Notice that Equation (13.31) can be written as

$$\log x_j^{k+1} = (1 - m_n^{-1} s_j^{-1} s_{nj}) \log x_j^k + m_n^{-1} s_j^{-1} \sum_{i \in B_n} A_{ij} \log\Big(x_j^k \frac{b_i}{(Ax^k)_i}\Big), \tag{13.32}$$

from which we see that x_j^{k+1} is a weighted geometric mean of x_j^k and the terms

$$(Q_i x^k)_j = x_j^k \Big(\frac{b_i}{(Ax^k)_i}\Big),$$

for $i \in B_n$. This will be helpful in deriving block-iterative versions of the EMML algorithm. The vectors $Q_i(x^k)$ are sometimes called weighted KL projections.

Let's look now at some of the other choices for these parameters that have been considered in the literature.

First, we notice that the OSSMART does not generally satisfy the requirements, since in (13.11) the choices are $\alpha_{ni} = 1$ and $\beta_{nj} = s_{nj}^{-1}$; the only times this is acceptable is if the s_{nj} are separable; that is, $s_{nj} = r_j t_n$ for some r_j and t_n. This is slightly more general than the condition of subset balance and is sufficient for convergence of OSSMART.

In [82] Censor and Segman make the choices $\beta_{nj} = 1$ and $\alpha_{ni} > 0$ such that $\sigma_{nj} \leq 1$ for all n and j. In those cases in which σ_{nj} is much less than 1 for each n and j their iterative scheme is probably excessively relaxed; it is hard to see how one might improve the rate of convergence by altering

only the weights α_{ni}, however. Limiting the choice to $\gamma_j\delta_n = 1$ reduces our ability to accelerate this algorithm.

The original SMART in Equation (13.5) uses $N = 1$, $\gamma_j = s_j^{-1}$ and $\alpha_{ni} = \alpha_i = 1$. Clearly the inequality (13.13) is satisfied; in fact it becomes an equality now.

For the row-action version of SMART, the *multiplicative* ART (MART), due to Gordon, Bender and Herman [138], we take $N = I$ and $B_n = B_i = \{i\}$ for $i = 1, ..., I$. The MART has the iterative

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i}\Big)^{m_i^{-1}A_{ij}}, \tag{13.33}$$

for $j = 1, 2, ..., J$, $i = k(\mathrm{mod}\, I) + 1$ and $m_i > 0$ chosen so that $m_i^{-1}A_{ij} \leq 1$ for all j. The smaller m_i is the faster the convergence, so a good choice is $m_i = \max\{A_{ij}|,\, j = 1, ..., J\}$. Although this particular choice for m_i is not explicitly mentioned in the various discussions of MART I have seen, it was used in implementations of MART from the beginning [151].

Darroch and Ratcliff included a discussion of a block-iterative version of SMART in their 1972 paper [98]. Close inspection of their version reveals that they require that $s_{nj} = \sum_{i\in B_n} A_{ij} = 1$ for all j. Since this is unlikely to be the case initially, we might try to rescale the equations or unknowns to obtain this condition. However, unless $s_{nj} = \sum_{i\in B_n} A_{ij}$ depends only on j and not on n, which is the *subset balance* property used in [157], we cannot redefine the unknowns in a way that is independent of n.

The MART fails to converge in the inconsistent case. What is always observed, but for which no proof exists, is that, for each fixed $i = 1, 2, ..., I$, as $m \to +\infty$, the MART subsequences $\{x^{mI+i}\}$ converge to separate limit vectors, say $x^{\infty,i}$. This *limit cycle* LC $= \{x^{\infty,i}|i = 1, ..., I\}$ reduces to a single vector whenever there is a nonnegative solution of $b = Ax$. The greater the minimum value of $KL(Ax, y)$ the more distinct from one another the vectors of the limit cycle are. An analogous result is observed for BI-SMART.

13.6 The RBI-EMML

As we did with SMART, we consider now a formulation of BI-EMML that is general enough to include all of the variants we wish to discuss. Once again, the formulation is too general and will need to be restricted in certain ways to obtain convergence. Let the iterative step be

$$x_j^{k+1} = x_j^k(1 - \beta_{nj}\sigma_{nj}) + x_j^k\beta_{nj}\sum_{i\in B_n} \alpha_{ni}A_{ij}\frac{b_i}{(Ax^k)_i}, \tag{13.34}$$

for $j = 1, 2, ..., J$, $n = k(\mathrm{mod}\, N)+1$ and β_{nj} and α_{ni} positive. As in the case of BI-SMART, our convergence proof will require that β_{nj} be separable,

that is,

$$b_{nj} = \gamma_j \delta_n \tag{13.35}$$

for each j and n and that the inequality (13.13) hold. With these conditions satisfied we have the following result.

Theorem 13.4 *Let x be a nonnegative solution of $b = Ax$. For any positive vector x^0 and any collection of blocks $\{B_n, n = 1, ..., N\}$ the sequence $\{x^k\}$ given by Equation (13.12) converges to a nonnegative solution of $b = Ax$.*

When there are multiple nonnegative solutions of $b = Ax$ the solution obtained by BI-EMML will depend on the starting point x^0, but precisely how it depends on x^0 is an open question. Also, in contrast to the case of BI-SMART, the solution can depend on the particular choice of the blocks. The inequality in the following lemma is the basis for the convergence proof.

Lemma 13.3 *Let $b = Ax$ for some nonnegative x. Then for $\{x^k\}$ as in Equation (13.34) we have*

$$\sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^k) - \sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^{k+1}) \geq \tag{13.36}$$

$$\delta_n \sum_{i \in B_n} \alpha_{ni} KL(b_i, (Ax^k)_i). \tag{13.37}$$

Proof: From the iterative step

$$x_j^{k+1} = x_j^k (1 - \gamma_j \delta_n \sigma_{nj}) + x_j^k \gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \frac{b_i}{(Ax^k)_i} \tag{13.38}$$

we have

$$\log(x_j^{k+1}/x_j^k) = \log\Big((1 - \gamma_j \delta_n \sigma_{nj}) + \gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \frac{b_i}{(Ax^k)_i}\Big). \tag{13.39}$$

By the concavity of the logarithm we obtain the inequality

$$\log(x_j^{k+1}/x_j^k) \geq \Big((1 - \gamma_j \delta_n \sigma_{nj}) \log 1 + \gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \log \frac{b_i}{(Ax^k)_i}\Big), \tag{13.40}$$

or

$$\log(x_j^{k+1}/x_j^k) \geq \gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \log \frac{b_i}{(Ax^k)_i}. \tag{13.41}$$

Therefore

$$\sum_{j=1}^{J} \gamma_j^{-1} x_j \log(x_j^{k+1}/x_j^k) \geq \delta_n \sum_{i \in B_n} \alpha_{ni} (\sum_{j=1}^{J} x_j A_{ij}) \log \frac{b_i}{(Ax^k)_i}. \quad (13.42)$$

Note that it is at this step that we used the separability of the β_{nj}. Also

$$\sum_{j=1}^{J} \gamma_j^{-1} (x_j^{k+1} - x_j^k) = \delta_n \sum_{i \in B_n} ((Ax^k)_i - b_i). \quad (13.43)$$

This concludes the proof of the lemma. ∎

From the inequality in (13.37) we conclude, as we did in the BI-SMART case, that the sequence $\{\sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^k)\}$ is decreasing, that $\{x^k\}$ is therefore bounded and the sequence $\{\sum_{i \in B_n} \alpha_{ni} KL(b_i, (Ax^k)_i)\}$ is converging to zero. Let x^* be any cluster point of the sequence $\{x\}$. Then it is not difficult to show that $b = Ax^*$. Replacing x with x^* we have that the sequence $\{\sum_{j=1}^{J} \gamma_j^{-1} KL(x_j^*, x_j^k)\}$ is decreasing; since a subsequence converges to zero, so does the whole sequence. Therefore x^* is the limit of the sequence $\{x^k\}$. This proves that the algorithm produces a nonnegative solution of $b = Ax$. So far, we have been unable to replace the inequality in (13.37) with an equation in which the right side is independent of the particular solution x chosen.

Having selected the γ_j, we see from the inequality in (13.37) that convergence will be accelerated if we select δ_n as large as permitted by the condition $\gamma_j \delta_n \sigma_{nj} \leq 1$. This suggests that once again we take

$$\delta_n = 1/\min\{\sigma_{nj}\gamma_j,\, j = 1, ..., J\}. \quad (13.44)$$

The *rescaled* BI-EMML (RBI-EMML) as presented in [50, 52, 53] uses this choice, but with $\alpha_{ni} = 1$ for each n and i. The original motivation for the RBI-EMML came from consideration of Equation (13.32), replacing the geometric means with arithmetic means. This RBI-EMML is as follows:

Algorithm 13.4 (RBI-EMML) *Let x^0 be an arbitrary positive vector. For $k = 0, 1, ...$, let $n = k(\mathrm{mod}\, N) + 1$. Then let*

$$x_j^{k+1} = (1 - m_n^{-1} s_j^{-1} s_{nj}) x_j^k + m_n^{-1} s_j^{-1} x_j^k \sum_{i \in B_n} (A_{ij} \frac{b_i}{(Ax^k)_i}). \quad (13.45)$$

Let's look now at some of the other choices for these parameters that have been considered in the literature.

First, we notice that the OSEM does not generally satisfy the requirements, since in (13.10) the choices are $\alpha_{ni} = 1$ and $\beta_{nj} = s_{nj}^{-1}$; the only times this is acceptable is if the s_{nj} are separable; that is, $s_{nj} = r_j t_n$ for

some r_j and t_n. This is slightly more general than the condition of subset balance and is sufficient for convergence of OSEM.

The original EMML in Equation (13.6) uses $N = 1$, $\gamma_j = s_j^{-1}$ and $\alpha_{ni} = \alpha_i = 1$. Clearly the inequality (13.13) is satisfied; in fact it becomes an equality now.

Notice that the calculations required to perform the BI-SMART are somewhat more complicated than those needed in BI-EMML. Because the MART converges rapidly in most cases there is considerable interest in the row-action version of EMML. It was clear from the outset that using the OSEM in a row-action mode does not work. We see from the formula for BI-EMML that the proper row-action version of EMML, which we call the EM-MART, is the following:

Algorithm 13.5 (EM-MART) *Let x^0 be an arbitrary positive vector and $i = k(\mathrm{mod}\, I) + 1$. Then let*

$$x_j^{k+1} = (1 - \delta_i \gamma_j \alpha_{ii} A_{ij}) x_j^k + \delta_i \gamma_j \alpha_{ii} x_j^k A_{ij} \frac{b_i}{(Ax^k)_i}, \tag{13.46}$$

with

$$\gamma_j \delta_i \alpha_{ii} A_{ij} \leq 1 \tag{13.47}$$

for all i and j.

The optimal choice would seem to be to take $\delta_i \alpha_{ii}$ as large as possible; that is, to select $\delta_i \alpha_{ii} = 1/\max\{\gamma_j A_{ij}, j = 1, ..., J\}$. With this choice the EM-MART is called the *rescaled* EM-MART (REM-MART).

The EM-MART fails to converge in the inconsistent case. What is always observed, but for which no proof exists, is that, for each fixed $i = 1, 2, ..., I$, as $m \to +\infty$, the EM-MART subsequences $\{x^{mI+i}\}$ converge to separate limit vectors, say $x^{\infty,i}$. This *limit cycle* LC $= \{x^{\infty,i} | i = 1, ..., I\}$ reduces to a single vector whenever there is a nonnegative solution of $b = Ax$. The greater the minimum value of $KL(y, Ax)$ the more distinct from one another the vectors of the limit cycle are. An analogous result is observed for BI-EMML.

We must mention a method that closely resembles the REM-MART, the *row-action maximum likelihood algorithm* (RAMLA), which was discovered independently by Browne and De Pierro [30]. The RAMLA avoids the limit cycle in the inconsistent case by using strong underrelaxation involving a decreasing sequence of relaxation parameters λ_k. The RAMLA is the following:

Algorithm 13.6 (RAMLA) *Let x^0 be an arbitrary positive vector, and $n = k(\mathrm{mod}\, N) + 1$. Let the positive relaxation parameters λ_k be chosen to*

converge to zero and $\sum_{k=0}^{+\infty} \lambda_k = +\infty$. *Then,*

$$x_j^{k+1} = (1 - \lambda_k \sum_{i \in B_n} A_{ij}) x_j^k + \lambda_k x_j^k \sum_{i \in B_n} A_{ij} \Big(\frac{b_i}{(Ax^k)_i} \Big), \tag{13.48}$$

13.7 RBI-SMART and Entropy Maximization

As we stated earlier, in the consistent case the sequence $\{x^k\}$ generated by the BI-SMART algorithm and given by Equation (13.16) converges to the unique solution of $b = Ax$ for which the distance $\sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^0)$ is minimized. In this section we sketch the proof of this result as a sequence of lemmas, each of which is easily established.

Lemma 13.4 *For any nonnegative vectors* a *and* b *with* $a_+ = \sum_{m=1}^{M} a_m$ *and* $b_+ = \sum_{m=1}^{M} b_m > 0$ *we have*

$$KL(a, b) = KL(a_+, b_+) + KL(a_+, \frac{a_+}{b_+} b). \tag{13.49}$$

For nonnegative vectors x and z let

$$G_n(x, z) = \sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, z_j) \tag{13.50}$$

$$+\delta_n \sum_{i \in B_n} \alpha_{ni} [KL((Ax)_i, b_i) - KL((Ax)_i, (Az)_i)]. \tag{13.51}$$

It follows from Equation 13.49 and the inequality

$$\gamma_j^{-1} - \delta_n \sigma_{nj} \geq 1 \tag{13.52}$$

that $G_n(x, z) \geq 0$ in all cases.

Lemma 13.5 *For every* x *we have*

$$G_n(x, x) = \delta_n \sum_{i \in B_n} \alpha_{ni} KL((Ax)_i, b_i) \tag{13.53}$$

so that

$$G_n(x, z) = G_n(x, x) + \sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, z_j) \tag{13.54}$$

$$-\delta_n \sum_{i \in B_n} \alpha_{ni} KL((Ax)_i, (Az)_i). \tag{13.55}$$

Therefore the distance $G_n(x,z)$ is minimized, as a function of z, by $z = x$. Now we minimize $G_n(x,z)$ as a function of x. The following lemma shows that the answer is

$$x_j = z'_j = z_j \exp\Big(\gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \log \frac{b_i}{(Az)_i}\Big). \tag{13.56}$$

Lemma 13.6 *For each x and z we have*

$$G_n(x,z) = G_n(z',z) + \sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, z'_j). \tag{13.57}$$

It is clear that $(x^k)' = x^{k+1}$ for all k.

Now let $b = Pu$ for some nonnegative vector u. We calculate $G_n(u, x^k)$ in two ways: using the definition we have

$$G_n(u,x^k) = \sum_{j=1}^{J} \gamma_j^{-1} KL(u_j, x_j^k) - \delta_n \sum_{i \in B_n} \alpha_{ni} KL(b_i, (Ax^k)_i), \tag{13.58}$$

while using Lemma 13.57 we find that

$$G_n(u,x^k) = G_n(x^{k+1}, x^k) + \sum_{j=1}^{J} \gamma_j^{-1} KL(u_j, x_j^{k+1}). \tag{13.59}$$

Therefore

$$\sum_{j=1}^{J} \gamma_j^{-1} KL(u_j, x_j^k) - \sum_{j=1}^{J} \gamma_j^{-1} KL(u_j, x_j^{k+1}) \tag{13.60}$$

$$= G_n(x^{k+1}, x^k) + \delta_n \sum_{i \in B_n} \alpha_{ni} KL(b_i, (Ax^k)_i). \tag{13.61}$$

We conclude several things from this.

First, the sequence $\{\sum_{j=1}^{J} \gamma_j^{-1} KL(u_j, x_j^k)\}$ is decreasing, so that the sequences $\{G_n(x^{k+1}, x^k)\}$ and $\{\delta_n \sum_{i \in B_n} \alpha_{ni} KL(b_i, (Ax^k)_i)\}$ converge to zero. Therefore the sequence $\{x^k\}$ is bounded and we may select an arbitrary cluster point x^*. It follows that $b = Ax^*$. We may therefore replace the generic solution u with x^* to find that $\{\sum_{j=1}^{J} \gamma_j^{-1} KL(x_j^*, x_j^k)\}$ is a decreasing sequence; but since a subsequence converges to zero, the entire sequence must converge to zero. Therefore $\{x^k\}$ converges to the solution x^*.

Finally, since the right side of Equation (13.61) does not depend on the particular choice of solution we made, neither does the left side. By *telescoping* we conclude that

$$\sum_{j=1}^{J} \gamma_j^{-1} KL(u_j, x_j^0) - \sum_{j=1}^{J} \gamma_j^{-1} KL(u_j, x_j^*) \tag{13.62}$$

is also independent of the choice of u. Consequently, minimizing the function $\sum_{j=1}^{J} \gamma_j^{-1} KL(u_j, x_j^0)$ over all solutions u is equivalent to minimizing $\sum_{j=1}^{J} \gamma_j^{-1} KL(u_j, x_j^*)$ over all solutions u; but the solution to the latter problem is obviously $u = x^*$. This completes the proof.

Chapter 14

Regularization

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, [192, 210] and the discussion in [62]. Penalty functions are also used for regularization [120, 2, 3].

14.1 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,

$$x_{LS} = (A^\dagger A)^{-1} A^\dagger b, \tag{14.1}$$

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

$$Q = \lambda_1 u_1 u_1^\dagger + \cdots + \lambda_J u_J u_J^\dagger, \tag{14.2}$$

where the (necessarily positive) eigenvalues of Q are

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_J > 0, \tag{14.3}$$

and the vectors u_j are the corresponding orthonormal eigenvectors.

14.1.1 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

$$A = \sqrt{\lambda_1} u_1 v_1^\dagger + \cdots + \sqrt{\lambda_I} u_J v_J^\dagger, \tag{14.4}$$

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.

14.1.2 The Inverse of $Q = A^\dagger A$

The inverse of Q can then be written

$$Q^{-1} = \lambda_1^{-1} u_1 u_1^\dagger + \cdots + \lambda_J^{-1} u_J u_J^\dagger, \tag{14.5}$$

so that, with $A^\dagger b = c$, we have

$$x_{LS} = \lambda_1^{-1} (u_1^\dagger c) u_1 + \cdots + \lambda_J^{-1} (u_J^\dagger c) u_J. \tag{14.6}$$

Because the eigenvectors are orthonormal, we can express $||A^\dagger b||_2^2 = ||c||_2^2$ as

$$||c||_2^2 = |u_1^\dagger c|^2 + \cdots + |u_J^\dagger c|^2, \tag{14.7}$$

and $||x_{LS}||_2^2$ as

$$||x_{LS}||_2^2 = \lambda_1^{-1} |u_1^\dagger c|^2 + \cdots + \lambda_J^{-1} |u_J^\dagger c|^2. \tag{14.8}$$

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. As with the ART, regularization can eliminate the problem.

14.1.3 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

$$F_\epsilon(x) = (1 - \epsilon)\|Ax - b\|_2^2 + \epsilon\|x - p\|_2^2. \tag{14.9}$$

Lemma 14.1 *The function F_ϵ always has a unique minimizer $\hat{x}_\epsilon$, given by*

$$\hat{x}_\epsilon = ((1 - \epsilon)A^\dagger A + \epsilon I)^{-1}((1 - \epsilon)A^\dagger b + \epsilon p); \tag{14.10}$$

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

$$\hat{x}_\epsilon = (A^\dagger A + \gamma^2 I)^{-1} A^\dagger b, \tag{14.11}$$

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 14.2 *Let ϵ be in $(0, 1)$, and let I be the identity matrix whose dimensions are understood from the context. Then*

$$((1-\epsilon)AA^\dagger + \epsilon I)^{-1}A = A((1-\epsilon)A^\dagger A + \epsilon I)^{-1}, \tag{14.12}$$

and, taking conjugate transposes,

$$A^\dagger((1-\epsilon)AA^\dagger + \epsilon I)^{-1} = ((1-\epsilon)A^\dagger A + \epsilon I)^{-1}A^\dagger. \tag{14.13}$$

Proof: Use the identity

$$A((1-\epsilon)A^\dagger A + \epsilon I) = ((1-\epsilon)AA^\dagger + \epsilon I)A. \tag{14.14}$$

∎

Lemma 14.3 *Any vector p in 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 14.4 *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

$$\hat{x}_\epsilon = A^\dagger((1-\epsilon)AA^\dagger + \epsilon I)^{-1}((1-\epsilon)b + \epsilon q) + ((1-\epsilon)A^\dagger A + \epsilon I)^{-1}(\epsilon r). \tag{14.15}$$

Lemma 14.5 (a) *We have*

$$((1-\epsilon)A^\dagger A + \epsilon I)^{-1}(\epsilon r) = r, \tag{14.16}$$

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

$$t_\epsilon = ((1-\epsilon)A^\dagger A + \epsilon I)^{-1}(\epsilon r). \tag{14.17}$$

Then, multiplying by A gives

$$At_\epsilon = A((1-\epsilon)A^\dagger A + \epsilon I)^{-1}(\epsilon r). \tag{14.18}$$

Now show that $At_\epsilon = 0$. For part (b) draw a diagram for the case of one equation in two unknowns. ∎

14.2 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 (14.9). For the case of $p = 0$, the solution to this problem is the vector $\hat{x}_\epsilon$ in Equation (14.11). However, we do not want to calculate $A^\dagger A + \gamma^2 I$, in order to solve

$$(A^\dagger A + \gamma^2 I)x = A^\dagger b, \tag{14.19}$$

when the matrix A is large. Fortunately, there are ways to find $\hat{x}_\epsilon$, using only the matrix A. We saw previously how this might be accomplished using the ART; now we show how the Landweber algorithm can be used to calculate this regularized solution.

14.2.1 Regularizing Landweber's Algorithm

Our goal is to minimize the function in Equation (14.9), with $p = 0$. Notice that this is equivalent to minimizing the function

$$F(x) = ||Bx - c||_2^2, \tag{14.20}$$

for

$$B = \begin{bmatrix} A \\ \gamma I \end{bmatrix}, \tag{14.21}$$

and

$$c = \begin{bmatrix} b \\ 0 \end{bmatrix}, \tag{14.22}$$

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

$$x^{k+1} = x^k + \alpha B^T(c - Bx^k), \tag{14.23}$$

for $0 < \alpha < 2/\rho(B^T B)$, where $\rho(B^T B)$ is the spectral radius of $B^T B$. Equation (14.23) can be written as

$$x^{k+1} = (1 - \alpha\gamma^2)x^k + \alpha A^T(b - Ax^k). \tag{14.24}$$

14.3 A Bayesian View of Reconstruction

The EMML iterative algorithm maximizes the likelihood function for the case in which the entries of the data vector $b = (b_1, ..., b_I)^T$ are assumed to be samples of independent Poisson random variables with mean values $(Ax)_i$; here, A is an I by J matrix with nonnegative entries and $x = (x_1, ..., x_J)^T$ is the vector of nonnegative parameters to be estimated. Equivalently, it minimizes the Kullback-Leibler distance $KL(b, Ax)$. This situation arises in single photon emission tomography, where the b_i are the number of photons counted at each detector i, x is the vectorized image to be reconstructed and its entries x_j are (proportional to) the radionuclide intensity levels at each voxel j. When the signal-to-noise ratio is low, which is almost always the case in medical applications, maximizing likelihood can lead to unacceptably noisy reconstructions, particularly when J is larger than I. One way to remedy this problem is simply to halt the EMML algorithm after a few iterations, to avoid over-fitting the x to the noisy data. A more mathematically sophisticated remedy is to employ a penalized-likelihood or Bayesian approach and seek a maximum *a posteriori* (MAP) estimate of x.

In the Bayesian approach we view x as an instance of a random vector having a probability density function $f(x)$. Instead of maximizing the likelihood given the data, we now maximize the posterior likelihood, given both the data and the prior distribution for x. This is equivalent to minimizing

$$F(x) = KL(b, Ax) - \log f(x). \tag{14.25}$$

The EMML algorithm is an example of an optimization method based on alternating minimization of a function $H(x, z) > 0$ of two vector variables. The alternating minimization works this way: let x and z be vector variables and $H(x, z) > 0$. If we fix z and minimize $H(x, z)$ with respect to x, we find that the solution is $x = z$, the vector we fixed; that is,

$$H(x, z) \geq H(z, z) \tag{14.26}$$

always. If we fix x and minimize $H(x, z)$ with respect to z, we get something new; call it Tx. The EMML algorithm has the iterative step $x^{k+1} = Tx^k$.

Obviously, we can't use an arbitrary function H; it must be related to the function $KL(b, Ax)$ that we wish to minimize, and we must be able to obtain each intermediate optimizer in closed form. The clever step is to select $H(x, z)$ so that $H(x, x) = KL(b, Ax)$, for any x. Now see what we have so far:

$$KL(b, Ax^k) = H(x^k, x^k) \geq H(x^k, x^{k+1}) \tag{14.27}$$

$$\geq H(x^{k+1}, x^{k+1}) = KL(b, Ax^{k+1}). \tag{14.28}$$

That tells us that the algorithm makes $KL(b, Ax^k)$ decrease with each iteration. The proof doesn't stop here, but at least it is now plausible that the EMML iteration could minimize $KL(b, Ax)$.

The function $H(x, z)$ used in the EMML case is the KL distance

$$H(x, z) = KL(r(x), q(z)) = \sum_{i=1}^{I} \sum_{j=i}^{J} KL(r(x)_{ij}, q(z)_{ij}); \tag{14.29}$$

we define, for each nonnegative vector x for which $(Ax)_i = \sum_{j=1}^{J} A_{ij}x_j > 0$, the arrays $r(x) = \{r(x)_{ij}\}$ and $q(x) = \{q(x)_{ij}\}$ with entries

$$r(x)_{ij} = x_j A_{ij} \frac{b_i}{(Ax)_i} \tag{14.30}$$

and

$$q(x)_{ij} = x_j A_{ij}. \tag{14.31}$$

With $x = x^k$ fixed, we minimize with respect to z to obtain the next EMML iterate x^{k+1}. Having selected the prior pdf $f(x)$, we want an iterative algorithm to minimize the function $F(x)$ in Equation (14.25). It would be a great help if we could mimic the alternating minimization formulation and obtain x^{k+1} by minimizing

$$KL(r(x^k), q(z)) - \log f(z) \tag{14.32}$$

with respect to z. Unfortunately, to be able to express each new x^{k+1} in closed form, we need to choose $f(x)$ carefully.

14.4 The Gamma Prior Distribution for x

In [175] Lange *et al.* suggest viewing the entries x_j as samples of independent gamma-distributed random variables. A gamma-distributed random variable x takes positive values and has for its pdf the *gamma distribution* defined for positive x by

$$\gamma(x) = \frac{1}{\Gamma(\alpha)} (\frac{\alpha}{\beta})^{\alpha} x^{\alpha-1} e^{-\alpha x/\beta}, \tag{14.33}$$

where α and β are positive parameters and Γ denotes the gamma function. The mean of such a gamma-distributed random variable is then $\mu = \beta$ and the variance is $\sigma^2 = \beta^2/\alpha$.

Lemma 14.6 *If the entries z_j of z are viewed as independent and gamma-distributed with means μ_j and variances σ_j^2, then minimizing the function in line (14.32) with respect to z is equivalent to minimizing the function*

$$KL(r(x^k), q(z)) + \sum_{j=1}^{J} \delta_j KL(\gamma_j, z_j), \tag{14.34}$$

for

$$\delta_j = \frac{\mu_j}{\sigma_j^2},\ \gamma_j = \frac{\mu_j^2 - \sigma_j^2}{\mu_j}, \tag{14.35}$$

under the assumption that the latter term is positive.

The resulting regularized EMML algorithm is the following:

Algorithm 14.1 (γ-prior Regularized EMML) *Let x^0 be an arbitrary positive vector. Then let*

$$x_j^{k+1} = \frac{\delta_j}{\delta_j + s_j}\gamma_j + \frac{1}{\delta_j + s_j} x_j^k \sum_{i=1}^{I} A_{ij} b_i / (Ax^k)_i, \tag{14.36}$$

where $s_j = \sum_{i=1}^{I} A_{ij}$.

We see from Equation (14.36) that the MAP iteration using the gamma priors generates a sequence of estimates each entry of which is a convex combination or weighted arithmetic mean of the result of one EMML step and the prior estimate γ_j. Convergence of the resulting iterative sequence is established by Lange, Bahn and Little in [175]; see also [48].

14.5 The One-Step-Late Alternative

It may well happen that we do not wish to use the gamma priors model and prefer some other $f(x)$. Because we will not be able to find a closed form expression for the z minimizing the function in line (14.32), we need some other way to proceed with the alternating minimization. Green [140] has offered the *one-step-late* (OSL) alternative.

When we try to minimize the function in line (14.32) by setting the gradient to zero we replace the variable z that occurs in the gradient of the term $-\log f(z)$ with x^k, the previously calculated iterate. Then, we can solve for z in closed form to obtain the new x^{k+1}. Unfortunately, negative entries can result and convergence is not guaranteed. There is a sizable literature on the use of MAP methods for this problem. In [57] an interior point algorithm (IPA) is presented that avoids the OSL issue. In [193] the IPA is used to regularize transmission tomographic images.

14.6 Regularizing the SMART

The SMART algorithm is not derived as a maximum likelihood method, so regularized versions do not take the form of MAP algorithms. Nevertheless, in the presence of noisy data, the SMART algorithm suffers from the same problem that afflicts the EMML, overfitting to noisy data resulting in an unacceptably noisy image. As we saw earlier, there is a close connection between the EMML and SMART algorithms. This suggests that a regularization method for SMART can be developed along the lines of the MAP with gamma priors used for EMML. Since the SMART is obtained by minimizing the function $KL(q(z), r(x^k))$ with respect to z to obtain x^{k+1}, it seems reasonable to attempt to derive a regularized SMART iterative scheme by minimizing

$$KL(q(z), r(x^k)) + \sum_{j=1}^{J} \delta_j KL(z_j, \gamma_j), \tag{14.37}$$

as a function of z, for selected positive parameters δ_j and γ_j. This leads to the following algorithm:

Algorithm 14.2 (Regularized SMART) *Let x^0 be an arbitrary positive vector. Then let*

$$\log x_j^{k+1} = \frac{\delta_j}{\delta_j + s_j} \log \gamma_j + \frac{1}{\delta_j + s_j} x_j^k \sum_{i=1}^{I} A_{ij} \log[b_i/(Ax^k)_i]. \tag{14.38}$$

In [48] it was shown that this iterative sequence converges to a minimizer of the function

$$KL(Ax, y) + \sum_{j=1}^{J} \delta_j KL(x_j, \gamma_j). \tag{14.39}$$

It is useful to note that, although it may be possible to rederive this minimization problem within the framework of Bayesian MAP estimation by carefully selecting a prior pdf for the vector x, we have not done so. The MAP approach is a special case of regularization through the use of penalty functions. These penalty functions need not arise through a Bayesian formulation of the parameter-estimation problem.

14.7 De Pierro's Surrogate-Function Method

In [101] De Pierro presents a modified EMML algorithm that includes regularization in the form of a penalty function. His objective is the same as ours was in the case of regularized SMART: to embed the penalty term

in the alternating minimization framework in such a way as to make it possible to obtain the next iterate in closed form. Because his *surrogate function* method has been used subsequently by others to obtain penalized likelihood algorithms [84], we consider his approach in some detail.

Let x and z be vector variables and $H(x,z) > 0$. Mimicking the behavior of the function $H(x,z)$ used in Equation (14.29), we require that if we fix z and minimize $H(x,z)$ with respect to x, the solution should be $x = z$, the vector we fixed; that is, $H(x,z) \geq H(z,z)$ always. If we fix x and minimize $H(x,z)$ with respect to z, we should get something new; call it Tx. As with the EMML, the algorithm will have the iterative step $x^{k+1} = Tx^k$.

Summarizing, we see that we need a function $H(x,z)$ with the properties (1) $H(x,z) \geq H(z,z)$ for all x and z; (2) $H(x,x)$ is the function $F(x)$ we wish to minimize; and (3) minimizing $H(x,z)$ with respect to z for fixed x is easy.

The function to be minimized is

$$F(x) = KL(b, Ax) + g(x), \tag{14.40}$$

where $g(x) \geq 0$ is some penalty function. De Pierro uses penalty functions $g(x)$ of the form

$$g(x) = \sum_{l=1}^{p} f_l(\langle s_l, x\rangle). \tag{14.41}$$

Let us define the matrix S to have for its lth row the vector s_l^T. Then $\langle s_l, x\rangle = (Sx)_l$, the lth entry of the vector Sx. Therefore,

$$g(x) = \sum_{l=1}^{p} f_l((Sx)_l). \tag{14.42}$$

Let $\lambda_{lj} > 0$ with $\sum_{j=1}^{J} \lambda_{lj} = 1$, for each l.

Assume that the functions f_l are convex. Therefore, for each l, we have

$$f_l((Sx)_l) = f_l(\sum_{j=1}^{J} S_{lj}x_j) = f_l(\sum_{j=1}^{J} \lambda_{lj}(S_{lj}/\lambda_{lj})x_j) \tag{14.43}$$

$$\leq \sum_{j=1}^{J} \lambda_{lj} f_l((S_{lj}/\lambda_{lj})x_j). \tag{14.44}$$

Therefore,

$$g(x) \leq \sum_{l=1}^{p}\sum_{j=1}^{J} \lambda_{lj} f_l((S_{lj}/\lambda_{lj})x_j). \tag{14.45}$$

So we have replaced $g(x)$ with a related function in which the x_j occur separately, rather than just in the combinations $(Sx)_l$. But we aren't quite done yet.

We would like to take for De Pierro's $H(x,z)$ the function used in the EMML algorithm, plus the function

$$\sum_{l=1}^{p}\sum_{j=1}^{J}\lambda_{lj}f_l((S_{lj}/\lambda_{lj})z_j). \tag{14.46}$$

But there is one slight problem: we need $H(z,z) = F(z)$, which we don't have yet.

De Pierro's clever trick is to replace $f_l((S_{lj}/\lambda_{lj})z_j)$ with

$$f_l\Big((S_{lj}/\lambda_{lj})z_j - (S_{lj}/\lambda_{lj})x_j\Big) + f_l((Sx)_l). \tag{14.47}$$

So, De Pierro's function $H(x,z)$ is the sum of the $H(x,z)$ used in the EMML case and the function

$$\sum_{l=1}^{p}\sum_{j=1}^{J}\lambda_{lj}f_l\Big((S_{lj}/\lambda_{lj})z_j - (S_{lj}/\lambda_{lj})x_j\Big) + \sum_{l=1}^{p}f_l((Sx)_l). \tag{14.48}$$

Now he has the three properties he needs. Once he has computed x^k, he minimizes $H(x^k,z)$ by taking the gradient and solving the equations for the correct $z = Tx^k = x^{k+1}$. For the choices of f_l he discusses, these intermediate calculations can either be done in closed form (the quadratic case) or with a simple Newton-Raphson iteration (the logcosh case).

14.8 Block-Iterative Regularization

We saw previously that it is possible to obtain a regularized least-squares solution $\hat{x}_\epsilon$, and thereby avoid the limit cycle, using only the matrix A and the ART algorithm. This prompts us to ask if it is possible to find regularized SMART solutions using block-iterative variants of SMART. Similarly, we wonder if it is possible to do the same for EMML.

Open Question: Can we use the MART to find the minimizer of the function

$$KL(Ax,b) + \epsilon KL(x,p)? \tag{14.49}$$

More generally, can we obtain the minimizer using RBI-SMART?

Open Question: Can we use the RBI-EMML methods to obtain the minimizer of the function

$$KL(b,Ax) + \epsilon KL(p,x)? \tag{14.50}$$

There have been various attempts to include regularization in block-iterative methods, to reduce noise sensitivity and avoid limit cycles; the paper by Ahn and Fessler [2] is a good source, as is [3]. Most of these approaches have been *ad hoc*, with little or no theoretical basis. Typically, they simply modify each iterative step by including an additional term that appears to be related to the regularizing penalty function. The case of the ART is instructive, however. In that case, we obtained the desired iterative algorithm by using an augmented set of variables, not simply by modifying each step of the original ART algorithm. How to do this for the MART and the other block-iterative algorithms is not obvious.

Recall that the RAMLA method in Equation (13.48) is similar to the RBI-EMML algorithm, but employs a sequence of decreasing relaxation parameters, which, if properly chosen, will cause the iterates to converge to the minimizer of $KL(b, Ax)$, thereby avoiding the limit cycle. In [103] De Pierro and Yamaguchi present a regularized version of RAMLA, but without guaranteed convergence.

Chapter 15

Block-Iterative ART

15.1 Introduction and Notation

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 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. [138] 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

$$x_j^k = x_j^{k-1} + A_{ij}\left(\frac{b_i - (Ax^{k-1})_i}{\sum_{t=1}^{J} |A_{it}|^2}\right), \tag{15.1}$$

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 [87] is a *simultaneous* method, in which all the equations are used at each step. The current vector x^{k-1} is projected orthog-

onally 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

$$x_j^k = \frac{1}{I}\sum_{i=1}^{I}\left(x_j^{k-1} + A_{ij}\left(\frac{b_i - (Ax^{k-1})_i}{\sum_{t=1}^{J}|A_{it}|^2}\right)\right),$$

which can also be written as

$$x_j^k = x_j^{k-1} + \sum_{i=1}^{I} A_{ij}\left(\frac{b_i - (Ax^{k-1})_i}{I\sum_{t=1}^{J}|A_{it}|^2}\right). \tag{15.2}$$

Landweber's iterative scheme [172] 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. [78] 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 [64] 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) [79], the CARP1 method of Gordon and Gordon [139], a block-iterative variant of CARP1 obtained from the DROP method of Censor et al. [76], and the SIRT method [241].

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

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

15.2 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 $i(k) = k(\mathrm{mod}\,)I$. 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

$$x_j^k = x_j^{k-1} + \frac{1}{I}\sum_{i=1}^{I}\left(\frac{A_{ij}(b_i - (Ax^{k-1})_i)}{\nu_i^2}\right). \tag{15.5}$$

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.

15.3 The Landweber Algorithms

For simplicity, we assume, in this section, that $\nu_i = 1$ for all i. The Landweber algorithm [172, 17], 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 15.2 and 15.3.

15.3.1 Finding the Optimum γ

The operator

$$Tx = x + \gamma A^\dagger(b - Ax) = (I - \gamma A^\dagger A)x + \gamma A^\dagger b$$

is affine linear and is av if and only if its linear part, the Hermitian matrix

$$B = I - \gamma A^\dagger A,$$

is av. To guarantee this 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 sc. 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 $(0, 2/\lambda_{max})$. The matrix B is still av, but it is no longer sc. 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 $\rho(B) = 1$. The operator B is av, in fact, it is fne, but it is not sc.

15.3.2 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 [17], 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 X, 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.

Both the Landweber and projected Landweber algorithms are special cases of the CQ algorithm [59], which, in turn, is a special case of the more general iterative fixed point algorithm, the Krasnoselskii/Mann (KM) method, with convergence governed by the KM Theorem (see [65]).

15.4 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 $A^\dagger A$.

15.4.1 Our Basic Eigenvalue Inequality

In [241] 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 $A^\dagger A$ ([64]). 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 15.1 *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

$$\|A^\dagger w\|^2 = \lambda \|w\|^2.$$

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,

$$\|A^\dagger w\|^2 \leq \sum_{j=1}^{J}\Big(c_{aj}(\sum_{i=1}^{I}|A_{ij}|^{2-a}|w_i|^2)\Big) = \sum_{i=1}^{I}\Big(\sum_{j=1}^{J}c_{aj}|A_{ij}|^{2-a}\Big)|w_i|^2$$

$$\leq \max_i\Big(\sum_{j=1}^{J}c_{aj}|A_{ij}|^{2-a}\Big)\|w\|^2.$$

The remaining two assertions follow in similar fashion. ∎

As a corollary, we obtain the following eigenvalue inequality, which is central to our discussion.

Theorem 15.2 *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 15.1 *Selecting $a = 1$, we have*

$$L = \|A\|_2^2 \leq \|A\|_1\|A\|_\infty = c_1r_1.$$

Corollary 15.2 *Selecting $a = 2$, we have*

$$L = \|A\|_2^2 \leq \|A\|_F^2,$$

where $\|A\|_F$ denotes the Frobenius norm of A.

Corollary 15.3 *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 15.2. ∎

Corollary 15.4 *If* $\sum_{j=1}^{J} s_j |A_{ij}|^2 \leq 1$ *for all* i*, then* $L \leq 1$*.*

Corollary 15.5 *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 15.6 *([59]; [240], Th. 4.2) If* $\sum_{j=1}^{J} |A_{ij}|^2 = 1$ *for each* i*, then* $L \leq s$*.*

Proof: For all i we have

$$p_i = \sum_{j=1}^{J} s_j |A_{ij}|^2 \leq s \sum_{j=1}^{J} |A_{ij}|^2 = s.$$

Therefore,

$$L \leq p \leq s.$$

∎

Corollary 15.7 *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, $\rho(G^\dagger G) \leq 1$. ∎

15.4.2 Another Upper Bound for L

The next theorem ([59]) 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 15.3 *([59]) 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 15.8 *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. ∎

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 [87], is acceptable, although perhaps much too small.

The proof of Theorem 15.3 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.

15.5 The Basic Convergence Theorem

The following theorem is a basic convergence result concerning block-iterative ART algorithms.

Theorem 15.4 *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 $\|x - x^0\|$ is minimized.

We begin with the following lemma.

Lemma 15.3 *Let T be any (not necessarily linear) operator on R^J, and $S = I - T$, where I denotes the identity operator. Then, for any x and y, we have*

$$\|x - y\|^2 - \|Tx - Ty\|^2 = 2\langle Sx - Sy, x - y\rangle - \|Sx - Sy\|^2. \tag{15.24}$$

The proof is a simple calculation and we omit it here.

Proof of Theorem 15.4: Let $Az = b$. Applying Equation (15.24) to the operator

$$Tx = x + \lambda_k A_n^\dagger (b^n - A_n x),$$

we obtain

$$\|z - x^{k-1}\|^2 - \|z - x^k\|^2 = 2\lambda_k \|b^n - A_n x^{k-1}\|^2 - \lambda_k^2 \|A_n^\dagger b^n - A_n^\dagger A_n x^{k-1}\|^2. \tag{15.25}$$

Since $L_n \leq 1$, it follows that

$$\|A_n^\dagger b^n - A_n^\dagger A_n x^{k-1}\|^2 \leq \|b^n - A_n x^{k-1}\|^2.$$

Therefore,

$$\|z - x^{k-1}\|^2 - \|z - x^k\|^2 \geq (2\lambda_k - \lambda_k^2)\|b^n - A_n x^{k-1}\|^2,$$

from which we draw several conclusions:

- the sequence $\{\|z - x^k\|\}$ is decreasing;
- the sequence $\{\|b^n - A_n x^{k-1}\|\}$ converges to zero.

In addition, for fixed $n = 1, ..., N$ and $m \to \infty$,

- the sequence $\{\|b^n - A_n x^{mN+n-1}\|\}$ 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 $\{\|x^* - x^k\|\}$ 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 $\|x - x^0\|$. 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 $\|x - x^0\|$. 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

$$\|x - x^0\|^2 - \|x - x^*\|^2$$

is independent of the choice of solution. Consequently, minimizing $\|x - x^0\|$ over all solutions x is equivalent to minimizing $\|x - x^*\|$ over all solutions x; the solution to the latter problem is clearly $x = x^*$. ∎

15.6 Simultaneous Iterative Algorithms

In this section we apply the previous theorems to obtain convergence of several simultaneous iterative algorithms for linear systems.

15.6.1 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 15.5 *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 15.7 we have that $\rho(G^\dagger G) \leq 1$. Convergence then follows from Theorem 15.4. ∎

Corollary 15.9 *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 15.3 that $\rho(G^\dagger G) \leq 1$. ∎

15.6.2 Some Convergence Results

We obtain convergence for several known algorithms as corollaries to the previous theorems.

The SIRT Algorithm:

Corollary 15.10 *([241]) 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 [145]. The SART algorithm [5] is a special case, in which it is assumed that $A_{ij} \geq 0$, for all i and j.

The CAV Algorithm:

Corollary 15.11 *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. [78].

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 $\|Ax - b\|$ for which the distance $\|x - x^0\|$ 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 [87], is far too large when A is sparse.

The Simultaneous DROP Algorithm:

Corollary 15.12 *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 [139] (see also [106, 78, 79]). The simultaneous DROP algorithm of [76] 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 [139] that the two methods differed only slightly in the simulated cases studied.

15.7 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.

15.7.1 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 $\|x - x^0\|$, 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 15.4 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 $i \in B_n$.

15.7.2 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 15.2 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 [79] uses

$$\gamma_i = \Big(\sum_{j=1}^{J} s_{nj} |A_{ij}|^2 \Big)^{-1}.$$

The iterative step of BICAV is

$$x^k = x^{k-1} + \lambda_k \sum_{i \in B_n} \left(\frac{b_i - (Ax^{k-1})_i}{\sum_{t=1}^{J} s_{nt} |A_{it}|^2} \right) A_i^\dagger. \tag{15.35}$$

15.7.3 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 [76], 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 [160] 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.

15.7.4 Using Sparseness

Suppose, for the sake of illustration, that each column of A has s non-zero 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).

15.8 Exercises

Exercise 15.1 *Prove Lemma 15.1.*

Exercise 15.2 (Computer Problem) *Compare the speed of convergence of the ART and Cimmino algorithms.*

Exercise 15.3 (Computer Problem) *By generating sparse matrices of various sizes, test the accuracy of the estimates of the largest singular-value given above.*

Chapter 16

The Split Feasibility Problem

The *split feasibility problem* (SFP) [74] is to find $c \in C$ with $Ac \in Q$, if such points exist, where A is a real I by J matrix and C and Q are nonempty, closed convex sets in R^J and R^I, respectively. In this chapter we discuss the CQ algorithm for solving the SFP, as well as recent extensions and applications.

16.1 The CQ Algorithm

In [59] the CQ algorithm for solving the SFP was presented, for the real case. It has the iterative step

$$x^{k+1} = P_C(x^k - \gamma A^T(I - P_Q)Ax^k), \tag{16.1}$$

where I is the identity operator and $\gamma \in (0, 2/\rho(A^T A))$, for $\rho(A^T A)$ the spectral radius of the matrix $A^T A$, which is also its largest eigenvalue. The CQ algorithm can be extended to the complex case, in which the matrix A has complex entries, and the sets C and Q are in C^J and C^I, respectively. The iterative step of the extended CQ algorithm is then

$$x^{k+1} = P_C(x^k - \gamma A^\dagger(I - P_Q)Ax^k). \tag{16.2}$$

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 [60]. The CQ algorithm employs the relaxation parameter γ in the interval $(0, 2/L)$, where L is the largest eigenvalue of the matrix A^TA. 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^TA 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.

16.2 Particular Cases of the CQ Algorithm

It is easy to find important examples of the SFP: if $C \subseteq R^J$ and $Q = \{b\}$ then solving the SFP amounts to solving the linear system of equations $Ax = b$; if C is a proper subset of R^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 [172] and projected Landweber methods (see [17]), are particular cases of the CQ algorithm.

16.2.1 The Landweber algorithm

With x^0 arbitrary and $k = 0, 1, ...$ let

$$x^{k+1} = x^k + \gamma A^T(b - Ax^k). \tag{16.3}$$

This is the Landweber algorithm.

16.2.2 The Projected Landweber Algorithm

For a general nonempty closed convex C, x^0 arbitrary, and $k = 0, 1, ...$, the projected Landweber method for finding a solution of $Ax = b$ in C has the iterative step

$$x^{k+1} = P_C(x^k + \gamma A^T(b - Ax^k)). \tag{16.4}$$

16.2.3 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.

16.2.4 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:

$$x_j^{k+1} = x_j^k + \frac{1}{A_{+j}} \sum_{i=1}^{I} A_{ij}(b_i - (Ax^k)_i)/A_{i+}.$$

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

$$z^{k+1} = z^k + B^T(c - Bz^k).$$

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 [59].

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 $B^Tv = u$; that is, u is an eigenvector of B^TB 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^TB is symmetric its eigenvectors are orthogonal; therefore u itself must be an eigenvector associated with the largest eigenvalue of B^TB. The convergence of SART follows.

16.2.5 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 matrix A of detection probabilities; our model is $Ax = b$, for x a nonnegative vector. We could then take $Q = \{b\}$ and $C = R^N_+$, the nonnegative cone in R^N.

In *dynamic* ET [118] 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.

16.2.6 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 [249].

Chapter 17

Conjugate-Direction Methods

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.

17.1 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

$$x^k = x^{k-1} + \alpha_k d^k. \tag{17.1}$$

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.

Exercise 17.1 *Differentiate the function $f(x^{k-1} + \alpha d^k)$ with respect to the variable α to show that*

$$\nabla f(x^k) \cdot d^k = 0. \tag{17.2}$$

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* [183]:

$$x^{k+1} = x^k - \alpha_{k+1} \nabla f(x^k).$$

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 [194] 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.

17.2 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

$$f(x) = \frac{1}{2} ||b - Ax||_2^2. \tag{17.3}$$

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 $Q = A^T A$ and $c = A^T b$. 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 (17.1) for $f(x)$ as in Equation (17.3). 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.

Exercise 17.2 *Show that the optimal α_k is*

$$\alpha_k = \frac{r^k \cdot d^k}{d^k \cdot Q d^k}, \tag{17.4}$$

where $r^k = c - Qx^{k-1}$.

Exercise 17.3 *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 R^J and we employ them cyclically, convergence follows.

Theorem 17.1 *Let $\{d^1, ..., d^J\}$ be any finite set whose span is all of R^J. Let α_k be chosen according to Equation (17.4). Then, for $k = 0, 1, ...$, $j = k(\mathrm{mod}\, J) + 1$, 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. ∎

There is an interesting corollary to this theorem that pertains to a modified version of the ART algorithm. For $k = 0, 1, ...$ and $i = k(\mathrm{mod}\, M) + 1$ and with the rows of A normalized to have length one, the ART iterative step is

$$x^{k+1} = x^k + (b_i - (Ax^k)_i)a^i,$$

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:

$$x^{k+1} = x^k + \frac{r^{k+1} \cdot a^i}{a^i \cdot Qa^i} a^i.$$

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 17.2 *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. ∎

17.3 Conjugate Bases for R^J

If the set $\{v^1, ..., v^J\}$ is a basis for R^J, then any vector x in 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 $m = 1, ..., M$. 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 the set $\{u^1, ..., u^M\}$ is an orthogonal basis, that is, then $u^j \cdot u^m = 0$, unless $j = m$, then the system of linear equations is now 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 $\hat{x}$.

17.3.1 Conjugate Directions

From Equation (17.2) we have

$$(c - Qx^{k+1}) \cdot d^k = 0,$$

which can be expressed as

$$(\hat{x} - x^{k+1}) \cdot Qd^k = (\hat{x} - x^{k+1})^T Qd^k = 0.$$

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+1} 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.*

Exercise 17.4 *Say that the set $\{p^1, ..., p^n\}$ is a conjugate set for 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.

17.3.2 The Gram-Schmidt Method

Let $\{v^1, ..., v^J\}$ be a linearly independent set of vectors in the space R^M, where $J \leq M$. The Gram-Schmidt method uses the v^j to create an orthogonal basis $\{u^1, ..., u^J\}$ for the span of the v^j. 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,

$$p^j = v^j - \frac{p^1 \cdot Qv^j}{p^1 \cdot Qp^1} p^1 - ... - \frac{p^{j-1} \cdot Qv^j}{p^{j-1} \cdot Qp^{j-1}} p^{j-1}. \tag{17.5}$$

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.

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 [183]. 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 17.3 *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

$$p^{n+1} = Qp^n - \frac{Qp^n \cdot Qp^n}{p^n \cdot Qp^n} p^n - \frac{Qp^{n-1} \cdot Qp^n}{p^{n-1} \cdot Qp^{n-1}} p^{n-1}. \tag{17.6}$$

Then, the set $\{p^1, ..., p^J\}$ is a conjugate set for R^J. If $p^n \neq 0$ for each n, then the set is a conjugate basis for 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 (17.6) 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 (17.6) also occur in the Gram-Schmidt approach in Equation (17.5); the point is that Equation (17.6) uses only the first three terms, in every case.

17.4 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

$$x^n = x^{n-1} + \alpha_n p^n. \tag{17.7}$$

The α_n is chosen so as to minimize the function of α defined by $f(x^{n-1} + \alpha p^n)$, and so we have

$$\alpha_n = \frac{r^n \cdot p^n}{p^n \cdot Qp^n},$$

where $r^n = c - Qx^{n-1}$. 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+1} to construct the next direction p^{n+1} 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

$$p^{n+1} = r^{n+1} - \frac{r^{n+1} \cdot Qp^n}{p^n \cdot Qp^n} p^n. \tag{17.8}$$

Since the α_n is the optimal choice and

$$r^{n+1} = -\nabla f(x^n),$$

we have, according to Equation (17.2),

$$r^{n+1} \cdot p^n = 0.$$

Exercise 17.5 *Prove that $r^{n+1} = 0$ whenever $p^{n+1} = 0$, in which case we have $c = Qx^n$, so that x^n is the least-squares solution.*

In theory, the CGM converges to the least squares solution in finitely many steps, since we either reach $p^{n+1} = 0$ or $n + 1 = J$. 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 17.3 establishes that the set $\{p^1, ..., p^J\}$ is a conjugate set [183, 194]. In fact, we can say more.

Theorem 17.4 *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 [194].

The proof uses induction on the number n. Throughout the following exercises assume that the statements in the theorem hold for some $n < J$. We prove that they hold also for $n + 1$.

Exercise 17.6 *Use the fact that*

$$r^{j+1} = r^j - \alpha_j Qp^j,$$

to show that Qp^j is in the span of the vectors r^j and r^{j+1}.

Exercise 17.7 *Show that $r^{n+1} \cdot r^n = 0$. Hint: establish that*

$$\alpha_n = \frac{r^n \cdot r^n}{p^n \cdot Qp^n}.$$

Exercise 17.8 *Show that $r^{n+1} \cdot r^j = 0$, for $j = 1, ..., n - 1$. Hint: use the induction hypothesis.*

Exercise 17.9 *Show that $r^{n+1} \cdot p^j = 0$, for $j = 1, ..., n$. Hint: first, establish that*

$$p^j = r^j - \beta_{j-1} p^{j-1},$$

where

$$\beta_{j-1} = \frac{r^j \cdot Qp^{j-1}}{p^{j-1} \cdot Qp^{j-1}},$$

and

$$r^{n+1} = r^n - \alpha_n Qp^n.$$

Exercise 17.10 *Show that $p^{n+1} \cdot Qp^j = 0$, for $j = 1, ..., n - 1$. Hint: use*

$$Qp^j = \alpha_j^{-1}(r^j - r^{j+1}).$$

The final step in the proof is contained in the following exercise.

Exercise 17.11 *Show that* $p^{n+1} \cdot Qp^n = 0$. *Hint: establish that*

$$\beta_n = -\frac{r^{n+1} \cdot r^{n+1}}{r^n \cdot r^n}.$$

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]).

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 [194].

Chapter 18

Constrained Iteration Methods

The ART and its simultaneous and block-iterative versions are designed to solve general systems of linear equations $Ax = b$. The SMART, EMML and RBI methods require that the entries of A be nonnegative, those of b positive and produce nonnegative x. In this chapter we present variations of the SMART and EMML that impose the constraints $u_j \leq x_j \leq v_j$, where the u_j and v_j are selected lower and upper bounds on the individual entries x_j. These algorithms were used in [193] as a method for including in transmission tomographic reconstruction spatially varying upper and lower bounds on the x-ray attenuation.

18.1 Modifying the KL distance

The SMART, EMML and RBI methods are based on the Kullback-Leibler distance between nonnegative vectors. To impose more general constraints on the entries of x we derive algorithms based on shifted KL distances, also called Fermi-Dirac generalized entropies.

For a fixed real vector u, the shifted KL distance $KL(x - u, z - u)$ is defined for vectors x and z having $x_j \geq u_j$ and $z_j \geq u_j$. Similarly, the shifted distance $KL(v - x, v - z)$ applies only to those vectors x and z for which $x_j \leq v_j$ and $z_j \leq v_j$. For $u_j \leq v_j$, the combined distance

$$KL(x - u, z - u) + KL(v - x, v - z)$$

is restricted to those x and z whose entries x_j and z_j lie in the interval $[u_j, v_j]$. Our objective is to mimic the derivation of the SMART, EMML and RBI methods, replacing KL distances with shifted KL distances, to obtain algorithms that enforce the constraints $u_j \leq x_j \leq v_j$, for each j.

The algorithms that result are the ABMART and ABEMML block-iterative methods. These algorithms were originally presented in [54], in which the vectors u and v were called a and b, hence the names of the algorithms. Throughout this chapter we shall assume that the entries of the matrix A are nonnegative. We shall denote by B_n, $n = 1, ..., N$ a partition of the index set $\{i = 1, ..., I\}$ into blocks. For $k = 0, 1, ...$ let $n(k) = k(\mathrm{mod}\, N)+1$.

The projected Landweber algorithm can also be used to impose the restrictions $u_j \leq x_j \leq v_j$; however, the projection step in that algorithm is implemented by clipping, or setting equal to u_j or v_j values of x_j that would otherwise fall outside the desired range. The result is that the values u_j and v_j can occur more frequently than may be desired. One advantage of the AB methods is that the values u_j and v_j represent barriers that can only be reached in the limit and are never taken on at any step of the iteration.

18.2 The ABMART Algorithm

We assume that $(Au)_i \leq b_i \leq (Av)_i$ and seek a solution of $Ax = b$ with $u_j \leq x_j \leq v_j$, for each j. The algorithm begins with an initial vector x^0 satisfying $u_j \leq x_j^0 \leq v_j$, for each j. Having calculated x^k, we take

$$x_j^{k+1} = \alpha_j^k v_j + (1 - \alpha_j^k) u_j, \tag{18.1}$$

with $n = n(k)$,

$$\alpha_j^k = \frac{c_j^k \prod^n (d_i^k)^{A_{ij}}}{1 + c_j^k \prod^n (d_i^k)^{A_{ij}}}, \tag{18.2}$$

$$c_j^k = \frac{(x_j^k - u_j)}{(v_j - x_j^k)}, \tag{18.3}$$

and

$$d_j^k = \frac{(b_i - (Au)_i)((Av)_i - (Ax^k)_i)}{((Av)_i - b_i)((Ax^k)_i - (Au)_i)}, \tag{18.4}$$

where $\prod^n$ denotes the product over those indices i in $B_{n(k)}$. Notice that, at each step of the iteration, x_j^k is a convex combination of the endpoints u_j and v_j, so that x_j^k lies in the interval $[u_j, v_j]$.

We have the following theorem concerning the convergence of the ABMART algorithm:

Theorem 18.1 *If there is a solution of the system $Ax = b$ that satisfies the constraints $u_j \leq x_j \leq v_j$ for each j, then, for any N and any choice of the*

blocks B_n, the ABMART sequence converges to that constrained solution of $Ax = b$ for which the Fermi-Dirac generalized entropic distance from x to x^0,

$$KL(x - u, x^0 - u) + KL(v - x, v - x^0),$$

is minimized. If there is no constrained solution of $Ax = b$, then, for $N = 1$, the ABMART sequence converges to the minimizer of

$$KL(Ax - Au, b - Au) + KL(Av - Ax, Av - b)$$

for which

$$KL(x - u, x^0 - u) + KL(v - x, v - x^0)$$

is minimized.

The proof is similar to that for RBI-SMART and is found in [54].

18.3 The ABEMML Algorithm

We make the same assumptions as in the previous section. The iterative step of the ABEMML algorithm is

$$x_j^{k+1} = \alpha_j^k v_j + (1 - \alpha_j^k) u_j, \tag{18.5}$$

where

$$\alpha_j^k = \gamma_j^k / d_j^k, \tag{18.6}$$

$$\gamma_j^k = (x_j^k - u_j) e_j^k, \tag{18.7}$$

$$\beta_j^k = (v_j - x_j^k) f_j^k, \tag{18.8}$$

$$d_j^k = \gamma_j^k + \beta_j^k, \tag{18.9}$$

$$e_j^k = \left(1 - \sum_{i \in B_n} A_{ij}\right) + \sum_{i \in B_n} A_{ij} \left(\frac{b_i - (Au)_i}{(Ax^k)_i - (Au)_i}\right), \tag{18.10}$$

and

$$f_j^k = \left(1 - \sum_{i \in B_n} A_{ij}\right) + \sum_{i \in B_n} A_{ij} \left(\frac{(Av)_i - b_i}{(Av)_i - (Ax^k)_i}\right). \tag{18.11}$$

We have the following theorem concerning the convergence of the ABEMML algorithm:

Theorem 18.2 *If there is a solution of the system $Ax = b$ that satisfies the constraints $u_j \leq x_j \leq v_j$ for each j, then, for any N and any choice of the blocks B_n, the ABEMML sequence converges to such a constrained solution of $Ax = b$. If there is no constrained solution of $Ax = b$, then, for $N = 1$, the ABMART sequence converges to a constrained minimizer of*

$$KL(Ax - Au, b - Au) + KL(Av - Ax, Av - b).$$

The proof is similar to that for RBI-EMML and is to be found in [54]. In contrast to the ABMART theorem, this is all we can say about the limits of the ABEMML sequences.

Open Question: How does the limit of the ABEMML iterative sequence depend, in the consistent case, on the choice of blocks, and, in general, on the choice of x^0?

Part IV

Applications

Chapter 19

Transmission Tomography I

In this part of the text we focus on transmission tomography. This chapter will provide a detailed description of how the data is gathered, the mathematical model of the scanning process, and the problem to be solved. In subsequent chapters we shall study the various mathematical techniques needed to solve this problem and the manner in which these techniques are applied.

19.1 X-ray Transmission Tomography

Although transmission tomography is not limited to scanning living beings, we shall concentrate here on the use of x-ray tomography in medical diagnosis and the issues that concern us in that application. The mathematical formulation will, of course, apply more generally.

In x-ray tomography, x-rays are transmitted through the body along many lines. In some, but not all, cases, the lines will all lie in the same plane. The strength of the x-rays upon entering the body is assumed known, and the strength upon leaving the body is measured. This data can then be used to estimate the amount of attenuation the x-ray encountered along that line, which is taken to be the integral, along that line, of the attenuation function. On the basis of these line integrals, we estimate the attenuation function. This estimate is presented to the physician as one or more two-dimensional images.

19.2 The Exponential-Decay Model

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

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

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. To see why this is the case, imagine the line L parameterized by the variable s and consider the intensity function $I(s)$ as a function of s. For small $\Delta s > 0$, the drop in intensity from the start to the end of the interval $[s, s + \Delta s]$ is approximately proportional to the intensity $I(s)$, to the attenuation $f(s)$ and to Δs, the length of the interval; that is,

$$I(s) - I(s + \Delta s) \approx f(s) I(s) \Delta s.$$

Dividing by Δs and letting Δs approach zero, we get

$$I'(s) = -f(s) I(s).$$

Exercise 19.1 *Show that the solution to this differential equation is*

$$I(s) = I(0) \exp(-\int_{u=0}^{u=s} f(u) du).$$

Hint: Use an integrating factor.

From knowledge of I_{in} and I_{out}, we can determine $\int_L f$. If we know $\int_L f$ for every line in the x, y-plane we can reconstruct the attenuation function f. In the real world we know line integrals only approximately and only for finitely many lines. The goal in x-ray transmission tomography is to estimate the attenuation function $f(x, y)$ in the slice, from finitely many noisy measurements of the line integrals. We usually have prior information about the values that $f(x, y)$ can take on. We also expect to find sharp boundaries separating regions where the function $f(x, y)$ varies only slightly. Therefore, we need algorithms capable of providing such images.

19.3 Difficulties to be Overcome

There are several problems associated with this model. X-ray beams are not exactly straight lines; the beams tend to spread out. The x-rays are not monochromatic, and their various frequency components are attenuated at

different rates, resulting in *beam hardening*, that is, changes in the spectrum of the beam as it passes through the object (see the appendix on the Laplace transform). The beams consist of photons obeying statistical laws, so our algorithms probably should be based on these laws. How we choose the line segments is determined by the nature of the problem; in certain cases we are somewhat limited in our choice of these segments. Patients move; they breathe, their hearts beat, and, occasionally, they shift position during the scan. Compensating for these motions is an important, and difficult, aspect of the image reconstruction process. Finally, to be practical in a clinical setting, the processing that leads to the reconstructed image must be completed in a short time, usually around fifteen minutes. This time constraint is what motivates viewing the three-dimensional attenuation function in terms of its two-dimensional slices.

As we shall see, the Fourier transform and the associated theory of convolution filters play important roles in the reconstruction of transmission tomographic images.

The data we actually obtain at the detectors are counts of detected photons. These counts are not the line integrals; they are random quantities whose means, or expected values, are related to the line integrals. The Fourier inversion methods for solving the problem ignore its statistical aspects; in contrast, other methods, such as likelihood maximization, are based on a statistical model that involves Poisson-distributed emissions.

19.4 Reconstruction from Line Integrals

We turn now to the underlying problem of reconstructing attenuation functions from line-integral data.

19.4.1 The Radon Transform

Our goal is to reconstruct the function $f(x, y) \geq 0$ from line-integral data. Let θ be a fixed angle in the interval $[0, \pi)$. Form the t, s-axis system with the positive t-axis making the angle θ with the positive x-axis, as shown in Figure 19.1. Each point (x, y) in the original coordinate system has coordinates (t, s) in the second system, where the t and s are given by

$$t = x \cos\theta + y \sin\theta,$$

and

$$s = -x \sin\theta + y \cos\theta.$$

If we have the new coordinates (t, s) of a point, the old coordinates are (x, y) given by

$$x = t \cos\theta - s \sin\theta,$$

and

$$y = t\sin\theta + s\cos\theta.$$

We can then write the function f as a function of the variables t and s. For each fixed value of t, we compute the integral

$$\int_L f(x,y)ds = \int f(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta)ds$$

along the single line L corresponding to the fixed values of θ and t. We repeat this process for every value of t and then change the angle θ and repeat again. In this way we obtain the integrals of f over every line L in the plane. We denote by $r_f(\theta,t)$ the integral

$$r_f(\theta,t) = \int_L f(x,y)ds.$$

The function $r_f(\theta,t)$ is called the *Radon transform* of f.

19.4.2 The Central Slice Theorem

For fixed θ the function $r_f(\theta,t)$ is a function of the single real variable t; let $R_f(\theta,\omega)$ be its Fourier transform. Then

$$R_f(\theta,\omega) = \int r_f(\theta,t)e^{i\omega t}dt$$

$$= \int\int f(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta)e^{i\omega t}dsdt$$

$$= \int\int f(x,y)e^{i\omega(x\cos\theta + y\sin\theta)}dxdy = F(\omega\cos\theta, \omega\sin\theta),$$

where $F(\omega\cos\theta, \omega\sin\theta)$ is the two-dimensional Fourier transform of the function $f(x,y)$, evaluated at the point $(\omega\cos\theta, \omega\sin\theta)$; this relationship is called the *Central Slice Theorem*. For fixed θ, as we change the value of ω, we obtain the values of the function F along the points of the line making the angle θ with the horizontal axis. As θ varies in $[0,\pi)$, we get all the values of the function F. Once we have F, we can obtain f using the formula for the two-dimensional inverse Fourier transform. We conclude that we are able to determine f from its line integrals. As we shall see, inverting the Fourier transform can be implemented by combinations of frequency-domain filtering and back-projection.

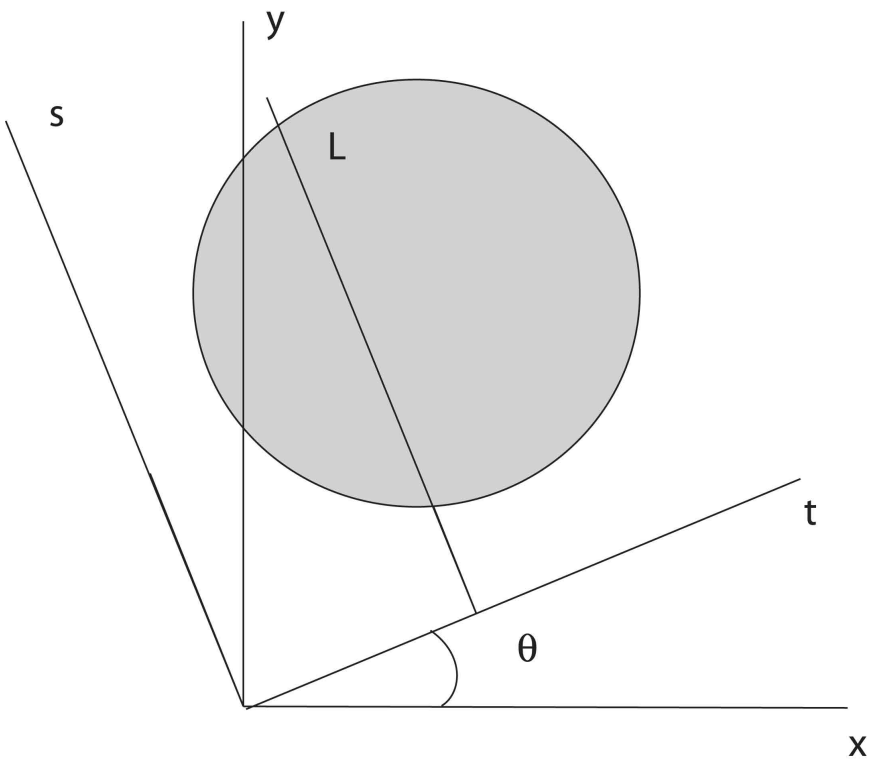

Figure 19.1: The Radon transform of f at (t, θ) is the line integral of f along line L.

Chapter 20

Transmission Tomography II

According to the Central Slice Theorem, if we have all the line integrals through the attenuation function $f(x, y)$ then we have the two-dimensional Fourier transform of $f(x, y)$. To get $f(x, y)$ we need to invert the two-dimensional Fourier transform.

20.1 Inverting the Fourier Transform

The Fourier-transform inversion formula for two-dimensional functions tells us that the function $f(x, y)$ can be obtained as

$$f(x, y) = \frac{1}{4\pi^2} \int \int F(u, v) e^{-i(xu+yv)} du dv. \tag{20.1}$$

We now derive alternative inversion formulas.

20.1.1 Back-Projection

Let $g(\theta, t)$ be any function of the variables θ and t; for example, it could be the Radon transform. As with the Radon transform, we imagine that each pair (θ, t) corresponds to one line through the x, y-plane. For each fixed point (x, y) we assign to this point the sum of the quantities $g(\theta, t)$ for every pair (θ, t) such that the point (x, y) lies on the associated line. The summing process is integration and the *back-projection* function at (x, y) is

$$BP_g(x, y) = \int g(\theta, x \cos\theta + y \sin\theta) d\theta.$$

The operation of back-projection will play an important role in what follows in this chapter.

20.1.2 Ramp Filter, then Back-project

Expressing the double integral in Equation (20.1) in polar coordinates (ω, θ), with $\omega \geq 0$, $u = \omega\cos\theta$, and $v = \omega\sin\theta$, we get

$$f(x,y) = \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{\infty} F(u,v)e^{-i(xu+yv)}\omega d\omega d\theta,$$

or

$$f(x,y) = \frac{1}{4\pi^2}\int_0^{\pi}\int_{-\infty}^{\infty} F(u,v)e^{-i(xu+yv)}|\omega| d\omega d\theta.$$

Now write

$$F(u,v) = F(\omega\cos\theta, \omega\sin\theta) = R_f(\theta,\omega),$$

where $R_f(\theta,\omega)$ is the FT with respect to t of $r_f(\theta,t)$, so that

$$\int_{-\infty}^{\infty} F(u,v)e^{-i(xu+yv)}|\omega| d\omega = \int_{-\infty}^{\infty} R_f(\theta,\omega)|\omega| e^{-i\omega t} d\omega.$$

The function $g_f(\theta,t)$ defined for $t = x\cos\theta + y\sin\theta$ by

$$g_f(\theta, x\cos\theta + y\sin\theta) = \frac{1}{2\pi}\int_{-\infty}^{\infty} R_f(\theta,\omega)|\omega| e^{-i\omega t} d\omega \tag{20.2}$$

is the result of a linear filtering of $r_f(\theta,t)$ using a *ramp filter* with transfer function $H(\omega) = |\omega|$. Then,

$$f(x,y) = \frac{1}{2\pi}\int_0^{\pi} g_f(\theta, x\cos\theta + y\sin\theta) d\theta \tag{20.3}$$

gives $f(x,y)$ as the result of a *back-projection operator*; for every fixed value of (θ,t) add $g_f(\theta,t)$ to the current value at the point (x,y) for all (x,y) lying on the straight line determined by θ and t by $t = x\cos\theta + y\sin\theta$. The final value at a fixed point (x,y) is then the average of all the values $g_f(\theta,t)$ for those (θ,t) for which (x,y) is on the line $t = x\cos\theta + y\sin\theta$. It is therefore said that $f(x,y)$ can be obtained by *filtered back-projection* (FBP) of the line-integral data.

Knowing that $f(x,y)$ is related to the complete set of line integrals by filtered back-projection suggests that, when only finitely many line integrals are available, a similar ramp filtering and back-projection can be used to estimate $f(x,y)$; in the clinic this is the most widely used method for the reconstruction of tomographic images.

20.1.3 Back-project, then Ramp Filter

There is a second way to recover $f(x,y)$ using back-projection and filtering, this time in the reverse order; that is, we back-project the Radon transform

and then ramp filter the resulting function of two variables. We begin again with the relation

$$f(x,y) = \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{\infty} F(u,v)e^{-i(xu+yv)}\omega d\omega d\theta,$$

which we write as

$$f(x,y) = \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{\infty} \frac{F(u,v)}{\sqrt{u^2+v^2}}\sqrt{u^2+v^2}e^{-i(xu+yv)}\omega d\omega d\theta$$

$$= \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{\infty} G(u,v)\sqrt{u^2+v^2}e^{-i(xu+yv)}\omega d\omega d\theta, \tag{20.4}$$

using

$$G(u,v) = \frac{F(u,v)}{\sqrt{u^2+v^2}}$$

for $(u,v) \neq (0,0)$. Equation (20.4) expresses $f(x,y)$ as the result of performing a two-dimensional ramp filtering of $g(x,y)$, the inverse Fourier transform of $G(u,v)$. We show now that $g(x,y)$ is the back-projection of the function $r_f(\theta,t)$; that is, we show that

$$g(x,y) = \frac{1}{2\pi}\int_0^{\pi} r_f(\theta, x\cos\theta + y\sin\theta)d\theta.$$

We have

$$g(x,y) = \frac{1}{4\pi^2}\int_0^{\pi}\int_{-\infty}^{\infty} G(\omega\cos\theta,\omega\sin\theta)|\omega|e^{-i\omega(x\cos\theta+y\sin\theta)}d\omega d\theta$$

$$= \frac{1}{4\pi^2}\int_0^{\pi}\int_{-\infty}^{\infty} F(\omega\cos\theta,\omega\sin\theta)e^{-i\omega(x\cos\theta+y\sin\theta)}d\omega d\theta$$

$$= \frac{1}{4\pi^2}\int_0^{\pi}\int_{-\infty}^{\infty} R_f(\theta,\omega)e^{-i\omega(x\cos\theta+y\sin\theta)}d\omega d\theta$$

$$= \frac{1}{2\pi}\int_0^{\pi} r_f(\theta, x\cos\theta + y\sin\theta)d\theta,$$

as required.

20.1.4 Radon's Inversion Formula

To get Radon's inversion formula, we need two basic properties of the Fourier transform. First, if $f(x)$ has Fourier transform $F(\gamma)$ then the derivative $f'(x)$ has Fourier transform $-i\gamma F(\gamma)$. Second, if $F(\gamma) = \text{sgn}(\gamma)$,

the function that is $\frac{\gamma}{|\gamma|}$ for $\gamma \neq 0$, and equal to zero for $\gamma = 0$, then its inverse Fourier transform is $f(x) = \frac{1}{i\pi x}$.

Writing equation (20.2) as

$$g_f(\theta, t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \omega R_f(\theta, \omega) \text{sgn}(\omega) e^{-i\omega t} d\omega,$$

we see that g_f is the inverse Fourier transform of the product of the two functions $\omega R_f(\theta, \omega)$ and $\text{sgn}(\omega)$. Consequently, g_f is the convolution of their individual inverse Fourier transforms, $i\frac{\partial}{\partial t} r_f(\theta, t)$ and $\frac{1}{i\pi t}$; that is,

$$g_f(\theta, t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{\partial}{\partial t} r_f(\theta, s) \frac{1}{t - s} ds,$$

which is the Hilbert transform of the function $\frac{\partial}{\partial t} r_f(\theta, t)$, with respect to the variable t. Radon's inversion formula is then

$$f(x, y) = \frac{1}{2\pi} \int_0^{\pi} HT(\frac{\partial}{\partial t} r_f(\theta, t)) d\theta.$$

20.2 From Theory to Practice

What we have just described is the theory. What happens in practice?

20.2.1 The Practical Problems

Of course, in reality we never have the Radon transform $r_f(\theta, t)$ for all values of its variables. Only finitely many angles θ are used, and, for each θ, we will have (approximate) values of line integrals for only finitely many t. Therefore, taking the Fourier transform of $r_f(\theta, t)$, as a function of the single variable t, is not something we can actually do. At best, we can approximate $R_f(\theta, \omega)$ for finitely many θ. From the Central Slice Theorem, we can then say that we have approximate values of $F(\omega \cos\theta, \omega \sin\theta)$, for finitely many θ. This means that we have (approximate) Fourier transform values for $f(x, y)$ along finitely many lines through the origin, like the spokes of a wheel. The farther from the origin we get, the fewer values we have, so the *coverage* in Fourier space is quite uneven. The low-spatial-frequencies are much better estimated than higher ones, meaning that we have a low-pass version of the desired $f(x, y)$. The filtered back-projection approaches we have just discussed both involve ramp filtering, in which the higher frequencies are increased, relative to the lower ones. This too can only be implemented approximately, since the data is noisy and careless ramp filtering will cause the reconstructed image to be unacceptably noisy.

20.2.2 A Practical Solution: Filtered Back-Projection

We assume, to begin with, that we have finitely many line integrals, that is, we have values $r_f(\theta, t)$ for finitely many θ and finitely many t. For each fixed θ we estimate the Fourier transform, $R_f(\theta, \omega)$. This step can be performed in various ways, and we can freely choose the values of ω at which we perform the estimation. The FFT will almost certainly be involved in calculating the estimates of $R_f(\theta, \omega)$.

For each fixed θ we multiply our estimated values of $R_f(\theta, \omega)$ by $|\omega|$ and then use the FFT again to inverse Fourier transform, to achieve a ramp filtering of $r_f(\theta, t)$ as a function of t. Note, however, that when $|\omega|$ is large, we may multiply by a smaller quantity, to avoid enhancing noise. We do this for each angle θ, to get a function of (θ, t), which we then back-project to get our final image. This is ramp-filtering, followed by back-projection, as applied to the finite data we have.

It is also possible to mimic the second approach to inversion, that is, to back-project onto the pixels each $r_f(\theta, t)$ that we have, and then to perform a ramp filtering of this two-dimensional array of numbers to obtain the final image. In this case, the two-dimensional ramp filtering involves many applications of the FFT.

There is a third approach. Invoking the Central Slice Theorem, we can say that we have finitely many approximate values of $F(u, v)$, the Fourier transform of the attenuation function $f(x, y)$, along finitely many lines through the origin. The first step is to use these values to estimate the values of $F(u, v)$ at the points of a rectangular grid. This step involves *interpolation* [233, 238]. Once we have (approximate) values of $F(u, v)$ on a rectangular grid, we perform a two-dimensional FFT to obtain our final estimate of the (discreteized) $f(x, y)$.

20.3 Summary

We have seen how the problem of reconstructing a function from line integrals arises in transmission tomography. The Central Slice Theorem connects the line integrals and the Radon transform to the Fourier transform of the desired attenuation function. Various approaches to implementing the Fourier Inversion Formula lead to filtered back-projection algorithms for the reconstruction. In x-ray tomography, as well as in PET, viewing the data as line integrals ignores the statistical aspects of the problem, and in SPECT, it ignores, as well, the important physical effects of attenuation. To incorporate more of the physics of the problem, iterative algorithms based on statistical models have been developed. We shall consider some of these algorithms later.

Chapter 21

Emission Tomography

In this chapter we describe the two modalities of emission tomography, *positron emission tomography* (PET) and *single photon emission computed tomography* (SPECT), and introduce the basic mathematical models for both.

21.1 Positron Emission Tomography

As we noted previously, 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.

Let the LOR be parameterized by the variable s, with $s = 0$ and $s = c$ denoting the two ends, and c the distance from one end to the other. For a fixed value $s = s_0$, let $P(s)$ be the probability of reaching s for a photon resulting from an emission at s_0. For small $\Delta s > 0$ the probability that a photon that reached s is absorbed in the interval $[s, s+\Delta s]$ is approximately $\mu(s)\Delta s$, where $\mu(s) \geq 0$ is the photon attenuation density at s. Then $P(s + \Delta s) \approx P(s)[1 - \mu(s)\Delta s]$, so that

$$P(s + \Delta s) - P(s) \approx -P(s)\mu(s)\Delta s.$$

Dividing by Δs and letting Δs go to zero, we get

$$P'(s) = -P(s)\mu(s).$$

It follows that

$$P(s) = e^{-\int_{s_0}^{s} \mu(t)dt}.$$

The probability that the photon will reach $s = c$ and be detected is then

$$P(c) = e^{-\int_{s_0}^{c} \mu(t)dt}.$$

Similarly, we find that the probability that a photon will succeed in reaching $s = 0$ from s_0 is

$$P(0) = e^{-\int_{0}^{s_0} \mu(t)dt}.$$

Since having one photon reach $s = 0$ and the other reach $s = c$ are independent events, their probabilities multiply, so that the probability that both photons reach their destinations and a coincident detection is recorded for this LOR is

$$e^{-\int_{0}^{c} \mu(t)dt}.$$

The expected number of coincident detections along the LOR is then proportional to

$$\int_0^c f(s)e^{-\int_0^c \mu(t)dt}ds = e^{-\int_0^c \mu(t)dt} \int_0^c f(s)ds, \tag{21.1}$$

where $f(s)$ is the intensity of radionuclide at s.

Let y_i be the number of coincidence detections associated with the ith LOR. If we are willing to equate the actual count with the expected count, and assuming we know the attenuation function $\mu(s)$, we can estimate the line integral $\int_0^c f(s)ds$ along the ith LOR as

$$\int_0^c f(s)ds = y_i e^{\int_0^c \mu(t)dt}.$$

So, once again, we have line-integral data pertaining to the function of interest.

21.2 Single-Photon Emission Tomography

We turn now to single-photon computed emission tomography (SPECT).

21.2.1 Sources of Degradation to be Corrected

We remarked earlier that there are at least three degradations that need to be corrected before the line-integral model and FBP can be successfully applied in the SPECT case [166]: attenuation, scatter, and spatially dependent resolution. There are mathematical ways to correct for both spatially

varying resolution and uniform attenuation [227]. Correcting for the more realistic non-uniform and patient-specific attenuation is more difficult and is the subject of on-going research.

Some photons never reach the detectors because they are absorbed in the body. As in the PET case, correcting for attenuation requires knowledge of the patient's body; this knowledge can be obtained by performing a transmission scan at the same time. In contrast to the PET case, the attenuation due to absorption is more difficult to correct, since it does not involve merely the line integral of the attenuation function, but a half-line integral that depends on the distribution of matter between each photon source and each detector.

While some photons are absorbed within the body, others are first deflected and then detected; this is called *scatter*. Consequently, some of the detected photons do not come from where they seem to come from. The scattered photons often have reduced energy, compared to *primary*, or non-scattered, photons, and scatter correction can be based on this energy difference; see [166].

Finally, even if there were no attenuation and no scatter, it would be incorrect to view the detected photons as having originated along a straight line from the detector. The detectors have a cone of acceptance that widens as it recedes from the detector. This results in spatially varying resolution.

It is not uncommon, however, to make the simplifying assumption that all photons detected at a given detector originated along a single line. As in the PET case previously discussed, the probability that a photon emitted at the point on the line corresponding to the variable $s = s_0$ will reach $s = c$ and be detected is then

$$P(s_0) = e^{-\int_{s_0}^{c} \mu(t)dt}.$$

If $f(s)$ is the expected number of photons emitted from point s during the scanning, then the expected number of photons detected at c and originating along this line is proportional to

$$\int_0^c f(s)e^{-\int_s^c \mu(t)dt}ds. \tag{21.2}$$

Notice the difference between the integral in Equation (21.2) and the one in Equation (21.1).

The integral in Equation (21.2) varies with the line being considered; the resulting function of lines is called the *attenuated Radon transform*.

If the attenuation function μ is constant, then the attenuated Radon transform is called the *exponential Radon transform*. Since

$$\int_s^c \mu dt = \mu(c - s),$$

the integral in (21.2) is now

$$e^{-\mu c}\int_0^c f(s)e^{\mu s}ds = e^{-\mu c}\int_0^\infty f(s)e^{-(-\mu)s}ds = e^{-\mu c}\mathcal{F}(-\mu),$$

where $\mathcal{F}$ denotes the Laplace transform of f. Since the function $f(s)$ is zero outside a bounded interval, we may safely assume that the Laplace transform is defined for all real values of the argument.

In practice, one sometimes assumes, initially, that $\mu = 0$ and that the counts at each detector are essentially integrals of f along a single line. Filtered back=projection is then used to reconstruct an image. Since the image does not reflect the effects of attenuation, it can be "corrected" during the back-projection phase.

Spatially varying resolution complicates the quantitation problem, which is the effort to determine the exact amount of radionuclide present within a given region of the body, by introducing the *partial volume effect* and *spill-over* (see [245]). To a large extent, these problems are shortcomings of reconstruction based on the line-integral model. If we assume that all photons detected at a particular detector came from points within a narrow strip perpendicular to the camera face, and we reconstruct the image using this assumption, then photons coming from locations outside this strip will be incorrectly attributed to locations within the strip (spill-over), and therefore not correctly attributed to their true source location. If the true source location also has its counts raised by spill-over, the net effect may not be significant; if, however, the true source is a hot spot surrounded by cold background, it gets no spill-over from its neighbors and its true intensity value is underestimated, resulting in the partial-volume effect. The term "partial volume" indicates that the hot spot is smaller than the region that the line-integral model offers as the source of the emitted photons. One way to counter these effects is to introduce a description of the spatially dependent blur into the reconstruction, which is then performed by iterative methods [205].

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 that incorporate more of the physics have become competitive.

21.2.2 The Discrete Model

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. In what follows we shall not distinguish the two cases, but as a linguistic shorthand, we shall refer

to 'pixels' indexed by $j = 1, ..., J$. The detectors are indexed by $i = 1, ..., I$, the count obtained at detector i is denoted y_i, and the vector $\mathbf{y} = (y_1, ..., y_I)^T$ is our data. In practice, for the fully three-dimensional case, I and J can be several hundred thousand.

We imagine that each pixel j 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; the average rate for j we denote by x_j. The goal is to determine the vector $\mathbf{x} = (x_1, ..., x_J)^T$ from $\mathbf{y}$.

21.2.3 Discrete Attenuated Radon Transform

To achieve our goal we must construct a model that relates $\mathbf{y}$ to $\mathbf{x}$. One way to do that is to discretize the attenuated Radon Transform [142, 234].

The objective is to describe the contribution to the count data from the intensity x_j at the jth pixel. We assume, for the moment, that all the radionuclide is concentrated within the jth pixel, and we compute the resulting attenuated Radon Transform. Following [142, 234], we adopt a ray model for detection, which means that corresponding to each detector is a line of acceptance and that all the counts recorded at that detector came from pixels that intersect this line. This is a simplification, of course, since each detector has a solid angle of acceptance, which leads to depth-dependent blur.

For notational simplicity, we suppose that the line of acceptance associated with the ith detector is parameterized by arc-length $s \geq 0$, with $s = c > 0$ corresponding to the point closest to the detector, within the body, $s = 0$ corresponding to the point farthest from the detector, at which the line leaves the body, $s = b < c$ the point closest to the detector within the jth pixel, and $s = a < b$ the point farthest from the detector at which the line leaves the jth pixel. The length of the intersection of the jth pixel with the line is then $d_{ij} = b - a$.

We are assuming that all the radionuclide is within the jth pixel, with intensity distribution (proportional to) x_j, so the value at detector i of the attenuated Radon Transform is

$$A_{ij} = \int_a^b x_j e^{-\int_s^c \mu(t)dt} ds. \tag{21.3}$$

We assume that the attenuation is uniformly equal to $\mu_j \geq 0$ within the jth pixel, so we can write

$$A_{ij} = \int_a^b x_j e^{-\int_s^b \mu_j dt - \int_b^c \mu(t)dt} ds,$$

or

$$A_{ij} = x_j e^{-\int_b^c \mu(t)dt} \int_a^b e^{(s-b)\mu_j} ds.$$

If $\mu_j = 0$, then we have

$$A_{ij} = x_j e^{-\int_b^c \mu(t)dt} d_{ij},$$

while if $\mu_j > 0$ we have

$$A_{ij} = \left(x_j e^{-\int_b^c \mu(t)dt} d_{ij}\right) S_{ij},$$

where

$$S_{ij} = \frac{1}{d_{ij}} \int_a^b e^{(b-s)\mu_j} ds = \frac{1}{\mu_j d_{ij}} (1 - e^{-\mu_j d_{ij}}).$$

We can then write

$$A_{ij} = x_j W_{ij},$$

for each j and i.

Since the function

$$g(t) = \frac{1}{t}(1 - e^{-t})$$

is positive for positive t, $g(0) = 1$, and $g(+\infty) = 0$, it is reasonable to view S_{ij} as the survival proportion associated with the jth pixel and the line from the ith detector. Expanding the exponential in S_{ij} in a power series, we find that

$$S_{ij} = \frac{1}{\mu_j d_{ij}} (1 - e^{-\mu_j d_{ij}}) \approx 1 - \frac{1}{2}\mu_j d_{ij},$$

so that the loss proportion is approximately $\frac{1}{2}\mu_j d_{ij}$. If we were to adopt the decaying exponential model for a photon surviving its passage through the jth pixel, and assume all the radionuclide was initially at the far side of the jth pixel, we would replace S_{ij} with $e^{-\mu_j d_{ij}}$, which is approximately $1 - \mu_j d_{ij}$, so that the loss proportion is approximately $\mu_j d_{ij}$. This is twice the loss proportion that we got using the other model, and is larger because we are assuming that all the radionuclide in the jth pixel has to attempt to travel through the entire jth pixel, whereas, due to the spreading of the radionuclide throughout the pixel, the average journey through the pixel is only half of the length d_{ij}.

Having found the values W_{ij}, we form the matrix W having these entries and then find a non-negative solution of the system of equations $Wx = y$, using one of a number of iterative algorithms, including the EMML. Contrary to what is stated in [234], it may not be appropriate to consider W_{ij} as the probability that a photon emitted at the jth pixel is detected at the ith detector, even though $0 \leq W_{ij} \leq 1$ for each i and j. If viewed that way, it would be the case that

$$\sum_{i=1}^{I} W_{ij}$$

would be the probability of detecting a photon emitted from the jth pixel; we have no guarantee, however, that this sum is not greater than one.

It is significant that the authors in [234] realize that the EMML iterative algorithm can be used to find a non-negative solution of $Wx = y$, even though no stochastic model for the data is assumed in their derivation. Their development involves discretizing the attenuated Radon Transform, which involves no randomness, and viewing the count data as approximate values of this discrete function.

There is another approach that can be used to relate the count data to the intensity levels x_j. This other approach is based on a stochastic model, as we describe next.

21.2.4 A Stochastic Model

Another way to relate the count data to the intensities x_j is to adopt the model of *independent Poisson emitters*. For $i = 1, ..., I$ and $j = 1, ..., J$, denote by Z_{ij} the random variable whose value is to be the number of photons emitted from pixel j, and detected at detector i, during the scanning time. We assume that the members of the collection $\{Z_{ij} | i = 1, ..., I, j = 1, ..., J\}$ are independent. In keeping with standard practice in modeling radioactivity, we also assume that the Z_{ij} are Poisson-distributed.

Generally, the signal-to-noise ratio (SNR) is the ratio of the mean of a distribution to its standard deviation (the square root of the variance). In the case of the Poisson distribution, the variance and the mean are the same, so the SNR is the square root of the mean; therefore, the higher the mean the higher the SNR.

We assume that Z_{ij} is a Poisson random variable whose mean value (and variance) is $\lambda_{ij} = P_{ij}x_j$. Here the $x_j \geq 0$ is the average rate of emission from pixel j, as discussed previously, and $P_{ij} \geq 0$ is the probability that a photon emitted from pixel j will be detected at detector i. The calculation of the P_{ij} can be quite similar to the derivation of the W_{ij} in the previous subsection, with the exception that we do need to have

$$\sum_{i=1}^{I} P_{ij} \leq 1.$$

We then define the random variables $Y_i = \sum_{j=1}^{J} Z_{ij}$, the total counts to be recorded at detector i; our actual count y_i is then the observed value of the random variable Y_i. Note that the actual values of the individual Z_{ij} are not observable.

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.

Having found the P_{ij}, we take P to be the matrix with these entries. Since Px is the vector of expected counts at the various detectors, and y is the vector of actual counts, trying to find a non-negative solution of the system $y = Px$ may not seem completely reasonable. However, this is what several well known iterative algorithms do, even ones such as the EMML that were not originally designed for this purpose.

21.2.5 Reconstruction as Parameter Estimation

The goal is to estimate the distribution of radionuclide intensity by calculating the vector $\mathbf{x}$. The entries of $\mathbf{x}$ are 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 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 know what the P_{ij} are. These values 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 i and pixel j, 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 values 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 P_{ij} in the absence of motion, and correcting for the effects of motion, are important parts of SPECT image reconstruction.

21.3 Relative Advantages

In [197], Ollinger and Fessler discuss some of the relative advantages of these two modes of emission tomography.

Attenuation, which is primarily the scattering of photons by the body to locations outside the field of view of the detecting cameras, is harder to correct in SPECT. The radiopharmaceuticals used in SPECT must incorporate heavy isotopes, such as thallium and technetium; since these do not occur naturally in biologically active molecules, the synthesis of physiologically useful tracers is a challenge. In contrast, in PET the positron-emitting isotopes of carbon, nitrogen, oxygen and fluorine that are used occur natu-

rally in many compounds of biological interest and can therefore be easily incorporated into useful radiopharmaceuticals.

Because collimation is performed by the computer in PET, while SPECT must employ lead collimators, which absorb many of the photons, the sensitivity of the detecting gamma cameras in SPECT is reduced, in comparison to PET.

On the other side of the balance sheet, the short half-life of most positron-emitting isotopes necessitates an on-site cyclotron, while the isotopes used in SPECT have longer half-lives and can be stored. Also, the scanners for PET are more expensive than those used in SPECT.

At any given time, computer speed limits the size of the problem that can be dealt with. While 2D reconstructions are clinically feasible, fully 3D imaging (not to mention dynamic, 4D imaging) poses more of a challenge, hence the need for continuing algorithm development.

Chapter 22

List-Mode Reconstruction in PET

22.1 Why List-Mode Processing?

In PET 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. When a LOR is recorded, it is assumed that a positron was emitted somewhere along that line.

In modern PET scanners the number of pairs of detectors, and therefore, the number of potential LOR, often exceeds the number of detections; the count recorded at any single i is typically one or zero. It makes sense, therefore, to record the data as a list of those LOR corresponding to a detection; this is list-mode data.

22.2 Correcting for Attenuation in PET

In SPECT attenuation correction is performed by modifying the probabilities P_{ij}. In PET the situation is at once simpler and more involved.

Let a given LOR be parameterized by the variable s, with $s = 0$ and $s = c$ denoting the two ends, and c the distance from one end to the other. For a fixed value $s = s_0$, let $P(s)$ be the probability of reaching s for a photon resulting from an emission at s_0. For small $\Delta s > 0$ the probability

that a photon that reached s is absorbed in the interval $[s, s + \Delta s]$ is approximately $\mu(s)\Delta s$, where $\mu(s) \geq 0$ is the photon attenuation density at s. Then $P(s + \Delta s) \approx P(s)[1 - \mu(s)\Delta s]$, so that

$$P(s + \Delta s) - P(s) \approx -P(s)\mu(s)\Delta s.$$

Dividing by Δs and letting Δs go to zero, we get

$$P'(s) = -P(s)\mu(s).$$

It follows that

$$P(s) = e^{-\int_{s_0}^{s} \mu(t)dt}.$$

The probability that the photon will reach $s = c$ and be detected is then

$$P(c) = e^{-\int_{s_0}^{c} \mu(t)dt}.$$

Similarly, we find that the probability that a photon will succeed in reaching $s = 0$ from s_0 is

$$P(0) = e^{-\int_{0}^{s_0} \mu(t)dt}.$$

Since having one photon reach $s = 0$ and the other reach $s = c$ are independent events, their probabilities multiply, so that the probability that both photons reach their destinations and a coincident detection is recorded for this LOR is

$$e^{-\int_{0}^{c} \mu(t)dt}.$$

The expected number of coincident detections along the LOR is then proportional to

$$\int_0^c f(s)e^{-\int_0^c \mu(t)dt}ds = e^{-\int_0^c \mu(t)dt} \int_0^c f(s)ds, \tag{22.1}$$

where $f(s)$ is the intensity of radionuclide at s.

For each LOR i and each pixel or voxel j, let A_{ij} be the *geometric probability* that an emission at j will result in two photons traveling along the LOR i. The probability A_{ij} is unrelated to the attenuation presented by the body of the patient. Then the probability that an emission at j will result in the LOR i being added to the list is

$$P_{ij} = a_i A_{ij},$$

where

$$a_i = e^{-\int_i \mu(s)ds},$$

and the integral is the line integral along the line segment associated with the LOR i. We then perform attenuation correction by using the probabilities P_{ij} in the reconstruction.

Note that, if the number I of potential LOR is not too large and the entries of the data vector y are not simply zero or one, we might correct for attenuation by replacing each y_i with y_i/a_i, which is approximately the count we would have seen for the LOR i if there had been no attenuation. However, in the more typical case of large I and zero or one values for the y_i, this approach does not make much sense. The effect of attenuation now is to prevent certain i from being recorded, not to diminish the values of the positive y_i of the LOR that were recorded. Therefore, at least in theory, it makes more sense to correct for attenuation by using the P_{ij}. There is an additional complication, though.

In list-mode processing, I, the number of potential LOR, is much larger than the size of the list. To employ the EMML algorithm or one of its block-iterative variants, we need to calculate the probabilities associated with those LOR on the list, but it is costly to do this for all the potential LOR; we do need to compute the sensitivities, or probabilities of detection, for each pixel, however. If we consider only the geometry of the scanner, calculating the sensitivities for each pixel is not difficult and can be done once and used repeatedly; it is much more problematic if we must include the patient-specific attenuation. For this reason, it makes sense, practically speaking, to correct for attenuation in list-mode PET by replacing y_i with y_i/a_i for those y_i equal to one. The reconstruction is probably much the same, either way.

22.3 Modeling the Possible LOR

We can model the potential LOR simply as pairs of detectors, so that I, the number of potential LOR, is very large, but finite, and finite probability vectors, rather than probability density functions, suffice in forming the likelihood function. The EMML algorithm applies directly to this list-mode model. This is the approach adopted by Huesman *et al.* [158].

Alternatively, one can assume that the end-point coordinates form a continuum, so that the set of potential LOR is uncountably infinite. Now we need probability density functions to form the likelihood function. This method, adopted by Parra and Barrett [201], makes the application of the EMML algorithm more complicated, as discussed in [58].

22.4 EMML: The Finite LOR Model

In this section we discuss the EMML iterative algorithm for list-mode reconstruction based on the finite model.

Let the list of recorded LOR be $\{i_1, ..., i_M\}$ and let

$$Q_{mj} = P_{i_m,j},$$

for $m = 1, ..., M$. Since the values of the y_i are typically zero or one, the i_m are typically distinct, but this is not essential here. The EMML iteration becomes

$$x_j^{k+1} = x_j^k s_j^{-1} \sum_{m=1}^{M} Q_{mj} \Big(\frac{1}{(Qx^k)_m} \Big). \tag{22.2}$$

Note that we still need to use the sensitivity values

$$s_j = \sum_{i=1}^{I} P_{ij},$$

which are the probabilities of detection. However, for imaging the radionuclide we do not need to calculate the s_j by first determining each of the P_{ij}; we need only that the $s_j > \sum_{m=1}^{M} Q_{mj}$ for each j and that the relative values of the various s_j be reasonably accurate. For quantitation, though, accurate absolute values of the s_j are needed.

22.5 List-mode RBI-EMML

We turn now to the block-iterative versions of EMML. For $n = 1, ..., N$ let C_n consist of all indices m such that the LOR i_m on the list is also in B_n. The list-mode BI-EMML (LMBI-EMML) has the iterative step

$$x_j^k = (1 - \gamma_n \delta_j s_{nj}) x_j^{k-1} + x_j^k \gamma_n \delta_j \sum_{m \in C_n} P_{ij} \Big(\frac{1}{(Qx^k)_m} \Big), \tag{22.3}$$

with $\gamma > 0$ chosen so that

$$s_{nj} \delta_j \gamma_n \leq 1.$$

When we select $\delta_j = s_j^{-1}$, we must then have $\gamma_n \leq \mu_n^{-1}$. When we have $\delta_j = 1$, we need $\gamma_n \leq m_n^{-1}$. Generally speaking, the larger the γ_n the faster the convergence. The *rescaled* LMBI-EMML (LMRBI-EMML) uses the largest values of γ_n consistent with these constraints.

Note that, as previously, we need s_j and now we also need s_{nj}. As before, though, we do not need to specify each of the P_{ij} to obtain reasonable choices for these values.

22.6 The Row-action LMRBI-EMML: LMEMART

The row-action or *event-by-event* version of the RBI-EMML algorithm, the LMEMART, is a special case of the LMRBI-EMML in which, for $m = 1, ..., M$, each LOR i_m on the list forms its own block or subset, denoted

C_m. Another way to say this is that we choose the original blocks B_n so that no B_n contains more than one i_m. For clarity, we shall assume that the blocks B_n are chosen so that $B_m = \{i_m\}$ and $C_m = \{m\}$, for $m = 1, ..., M$. We then let B_{M+1} consist of all the i not equal to some I_m on the list, and $N = M + 1$. Therefore, for $n = 1, ..., M$, we have

$$s_{nj} = Q_{nj}.$$

In the LMEMART each iteration employs a single member of the list and we cycle through the list repeatedly. The iteration index is now $m = 1, ..., M$, with $m = m(k) = k(\mathrm{mod}\, M) + 1$.

The LMEMART has the iterative step

$$x_j^{k+1} = (1 - \gamma_m \delta_j Q_{mj}) x_j^k + x_j^k \gamma_m \delta_j Q_{mj} \Big(\frac{1}{(Qx^k)_m} \Big), \tag{22.4}$$

with $Q_{mj}\delta_j\gamma_m \leq 1$.

22.7 EMML: The Continuous LOR Model

When the end points of the potential LOR are allowed to take on values in a continuum, the likelihood function involves probability density functions, rather than finite probabilities. This poses a difficulty, in that the values of probability density functions can be any non-negative real number; only their integrals are required to be one. As a result, the convergence theory for the EMML algorithm and its various block-iterative versions does not apply unchanged.

For each pixel index j, let $f_j(\cdot)$ be the probability density function (pdf) whose domain is the (uncountably infinite) set of potential LOR with the property that the probability that an emission at j results in an LOR from the set S being recorded is the integral of f_j over S. With x_j the expected number of emissions from j during the scanning time, and

$$x_+ = \sum_{j=1}^{J} x_j,$$

the probability that an emission came from j, given that an emission has happened, is x_j/x_+. Therefore, the probability that an LOR in the set S will be recorded, given that an emission has happened, is the integral over S of the pdf

$$f(\cdot) = \frac{1}{x_+} \sum_{j=1}^{J} x_j f_j(\cdot).$$

For each j let d_j be the probability that an emission from j will be detected, and let

$$d = \frac{1}{x_+} \sum_{j=1}^{J} x_j d_j$$

be the probability that an emission will be detected.

The number of items on the list, M, is also a random variable, which we model as having a Poisson distribution with mean value dx_+. Therefore, the probability of M is

$$p(M) = \exp(-x_+ d)(x_+ d)^M / M!.$$

Given the list of recorded LOR, the likelihood function is then

$$L(x) = p(M) \prod_{m=1}^{M} f(i_m),$$

and the log likelihood function to be maximized is

$$LL(x) = -x_+ d + \sum_{m=1}^{M} \log(Px)_m,$$

where the matrix P has entries

$$P_{mj} = f_j(i_m).$$

Note that

$$(Px)_m = \sum_{j=1}^{J} P_{mj} x_j,$$

so that

$$\sum_{m=1}^{M} (Px)_m = \sum_{j=1}^{J} \Big(\sum_{m=1}^{M} P_{mj} \Big) x_j = \sum_{j=1}^{J} c_j x_j,$$

for

$$c_j = \sum_{m=1}^{M} P_{mj}.$$

Maximizing the log likelihood function is equivalent to minimizing

$$KL(u, Px) - \sum_{m=1}^{M} (Px)_m + x_+ d + \text{constants},$$

where u is the vector whose entries are all one, and therefore equivalent to minimizing

$$F(x) = KL(u, Px) + \sum_{j=1}^{J} (d_j - c_j) x_j.$$

The EMML algorithm itself will minimize only $KL(u, Px)$. The basic problem now is that we have values of probability density functions and the quantities c_j, which can be any positive real numbers, are unrelated to the detectability or sensitivity d_j.

It was shown in [58] that the EMML algorithm can be modified to provide a convergent iterative method for minimizing $F(x)$. This modified EMML algorithm has the iterative step

$$x_j^{k+1} = x_j^k d_j^{-1} \sum_{m=1}^{M} \Big(\frac{1}{(Px^k)_m} \Big).$$

For the finite model, as in [158], this is just the usual EMML and convergence follows from known results, but for the continuous model, as in [201], this iterative scheme falls outside the EMML framework and convergence needed to be established, as in [58].

Just as the EMML algorithm must be modified before it can be applied to the continuous model, we must adapt the block-iterative versions as well; see [58] for details.

Chapter 23

Magnetic Resonance Imaging

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. This is achieved by forcing the individual spinning nuclei to emit signals that, while too weak to be detected alone, are detectable in the aggregate. Fourier-transform estimation and extrapolation techniques play a major role in the rapidly expanding field of magnetic resonance imaging [247, 143].

23.1 Slice Isolation

When the external magnetic field is the *static field* $B_0\mathbf{k}$, that is, the magnetic field has strength B_0 and axis $\mathbf{k} = (0, 0, 1)$, then the Larmor frequency is the same everywhere and equals $\omega_0 = \gamma B_0$, where γ is the gyromagnetic constant. If, instead, we impose an external magnetic field $(B_0 + G_z(z - z_0))\mathbf{k}$, for some constant G_z, then the Larmor frequency is ω_0 only within the plane $z = z_0$. This external field now includes a *gradient field*.

23.2 Tipping

When a magnetic dipole moment that is aligned with $\mathbf{k}$ is given a component in the x, y-plane, it begins to precess around the z-axis, with frequency equal to its Larmor frequency. To create this x, y-plane component, we ap-

ply a *radio-frequency field* (rf field)

$$H_1(t)(\cos(\omega t)\mathbf{i} + \sin(\omega t)\mathbf{j}).$$

The function $H_1(t)$ typically lasts only for a short while, and the effect of imposing this rf field is to tip the aligned magnetic dipole moment axes away from the z-axis, initiating precession. Those dipole axes that tip most are those whose Larmor frequency is ω. Therefore, if we first isolate the slice $z = z_0$ and then choose $\omega = \omega_0$, we tip primarily those dipole axes within the plane $z = z_0$. The dipoles that have been tipped ninety degrees into the x, y-plane generate the strongest signal. How much tipping occurs also depends on $H_1(t)$, so it is common to select $H_1(t)$ to be constant over the time interval $[0, \tau]$, and zero elsewhere, with integral $\frac{\pi}{2\gamma}$. This $H_1(t)$ is called a $\frac{\pi}{2}$-pulse, and tips those axes with Larmor frequency ω_0 into the x, y-plane.

23.3 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.

23.3.1 The Line-Integral Approach

Suppose that we have isolated the plane $z = z_0$ and tipped the aligned axes using a $\frac{\pi}{2}$-pulse. After the tipping has been completed, we introduce an external field $(B_0 + G_x x)\mathbf{k}$, so that now the Larmor frequency of dipoles within the plane $z = z_0$ is $\omega(x) = \omega_0 + \gamma G_x x$, which depends on the x-coordinate of the point. The result is that the component of the received signal associated with the frequency $\omega(x)$ is due solely to those dipoles having that x coordinate. Performing an FFT of the received signal gives us line integrals of the density function along lines in the x, y-plane having fixed x-coordinate.

More generally, if we introduce an external field $(B_0 + G_x x + G_y y)\mathbf{k}$, the Larmor frequency is constant at $\omega(x, y) = \omega_0 + \gamma(G_x x + G_y y) = \omega_0 + \gamma s$ along lines in the x, y-plane with equation

$$G_x x + G_y y = s.$$

Again performing an FFT on the received signal, we obtain the integral of the density function along these lines. 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 we can obtain the Fourier transform of the density function by performing a one-dimensional Fourier transform with respect to the variable s. For each fixed (G_x, G_y) we obtain this Fourier transform along a ray through the origin. By varying the (G_x, G_y) we get the entire Fourier transform. The desired density function is then obtained by Fourier inversion.

23.3.2 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 to have Fourier-transform values on the points of a rectangular grid. We can obtain this by selecting the gradient fields to achieve *phase encoding*.

Suppose that, after the tipping has been performed, we impose the external field $(B_0+G_y y)\mathbf{k}$ for T seconds. The effect is to alter the precession frequency from ω_0 to $\omega(y) = \omega_0 + \gamma G_y y$. A harmonic $e^{i\omega_0 t}$ is changed to

$$e^{i\omega_0 t} e^{i\gamma G_y y t},$$

so that, after T seconds,we have

$$e^{i\omega_0 T} e^{i\gamma G_y y T}.$$

For $t \geq T$, the harmonic $e^{i\omega_0 t}$ returns, but now it is

$$e^{i\omega_0 t} e^{i\gamma G_y y T}.$$

The effect is to introduce a phase shift of $\gamma G_y y T$. Each point with the same y-coordinate has the same phase shift.

After time T, when this gradient field is turned off, we impose a second external field, $(B_0 + G_x x)\mathbf{k}$. Because this gradient field alters the Larmor frequencies, at times $t \geq T$ the harmonic $e^{i\omega_0 t} e^{i\gamma G_y y T}$ is transformed into

$$e^{i\omega_0 t} e^{i\gamma G_y y T} e^{i\gamma G_x x t}.$$

The received signal is now

$$S(t) = e^{i\omega_0 t} \int\int \rho(x, y) e^{i\gamma G_y y T} e^{i\gamma G_x x t} dx dy,$$

where $\rho(x, y)$ is the value of the proton density function at (x, y). Removing the $e^{i\omega_0 t}$ factor, we have

$$\int\int \rho(x, y) e^{i\gamma G_y y T} e^{i\gamma G_x x t} dx dy,$$

which is the Fourier transform of $\rho(x, y)$ at the point $(\gamma G_x t, \gamma G_y T)$. By selecting equi-spaced values of t and altering the G_y, we can get the Fourier transform values on a rectangular grid.

23.4 The General Formulation

The external magnetic field generated in the MRI scanner is generally described by

$$H(r,t) = (H_0 + \mathbf{G}(t)\cdot\mathbf{r})\mathbf{k} + H_1(t)(\cos(\omega t)\mathbf{i} + \sin(\omega t)\mathbf{j}). \tag{23.1}$$

The vectors $\mathbf{i}, \mathbf{j}$, and $\mathbf{k}$ are the unit vectors along the coordinate axes, and $\mathbf{r} = (x, y, z)$. The vector-valued function $\mathbf{G}(t) = (G_x(t), G_y(t), G_z(t))$ produces the *gradient field*

$$\mathbf{G}(t)\cdot\mathbf{r}.$$

The magnetic field component in the x, y plane is the *radio frequency* (rf) field.

If $\mathbf{G}(t) = 0$, then the Larmor frequency is ω_0 everywhere. Using $\omega = \omega_0$ in the rf field, with a $\frac{\pi}{2}$-pulse, will then tip the aligned axes into the x, y-plane and initiate precession. If $\mathbf{G}(t) = \theta$, for some direction vector θ, then the Larmor frequency is constant on planes $\theta\cdot\mathbf{r} = s$. Using an rf field with frequency $\omega = \gamma(H_0 + s)$ and a $\frac{\pi}{2}$-pulse will then tip the axes in this plane into the x, y-plane. The strength of the received signal will then be proportional to the integral, over this plane, of the proton density function. Therefore, the measured data will be values of the three-dimensional Radon transform of the proton density function, which is related to its three-dimensional Fourier transform by the Central Slice Theorem. Later, we shall consider two more widely used examples of $\mathbf{G}(t)$.

23.5 The Received Signal

We assume now that the function $H_1(t)$ is a *short* $\frac{\pi}{2}$*-pulse*, that is, it has constant value over a short time interval $[0, \tau]$ and has integral $\frac{\pi}{2\gamma}$. The received signal produced by the precessing magnetic dipole moments is approximately

$$S(t) = \int_{R^3} \rho(\mathbf{r}) \exp(-i\gamma(\int_0^t \mathbf{G}(s)ds)\cdot\mathbf{r}) \exp(-t/T_2)d\mathbf{r}, \tag{23.2}$$

where $\rho(\mathbf{r})$ is the proton density function, and T_2 is the *transverse* or *spin-spin* relaxation time. The vector integral in the exponent is

$$\int_0^t \mathbf{G}(s)ds = (\int_0^t G_x(s)ds, \int_0^t G_y(s)ds, \int_0^t G_z(s)ds).$$

Now imagine approximating the function $G_x(s)$ over the interval $[0, t]$ by a step function that is constant over small subintervals, that is, $G_x(s)$ is approximately $G_x(n\Delta)$ for s in the interval $[n\Delta, (n+1)\Delta)$, with $n =$

$1, ..., N$ and $\Delta = \frac{t}{N}$. During the interval $[n\Delta, (n+1)\Delta)$, the presence of this gradient field component causes the phase to change by the amount $x\gamma G_x(n\Delta)\Delta$, so that by the time we reach $s = t$ the phase has changed by

$$x \sum_{n=1}^{N} G_x(n\Delta)\Delta,$$

which is approximately $x \int_0^t G_x(s)ds$.

23.5.1 An Example of $\mathbf{G}(t)$

Suppose now that $g > 0$ and θ is an arbitrary direction vector. Let

$$\mathbf{G}(t) = g\theta, \text{ for } \tau \leq t, \tag{23.3}$$

and $\mathbf{G}(t) = 0$ otherwise. Then the received signal $S(t)$ is

$$S(t) = \int_{R^3} \rho(\mathbf{r}) \exp(-i\gamma g(t-\tau)\theta \cdot \mathbf{r}) d\mathbf{r}$$

$$= (2\pi)^{3/2} \hat{\rho}(\gamma g(t-\tau)\theta), \tag{23.4}$$

for $\tau \leq t << T_2$, where $\hat{\rho}$ denotes the three-dimensional Fourier transform of the function $\rho(\mathbf{r})$.

From Equation (23.4) we see that, by selecting different direction vectors and by sampling the received signal $S(t)$ at various times, we can obtain values of the Fourier transform of ρ along lines through the origin in the Fourier domain, called *k-space*. If we had these values for all θ and for all t we would be able to determine $\rho(\mathbf{r})$ exactly. Instead, we have much the same problem as in transmission tomography; only finitely many θ and only finitely many samples of $S(t)$. Noise is also a problem, because the resonance signal is not strong, even though the external magnetic field is.

We may wish to avoid having to estimate the function $\rho(\mathbf{r})$ from finitely many noisy values of its Fourier transform. We can do this by selecting the gradient field $\mathbf{G}(t)$ differently.

23.5.2 Another Example of $\mathbf{G}(t)$

The vector-valued function $\mathbf{G}(t)$ can be written as

$$\mathbf{G}(t) = (G_1(t), G_2(t), G_3(t)).$$

Now we let

$$G_2(t) = g_2,$$

and

$$G_3(t) = g_3,$$

for $0 \leq t \leq \tau$, and zero otherwise, and

$$G_1(t) = g_1,$$

for $\tau \leq t$, and zero otherwise. This means that only $H_0\mathbf{k}$ and the rf field are present up to time τ, and then the rf field is shut off and the gradient field is turned on. Then, for $t \geq \tau$, we have

$$S(t) = (2\pi)^{3/2}\hat{M}_0(\gamma(t-\tau)g_1, \gamma\tau g_2, \gamma\tau g_3).$$

By selecting

$$t_n = n\Delta t + \tau, \text{for } n = 1, ..., N,$$

$$g_{2k} = k\Delta g,$$

and

$$g_{3i} = i\Delta g,$$

for $i, k = -m, ..., m$ we have values of the Fourier transform, $\hat{M}_0$, on a Cartesian grid in three-dimensional k-space. The proton density function, ρ, can then be approximated using the fast Fourier transform.

Although the reconstruction employs the FFT, obtaining the Fourier-transform values on the Cartesian grid can take time. An abdominal scan can last for a couple of hours, during which the patient is confined, motionless and required to hold his or her breath repeatedly. Recent work on *compressed sensing* is being applied to reduce the number of Fourier-transform values that need to be collected, and thereby reduce the scan time [250, 184].

23.6 Compressed Sensing in Image Reconstruction

As we have seen, 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.

23.6.1 Incoherent Bases

The objective in CS is to exploit sparseness to reconstruct a vector f in R^J from relatively few linear functional measurements [107].

Let $U = \{u^1, u^2, ..., u^J\}$ and $V = \{v^1, v^2, ..., v^J\}$ be two orthonormal bases for R^J, with all members of R^J represented as column vectors. For $i = 1, 2, ..., J$, let

$$\mu_i = \max_{1 \leq j \leq J} \{|\langle u^i, v^j \rangle|\}$$

and

$$\mu(U, V) = \max\{mu_i \, | i = 1, ..., I\}.$$

We know from Cauchy's Inequality that

$$|\langle u^i, v^j \rangle| \leq 1,$$

and from Parseval's Equation

$$\sum_{j=1}^{J} |\langle u^i, v^j \rangle|^2 = ||u^i||^2 = 1.$$

Therefore, we have

$$\frac{1}{\sqrt{J}} \leq \mu(U, V) \leq 1.$$

The quantity $\mu(U, V)$ is the *coherence* measure of the two bases; the closer $\mu(U, V)$ is to the lower bound of $\frac{1}{\sqrt{J}}$, the more *incoherent* the two bases are.

Let f be a fixed member of R^J; we expand f in the V basis as

$$f = x_1 v^1 + x_2 v^2 + ... + x_J v^J.$$

We say that the coefficient vector $x = (x_1, ..., x_J)$ is S-sparse if S is the number of non-zero x_j.

23.6.2 Exploiting Sparseness

If S is small, most of the x_j are zero, but since we do not know which ones these are, we would have to compute all the linear functional values

$$x_j = \langle f, v^j \rangle$$

to recover f exactly. In fact, the smaller S is, the harder it would be to learn anything from randomly selected x_j, since most would be zero. The idea in CS is to obtain measurements of f with members of a different orthonormal basis, which we call the U basis. If the members of U are very

much like the members of V, then nothing is gained. But, if the members of U are quite unlike the members of V, then each inner product measurement

$$y_i = \langle f, u^i \rangle = f^T u^i$$

should tell us something about f. If the two bases are sufficiently incoherent, then relatively few y_i values should tell us quite a bit about f. Specifically, we have the following result due to Candès and Romberg [67]: suppose the coefficient vector x for representing f in the V basis is S-sparse. Select uniformly randomly $M \leq J$ members of the U basis and compute the measurements $y_i = \langle f, u^i \rangle$. Then, if M is sufficiently large, it is highly probable that $z = x$ also solves the problem of minimizing the one-norm

$$||z||_1 = |z_1| + |z_2| + ... + |z_J|,$$

subject to the conditions

$$y_i = \langle g, a^i \rangle = g^T u^i,$$

for those M randomly selected u^i, where

$$g = z_1 v^1 + z_2 v^2 + ... + z_J v^J.$$

This can be formulated as a linear programming problem. The smaller $\mu(U, V)$ is, the smaller the M is permitted to be without reducing the probability of perfect reconstruction.

Chapter 24

Intensity Modulated Radiation Therapy

In *intensity modulated radiation therapy* (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). In our discussion here we follow Censor et al. [75].

24.1 The Forward and Inverse Problems

The *forward problem* is to calculate the radiation dose absorbed in the irradiated tissue based on a given distribution of the beamlet intensities. The *inverse problem* is to find a distribution of beamlet intensities, the radiation intensity map, that will result in a clinically acceptable dose distribution. One important constraint is that the radiation intensity map must be implementable, that is, it is physically possible to produce such an intensity map, given the machine's design. There will be limits on the change in intensity between two adjacent beamlets, for example.

24.2 Equivalent Uniform Dosage

The *equivalent uniform dose* (EUD) for tumors is the biologically equivalent dose which, if given uniformly, will lead to the same cell-kill within the tumor volume as the actual non-uniform dose.

24.3 Constraints

Constraints on the EUD received by each voxel of the body are described in *dose space*, the space of vectors whose entries are the doses received at each voxel. Constraints on the deliverable radiation intensities of the beamlets are best described in *intensity space*, the space of vectors whose entries are the intensity levels associated with each of the beamlets. The constraints in dose space will be upper bounds on the dosage received by the OAR and lower bounds on the dosage received by the PTV. The constraints in intensity space are limits on the complexity of the intensity map and on the delivery time, and, obviously, that the intensities be non-negative. Because the constraints operate in two different domains, it is convenient to formulate the problem using these two domains. This leads to a split-feasibility problem.

24.4 The Multi-Set Split-Feasibilty-Problem Model

The *split feasibility problem* (SFP) is to find an x in a given closed convex subset C of R^J such that Ax is in a given closed convex subset Q of R^I, where A is a given real I by J matrix. Because the constraints are best described in terms of several sets in dose space and several sets in intensity space, the SFP model needs to be expanded into the *multi-set* SFP (MSSFP) [77].

It is not uncommon to find that, once the various constraints have been specified, there is no intensity map that satisfies them all. In such cases, it is desirable to find an intensity map that comes as close as possible to satisfying all the constraints. One way to do this, as we shall see, is to minimize a *proximity function*.

24.5 Formulating the Proximity Function

For $i = 1, ..., I$, and $j = 1, ..., J$, let $h_i \geq 0$ be the dose absorbed by the i-th voxel of the patient's body, $x_j \geq 0$ be the intensity of the j-th beamlet of radiation, and $D_{ij} \geq 0$ be the dose absorbed at the i-th voxel due to a unit intensity of radiation at the j-th beamlet. The non-negative matrix D with entries D_{ij} is the *dose influence matrix*.

In intensity space, we have the obvious constraints that $x_j \geq 0$. 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 dosage delivered to those regions designated as *planned tar-*

get volumes (PTV), and an upper bound on the dosage delivered to those regions designated as *organs at risk* (OAR).

24.6 Equivalent Uniform Dosage Functions

Suppose that S_t is either a PTV or a OAR, and suppose that S_t contains N_t voxels. For each dosage vector $h = (h_1, ..., h_I)^T$ define the *equivalent uniform dosage function* (EUD-function) $e_t(h)$ by

$$e_t(h) = (\frac{1}{N_t} \sum_{i \in S_t} (h_i)^\alpha)^{1/\alpha}, \tag{24.1}$$

where $0 < \alpha < 1$ if S_t is a PTV, and $\alpha > 1$ if S_t is an OAR. The function $e_t(h)$ is convex, for h nonnegative, when S_t is an OAR, and $-e_t(h)$ is convex, when S_t is a PTV. The constraints in dosage space take the form

$$e_t(h) \leq a_t,$$

when S_t is an OAR, and

$$-e_t(h) \leq b_t,$$

when S_t is a PTV. Therefore, we require that $h = Dx$ lie within the intersection of these convex sets.

Chapter 25

Planewave Propagation

In this chapter we demonstrate how the Fourier transform arises naturally as we study the signals received in the farfield from an array of tranmitters or reflectors. We restrict our attention to single-frequency, or narrowband, signals.

25.1 Transmission and Remote-Sensing

For pedagogical reasons, we shall discuss separately what we shall call the transmission and the remote-sensing problems, although the two problems are opposite sides of the same coin, in a sense. In the one-dimensional transmission problem, it is convenient to imagine the transmitters located at points $(x, 0)$ within a bounded interval $[-A, A]$ of the x-axis, and the measurements taken at points P lying on a circle of radius D, centered at the origin. The radius D is large, with respect to A. It may well be the case that no actual sensing is to be performed, but rather, we are simply interested in what the received signal pattern is at points P distant from the transmitters. Such would be the case, for example, if we were analyzing or constructing a transmission pattern of radio broadcasts. In the remote-sensing problem, in contrast, we imagine, in the one-dimensional case, that our sensors occupy a bounded interval of the x-axis, and the transmitters or reflectors are points of a circle whose radius is large, with respect to the size of the bounded interval. The actual size of the radius does not matter and we are interested in determining the amplitudes of the transmitted or reflected signals, as a function of angle only. Such is the case in astronomy, farfield sonar or radar, and the like. Both the transmission and remote-sensing problems illustrate the important role played by the Fourier transform.

25.2 The Transmission Problem

We identify two distinct transmission problems: the direct problem and the inverse problem. In the direct transmission problem, we wish to determine the farfield pattern, given the complex amplitudes of the transmitted signals. In the inverse transmission problem, the array of transmitters or reflectors is the object of interest; we are given, or we measure, the farfield pattern and wish to determine the amplitudes. For simplicity, we consider only single-frequency signals.

We suppose that each point x in the interval $[-A, A]$ transmits the signal $f(x)e^{i\omega t}$, where $f(x)$ is the complex amplitude of the signal and $\omega > 0$ is the common fixed frequency of the signals. Let $D > 0$ be large, with respect to A, and consider the signal received at each point P given in polar coordinates by $P = (D, \theta)$. The distance from $(x, 0)$ to P is approximately $D - x\cos\theta$, so that, at time t, the point P receives from $(x, 0)$ the signal $f(x)e^{i\omega(t-(D-x\cos\theta)/c)}$, where c is the propagation speed. Therefore, the combined signal received at P is

$$B(P,t) = e^{i\omega t}e^{-i\omega D/c}\int_{-A}^{A} f(x)e^{ix\frac{\omega\cos\theta}{c}}dx. \tag{25.1}$$

The integral term, which gives the farfield pattern of the transmission, is

$$F(\frac{\omega\cos\theta}{c}) = \int_{-A}^{A} f(x)e^{ix\frac{\omega\cos\theta}{c}}dx, \tag{25.2}$$

where $F(\gamma)$ is the Fourier transform of $f(x)$, given by

$$F(\gamma) = \int_{-A}^{A} f(x)e^{ix\gamma}dx. \tag{25.3}$$

How $F(\frac{\omega\cos\theta}{c})$ behaves, as a function of θ, as we change A and ω, is discussed in some detail in the chapter in [62] on direct transmission.

Consider, for example, the function $f(x) = 1$, for $|x| \leq A$, and $f(x) = 0$, otherwise. The Fourier transform of $f(x)$ is

$$F(\gamma) = 2A\text{sinc}(A\gamma), \tag{25.4}$$

where $\text{sinc}(t)$ is defined to be

$$\text{sinc}(t) = \frac{\sin(t)}{t}, \tag{25.5}$$

for $t \neq 0$, and $\text{sinc}(0) = 1$. Then $F(\frac{\omega\cos\theta}{c}) = 2A$ when $\cos\theta = 0$, so when $\theta = \frac{\pi}{2}$ and $\theta = \frac{3\pi}{2}$. We will have $F(\frac{\omega\cos\theta}{c}) = 0$ when $A\frac{\omega\cos\theta}{c} = \pi$, or $\cos\theta = \frac{\pi c}{A\omega}$. Therefore, the transmission pattern has no nulls if $\frac{\pi c}{A\omega} > 1$. In

order for the transmission pattern to have nulls, we need $A > \frac{\lambda}{2}$, where $\lambda = \frac{2\pi c}{\omega}$ is the wavelength. This rather counterintuitive fact, namely that we need more signals transmitted in order to receive less at certain locations, illustrates the phenomenon of destructive interference.

25.3 Reciprocity

For certain remote-sensing applications, such as sonar and radar array processing and astronomy, it is convenient to switch the roles of sender and receiver. Imagine that superimposed planewave fields are sensed at points within some bounded region of the interior of the sphere, having been transmitted or reflected from the points P on the surface of a sphere whose radius D is large with respect to the bounded region. The *reciprocity principle* tells us that the same mathematical relation holds between points P and $(x, 0)$, regardless of which is the sender and which the receiver. Consequently, the data obtained at the points $(x, 0)$ are then values of the inverse Fourier transform of the function describing the amplitude of the signal sent from each point P.

25.4 Remote Sensing

A basic problem in remote sensing is to determine the nature of a distant object by measuring signals transmitted by or reflected from that object. If the object of interest is sufficiently remote, that is, is in the *farfield*, the data we obtain by sampling the propagating spatio-temporal field is related, approximately, to what we want by *Fourier transformation*. The problem is then to estimate a function from finitely many (usually noisy) values of its *Fourier transform*. The application we consider here is a common one of remote-sensing of transmitted or reflected waves propagating from distant sources. Examples include optical imaging of planets and asteroids using reflected sunlight, radio-astronomy imaging of distant sources of radio waves, active and passive sonar, and radar imaging.

25.5 The Wave Equation

In many areas of remote sensing, what we measure are the fluctuations in time of an electromagnetic or acoustic field. Such fields are described mathematically as solutions of certain partial differential equations, such as the *wave equation*. A function $u(x, y, z, t)$ is said to satisfy the *three-dimensional wave equation* if

$$u_{tt} = c^2(u_{xx} + u_{yy} + u_{zz}) = c^2\nabla^2 u, \tag{25.6}$$

where u_{tt} denotes the second partial derivative of u with respect to the time variable t twice and $c > 0$ is the (constant) speed of propagation. More complicated versions of the wave equation permit the speed of propagation c to vary with the spatial variables x, y, z, but we shall not consider that here.

We use the method of *separation of variables* at this point, to get some idea about the nature of solutions of the wave equation. Assume, for the moment, that the solution $u(t, x, y, z)$ has the simple form

$$u(t, x, y, z) = f(t)g(x, y, z). \tag{25.7}$$

Inserting this separated form into the wave equation, we get

$$f''(t)g(x, y, z) = c^2 f(t)\nabla^2 g(x, y, z) \tag{25.8}$$

or

$$f''(t)/f(t) = c^2\nabla^2 g(x, y, z)/g(x, y, z). \tag{25.9}$$

The function on the left is independent of the spatial variables, while the one on the right is independent of the time variable; consequently, they must both equal the same constant, which we denote $-\omega^2$. From this we have two separate equations,

$$f''(t) + \omega^2 f(t) = 0, \tag{25.10}$$

and

$$\nabla^2 g(x, y, z) + \frac{\omega^2}{c^2} g(x, y, z) = 0. \tag{25.11}$$

Equation (25.11) is the *Helmholtz equation*.

Equation (25.10) has for its solutions the functions $f(t) = \cos(\omega t)$ and $\sin(\omega t)$, or, in complex form, the complex exponential functions $f(t) = e^{i\omega t}$ and $f(t) = e^{-i\omega t}$. Functions $u(t, x, y, z) = f(t)g(x, y, z)$ with such time dependence are called *time-harmonic* solutions.

25.6 Planewave Solutions

Suppose that, beginning at time $t = 0$, there is a localized disturbance. As time passes, that disturbance spreads out spherically. When the radius of the sphere is very large, the surface of the sphere appears planar, to an observer on that surface, who is said then to be in the *far field*. This motivates the study of solutions of the wave equation that are constant on planes; the so-called *planewave solutions*.

Let $\mathbf{s} = (x, y, z)$ and $u(\mathbf{s}, t) = u(x, y, z, t) = e^{i\omega t}e^{i\mathbf{k}\cdot\mathbf{s}}$. Then we can show that u satisfies the wave equation $u_{tt} = c^2\nabla^2 u$ for any real vector $\mathbf{k}$, so long as $||\mathbf{k}||^2 = \omega^2/c^2$. This solution is a planewave associated with frequency ω and *wavevector* $\mathbf{k}$; at any fixed time the function $u(\mathbf{s}, t)$ is constant on any plane in three-dimensional space having $\mathbf{k}$ as a normal vector.

In radar and sonar, the field $u(\mathbf{s}, t)$ being sampled is usually viewed as a discrete or continuous superposition of planewave solutions with various amplitudes, frequencies, and wavevectors. We sample the field at various spatial locations $\mathbf{s}$, for various times t. Here we simplify the situation a bit by assuming that all the planewave solutions are associated with the same frequency, ω. If not, we can perform an FFT on the functions of time received at each sensor location $\mathbf{s}$ and keep only the value associated with the desired frequency ω.

25.7 Superposition and the Fourier Transform

In the continuous superposition model, the field is

$$u(\mathbf{s}, t) = e^{i\omega t} \int F(\mathbf{k})e^{i\mathbf{k}\cdot\mathbf{s}}d\mathbf{k}. \tag{25.12}$$

Our measurements at the sensor locations $\mathbf{s}$ give us the values

$$f(\mathbf{s}) = \int F(\mathbf{k})e^{i\mathbf{k}\cdot\mathbf{s}}d\mathbf{k}. \tag{25.13}$$

The data are then Fourier transform values of the complex function $F(\mathbf{k})$; $F(\mathbf{k})$ is defined for all three-dimensional real vectors $\mathbf{k}$, but is zero, in theory, at least, for those $\mathbf{k}$ whose squared length $||\mathbf{k}||^2$ is not equal to ω^2/c^2. Our goal is then to estimate $F(\mathbf{k})$ from measured values of its Fourier transform. Since each $\mathbf{k}$ is a normal vector for its planewave field component, determining the value of $F(\mathbf{k})$ will tell us the strength of the planewave component coming from the direction $\mathbf{k}$.

25.7.1 The Spherical Model

We can imagine that the sources of the planewave fields are the points P that lie on the surface of a large sphere centered at the origin. For each P, the ray from the origin to P is parallel to some wavevector $\mathbf{k}$. The function $F(\mathbf{k})$ can then be viewed as a function $F(P)$ of the points P. Our measurements will be taken at points $\mathbf{s}$ inside this sphere. The radius of the sphere is assumed to be orders of magnitude larger than the distance between sensors. The situation is that of astronomical observation of the heavens using ground-based antennas. The sources of the optical or electromagnetic signals reaching the antennas are viewed as lying on a large sphere

surrounding the earth. Distance to the sources is not considered now, and all we are interested in are the amplitudes $F(\mathbf{k})$ of the fields associated with each direction $\mathbf{k}$.

25.8 Sensor Arrays

In some applications the sensor locations are essentially arbitrary, while in others their locations are carefully chosen. Sometimes, the sensors are collinear, as in sonar towed arrays.

25.8.1 The Two-Dimensional Array

Suppose now that the sensors are in locations $\mathbf{s} = (x, y, 0)$, for various x and y; then we have a *planar array* of sensors. Then the dot product $\mathbf{s} \cdot \mathbf{k}$ that occurs in Equation (25.13) is

$$\mathbf{s} \cdot \mathbf{k} = xk_1 + yk_2; \tag{25.14}$$

we cannot *see* the third component, k_3. However, since we know the size of the vector $\mathbf{k}$, we can determine $|k_3|$. The only ambiguity that remains is that we cannot distinguish sources on the upper hemisphere from those on the lower one. In most cases, such as astronomy, it is obvious in which hemisphere the sources lie, so the ambiguity is resolved.

The function $F(\mathbf{k})$ can then be viewed as $F(k_1, k_2)$, a function of the two variables k_1 and k_2. Our measurements give us values of $f(x, y)$, the two-dimensional Fourier transform of $F(k_1, k_2)$. Because of the limitation $||\mathbf{k}|| = \frac{\omega}{c}$, the function $F(k_1, k_2)$ has bounded support. Consequently, its Fourier transform cannot have bounded support. As a result, we can never have all the values of $f(x, y)$, and so cannot hope to reconstruct $F(k_1, k_2)$ exactly, even for noise-free data.

25.8.2 The One-Dimensional Array

If the sensors are located at points $\mathbf{s}$ having the form $\mathbf{s} = (x, 0, 0)$, then we have a *line array* of sensors. The dot product in Equation (25.13) becomes

$$\mathbf{s} \cdot \mathbf{k} = xk_1. \tag{25.15}$$

Now the ambiguity is greater than in the planar array case. Once we have k_1, we know that

$$k_2^2 + k_3^2 = (\frac{\omega}{c})^2 - k_1^2, \tag{25.16}$$

which describes points P lying on a circle on the surface of the distant sphere, with the vector $(k_1, 0, 0)$ pointing at the center of the circle. It

is said then that we have a *cone of ambiguity*. One way to resolve the situation is to assume $k_3 = 0$; then $|k_2|$ can be determined and we have remaining only the ambiguity involving the sign of k_2. Once again, in many applications, this remaining ambiguity can be resolved by other means.

Once we have resolved any ambiguity, we can view the function $F(\mathbf{k})$ as $F(k_1)$, a function of the single variable k_1. Our measurements give us values of $f(x)$, the Fourier transform of $F(k_1)$. As in the two-dimensional case, the restriction on the size of the vectors $\mathbf{k}$ means that the function $F(k_1)$ has bounded support. Consequently, its Fourier transform, $f(x)$, cannot have bounded support. Therefore, we shall never have all of $f(x)$, and so cannot hope to reconstruct $F(k_1)$ exactly, even for noise-free data.

25.8.3 Limited Aperture

In both the one- and two-dimensional problems, the sensors will be placed within some bounded region, such as $|x| \leq A$, $|y| \leq B$ for the two-dimensional problem, or $|x| \leq A$ for the one-dimensional case. These bounded regions are the *apertures* of the arrays. The larger these apertures are, in units of the wavelength, the better the resolution of the reconstructions.

In digital array processing there are only finitely many sensors, which then places added limitations on our ability to reconstruction the field amplitude function $F(\mathbf{k})$.

25.9 The Remote-Sensing Problem

We shall begin our discussion of the remote-sensing problem by considering an extended object transmitting or reflecting a single-frequency, or *narrowband*, signal. The narrowband, extended-object case is a good place to begin, since a point object is simply a limiting case of an extended object, and broadband received signals can always be filtered to reduce their frequency band.

25.9.1 The Solar-Emission Problem

In [26] Bracewell discusses the *solar-emission* problem. In 1942, it was observed that radio-wave emissions in the one-meter wavelength range were arriving from the sun. Were they coming from the entire disk of the sun or were the sources more localized, in sunspots, for example? The problem then was to view each location on the sun's surface as a potential source of these radio waves and to determine the intensity of emission corresponding to each location.

For electromagnetic waves the propagation speed is the speed of light in a vacuum, which we shall take here to be $c = 3 \times 10^8$ meters per second.

The wavelength λ for gamma rays is around one Angstrom, which is 10^{-10} meters; for x-rays it is about one millimicron, or 10^{-9} meters. The visible spectrum has wavelengths that are a little less than one micron, that is, 10^{-6} meters. Shortwave radio has a wavelength around one millimeter; microwaves have wavelengths between one centimeter and one meter. Broadcast radio has a λ running from about 10 meters to 1000 meters, while the so-called long radio waves can have wavelengths several thousand meters long.

The sun has an angular diameter of 30 min. of arc, or one-half of a degree, when viewed from earth, but the needed resolution was more like 3 min. of arc. As we shall see shortly, such resolution requires a radio telescope 1000 wavelengths across, which means a diameter of 1km at a wavelength of 1 meter; in 1942 the largest military radar antennas were less than 5 meters across. A solution was found, using the method of reconstructing an object from line-integral data, a technique that surfaced again in tomography. The problem here is inherently two-dimensional, but, for simplicity, we shall begin with the one-dimensional case.

25.10 Sampling

In the one-dimensional case, the signal received at the point $(x, 0, 0)$ is essentially the inverse Fourier transform $f(x)$ of the function $F(k_1)$; for notational simplicity, we write $k = k_1$. The $F(k)$ supported on a bounded interval $|k| \leq \frac{\omega}{c}$, so $f(x)$ cannot have bounded support. As we noted earlier, to determine $F(k)$ exactly, we would need measurements of $f(x)$ on an unbounded set. But, which unbounded set?

Because the function $F(k)$ is zero outside the interval $[-\frac{\omega}{c}, \frac{\omega}{c}]$, the function $f(x)$ is *band-limited*. The *Nyquist spacing* in the variable x is therefore

$$\Delta_x = \frac{\pi c}{\omega}. \tag{25.17}$$

The wavelength λ associated with the frequency ω is defined to be

$$\lambda = \frac{2\pi c}{\omega}, \tag{25.18}$$

so that

$$\Delta_x = \frac{\lambda}{2}. \tag{25.19}$$

The significance of the Nyquist spacing comes from *Shannon's Sampling Theorem*, which says that if we have the values $f(m\Delta_x)$, for all integers m, then we have enough information to recover $F(k)$ exactly. In practice, of course, this is never the case.

25.11 The Limited-Aperture Problem

In the remote-sensing problem, our measurements at points $(x, 0, 0)$ in the farfield give us the values $f(x)$. Suppose now that we are able to take measurements only for limited values of x, say for $|x| \leq A$; then $2A$ is the *aperture* of our antenna or array of sensors. We describe this by saying that we have available measurements of $f(x)h(x)$, where $h(x) = \chi_A(x) = 1$, for $|x| \leq A$, and zero otherwise. So, in addition to describing blurring and low-pass filtering, the convolution-filter model can also be used to model the limited-aperture problem. As in the low-pass case, the limited-aperture problem can be attacked using extrapolation, but with the same sort of risks described for the low-pass case. A much different approach is to increase the aperture by physically moving the array of sensors, as in *synthetic aperture radar* (SAR).

Returning to the farfield remote-sensing model, if we have Fourier transform data only for $|x| \leq A$, then we have $f(x)$ for $|x| \leq A$. Using $h(x) = \chi_A(x)$ to describe the limited aperture of the system, the point-spread function is $H(\gamma) = 2A\text{sinc}(\gamma A)$, the Fourier transform of $h(x)$. The first zeros of the numerator occur at $|\gamma| = \frac{\pi}{A}$, so the main lobe of the point-spread function has width $\frac{2\pi}{A}$. For this reason, the resolution of such a limited-aperture imaging system is said to be on the order of $\frac{1}{A}$. Since $|k| \leq \frac{\omega}{c}$, we can write $k = \frac{\omega}{c}\cos\theta$, where θ denotes the angle between the positive x-axis and the vector $\mathbf{k} = (k_1, k_2, 0)$; that is, θ points in the direction of the point P associated with the wavevector $\mathbf{k}$. The resolution, as measured by the width of the main lobe of the point-spread function $H(\gamma)$, in units of k, is $\frac{2\pi}{A}$, but, the angular resolution will depend also on the frequency ω. Since $k = \frac{2\pi}{\lambda}\cos\theta$, a distance of one unit in k may correspond to a large change in θ when ω is small, but only to a relatively small change in θ when ω is large. For this reason, the aperture of the array is usually measured in units of the wavelength; an aperture of $A = 5$ meters may be acceptable if the frequency is high, so that the wavelength is small, but not if the radiation is in the one-meter-wavelength range.

25.12 Resolution

If $F(k) = \delta(k)$ and $h(x) = \chi_A(x)$ describes the aperture-limitation of the imaging system, then the point-spread function is $H(\gamma) = 2A\text{sinc}(\gamma A)$. The maximum of $H(\gamma)$ still occurs at $\gamma = 0$, but the main lobe of $H(\gamma)$ extends from $-\frac{\pi}{A}$ to $\frac{\pi}{A}$; the point source has been spread out. If the point-source object shifts, so that $F(k) = \delta(k - a)$, then the reconstructed image of the object is $H(k - a)$, so the peak is still in the proper place. If we know *a priori* that the object is a single point source, but we do not know its location, the spreading of the point poses no problem; we simply look for

the maximum in the reconstructed image. Problems arise when the object contains several point sources, or when we do not know *a priori* what we are looking at, or when the object contains no point sources, but is just a continuous distribution.

Suppose that $F(k) = \delta(k-a) + \delta(k-b)$; that is, the object consists of two point sources. Then Fourier transformation of the aperture-limited data leads to the reconstructed image

$$R(k) = 2A\Big(\text{sinc}(A(k-a)) + \text{sinc}(A(k-b))\Big). \tag{25.20}$$

If $|b-a|$ is large enough, $R(k)$ will have two distinct maxima, at approximately $k = a$ and $k = b$, respectively. For this to happen, we need π/A, the width of the main lobe of the function $\text{sinc}(Ak)$, to be less than $|b-a|$. In other words, to resolve the two point sources a distance $|b-a|$ apart, we need $A \geq \pi/|b-a|$. However, if $|b-a|$ is too small, the distinct maxima merge into one, at $k = \frac{a+b}{2}$ and resolution will be lost. How small is too small will depend on both A and ω.

Suppose now that $F(k) = \delta(k-a)$, but we do not know *a priori* that the object is a single point source. We calculate

$$R(k) = H(k-a) = 2A\text{sinc}(A(k-a)) \tag{25.21}$$

and use this function as our reconstructed image of the object, for all k. What we see when we look at $R(k)$ for some $k = b \neq a$ is $R(b)$, which is the same thing we see when the point source is at $k = b$ and we look at $k = a$. Point-spreading is, therefore, more than a cosmetic problem. When the object is a point source at $k = a$, but we do not know *a priori* that it is a point source, the spreading of the point causes us to believe that the object function $F(k)$ is nonzero at values of k other than $k = a$. When we look at, say, $k = b$, we see a nonzero value that is caused by the presence of the point source at $k = a$.

Suppose now that the object function $F(k)$ contains no point sources, but is simply an ordinary function of k. If the aperture A is very small, then the function $H(k)$ is nearly constant over the entire extent of the object. The convolution of $F(k)$ and $H(k)$ is essentially the integral of $F(k)$, so the reconstructed object is $R(k) = \int F(k)dk$, for all k.

Let's see what this means for the solar-emission problem discussed earlier.

25.12.1 The Solar-Emission Problem Revisited

The wavelength of the radiation is $\lambda = 1$ meter. Therefore, $\frac{\omega}{c} = 2\pi$, and k in the interval $[-2\pi, 2\pi]$ corresponds to the angle θ in $[0, \pi]$. The sun has an angular diameter of 30 minutes of arc, which is about 10^{-2} radians. Therefore, the sun subtends the angles θ in $[\frac{\pi}{2} - (0.5)\cdot 10^{-2}, \frac{\pi}{2} + (0.5)\cdot 10^{-2}]$,

which corresponds roughly to the variable k in the interval $[-3 \cdot 10^{-2}, 3 \cdot 10^{-2}]$. Resolution of 3 minutes of arc means resolution in the variable k of $3 \cdot 10^{-3}$. If the aperture is $2A$, then to achieve this resolution, we need

$$\frac{\pi}{A} \leq 3 \cdot 10^{-3}, \tag{25.22}$$

or

$$A \geq \frac{\pi}{3} \cdot 10^{3} \tag{25.23}$$

meters, or A not less than about 1000 meters.

The radio-wave signals emitted by the sun are focused, using a parabolic radio-telescope. The telescope is pointed at the center of the sun. Because the sun is a great distance from the earth and the subtended arc is small (30 min.), the signals from each point on the sun's surface arrive at the parabola nearly head-on, that is, parallel to the line from the vertex to the focal point, and are reflected to the receiver located at the focal point of the parabola. The effect of the parabolic antenna is not to discriminate against signals coming from other directions, since there are none, but to effect a summation of the signals received at points $(x, 0, 0)$, for $|x| \leq A$, where $2A$ is the diameter of the parabola. When the aperture is large, the function $h(x)$ is nearly one for all x and the signal received at the focal point is essentially

$$\int f(x)dx = F(0); \tag{25.24}$$

we are now able to distinguish between $F(0)$ and other values $F(k)$. When the aperture is small, $h(x)$ is essentially $\delta(x)$ and the signal received at the focal point is essentially

$$\int f(x)\delta(x)dx = f(0) = \int F(k)dk; \tag{25.25}$$

now all we get is the contribution from all the k, superimposed, and all resolution is lost.

Since the solar emission problem is clearly two-dimensional, and we need 3 min. resolution in both dimensions, it would seem that we would need a circular antenna with a diameter of about one kilometer, or a rectangular antenna roughly one kilometer on a side. We shall return to this problem later, once when we discuss multi-dimensional Fourier transforms, and then again when we consider tomographic reconstruction of images from line integrals.

25.13 Discrete Data

A familiar topic in signal processing is the passage from functions of continuous variables to discrete sequences. This transition is achieved by *sam-*

pling, that is, extracting values of the continuous-variable function at discrete points in its domain. Our example of farfield propagation can be used to explore some of the issues involved in sampling.

Imagine an infinite *uniform line array* of sensors formed by placing receivers at the points $(n\Delta, 0, 0)$, for some $\Delta > 0$ and all integers n. Then our data are the values $f(n\Delta)$. Because we defined $k = \frac{\omega}{c}\cos\theta$, it is clear that the function $F(k)$ is zero for k outside the interval $[-\frac{\omega}{c}, \frac{\omega}{c}]$.

Our discrete array of sensors cannot distinguish between the signal arriving from θ and a signal with the same amplitude, coming from an angle α with

$$\frac{\omega}{c}\cos\alpha = \frac{\omega}{c}\cos\theta + \frac{2\pi}{\Delta}m, \tag{25.26}$$

where m is an integer. To resolve this ambiguity, we select $\Delta > 0$ so that

$$-\frac{\omega}{c} + \frac{2\pi}{\Delta} \geq \frac{\omega}{c}, \tag{25.27}$$

or

$$\Delta \leq \frac{\pi c}{\omega} = \frac{\lambda}{2}. \tag{25.28}$$

The sensor spacing $\Delta_s = \frac{\lambda}{2}$ is the *Nyquist spacing*.

In the sunspot example, the object function $F(k)$ is zero for k outside of an interval much smaller than $[-\frac{\omega}{c}, \frac{\omega}{c}]$. Knowing that $F(k) = 0$ for $|k| > K$, for some $0 < K < \frac{\omega}{c}$, we can accept ambiguities that confuse θ with another angle that lies outside the angular diameter of the object. Consequently, we can redefine the Nyquist spacing to be

$$\Delta_s = \frac{\pi}{K}. \tag{25.29}$$

This tells us that when we are imaging a distant object with a small angular diameter, the Nyquist spacing is greater than $\frac{\lambda}{2}$. If our sensor spacing has been chosen to be $\frac{\lambda}{2}$, then we have *oversampled*. In the oversampled case, band-limited extrapolation methods can be used to improve resolution .

25.13.1 Reconstruction from Samples

From the data gathered at our infinite array we have extracted the Fourier transform values $f(n\Delta)$, for all integers n. The obvious question is whether or not the data is sufficient to reconstruct $F(k)$. We know that, to avoid ambiguity, we must have $\Delta \leq \frac{\pi c}{\omega}$. The good news is that, provided this condition holds, $F(k)$ is uniquely determined by this data and formulas exist for reconstructing $F(k)$ from the data; this is the content of the *Shannon's Sampling Theorem*. Of course, this is only of theoretical interest, since we never have infinite data. Nevertheless, a considerable amount of traditional signal-processing exposition makes use of this infinite-sequence model. The real problem, of course, is that our data is always finite.

25.14 The Finite-Data Problem

Suppose that we build a *uniform line array* of sensors by placing receivers at the points $(n\Delta, 0, 0)$, for some $\Delta > 0$ and $n = -N, ..., N$. Then our data are the values $f(n\Delta)$, for $n = -N, ..., N$. Suppose, as previously, that the object of interest, the function $F(k)$, is nonzero only for values of k in the interval $[-K, K]$, for some $0 < K < \frac{\omega}{c}$. Once again, we must have $\Delta \leq \frac{\pi c}{\omega}$ to avoid ambiguity; but this is not enough, now. The finite Fourier data is no longer sufficient to determine a unique $F(k)$. The best we can hope to do is to estimate the true $F(k)$, using both our measured Fourier data and whatever prior knowledge we may have about the function $F(k)$, such as where it is nonzero, if it consists of Dirac delta point sources, or if it is nonnegative. The data is also noisy, and that must be accounted for in the reconstruction process.

In certain applications, such as sonar array processing, the sensors are not necessarily arrayed at equal intervals along a line, or even at the grid points of a rectangle, but in an essentially arbitrary pattern in two, or even three, dimensions. In such cases, we have values of the Fourier transform of the object function, but at essentially arbitrary values of the variable. How best to reconstruct the object function in such cases is not obvious.

25.15 Functions of Several Variables

Fourier transformation applies, as well, to functions of several variables. As in the one-dimensional case, we can motivate the multi-dimensional Fourier transform using the farfield propagation model. As we noted earlier, the solar emission problem is inherently a two-dimensional problem.

25.15.1 Two-Dimensional Farfield Object

Assume that our sensors are located at points $\mathbf{s} = (x, y, 0)$ in the x,y-plane. As discussed previously, we assume that the function $F(\mathbf{k})$ can be viewed as a function $F(k_1, k_2)$. Since, in most applications, the distant object has a small angular diameter when viewed from a great distance - the sun's is only 30 minutes of arc - the function $F(k_1, k_2)$ will be supported on a small subset of vectors (k_1, k_2).

25.15.2 Limited Apertures in Two Dimensions

Suppose we have the values of the Fourier transform, $f(x, y)$, for $|x| \leq A$ and $|y| \leq A$. We describe this limited-data problem using the function $h(x, y)$ that is one for $|x| \leq A$, and $|y| \leq A$, and zero, otherwise. Then the

point-spread function is the Fourier transform of this $h(x, y)$, given by

$$H(\alpha, \beta) = 4AB\text{sinc}(A\alpha)\text{sinc}(B\beta). \tag{25.30}$$

The resolution in the horizontal (x) direction is on the order of $\frac{1}{A}$, and $\frac{1}{B}$ in the vertical, where, as in the one-dimensional case, aperture is best measured in units of wavelength.

Suppose our aperture is circular, with radius A. Then we have Fourier transform values $f(x, y)$ for $\sqrt{x^2 + y^2} \leq A$. Let $h(x, y)$ equal one, for $\sqrt{x^2 + y^2} \leq A$, and zero, otherwise. Then the point-spread function of this limited-aperture system is the Fourier transform of $h(x, y)$, given by $H(\alpha, \beta) = \frac{2\pi A}{r} J_1(rA)$, with $r = \sqrt{\alpha^2 + \beta^2}$. The resolution of this system is roughly the distance from the origin to the first null of the function $J_1(rA)$, which means that $rA = 4$, roughly.

For the solar emission problem, this says that we would need a circular aperture with radius approximately one kilometer to achieve 3 minutes of arc resolution. But this holds only if the antenna is stationary; a moving antenna is different! The solar emission problem was solved by using a rectangular antenna with a large A, but a small B, and exploiting the rotation of the earth. The resolution is then good in the horizontal, but bad in the vertical, so that the imaging system discriminates well between two distinct vertical lines, but cannot resolve sources within the same vertical line. Because B is small, what we end up with is essentially the integral of the function $f(x, z)$ along each vertical line. By tilting the antenna, and waiting for the earth to rotate enough, we can get these integrals along any set of parallel lines. The problem then is to reconstruct $F(k_1, k_2)$ from such line integrals. This is also the main problem in tomography.

25.16 Broadband Signals

We have spent considerable time discussing the case of a distant point source or an extended object transmitting or reflecting a single-frequency signal. If the signal consists of many frequencies, the so-called broadband case, we can still analyze the received signals at the sensors in terms of time delays, but we cannot easily convert the delays to phase differences, and thereby make good use of the Fourier transform. One approach is to filter each received signal, to remove components at all but a single frequency, and then to proceed as previously discussed. In this way we can process one frequency at a time. The object now is described in terms of a function of both $\mathbf{k}$ and ω, with $F(\mathbf{k}, \omega)$ the complex amplitude associated with the wave vector $\mathbf{k}$ and the frequency ω. In the case of radar, the function $F(\mathbf{k}, \omega)$ tells us how the material at P reflects the radio waves at the various frequencies ω, and thereby gives information about the nature of the material making up the object near the point P.

There are times, of course, when we do not want to decompose a broadband signal into single-frequency components. A satellite reflecting a TV signal is a broadband point source. All we are interested in is receiving the broadband signal clearly, free of any other interfering sources. The direction of the satellite is known and the antenna is turned to face the satellite. Each location on the parabolic dish reflects the same signal. Because of its parabolic shape, the signals reflected off the dish and picked up at the focal point have exactly the same travel time from the satellite, so they combine coherently, to give us the desired TV signal.

Part V

Appendices

Chapter 26

Complex Exponentials

The most important signals considered in signal processing are *sinusoids*, that is, sine or cosine functions. A *complex sinusoid* is a function of the real variable t having the form

$$f(t) = \cos \omega t + i \sin \omega t, \tag{26.1}$$

for some real frequency ω. Complex sinusoids are also called *complex exponential functions*.

26.1 Why "Exponential"?

Complex exponential functions have the property $f(t + u) = f(t)f(u)$, which is characteristic of exponential functions. This property can be easily verified for $f(t)$ using trigonometric identities.

Exponential functions in calculus take the form $g(t) = a^t$, for some positive constant a; the most famous of these is $g(t) = e^t$. The function $f(t)$ in Equation (26.1) has complex values, so cannot be $f(t) = a^t$ for any positive a. But, what if we let a be complex? If it is the case that $f(t) = a^t$ for some complex a, then, setting $t = 1$, we would have $a = f(1) = \cos \omega + i \sin \omega$. This is the complex number denoted e^i; to see why we consider Taylor series expansions.

26.2 Taylor-series expansions

The Taylor series expansion for the exponential function $g(t) = e^t$ is

$$e^t = 1 + t + \frac{1}{2!}t^2 + \frac{1}{3!}t^3 + ... \tag{26.2}$$

If we replace t with $i\omega$, where $i = \sqrt{-1}$, we obtain

$$e^{i\omega} = (1 - \frac{1}{2!}\omega^2 + \frac{1}{4!}\omega^4 - ...) + i(\omega - \frac{1}{3!}\omega^3 + \frac{1}{5!}\omega^5 - ...). \qquad (26.3)$$

We recognize the two series in Equation (26.3) as the Taylor-series expansions for $\cos\omega$ and $\sin\omega$, respectively, so we can write

$$e^{i\omega} = \cos\omega + i\sin\omega.$$

Therefore the complex exponential function in Equation (26.1) can be written

$$f(t) = (e^{i\omega})^t = e^{i\omega t}.$$

If $A = |A|e^{i\theta}$, then the signal $h(t) = Ae^{i\omega t}$ can be written

$$h(t) = |A|e^{i(\omega t+\theta)};$$

here A is called the *complex amplitude* of the signal $h(t)$, with positive amplitude $|A|$ and phase θ.

26.3 Basic Properties

The laws of exponents apply to the complex exponential functions, so, for example, we can write

$$e^{i\omega t}e^{i\omega u} = e^{i\omega(t+u)}.$$

Note also that the complex conjugate of $e^{i\omega t}$ is

$$\overline{e^{i\omega t}} = e^{-i\omega t}$$

It follows directly from the definition of $e^{i\omega t}$ that

$$\sin(\omega t) = \frac{1}{2i}[e^{i\omega t} - e^{-i\omega t}],$$

and

$$\cos(\omega t) = \frac{1}{2}[e^{i\omega t} + e^{-i\omega t}].$$

Exercise 26.1 *Show that*

$$e^{ia} + e^{ib} = e^{i\frac{a+b}{2}}[e^{i\frac{a-b}{2}} + e^{-i\frac{a-b}{2}}] = 2e^{i\frac{a+b}{2}}\cos(\frac{a-b}{2}),$$

and

$$e^{ia} - e^{ib} = e^{i\frac{a+b}{2}}[e^{i\frac{a-b}{2}} - e^{-i\frac{a-b}{2}}] = 2ie^{i\frac{a+b}{2}}\sin(\frac{a-b}{2}).$$

Exercise 26.2 *Use the formula for the sum of a geometric progression,*

$$1 + r + r^2 + ... + r^k = (1 - r^{k+1})/(1 - r),$$

to show that

$$\sum_{n=M}^{N} e^{i\omega n} = e^{i\frac{M+N}{2}} \frac{\sin(\omega\frac{N-M+1}{2})}{\sin(\frac{\omega}{2})}. \tag{26.4}$$

Exercise 26.3 *Express the result in the previous exercise in terms of real and imaginary parts to show that*

$$\sum_{n=M}^{N} \cos(\omega n) = \cos(\frac{M+N}{2}) \frac{\sin(\omega\frac{N-M+1}{2})}{\sin(\frac{\omega}{2})},$$

and

$$\sum_{n=M}^{N} \sin(\omega n) = \sin(\frac{M+N}{2}) \frac{\sin(\omega\frac{N-M+1}{2})}{\sin(\frac{\omega}{2})}.$$

Chapter 27

The Fourier Transform

As we noted previously, the Fourier transform in one and two dimensions plays an important role in transmission tomographic image reconstruction, both in the theoretical formulation and in the practical implementation. In fact, in many areas of remote sensing, including MRI, what we want is related by the Fourier transform to what we can measure.

In this chapter we review the basic properties of the Fourier transform.

27.1 Fourier-Transform Pairs

Let $f(x)$ be defined for the real variable x in $(-\infty, \infty)$. The *Fourier transform* of $f(x)$ is the function of the real variable γ given by

$$F(\gamma) = \int_{-\infty}^{\infty} f(x) e^{i\gamma x} dx. \tag{27.1}$$

Precisely how we interpret the infinite integrals that arise in the discussion of the Fourier transform will depend on the properties of the function $f(x)$. A detailed treatment of this issue, which is beyond the scope of this book, can be found in almost any text on the Fourier transform (see, for example, [129]).

27.1.1 The Issue of Units

When we write $\cos \pi = -1$, it is with the understanding that π is a measure of angle, in radians; the function cos will always have an independent variable in units of radians. By extension, the same is true of the complex exponential functions. Therefore, when we write $e^{ix\gamma}$, we understand the product $x\gamma$ to be in units of radians. If x is measured in seconds, then γ is in units of radians per second; if x is in meters, then γ is in units of

radians per meter. When x is in seconds, we sometimes use the variable $\frac{\gamma}{2\pi}$; since 2π is then in units of radians per cycle, the variable $\frac{\gamma}{2\pi}$ is in units of cycles per second, or Hertz. When we sample $f(x)$ at values of x spaced Δ apart, the Δ is in units of x-units per sample, and the reciprocal, $\frac{1}{\Delta}$, which is called the *sampling frequency*, is in units of samples per x-units. If x is in seconds, then Δ is in units of seconds per sample, and $\frac{1}{\Delta}$ is in units of samples per second.

27.1.2 Reconstructing from Fourier-Transform Data

Our goal is often to reconstruct the function $f(x)$ from measurements of its Fourier transform $F(\gamma)$. But, how?

If we have $F(\gamma)$ for all real γ, then we can recover the function $f(x)$ using the *Fourier Inversion Formula*:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\gamma) e^{-i\gamma x} d\gamma. \tag{27.2}$$

The functions $f(x)$ and $F(\gamma)$ are called a *Fourier-transform pair*. Once again, the proper interpretation of Equation (27.2) will depend on the properties of the functions involved. If both $f(x)$ and $F(\gamma)$ are measurable and absolutely integrable then both functions are continuous. In the next chapter, we prove the Fourier Inversion Formula for the functions in the Schwartz class [129].

27.1.3 An Example

Consider the function $f(x) = \frac{1}{2A}$, for $|x| \leq A$, and $f(x) = 0$, otherwise. The Fourier transform of this $f(x)$ is

$$F(\gamma) = \frac{\sin(A\gamma)}{A\gamma},$$

for all real $\gamma \neq 0$, and $F(0) = 1$. Note that $F(\gamma)$ is nonzero throughout the real line, except for isolated zeros, but that it goes to zero as we go to the infinities. This is typical behavior. Notice also that the smaller the A, the slower $F(\gamma)$ dies out; the first zeros of $F(\gamma)$ are at $|\gamma| = \frac{\pi}{A}$, so the main lobe widens as A goes to zero. The function $f(x)$ is not continuous, so its Fourier transform cannot be absolutely integrable. In this case, the Fourier Inversion Formula must be interpreted as involving convergence in the L^2 norm.

It may seem paradoxical that when A is larger, its Fourier transform dies off more quickly. The Fourier transform $F(\gamma)$ goes to zero faster for larger A because of destructive interference. Because of differences in their complex phases as x varies, the magnitude of the sum of the complex exponential

functions $e^{i\gamma x}$ is much smaller than we might expect, especially when A is large. For smaller A the x are more similar to one another and so the complex exponential functions are much more *in phase* with one another; consequently, the magnitude of the sum remains large. A more quantitative statement of this phenomenon is provided by the *uncertainty principle* (see [61]).

27.1.4 The Dirac Delta

Consider what happens in the limit, as $A \to 0$. Then we have an infinitely high point source at $x = 0$; we denote this by $\delta(x)$, the *Dirac delta*. The Fourier transform approaches the constant function with value 1, for all γ; the Fourier transform of $f(x) = \delta(x)$ is the constant function $F(\gamma) = 1$, for all γ. The Dirac delta $\delta(x)$ has the *sifting property*:

$$\int h(x)\delta(x)dx = h(0),$$

for each function $h(x)$ that is continuous at $x = 0$.

Because the Fourier transform of $\delta(x)$ is the function $F(\gamma) = 1$, the Fourier inversion formula tells us that

$$\delta(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\gamma x}d\gamma. \tag{27.3}$$

Obviously, this integral cannot be understood in the usual way. The integral in Equation (27.3) is a symbolic way of saying that

$$\int h(x)(\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\gamma x}d\gamma)dx = \int h(x)\delta(x)dx = h(0), \tag{27.4}$$

for all $h(x)$ that are continuous at $x = 0$; that is, the integral in Equation (27.3) has the sifting property, so it acts like $\delta(x)$. Interchanging the order of integration in Equation (27.4), we obtain

$$\int h(x)(\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\gamma x}d\gamma)dx = \frac{1}{2\pi}\int_{-\infty}^{\infty}(\int h(x)e^{-i\gamma x}dx)d\gamma$$

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} H(-\gamma)d\gamma = \frac{1}{2\pi}\int_{-\infty}^{\infty} H(\gamma)d\gamma = h(0).$$

We shall return to the Dirac delta when we consider farfield point sources.

27.2 Practical Limitations

In actual remote-sensing problems, arrays of sensors cannot be of infinite extent. In digital signal processing, moreover, there are only finitely many

sensors. We never measure the entire Fourier transform $F(\gamma)$, but, at best, just part of it; as we shall see in the chapter on planewave propagation, in the direct transmission problem we measure $F(\gamma)$ only for $\gamma = k$, with $|k| \leq \frac{\omega}{c}$, with ω the frequency and c the propagation speed. In fact, the data we are able to measure is almost never exact values of $F(\gamma)$, but rather, values of some distorted or blurred version. To describe such situations, we usually resort to *convolution-filter* models.

27.3 Convolution Filtering

Imagine that what we measure are not values of $F(\gamma)$, but of $F(\gamma)H(\gamma)$, where $H(\gamma)$ is a function that describes the limitations and distorting effects of the measuring process, including any blurring due to the medium through which the signals have passed, such as refraction of light as it passes through the atmosphere. If we apply the Fourier Inversion Formula to $F(\gamma)H(\gamma)$, instead of to $F(\gamma)$, we get

$$g(x) = \frac{1}{2\pi} \int F(\gamma)H(\gamma)e^{-i\gamma x} dx. \tag{27.5}$$

The function $g(x)$ that results is $g(x) = (f * h)(x)$, the *convolution* of the functions $f(x)$ and $h(x)$, with the latter given by

$$h(x) = \frac{1}{2\pi} \int H(\gamma)e^{-i\gamma x} dx.$$

Note that, if $f(x) = \delta(x)$, then $g(x) = h(x)$; that is, our reconstruction of the object from distorted data is the function $h(x)$ itself. For that reason, the function $h(x)$ is called the *point-spread function* of the imaging system.

Convolution filtering refers to the process of converting any given function, say $f(x)$, into a different function, say $g(x)$, by convolving $f(x)$ with a fixed function $h(x)$. Since this process can be achieved by multiplying $F(\gamma)$ by $H(\gamma)$ and then inverse Fourier transforming, such convolution filters are studied in terms of the properties of the function $H(\gamma)$, known in this context as the *system transfer function*, or the *optical transfer function* (OTF); when γ is a frequency, rather than a spatial frequency, $H(\gamma)$ is called the *frequency-response function* of the filter. The function $|H(\gamma)|$, the magnitude of $H(\gamma)$, is called the *modulation transfer function* (MTF). The study of convolution filters is a major part of signal processing. Such filters provide both reasonable models for the degradation signals undergo, and useful tools for reconstruction.

Let us rewrite Equation (27.5), replacing $F(\gamma)$ and $H(\gamma)$ with their definitions, as given by Equation (27.1). Then we have

$$g(x) = \frac{1}{2\pi} \int (\int f(t)e^{i\gamma t} dt)(\int h(s)e^{i\gamma s} ds)e^{-i\gamma x} d\gamma.$$

Interchanging the order of integration, we get

$$g(x) = \frac{1}{2\pi} \int \int f(t)h(s)(\int e^{i\gamma(x-(t+s))} d\gamma)dsdt.$$

Now using Equation (27.3) to replace the inner integral with $2\pi\delta(x-(t+s))$, the next integral becomes

$$2\pi \int h(s)\delta(x-(t+s))ds = 2\pi h(x-t).$$

Finally, we have

$$g(x) = \int f(t)h(x-t)dt; \tag{27.6}$$

this is the definition of the convolution of the functions f and h.

27.4 Low-Pass Filtering

A major problem in image reconstruction is the removal of blurring, which is often modeled using the notion of convolution filtering. In the one-dimensional case, we describe blurring by saying that we have available measurements not of $F(\gamma)$, but of $F(\gamma)H(\gamma)$, where $H(\gamma)$ is the frequency-response function describing the blurring. If we know the nature of the blurring, then we know $H(\gamma)$, at least to some degree of precision. We can try to remove the blurring by taking measurements of $F(\gamma)H(\gamma)$, dividing these numbers by the value of $H(\gamma)$, and then inverse Fourier transforming. The problem is that our measurements are always noisy, and typical functions $H(\gamma)$ have many zeros and small values, making division by $H(\gamma)$ dangerous, except where the values of $H(\gamma)$ are not too small. These values of γ tend to be the smaller ones, centered around zero, so that we end up with estimates of $F(\gamma)$ itself only for the smaller values of γ. The result is a *low-pass filtering* of the object $f(x)$.

To investigate such low-pass filtering, we suppose that $H(\gamma) = 1$, for $|\gamma| \leq \Gamma$, and is zero, otherwise. Then the filter is called the ideal Γ-lowpass filter. In the farfield propagation model, the variable x is spatial, and the variable γ is spatial frequency, related to how the function $f(x)$ changes spatially, as we move x. Rapid changes in $f(x)$ are associated with values of $F(\gamma)$ for large γ. For the case in which the variable x is time, the variable γ becomes frequency, and the effect of the low-pass filter on $f(x)$ is to remove its higher-frequency components.

One effect of low-pass filtering in image processing is to smooth out the more rapidly changing features of an image. This can be useful if these features are simply unwanted oscillations, but if they are important detail, the smoothing presents a problem. Restoring such wanted detail is

often viewed as removing the unwanted effects of the low-pass filtering; in other words, we try to recapture the missing high-spatial-frequency values that have been zeroed out. Such an approach to image restoration is called *frequency-domain extrapolation* . How can we hope to recover these missing spatial frequencies, when they could have been anything? To have some chance of estimating these missing values we need to have some prior information about the image being reconstructed.

27.5 Two-Dimensional Fourier Transforms

More generally, we consider a function $f(x, y)$ of two real variables. Its Fourier transformation is

$$F(\alpha, \beta) = \int \int f(x, y) e^{i(x\alpha + y\beta)} dx dy. \tag{27.7}$$

For example, suppose that $f(x, y) = 1$ for $\sqrt{x^2 + y^2} \leq R$, and zero, otherwise. Then we have

$$F(\alpha, \beta) = \int_{-\pi}^{\pi} \int_{0}^{R} e^{-i(\alpha r \cos\theta + \beta r \sin\theta)} r dr d\theta.$$

In polar coordinates, with $\alpha = \rho \cos\phi$ and $\beta = \rho \sin\phi$, we have

$$F(\rho, \phi) = \int_{0}^{R} \int_{-\pi}^{\pi} e^{ir\rho\cos(\theta - \phi)} d\theta r dr.$$

The inner integral is well known;

$$\int_{-\pi}^{\pi} e^{ir\rho\cos(\theta - \phi)} d\theta = 2\pi J_0(r\rho),$$

where J_0 denotes the 0th order Bessel function. Using the identity

$$\int_{0}^{z} t^n J_{n-1}(t) dt = z^n J_n(z),$$

we have

$$F(\rho, \phi) = \frac{2\pi R}{\rho} J_1(\rho R).$$

Notice that, since $f(x, y)$ is a radial function, that is, dependent only on the distance from $(0, 0)$ to (x, y), its Fourier transform is also radial.

The first positive zero of $J_1(t)$ is around $t = 4$, so when we measure F at various locations and find $F(\rho, \phi) = 0$ for a particular (ρ, ϕ), we can estimate $R \approx 4/\rho$. So, even when a distant spherical object, like a star, is too far away to be imaged well, we can sometimes estimate its size by finding where the intensity of the received signal is zero [168].

27.5.1 Two-Dimensional Fourier Inversion

Just as in the one-dimensional case, the Fourier transformation that produced $F(\alpha, \beta)$ can be inverted to recover the original $f(x, y)$. The Fourier Inversion Formula in this case is

$$f(x,y) = \frac{1}{4\pi^2} \int \int F(\alpha, \beta) e^{-i(\alpha x + \beta y)} d\alpha d\beta. \tag{27.8}$$

It is important to note that this procedure can be viewed as two one-dimensional Fourier inversions: first, we invert $F(\alpha, \beta)$, as a function of, say, β only, to get the function of α and y

$$g(\alpha, y) = \frac{1}{2\pi} \int F(\alpha, \beta) e^{-i\beta y} d\beta;$$

second, we invert $g(\alpha, y)$, as a function of α, to get

$$f(x,y) = \frac{1}{2\pi} \int g(\alpha, y) e^{-i\alpha x} d\alpha.$$

If we write the functions $f(x, y)$ and $F(\alpha, \beta)$ in polar coordinates, we obtain alternative ways to implement the two-dimensional Fourier inversion. We shall consider these other ways when we discuss the tomography problem of reconstructing a function $f(x, y)$ from line-integral data.

27.6 Fourier Series

Students typically encounter Fourier series before they see Fourier transforms. Suppose that $F(\gamma)$ is zero outside of the interval $[-\Gamma, \Gamma]$. For integers n and $\Delta = \frac{\pi}{\Gamma}$, the complex exponential functions $e^{i\gamma n\Delta}$ are 2Γ-periodic, and mutually orthogonal; that is, for $m \neq n$, we have

$$\int_{-\Gamma}^{\Gamma} e^{i\gamma n\Delta} e^{-i\gamma m\Delta} d\gamma = 0.$$

The objective in Fourier series is to express the function $F(\gamma)$, for γ in $[-\Gamma, \Gamma]$, as a sum of these complex exponential functions,

$$F(\gamma) = \sum_{n=-\infty}^{\infty} a_n e^{i\gamma n\Delta}, \tag{27.9}$$

for some choice of the coefficients a_n.

Multiplying both sides of Equation (27.9) by $e^{-i\gamma m\Delta}$ and integrating from $-\Gamma$ to Γ, we find that

$$\int_{-\Gamma}^{\Gamma} F(\gamma) e^{-i\gamma m\Delta} d\gamma = 2\Gamma a_m.$$

Notice that

$$\int_{-\Gamma}^{\Gamma} F(\gamma)e^{-i\gamma m\Delta}d\gamma = 2\pi f(m\Delta)$$

also. Consequently, we have

$$a_m = \Delta f(m\Delta).$$

This gives us the important result that whenever $F(\gamma)$ is zero outside an interval $[-\Gamma, \Gamma]$, we can recover $F(\gamma)$, and thereby $f(x)$ also, from the infinite discrete set of samples $f(m\Delta)$, for $\Delta = \frac{\pi}{\Gamma}$. In signal processing this result is called *Shannon's Sampling Theorem.*

If $G(\gamma)$ is also zero for $|\gamma| > \Gamma$, then it follows from the orthogonality of the complex exponential functions $e^{i\gamma n\Delta}$ that

$$\frac{1}{2\pi}\int_{-\Gamma}^{\Gamma} F(\gamma)\overline{G(\gamma)}d\gamma = \Delta \sum_{n=-\infty}^{\infty} f(n\Delta)\overline{g(n\Delta)};$$

this is Parseval's Equation.

Note that if $F(\gamma) = 0$ for $|\gamma| > \Gamma$, then the same is true if we replace Γ with any larger value. It follows that in Shannon's Sampling Theorem we need only that $\Delta \leq \frac{\pi}{\Gamma}$.

27.7 The Discrete Fourier Transform

Suppose again that $F(\gamma)$ is zero for $|\gamma| > \Gamma$ and let $\Delta = \frac{\pi}{\Gamma}$. In real applications we never have the entire infinite set of samples $\{f(n\Delta)\}$; at best, we would have a finite subset of these, say for $n = 1$ to $n = N$. If our goal is to estimate $F(\gamma)$, we might choose the *discrete Fourier transform* (DFT) estimate

$$F_{DFT}(\gamma) = \Delta \sum_{n=1}^{N} f(n\Delta)e^{in\Delta\gamma}.$$

The DFT estimate $F_{DFT}(\gamma)$ is data consistent; its inverse Fourier-transform value at $x = n\Delta$ is $f(n\Delta)$ for $n = 1, ..., N$. The DFT is sometimes used in a slightly more general context in which the coefficients are not necessarily viewed as samples of a function $f(x)$.

Once we have decided to use the DFT estimate for the function $F(\gamma)$, we would want to evaluate this estimate at some number of values of γ, so that, for example, we could plot this function. When N is not large (say, several hundred), this poses no problem. But in many applications, especially image processing, N is in the thousands or more, and evaluating the DFT estimate at that many points without a fast algorithm is too costly and time-consuming. The *fast Fourier transform* is an algorithm for performing this calculation quickly.

27.8 The Fast Fourier Transform

A fundamental problem in signal processing is to estimate finitely many values of the function $F(\gamma)$ from finitely many values of its (inverse) Fourier transform, $f(x)$. As we shall see, the DFT arises in several ways in that estimation effort. The *fast Fourier transform* (FFT), discovered in 1965 by Cooley and Tukey, is an important and efficient algorithm for calculating the vector DFT [93]. 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.

27.8.1 Evaluating a Polynomial

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.

27.8.2 The DFT and the Vector DFT

Given the complex N-dimensional column vector $\mathbf{f} = (f_0, f_1, ..., f_{N-1})^T$, define the DFT of vector $\mathbf{f}$ to be the function $DFT_{\mathbf{f}}(\gamma)$, defined for γ in $[0, 2\pi)$, given by

$$DFT_{\mathbf{f}}(\gamma) = \sum_{n=0}^{N-1} f_n e^{in\gamma}.$$

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 equi-spaced points $2\pi/N$ apart in $[0, 2\pi)$.

From the formula for $DFT_{\mathbf{f}}$ we have, for $k = 0, 1, ..., N-1$,

$$F_k = F(2\pi k/N) = \sum_{n=0}^{N-1} f_n e^{2\pi i n k/N}. \tag{27.10}$$

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.

27.8.3 Exploiting Redundancy

Suppose that $N = 2M$ is even. We can rewrite Equation (27.10) 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,

$$F_k = \sum_{m=0}^{M-1} f_{2m} e^{2\pi i m k/M} + e^{2\pi i k/N} \sum_{m=0}^{M-1} f_{2m+1} e^{2\pi i m k/M}. \tag{27.11}$$

Note that if $0 \leq k \leq M-1$ then

$$F_{k+M} = \sum_{m=0}^{M-1} f_{2m} e^{2\pi i m k/M} - e^{2\pi i k/N} \sum_{m=0}^{M-1} f_{2m+1} e^{2\pi i m k/M}, \tag{27.12}$$

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 (27.11) and (27.12) 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 (27.11) 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 $N \log_2 N$.

The FFT can be used to calculate the periodic convolution (or even the nonperiodic convolution) of finite length vectors.

27.8.4 Estimating the Fourier Transform

Finally, let's return to the original context of estimating the Fourier transform $F(\gamma)$ of function $f(x)$ from finitely many samples of $f(x)$. If we have N equi-spaced 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 γ, 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 $N + J$ equi-spaced values of γ in $[0, 2\pi)$.

27.8.5 The Two-Dimensional Case

Suppose now that we have the data $\{f(m\Delta_x, n\Delta_y)\}$, for $m = 1, ..., M$ and $n = 1, ..., N$, where $\Delta_x > 0$ and $\Delta_y > 0$ are the sample spacings in the x and y directions, respectively. The DFT of this data is the function $F_{DFT}(\alpha, \beta)$ defined by

$$F_{DFT}(\alpha, \beta) = \Delta_x \Delta_y \sum_{m=1}^{M} \sum_{n=1}^{N} f(m\Delta_x, n\Delta_y) e^{i(\alpha m \Delta_x + \beta n \Delta_y)},$$

for $|\alpha| \leq \pi/\Delta_x$ and $|\beta| \leq \pi/\Delta_y$. The two-dimensional FFT produces MN values of $F_{DFT}(\alpha, \beta)$ on a rectangular grid of M equi-spaced values of α and N equi-spaced values of β. This calculation proceeds as follows. First, for each fixed value of n, a FFT of the M data points $\{f(m\Delta_x, n\Delta_y)\}, m = 1, ..., M$ is calculated, producing a function, say $G(\alpha_m, n\Delta_y)$, of M equi-spaced values of α and the N equi-spaced values $n\Delta_y$. Then, for each

of the M equi-spaced values of α, the FFT is applied to the N values $G(\alpha_m, n\Delta_y), n = 1, ..., N$, to produce the final result.

Chapter 28

Prony's Method

The date of publication of [209] is often taken by editors to be a typographical error and is replaced by 1995; or, since it is not written in English, perhaps 1895. But the 1795 date is the correct one. The mathematical problem Prony solved arises also in signal processing, and his method for solving it is still used today. Prony's method is also the inspiration for the eigenvector methods described in our next chapter.

28.1 Prony's Problem

Prony considers a function of the form

$$s(t) = \sum_{n=1}^{N} a_n e^{\gamma_n t}, \tag{28.1}$$

where we allow the a_n and the γ_n to be complex. If we take the $\gamma_n = i\omega_n$ to be imaginary, $s(t)$ becomes the sum of complex exponentials; if we take γ_n to be real, then $s(t)$ is the sum of real exponentials, either increasing with t or decreasing with t. The problem is to determine from samples of $s(t)$ the number N, the γ_n, and the a_n.

28.2 Prony's Method

Suppose that we have data $y_m = s(m\Delta)$, for some $\Delta > 0$ and for $m = 1, ..., M$, where we assume that $M = 2N$. We seek a vector $\mathbf{c}$ with entries c_j, $j = 0, ..., N$ such that

$$c_0 y_{k+1} + c_1 y_{k+2} + c_2 y_{k+3} + ... + c_N y_{k+N+1} = 0, \tag{28.2}$$

for $k = 0, 1, ..., M - N - 1$. So, we want a complex vector $\mathbf{c}$ in C^{N+1} orthogonal to $M - N = N$ other vectors. In matrix-vector notation we are solving the linear system

$$\begin{bmatrix} y_1 & y_2 & \cdots & y_{N+1} \\ y_2 & y_3 & \cdots & y_{N+2} \\ . & & & \\ . & & & \\ . & & & \\ y_N & y_{N+1} & \cdots & y_M \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ . \\ . \\ . \\ c_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix},$$

which we write as $Y\mathbf{c} = \mathbf{0}$. Since $Y^\dagger Y\mathbf{c} = \mathbf{0}$ also, we see that $\mathbf{c}$ is an eigenvector associated with the eigenvalue zero of the hermitian nonnegative definite matrix $Y^\dagger Y$.

Fix a value of k and replace each of the y_{k+j} in Equation (28.2) with the value given by Equation (28.1) to get

$$0 = \sum_{n=0}^{N} a_n [\sum_{j=0}^{N} c_j e^{\gamma_n (k+j+1)\Delta}]$$

$$= \sum_{n=0}^{N} a_n e^{\gamma_n (k+1)\Delta} [\sum_{j=0}^{N} c_j (e^{\gamma_n \Delta})^j].$$

Since this is true for each of the N fixed values of k, we conclude that the inner sum is zero for each n; that is,

$$\sum_{j=0}^{N} c_j (e^{\gamma_n \Delta})^j = 0,$$

for each n. Therefore, the polynomial

$$C(x) = \sum_{j=0}^{N} c_j x^j$$

has for its roots the N values $x = e^{\gamma_n \Delta}$. Once we find the roots of this polynomial we have the values of γ_n. Then, we obtain the a_n by solving a linear system of equations. In practice we would not know N so would overestimate N somewhat in selecting M. As a result, some of the a_n would be zero.

If we believe that the number N is considerably smaller than M, we do not assume that $2N = M$. Instead, we select L somewhat larger than we

believe N is and then solve the linear system

$$\begin{bmatrix} y_1 & y_2 & \cdots & y_{L+1} \\ y_2 & y_3 & \cdots & y_{L+2} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ y_{M-L} & y_{M-L+1} & \cdots & y_M \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \cdot \\ \cdot \\ \cdot \\ c_L \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 0 \end{bmatrix}.$$

This system has $M - L$ equations and $L + 1$ unknowns, so is quite overdetermined. We would then use the least-squares approach to obtain the vector $\mathbf{c}$. Again writing the system as $Y\mathbf{c} = \mathbf{0}$, we note that the matrix $Y^\dagger Y$ is $L+1$ by $L+1$ and has $\lambda = 0$ for its lowest eigenvalue; therefore, it is not invertible. When there is noise in the measurements, this matrix may become invertible, but will still have at least one very small eigenvalue.

Finding the vector $\mathbf{c}$ in either case can be tricky because we are looking for a nonzero solution of a homogeneous system of linear equations. For a discussion of the numerical issues involved in these calculations, the interested reader should consult the book by Therrien [232].

Chapter 29

Eigenvector Methods

Prony's method showed that information about the signal can sometimes be obtained from the roots of certain polynomials formed from the data. Eigenvector methods assume the data are correlation values and involve polynomials formed from the eigenvectors of the correlation matrix. Schmidt's *multiple signal classification* (MUSIC) algorithm is one such method [219]. A related technique used in direction-of-arrival array processing is the *estimation of signal parameters by rotational invariance techniques* (ESPRIT) of Paulraj, Roy, and Kailath [202].

29.1 The Sinusoids-in-Noise Model

We suppose now that the function $f(t)$ being measured is signal plus noise, with the form

$$f(t) = \sum_{j=1}^{J} |A_j| e^{i\theta_j} e^{-i\omega_j t} + n(t) = s(t) + n(t),$$

where the phases θ_j are random variables, independent and uniformly distributed in the interval $[0, 2\pi)$, and $n(t)$ denotes the random complex stationary noise component. Assume that $E(n(t)) = 0$ for all t and that the noise is independent of the signal components. We want to estimate J, the number of sinusoidal components, their magnitudes $|A_j|$ and their frequencies ω_j.

29.2 Autocorrelation

The autocorrelation function associated with $s(t)$ is

$$r_s(\tau) = \sum_{j=1}^{J} |A_j|^2 e^{-i\omega_j \tau},$$

and the signal power spectrum is the Fourier transform of $r_s(\tau)$,

$$R_s(\omega) = \sum_{j=1}^{J} |A_j|^2 \delta(\omega - \omega_j).$$

The noise autocorrelation is denoted $r_n(\tau)$ and the noise power spectrum is denoted $R_n(\omega)$. For the remainder of this section we shall assume that the noise is *white noise*; that is, $R_n(\omega)$ is constant and $r_n(\tau) = 0$ for $\tau \neq 0$.

We collect samples of the function $f(t)$ and use them to estimate some of the values of $r_s(\tau)$. From these values of $r_s(\tau)$, we estimate $R_s(\omega)$, primarily looking for the locations ω_j at which there are delta functions.

We assume that the samples of $f(t)$ have been taken over an interval of time sufficiently long to take advantage of the independent nature of the phase angles θ_j and the noise. This means that when we estimate the $r_s(\tau)$ from products of the form $f(t+\tau)\overline{f(t)}$, the cross terms between one signal component and another, as well as between a signal component and the noise, are nearly zero, due to destructive interference coming from the random phases.

29.3 The Autocorrelation Matrix

Suppose now that we have the values $r_f(m)$ for $m = -(M-1), ..., M-1$, where $M > J$, $r_f(m) = r_s(m)$ for $m \neq 0$, and $r_f(0) = r_s(0) + \sigma^2$, for σ^2 the variance (or *power*) of the noise. We form the M by M autocorrelation matrix R with entries $R_{m,k} = r_f(m-k)$.

Exercise 29.1 *Show that the matrix R has the following form:*

$$R = \sum_{j=1}^{J} |A_j|^2 \mathbf{e}_j \mathbf{e}_j^\dagger + \sigma^2 I,$$

where $\mathbf{e}_j$ is the column vector with entries $e^{-i\omega_j m}$, for $m = 0, 1, ..., M-1$.

Let $\mathbf{u}$ be an eigenvector of R with $\|u\| = 1$ and associated eigenvalue λ. Then we have

$$\lambda = \mathbf{u}^\dagger R\mathbf{u} = \sum_{j=1}^{J} |A_j|^2 |\mathbf{e}_j^\dagger \mathbf{u}|^2 + \sigma^2 \geq \sigma^2.$$

Therefore, the smallest eigenvalue of R is σ^2

Because $M > J$, there must be non-zero M-dimensional vectors $\mathbf{v}$ that are orthogonal to all of the $\mathbf{e}_j$; in fact, we can say that there are $M - J$ linearly independent such $\mathbf{v}$. For each such vector $\mathbf{v}$ we have

$$R\mathbf{v} = \sum_{j=1}^{J} |A_j|^2 \mathbf{e}_j^\dagger \mathbf{v} \mathbf{e}_j + \sigma^2 \mathbf{v} = \sigma^2 \mathbf{v};$$

consequently, $\mathbf{v}$ is an eigenvector of R with associated eigenvalue σ^2.

Let $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_M > 0$ be the eigenvalues of R and let $\mathbf{u}^m$ be a norm-one eigenvector associated with λ_m. It follows from the previous paragraph that $\lambda_m = \sigma^2$, for $m = J + 1, ..., M$, while $\lambda_m > \sigma^2$ for $m = 1, ..., J$. This leads to the MUSIC method.

29.4 The MUSIC Method

By calculating the eigenvalues of R and noting how many of them are greater than the smallest one, we find J. Now we seek the ω_j.

For each ω let $\mathbf{e}_\omega$ have the entries $e^{-i\omega m}$ and form the function

$$T(\omega) = \sum_{m=J+1}^{M} |\mathbf{e}_\omega^\dagger \mathbf{u}^m|^2.$$

This function $T(\omega)$ will have zeros at precisely the values $\omega = \omega_j$, for $j = 1, ..., J$. Once we have determined J and the ω_j, we estimate the magnitudes $|A_j|$ using Fourier transform estimation techniques already discussed. This is basically Schmidt's MUSIC method [219].

Chapter 30

A Little Optimization

30.1 Image Reconstruction Through Optimization

In our discussion of both transmission and emission tomography we saw that discretization leads to systems of linear equations to be solved for the vectorized image x. Typically, these systems are quite large, the measured data is noisy, and there will be no non-negative x satisfying the system exactly. In such cases, one can turn to optimization, and calculate a non-negatively constrained least-squares solution, with or without a penalty term.

In the stochastic approach to emission tomography, we maximize the likelihood function with respect to the unknown image vector x. Here again, optimization plays a role. It is reasonable, therefore, to take a brief look at the theory of optimization, particularly constrained optimization. In this chapter we discuss optimization with equality constraints and the area known as *convex programming* (CP).

30.2 Eigenvalues and Eigenvectors Through Optimization

Let B be any real I by J matrix. We want to find the maximum value of the ratio $||Bx||/||x||$, over all non-zero vectors x. If $\hat{x}$ solves this problem, so does $c\hat{x}$ for every non-zero real number c; therefore, we may and do constrain the vectors x to have $||x|| = 1$.

We reformulate the problem as follows: maximize $f(x) = ||Bx||^2$, subject to $g(x) = ||x||^2 = 1$. Our approach will be to use the method of *Lagrange multipliers*. Suppose that $\hat{x}$ is a solution and S is the level sur-

face of the function $f(x)$ containing the vector $\hat{x}$, that is,

$$S = \{x | f(x) = f(\hat{x})\}.$$

The gradient of $f(x)$ at $\hat{x}$ is a vector normal to S at $\hat{x}$. Now let U be the unit surface of all x with $||x|| = 1$. We claim that S and U must be tangent at $x = \hat{x}$. If that is not the case, then U cuts through S, making it possible to move from one side of S to the other side of S, while remaining on the surface U. Therefore, we would be able to move along U to another vector x with $f(x) > f(\hat{x})$, which cannot happen.

Since the two surfaces are tangent at $x = \hat{x}$, their gradients are parallel, so that

$$\nabla f(\hat{x}) = \alpha \nabla g(\hat{x}),$$

for some constant α. Equivalently,

$$\nabla f(\hat{x}) + (-\alpha)\nabla g(\hat{x}) = 0.$$

The main idea of the Lagrange-multiplier method is to define the Lagrangian as

$$L(x; \lambda) = f(x) + \lambda g(x),$$

so that, for some value of the parameter λ the gradient of $L(x; \lambda)$ is zero; here $\lambda = -\alpha$ works.

The *Lagrangian* for this problem is

$$L(x, \lambda) = f(x) + \lambda g(x) = ||Bx||^2 + \lambda ||x||^2.$$

Therefore, we have

$$2B^T B\hat{x} + 2\lambda \hat{x} = 0,$$

or

$$B^T B\hat{x} = \alpha \hat{x},$$

which tells us that $\hat{x}$ is an *eigenvector* of the matrix $B^T B$ corresponding to the *eigenvalue* α. Since the matrix $B^T B$ is symmetric, all its eigenvalues are real numbers; in fact, $B^T B$ is non-negative definite, so all its eigenvalues are non-negative.

Since

$$||B\hat{x}||^2 = \hat{x}^T B^T B\hat{x} = \alpha \hat{x}^T \hat{x} = \alpha ||\hat{x}||^2 = \alpha,$$

we see that the largest value of $||Bx||^2$, subject to $||x|| = 1$, must be α. So α is the largest eigenvalue of the matrix $B^T B$ and $\hat{x}$ is an associated eigenvector.

The largest eigenvalue of $B^T B$ is also the largest eigenvalue of the matrix BB^T and is denoted $\rho(B^T B) = \rho(BB^T)$, and called the *spectral radius* of $B^T B$. We can therefore write

$$||Bz||^2 \leq \rho(B^T B)||z||^2, \tag{30.1}$$

for all vectors z.

30.3 Convex Sets and Convex Functions

A subset C of R^J is said to be *convex* if, for every collection $c_1, c_2, ..., c_N$ of points in C and all positive constants $a_1, a_2, ..., a_N$ summing to one, the point $a_1c_1 + ... + a_Nc_N$ is again in C. A function $f : R^J \rightarrow R$ is said to be a *convex function* on the convex set C if, for all such combinations as above, we have

$$f(a_1c_1 + ... + a_Nc_N) \leq a_1f(c_1) + ... + a_Nf(c_N).$$

The function $f(x) = ||Ax - b||^2$ is convex on $C = R^J$ and the function $f(x) = KL(b, Ax)$ is convex on the set C of non-negative x in R^J.

30.4 The Convex Programming Problem

Let f and g_i, $i = 1, ..., I$, be convex functions defined on a non-empty closed convex subset C of R^J. The *primal problem* in *convex programming* (CP) is the following:

$$\text{minimize } f(x), \text{ subject to } g_i(x) \leq 0, \text{ for } i = 1, ..., I. \quad \text{(P)} \tag{30.2}$$

For notational convenience, we define $g(x) = (g_1(x), ..., g_I(x))$. Then (P) becomes

$$\text{minimize } f(x), \text{ subject to } g(x) \leq 0. \quad \text{(P)} \tag{30.3}$$

The *feasible set* for (P) is

$$F = \{x | g(x) \leq 0\}. \tag{30.4}$$

Definition 30.1 *The problem (P) is said to be* consistent *if F is not empty, and* super-consistent *if there is x in F with $g_i(x) < 0$ for all $i = 1, ..., I$. Such a point x is then called a Slater point.*

Definition 30.2 *The* Lagrangian *for the problem (P) is the function*

$$L(x, \lambda) = f(x) + \sum_{i=1}^{I} \lambda_i g_i(x), \tag{30.5}$$

defined for all x in C and $\lambda \geq 0$.

30.5 A Simple Example

Let us minimize the function $f : R^2 \rightarrow R$ given by

$$f(x, y) = (x + 1)^2 + y^2,$$

subject to $x \geq 0$ and $y \geq 0$. To get this problem into the form of the CP problem we introduce the functions

$$g_1(x, y) = -x,$$

and

$$g_2(x, y) = -y.$$

The partial derivative of f, with respect to x, is

$$\frac{\partial f}{\partial x}(x, y) = 2(x + 1),$$

and the partial derivative of f, with respect to y, is

$$\frac{\partial f}{\partial y}(x, y) = 2y.$$

If we simply set both partial derivatives to zero, we get $x = -1$ and $y = 0$, which is, of course, the unconstrained minimizing point for f. But this point does not satisfy our constraints.

If we graph the function, we see immediately that the constrained solution is the origin, $x = 0$ and $y = 0$. At this point, we can move up or down without decreasing f, and this is reflected in the fact that the y-partial derivative at $(0, 0)$ is zero. The x-partial derivative at $(0, 0)$ is not zero, however, since, if we move horizontally to the left, the function f decreases. However, we are prevented from moving left by the constraint that $x \geq 0$, so it is not necessary that the x-partial derivative be zero at the solution. We only need to know that if we move to the right, which is permitted by the constraints, the function f increases; the fact that the x-partial derivative is positive at $(0, 0)$ guarantees this.

30.6 The Karush-Kuhn-Tucker Theorem

As we have just seen, at the solution of a CP problem it is not necessarily the case that the partial derivatives all be zero. But what does have to be the case?

The Karush-Kuhn Tucker Theorem gives necessary and sufficient conditions for a vector x^* to be a solution of a super-consistent problem (P).

Theorem 30.1 *Let (P) be super-consistent. Then x^* solves (P) if and only if there is a vector λ^* such that*

- **1)** $\lambda^* \geq 0$*;*
- **2)** $\lambda_i^* g_i(x^*) = 0$*, for all $i = 1, ..., I$;*

- **3)** $\nabla f(x^*) + \sum_{i=1}^{I} \lambda_i^* \nabla g_i(x^*) = 0.$

We saw in the first section that when we optimize subject to an equality constraint the first condition of the KKT Theorem need not hold, that is, the Lagrange multipliers need not be non-negative, and the second condition is automatically true, since the constraints are now $g_i(x) = 0$ for all i.

30.7 Back to our Example

Once again, the problem is to minimize $f(x, y) = (x + 1)^2 + y^2$, subject to $g_1(x, y) = -x \leq 0$ and $g_2(x, y) = -y \leq 0$. Applying Condition 3 of the KKT Theorem, we get

$$0 = 2(x + 1) - \lambda_1^*,$$

and

$$0 = 2y - \lambda_2^*.$$

From Condition 2 we know that either $\lambda_1^* = 0$, which can't happen, since then $x = -1$, or $x = 0$; therefore $x = 0$. Also from Condition 2 we know that either $\lambda_2^* = 0$ or $y = 0$; therefore, $y = 0$. We have found the solution to our constrained minimization problem.

30.8 Two More Examples

We illustrate the use of the gradient form of the KKT Theorem with two more examples that appeared in the paper of Driscoll and Fox [108].

30.8.1 A Linear Programming Problem

Minimize $f(x_1, x_2) = 3x_1 + 2x_2$, subject to the constraints $2x_1 + x_2 \geq 100$, $x_1 + x_2 \geq 80$, $x_1 \geq 0$ and $x_2 \geq 0$. We define

$$g_1(x_1, x_2) = 100 - 2x_1 - x_2 \leq 0, \tag{30.6}$$

$$g_2(x_1, x_2) = 80 - x_1 - x_2, \tag{30.7}$$

$$g_3(x_1, x_2) = -x_1, \tag{30.8}$$

and

$$g_4(x_1, x_2) = -x_2. \tag{30.9}$$

The Lagrangian is then

$$L(x,\lambda) = 3x_1 + 2x_2 + \lambda_1(100 - 2x_1 - x_2)$$

$$+\lambda_2(80 - x_1 - x_2) - \lambda_3 x_1 - \lambda_4 x_2. \tag{30.10}$$

From the KKT Theorem, we know that if there is a solution x^*, then there is $\lambda^* \geq 0$ with

$$f(x^*) = L(x^*, \lambda^*) \leq L(x, \lambda^*),$$

for all x. For notational simplicity, we write λ in place of λ^*.

Taking the partial derivatives of $L(x,\lambda)$ with respect to the variables x_1 and x_2, we get

$$3 - 2\lambda_1 - \lambda_2 - \lambda_3 = 0, \tag{30.11}$$

and

$$2 - \lambda_1 - \lambda_2 - \lambda_4 = 0. \tag{30.12}$$

The complementary slackness conditions are

$$\lambda_1 = 0 \ , \text{ if } 2x_1 + x_2 \neq 100, \tag{30.13}$$

$$\lambda_2 = 0 \ , \text{ if } x_1 + x_2 \neq 80, \tag{30.14}$$

$$\lambda_3 = 0 \ , \text{ if } x_1 \neq 0, \tag{30.15}$$

and

$$\lambda_4 = 0 \ , \text{ if } x_2 \neq 0. \tag{30.16}$$

A little thought reveals that precisely two of the four constraints must be binding. Examining the six cases, we find that the only case satisfying all the conditions of the KKT Theorem is $\lambda_3 = \lambda_4 = 0$. The minimum occurs at $x_1 = 20$ and $x_2 = 60$ and the minimum value is $f(20, 60) = 180$.

30.8.2 A Nonlinear Convex Programming Problem

Minimize the function

$$f(x_1, x_2) = (x_1 - 14)^2 + (x_2 - 11)^2,$$

subject to

$$g_1(x_1, x_2) = (x_1 - 11)^2 + (x_2 - 13)^2 - 49 \leq 0,$$

and

$$g_2(x_1, x_2) = x_1 + x_2 - 19 \leq 0.$$

The Lagrangian is then

$$L(x, \lambda) = (x_1 - 14)^2 + (x_2 - 11)^2 +$$

$$\lambda_1\Big((x_1 - 11)^2 + (x_2 - 13)^2 - 49\Big) + \lambda_2\Big(x_1 + x_2 - 19\Big). \qquad (30.17)$$

Again, we write λ in place of λ^*. Setting the partial derivatives, with respect to x_1 and x_2, to zero, we get the KKT equations

$$2x_1 - 28 + 2\lambda_1 x_1 - 22\lambda_1 + \lambda_2 = 0, \qquad (30.18)$$

and

$$2x_2 - 22 + 2\lambda_1 x_2 - 26\lambda_1 + \lambda_2 = 0. \qquad (30.19)$$

The complementary slackness conditions are

$$\lambda_1 = 0\,, \quad \text{if } (x_1 - 11)^2 + (x_2 - 13)^2 \neq 49, \qquad (30.20)$$

and

$$\lambda_2 = 0\,, \quad \text{if } x_1 + x_2 \neq 19. \qquad (30.21)$$

There are four cases to consider. First, if neither constraint is binding, the KKT equations have solution $x_1 = 14$ and $x_2 = 11$, which is not feasible. If only the first constraint is binding, we obtain two solutions, neither feasible. If only the second constraint is binding, we obtain $x_1^* = 11$, $x_2^* = 8$, and $\lambda_2 = 6$. This is the optimal solution. If both constraints are binding, we obtain, with a bit of calculation, two solutions, neither feasible. The minimum value is $f(11, 8) = 18$, and the sensitivity vector is $\lambda^* = (0, 6)$.

30.9 Non-Negatively Constrained Least-Squares

If there is no solution to a system of linear equations $Ax = b$, then we may seek a *least-squares* "solution" , which is a minimizer of the function

$$f(x) = \sum_{i=1}^{I} \Big((\sum_{m=1}^{J} A_{im} x_m) - b_i \Big)^2 = ||Ax - b||^2.$$

The partial derivative of $f(x)$ with respect to the variable x_j is

$$\frac{\partial f}{\partial x_j}(x) = 2 \sum_{i=1}^{I} A_{ij} \Big((\sum_{m=1}^{J} A_{im} x_m) - b_i \Big).$$

Setting the gradient equal to zero, we find that to get a least-squares solution we must solve the system of equations

$$A^T(Ax - b) = 0.$$

Now we consider what happens when the additional constraints $x_j \geq 0$ are imposed.

This problem fits into the CP framework, when we define

$$g_j(x) = -x_j,$$

for each j. Let $\hat{x}$ be a least-squares solution. According to the KKT Theorem, for those values of j for which $\hat{x}_j$ is not zero we have $\lambda_j^* = 0$ and $\frac{\partial f}{\partial x_j}(\hat{x}) = 0$. Therefore, if $\hat{x}_j \neq 0$,

$$0 = \sum_{i=1}^{I} A_{ij}\Big((\sum_{m=1}^{J} A_{im}\hat{x}_m) - b_i\Big).$$

Let Q be the matrix obtained from A by deleting rows j for which $\hat{x}_j = 0$. Then we can write

$$Q^T(A\hat{x} - b) = 0.$$

If Q has at least I columns and has full rank, then Q^T is a one-to-one linear transformation, which implies that $A\hat{x} = b$. Therefore, when there is no non-negative solution of $Ax = b$, Q must have fewer than I columns, which means that $\hat{x}$ has fewer than I non-zero entries. This is the proof of Theorem 11.1.

This result has some practical implications in medical image reconstruction. In the hope of improving the resolution of the reconstructed image, we may be tempted to take J, the number of pixels, larger than I, the number of equations arising from photon counts or line integrals. Since the vector b consists of measured data, it is noisy and there may well not be a non-negative solution of $Ax = b$. As a result, the image obtained by non-negatively constrained least-squares will have at most $I - 1$ non-zero entries; many of the pixels will be zero and they will be scattered throughout the image, making it unusable for diagnosis. The reconstructed images resemble stars in a night sky, and, as a result, the theorem is sometimes described as the "night sky" theorem.

This "night sky" phenomenon is not restricted to least squares. The same thing happens with methods based on the Kullback-Leibler distance, such as MART, EMML and SMART.

30.10 The EMML Algorithm

Maximizing the likelihood function in SPECT is equivalent to minimizing the KL distance $KL(b, Ax)$ over non-negative vectors x, where b is the

vector of photon counts at the detectors and A the matrix of detection probabilities. With $f(x) = KL(b, Ax)$ and $g_j(x) = -x_j$, the problem becomes a CP problem. We have

$$\frac{\partial f}{\partial x_j}(x) = \sum_{i=1}^{I} A_{ij}\Big(1 - b_i/(Ax)_i\Big),$$

where

$$(Ax)_i = \sum_{m=1}^{J} A_{im}x_m.$$

Let $\hat{x}$ be the solution. According to the KKT Theorem, one of two things are possible: for each j either 1): $\hat{x}_j = 0$ or 2): both $\lambda_j^* = 0$ and, consequently,

$$\frac{\partial f}{\partial x_j}(\hat{x}) = 0.$$

Therefore, for all values of the index j we have

$$0 = \hat{x}_j \sum_{i=1}^{I} A_{ij}\Big(1 - b_i/(A\hat{x})_i\Big),$$

or, equivalently,

$$\hat{x}_j = s_j^{-1} \sum_{i=1}^{I} A_{ij}\Big(b_i/(A\hat{x})_i\Big),$$

where $s_j = \sum_{i=1}^{I} A_{ij}$.

This suggests an iterative optimization algorithm whereby we insert the current value of the vector, call it x^k, into the right side of the last equation, and call the resulting vector the next iterate, x^{k+1}. For simplicity, we assume $s_j = 1$. Then the iteration becomes

$$x_j^{k+1} = x_j^k\Big(\sum_{i=1}^{I} A_{ij}(b_i/(Ax^k)_i)\Big). \tag{30.22}$$

This is the EMML iterative algorithm.

30.11 The Simultaneous MART Algorithm

The MART algorithm has the following iterative step:

$$x_j^{k+1} = x_j^k\Big(b_i/(Ax^k)_i\Big)^{A_{ij}},$$

where $i = k(\mathrm{mod}\, I) + 1$. The MART uses only one equation at each step. The simultaneous MART (SMART) uses all the equations at each step. Assuming once again that $s_j = 1$ for all j, the iterative step of the SMART is

$$x_j^{k+1} = x_j^k \exp\Big(\sum_{i=1}^{I} A_{ij} \log(b_i/(Ax^k)_i)\Big). \tag{30.23}$$

The SMART is clearly closely related to the EMML algorithm, with subtle differences, namely the exponentiation and the logarithm. As we shall show in the next chapter, the SMART algorithm minimizes the function $KL(Ax, b)$, while the EMML minimizes $KL(b, Ax)$.

Chapter 31

Using Prior Knowledge

A basic problem in signal processing is the estimation of the function $F(\gamma)$ from finitely many values of its inverse Fourier transform $f(x)$. The DFT is one such estimator. As we shall see in this section, there are other estimators that are able to make better use of prior information about $F(\gamma)$ and thereby provide a better estimate.

31.1 Over-Sampling

We assume, for the moment, that $F(\gamma) = 0$ for $|\gamma| > \Gamma$ and that $\Delta = \frac{\pi}{\Gamma}$. In Figure 31.1 below, we show the DFT estimate for $F(\gamma)$ for a case in which $\Gamma = \frac{\pi}{30}$. This would tell us that the proper sampling spacing is $\Delta = 30$. However, it is not uncommon to have situations in which x is time and we can take as many samples of $f(x)$ as we wish, but must take the samples at points x within some limited time interval, say $[0, A]$. In the case considered in the figure, $A = 130$. If we had used $\Delta = 30$, we would have obtained only four data points, which is not sufficient information. Instead, we used $\Delta = 1$ and took $N = 129$ data points; we *over-sampled*. There is a price to be paid for over-sampling, however.

The DFT estimation procedure does not "know" about the true value of Γ; it only "sees" Δ. It "assumes" incorrectly that Γ must be π, since $\Delta = 1$. Consequently, it "thinks" that we want it to estimate $F(\gamma)$ on the interval $[-\pi, \pi]$. It doesn't "know" that we know that $F(\gamma)$ is zero on most of this interval. Therefore, the DFT spends a lot of its energy trying to describe the part of the graph of $F(\gamma)$ where it is zero, and relatively little of its energy describing what is happening within the interval $[-\Gamma, \Gamma]$, which is all that we are interested in. This is why the bottom graph in the figure shows the DFT to be poor within $[-\Gamma, \Gamma]$. There is a second graph in the figure. It looks quite a bit better. How was that graph obtained?

We know that $F(\gamma) = 0$ outside the interval $[-\Gamma, \Gamma]$. Can we somehow let the estimation process know that we know this, so that it doesn't waste its energy outside this interval? Yes, we can.

The *characteristic function* of the interval $[-\Gamma, \Gamma]$ is

$$\chi_\Gamma(\gamma) = \begin{cases} 1, \text{if } |\gamma| \leq \Gamma\,; \\ 0, \text{if } |\gamma| > \Gamma\,. \end{cases}$$

We take as our estimator of $F(\gamma)$ a function called the *modified* DFT, (MDFT) having the form

$$MDFT(\gamma) = \chi_\Gamma(\gamma) \sum_{m=0}^{N-1} a_m e^{im\Delta\gamma}. \tag{31.1}$$

We determine the coefficients a_m by making $MDFT(\gamma)$ consistent with the data. Inserting $MDFT(\gamma)$ into the integral in Equation (27.2) and setting $x = n\Delta$, for each $n = 0, 1, ..., N-1$, in turn, we find that we must have

$$f(n\Delta) = \frac{1}{2\pi} \sum_{m=0}^{N-1} a_m \int_{-\Gamma}^{\Gamma} e^{i(m-n)\Delta\gamma} d\gamma.$$

Performing the integration, we find that we need

$$f(n\Delta) = \sum_{m=0}^{N-1} a_m \frac{\sin(\Gamma(n-m)\Delta)}{\pi(n-m)\Delta}, \tag{31.2}$$

for $n = 0, 1, ..., N-1$. We solve for the a_m and insert these coefficients into the formula for the MDFT. The graph of the MDFT is the top graph in the figure.

The main idea in the MDFT is to use a form of the estimator that already includes whatever important features of $F(\gamma)$ we may know a priori. In the case of the MDFT, we knew that $F(\gamma) = 0$ outside the interval $[-\Gamma, \Gamma]$, so we introduced a factor of $\chi_\Gamma(\gamma)$ in the estimator. Now, whatever coefficients we use, any estimator of the form given in Equation (31.1) will automatically be zero outside $[-\Gamma, \Gamma]$. We are then free to select the coefficients so as to make the MDFT consistent with the data. This involves solving the system of linear equations in (31.2).

31.2 Using Other Prior Information

The approach that led to the MDFT estimate suggests that we can introduce other prior information besides the support of $F(\gamma)$. For example, if we have some idea of the overall shape of the function $F(\gamma)$, we could

choose $P(\gamma) > 0$ to indicate this shape and use it instead of $\chi_\Gamma(\gamma)$ in our estimator. This leads to the PDFT estimator, which has the form

$$PDFT(\gamma) = P(\gamma) \sum_{n=0}^{N-1} b_m e^{im\Delta\gamma}. \tag{31.3}$$

Now we find the b_m by forcing the right side of Equation (31.3) to be consistent with the data. Inserting the function $PDFT(\gamma)$ into the integral in Equation (27.2), we find that we must have

$$f(n\Delta) = \frac{1}{2\pi} \sum_{m=0}^{N-1} b_m \int_{-\infty}^{\infty} P(\gamma) e^{i(m-n)\Delta\gamma} d\gamma. \tag{31.4}$$

Using $p(x)$, the inverse Fourier transform of $P(\gamma)$, given by

$$p(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} P(\gamma) e^{-ix\gamma} d\gamma,$$

we find that we must have

$$f(n\Delta) = \sum_{m=0}^{N-1} b_m p((n-m)\Delta), \tag{31.5}$$

for $n = 0, 1, ..., N-1$. We solve this system of equations for the b_m and insert them into the PDFT estimator in Equation (31.3).

In Figure 31.2 we have the function $F(\gamma)$ in the upper left corner. It consists of one large bump in the center and one smaller bump toward the right side. The DFT on the upper right side gives only slight indication that the smaller bump exists. The data here is somewhat over-sampled, so we can try the MDFT. The prior for the MDFT is $P(\gamma) = \chi_\Gamma(\gamma)$, which is pictured in the center left frame; it is shown only over $[-\Gamma, \Gamma]$, where it is just one. The MDFT estimate is in the center right frame; it shows only slight improvement over the DFT. Now, suppose we know that there is a large bump in the center. Both the DFT and the MDFT tell us clearly that this is the case, so even if we did not know it at the start, we know it now. Let's select as our prior a function $P(\gamma)$ that includes the big bump in the center, as shown in the lower left. The PDFT on the lower right now shows the smaller bump more clearly.

A more dramatic illustration of the use of the PDFT is shown in Figure 31.3. The function $F(\gamma)$ is a function of two variables simulating a slice of a head. It has been approximated by a discrete image, called here the "original" . The data was obtained by taking the two-dimensional vector DFT of the discrete image and replacing most of its values with zeros. When we formed the inverse vector DFT, we obtained the estimate in the lower

right. This is essentially the DFT estimate, and it tells us nothing about the inside of the head. From prior information, or even from the DFT estimate itself, we know that the true $F(\gamma)$ includes a skull. We therefore select as our prior the (discretized) function of two variables shown in the upper left. The PDFT estimate is the image in the lower left. The important point to remember here is that the same data was used to generate both pictures.

We saw previously how the MDFT can improve the estimate of $F(\gamma)$, by incorporating the prior information about its support. Precisely why the improvement occurs is the subject of the next section.

31.3 Analysis of the MDFT

Let our data be $f(x_m)$, $m = 1, ..., M$, where the x_m are arbitrary values of the variable x. If $F(\gamma)$ is zero outside $[-\Gamma, \Gamma]$, then minimizing the energy over $[-\Gamma, \Gamma]$ subject to data consistency produces an estimate of the form

$$F_\Gamma(\gamma) = \chi_\Gamma(\gamma) \sum_{m=1}^{M} b_m \exp(ix_m\gamma),$$

with the b_m satisfying the equations

$$f(x_n) = \sum_{m=1}^{M} b_m \frac{\sin(\Gamma(x_m - x_n))}{\pi(x_m - x_n)},$$

for $n = 1, ..., M$. The matrix S_Γ with entries $\frac{\sin(\Gamma(x_m - x_n))}{\pi(x_m - x_n)}$ we call a *sinc* matrix.

31.3.1 Eigenvector Analysis of the MDFT

Although it seems reasonable that incorporating the additional information about the support of $F(\gamma)$ should improve the estimation, it would be more convincing if we had a more mathematical argument to make. For that we turn to an analysis of the eigenvectors of the sinc matrix. Throughout this subsection we make the simplification that $x_n = n$.

Exercise 31.1 *The purpose of this exercise is to show that, for an Hermitian nonnegative-definite M by M matrix Q, a norm-one eigenvector $\mathbf{u}^1$ of Q associated with its largest eigenvalue, λ_1, maximizes the quadratic form $\mathbf{a}^\dagger Q\mathbf{a}$ over all vectors $\mathbf{a}$ with norm one. Let $Q = ULU^\dagger$ be the eigenvector decomposition of Q, where the columns of U are mutually orthogonal eigenvectors $\mathbf{u}^n$ with norms equal to one, so that $U^\dagger U = I$, and*

$L = diag\{\lambda_1, ..., \lambda_M\}$ is the diagonal matrix with the eigenvalues of Q as its entries along the main diagonal. Assume that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_M$. Then maximize

$$\mathbf{a}^\dagger Q\mathbf{a} = \sum_{n=1}^{M} \lambda_n \, |\mathbf{a}^\dagger \mathbf{u}^n|^2,$$

subject to the constraint

$$\mathbf{a}^\dagger \mathbf{a} = \mathbf{a}^\dagger U^\dagger U\mathbf{a} = \sum_{n=1}^{M} |\mathbf{a}^\dagger \mathbf{u}^n|^2 = 1.$$

Hint: Show $\mathbf{a}^\dagger Q\mathbf{a}$ is a convex combination of the eigenvalues of Q.

Exercise 31.2 *Show that, for the sinc matrix $Q = S_\Gamma$, the quadratic form $\mathbf{a}^\dagger Q\mathbf{a}$ in the previous exercise becomes*

$$\mathbf{a}^\dagger S_\Gamma \mathbf{a} = \frac{1}{2\pi} \int_{-\Gamma}^{\Gamma} |\sum_{n=1}^{M} a_n e^{in\gamma}|^2 d\gamma.$$

Show that the norm of the vector $\mathbf{a}$ is the integral

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} |\sum_{n=1}^{M} a_n e^{in\gamma}|^2 d\gamma.$$

Exercise 31.3 *For $M = 30$ compute the eigenvalues of the matrix S_Γ for various choices of Γ, such as $\Gamma = \frac{\pi}{k}$, for $k = 2, 3, ..., 10$. For each k arrange the set of eigenvalues in decreasing order and note the proportion of them that are not near zero. The set of eigenvalues of a matrix is sometimes called its eigenspectrum and the nonnegative function $\chi_\Gamma(\gamma)$ is a power spectrum; here is one time in which different notions of a spectrum are related.*

31.3.2 The Eigenfunctions of S_Γ

Suppose that the vector $\mathbf{u}^1 = (u_1^1, ..., u_M^1)^T$ is an eigenvector of S_Γ corresponding to the largest eigenvalue, λ_1. Associate with $\mathbf{u}^1$ the *eigenfunction*

$$U^1(\gamma) = \sum_{n=1}^{M} u_n^1 e^{in\gamma}.$$

Then

$$\lambda_1 = \int_{-\Gamma}^{\Gamma} |U^1(\gamma)|^2 d\gamma / \int_{-\pi}^{\pi} |U^1(\gamma)|^2 d\gamma$$

and $U^1(\gamma)$ is the function of its form that is most concentrated within the interval $[-\Gamma, \Gamma]$.

Similarly, if $\mathbf{u}^M$ is an eigenvector of S_Γ associated with the smallest eigenvalue λ_M, then the corresponding eigenfunction $U^M(\gamma)$ is the function of its form least concentrated in the interval $[-\Gamma, \Gamma]$.

Exercise 31.4 *Plot for $|\gamma| \leq \pi$ the functions $|U^m(\gamma)|$ corresponding to each of the eigenvectors of the sinc matrix S_Γ. Pay particular attention to the places where each of these functions is zero.*

The eigenvectors of S_Γ corresponding to different eigenvalues are orthogonal, that is $(\mathbf{u}^m)^\dagger \mathbf{u}^n = 0$ if m is not n. We can write this in terms of integrals:

$$\int_{-\pi}^{\pi} U^n(\gamma) \overline{U^m(\gamma)} d\gamma = 0$$

if m is not n. The mutual orthogonality of these eigenfunctions is related to the locations of their roots, which were studied in the previous exercise.

Any Hermitian matrix Q is invertible if and only if none of its eigenvalues is zero. With λ_m and $\mathbf{u}^m$, $m = 1, ..., M$, the eigenvalues and eigenvectors of Q, the inverse of Q can then be written as

$$Q^{-1} = (1/\lambda_1)\mathbf{u}^1(\mathbf{u}^1)^\dagger + ... + (1/\lambda_M)\mathbf{u}^M(\mathbf{u}^M)^\dagger.$$

Exercise 31.5 *Show that the MDFT estimator given by Equation (31.1) $F_\Gamma(\gamma)$ can be written as*

$$F_\Gamma(\gamma) = \chi_\Gamma(\gamma) \sum_{m=1}^{M} \frac{1}{\lambda_m} (\mathbf{u}^m)^\dagger \mathbf{d}\, U^m(\gamma),$$

where $\mathbf{d} = (f(1), f(2), ..., f(M))^T$ is the data vector.

Exercise 31.6 *Show that the DFT estimate of $F(\gamma)$, restricted to the interval $[-\Gamma, \Gamma]$, is*

$$F_{DFT}(\gamma) = \chi_\Gamma(\gamma) \sum_{m=1}^{M} (\mathbf{u}^m)^\dagger \mathbf{d}\, U^m(\gamma).$$

From these two exercises we can learn why it is that the estimate $F_\Gamma(\gamma)$ resolves better than the DFT. The former makes more use of the eigenfunctions $U^m(\gamma)$ for higher values of m, since these are the ones for which λ_m is closer to zero. Since those eigenfunctions are the ones having most of their roots within the interval $[-\Gamma, \Gamma]$, they have the most flexibility within that region and are better able to describe those features in $F(\gamma)$ that are not resolved by the DFT.

31.4 The Discrete PDFT (DPDFT)

The derivation of the PDFT assumes a function $f(x)$ of one or more continuous real variables, with the data obtained from $f(x)$ by integration. The discrete PDFT (DPDFT) begins with $f(x)$ replaced by a finite vector $f = (f_1, ..., f_J)^T$ that is a discretization of $f(x)$; say that $f_j = f(x_j)$ for some point x_j. The integrals that describe the Fourier transform data can be replaced by finite sums,

$$F(\gamma_n) = \sum_{j=1}^{J} f_j E_{nj}, \tag{31.6}$$

where $E_{nj} = e^{ix_j\gamma_n}$. We have used a Riemann-sum approximation of the integrals here, but other choices are also available. The problem then is to solve this system of equations for the f_j.

Since the N is fixed, but the J is under our control, we select $J > N$, so that the system becomes under-determined. Now we can use minimum-norm and minimum-weighted-norms solutions of the finite-dimensional problem to obtain an approximate, discretized PDFT solution.

Since the PDFT is a minimum-weighted norm solution in the continous-variable formulation, it is reasonable to let the DPDFT be the corresponding minimum-weighted-norm solution obtained with the positive-definite matrix Q the diagonal matrix having for its jth diagonal entry

$$Q_{jj} = 1/p(x_j), \tag{31.7}$$

if $p(x_j) > 0$, and zero, otherwise.

31.4.1 Calculating the DPDFT

The DPDFT is a minimum-weighted-norm solution, which can be calculated using, say, the ART algorithm. We know that, in the under-determined case, the ART provides the the solution closest to the starting vector, in the sense of the Euclidean distance. We therefore reformulate the system, so that the minimum-weighted norm solution becomes a minimum-norm solution, as we did earlier, and then begin the ART iteration with zero. For recent work involving the DPDFT se [223, 222, 224].

31.4.2 Regularization

We noted earlier that one of the principles guiding the estimation of $f(x)$ from Fourier transform data should be that we do not want to overfit the estimate to noisy data. In the PDFT, this can be avoided by adding a small positive quantity to the main diagonal of the matrix P. In the DPDFT, sensitivity to noise is reduced by using the iterative regularized ART [63].

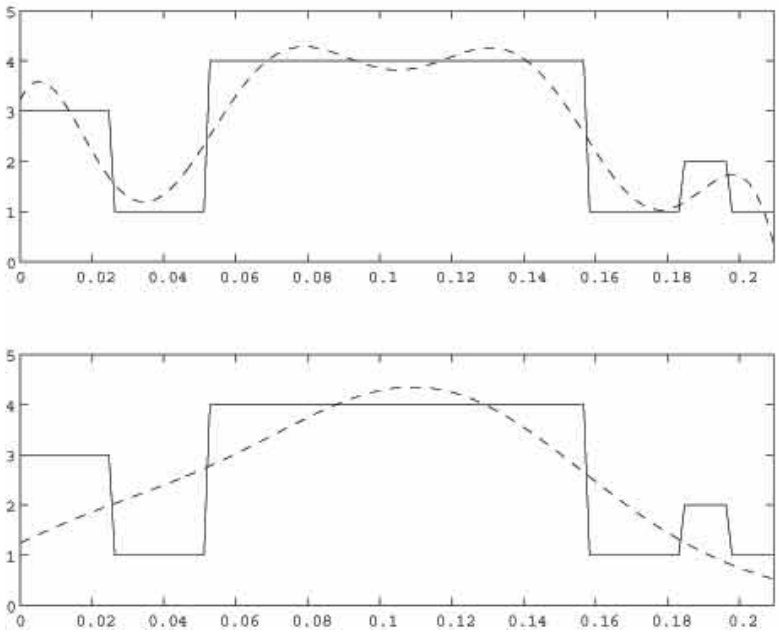

Figure 31.1: The non-iterative band-limited extrapolation method (MDFT) (top) and the DFT (bottom) for $N = 129$, $\Delta = 1$ and $\Gamma = \pi/30$.

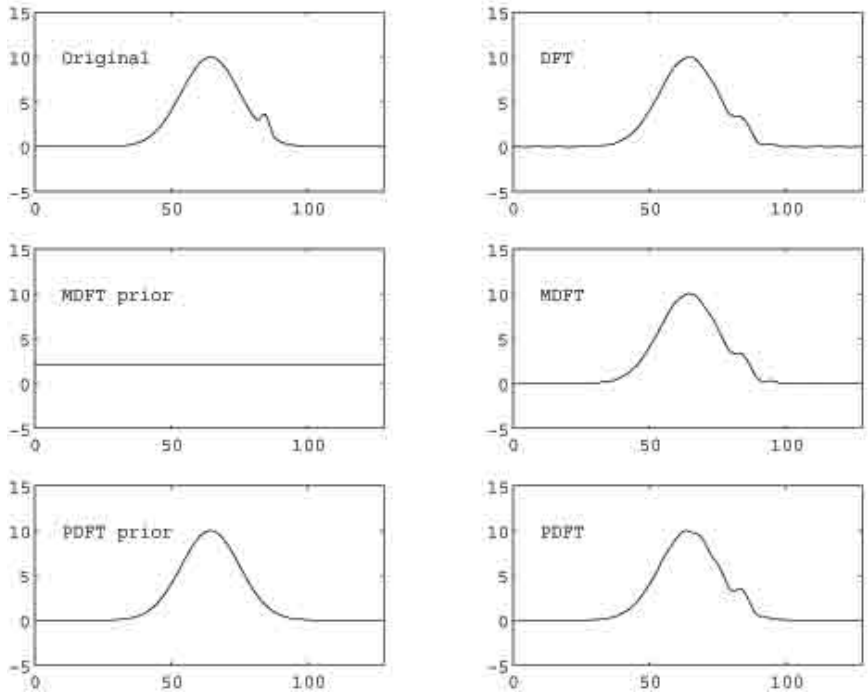

Figure 31.2: The DFT, the MDFT, and the PDFT.

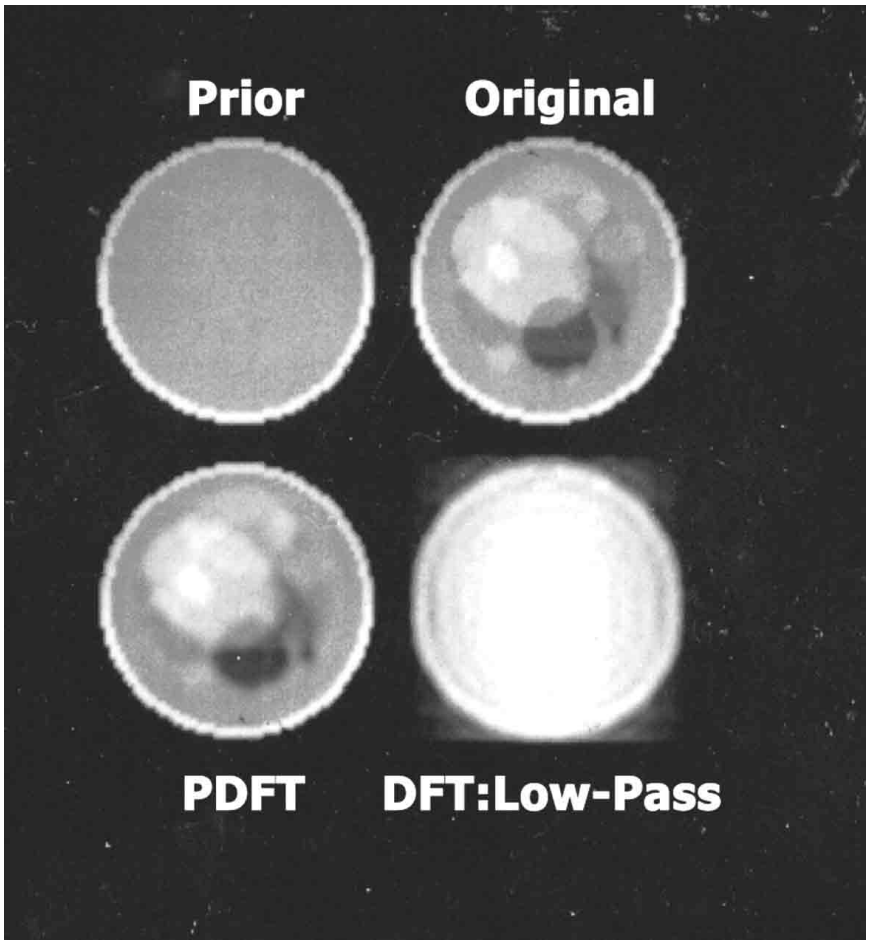

Figure 31.3: The PDFT in image reconstruction.

Chapter 32

Convex Sets

Convex sets and convex functions play important roles in optimization. In this chapter we survey the basic facts concerning the geometry of convex sets.

32.1 A Bit of Topology

Having the norm allows us to define the distance between two points x and y in R^J as $||x - y||$. Being able to talk about how close points are to each other enables us to define continuity of functions on R^J and to consider topological notions of closed set, open set, interior of a set and boundary of a set.

Definition 32.1 *A subset B of R^J is* closed *if, whenever x^k is in B for each non-negative integer k and $||x - x^k|| \to 0$, as $k \to +\infty$, then x is in B.*

For example, $B = [0, 1]$ is closed as a subset of R, but $B = (0, 1)$ is not.

Definition 32.2 *We say that $d \geq 0$ is the* distance from the point x *to the set B if, for every $\epsilon > 0$, there is b_ϵ in B, with $||x - b_\epsilon||_2 < d + \epsilon$, and no b in B with $||x - b||_2 < d$.*

The distance from the point 0 in R to the set $(0, 1)$ is zero, while its distance to the set $(1, 2)$ is one. It follows easily from the definitions that, if B is closed and $d = 0$, then x is in B.

Definition 32.3 *The* closure *of a set B is the set of all points x whose distance from B is zero.*

The closure of the interval $B = (0, 1)$ is $[0, 1]$.

Definition 32.4 *A subset U of R^J is* open *if its complement, the set of all points not in U, is closed.*

Definition 32.5 *Let C be a subset of R^J. A point x in C is said to be an* interior point *of set C if there is $\epsilon > 0$ such that every point z with $||x - z|| < \epsilon$ is in C. The* interior *of the set C, written int(C), is the set of all interior points of C. It is also the largest open set contained within C.*

For example, the open interval $(0, 1)$ is the interior of the intervals $(0, 1]$ and $[0, 1]$. A set C is open if and only if $C = \text{int}(C)$.

Definition 32.6 *A point x in R^J is said to be a* boundary point *of set C if, for every $\epsilon > 0$, there are points y_ϵ in C and z_ϵ not in C, both depending on the choice of ϵ, with $||x - y_\epsilon|| < \epsilon$ and $||x - z_\epsilon|| < \epsilon$. The* boundary *of C is the set of all boundary points of C. It is also the intersection of the closure of C with the closure of its complement.*

For example, the points $x = 0$ and $x = 1$ are boundary points of the set $(0, 1]$.

Definition 32.7 *For $k = 0, 1, 2, ...$, let x^k be a vector in R^J. The sequence of vectors $\{x^k\}$ is said to* converge *to the vector z if, given any $\epsilon > 0$, there is positive integer n, usually depending on ϵ, such that, for every $k > n$, we have $||z - x^k|| \leq \epsilon$. Then we say that z is the* limit of the sequence.

For example, the sequence $\{x^k = \frac{1}{k+1}\}$ in R converges to $z = 0$. The sequence $\{(-1)^k\}$ alternates between 1 and -1, so does not converge. However, the subsequence associated with odd k converges to $z = -1$, while the subsequence associated with even k converges to $z = 1$. The values $z = -1$ and $z = 1$ are called *subsequential limit points*, or, sometimes, *cluster points* of the sequence.

Definition 32.8 *A sequence $\{x^k\}$ of vectors in R^J is said to be* bounded *if there is a constant $b > 0$, such that $||x^k|| \leq b$, for all k.*

A fundamental result in analysis is the following.

Proposition 32.1 *Every convergent sequence of vectors in R^J is bounded. Every bounded sequence of vectors in R^J has at least one convergent subsequence, therefore, has at least one cluster point.*

32.2 Convex Sets in R^J

In preparation for our discussion of linear and nonlinear programming, we consider some of the basic concepts from the geometry of convex sets.

32.2.1 Basic Definitions

We begin with the basic definitions.

Definition 32.9 *A vector z is said to be a* convex combination *of the vectors x and y if there is α in the interval $[0,1]$ such that $z = (1-\alpha)x+\alpha y$.*

Definition 32.10 *A nonempty set C in R^J is said to be* convex *if, for any distinct points x and y in C, and for any real number α in the interval $(0,1)$, the point $(1-\alpha)x+\alpha y$ is also in C; that is, C is closed to convex combinations.*

For example, the unit ball B in R^J, consisting of all x with $||x||_2 \leq 1$, is convex, while the surface of the ball, the set of all x with $||x||_2 = 1$, is not convex.

Definition 32.11 *The* convex hull *of a set S, denoted conv(S), is the smallest convex set containing S.*

Proposition 32.2 *The convex hull of a set S is the set C of all convex combinations of members of S.*

Definition 32.12 *A subset S of R^J is a* subspace *if, for every x and y in S and scalars α and β, the linear combination $\alpha x+\beta y$ is again in S.*

A subspace is necessarily a convex set.

Definition 32.13 *The* orthogonal complement *of a subspace S is the set*

$$S^{\perp} = \{u | u^T s = 0, \text{for every } s \in S\}, \tag{32.1}$$

the set of all vectors u in R^J that are orthogonal to every member of S.

For example, in R^3, the x,y-plane is a subspace and has for its orthogonal complement the z-axis.

Definition 32.14 *A subset M of R^J is a* linear manifold *if there is a subspace S and a vector b such that*

$$M = S + b = \{x | x = s + b, \text{for some } s \text{ in } S\}.$$

Any linear manifold is convex.

Definition 32.15 *For a fixed column vector a with Euclidean length one and a fixed scalar γ the* hyperplane *determined by a and γ is the set*

$$H(a,\gamma) = \{z | \langle a, z\rangle = \gamma\}.$$

The hyperplanes $H(a, \gamma)$ are linear manifolds, and the hyperplanes $H(a, 0)$ are subspaces.

Definition 32.16 *Given a subset C of R^J, the* affine hull *of C, denoted aff(C), is the smallest linear manifold containing C.*

For example, let C be the line segment connecting the two points $(0, 1)$ and $(1, 2)$ in R^2. The affine hull of C is the straight line whose equation is $y = x + 1$.

Definition 32.17 *The* dimension *of a subset of R^J is the dimension of its affine hull, which is the dimension of the subspace of which it is a translate.*

The set C above has dimension one. A set containing only one point is its own affine hull, since it is a translate of the subspace $\{0\}$.

In R^2, the line segment connecting the points $(0, 1)$ and $(1, 2)$ has no interior; it is a one-dimensional subset of a two-dimensional space and can contain no two-dimensional ball. But, the part of this set without its two end points is a sort of interior, called the *relative interior*.

Definition 32.18 *The* relative interior *of a subset C of R^J, denoted ri(C), is the interior of C, as defined by considering C as a subset of its affine hull.*

Since a set consisting of a single point is its own affine hull, it is its own relative interior.

Definition 32.19 *A point x in a convex set C is said to be an* extreme point *of C if the set obtained by removing x from C remains convex.*

Said another way, $x \in C$ is an extreme point of C if x cannot be written as

$$x = (1 - \alpha)y + \alpha z, \tag{32.2}$$

for $y, z \neq x$ and $\alpha \in (0, 1)$. For example, the point $x = 1$ is an extreme point of the convex set $C = [0, 1]$. Every point on the boundary of a sphere in R^J is an extreme point of the sphere. The set of all extreme points of a convex set is denoted Ext(C).

Definition 32.20 *A non-zero vector d is said to be a* direction of unboundedness *of a convex set C if, for all x in C and all $\gamma \geq 0$, the vector $x + \gamma d$ is in C.*

For example, if C is the non-negative orthant in R^J, then any non-negative vector d is a direction of unboundedness.

Definition 32.21 *A vector a is* normal *to a convex set C at the point s in C if*

$$\langle a, c - s\rangle \leq 0, \tag{32.3}$$

for all c in C.

Definition 32.22 *Let C be convex and s in C. The* normal cone *to C at s, denoted $N_C(s)$, is the set of all vectors a that are normal to C at s.*

32.2.2 Orthogonal Projection onto Convex Sets

The following proposition is fundamental in the study of convexity and can be found in most books on the subject; see, for example, the text by Goebel and Reich [134].

Proposition 32.3 *Given any nonempty closed convex set C and an arbitrary vector x in R^J, there is a unique member of C closest to x, denoted $P_C x$, the orthogonal (or metric) projection of x onto C.*

Proof: If x is in C, then $P_C x = x$, so assume that x is not in C. Then $d > 0$, where d is the distance from x to C. For each positive integer n, select c_n in C with $||x - c_n||_2 < d + \frac{1}{n}$, and $||x - c_n||_2 < ||x - c_{n-1}||_2$. Then the sequence $\{c_n\}$ is bounded; let c^* be any cluster point. It follows easily that $||x - c^*||_2 = d$ and that c^* is in C. If there is any other member c of C with $||x - c||_2 = d$, then, by the Parallelogram Law, we would have $||x - (c^* + c)/2||_2 < d$, which is a contradiction. Therefore, c^* is $P_C x$. ∎

For example, if $C = U$, the unit ball, then $P_C x = x/||x||_2$, for all x such that $||x||_2 > 1$, and $P_C x = x$ otherwise. If C is R^J_+, the nonnegative cone of R^J, consisting of all vectors x with $x_j \geq 0$, for each j, then $P_C x = x_+$, the vector whose entries are $\max(x_j, 0)$. For any closed, convex set C, the distance from x to C is $||x - P_C x||$.

If a nonempty set S is not convex, then the orthogonal projection of a vector x onto S need not be well defined; there may be more than one vector in S closest to x. In fact, it is known that a set S is convex if and only if, for every x not in S, there is a unique point in S closest to x. Note that there may well be some x for which there is a unique closest point in S, but if S is not convex, then there must be at least one point without a unique closest point in S.

Lemma 32.1 *For $H = H(a, \gamma)$, $z = P_H x$ is the vector*

$$z = P_H x = x + (\gamma - \langle a, x\rangle)a. \tag{32.4}$$

We shall use this fact in our discussion of the ART algorithm.

For an arbitrary nonempty closed convex set C in R^J, the orthogonal projection $T = P_C$ is a nonlinear operator, unless, of course, C is a subspace. We may not be able to describe $P_C x$ explicitly, but we do know a useful property of $P_C x$.

Proposition 32.4 *For a given x, a vector z in C is $P_C x$ if and only if*

$$\langle c - z, z - x\rangle \geq 0, \tag{32.5}$$

for all c in the set C.

Proof: Let c be arbitrary in C and α in $(0, 1)$. Then

$$||x - P_C x||_2^2 \leq ||x - (1-\alpha)P_C x - \alpha c||_2^2 = ||x - P_C x + \alpha(P_C x - c)||_2^2$$

$$= ||x - P_C x||_2^2 - 2\alpha\langle x - P_C x, c - P_C x\rangle + \alpha^2||P_C x - c||_2^2. \tag{32.6}$$

Therefore,

$$-2\alpha\langle x - P_C x, c - P_C x\rangle + \alpha^2||P_C x - c||_2^2 \geq 0, \tag{32.7}$$

so that

$$2\langle x - P_C x, c - P_C x\rangle \leq \alpha||P_C x - c||_2^2. \tag{32.8}$$

Taking the limit, as $\alpha \to 0$, we conclude that

$$\langle c - P_C x, P_C x - x\rangle \geq 0. \tag{32.9}$$

If z is a member of C that also has the property

$$\langle c - z, z - x\rangle \geq 0, \tag{32.10}$$

for all c in C, then we have both

$$\langle z - P_C x, P_C x - x\rangle \geq 0, \tag{32.11}$$

and

$$\langle z - P_C x, x - z\rangle \geq 0. \tag{32.12}$$

Adding on both sides of these two inequalities lead to

$$\langle z - P_C x, P_C x - z\rangle \geq 0. \tag{32.13}$$

But,

$$\langle z - P_C x, P_C x - z\rangle = -||z - P_C x||_2^2, \tag{32.14}$$

so it must be the case that $z = P_C x$. This completes the proof. ∎

32.3 Some Results on Projections

The characterization of the orthogonal projection operator P_C given by Proposition 32.4 has a number of important consequences.

Corollary 32.1 *Let S be any subspace of R^J. Then, for any x in R^J and s in S, we have*

$$\langle P_S x - x, s \rangle = 0. \tag{32.15}$$

Proof: Since S is a subspace, $s + P_S x$ is again in S, for all s, as is cs, for every scalar c. ∎

This corollary enables us to prove the Decomposition Theorem.

Theorem 32.1 *Let S be any subspace of R^J and x any member of R^J. Then there are unique vectors s in S and u in $S^\perp$ such that $x = s + u$. The vector s is $P_S x$ and the vector u is $P_{S^\perp} x$.*

Proof: For the given x we take $s = P_S x$ and $u = x - P_S x$. Corollary 32.1 assures us that u is in $S^\perp$. Now we need to show that this decomposition is unique. To that end, suppose that we can write $x = s_1 + u_1$, with s_1 in S and u_1 in $S^\perp$. Then Proposition 32.4 tells us that, since $s_1 - x$ is orthogonal to every member of S, s_1 must be $P_S x$. ∎

This theorem is often presented in a slightly different manner.

Theorem 32.2 *Let A be a real I by J matrix. Then every vector b in R^I can be written uniquely as $b = Ax + w$, where $A^T w = 0$.*

To derive Theorem 32.2 from Theorem 32.1, we simply let $S = \{Ax | x \in R^J\}$. Then $S^\perp$ is the set of all w such that $A^T w = 0$. It follows that w is the member of the null space of A^T closest to b.

Here are additional consequences of Proposition 32.4.

Corollary 32.2 *Let S be any subspace of R^J, d a fixed vector, and V the linear manifold $V = S + d = \{v = s + d | s \in S\}$, obtained by translating the members of S by the vector d. Then, for every x in R^J and every v in V, we have*

$$\langle P_V x - x, v - P_V x \rangle = 0. \tag{32.16}$$

Proof: Since v and $P_V x$ are in V, they have the form $v = s + d$, and $P_V x = \hat{s} + d$, for some s and $\hat{s}$ in S. Then $v - P_V x = s - \hat{s}$. ∎

Corollary 32.3 *Let H be the hyperplane $H(a, \gamma)$. Then, for every x, and every h in H, we have*

$$\langle P_H x - x, h - P_H x \rangle = 0. \tag{32.17}$$

Corollary 32.4 *Let S be a subspace of R^J. Then $(S^\perp)^\perp = S$.*

Proof: Every x in R^J has the form $x = s + u$, with s in S and u in $S^\perp$. Suppose x is in $(S^\perp)^\perp$. Then $u = 0$. ∎

Chapter 33

Inner Product Spaces

An *inner product* is a generalization of the dot product between two vectors. An *inner product space* or *pre-Hilbert space* is a vector space on which we have defined an inner product. Such spaces arise in many areas of mathematics and provide a convenient setting for performing optimal approximation.

33.1 Background

We begin by recalling the solution of the vibrating string problem and Sturm-Liouville problems.

33.1.1 The Vibrating String

When we solve the problem of the vibrating string using the technique of separation of variables, the differential equation involving the space variable x, and assuming constant mass density, is

$$y''(x) + \frac{\omega^2}{c^2} y(x) = 0, \tag{33.1}$$

which we can write as an eigenvalue problem

$$y''(x) + \lambda y(x) = 0. \tag{33.2}$$

The solutions to Equation (33.1) are

$$y(x) = \alpha \sin\left(\frac{\omega}{c} x\right).$$

In the vibrating string problem, the string is fixed at both ends, $x = 0$ and $x = L$, so that

$$\phi(0, t) = \phi(L, t) = 0,$$

for all t. Therefore, we must have $y(0) = y(L) = 0$, so that the *eigenfunction solution* that corresponds to the eigenvalue $\lambda_m = \left(\frac{\pi m}{L}\right)^2$ must have the form

$$y(x) = A_m \sin\left(\frac{\omega_m}{c}x\right) = A_m \sin\left(\frac{\pi m}{L}x\right),$$

where $\omega_m = \frac{\pi c m}{L}$, for any positive integer m. Therefore, the boundary conditions limit the choices for the separation constant ω.

We then discover that the eigenfunction solutions corresponding to different λ are *orthogonal*, in the sense that

$$\int_0^L \sin\left(\frac{\pi m}{L}x\right)\sin\left(\frac{\pi n}{L}x\right)dx = 0,$$

for $m \neq n$.

33.1.2 The Sturm-Liouville Problem

The general form for the Sturm-Liouville Problem is

$$\frac{d}{dx}\Big(p(x)y'(x)\Big) + \lambda w(x)y(x) = 0. \tag{33.3}$$

As with the one-dimensional wave equation, boundary conditions, such as $y(a) = y(b) = 0$, where $a = -\infty$ and $b = +\infty$ are allowed, restrict the possible eigenvalues λ to an increasing sequence of positive numbers λ_m. The corresponding eigenfunctions $y_m(x)$ will be $w(x)$-orthogonal, meaning that

$$0 = \int_a^b y_m(x)y_n(x)w(x)dx,$$

for $m \neq n$. For various choices of $w(x)$ and $p(x)$ and various choices of a and b, we obtain several famous sets of "orthogonal" functions.

Well known examples of Sturm-Liouville problems include

- **Legendre:**
$$\frac{d}{dx}\Big((1-x^2)\frac{dy}{dx}\Big) + \lambda y = 0;$$
- **Chebyshev:**
$$\frac{d}{dx}\Big(\sqrt{1-x^2}\frac{dy}{dx}\Big) + \lambda(1-x^2)^{-1/2}y = 0;$$
- **Hermite:**
$$\frac{d}{dx}\Big(e^{-x^2}\frac{dy}{dx}\Big) + \lambda e^{-x^2}y = 0;$$

  and

- **Laguerre:**
$$\frac{d}{dx}\Big(xe^{-x}\frac{dy}{dx}\Big) + \lambda e^{-x}y = 0.$$

Each of these examples involves an inner product space and an orthogonal basis for that space.

33.2 The Complex Vector Dot Product

An *inner product* is a generalization of the notion of the dot product between two complex vectors.

33.2.1 The Two-Dimensional Case

Let $\mathbf{u} = (a, b)$ and $\mathbf{v} = (c, d)$ be two vectors in two-dimensional space. Let $\mathbf{u}$ make the angle $\alpha > 0$ with the positive x-axis and $\mathbf{v}$ the angle $\beta > 0$. Let $||\mathbf{u}|| = \sqrt{a^2 + b^2}$ denote the length of the vector $\mathbf{u}$. Then $a = ||\mathbf{u}|| \cos\alpha$, $b = ||\mathbf{u}|| \sin\alpha$, $c = ||\mathbf{v}|| \cos\beta$ and $d = ||\mathbf{v}|| \sin\beta$. So $\mathbf{u} \cdot \mathbf{v} = ac + bd = ||\mathbf{u}||||\mathbf{v}||(\cos\alpha\cos\beta + \sin\alpha\sin\beta = ||\mathbf{u}||\,||\mathbf{v}||\cos(\alpha - \beta)$. Therefore, we have

$$\mathbf{u} \cdot \mathbf{v} = ||\mathbf{u}||\,||\mathbf{v}||\cos\theta, \tag{33.4}$$

where $\theta = \alpha - \beta$ is the angle between $\mathbf{u}$ and $\mathbf{v}$. Cauchy's inequality is

$$|\mathbf{u} \cdot \mathbf{v}| \leq ||\mathbf{u}||\,||\mathbf{v}||,$$

with equality if and only if $\mathbf{u}$ and $\mathbf{v}$ are parallel. From Equation (33.4) we know that the dot product $\mathbf{u} \cdot \mathbf{v}$ is zero if and only if the angle between these two vectors is a right angle; we say then that $\mathbf{u}$ and $\mathbf{v}$ are mutually *orthogonal*.

Cauchy's inequality extends to complex vectors $\mathbf{u}$ and $\mathbf{v}$:

$$\mathbf{u} \cdot \mathbf{v} = \sum_{n=1}^{N} u_n \overline{v_n}, \tag{33.5}$$

and Cauchy's Inequality still holds.

Proof of Cauchy's Inequality: To prove Cauchy's inequality for the complex vector dot product, we write $\mathbf{u} \cdot \mathbf{v} = |\mathbf{u} \cdot \mathbf{v}|e^{i\theta}$. Let t be a real variable and consider

$$\begin{aligned} 0 \leq ||e^{-i\theta}\mathbf{u} - t\mathbf{v}||^2 &= (e^{-i\theta}\mathbf{u} - t\mathbf{v}) \cdot (e^{-i\theta}\mathbf{u} - t\mathbf{v}) \\ &= ||\mathbf{u}||^2 - t[(e^{-i\theta}\mathbf{u}) \cdot \mathbf{v} + \mathbf{v} \cdot (e^{-i\theta}\mathbf{u})] + t^2||\mathbf{v}||^2 \\ &= ||\mathbf{u}||^2 - t[(e^{-i\theta}\mathbf{u}) \cdot \mathbf{v} + \overline{(e^{-i\theta}\mathbf{u}) \cdot \mathbf{v}}] + t^2||\mathbf{v}||^2 \end{aligned}$$

$$= ||\mathbf{u}||^2 - 2Re(te^{-i\theta}(\mathbf{u}\cdot\mathbf{v})) + t^2||\mathbf{v}||^2$$
$$= ||\mathbf{u}||^2 - 2Re(t|\mathbf{u}\cdot\mathbf{v}|) + t^2||\mathbf{v}||^2 = ||\mathbf{u}||^2 - 2t|\mathbf{u}\cdot\mathbf{v}| + t^2||\mathbf{v}||^2.$$

This is a nonnegative quadratic polynomial in the variable t, so it cannot have two distinct real roots. Therefore, the discriminant $4|\mathbf{u}\cdot\mathbf{v}|^2 - 4||\mathbf{v}||^2||\mathbf{u}||^2$ must be non-positive; that is, $|\mathbf{u}\cdot\mathbf{v}|^2 \leq ||\mathbf{u}||^2||\mathbf{v}||^2$. This is Cauchy's inequality. ∎

A careful examination of the proof just presented shows that we did not explicitly use the definition of the complex vector dot product, but only some of its properties. This suggested to mathematicians the possibility of abstracting these properties and using them to define a more general concept, an *inner product*, between objects more general than complex vectors, such as infinite sequences, random variables, and matrices. Such an inner product can then be used to define the *norm* of these objects and thereby a distance between such objects. Once we have an inner product defined, we also have available the notions of orthogonality and best approximation.

33.2.2 Orthogonality

Consider the problem of writing the two-dimensional real vector $(3, -2)$ as a linear combination of the vectors $(1, 1)$ and $(1, -1)$; that is, we want to find constants a and b so that $(3, -2) = a(1, 1) + b(1, -1)$. One way to do this, of course, is to compare the components: $3 = a + b$ and $-2 = a - b$; we can then solve this simple system for the a and b. In higher dimensions this way of doing it becomes harder, however. A second way is to make use of the dot product and orthogonality.

The dot product of two vectors (x, y) and (w, z) in R^2 is $(x, y)\cdot(w, z) = xw+yz$. If the dot product is zero then the vectors are said to be *orthogonal*; the two vectors $(1, 1)$ and $(1, -1)$ are orthogonal. We take the dot product of both sides of $(3, -2) = a(1, 1) + b(1, -1)$ with $(1, 1)$ to get

$$1 = (3, -2)\cdot(1, 1) = a(1, 1)\cdot(1, 1)+b(1, -1)\cdot(1, 1) = a(1, 1)\cdot(1, 1)+0 = 2a,$$

so we see that $a = \frac{1}{2}$. Similarly, taking the dot product of both sides with $(1, -1)$ gives

$$5 = (3, -2)\cdot(1, -1) = a(1, 1)\cdot(1, -1) + b(1, -1)\cdot(1, -1) = 2b,$$

so $b = \frac{5}{2}$. Therefore, $(3, -2) = \frac{1}{2}(1, 1) + \frac{5}{2}(1, -1)$. The beauty of this approach is that it does not get much harder as we go to higher dimensions.

Since the cosine of the angle θ between vectors $\mathbf{u}$ and $\mathbf{v}$ is

$$\cos\theta = \mathbf{u}\cdot\mathbf{v}/||\mathbf{u}||\,||\mathbf{v}||,$$

where $||\mathbf{u}||^2 = \mathbf{u}\cdot\mathbf{u}$, the projection of vector $\mathbf{v}$ on to the line through the origin parallel to $\mathbf{u}$ is

$$\text{Proj}_{\mathbf{u}}(\mathbf{v}) = \frac{\mathbf{u}\cdot\mathbf{v}}{\mathbf{u}\cdot\mathbf{u}}\mathbf{u}.$$

Therefore, the vector $\mathbf{v}$ can be written as

$$\mathbf{v} = \text{Proj}_{\mathbf{u}}(\mathbf{v}) + (\mathbf{v} - \text{Proj}_{\mathbf{u}}(\mathbf{v})),$$

where the first term on the right is parallel to $\mathbf{u}$ and the second one is orthogonal to $\mathbf{u}$.

How do we find vectors that are mutually orthogonal? Suppose we begin with $(1, 1)$. Take a second vector, say $(1, 2)$, that is not parallel to $(1, 1)$ and write it as we did $\mathbf{v}$ earlier, that is, as a sum of two vectors, one parallel to $(1, 1)$ and the second orthogonal to $(1, 1)$. The projection of $(1, 2)$ onto the line parallel to $(1, 1)$ passing through the origin is

$$\frac{(1,1)\cdot(1,2)}{(1,1)\cdot(1,1)}(1,1) = \frac{3}{2}(1,1) = (\frac{3}{2}, \frac{3}{2})$$

so

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

The vectors $(-\frac{1}{2}, \frac{1}{2}) = -\frac{1}{2}(1, -1)$ and, therefore, $(1, -1)$ are then orthogonal to $(1, 1)$. This approach is the basis for the *Gram-Schmidt* method for constructing a set of mutually orthogonal vectors.

33.3 Generalizing the Dot Product: Inner Products

The proof of Cauchy's Inequality rests not on the actual definition of the complex vector dot product, but rather on four of its most basic properties. We use these properties to extend the concept of the complex vector dot product to that of *inner product*. Later in this chapter we shall give several examples of inner products, applied to a variety of mathematical objects, including infinite sequences, functions, random variables, and matrices. For now, let us denote our mathematical objects by $\mathbf{u}$ and $\mathbf{v}$ and the inner product between them as $\langle \mathbf{u}, \mathbf{v} \rangle$. The objects will then be said to be members of an *inner-product space*. We are interested in inner products because they provide a notion of orthogonality, which is fundamental to best approximation and optimal estimation.

33.3.1 Defining an Inner Product and Norm

The four basic properties that will serve to define an inner product are:

- **1:** $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$, with equality if and only if $\mathbf{u} = \mathbf{0}$;
- **2:** $\langle \mathbf{v}, \mathbf{u} \rangle = \overline{\langle \mathbf{u}, \mathbf{v} \rangle}$;

- **3:** $\langle \mathbf{u}, \mathbf{v} + \mathbf{w} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle + \langle \mathbf{u}, \mathbf{w} \rangle$;
- **4:** $\langle c\mathbf{u}, \mathbf{v} \rangle = c\langle \mathbf{u}, \mathbf{v} \rangle$ for any complex number c.

The inner product is the basic ingredient in Hilbert space theory. Using the inner product, we define the *norm* of $\mathbf{u}$ to be

$$||\mathbf{u}|| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$$

and the distance between $\mathbf{u}$ and $\mathbf{v}$ to be $||\mathbf{u} - \mathbf{v}||$.

The Cauchy-Schwarz Inequality: Because these four properties were all we needed to prove the Cauchy inequality for the complex vector dot product, we obtain the same inequality whenever we have an inner product. This more general inequality is the Cauchy-Schwarz Inequality:

$$|\langle \mathbf{u}, \mathbf{v} \rangle| \leq \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle} \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}$$

or

$$|\langle \mathbf{u}, \mathbf{v} \rangle| \leq ||\mathbf{u}|| \, ||\mathbf{v}||,$$

with equality if and only if there is a scalar c such that $\mathbf{v} = c\mathbf{u}$. We say that the vectors $\mathbf{u}$ and $\mathbf{v}$ are *orthogonal* if $\langle \mathbf{u}, \mathbf{v} \rangle = 0$. We turn now to some examples.

33.3.2 Some Examples of Inner Products

Here are several examples of inner products.

- **Inner product of infinite sequences:** Let $\mathbf{u} = \{u_n\}$ and $\mathbf{v} = \{v_n\}$ be infinite sequences of complex numbers. The inner product is then

  $$\langle \mathbf{u}, \mathbf{v} \rangle = \sum u_n \overline{v_n},$$

  and

  $$||\mathbf{u}|| = \sqrt{\sum |u_n|^2}.$$

  The sums are assumed to be finite; the index of summation n is singly or doubly infinite, depending on the context. The Cauchy-Schwarz inequality says that

  $$|\sum u_n \overline{v_n}| \leq \sqrt{\sum |u_n|^2} \sqrt{\sum |v_n|^2}.$$

- **Inner product of functions:** Now suppose that $\mathbf{u} = f(x)$ and $\mathbf{v} = g(x)$. Then,

$$\langle \mathbf{u}, \mathbf{v} \rangle = \int f(x)\overline{g(x)}dx$$

and

$$||\mathbf{u}|| = \sqrt{\int |f(x)|^2 dx}.$$

The integrals are assumed to be finite; the limits of integration depend on the support of the functions involved. The Cauchy-Schwarz inequality now says that

$$|\int f(x)\overline{g(x)}dx| \leq \sqrt{\int |f(x)|^2 dx}\sqrt{\int |g(x)|^2 dx}.$$

- **Inner product of random variables:** Now suppose that $\mathbf{u} = X$ and $\mathbf{v} = Y$ are random variables. Then,

$$\langle \mathbf{u}, \mathbf{v} \rangle = E(X\overline{Y})$$

and

$$||\mathbf{u}|| = \sqrt{E(|X|^2)},$$

which is the standard deviation of X if the mean of X is zero. The expected values are assumed to be finite. The Cauchy-Schwarz inequality now says that

$$|E(X\overline{Y})| \leq \sqrt{E(|X|^2)}\sqrt{E(|Y|^2)}.$$

If $E(X) = 0$ and $E(Y) = 0$, the random variables X and Y are orthogonal if and only if they are *uncorrelated*.

- **Inner product of complex matrices:** Now suppose that $\mathbf{u} = A$ and $\mathbf{v} = B$ are complex matrices. Then,

$$\langle \mathbf{u}, \mathbf{v} \rangle = \text{trace}(B^\dagger A)$$

and

$$||\mathbf{u}|| = \sqrt{\text{trace}(A^\dagger A)},$$

where the trace of a square matrix is the sum of the entries on the main diagonal. As we shall see later, this inner product is simply the complex vector dot product of the vectorized versions of the matrices involved. The Cauchy-Schwarz inequality now says that

$$|\text{trace}(B^\dagger A)| \leq \sqrt{\text{trace}(A^\dagger A)}\sqrt{\text{trace}(B^\dagger B)}.$$

- **Weighted inner product of complex vectors:** Let $\mathbf{u}$ and $\mathbf{v}$ be complex vectors and let Q be a Hermitian positive-definite matrix; that is, $Q^\dagger = Q$ and $\mathbf{u}^\dagger Q\mathbf{u} > 0$ for all nonzero vectors $\mathbf{u}$. The inner product is then
$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{v}^\dagger Q\mathbf{u}$$
and
$$||\mathbf{u}|| = \sqrt{\mathbf{u}^\dagger Q\mathbf{u}}.$$
We know from the eigenvector decomposition of Q that $Q = C^\dagger C$ for some matrix C. Therefore, the inner product is simply the complex vector dot product of the vectors $C\mathbf{u}$ and $C\mathbf{v}$. The Cauchy-Schwarz inequality says that
$$|\mathbf{v}^\dagger Q\mathbf{u}| \leq \sqrt{\mathbf{u}^\dagger Q\mathbf{u}}\sqrt{\mathbf{v}^\dagger Q\mathbf{v}}.$$

- **Weighted inner product of functions:** Now suppose that $\mathbf{u} = f(x)$ and $\mathbf{v} = g(x)$ and $w(x) > 0$. Then define
$$\langle \mathbf{u}, \mathbf{v} \rangle = \int f(x)\overline{g(x)}w(x)dx$$
and
$$||\mathbf{u}|| = \sqrt{\int |f(x)|^2 w(x)dx}.$$
The integrals are assumed to be finite; the limits of integration depend on the support of the functions involved. This inner product is simply the inner product of the functions $f(x)\sqrt{w(x)}$ and $g(x)\sqrt{w(x)}$. The Cauchy-Schwarz inequality now says that
$$|\int f(x)\overline{g(x)}w(x)dx| \leq \sqrt{\int |f(x)|^2 w(x)dx}\sqrt{\int |g(x)|^2 w(x)dx}.$$

Once we have an inner product defined, we can speak about orthogonality and best approximation. Important in that regard is the orthogonality principle.

33.4 Best Approximation and the Orthogonality Principle

Imagine that you are standing and looking down at the floor. The point B on the floor that is closest to N, the tip of your nose, is the unique

point on the floor such that the vector from B to any other point A on the floor is perpendicular to the vector from N to B; that is, $\langle BN, BA\rangle = 0$. This is a simple illustration of the *orthogonality principle*. Whenever we have an inner product defined we can speak of orthogonality and apply the orthogonality principle to find best approximations. For notational simplicity, we shall consider only real inner product spaces.

33.4.1 Best Approximation

Let $\mathbf{u}$ and $\mathbf{v}^1, ..., \mathbf{v}^N$ be members of a real inner-product space. For all choices of scalars $a_1, ..., a_N$, we can compute the distance from $\mathbf{u}$ to the member $a_1\mathbf{v}^1 + ...a_N\mathbf{v}^N$. Then, we minimize this distance over all choices of the scalars; let $b_1, ..., b_N$ be this best choice.

The distance squared from $\mathbf{u}$ to $a_1\mathbf{v}^1 + ...a_N\mathbf{v}^N$ is

$$||\mathbf{u} - (a_1\mathbf{v}^1 + ...a_N\mathbf{v}^N)||^2 = \langle \mathbf{u} - (a_1\mathbf{v}^1 + ...a_N\mathbf{v}^N), \mathbf{u} - (a_1\mathbf{v}^1 + ...a_N\mathbf{v}^N)\rangle,$$

$$= ||\mathbf{u}||^2 - 2\langle \mathbf{u}, \sum_{n=1}^{N} a_n\mathbf{v}^n\rangle + \sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m \langle \mathbf{v}^n\mathbf{v}^m\rangle.$$

Setting the partial derivative with respect to a_n equal to zero, we have

$$\langle \mathbf{u}, \mathbf{v}^n\rangle = \sum_{m=1}^{N} a_m \langle \mathbf{v}^m\mathbf{v}^n\rangle.$$

With $\mathbf{a} = (a_1, ..., a_N)^T$,

$$\mathbf{d} = (\langle \mathbf{u}, \mathbf{v}^1\rangle, ..., \langle \mathbf{u}, \mathbf{v}^N\rangle)^T$$

and V the matrix with entries

$$V_{mn} = \langle \mathbf{v}^m, \mathbf{v}^n\rangle,$$

we find that we must solve the system of equations $V\mathbf{a} = \mathbf{d}$. When the vectors $\mathbf{v}^n$ are mutually orthogonal and each has norm equal to one, then $V = I$, the identity matrix, and the desired vector $\mathbf{a}$ is simply $\mathbf{d}$.

33.4.2 The Orthogonality Principle

The *orthogonality principle* provides another way to view the calculation of the best approximation: let the best approximation of $\mathbf{u}$ be the vector

$$\hat{\mathbf{v}} = b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N.$$

Then

$$\langle \mathbf{u} - \hat{\mathbf{v}}, \mathbf{v}^n\rangle = \langle \mathbf{u} - (b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N), \mathbf{v}^n\rangle = 0,$$

for $n = 1, 2, ..., N$. This leads directly to the system of equations

$$\mathbf{d} = V\mathbf{b},$$

which, as we just saw, provides the optimal coefficients.

To see why the orthogonality principle is valid, fix a value of n and consider the problem of minimizing the distance

$$||\mathbf{u} - (b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N + \alpha\mathbf{v}^n)||$$

as a function of α. Writing the norm squared in terms of the inner product, expanding the terms, and differentiating with respect to α, we find that the minimum occurs when

$$\alpha = \langle \mathbf{u} - b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N, \mathbf{v}^n \rangle.$$

But we already know that the minimum occurs when $\alpha = 0$. This completes the proof of the orthogonality principle.

33.5 Gram-Schmidt Orthogonalization

We have seen that the best approximation is easily calculated if the vectors $\mathbf{v}^n$ are mutually orthogonal. But how do we get such a mutually orthogonal set, in general? The Gram-Schmidt Orthogonalization Method is one way to proceed.

Let $\{\mathbf{v}^1, ..., \mathbf{v}^N\}$ be a linearly independent set of vectors in the space R^M, where $N \leq M$. The Gram-Schmidt method uses the $\mathbf{v}^n$ to create an orthogonal basis $\{\mathbf{u}^1, ..., \mathbf{u}^N\}$ for the span of the $\mathbf{v}^n$. Begin by taking $\mathbf{u}^1 = \mathbf{v}^1$. For $j = 2, ..., N$, let

$$\mathbf{u}^j = \mathbf{v}^j - \frac{\mathbf{u}^1 \cdot \mathbf{v}^j}{\mathbf{u}^1 \cdot \mathbf{u}^1}\mathbf{u}^1 - ... - \frac{\mathbf{u}^{j-1} \cdot \mathbf{v}^j}{\mathbf{u}^{j-1} \cdot \mathbf{u}^{j-1}}\mathbf{u}^{j-1}. \tag{33.6}$$

One obvious problem with this approach is that the calculations become increasingly complicated and lengthy as the j increases. In many of the important examples of orthogonal functions that we study in connection with Sturm-Liouville problems, there is a two-term recursive formula that enables us to generate the next orthogonal function from the two previous ones.

Chapter 34

Reconstruction from Limited Data

The problem is to reconstruct a (possibly complex-valued) function $f : R^D \rightarrow C$ from finitely many measurements g_n, $n = 1, ..., N$, pertaining to f. The function $f(r)$ represents the physical object of interest, such as the spatial distribution of acoustic energy in sonar, the distribution of x-ray-attenuating material in transmission tomography, the distribution of radionuclide in emission tomography, the sources of reflected radio waves in radar, and so on. Often the reconstruction, or estimate, of the function f takes the form of an image in two or three dimensions; for that reason, we also speak of the problem as one of *image reconstruction*. The data are obtained through measurements. Because there are only finitely many measurements, the problem is highly under-determined and even noise-free data are insufficient to specify a unique solution.

34.1 The Optimization Approach

One way to solve such under-determined problems is to replace $f(r)$ with a vector in C^N and to use the data to determine the N entries of this vector. An alternative method is to model $f(r)$ as a member of a family of linear combinations of N preselected basis functions of the multivariable r. Then the data is used to determine the coefficients. This approach offers the user the opportunity to incorporate prior information about $f(r)$ in the choice of the basis functions. Such finite-parameter models for $f(r)$ can be obtained through the use of the minimum-norm estimation procedure, as we shall see. More generally, we can associate a *cost* with each data-consistent function of r, and then minimize the cost over all the potential solutions to the problem. Using a norm as a cost function is one way to

proceed, but there are others. These optimization problems can often be solved only through the use of discretization and iterative algorithms.

34.2 Introduction to Hilbert Space

In many applications the data are related linearly to f. To model the operator that transforms f into the data vector, we need to select an ambient space containing f. Typically, we choose a Hilbert space. The selection of the inner product provides an opportunity to incorporate prior knowledge about f into the reconstruction. The inner product induces a norm and our reconstruction is that function, consistent with the data, for which this norm is minimized. We shall illustrate the method using Fourier-transform data and prior knowledge about the support of f and about its overall shape.

Our problem, then, is to estimate a (possibly complex-valued) function $f(r)$ of D real variables $r = (r_1, ..., r_D)$ from finitely many measurements, g_n, $n = 1, ..., N$. We shall assume, in this chapter, that these measurements take the form

$$g_n = \int_S f(r)\overline{h_n(r)}dr, \tag{34.1}$$

where S denotes the support of the function $f(r)$, which, in most cases, is a bounded set. For the purpose of estimating, or reconstructing, $f(r)$, it is convenient to view Equation (34.1) in the context of a Hilbert space, and to write

$$g_n = \langle f, h_n \rangle, \tag{34.2}$$

where the usual Hilbert space inner product is defined by

$$\langle f, h \rangle_2 = \int_S f(r)\overline{h(r)}dr, \tag{34.3}$$

for functions $f(r)$ and $h(r)$ supported on the set S. Of course, for these integrals to be defined, the functions must satisfy certain additional properties, but a more complete discussion of these issues is outside the scope of this chapter. The Hilbert space so defined, denoted $L^2(S)$, consists (essentially) of all functions $f(r)$ for which the norm

$$||f||_2 = \sqrt{\int_S |f(r)|^2 dr} \tag{34.4}$$

is finite.

34.2.1 Minimum-Norm Solutions

Our estimation problem is highly under-determined; there are infinitely many functions in $L^2(S)$ that are consistent with the data and might be the right answer. Such under-determined problems are often solved by acting conservatively, and selecting as the estimate that function consistent with the data that has the smallest norm. At the same time, however, we often have some prior information about f that we would like to incorporate in the estimate. One way to achieve both of these goals is to select the norm to incorporate prior information about f, and then to take as the estimate of f the function consistent with the data, for which the chosen norm is minimized.

The data vector $g = (g_1, ..., g_N)^T$ is in C^N and the linear operator $\mathcal{H}$ from $L^2(S)$ to C^N takes f to g; so we write $g = \mathcal{H}f$. Associated with the mapping $\mathcal{H}$ is its adjoint operator, $\mathcal{H}^\dagger$, going from C^N to $L^2(S)$ and given, for each vector $a = (a_1, ..., a_N)^T$, by

$$\mathcal{H}^\dagger a(r) = a_1 h_1(r) + ... + a_N h_N(r). \tag{34.5}$$

The operator from C^N to C^N defined by $\mathcal{H}\mathcal{H}^\dagger$ corresponds to an N by N matrix, which we shall also denote by $\mathcal{H}\mathcal{H}^\dagger$. If the functions $h_n(r)$ are linearly independent, then this matrix is positive-definite, therefore invertible.

Given the data vector g, we can solve the system of linear equations

$$g = \mathcal{H}\mathcal{H}^\dagger a \tag{34.6}$$

for the vector a. Then the function

$$\hat{f}(r) = \mathcal{H}^\dagger a(r) \tag{34.7}$$

is consistent with the measured data and is the function in $L^2(S)$ with the smallest norm for which this is true. The function $w(r) = f(r) - \hat{f}(r)$ has the property $\mathcal{H}w = 0$. It is easy to see that

$$||f||_2^2 = ||\hat{f}||_2^2 + ||w||_2^2 \tag{34.8}$$

The estimate $\hat{f}(r)$ is the *minimum-norm solution*, with respect to the norm defined in Equation (34.4). If we change the norm on $L^2(S)$, or, equivalently, the inner product, then the minimum-norm solution will change.

For any continuous linear operator $\mathcal{T}$ on $L^2(S)$, the adjoint operator, denoted $\mathcal{T}^\dagger$, is defined by

$$\langle \mathcal{T}f, h\rangle_2 = \langle f, \mathcal{T}^\dagger h\rangle_2. \tag{34.9}$$

The adjoint operator will change when we change the inner product.

34.3 A Class of Inner Products

Let $\mathcal{T}$ be a continuous, linear, and invertible operator on $L^2(S)$. Define the $\mathcal{T}$ inner product to be

$$\langle f, h\rangle_{\mathcal{T}} = \langle \mathcal{T}^{-1} f, \mathcal{T}^{-1} h\rangle_2. \tag{34.10}$$

We can then use this inner product to define the problem to be solved. We now say that

$$g_n = \langle f, t^n\rangle_{\mathcal{T}}, \tag{34.11}$$

for known functions $t^n(r)$. Using the definition of the $\mathcal{T}$ inner product, we find that

$$g_n = \langle f, h^n\rangle_2 = \langle \mathcal{T} f, \mathcal{T} h^n\rangle_{\mathcal{T}}. \tag{34.12}$$

The adjoint operator for $\mathcal{T}$, with respect to the $\mathcal{T}$-norm, is denoted $\mathcal{T}^*$, and is defined by

$$\langle \mathcal{T} f, h\rangle_{\mathcal{T}} = \langle f, \mathcal{T}^* h\rangle_{\mathcal{T}}. \tag{34.13}$$

Therefore,

$$g_n = \langle f, \mathcal{T}^* \mathcal{T} h^n\rangle_{\mathcal{T}}. \tag{34.14}$$

Lemma 34.1 . *We have* $\mathcal{T}^* \mathcal{T} = \mathcal{T}\mathcal{T}^\dagger$.

Consequently, we have

$$g_n = \langle f, \mathcal{T}\mathcal{T}^\dagger h^n\rangle_{\mathcal{T}}. \tag{34.15}$$

34.4 Minimum-$\mathcal{T}$-Norm Solutions

The function $\tilde{f}$ consistent with the data and having the smallest $\mathcal{T}$-norm has the algebraic form

$$\hat{f} = \sum_{m=1}^{N} a_m \mathcal{T}\mathcal{T}^\dagger h^m. \tag{34.16}$$

Applying the $\mathcal{T}$-inner product to both sides of Equation (34.16), we get

$$g_n = \langle \hat{f}, \mathcal{T}\mathcal{T}^\dagger h^n\rangle_{\mathcal{T}} \tag{34.17}$$

$$= \sum_{m=1}^{N} a_m \langle \mathcal{T}\mathcal{T}^\dagger h^m, \mathcal{T}\mathcal{T}^\dagger h^n\rangle_{\mathcal{T}}. \tag{34.18}$$

Therefore,

$$g_n = \sum_{m=1}^{N} a_m \langle \mathcal{T}^\dagger h^m, \mathcal{T}^\dagger h^n \rangle_2. \tag{34.19}$$

We solve this system for the a_m and insert them into Equation (34.16) to get our reconstruction. The Gram matrix that appears in Equation (34.19) is positive-definite, but is often ill-conditioned; increasing the main diagonal by a percent or so usually is sufficient regularization.

34.5 The Case of Fourier-Transform Data

To illustrate these minimum-$\mathcal{T}$-norm solutions, we consider the case in which the data are values of the Fourier transform of f. Specifically, suppose that

$$g_n = \int_S f(x) e^{-i\omega_n x} dx, \tag{34.20}$$

for arbitrary values ω_n.

34.5.1 The $L^2(-\pi, \pi)$ Case

Assume that $f(x) = 0$, for $|x| > \pi$. The minimum-2-norm solution has the form

$$\hat{f}(x) = \sum_{m=1}^{N} a_m e^{i\omega_m x}, \tag{34.21}$$

with

$$g_n = \sum_{m=1}^{N} a_m \int_{-\pi}^{\pi} e^{i(\omega_m - \omega_n)x} dx. \tag{34.22}$$

For the equi-spaced values $\omega_n = n$ we find that $a_m = g_m$ and the minimum-norm solution is

$$\hat{f}(x) = \sum_{n=1}^{N} g_n e^{inx}. \tag{34.23}$$

34.5.2 The Over-Sampled Case

Suppose that $f(x) = 0$ for $|x| > A$, where $0 < A < \pi$. Then we use $L^2(-A, A)$ as the Hilbert space. For equi-spaced data at $\omega_n = n$, we have

$$g_n = \int_{-\pi}^{\pi} f(x) \chi_A(x) e^{-inx} dx, \tag{34.24}$$

so that the minimum-norm solution has the form

$$\hat{f}(x) = \chi_A(x) \sum_{m=1}^{N} a_m e^{imx}, \tag{34.25}$$

with

$$g_n = 2 \sum_{m=1}^{N} a_m \frac{\sin A(m-n)}{m-n}. \tag{34.26}$$

The minimum-norm solution is support-limited to $[-A, A]$ and consistent with the Fourier-transform data.

34.5.3 Using a Prior Estimate of f

Suppose that $f(x) = 0$ for $|x| > \pi$ again, and that $p(x)$ satisfies

$$0 < \epsilon \leq p(x) \leq E < +\infty, \tag{34.27}$$

for all x in $[-\pi, \pi]$. Define the operator $\mathcal{T}$ by $(\mathcal{T}f)(x) = \sqrt{p(x)}f(x)$. The $\mathcal{T}$-norm is then

$$\langle f, h \rangle_{\mathcal{T}} = \int_{-\pi}^{\pi} f(x)\overline{h(x)}p(x)^{-1}dx. \tag{34.28}$$

It follows that

$$g_n = \int_{-\pi}^{\pi} f(x)p(x)e^{-i\omega_n x}p(x)^{-1}dx, \tag{34.29}$$

so that the minimum $\mathcal{T}$-norm solution is

$$\hat{f}(x) = \sum_{m=1}^{N} a_m p(x) e^{i\omega_m x} = p(x) \sum_{m=1}^{N} a_m e^{i\omega_m x}, \tag{34.30}$$

where

$$g_n = \sum_{m=1}^{N} a_m \int_{-\pi}^{\pi} p(x) e^{i(\omega_m - \omega_n)x} dx. \tag{34.31}$$

If we have prior knowledge about the support of f, or some idea of its shape, we can incorporate that prior knowledge into the reconstruction through the choice of $p(x)$.

The reconstruction in Equation (34.30) was presented in [39], where it was called the PDFT method. The PDFT was based on a non-iterative version of the Gerchberg-Papoulis bandlimited extrapolation procedure,

discussed earlier in [38]. The PDFT was then applied to image reconstruction problems in [40]. An application of the PDFT was presented in [43]. In [42] we extended the PDFT to a nonlinear version, the indirect PDFT (IPDFT), that generalizes Burg's maximum entropy spectrum estimation method. The PDFT was applied to the phase problem in [45] and in [46] both the PDFT and IPDFT were examined in the context of Wiener filter approximation. More recent work on these topics is discussed in the book [62].

When N, the number of data values, is not large, the PDFT can be implemented in a straight-forward manner, by first calculating the matrix P that appears in Equation (34.31), with entries

$$P_{n,m} = \int_{-\pi}^{\pi} p(x) e^{i(\omega_m - \omega_n)x} dx,$$

solving Equation (34.31) for the coefficients a_m, and finally, inserting these coefficients in Equation (34.30). When N is large, calculating the entries of the matrix P can be an expensive step. Since, in such cases, solving the system in Equation (34.31) will probably be done iteratively, it makes sense to consider an iterative alternative to the PDFT that avoids the use of the matrix P. This is the *discrete* PDFT (DPDFT).

The Discrete PDFT (DPDFT)

The PDFT uses the estimate $\hat{f}(x)$ of $f(x)$, consistent with the data, that has the minimum weighted norm

$$\int_{-\pi}^{\pi} |\hat{f}(x)|^2 p(x)^{-1} dx.$$

The discrete PDFT (DPDFT) replaces the functions $f(x)$ and $p(x)$ with finite vectors $f = (f_1, ..., f_J)^T$ and $p = (p_1, ..., p_J)^T$, for some $J > N$; for example, we could have $f_j = f(x_j)$ for some sample points x_j in $(-\pi, \pi)$. The vector p must have positive entries. The integrals that appear in Equation (34.20) are replaced by sums

$$g_n = \sum_{j=1}^{J} f_j E_{n,j}; \tag{34.32}$$

for example, we could use $E_{n,j} = \exp(-i\omega_n x_j)$. Now our estimate is the solution of the system $g = Ef$ for which the weighted norm

$$\sum_{j=1}^{J} |f_j|^2 p_j^{-1}$$

is minimized. To obtain this minimum-weighted-norm solution, we can use the ART algorithm.

The ART will give the minimum-norm solution of $Au = v$ if we begin the iteration at $u^0 = 0$. To obtain the solution with minimum weighted norm

$$\sum_{j=1}^{J} |u_j|^2 p_j^{-1},$$

we replace u_j with $u_j p_j^{-1/2}$, and $A_{n,j}$ with $A_{n,j} p_j^{1/2}$, and then apply the ART.

Chapter 35

Compressed Sensing

One area that has attracted much attention lately is *compressed sensing* or *compressed sampling* (CS) [107]. For applications such as medical imaging, CS may provide a means of reducing radiation dosage to the patient without sacrificing image quality. An important aspect of CS is finding sparse solutions of under-determined systems of linear equations, which can often be accomplished by one-norm minimization. Perhaps the best reference to date on CS is [32].

35.1 Compressed Sensing

The objective in CS is exploit sparseness to reconstruct a vector f in R^J from relatively few linear functional measurements [107].

Let $U = \{u^1, u^2, ..., u^J\}$ and $V = \{v^1, v^2, ..., v^J\}$ be two orthonormal bases for R^J, with all members of R^J represented as column vectors. For $i = 1, 2, ..., J$, let

$$\mu_i = \max_{1 \leq j \leq J} \{|\langle u^i, v^j \rangle|\}$$

and

$$\mu(U, V) = \max_{i=1}^{I} \mu_i.$$

We know from Cauchy's Inequality that

$$|\langle u^i, v^j \rangle| \leq 1,$$

and from Parseval's Equation

$$\sum_{j=1}^{J} |\langle u^i, v^j \rangle|^2 = ||u^i||^2 = 1.$$

Therefore, we have

$$\frac{1}{\sqrt{J}} \leq \mu(U,V) \leq 1.$$

The quantity $\mu(U,V)$ is the *coherence* measure of the two bases; the closer $\mu(U,V)$ is to the lower bound of $\frac{1}{\sqrt{J}}$, the more *incoherent* the two bases are.

Let f be a fixed member of R^J; we expand f in the V basis as

$$f = x_1 v^1 + x_2 v^2 + ... + x_J v^J.$$

We say that the coefficient vector $x = (x_1, ..., x_J)$ is S-sparse if S is the number of non-zero x_j.

If S is small, most of the x_j are zero, but since we do not know which ones these are, we would have to compute all the linear functional values

$$x_j = \langle f, v^j \rangle$$

to recover f exactly. In fact, the smaller S is, the harder it would be to learn anything from randomly selected x_j, since most would be zero. The idea in CS is to obtain measurements of f with members of a different orthonormal basis, which we call the U basis. If the members of U are very much like the members of V, then nothing is gained. But, if the members of U are quite unlike the members of V, then each inner product measurement

$$y_i = \langle f, u^i \rangle = f^T u^i$$

should tell us something about f. If the two bases are sufficiently incoherent, then relatively few y_i values should tell us quite a bit about f. Specifically, we have the following result due to Candès and Romberg [67]: suppose the coefficient vector x for representing f in the V basis is S-sparse. Select uniformly randomly $M \leq J$ members of the U basis and compute the measurements $y_i = \langle f, u^i \rangle$. Then, if M is sufficiently large, it is highly probable that $z = x$ also solves the problem of minimizing the one-norm

$$||z||_1 = |z_1| + |z_2| + ... + |z_J|,$$

subject to the conditions

$$y_i = \langle g, a^i \rangle = g^T u^i,$$

for those M randomly selected u^i, where

$$g = z_1 v^1 + z_2 v^2 + ... + z_J v^J.$$

The smaller $\mu(U,V)$ is, the smaller the M is permitted to be without reducing the probability of perfect reconstruction.

35.2 Sparse Solutions

Suppose that A is a real M by N matrix, with $M < N$, and that the linear system $Ax = b$ has infinitely many 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 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.

35.2.1 Maximally Sparse Solutions

Consider the problem P_0: among all solutions x of the consistent system $b = Ax$, find one, call it $\hat{x}$, 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.

35.2.2 Minimum One-Norm Solutions

Instead, we can seek a *minimum one-norm* solution, that is, solve the problem P_1: minimize

$$||x||_1 = \sum_{n=1}^{N} |x_n|,$$

subject to $Ax = b$. Problem P_1 can be formulated as a linear programming problem, so is more easily solved. The big questions are: when does P_1 have a unique solution, and when is it $\hat{x}$? The problem P_1 will have a unique solution if and only if A is such that the one-norm satisfies

$$||\hat{x}||_1 < ||\hat{x} + v||_1,$$

for all non-zero v in the null space of A.

35.2.3 Minimizing $\|x\|_1$ as Linear Programming

The entries of x need not be non-negative, so the problem is not yet a linear programming problem. Let

$$B = [\, A \quad -A \,],$$

and consider the linear programming problem of minimizing the function

$$c^T z = \sum_{j=1}^{2J} z_j,$$

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 $u_j^* v_j^* = 0$, for each j. If, say, there is a j such that $0 < v_j < u_j$, then we can create a new vector z by replacing the old u_j^* with $u_j^* - v_j^*$ and the old v_j^* with zero, while maintaining $Bz = b$. But then, since $u_j^* - v_j^* < u_j^* + v_j^*$, 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 $u_j = x_j$, if $x_j \geq 0$, and $u_j = 0$, otherwise. Let $v_j = u_j - x_j$, so that $x = u - v$. Then let

$$z = \begin{bmatrix} u \\ v \end{bmatrix}.$$

Then $b = Ax = Bz$, and $c^T z = \|x\|_1$. Consequently,

$$\|x^*\|_1 = c^T z^* \leq c^T z = \|x\|_1,$$

and x^* must be a minimum one-norm solution.

35.2.4 Why the One-Norm?

When a system of linear equations $Ax = b$ is under-determined, we can find the *minimum-two-norm solution* that minimizes the square of the two-norm,

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

subject to $Ax = b$. 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 solution for which the one-norm,

$$||x||_1 = \sum_{n=1}^{N} |x_n|,$$

is minimized. The one-norm still penalizes relatively large entries x_n more than the smaller ones, but much less than the two-norm does. As a result, it often happens that the minimum one-norm solution actually solves P_0 as well.

35.2.5 Comparison with the PDFT

The PDFT approach to solving the under-determined system $Ax = b$ is to select weights $w_n > 0$ and then to find the solution $\tilde{x}$ that minimizes the weighted two-norm given by

$$\sum_{n=1}^{N} |x_n|^2 w_n.$$

Our intention is to select weights w_n so that w_n^{-1} is reasonably close to $|\hat{x}_n|$; consider, therefore, what happens when $w_n^{-1} = |\hat{x}_n|$. 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

$$\sum_{n=1}^{N} |x_n| = \sum_{n=1}^{N} \frac{|x_n|}{\sqrt{|\hat{x}_n|}} \sqrt{|\hat{x}_n|}$$

$$\leq \sqrt{\sum_{n=1}^{N} \frac{|x_n|^2}{|\hat{x}_n|}} \sqrt{\sum_{n=1}^{N} |\hat{x}_n|}.$$

Therefore,

$$\sum_{n=1}^{N} |\tilde{x}_n| \leq \sqrt{\sum_{n=1}^{N} \frac{|\tilde{x}_n|^2}{|\hat{x}_n|}} \sqrt{\sum_{n=1}^{N} |\hat{x}_n|}$$

$$\leq \sqrt{\sum_{n=1}^{N} \frac{|\hat{x}_n|^2}{|\hat{x}_n|}} \sqrt{\sum_{n=1}^{N} |\hat{x}_n|} = \sum_{n=1}^{N} |\hat{x}_n|.$$

Therefore, $\tilde{x}$ also minimizes the one-norm.

35.2.6 Iterative Reweighting

We want each weight w_n to be a good prior estimate of the reciprocal of $|\hat{x}_n|$. Because we do not yet know $\hat{x}$, we may take a sequential-optimization approach, beginning with weights $w_n^0 > 0$, finding the PDFT solution using these weights, then using this PDFT solution to get a (we hope!) a 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 [123].

In [69], 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 $|\hat{x}_n|$. Again, not yet knowing $\hat{x}$, 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 PDFT has been the nature of the prior weighting. Does w_n approximate $|x_n|$ or $|x_n|^2$? This is close to the issue treated in [69], 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 PDFT, 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 $\hat{x}$, their use reduces the penalty associated with the larger entries of an estimated solution.

35.3 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.

35.3.1 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 [68]:

Theorem 35.1 *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 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.

35.3.2 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.

35.3.3 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.

We turn now to the more general problem of compressed sampling.

35.4 Compressed Sampling

Our goal is to recover the vector $f = (f_1, ..., f_N)^T$ from M linear functional values of f, where M is much less than N. In general, this is not possible

without prior information about the vector f. In compressed sampling, the prior information concerns the sparseness of either f itself, or another vector linearly related to f.

Let U and V be unitary N by N matrices, so that the column vectors of both U and V form orthonormal bases for C^N. We shall refer to the bases associated with U and V as the *sampling basis* and the *representing basis*, respectively. The first objective is to find a unitary matrix V so that $f = Vx$, where x is sparse. Then we want to find a second unitary matrix U such that, when an M by N matrix R is obtained from U by deleting rows, the sparse vector x can be determined from the data $b = RVx = Ax$. Theorems in compressed sensing describe properties of the matrices U and V such that, when R is obtained from U by a random selection of the rows of U, the vector x will be uniquely determined, with high probability, as the unique solution that minimizes the one-norm.

Chapter 36

The BLUE and The Kalman Filter

In most signal- and image-processing applications the measured data includes (or may include) a signal component we want and unwanted components called *noise*. Estimation involves determining the precise nature and strength of the signal component; deciding if that strength is zero or not is detection.

Noise often appears as an additive term, which we then try to remove. If we knew precisely the noisy part added to each data value we would simply subtract it; of course, we never have such information. How then do we remove something when we don't know what it is? Statistics provides a way out.

The basic idea in statistics is to use procedures that perform well on average, when applied to a class of problems. The procedures are built using properties of that class, usually involving probabilistic notions, and are evaluated by examining how they would have performed had they been applied to every problem in the class. To use such methods to remove additive noise, we need a description of the class of noises we expect to encounter, not specific values of the noise component in any one particular instance. We also need some idea about what signal components look like. In this chapter we discuss solving this noise removal problem using the *best linear unbiased estimation* (BLUE). We begin with the simplest case and then proceed to discuss increasingly complex scenarios.

An important application of the BLUE is in Kalman filtering. The connection between the BLUE and Kalman filtering is best understood by considering the case of the BLUE with a prior estimate of the signal component, and mastering the various matrix manipulations that are involved in this problem. These calculations then carry over, almost unchanged, to

the Kalman filtering.

Kalman filtering is usually presented in the context of estimating a sequence of vectors evolving in time. Kalman filtering for image processing is derived by analogy with the temporal case, with certain parts of the image considered to be in the "past" of a fixed pixel.

36.1 The Simplest Case

Suppose our data is $z_j = c + v_j$, for $j = 1, ..., J$, where c is an unknown constant to be estimated and the v_j are additive noise. We assume that $E(v_j) = 0, E(v_j\overline{v_k}) = 0$ for $j \neq k$, and $E(|v_j|^2) = \sigma_j^2$. So, the additive noises are assumed to have mean zero and to be independent (or at least uncorrelated). In order to estimate c, we adopt the following rules:

1. The estimate $\hat{c}$ is *linear* in the data $\mathbf{z} = (z_1, ..., z_J)^T$; that is, $\hat{c} = \mathbf{k}^\dagger \mathbf{z}$, for some vector $\mathbf{k} = (k_1, ..., k_J)^T$.

2. The estimate is *unbiased*; that is $E(\hat{c}) = c$. This means $\sum_{j=1}^J k_j = 1$.

3. The estimate is best in the sense that it minimizes the expected error squared; that is, $E(|\hat{c} - c|^2)$ is minimized.

The resulting vector $\mathbf{k}$ is calculated to be

$$k_i = \sigma_i^{-2} / (\sum_{j=1}^J \sigma_j^{-2}),$$

and the BLUE estimator of c is then

$$\hat{c} = \sum_{i=1}^J z_i \sigma_i^{-2} / (\sum_{j=1}^J \sigma_j^{-2}).$$

36.2 A More General Case

Suppose now that our data vector is $\mathbf{z} = H\mathbf{x} + \mathbf{v}$. Here, $\mathbf{x}$ is an unknown vector whose value is to be estimated, the random vector $\mathbf{v}$ is additive noise whose mean is $E(\mathbf{v}) = 0$ and whose known covariance matrix is $Q = E(\mathbf{v}\mathbf{v}^\dagger)$, not necessarily diagonal, and the known matrix H is J by N, with $J > N$. Now we seek an estimate of the vector $\mathbf{x}$. We now use the following rules:

1. The estimate $\hat{\mathbf{x}}$ must have the form $\hat{\mathbf{x}} = K^\dagger \mathbf{z}$, where the matrix K is to be determined.

2. The estimate is unbiased; that is, $E(\hat{\mathbf{x}}) = \mathbf{x}$.

3. The K is determined as the minimizer of the expected squared error; that is, once again we minimize $E(|\hat{\mathbf{x}} - \mathbf{x}|^2)$.

Exercise 36.1 *Show that for the estimator to be unbiased we need $K^\dagger H = I$, the identity matrix.*

Exercise 36.2 *Show that*

$$E(|\hat{\mathbf{x}} - \mathbf{x}|^2) = \operatorname{trace} K^\dagger Q K.$$

Hints: Write the left side as

$$E(\operatorname{trace}((\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})^\dagger)).$$

Also use the fact that the trace and expected-value operations commute.

The problem then is to minimize trace $K^\dagger Q K$ subject to the constraint equation $K^\dagger H = I$. We solve this problem using a technique known as *prewhitening*.

Since the noise covariance matrix Q is Hermitian and nonnegative definite, we have $Q = UDU^\dagger$, where the columns of U are the (mutually orthogonal) eigenvectors of Q and D is a diagonal matrix whose diagonal entries are the (necessarily nonnegative) eigenvalues of Q; therefore, $U^\dagger U = I$. We call $C = UD^{1/2}U^\dagger$ the Hermitian square root of Q, since $C^\dagger = C$ and $C^2 = Q$. We assume that Q is invertible, so that C is also. Given the system of equations

$$\mathbf{z} = H\mathbf{x} + \mathbf{v},$$

as before, we obtain a new system

$$\mathbf{y} = G\mathbf{x} + \mathbf{w}$$

by multiplying both sides by $C^{-1} = Q^{-1/2}$; here, $G = C^{-1}H$ and $\mathbf{w} = C^{-1}\mathbf{v}$. The new noise correlation matrix is

$$E(\mathbf{w}\mathbf{w}^\dagger) = C^{-1}QC^{-1} = I,$$

so the new noise is white. For this reason the step of multiplying by C^{-1} is called *prewhitening*.

With $J = CK$ and $M = C^{-1}H$, we have

$$K^\dagger Q K = J^\dagger J$$

and

$$K^\dagger H = J^\dagger M.$$

Our problem then is to minimize trace $J^\dagger J$, subject to $J^\dagger M = I$.

Let $L = L^\dagger = (M^\dagger M)^{-1}$ and let $f(J)$ be the function

$$f(J) = \text{trace}[(J^\dagger - L^\dagger M^\dagger)(J - ML)].$$

The minimum value of $f(J)$ is zero, which occurs when $J = ML$. Note that this choice for J has the property $J^\dagger M = I$. So, minimizing $f(J)$ is equivalent to minimizing $f(J)$ subject to the constraint $J^\dagger M = I$ and both problems have the solution $J = ML$. But minimizing $f(J)$ subject to $J^\dagger M = I$ is equivalent to minimizing trace $J^\dagger J$ subject to $J^\dagger M = I$, which is our original problem. Therefore, the optimal choice for J is $J = ML$. Consequently, the optimal choice for K is

$$K = Q^{-1}HL = Q^{-1}H(H^\dagger Q^{-1}H)^{-1},$$

and the BLUE estimate of $\mathbf{x}$ is

$$\mathbf{x}_{BLUE} = \hat{\mathbf{x}} = K^\dagger \mathbf{z} = (H^\dagger Q^{-1}H)^{-1}H^\dagger Q^{-1}\mathbf{z}.$$

The simplest case can be obtained from this more general formula by taking $N = 1$, $H = (1, 1, ..., 1)^T$ and $\mathbf{x} = c$.

Note that if the noise is *white*, that is, $Q = \sigma^2 I$, then $\hat{\mathbf{x}} = (H^\dagger H)^{-1}H^\dagger \mathbf{z}$, which is the least-squares solution of the equation $\mathbf{z} = H\mathbf{x}$. The effect of requiring that the estimate be unbiased is that, in this case, we simply ignore the presence of the noise and calculate the least squares solution of the noise-free equation $\mathbf{z} = H\mathbf{x}$.

The BLUE estimator involves nested inversion, making it difficult to calculate, especially for large matrices. In the exercise that follows, we discover an approximation of the BLUE that is easier to calculate.

Exercise 36.3 *Show that for $\epsilon > 0$ we have*

$$(H^\dagger Q^{-1}H + \epsilon I)^{-1}H^\dagger Q^{-1} = H^\dagger (HH^\dagger + \epsilon Q)^{-1}. \tag{36.1}$$

Hint: Use the identity

$$H^\dagger Q^{-1}(HH^\dagger + \epsilon Q) = (H^\dagger Q^{-1}H + \epsilon I)H^\dagger.$$

It follows from Equation (36.1) that

$$\mathbf{x}_{BLUE} = \lim_{\epsilon \to 0} H^\dagger (HH^\dagger + \epsilon Q)^{-1}\mathbf{z}. \tag{36.2}$$

Therefore, we can get an approximation of the BLUE estimate by selecting $\epsilon > 0$ near zero, solving the system of linear equations

$$(HH^\dagger + \epsilon Q)\mathbf{a} = \mathbf{z}$$

for $\mathbf{a}$ and taking $\mathbf{x} = H^\dagger \mathbf{a}$.

36.3 Some Useful Matrix Identities

In the exercise that follows we consider several matrix identities that are useful in developing the Kalman filter.

Exercise 36.4 *Establish the following identities, assuming that all the products and inverses involved are defined:*

$$CDA^{-1}B(C^{-1} - DA^{-1}B)^{-1} = (C^{-1} - DA^{-1}B)^{-1} - C; \tag{36.3}$$

$$(A - BCD)^{-1} = A^{-1} + A^{-1}B(C^{-1} - DA^{-1}B)^{-1}DA^{-1}; \tag{36.4}$$

$$A^{-1}B(C^{-1} - DA^{-1}B)^{-1} = (A - BCD)^{-1}BC; \tag{36.5}$$

$$(A - BCD)^{-1} = (I + GD)A^{-1}, \tag{36.6}$$

for

$$G = A^{-1}B(C^{-1} - DA^{-1}B)^{-1}.$$

Hints: To get Equation (36.3) use

$$C(C^{-1} - DA^{-1}B) = I - CDA^{-1}B.$$

For the second identity, multiply both sides of Equation (36.4) on the left by $A - BCD$ and at the appropriate step use Equation (36.3). For Equation (36.5) show that

$$BC(C^{-1} - DA^{-1}B) = B - BCDA^{-1}B = (A - BCD)A^{-1}B.$$

For Equation (36.6), substitute what G is and use Equation (36.4).

36.4 The BLUE with a Prior Estimate

In Kalman filtering we have the situation in which we want to estimate an unknown vector $\mathbf{x}$ given measurements $\mathbf{z} = H\mathbf{x} + \mathbf{v}$, but also given a prior estimate $\mathbf{y}$ of $\mathbf{x}$. It is the case there that $E(\mathbf{y}) = E(\mathbf{x})$, so we write $\mathbf{y} = \mathbf{x} + \mathbf{w}$, with $\mathbf{w}$ independent of both $\mathbf{x}$ and $\mathbf{v}$ and $E(\mathbf{w}) = \mathbf{0}$. The covariance matrix for $\mathbf{w}$ we denote by $E(\mathbf{w}\mathbf{w}^\dagger) = R$. We now require that the estimate $\hat{\mathbf{x}}$ be linear in both $\mathbf{z}$ and $\mathbf{y}$; that is, the estimate has the form

$$\hat{\mathbf{x}} = C^\dagger\mathbf{z} + D^\dagger\mathbf{y},$$

for matrices C and D to be determined.

The approach is to apply the BLUE to the combined system of linear equations

$$\mathbf{z} = H\mathbf{x} + \mathbf{v} \quad \text{and}$$

$$\mathbf{y} = \mathbf{x} + \mathbf{w}.$$

In matrix language this combined system becomes $\mathbf{u} = J\mathbf{x} + \mathbf{n}$, with $\mathbf{u}^T = [\mathbf{z}^T\ \mathbf{y}^T]$, $J^T = [H^T\ I^T]$, and $\mathbf{n}^T = [\mathbf{v}^T\ \mathbf{w}^T]$. The noise covariance matrix becomes

$$P = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}.$$

The BLUE estimate is $K^\dagger \mathbf{u}$, with $K^\dagger J = I$. Minimizing the variance, we find that the optimal $K^\dagger$ is

$$K^\dagger = (J^\dagger P^{-1} J)^{-1} J^\dagger P^{-1}.$$

The optimal estimate is then

$$\hat{\mathbf{x}} = (H^\dagger Q^{-1} H + R^{-1})^{-1}(H^\dagger Q^{-1}\mathbf{z} + R^{-1}\mathbf{y}).$$

Therefore,

$$C^\dagger = (H^\dagger Q^{-1} H + R^{-1})^{-1} H^\dagger Q^{-1}$$

and

$$D^\dagger = (H^\dagger Q^{-1} H + R^{-1})^{-1} R^{-1}.$$

Using the matrix identities in Equations (36.4) and (36.5) we can rewrite this estimate in the more useful form

$$\hat{\mathbf{x}} = \mathbf{y} + G(\mathbf{z} - H\mathbf{y}),$$

for

$$G = RH^\dagger (Q + HRH^\dagger)^{-1}. \tag{36.7}$$

The covariance matrix of the optimal estimator is $K^\dagger P K$, which can be written as

$$K^\dagger P K = (R^{-1} + H^\dagger Q^{-1} H)^{-1} = (I - GH)R.$$

In the context of the Kalman filter, R is the covariance of the prior estimate of the current state, G is the Kalman gain matrix, and $K^\dagger P K$ is the posterior covariance of the current state. The algorithm proceeds recursively from one state to the next in time.

36.5 Adaptive BLUE

We have assumed so far that we know the covariance matrix Q corresponding to the measurement noise. If we do not, then we may attempt to estimate Q from the measurements themselves; such methods are called *noise-adaptive*. To illustrate, let the *innovations* vector be $\mathbf{e} = \mathbf{z} - H\mathbf{y}$. Then the covariance matrix of $\mathbf{e}$ is $S = HRH^\dagger + Q$. Having obtained an estimate $\hat{S}$ of S from the data, we use $\hat{S} - HRH^\dagger$ in place of Q in Equation (36.7).

36.6 The Kalman Filter

So far in this chapter we have focused on the filtering problem: given the data vector $\mathbf{z}$, estimate $\mathbf{x}$, assuming that $\mathbf{z}$ consists of noisy measurements of $H\mathbf{x}$; that is, $\mathbf{z} = H\mathbf{x} + \mathbf{v}$. An important extension of this problem is that of stochastic prediction. Shortly, we discuss the Kalman-filter method for solving this more general problem. One area in which prediction plays an important role is the tracking of moving targets, such as ballistic missiles, using radar. The range to the target, its angle of elevation, and its azimuthal angle are all functions of time governed by linear differential equations. The *state vector* of the system at time t might then be a vector with nine components, the three functions just mentioned, along with their first and second derivatives. In theory, if we knew the initial state perfectly and our differential equations model of the physics was perfect, that would be enough to determine the future states. In practice neither of these is true, and we need to assist the differential equation by taking radar measurements of the state at various times. The problem then is to estimate the state at time t using both the measurements taken prior to time t and the estimate based on the physics.

When such tracking is performed digitally, the functions of time are replaced by discrete sequences. Let the state vector at time $k\Delta t$ be denoted by $\mathbf{x}_k$, for k an integer and $\Delta t > 0$. Then, with the derivatives in the differential equation approximated by divided differences, the physical model for the evolution of the system in time becomes

$$\mathbf{x}_k = A_{k-1}\mathbf{x}_{k-1} + \mathbf{m}_{k-1}.$$

The matrix A_{k-1}, which we assume is known, is obtained from the differential equation, which may have nonconstant coefficients, as well as from the divided difference approximations to the derivatives. The random vector sequence $\mathbf{m}_{k-1}$ represents the error in the physical model due to the discretization and necessary simplification inherent in the original differential equation itself. We assume that the expected value of $\mathbf{m_k}$ is zero for each k. The covariance matrix is $E(\mathbf{m}_k\mathbf{m}_k^\dagger) = M_k$.

At time $k\Delta t$ we have the measurements

$$\mathbf{z}_k = H_k\mathbf{x}_k + \mathbf{v}_k,$$

where H_k is a known matrix describing the nature of the linear measurements of the state vector and the random vector $\mathbf{v}_k$ is the noise in these measurements. We assume that the mean value of $\mathbf{v}_k$ is zero for each k. The covariance matrix is $E(\mathbf{v}_k\mathbf{v}_k^\dagger) = Q_k$. We assume that the initial state vector $\mathbf{x}_0$ is arbitrary.

Given an unbiased estimate $\hat{\mathbf{x}}_{k-1}$ of the state vector $\mathbf{x}_{k-1}$, our prior estimate of $\mathbf{x}_k$ based solely on the physics is

$$\mathbf{y}_k = A_{k-1}\hat{\mathbf{x}}_{k-1}.$$

Exercise 36.5 *Show that $E(\mathbf{y}_k - \mathbf{x}_k) = 0$, so the prior estimate of $\mathbf{x}_k$ is unbiased. We can then write $\mathbf{y}_k = \mathbf{x}_k + \mathbf{w}_k$, with $E(\mathbf{w}_k) = \mathbf{0}$.*

36.7 Kalman Filtering and the BLUE

The *Kalman filter* [163, 130, 86] is a recursive algorithm to estimate the state vector $\mathbf{x}_k$ at time $k\Delta t$ as a linear combination of the vectors $\mathbf{z}_k$ and $\mathbf{y}_k$. The estimate $\hat{\mathbf{x}}_k$ will have the form

$$\hat{\mathbf{x}}_k = C_k^\dagger\mathbf{z}_k + D_k^\dagger\mathbf{y}_k, \tag{36.8}$$

for matrices C_k and D_k to be determined. As we shall see, this estimate can also be written as

$$\hat{\mathbf{x}}_k = \mathbf{y}_k + G_k(\mathbf{z}_k - H_k\mathbf{y}_k), \tag{36.9}$$

which shows that the estimate involves a prior prediction step, the $\mathbf{y}_k$, followed by a correction step, in which $H_k\mathbf{y}_k$ is compared to the measured data vector $\mathbf{z}_k$; such estimation methods are sometimes called *predictor-corrector methods*.

In our discussion of the BLUE, we saw how to incorporate a prior estimate of the vector to be estimated. The trick was to form a larger matrix equation and then to apply the BLUE to that system. The Kalman filter does just that.

The correction step in the Kalman filter uses the BLUE to solve the combined linear system

$$\mathbf{z}_k = H_k\mathbf{x}_k + \mathbf{v}_k$$

and

$$\mathbf{y}_k = \mathbf{x}_k + \mathbf{w}_k.$$

The covariance matrix of $\hat{\mathbf{x}}_{k-1} - \mathbf{x}_{k-1}$ is denoted by P_{k-1}, and we let $Q_k = E(\mathbf{w}_k \mathbf{w}_k^\dagger)$. The covariance matrix of $\mathbf{y}_k - \mathbf{x}_k$ is

$$\text{cov}(\mathbf{y}_k - \mathbf{x}_k) = R_k = M_{k-1} + A_{k-1} P_{k-1} A_{k-1}^\dagger.$$

It follows from our earlier discussion of the BLUE that the estimate of $\mathbf{x}_k$ is

$$\hat{\mathbf{x}}_k = \mathbf{y}_k + G_k(\mathbf{z}_k - H\mathbf{y}_k),$$

with

$$G_k = R_k H_k^\dagger (Q_k + H_k R_k H_k^\dagger)^{-1}.$$

Then, the covariance matrix of $\hat{\mathbf{x}}_k - \mathbf{x}_k$ is

$$P_k = (I - G_k H_k) R_k.$$

The recursive procedure is to go from P_{k-1} and M_{k-1} to R_k, then to G_k, from which $\hat{\mathbf{x}}_k$ is formed, and finally to P_k, which, along with the known matrix M_k, provides the input to the next step. The time-consuming part of this recursive algorithm is the matrix inversion in the calculation of G_k. Simpler versions of the algorithm are based on the assumption that the matrices Q_k are diagonal, or on the convergence of the matrices G_k to a limiting matrix G [86].

There are many variants of the Kalman filter, corresponding to variations in the physical model, as well as in the statistical assumptions. The differential equation may be nonlinear, so that the matrices A_k depend on $\mathbf{x}_k$. The system noise sequence $\{\mathbf{w}_k\}$ and the measurement noise sequence $\{\mathbf{v}_k\}$ may be correlated. For computational convenience the various functions that describe the state may be treated separately. The model may include known external inputs to drive the differential system, as in the tracking of spacecraft capable of firing booster rockets. Finally, the noise covariance matrices may not be known *a priori* and adaptive filtering may be needed. We discuss this last issue briefly in the next section.

36.8 Adaptive Kalman Filtering

As in [86] we consider only the case in which the covariance matrix Q_k of the measurement noise $\mathbf{v}_k$ is unknown. As we saw in the discussion of adaptive BLUE, the covariance matrix of the innovations vector $\mathbf{e}_k = \mathbf{z}_k - H_k \mathbf{y}_k$ is

$$S_k = H_k R_k H_k^\dagger + Q_k.$$

Once we have an estimate for S_k, we estimate Q_k using

$$\hat{Q}_k = \hat{S}_k - H_k R_k H_k^\dagger.$$

We might assume that S_k is independent of k and estimate $S_k = S$ using past and present innovations; for example, we could use

$$\hat{S} = \frac{1}{k-1} \sum_{j=1}^{k} (\mathbf{z}_j - H_j \mathbf{y}_j)(\mathbf{z}_j - H_j \mathbf{y}_j)^\dagger.$$

Chapter 37

The BLUE and the Least Squares Estimators

37.1 Difficulties with the BLUE

As we saw in the previous chapter, the best linear unbiased estimate of $\mathbf{x}$, given the observed vector $\mathbf{z} = H\mathbf{x} + \mathbf{v}$, is

$$\mathbf{x}_{BLUE} = (H^\dagger Q^{-1} H)^{-1} H^\dagger Q^{-1} \mathbf{z}, \tag{37.1}$$

where Q is the invertible covariance matrix of the mean zero noise vector $\mathbf{v}$ and H is a J by N matrix with $J \geq N$ and $H^\dagger H$ invertible. Even if we know Q exactly, the double inversion in Equation (37.1) makes it difficult to calculate the BLUE estimate, especially for large vectors $\mathbf{z}$. It is often the case in practice that we do not know Q precisely and must estimate or model it. Because good approximations of Q do not necessarily lead to good approximations of Q^{-1}, the calculation of the BLUE is further complicated. For these reasons one may decide to use the least squares estimate

$$\mathbf{x}_{LS} = (H^\dagger H)^{-1} H^\dagger \mathbf{z} \tag{37.2}$$

instead. We are therefore led to consider when the two estimation methods produce the same answers; that is, when we have

$$(H^\dagger H)^{-1} H^\dagger = (H^\dagger Q^{-1} H)^{-1} H^\dagger Q^{-1}. \tag{37.3}$$

In this chapter we state and prove a theorem that answers this question. The proof relies on the results of several exercises, useful in themselves, that involve basic facts of linear algebra.

37.2 Preliminaries from Linear Algebra

We begin with some definitions. Let S be a subspace of finite-dimensional Euclidean space R^J and Q a J by J Hermitian matrix. We denote by $Q(S)$ the set

$$Q(S) = \{\mathbf{t} | \text{there exists } \mathbf{s} \in S \text{ with } \mathbf{t} = Q\mathbf{s}\}$$

and by $Q^{-1}(S)$ the set

$$Q^{-1}(S) = \{\mathbf{u} | Q\mathbf{u} \in S\}.$$

Note that the set $Q^{-1}(S)$ is defined whether or not Q is invertible.

We denote by $S^{\perp}$ the set of vectors $\mathbf{u}$ that are orthogonal to every member of S; that is,

$$S^{\perp} = \{\mathbf{u} | \mathbf{u}^{\dagger}\mathbf{s} = 0, \text{for every } \mathbf{s} \in S\}.$$

Let H be a J by N matrix. Then $CS(H)$, the column space of H, is the subspace of R^J consisting of all the linear combinations of the columns of H. The null space of $H^{\dagger}$, denoted $NS(H^{\dagger})$, is the subspace of R^J containing all the vectors $\mathbf{w}$ for which $H^{\dagger}\mathbf{w} = 0$.

Exercise 37.1 *Show that* $CS(H)^{\perp} = NS(H^{\dagger})$.

Hint: If $\mathbf{v} \in CS(H)^{\perp}$, then $\mathbf{v}^{\dagger}H\mathbf{x} = 0$ for all $\mathbf{x}$, including $\mathbf{x} = H^{\dagger}\mathbf{v}$.

Exercise 37.2 *Show that* $CS(H) \cap NS(H^{\dagger}) = \{\mathbf{0}\}$.

Hint: If $\mathbf{y} = H\mathbf{x} \in NS(H^{\dagger})$ consider $||\mathbf{y}||^2 = \mathbf{y}^{\dagger}\mathbf{y}$.

Exercise 37.3 *Let S be any subspace of R^J. Show that if Q is invertible and $Q(S) = S$ then $Q^{-1}(S) = S$.*

Hint: If $Q\mathbf{t} = Q\mathbf{s}$ then $\mathbf{t} = \mathbf{s}$.

Exercise 37.4 *Let Q be Hermitian. Show that $Q(S)^{\perp} = Q^{-1}(S^{\perp})$ for every subspace S. If Q is also invertible then $Q^{-1}(S)^{\perp} = Q(S^{\perp})$. Find an example of a non-invertible Q for which $Q^{-1}(S)^{\perp}$ and $Q(S^{\perp})$ are different.*

We assume, for the remainder of this chapter, that Q is Hermitian and invertible and that the matrix $H^\dagger H$ is invertible. Note that the matrix $H^\dagger Q^{-1} H$ need not be invertible under these assumptions. We shall denote by S an arbitrary subspace of R^J.

Exercise 37.5 *Show that $Q(S) = S$ if and only if $Q(S^\perp) = S^\perp$.*

Hint: Use Exercise 37.4.

Exercise 37.6 *Show that if $Q(CS(H)) = CS(H)$ then $H^\dagger Q^{-1} H$ is invertible.*

Hint: Show that $H^\dagger Q^{-1} H\mathbf{x} = \mathbf{0}$ if and only if $\mathbf{x} = \mathbf{0}$. Recall that $Q^{-1}H\mathbf{x} \in CS(H)$, by Exercise 37.4. Then use Exercise 37.2.

37.3 When are the BLUE and the LS Estimator the Same?

We are looking for conditions on Q and H that imply Equation (37.3), which we rewrite as

$$H^\dagger = (H^\dagger Q^{-1} H)(H^\dagger H)^{-1} H^\dagger Q \tag{37.4}$$

or

$$H^\dagger T\mathbf{x} = \mathbf{0}$$

for all $\mathbf{x}$, where

$$T = I - Q^{-1}H(H^\dagger H)^{-1}H^\dagger Q.$$

In other words, we want $T\mathbf{x} \in NS(H^\dagger)$ for all $\mathbf{x}$. The theorem is the following:

Theorem 37.1 *We have $T\mathbf{x} \in NS(H^\dagger)$ for all $\mathbf{x}$ if and only if $Q(CS(H)) = CS(H)$.*

An equivalent form of this theorem was proven by Anderson in [4]; he attributes a portion of the proof to Magness and McQuire [185]. The proof we give here is due to Kheifets [165] and is much simpler than Anderson's proof. The proof of the theorem is simplified somewhat by first establishing the result in the next exercise.

Exercise 37.7 *Show that if Equation (37.4) holds, then the matrix $H^\dagger Q^{-1} H$ is invertible.*

Hints: Recall that we have assumed that $CS(H^\dagger) = R^J$ when we assumed that $H^\dagger H$ is invertible. From equation (37.4) it follows that $CS(H^\dagger Q^{-1} H) = R^J$.

The Proof of Theorem 37.1: Assume first that $Q(CS(H)) = CS(H)$, which, as we now know, also implies $Q(NS(H^\dagger)) = NS(H^\dagger)$, as well as $Q^{-1}(CS(H)) = CS(H)$, $Q^{-1}(NS(H^\dagger)) = NS(H^\dagger)$, and the invertibility of the matrix $H^\dagger Q^{-1} H$. Every $\mathbf{x} \in R^J$ has the form $\mathbf{x} = H\mathbf{a} + \mathbf{w}$, for some $\mathbf{a}$ and $\mathbf{w} \in NS(H^\dagger)$. We show that $T\mathbf{x} = \mathbf{w}$, so that $T\mathbf{x} \in NS(H^\dagger)$ for all $\mathbf{x}$. We have

$$T\mathbf{x} = TH\mathbf{a} + T\mathbf{w} =$$

$$\mathbf{x} - Q^{-1}H(H^\dagger H)^{-1}H^\dagger QH\mathbf{a} - Q^{-1}H(H^\dagger H)^{-1}H^\dagger Q\mathbf{w}.$$

We know that $QH\mathbf{a} = H\mathbf{b}$ for some $\mathbf{b}$, so that $H\mathbf{a} = Q^{-1}H\mathbf{b}$. We also know that $Q\mathbf{w} = \mathbf{v} \in NS(H^\dagger)$, so that $\mathbf{w} = Q^{-1}\mathbf{v}$. Then, continuing our calculations, we have

$$T\mathbf{x} = \mathbf{x} - Q^{-1}H\mathbf{b} - \mathbf{0} = \mathbf{x} - H\mathbf{a} = \mathbf{w},$$

so $T\mathbf{x} \in NS(H^\dagger)$.

Conversely, suppose now that $T\mathbf{x} \in NS(H^\dagger)$ for all $\mathbf{x}$, which, as we have seen, is equivalent to Equation (37.4). We show that $Q^{-1}(NS(H^\dagger) = NS(H^\dagger)$. First, let $\mathbf{v} \in Q^{-1}(NS(H^\dagger))$; we show $\mathbf{v} \in NS(H^\dagger)$. We have

$$H^\dagger\mathbf{v} = (H^\dagger Q^{-1}H)(H^\dagger H)^{-1}H^\dagger Q\mathbf{v},$$

which is zero, since $H^\dagger Q\mathbf{v} = \mathbf{0}$. So, we have shown that $Q^{-1}(NS(H^\dagger)) \subseteq NS(H^\dagger)$. To complete the proof, we take an arbitrary member $\mathbf{v}$ of $NS(H^\dagger)$ and show that $\mathbf{v}$ is in $Q^{-1}(NS(H^\dagger))$; that is, $Q\mathbf{v} \in NS(H^\dagger)$. We know that $Q\mathbf{v} = H\mathbf{a} + \mathbf{w}$, for $\mathbf{w} \in NS(H^\dagger)$, and

$$\mathbf{a} = (H^\dagger H)^{-1}H^\dagger Q\mathbf{v},$$

so that

$$H\mathbf{a} = H(H^\dagger H)^{-1}H^\dagger Q\mathbf{v}.$$

Then, using Exercise 37.7, we have

$$Q\mathbf{v} = H(H^\dagger H)^{-1}H^\dagger Q\mathbf{v} + \mathbf{w}$$

$$= H(H^\dagger Q^{-1}H)^{-1}H^\dagger Q^{-1}Q\mathbf{v} + \mathbf{w}$$

$$= H(H^\dagger Q^{-1}H)^{-1}H^\dagger\mathbf{v} + \mathbf{w} = \mathbf{w}.$$

So $Q\mathbf{v} = \mathbf{w}$, which is in $NS(H^\dagger)$. This completes the proof. ∎

Chapter 38

Linear Inequalities

Most books on linear algebra devote considerable space to the problem of solving a consistent or inconsistent system of linear equations, say $Ax = b$. Problems involving linear inequalities, such as solving $Ax \geq b$, attract less attention, although such problems play a crucial role in linear programming. The term *linear programming* (LP) refers to the problem of optimizing a linear function of several variables over linear equality or inequality constraints. Such problems arise in many areas of applications. It is common, in applications, for A to be quite large, necessitating the use of an iterative algorithm to solve the problem. Dantzig's *Simplex Method* (see [65]) is the best known iterative method for solving LP problems.

38.1 Theorems of the Alternative

Later in this chapter we shall present David Gale's proof of his *strong duality theorem* in linear programming ([128]). His proof makes use of a theorem concerning linear inequalities known as a *theorem of the alternative*. For that reason, we begin with a discussion of these types of theorems.

38.1.1 A Theorem of the Alternative

The following theorem is a good illustration of a type of theorem known as *Theorems of the Alternative*. These theorems assert that precisely one of two problems will have a solution. The proof illustrates how we should go about proving such theorems.

Theorem 38.1 (Gale I)[128] *Precisely one of the following is true:*

- **(1)** *there is x such that $Ax = b$;*
- **(2)** *there is y such that $A^T y = 0$ and $b^T y = 1$.*

Proof: First, we show that it is not possible for both to be true at the same time. Suppose that $Ax = b$ and $A^T y = 0$. Then $b^T y = x^T A^T y = 0$, so that we cannot have $b^T y = 1$. By Theorem 32.1, the fundamental decomposition theorem from linear algebra, we know that, for any b, there are unique x and w with $A^T w = 0$ such that $b = Ax + w$. Clearly, $b = Ax$ if and only if $w = 0$. Also, $b^T y = w^T y$. Therefore, if alternative **(1)** does not hold, we must have w non-zero, in which case $A^T y = 0$ and $b^T y = 1$, for $y = w/||w||^2$, so alternative **(2)** holds. ∎

In this section we consider several other theorems of this type.

38.1.2 More Theorems of the Alternative

Theorem 38.2 (Farkas' Lemma)[117] *Precisely one of the following is true:*

- **(1)** *there is $x \geq 0$ such that $Ax = b$;*
- **(2)** *there is y such that $A^T y \geq 0$ and $b^T y < 0$.*

Proof: We can restate the lemma as follows: there is a vector y with $A^T y \geq 0$ and $b^T y < 0$ if and only if b is not a member of the convex set $C = \{Ax | x \geq 0\}$. If b is not in C, which is closed and convex, then, by the Separation Theorem (see [65]), there is a non-zero vector a and real α with

$$a^T b < \alpha \leq a^T Ax = (A^T a)^T x,$$

for all $x \geq 0$. Since $(A^T a)^T x$ is bounded below, as x runs over all non-negative vectors, it follows that $A^T a \geq 0$. Choosing $x = 0$, we have $\alpha \leq 0$. Then let $y = a$. Conversely, if $Ax = b$ does have a non-negative solution x, then $A^T y \geq 0$ implies that $0 \leq y^T Ax = y^T b \geq 0$. ∎

The next theorem can be obtained from Farkas' Lemma.

Theorem 38.3 (Gale II)[128] *Precisely one of the following is true:*

- **(1)** *there is x such that $Ax \leq b$;*
- **(2)** *there is $y \geq 0$ such that $A^T y = 0$ and $b^T y < 0$.*

Proof: First, if both are true, then $0 \leq y^T(b - Ax) = y^T b - 0 = y^T b$, which is a contradiction. Now assume that **(2)** does not hold. Therefore, for every $y \geq 0$ with $A^T y = 0$, we have $b^T y \geq 0$. Let $B = [\, A \quad b \,]$. Then the system $B^T y = [\, 0 \quad -1 \,]^T$ has no non-negative solution. Applying Farkas' Lemma, we find that there is a vector $w = [\, z \quad \gamma \,]^T$ with $Bw \geq 0$ and $[\, 0 \quad -1 \,]\, w < 0$. So, $Az + \gamma b \geq 0$ and $\gamma > 0$. Let $x = -\frac{1}{\gamma} z$ to get $Ax \leq b$, so that **(1)** holds. ∎

Theorem 38.4 (Gordan)[137] *Precisely one of the following is true:*

- **(1)** *there is x such that $Ax < 0$;*
- **(2)** *there is $y \geq 0$, $y \neq 0$, such that $A^T y = 0$.*

Proof: First, if both are true, then $0 < -y^T Ax = 0$, which cannot be true. Now assume that there is no non-zero $y \geq 0$ with $A^T y = 0$. Then, with $e = (1, 1, ..., 1)^T$, $C = [\, A \quad e \,]$, and $d = (0, 0, ..., 0, 1)^T$, there is no non-negative solution of $C^T y = d$. From Farkas' Lemma we then know that there is a vector $z = [\, u \quad \gamma \,]^T$, with $Cz = Au + \gamma e \geq 0$, and $d^T z < 0$. Then $Ax < 0$ for $x = -u$. ∎

Here are several more theorems of the alternative.

Theorem 38.5 (Stiemke I)[229] *Precisely one of the following is true:*

- **(1)** *there is x such that $Ax \leq 0$ and $Ax \neq 0$;*
- **(2)** *there is $y > 0$ such that $A^T y = 0$.*

Theorem 38.6 (Stiemke II)[229] *Let c be a fixed non-zero vector. Precisely one of the following is true:*

- **(1)** *there is x such that $Ax \leq 0$ and $c^T x \geq 0$ and not both $Ax = 0$ and $c^T x = 0$;*
- **(2)** *there is $y > 0$ such that $A^T y = c$.*

Theorem 38.7 (Gale III)[128] *Let c be a fixed non-zero vector. Precisely one of the following is true:*

- **(1)** *there is $x \geq 0$ such that $Ax \geq 0$ and $c^T x < 0$;*
- **(2)** *there is $y \geq 0$ such that $A^T y \leq c$.*

Proof: First, note that we cannot have both true at the same time, since we would then have

$$0 < x^T (c - A^T y) = c^T x - (Ax)^T y \leq c^T x,$$

which is a contradiction. Now suppose that **(2)** does not hold. Then there is no $w \geq 0$ such that

$$[\, A^T \quad I \,]\, w = c.$$

By Farkas' Lemma (Theorem 38.2), it follows that there is x with

$$\begin{bmatrix} A \\ I \end{bmatrix} x \geq 0,$$

and $c^T x < 0$. Therefore, $Ax \geq 0$, $Ix = x \geq 0$, and $c^T x < 0$; therefore, **(1)** holds. ∎

Theorem 38.8 (Von Neumann)[243] *Precisely one of the following is true:*

- **(1)** *there is* $x \geq 0$ *such that* $Ax > 0$*;*
- **(2)** *there is* $y \geq 0$*,* $y \neq 0$*, such that* $A^T y \leq 0$*.*

Proof: If both were true, then we would have

$$0 < (Ax)^T y = x^T (A^T y),$$

so that $A^T y \leq 0$ would be false. Now suppose that **(2)** does not hold. Then there is no $y \geq 0$, $y \neq 0$, with $A^T y \leq 0$. Consequently, there is no $y \geq 0$, $y \neq 0$, such that

$$\begin{bmatrix} A^T \\ -u^T \end{bmatrix} y = \begin{bmatrix} A^T y \\ -u^T y \end{bmatrix} \leq \begin{bmatrix} 0 \\ -1 \end{bmatrix},$$

where $u^T = (1, 1, ..., 1)$. By Theorem 38.7, there is

$$z = \begin{bmatrix} x \\ \alpha \end{bmatrix} \geq 0,$$

such that

$$[\, A \quad -u \,]\, z = [\, A \quad -u \,] \begin{bmatrix} x \\ \alpha \end{bmatrix} \geq 0,$$

and

$$[\, 0 \quad -1 \,]\, z = [\, 0 \quad -1 \,] \begin{bmatrix} x \\ \alpha \end{bmatrix} = -\alpha < 0.$$

Therefore, $\alpha > 0$ and $(Ax)_i - \alpha \geq 0$ for each i, and so $Ax > 0$ and **(1)** holds. ∎

Theorem 38.9 (Tucker)[235] *Precisely one of the following is true:*

- **(1)** *there is* $x \geq 0$ *such that* $Ax \geq 0$*,* $Ax \neq 0$*;*
- **(2)** *there is* $y > 0$ *such that* $A^T y \leq 0$*.*

38.1.3 Another Proof of Farkas' Lemma

In the previous section, we proved Farkas' Lemma, Theorem 38.2, using the Separation Theorem, the proof of which, in turn, depended here on the existence of the orthogonal projection onto any closed convex set. It is possible to prove Farkas' Lemma directly, along the lines of Gale [128].

Suppose that $Ax = b$ has no non-negative solution. If, indeed, it has no solution whatsoever, then $b = Ax + w$, where $w \neq 0$ and $A^T w = 0$.

Then we take $y = -w/||w||^2$. So suppose that $Ax = b$ does have solutions, but not any non-negative ones. The approach is to use induction on the number of columns of the matrix involved in the lemma.

If A has only one column, denoted a^1, then $Ax = b$ can be written as

$$x_1 a^1 = b.$$

Assuming that there are no non-negative solutions, it must follow that $x_1 < 0$. We take $y = -b$. Then

$$b^T y = -b^T b = -||b||^2 < 0,$$

while

$$A^T y = (a^1)^T(-b) = \frac{-1}{x_1} b^T b > 0.$$

Now assume that the lemma holds whenever the involved matrix has no more than $m-1$ columns. We show the same is true for m columns.

If there is no non-negative solution of the system $Ax = b$, then clearly there are no non-negative real numbers $x_1, x_2, ..., x_{m-1}$ such that

$$x_1 a^1 + x_2 a^2 + ... + x_{m-1} a^{m-1} = b,$$

where a^j denotes the jth column of the matrix A. By the induction hypothesis, there must be a vector v with

$$(a^j)^T v \geq 0,$$

for $j = 1, ..., m-1$, and $b^T v < 0$. If it happens that $(a^m)^T v \geq 0$ also, then we are done. If, on the other hand, we have $(a^m)^T v < 0$, then let

$$c^j = (a^j)^T a^m - (a^m)^T a^j,\ j = 1, ..., m-1,$$

and

$$d = (b^T v) a^m - ((a^m)^T v) b.$$

Then there are no non-negative real numbers $z_1, ..., z_{m-1}$ such that

$$z_1 c^1 + z_2 c^2 + ... + z_{m-1} c^{m-1} = d, \tag{38.1}$$

since, otherwise, it would follow from simple calculations that

$$\frac{-1}{(a^m)^T v}\Big([\sum_{j=1}^{m-1} z_j((a^j)^T v)] - b^T v\Big) a^m - \sum_{j=1}^{m-1} z_j((a^m)^T v) a^j = b.$$

Close inspection of this shows all the coefficients to be non-negative, which implies that the system $Ax = b$ has a non-negative solution, contrary to

our assumption. It follows, therefore, that there can be no non-negative solution to the system in Equation (38.1).

By the induction hypothesis, it follows that there is a vector u such that

$$(c^j)^T u \geq 0,\ j = 1, ..., m-1,$$

and

$$d^T u < 0.$$

Now let

$$y = ((a^m)^T u)v - ((a^m)^T v)u.$$

We can easily verify that

$$(a^j)^T y = (c^j)^T u \geq 0,\ j = 1, ..., m-1,$$

$$b^T y = d^T u < 0,$$

and

$$(a^m)^T y = 0,$$

so that

$$A^T y \geq 0,$$

and

$$b^T y < 0.$$

This completes the proof.

38.2 Linear Programming

We begin with an example.

38.2.1 An Example

Consider the problem of maximizing the function $f(x_1, x_2) = x_1 + 2x_2$, over all $x_1 \geq 0$ and $x_2 \geq 0$, for which the inequalities

$$x_1 + x_2 \leq 40,$$

and

$$2x_1 + x_2 \leq 60$$

are satisfied. The set of points satisfying all four inequalities is the quadrilateral with vertices $(0, 0)$, $(30, 0)$, $(20, 20)$, and $(0, 40)$; draw a picture. Since the level curves of the function f are straight lines, the maximum value must occur at one of these vertices; in fact, it occurs at $(0, 40)$ and

the maximum value of f over the constraint set is 80. Rewriting the problem as minimizing the function $-x_1 - 2x_2$, subject to $x_1 \geq 0$, $x_2 \geq 0$,

$$-x_1 - x_2 \geq -40,$$

and

$$-2x_1 - x_2 \geq -60,$$

the problem is now in what is called *primal canonical form*.

38.2.2 Canonical and Standard Forms

Let b and c be fixed vectors and A a fixed matrix. The problem

$$\text{minimize } z = c^T x, \text{ subject to } Ax \geq b, x \geq 0 \text{ (PC)} \tag{38.2}$$

is the so-called *primary problem* of LP, in *canonical form*. The *dual problem* in canonical form is

$$\text{maximize } w = b^T y, \text{ subject to } A^T y \leq c, y \geq 0. \text{ (DC)} \tag{38.3}$$

The primary problem, in *standard form*, is

$$\text{minimize } z = c^T x, \text{ subject to } Ax = b, x \geq 0 \text{ (PS)} \tag{38.4}$$

with the dual problem in standard form given by

$$\text{maximize } w = b^T y, \text{ subject to } A^T y \leq c. \text{ (DS)} \tag{38.5}$$

Notice that the dual problem in standard form does not require that y be nonnegative. Note also that (PS) makes sense only if the system $Ax = b$ has solutions. For that reason, we shall assume, for the standard problems, that the I by J matrix A has at least as many columns as rows, so $J \geq I$, and A has full rank I.

If we are given the primary problem in canonical form, we can convert it to standard form by augmenting the variables, that is, by defining

$$u_i = (Ax)_i - b_i, \tag{38.6}$$

for $i = 1, ..., I$, and rewriting $Ax \geq b$ as

$$\tilde{A}\tilde{x} = b, \tag{38.7}$$

for $\tilde{A} = [\, A \quad -I \,]$ and $\tilde{x} = [x^T u^T]^T$.

If we are given the primary problem in standard form, we can convert it to canonical form by writing the equations as inequalities, that is, by replacing $Ax = b$ with the two matrix inequalities $Ax \geq b$, and $(-A)x \geq -b$.

38.2.3 Weak Duality

Consider the problems (PS) and (DS). Say that x is *feasible* if $x \geq 0$ and $Ax = b$. Let F be the set of feasible x. Say that y is *feasible* if $A^T y \leq c$. The *Weak Duality Theorem* is the following:

Theorem 38.10 *Let x and y be feasible vectors. Then*

$$z = c^T x \geq b^T y = w. \tag{38.8}$$

Corollary 38.1 *If z is not bounded below, then there are no feasible y.*

Corollary 38.2 *If x and y are both feasible, and $z = w$, then both x and y are optimal for their respective problems.*

The proof of the theorem and its corollaries are left as exercises.

The nonnegative quantity $c^T x - b^T y$ is called the *duality gap*. The *complementary slackness condition* says that, for optimal x and y, we have

$$x_j(c_j - (A^T y)_j) = 0, \tag{38.9}$$

for each j, which says that the duality gap is zero. Primal-dual algorithms for solving linear programming problems are based on finding sequences $\{x^k\}$ and $\{y^k\}$ that drive the duality gap down to zero [194].

38.2.4 Strong Duality

The *Strong Duality Theorems* make a stronger statement. The following theorems are well known examples.

Theorem 38.11 *If one of the problems (PS) or (DS) has an optimal solution, then so does the other and $z = w$ for the optimal vectors.*

Theorem 38.12 Gale's Strong Duality Theorem*[128] If both problems (PC) and (DC) have feasible solutions, then both have optimal solutions and the optimal values are equal.*

Proof: We show that there are non-negative vectors x and y such that $Ax \geq b$, $A^T y \leq c$, and $b^T y - c^T x \geq 0$. It will then follow that $z = c^T x = b^T y = w$, so that x and y are both optimal. In matrix notation, we want to find $x \geq 0$ and $y \geq 0$ such that

$$\begin{bmatrix} A & 0 \\ 0 & -A^T \\ -c^T & b^T \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \geq \begin{bmatrix} b \\ -c \\ 0 \end{bmatrix}. \tag{38.10}$$

We assume that there are no $x \geq 0$ and $y \geq 0$ for which the inequalities in (38.10) hold. Then, according to Theorem 38.7, there are non-negative vectors s and t, and non-negative scalar ρ such that

$$\begin{bmatrix} -A^T & 0 & c \\ 0 & A & -b \end{bmatrix} \begin{bmatrix} s \\ t \\ \rho \end{bmatrix} \geq 0, \tag{38.11}$$

and

$$\begin{bmatrix} -b^T & c^T & 0 \end{bmatrix} \begin{bmatrix} s \\ t \\ \rho \end{bmatrix} < 0. \tag{38.12}$$

Note that ρ cannot be zero, for then we would have $A^T s \leq 0$ and $At \geq 0$. Taking feasible vectors x and y, we would find that $s^T Ax \leq 0$, which implies that $b^T s \leq 0$, and $t^T A^T y \geq 0$, which implies that $c^T t \geq 0$. Therefore, we could not also have $c^T t - b^T s < 0$.

Writing out the inequalities, we have

$$\rho c^T t \geq s^T At \geq s^T(\rho b) = \rho s^T b.$$

Using $\rho > 0$, we find that

$$c^T t \geq b^T s,$$

which is a contradiction. Therefore, there do exist $x \geq 0$ and $y \geq 0$ such that $Ax \geq b$, $A^T y \leq c$, and $b^T y - c^T x \geq 0$. ∎

In his book [128] Gale uses his strong duality theorem to obtain a proof of the *min-max* theorem in game theory (see [65]).

Chapter 39

Geometric Programming and the MART

Geometric Programming (GP) involves the minimization of functions of a special type, known as posynomials. The first systematic treatment of geometric programming appeared in the book [110], by Duffin, Peterson and Zener, the founders of geometric programming. As we shall see, the Generalized Arithmetic-Geometric Mean Inequality plays an important role in the theoretical treatment of geometric programming.

39.1 An Example of a GP Problem

The following optimization problem was presented originally by Duffin, *et al.* [110] and discussed by Peressini *et al.* in [203]. It illustrates well the type of problem considered in geometric programming. Suppose that 400 cubic yards of gravel must be ferried across a river in an open box of length t_1, width t_2 and height t_3. Each round-trip cost ten cents. The sides and the bottom of the box cost 10 dollars per square yard to build, while the ends of the box cost twenty dollars per square yard. The box will have no salvage value after it has been used. Determine the dimensions of the box that minimize the total cost.

With $t = (t_1, t_2, t_3)$, the cost function is

$$g(t) = \frac{40}{t_1 t_2 t_3} + 20t_1t_3 + 10t_1t_2 + 40t_2t_3, \tag{39.1}$$

which is to be minimized over $t_j > 0$, for $j = 1, 2, 3$. The function $g(t)$ is an example of a posynomial.

39.2 Posynomials and the GP Problem

Functions $g(t)$ of the form

$$g(t) = \sum_{i=1}^{n} c_i \Big(\prod_{j=1}^{m} t_j^{a_{ij}} \Big), \tag{39.2}$$

with $t = (t_1, ..., t_m)$, the $t_j > 0$, $c_i > 0$ and a_{ij} real, are called *posynomials*. The *geometric programming problem*, denoted (GP), is to minimize a given posynomial over positive t. In order for the minimum to be greater than zero, we need some of the a_{ij} to be negative.

We denote by $u_i(t)$ the function

$$u_i(t) = c_i \prod_{j=1}^{m} t_j^{a_{ij}}, \tag{39.3}$$

so that

$$g(t) = \sum_{i=1}^{n} u_i(t). \tag{39.4}$$

For any choice of $\delta_i > 0$, $i = 1, ..., n$, with

$$\sum_{i=1}^{n} \delta_i = 1,$$

we have

$$g(t) = \sum_{i=1}^{n} \delta_i \Big(\frac{u_i(t)}{\delta_i} \Big). \tag{39.5}$$

Applying the Generalized Arithmetic-Geometric Mean (GAGM) Inequality, we have

$$g(t) \geq \prod_{i=1}^{n} \Big(\frac{u_i(t)}{\delta_i} \Big)^{\delta_i}. \tag{39.6}$$

Therefore,

$$g(t) \geq \prod_{i=1}^{n} \Big(\frac{c_i}{\delta_i} \Big)^{\delta_i} \left(\prod_{i=1}^{n} \prod_{j=1}^{m} t_j^{a_{ij}\delta_i} \right), \tag{39.7}$$

or

$$g(t) \geq \prod_{i=1}^{n} \Big(\frac{c_i}{\delta_i} \Big)^{\delta_i} \Big(\prod_{j=1}^{m} t_j^{\sum_{i=1}^{n} a_{ij}\delta_i} \Big), \tag{39.8}$$

Suppose that we can find $\delta_i > 0$ with

$$\sum_{i=1}^{n} a_{ij}\delta_i = 0, \tag{39.9}$$

for each j. Then the inequality in (39.8) becomes

$$g(t) \geq v(\delta), \tag{39.10}$$

for

$$v(\delta) = \prod_{i=1}^{n} \left(\frac{c_i}{\delta_i}\right)^{\delta_i}. \tag{39.11}$$

39.3 The Dual GP Problem

The *dual geometric programming problem*, denoted (DGP), is to maximize the function $v(\delta)$, over all *feasible* $\delta = (\delta_1, ..., \delta_n)$, that is, all positive δ for which

$$\sum_{i=1}^{n} \delta_i = 1, \tag{39.12}$$

and

$$\sum_{i=1}^{n} a_{ij}\delta_i = 0, \tag{39.13}$$

for each $j = 1, ..., m$. Clearly, we have

$$g(t) \geq v(\delta), \tag{39.14}$$

for any positive t and feasible δ. Of course, there may be no feasible δ, in which case (DGP) is said to be *inconsistent*.

As we have seen, the inequality in (39.14) is based on the GAGM Inequality. We have equality in the GAGM Inequality if and only if the terms in the arithmetic mean are all equal. In this case, this says that there is a constant λ such that

$$\frac{u_i(t)}{\delta_i} = \lambda, \tag{39.15}$$

for each $i = 1, ..., n$. Using the fact that the δ_i sum to one, it follows that

$$\lambda = \sum_{i=1}^{n} u_i(t) = g(t), \tag{39.16}$$

and

$$\delta_i = \frac{u_i(t)}{g(t)}, \tag{39.17}$$

for each $i = 1, ..., n$. As the theorem below asserts, if t^* is positive and minimizes $g(t)$, then δ^*, the associated δ from Equation (39.17), is feasible and solves (DGP). Since we have equality in the GAGM Inequality now, we have

$$g(t^*) = v(\delta^*).$$

The main theorem in geometric programming is the following.

Theorem 39.1 *If $t^* > 0$ minimizes $g(t)$, then (DGP) is consistent. In addition, the choice*

$$\delta_i^* = \frac{u_i(t^*)}{g(t^*)} \tag{39.18}$$

is feasible and solves (DGP). Finally,

$$g(t^*) = v(\delta^*); \tag{39.19}$$

that is, there is no duality gap.

Proof: We have

$$\frac{\partial u_i}{\partial t_j}(t^*) = \frac{a_{ij} u_i(t^*)}{t_j^*}, \tag{39.20}$$

so that

$$t_j^* \frac{\partial u_i}{\partial t_j}(t^*) = a_{ij} u_i(t^*), \tag{39.21}$$

for each $j = 1, ..., m$. Since t^* minimizes $g(t)$, we have

$$0 = \frac{\partial g}{\partial t_j}(t^*) = \sum_{i=1}^{n} \frac{\partial u_i}{\partial t_j}(t^*), \tag{39.22}$$

so that, from Equation (39.21), we have

$$0 = \sum_{i=1}^{n} a_{ij} u_i(t^*), \tag{39.23}$$

for each $j = 1, ..., m$. It follows that δ^* is feasible. Since we have equality in the GAGM Inequality, we know

$$g(t^*) = v(\delta^*). \tag{39.24}$$

Therefore, δ^* solves (DGP). This completes the proof. ∎

39.4 Solving the GP Problem

The theorem suggests how we might go about solving (GP). First, we try to find a feasible δ^* that maximizes $v(\delta)$. This means we have to find a positive solution to the system of $m+1$ linear equations in n unknowns, given by

$$\sum_{i=1}^{n} \delta_i = 1, \tag{39.25}$$

and

$$\sum_{i=1}^{n} a_{ij}\delta_i = 0, \tag{39.26}$$

for $j = 1, ..., m$, such that $v(\delta)$ is maximized. As we shall see, the *multiplicative algebraic reconstruction technique* (MART) is an iterative procedure that we can use to find such δ. If there is no such vector, then (GP) has no minimizer. Once the desired δ^* has been found, we set

$$\delta_i^* = \frac{u_i(t^*)}{v(\delta^*)}, \tag{39.27}$$

for each $i = 1, ..., n$, and then solve for the entries of t^*. This last step can be simplified by taking logs; then we have a system of linear equations to solve for the values $\log t_j^*$.

39.5 Solving the DGP Problem

The iterative multiplicative algebraic reconstruction technique MART can be used to minimize the function $v(\delta)$, subject to linear equality constraints, provided that the matrix involved has nonnegative entries. We cannot apply the MART yet, because the matrix A^T does not satisfy these conditions.

39.5.1 The MART

The Kullback-Leibler, or KL distance [171] between positive numbers a and b is

$$KL(a, b) = a \log \frac{a}{b} + b - a. \tag{39.28}$$

We also define $KL(a, 0) = +\infty$ and $KL(0, b) = b$. Extending to nonnegative vectors $a = (a_1, ..., a_J)^T$ and $b = (b_1, ..., b_J)^T$, we have

$$KL(a, b) = \sum_{j=1}^{J} KL(a_j, b_j) = \sum_{j=1}^{J} \Big(a_j \log \frac{a_j}{b_j} + b_j - a_j \Big).$$

The MART is an iterative algorithm for finding a non-negative solution of the system $Px = y$, for an I by J matrix P with non-negative entries and vector y with positive entries. We also assume that

$$p_j = \sum_{i=1}^{I} P_{ij} > 0,$$

for all $i = 1, ..., I$. When discussing the MART, we say that the system $Px = y$ is *consistent* when it has non-negative solutions. We consider two different versions of the MART.

MART I

The iterative step of the first version of MART, which we shall call MART I, is the following: for $k = 0, 1, ...,$ and $i = k(\mathrm{mod}\, I) + 1$, let

$$x_j^{k+1} = x_j^k \Big(\frac{y_i}{(Px^k)_i} \Big)^{P_{ij}/m_i},$$

for $j = 1, ..., J$, where the parameter m_i is defined to be

$$m_i = \max\{P_{ij} | j = 1, ..., J\}.$$

The MART I algorithm converges, in the consistent case, to the non-negative solution for which the KL distance $KL(x, x^0)$ is minimized.

MART II

The iterative step of the second version of MART, which we shall call MART II, is the following: for $k = 0, 1, ...,$ and $i = k(\mathrm{mod}\, I) + 1$, let

$$x_j^{k+1} = x_j^k \Big(\frac{y_i}{(Px^k)_i} \Big)^{P_{ij}/p_j n_i},$$

for $j = 1, ..., J$, where the parameter n_i is defined to be

$$n_i = \max\{P_{ij} p_j^{-1} | j = 1, ..., J\}.$$

The MART II algorithm converges, in the consistent case, to the non-negative solution for which the KL distance

$$\sum_{j=1}^{J} p_j KL(x_j, x_j^0)$$

is minimized.

39.5.2 Using the MART to Solve the DGP Problem

The entries on the bottom row of A^T are all one, as is the bottom entry of the column vector u, since these entries correspond to the equation $\sum_{i=1}^{I} \delta_i = 1$. By adding suitably large positive multiples of this last equation to the other equations in the system, we obtain an equivalent system, $B^T\delta = s$, for which the new matrix B^T and the new vector s have only positive entries. Now we can apply the MART I algorithm to the system $B^T\delta = s$, letting $P = B^T$, $p_i = \sum_{j=1}^{J+1} B_{ij}$, $\delta = x$, $x^0 = c$ and $y = s$. In the consistent case, the MART I algorithm will find the non-negative solution that minimizes $KL(x, x^0)$, so we select $x^0 = c$. Then the MART I algorithm finds the non-negative δ^* satisfying $B^T\delta^* = s$, or, equivalently, $A^T\delta^* = u$, for which the KL distance

$$KL(\delta, c) = \sum_{i=1}^{I} \left(\delta_i \log \frac{\delta_i}{c_i} + c_i - \delta_i\right)$$

is minimized. Since we know that

$$\sum_{i=1}^{I} \delta_i = 1,$$

it follows that minimizing $KL(\delta, c)$ is equivalent to maximizing $v(\delta)$. Using δ^*, we find the optimal t^* solving the GP problem.

For example, the linear system of equations $A^T\delta = u$ corresponding to the posynomial in Equation (39.1) is

$$A^T\delta = u = \begin{bmatrix} -1 & 1 & 1 & 0 \\ -1 & 0 & 1 & 1 \\ -1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Adding two times the last row to the other rows, the system becomes

$$B^T\delta = s = \begin{bmatrix} 1 & 3 & 3 & 2 \\ 1 & 2 & 3 & 3 \\ 1 & 3 & 2 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 1 \end{bmatrix}.$$

The matrix B^T and the vector s are now positive. We are ready to apply the MART.

The MART iteration is as follows. With $j = k(\mathrm{mod}\,(J+1)) + 1$, $m_j = \max\{B_{ij} \,|i = 1, 2, ..., I\}$ and $k = 0, 1, ...$, let

$$\delta_i^{k+1} = \delta_i^k \left(\frac{s_j}{(B^T\delta^k)_j}\right)^{m_j^{-1}B_{ij}}.$$

The optimal δ^* is $\delta^* = (.4, .2, .2, .2)^T$, the optimal t^* is $t^* = (2, 1, .5)$, and the lowest cost is one hundred dollars.

39.6 Constrained Geometric Programming

Consider now the following variant of the problem of transporting the gravel across the river. Suppose that the bottom and the two sides will be constructed for free from scrap metal, but only four square yards are available. The cost function to be minimized becomes

$$g_0(t) = \frac{40}{t_1 t_2 t_3} + 40 t_2 t_3, \tag{39.29}$$

and the constraint is

$$g_1(t) = \frac{t_1 t_3}{2} + \frac{t_1 t_2}{4} \leq 1. \tag{39.30}$$

With $\delta_1 > 0$, $\delta_2 > 0$, and $\delta_1 + \delta_2 = 1$, we write

$$g_0(t) = \delta_1 \frac{40}{\delta_1 t_1 t_2 t_3} + \delta_2 \frac{40 t_2 t_3}{\delta_2}. \tag{39.31}$$

Since $0 \leq g_1(t) \leq 1$, we have

$$g_0(t) \geq \Big(\delta_1 \frac{40}{\delta_1 t_1 t_2 t_3} + \delta_2 \frac{40 t_2 t_3}{\delta_2}\Big)\Big(g_1(t)\Big)^{\lambda}, \tag{39.32}$$

for any positive λ. The GAGM Inequality then tells us that

$$g_0(t) \geq \left(\Big(\frac{40}{\delta_1 t_1 t_2 t_3}\Big)^{\delta_1}\Big(\frac{40 t_2 t_3}{\delta_2}\Big)^{\delta_2}\right)\Big(g_1(t)\Big)^{\lambda}, \tag{39.33}$$

so that

$$g_0(t) \geq \left(\Big(\frac{40}{\delta_1}\Big)^{\delta_1}\Big(\frac{40}{\delta_2}\Big)^{\delta_2}\right) t_1^{-\delta_1} t_2^{\delta_2 - \delta_1} t_3^{\delta_2 - \delta_1}\Big(g_1(t)\Big)^{\lambda}. \tag{39.34}$$

From the GAGM Inequality, we also know that, for $\delta_3 > 0$, $\delta_4 > 0$ and $\lambda = \delta_3 + \delta_4$,

$$\Big(g_1(t)\Big)^{\lambda} \geq (\lambda)^{\lambda}\left(\Big(\frac{1}{2\delta_3}\Big)^{\delta_3}\Big(\frac{1}{4\delta_4}\Big)^{\delta_4}\right) t_1^{\delta_3+\delta_4} t_2^{\delta_4} t_3^{\delta_3}. \tag{39.35}$$

Combining the inequalities in (39.34) and (39.35), we obtain

$$g_0(t) \geq v(\delta) t_1^{-\delta_1+\delta_3+\delta_4} t_2^{-\delta_1+\delta_2+\delta_4} t_3^{-\delta_1+\delta_2+\delta_3}, \tag{39.36}$$

with

$$v(\delta) = \Big(\frac{40}{\delta_1}\Big)^{\delta_1}\Big(\frac{40}{\delta_2}\Big)^{\delta_2}\Big(\frac{1}{2\delta_3}\Big)^{\delta_3}\Big(\frac{1}{4\delta_4}\Big)^{\delta_4}\Big(\delta_3 + \delta_4\Big)^{\delta_3+\delta_4}, \tag{39.37}$$

and $\delta = (\delta_1, \delta_2, \delta_3, \delta_4)$. If we can find a positive vector δ with

$$\begin{aligned} \delta_1 + \delta_2 &= 1, \\ \delta_3 + \delta_4 &= \lambda, \\ -\delta_1 + \delta_3 + \delta_4 &= 0, \\ -\delta_1 + \delta_2 + \delta_4 &= 0 \\ -\delta_1 + \delta_2 + \delta_3 &= 0, \end{aligned} \tag{39.38}$$

then

$$g_0(t) \geq v(\delta). \tag{39.39}$$

In this particular case, there is a unique positive δ satisfying the equations (39.38), namely

$$\delta_1^* = \frac{2}{3}, \delta_2^* = \frac{1}{3}, \delta_3^* = \frac{1}{3}, \text{and } \delta_4^* = \frac{1}{3}, \tag{39.40}$$

and

$$v(\delta^*) = 60. \tag{39.41}$$

Therefore, $g_0(t)$ is bounded below by 60. If there is t^* such that

$$g_0(t^*) = 60, \tag{39.42}$$

then we must have

$$g_1(t^*) = 1, \tag{39.43}$$

and equality in the GAGM Inequality. Consequently,

$$\frac{3}{2}\frac{40}{t_1^* t_2^* t_3^*} = 3(40 t_2^* t_3^*) = 60, \tag{39.44}$$

and

$$\frac{3}{2} t_1^* t_3^* = \frac{3}{4} t_1^* t_2^* = K. \tag{39.45}$$

Since $g_1(t^*) = 1$, we must have $K = \frac{3}{2}$. We solve these equations by taking logarithms, to obtain the solution

$$t_1^* = 2,\ t_2^* = 1, \text{ and } t_3^* = \frac{1}{2}. \tag{39.46}$$

The change of variables $t_j = e^{x_j}$ converts the constrained (GP) problem into a constrained convex programming problem. The theory of the constrained (GP) problem can then be obtained as a consequence of the theory for the convex programming problem.

39.7 Exercises

Exercise 39.1 *Show that there is no solution to the problem of minimizing the function*

$$g(t_1, t_2) = \frac{2}{t_1 t_2} + t_1 t_2 + t_1, \tag{39.47}$$

over $t_1 > 0$, $t_2 > 0$.

Exercise 39.2 *Minimize the function*

$$g(t_1, t_2) = \frac{1}{t_1 t_2} + t_1 t_2 + t_1 + t_2, \tag{39.48}$$

over $t_1 > 0$, $t_2 > 0$. *This will require some iterative numerical method for solving equations.*

Exercise 39.3 *Program the MART algorithm and use it to verify the assertions made previously concerning the solutions of the two numerical examples.*

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] Bertero, M. (1992) "Sampling theory, resolution limits and inversion methods."in [18], pp. 71–94.

[17] Bertero, M., and Boccacci, P. (1998) *Introduction to Inverse Problems in Imaging* Bristol, UK: Institute of Physics Publishing.

[18] Bertero, M. and Pike, E.R., editors (1992) *Inverse Problems in Scattering and Imaging* Malvern Physics Series, Adam Hilger, IOP Publishing, London.

[19] Bertsekas, D.P. (1997) "A new class of incremental gradient methods for least squares problems."*SIAM J. Optim.* **7**, pp. 913–926.

[20] Blackman, R. and Tukey, J. (1959) *The Measurement of Power Spectra*. New York: Dover Publications.

[21] 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.

[22] Bochner, S. and Chandrasekharan, K. (1949) *Fourier Transforms*, Annals of Mathematical Studies, No. 19. Princeton, NJ: Princeton University Press.

[23] Born, M. and Wolf, E. (1999) *Principles of Optics: 7th edition*. Cambridge, UK: Cambridge University Press.

[24] 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.

[25] Borwein, J. and Lewis, A. (2000) *Convex Analysis and Nonlinear Optimization*. Canadian Mathematical Society Books in Mathematics, New York: Springer-Verlag.

[26] Bracewell, R.C. (1979) Image Reconstruction in Radio Astronomy, in [148], pp. 81–104.

[27] 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.

[28] 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.

[29] Brooks, D., and MacLeod, R. (1997) "Electrical imaging of the heart." *IEEE Signal Processing Magazine*, **14 (1)**, pp. 24–42.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] Budinger, T., Gullberg, G., and Huesman, R. (1979) "Emission computed tomography." in [148], pp. 147–246.

[35] Burg, J. (1967) "Maximum entropy spectral analysis." *paper presented at the 37th Annual SEG meeting, Oklahoma City, OK.*

[36] Burg, J. (1972) "The relationship between maximum entropy spectra and maximum likelihood spectra." *Geophysics* **37**, pp. 375–376.

[37] Burg, J. (1975) *Maximum Entropy Spectral Analysis*, Ph.D. dissertation, Stanford University.

[38] 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.

[39] Byrne, C. and Fitzgerald, R. (1982) "Reconstruction from partial information, with applications to tomography."*SIAM J. Applied Math.* **42(4)**, pp. 933–940.

[40] 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.

[41] Byrne, C., and Wells, D. (1983) "Limit of continuous and discrete finite-band Gerchberg iterative spectrum extrapolation."*Optics Letters* **8 (10)**, pp. 526–527.

[42] 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.

[43] 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.

[44] 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.

[45] Byrne, C. and Fiddy, M. (1987) "Estimation of continuous object distributions from Fourier magnitude measurements."*JOSA A* **4**, pp. 412–417.

[46] Byrne, C. and Fiddy, M. (1988) "Images as power spectra; reconstruction as Wiener filter approximation."*Inverse Problems* **4**, pp. 399–409.

[47] Byrne, C., Haughton, D., and Jiang, T. (1993) "High-resolution inversion of the discrete Poisson and binomial transformations."*Inverse Problems* **9**, pp. 39–56.

[48] Byrne, C. (1993) "Iterative image reconstruction algorithms based on cross-entropy minimization."*IEEE Transactions on Image Processing* **IP-2**, pp. 96–103.

[49] 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.

[50] 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.

[51] Byrne, C. (1996) "Block-iterative methods for image reconstruction from projections."*IEEE Transactions on Image Processing* **IP-5**, pp. 792–794.

[52] Byrne, C. (1997) "Convergent block-iterative algorithms for image reconstruction from inconsistent data."*IEEE Transactions on Image Processing* **IP-6**, pp. 1296–1304.

[53] 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.

[54] Byrne, C. (1998) "Iterative deconvolution and deblurring with constraints." *Inverse Problems*, **14**, pp. 1455-1467.

[55] Byrne, C. (1999) "Iterative projection onto convex sets using multiple Bregman distances." *Inverse Problems* **15**, pp. 1295–1313.

[56] Byrne, C. (2000) "Block-iterative interior point optimization methods for image reconstruction from limited data."*Inverse Problems* **16**, pp. 1405–1419.

[57] 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.,

[58] Byrne, C. (2001) "Likelihood maximization for list-mode emission tomographic image reconstruction."*IEEE Transactions on Medical Imaging* **20(10)**, pp. 1084–1092.

[59] Byrne, C. (2002) "Iterative oblique projection onto convex sets and the split feasibility problem."*Inverse Problems* **18**, pp. 441–453.

[60] Byrne, C. (2004) "A unified treatment of some iterative algorithms in signal processing and image reconstruction."*Inverse Problems* **20**, pp. 103–120.

[61] Byrne, C. (2005) "Choosing parameters in block-iterative or ordered-subset reconstruction algorithms." *IEEE Transactions on Image Processing*, **14 (3)**, pp. 321–327.

[62] Byrne, C. (2005) *Signal Processing: A Mathematical Approach*, AK Peters, Publ., Wellesley, MA.

[63] Byrne, C. (2007) *Applied Iterative Methods*, AK Peters, Publ., Wellesley, MA.

[64] 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*, to appear.

[65] Byrne, C. (2009) *A First Course in Optimization*, unpublished text available at my website.

[66] 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.

[67] Candès, E., and Romberg, J. (2007) "Sparsity and incoherence in compressive sampling." *Inverse Problems*, **23(3)**, pp. 969–985.

[68] 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.

[69] 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 .

[70] Candy, J. (1988) *Signal Processing: The Modern Approach* New York: McGraw-Hill Publ.

[71] 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.

[72] Censor, Y. (1981) "Row-action methods for huge and sparse systems and their applications." *SIAM Review*, **23**: 444–464.

[73] 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.

[74] Censor, Y. and Elfving, T. (1994) "A multi-projection algorithm using Bregman projections in a product space." *Numerical Algorithms*, **8**, pp. 221–239.

[75] 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.

[76] 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.

[77] 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.

[78] 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.

[79] 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.

[80] Censor, Y., and Reich, S. (1996) "Iterations of paracontractions and firmly nonexpansive operators with applications to feasibility and optimization" , *Optimization*, **37**, pp. 323–339.

[81] Censor, Y., and Reich, S. (1998) "The Dykstra algorithm for Bregman projections." *Communications in Applied Analysis*, **2**, pp. 323–339.

[82] Censor, Y. and Segman, J. (1987) "On block-iterative maximization." *J. of Information and Optimization Sciences* **8**, pp. 275–291.

[83] Censor, Y. and Zenios, S.A. (1997) *Parallel Optimization: Theory, Algorithms and Applications*. New York: Oxford University Press.

[84] 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.

[85] Childers, D., editor (1978) *Modern Spectral Analysis*. New York:IEEE Press.

[86] Chui, C. and Chen, G. (1991) *Kalman Filtering*, second edition. Berlin: Springer-Verlag.

[87] Cimmino, G. (1938) "Calcolo approssimato per soluzioni dei sistemi di equazioni lineari." *La Ricerca Scientifica XVI, Series II, Anno IX* **1**, pp. 326–333.

[88] Combettes, P. (1993) "The foundations of set theoretic estimation." *Proceedings of the IEEE* **81 (2)**, pp. 182–208.

[89] Combettes, P. (1996) "The convex feasibility problem in image recovery." *Advances in Imaging and Electron Physics* **95**, pp. 155–270.

[90] Combettes, P. (2000) "Fejér monotonicity in convex optimization." in *Encyclopedia of Optimization*, C.A. Floudas and P. M. Pardalos, editors, Boston: Kluwer Publ.

[91] 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.

[92] Combettes, P., and Wajs, V. (2005) "Signal recovery by proximal forward-backward splitting." *Multi-scale Modeling and Simulation*, **4(4)**, pp. 1168–1200.

[93] Cooley, J. and Tukey, J. (1965) "An algorithm for the machine calculation of complex Fourier series." *Math. Comp.*, **19**, pp. 297–301.

[94] Csiszár, I. (1989) "A geometric interpretation of Darroch and Ratcliff's generalized iterative scaling." *The Annals of Statistics* **17 (3)**, pp. 1409–1413.

[95] 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.

[96] Csiszár, I. and Tusnády, G. (1984) "Information geometry and alternating minimization procedures." *Statistics and Decisions* **Supp. 1**, pp. 205–237.

[97] Dainty, J. C. and Fiddy, M. (1984) "The essential role of prior knowledge in phase retrieval." *Optica Acta* **31**, pp. 325–330.

[98] Darroch, J. and Ratcliff, D. (1972) "Generalized iterative scaling for log-linear models." *Annals of Mathematical Statistics* **43**, pp. 1470–1480.

[99] Dax, A. (1990) "The convergence of linear stationary iterative processes for solving singular unstructured systems of linear equations." *SIAM Review*, **32**, pp. 611–635.

[100] 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.

[101] 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.

[102] 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.

[103] 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)**.

[104] 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.

[105] 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.

[106] Dines, K., and Lyttle, R. (1979) "Computerized geophysical tomography." *Proc. IEEE*, **67**, pp. 1065–1073.

[107] Donoho, D. (2006) "Compressed sampling." *IEEE Transactions on Information Theory*, **52 (4)**. (download preprints at http://www.stat.stanford.edu/ donoho/Reports).

[108] Driscoll, P., and Fox, W. (1996) "Presenting the Kuhn-Tucker conditions using a geometric method." *The College Mathematics Journal*, **38 (1)**, pp. 101–108.

[109] Duda, R., Hart, P., and Stork, D. (2001) *Pattern Classification*, Wiley.

[110] Duffin, R., Peterson, E., and Zener, C. (1967) *Geometric Programming: Theory and Applications*. New York: Wiley.

[111] Dugundji, J. (1970) *Topology* Boston: Allyn and Bacon, Inc.

[112] Dykstra, R. (1983) "An algorithm for restricted least squares regression." *J. Amer. Statist. Assoc.*, **78 (384)**, pp. 837–842.

[113] 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.

[114] Elsner, L., Koltracht, L., and Neumann, M. (1992) "Convergence of sequential and asynchronous nonlinear paracontractions." *Numerische Mathematik*, **62**, pp. 305–319.

[115] Erdogan, H., and Fessler, J. (1999) "Fast monotonic algorithms for transmission tomography." *IEEE Transactions on Medical Imaging*, **18(9)**, pp. 801–814.

[116] Everitt, B. and Hand, D. (1981) *Finite Mixture Distributions* London: Chapman and Hall.

[117] Farkas, J. (1902) "Über die Theorie der einfachen Ungleichungen." *J. Reine Angew. Math.*, **124**, pp. 1–24.

[118] Farncombe, T. (2000) "Functional dynamic SPECT imaging using a single slow camera rotation." *Ph.D. thesis, Dept. of Physics, University of British Columbia.*

[119] 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.

[120] 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.

[121] Feynman, R., Leighton, R., and Sands, M. (1963) *The Feynman Lectures on Physics, Vol. 1.* Boston: Addison-Wesley.

[122] Fiddy, M. (1983) "The phase retrieval problem." in *Inverse Optics*, SPIE Proceedings 413 (A.J. Devaney, editor), pp. 176–181.

[123] Fiddy, M. (2008) *private communication.*

[124] Fienup, J. (1979) "Space object imaging through the turbulent atmosphere." *Optical Engineering* **18**, pp. 529–534.

[125] 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.

[126] Fleming, W. (1965) *Functions of Several Variables*, Addison-Wesley Publ., Reading, MA.

[127] Frieden, B. R. (1982) *Probability, Statistical Optics and Data Testing.* Berlin: Springer-Verlag.

[128] Gale, D. (1960) *The Theory of Linear Economic Models*. New York: McGraw-Hill.

[129] Gasquet, C. and Witomski, F. (1998) *Fourier Analysis and Applications*. Berlin: Springer-Verlag.

[130] Gelb, A., editor, (1974) *Applied Optimal Estimation*, written by the technical staff of The Analytic Sciences Corporation, MIT Press, Cambridge, MA.

[131] 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.

[132] Gerchberg, R. W. (1974) "Super-restoration through error energy reduction." *Optica Acta* **21**, pp. 709–720.

[133] 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.

[134] Goebel, K., and Reich, S. (1984) *Uniform Convexity, Hyperbolic Geometry, and Nonexpansive Mappings*, New York: Dekker.

[135] Golshtein, E., and Tretyakov, N. (1996) *Modified Lagrangians and Monotone Maps in Optimization*. New York: John Wiley and Sons, Inc.

[136] 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.

[137] Gordan, P. (1873) "Über die Auflösungen linearer Gleichungen mit reelen Coefficienten." *Math. Ann.*, **6**, pp. 23–28.

[138] 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.

[139] 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.

[140] Green, P. (1990) "Bayesian reconstructions from emission tomography data using a modified EM algorithm." *IEEE Transactions on Medical Imaging* **9**, pp. 84–93.

[141] 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.

[142] 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.

[143] Haacke, E., Brown, R., Thompson, M., and Venkatesan, R. (1999) *Magnetic Resonance Imaging*. New York: Wiley-Liss.

[144] Hager, W. (1988) *Applied Numerical Linear Algebra*, Englewood Cliffs, NJ: Prentice- Hall.

[145] 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.

[146] Haykin, S. (1985) *Array Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall.

[147] 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.

[148] Herman, G.T. (ed.) (1979) *Image Reconstruction from Projections*, Topics in Applied Physics, Vol. 32, Springer-Verlag, Berlin.

[149] Herman, G.T., and Natterer, F. (eds.) (1981) *Mathematical Aspects of Computerized Tomography*, Lecture Notes in Medical Informatics, Vol. 8, Springer-Verlag, Berlin.

[150] Herman, G.T., Censor, Y., Gordon, D., and Lewitt, R. (1985) "Comment." (on the paper [242]), *Journal of the American Statistical Association* **80**, pp. 22–25.

[151] Herman, G. T. (1999) *private communication*.

[152] Herman, G. T. and Meyer, L. (1993) "Algebraic reconstruction techniques can be made computationally efficient." *IEEE Transactions on Medical Imaging* **12**, pp. 600–609.

[153] Hildreth, C. (1957) "A quadratic programming procedure." *Naval Research Logistics Quarterly* **4**, pp. 79–85. Erratum, p. 361.

[154] Hogg, R. and Craig, A. (1978) *Introduction to Mathematical Statistics*, MacMillan, New York.

[155] 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.

[156] Hudson, M., Hutton, B., and Larkin, R. (1992) "Accelerated EM reconstruction using ordered subsets." *Journal of Nuclear Medicine*, **33**, p.960.

[157] 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.

[158] 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.

[159] 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.

[160] Jiang, M., and Wang, G. (2003) "Convergence studies on iterative algorithms for image reconstruction." *IEEE Transactions on Medical Imaging*, **22(5)**, pp. 569–579.

[161] Kaczmarz, S. (1937) "Angenäherte Auflösung von Systemen linearer Gleichungen." *Bulletin de l'Academie Polonaise des Sciences et Lettres* **A35**, pp. 355–357.

[162] Kak, A., and Slaney, M. (2001) *Principles of Computerized Tomographic Imaging*. SIAM, Philadelphia, PA.

[163] Kalman, R. (1960) "A new approach to linear filtering and prediction problems." *Trans. ASME, J. Basic Eng.* **82**, pp. 35–45.

[164] Katznelson, Y. (1983) *An Introduction to Harmonic Analysis*. New York: John Wiley and Sons, Inc.

[165] Kheifets, A. (2004) *private communication*.

[166] 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 [245], pp. 473–498.

[167] Koltracht, L., and Lancaster, P. (1990) "Constraining strategies for linear iterative processes." *IMA J. Numer. Anal.*, **10**, pp. 555–567.

[168] Körner, T. (1988) *Fourier Analysis*. Cambridge, UK: Cambridge University Press.

[169] Körner, T. (1996) *The Pleasures of Counting*. Cambridge, UK: Cambridge University Press.

[170] Kuhn, H., and Tucker, A. (eds.) (1956) *Linear Inequalities and Related Systems*. Annals of Mathematical Studies, No. 38. New Jersey: Princeton University Press.

[171] Kullback, S. and Leibler, R. (1951) "On information and sufficiency." *Annals of Mathematical Statistics* **22**, pp. 79–86.

[172] Landweber, L. (1951) "An iterative formula for Fredholm integral equations of the first kind." *Amer. J. of Math.* **73**, pp. 615–624.

[173] Lane, R. (1987) "Recovery of complex images from Fourier magnitude." *Optics Communications* **63(1)**, pp. 6–10.

[174] Lange, K. and Carson, R. (1984) "EM reconstruction algorithms for emission and transmission tomography." *Journal of Computer Assisted Tomography* **8**, pp. 306–316.

[175] 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.

[176] 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.

[177] 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.

[178] 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.

[179] 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.

[180] Lent, A. (1998) *private communication*.

[181] 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.

[182] 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.

[183] Luenberger, D. (1969) *Optimization by Vector Space Methods*. New York: John Wiley and Sons, Inc.

[184] Lustig, M., Donoho, D., and Pauly, J. (2008) *Magnetic Resonance in Medicine*, to appear.

[185] 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.

[186] Mann, W. (1953) "Mean value methods in iteration." *Proc. Amer. Math. Soc.* **4**, pp. 506–510.

[187] McLachlan, G.J. and Krishnan, T. (1997) *The EM Algorithm and Extensions*. New York: John Wiley and Sons, Inc.

[188] McVeigh, E., and Ozturk, C. (2001) "Imaging myocardial strain." *IEEE Signal Processing Magazine*, **18 (6)**, pp. 44–56.

[189] 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*.

[190] Meijering, E., Smal, I., and Danuser, G. (2006) "Tracking in molecular bioimaging." *IEEE Signal Processing Magazine*, **23 (3)**, pp. 46–53.

[191] Motzkin, T. and Schoenberg, I. (1954) "The relaxation method for linear inequalities." *Canadian Journal of Mathematics* **6**, pp. 393–404.

[192] Mumcuoglu, E., Leahy, R., and Cherry, S. (1996) "Bayesian reconstruction of PET images: Methodology and performance analysis." *Phys. Med. Biol.*, **41**, pp. 1777–1807.

[193] 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.

[194] Nash, S. and Sofer, A. (1996) *Linear and Nonlinear Programming*. New York: McGraw-Hill.

[195] Natterer, F. (1986) *Mathematics of Computed Tomography*. New York: John Wiley and Sons, Inc.

[196] Natterer, F., and Wübbeling, F. (2001) *Mathematical Methods in Image Reconstruction*. Philadelphia, PA: SIAM Publ.

[197] Ollinger, J., and Fessler, J. (1997) "Positron-emission tomography." *IEEE Signal Processing Magazine*, **14 (1)**, pp. 43–55.

[198] Oppenheim, A. and Schafer, R. (1975) *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall.

[199] Papoulis, A. (1975) "A new algorithm in spectral analysis and band-limited extrapolation." *IEEE Transactions on Circuits and Systems* **22**, pp. 735–742.

[200] Papoulis, A. (1977) *Signal Analysis*. New York: McGraw-Hill.

[201] 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.

[202] Paulraj, A., Roy, R., and Kailath, T. (1986) "A subspace rotation approach to signal parameter estimation." *Proceedings of the IEEE* **74**, pp. 1044–1045.

[203] Peressini, A., Sullivan, F., and Uhl, J. (1988) *The Mathematics of Nonlinear Programming*. Berlin: Springer-Verlag.

[204] Peters, T. (1981) "Resolution improvement to CT systems using aperture-function correction." in [149], pp. 241–251.

[205] 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.

[206] 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.

[207] Poggio, T. and Smale, S. (2003) "The mathematics of learning: dealing with data." *Notices of the American Mathematical Society* **50 (5)**, pp. 537–544.

[208] Priestley, M. B. (1981) *Spectral Analysis and Time Series*. Boston: Academic Press.

[209] 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.

[210] 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.

[211] Qian, H. (1990) "Inverse Poisson transformation and shot noise filtering." *Rev. Sci. Instrum.* **61**, pp. 2088–2091.

[212] Quistgaard, J. (1997) "Signal acquisition and processing in medical diagnostic ultrasound." *IEEE Signal processing Magazine*, **14 (1)**, pp. 67–74.

[213] Reich, S. (1979) "Weak convergence theorems for nonexpansive mappings in Banach spaces." *Journal of Mathematical Analysis and Applications*, **67**, pp. 274–276.

[214] Rockafellar, R. (1970) *Convex Analysis*. Princeton, NJ: Princeton University Press.

[215] 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.

[216] Sarder, P., and Nehorai, A. (2006) "Deconvolution methods for 3-D fluoresence microscopy images." *IEEE Signal Processing Magazine*, **23 (3)**, pp. 32–45.

[217] Saulnier, G., Blue, R., Newell, J., Isaacson, D., and Edic, P. (2001) "Electrical impedance tomography." *IEEE Signal Processing Magazine*, **18 (6)**, pp. 31–43.

[218] Schmidlin, P. (1972) "Iterative separation of sections in tomographic scintigrams." *Nucl. Med.* **15(1)**.

[219] Schmidt, R. (1981) "A signal subspace approach to multiple emitter location and spectral estimation." *PhD thesis, Stanford University.*

[220] 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.

[221] Shepp, L., and Vardi, Y. (1982) "Maximum likelihood reconstruction for emission tomography." *IEEE Transactions on Medical Imaging*, **MI-1**, pp. 113–122.

[222] 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.

[223] 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.

[224] Shieh, M., and Byrne, C. (2006) "Image reconstruction from limited Fourier data." *Journal of the Optical Society of America, A*, **23(11)**.

[225] Smith, C. Ray and Grandy, W.T., editors (1985) *Maximum-Entropy and Bayesian Methods in Inverse Problems*. Dordrecht: Reidel Publ.

[226] Smith, C. Ray and Erickson, G., editors (1987) *Maximum-Entropy and Bayesian Spectral Analysis and Estimation Problems*. Dordrecht: Reidel Publ.

[227] 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.

[228] 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.

[229] Stiemke, E. (1915) "Über positive Lösungen homogener linearer Gleichungen." *Math. Ann*, **76**, pp. 340–342.

[230] Strang, G. (1980) *Linear Algebra and its Applications*. New York: Academic Press.

[231] Tanabe, K. (1971) "Projection method for solving a singular system of linear equations and its applications." *Numer. Math.* **17**, pp. 203–214.

[232] Therrien, C. (1992) *Discrete Random Signals and Statistical Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall.

[233] Thévenaz, P., Blu, T., and Unser, M. (2000) "Interpolation revisited." *IEEE Transactions on Medical Imaging*, **19**, pp.739–758.

[234] 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.

[235] Tucker, A. (1956) "Dual systems of homogeneous linear relations." in [170], pp. 3–18.

[236] Twomey, S. (1996) *Introduction to the Mathematics of Inversion in Remote Sensing and Indirect Measurement*. New York: Dover Publ.

[237] 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.

[238] Unser, M. (1999) "Splines: A perfect fit for signal and image processing." *IEEE Signal Processing Magazine*, **16**, pp. 22–38.

[239] Van Trees, H. (1968) *Detection, Estimation and Modulation Theory*. New York: John Wiley and Sons, Inc.

[240] van der Sluis, A. (1969) "Condition numbers and equilibration of matrices." *Numer. Math.*, **14**, pp. 14–23.

[241] 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.

[242] 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.

[243] von Neumann, J., and Morgenstern, O. (1944) *Theory of Games and Economic Behavior*. New Jersey: Princeton University Press.

[244] 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.

[245] Wernick, M. and Aarsvold, J., editors (2004) *Emission Tomography: The Fundamentals of PET and SPECT*. San Diego: Elsevier Academic Press.

[246] Wiener, N. (1949) *Time Series*. Cambridge, MA: MIT Press.

[247] Wright, G.A. (1997) "Magnetic resonance imaging." *IEEE Signal Processing Magazine*, **14 (1)**, pp. 56–66.

[248] Wright, W., Pridham, R., and Kay, S. (1981) "Digital signal processing for sonar." *Proc. IEEE* **69**, pp. 1451–1506.

[249] Yang, Q. (2004) "The relaxed CQ algorithm solving the split feasibility problem." *Inverse Problems*, **20**, pp. 1261–1266.

[250] 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.

[251] Youla, D. (1978) "Generalized image restoration by the method of alternating projections." *IEEE Transactions on Circuits and Systems* **CAS-25 (9)**, pp. 694–702.

[252] 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.

[253] Young, R. (1980) *An Introduction to Nonharmonic Fourier Analysis*. Boston: Academic Press.

[254] Zhou, X., and Wong, S. (2006) "Informatics challenges of high-throughput microscopy." *IEEE Signal Processing Magazine*, **23 (3)**, pp. 63–72.

[255] 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