Signal Processing for Medical Imaging

Charles L. Byrne

December 14, 2006

Contents

Part I

Preliminaries

Chapter 1

Preface

The term *image* is used here to denote any single- or multi-dimensional representation of a distribution of interest. The term *signal processing* is also used broadly to denote the extraction of information from measured data, usually obtained through some mode of remote sensing. This is not a survey of the ever-growing field of medical imaging, nor is it a summary of the history of the subject. The emphasis here is on mathematical tools that feature prominently in medical imaging. Several areas of applications, such as transmission and emission tomography, magnetic-resonance imaging (MRI), and intensity-modulated radiation therapy, are described in some detail, both to illustrate the importance of mathematical tools such as the Fourier transform, iterative optimization and statistical parameter estimation, and to provide concrete examples of medical applications.

The reader interested in learning more about computerized tomography should consult the classical books by Kak and Slaney [129], Natterer [155], and those edited by Herman [118] and by Herman and Natterer [119]. More recent volumes, such as [156] and [189], should also be *required reading.*

Helpful introductory articles on emerging applications have appeared in recent issues of the IEEE Signal Processing Magazine, specifically the January 1997, November 2001, and May 2006 issues. The January 1997 issue, described as a *special issue on medical imaging modalities*, includes articles on electrical heart imaging [26], positron-emission tomography (PET) [157], MRI [191], and ultrasound [170]. Each of these topics was fairly well established by 1997. In contrast, the January 2001 issue, describing *emerging medical imaging technologies*, looks at such newer techniques as electromagnetic brain mapping [7], electrical impedance tomography [174], heart strain imaging [149], and diffuse optical tomography [19]. The latest issue, in May 2006, surveys the imaging being done now at the cellular and molecular level, with articles on fluoresence microscopy [173], molecular bioimaging [151], electron microscopy [98], cryo-electron tomography [143], and several other topics (see also [188, 198, 185, 197]).

Books on subjects such as tomographic imaging necessarily contain material on signal processing, but their treatment is often inadequate. The main reason for this, I believe, is that the concepts and problems of signal processing are best presented to students through the use of physical examples; often the best examples do not fall within the subject area of the book and the authors hesitate to include such apparently tangential material. In contrast, I have included in these notes what I consider to be the best real-world examples that illustrate the main ideas of signal processing, without regard to subject area. As a result, the reader will find extended discussions of solar radio-emission problems, sonar and radar imaging, ocean acoustic tomography, and the like.

These notes are designed to be used either for a one-semester course on signal processing in medical imaging, or a two-semester course that also includes an in-depth treatment of iterative reconstruction methods. The one-semester course should cover chapters 3 through 8 in detail, and touch on the highlights of chapters 9 through 15. The two-semester course should treat chapters 9 through 15 in greater detail. Topics from the appendices should be included as needed.

Many of the referenced articles authored or co-authored by me are available for downloading at http://faculty.uml.edu/cbyrne/cbyrne.html. If you find any typographical errors, please email me.

Chapter 2

Introduction

The Fourier transform and Fourier series play important roles in signal processing, and, therefore, in applications in which signal processing features prominently. These notions are often first encountered as convenient mathematical devices for simplifying the discussion of ordinary and partial differential equations. I feel that it was an advantage for me that my own introduction to the Fourier transform came in the context of sonar signal processing, rather than in a more purely mathematical context. Consequently, I begin with a discussion of farfield propagation, which I believe to be the best way to introduce the Fourier transform and demonstrate its physical significance.

Convolution filtering and point-spread functions arise naturally as we discuss blurring and the effects of limited aperture on resolution. Nyquist spacing of sensors is then related to the accurate imaging of a distant object of finite extent. Dirac-delta models are introduced as limits of finite-extent objects and are used to facilitate direction-of-arrival array processing in sonar and radar.

The example of localizing the sources of solar radio emissions is used to introduce the problem of resolving point sources with limited-aperture antennas, and to provide a transition to the subject of reconstructing an image from line-integral data.

The Fourier transform arises explicitly in the discussion of farfield propagation, but occurs, somewhat less obviously, in reconstruction from line-integral data, through the Central Slice Theorem. These notes include a variety of examples leading to such line-integral data, along with several methods for reconstructing the object of interest from such data.

Once we have motivated the Fourier transform and revealed its several roles in tomographic imaging, we turn to the central concepts of Fourier methods. Following the tradition, I have chosen to discuss these notions within the context of discrete infinite sequences, although functions of continuous variables seem more physically realistic and vectors of finite data more accurately describe what we obtain through measurements. What we have, then, is a fairly standard treatment of discrete signal processing. I do dwell longer than most on the issue of transition between functions of continuous variables and discrete sequences, and between vectors of finite data and discrete sequences, mainly because these transitions can be the source of much confusion.

In signal processing, the data obtained through measurements usually contains a component called *signal* that contains the information we seek, and an unwanted component called *noise*. In order to process the data in a manner that respects the presence of the noise, we adopt a mathematical model of noise called a *random variable*. An infinite sequence of such random variables is a *random process*; a vector whose entries are random variables is a *random vector*. Our prior information about the noise will be in terms of correlations.

In most applications of remote sensing, the object of interest is a two- or three-dimensional distribution of something, such as acoustic energy in sonar, or radionuclide in positron-emission tomography (PET). Such distributions are usually reconstructed as images. It is important to recall that the production of a nice image is rarely the ultimate goal; in PET making a correct medical diagnosis using the images is the objective. For such purposes, estimation, detection, discrimination and pattern recognition also play important roles.

To provide some focus for our discussions, we will emphasize those mathematical notions and algorithms that play significant roles in transmission and emission tomography, magnetic-resonance imaging and, to a lesser extent, radiation therapy. We pay particular attention to Fourier-transform estimation, and iterative optimization algorithms, and solving large systems of linear equations, often with side constraints.

The several appendices deal with important, but somewhat specialized, topics, such as the Fast Fourier Transform (FFT), complex exponential functions, imaging in a Hilbert-space context, sensitivity to noise and the use of regularization methods, and the optimization of convex functions.

Part II

Signal Processing Fundamentals

Chapter 3

Far-field Propagation and the Fourier Transform

The Fourier transform (FT), in both one and two dimensions, will play a prominent role in our discussions. One often has the impression that the FT is introduced into discussions primarily for mathematical convenience. It is our goal, in this chapter, to convince the reader that the FT arises naturally in remote-sensing applications. To illustrate this natural occurrence of the FT, we study the signals received in the far-field from an array of transmitters or reflectors. We restrict our attention to single-frequency, or narrowband, signals.

3.1 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, far-field sonar or radar, and the like. Both the transmission and remote-sensing problems illustrate the important role played by the Fourier transform.

3.2 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 far-field 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 far-field 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

$$B(P,t) = e^{i\omega t}e^{-i\omega D/c}\int_{-A}^{A} f(x)e^{ix\frac{\omega\cos\theta}{c}}dx.$$

The integral term, which gives the far-field pattern of the transmission, is

$$F(\frac{\omega\cos\theta}{c}) = \int_{-A}^{A} f(x)e^{ix\frac{\omega\cos\theta}{c}}dx,$$

where $F(\gamma)$ is the Fourier transform of $f(x)$, given by

$$F(\gamma) = \int_{-A}^{A} f(x)e^{ix\gamma}dx.$$

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 Chapter 12 of [56].

Consider, for example, the function $f(x) = 1$, for $|x| \leq A$, and $f(x) = 0$, otherwise. The Fourier transform of $f(x)$ is

$$F(\gamma) = 2A\text{sinc}(A\gamma),$$

where sinc(t) is defined to be

$$\mathrm{sinc}(t) = \frac{\sin(t)}{t},$$

for $t \neq 0$, and $\mathrm{sinc}(0) = 1$. 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.

3.3 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 Fourier transform of the function describing the amplitude of the signal sent from each point P.

3.4 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 *far-field*, 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.

3.5 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

$$u_{tt} = c^2(u_{xx} + u_{yy} + u_{zz}) = c^2\nabla^2 u,$$

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

$$u(t, x, y, z) = f(t)g(x, y, z).$$

Inserting this separated form into the wave equation, we get

$$f''(t)g(x, y, z) = c^2 f(t)\nabla^2 g(x, y, z)$$

or

$$f''(t)/f(t) = c^2\nabla^2 g(x, y, z)/g(x, y, z).$$

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,

$$f''(t) + \omega^2 f(t) = 0, \tag{3.1}$$

and

$$\nabla^2 g(x, y, z) + \frac{\omega^2}{c^2} g(x, y, z) = 0. \tag{3.2}$$

Equation (3.2) is the *Helmholtz equation.*

Equation (3.1) 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.

3.6 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*.

Exercise 3.1 *Let* $\mathbf{s} = (x, y, z)$ *and* $u(\mathbf{s}, t) = u(x, y, z, t) = e^{i\omega t}e^{i\mathbf{k}\cdot\mathbf{s}}$. *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.

3.7 Superposition and the Fourier Transform

In the continuous superposition model, the field is

$$u(\mathbf{s}, t) = e^{i\omega t}\int F(\mathbf{k})e^{i\mathbf{k}\cdot\mathbf{s}}d\mathbf{k}.$$

Our measurements at the sensor locations $\mathbf{s}$ give us the values

$$f(\mathbf{s}) = \int F(\mathbf{k})e^{i\mathbf{k}\cdot\mathbf{s}}d\mathbf{k}. \tag{3.3}$$

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, at least in theory, 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 $\mathbf{k}$.

3.7.1 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 $\mathbf{k}$.

3.8 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.

3.8.1 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 (3.3) is

$$\mathbf{s} \cdot \mathbf{k} = xk_1 + yk_2;$$

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.

3.8.2 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 (3.3) becomes

$$\mathbf{s} \cdot \mathbf{k} = xk_1.$$

Now the ambiguity is greater than in the planar array case. Once we have k_1, we know that

$$k_2^2 + k_3^2 = (\frac{\omega}{c})^2 - k_1^2,$$

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.

3.8.3 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 $F(\mathbf{k})$.

3.9 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.

3.9.1 The Solar-Emission Problem

In [23] 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.

3.10 Sampling

In the one-dimensional case, the signal received at the point $(x, 0, 0)$ is essentially the 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

$$\Delta_x = \frac{\pi c}{\omega}.$$

The wavelength λ associated with the frequency ω is defined to be

$$\lambda = \frac{2\pi c}{\omega},$$

so that

$$\Delta_x = \frac{\lambda}{2}.$$

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.

3.11 The Limited-Aperture Problem

In the remote-sensing problem, our measurements at points $(x, 0, 0)$ in the far-field 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, as described in the Appendix on the Fourier transform, 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 far-field 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.

3.12 Resolution

The Dirac delta plays an important role in any discussion of resolution of point sources; for details, see the Appendix on the Fourier transform. 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 $H(\gamma) = 2A\text{sinc}(A\gamma)$. 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

$$R(k) = 2A\Big(\text{sinc}(A(k-a)) + \text{sinc}(A(k-b))\Big).$$

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

$$R(k) = H(k-a) = \text{sinc}(A(k-a))$$

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.

3.12.1 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

$$\frac{\pi}{A} \leq 3 \cdot 10^{-3},$$

or

$$A \geq \frac{\pi}{3} \cdot 10^{3}$$

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

$$\int f(x)dx = F(0);$$

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

$$\int f(x)\delta(x)dx = f(0) = \int F(k)dk;$$

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.

3.13 Discrete Data

A familiar topic in signal processing is the passage from functions of continuous variables to discrete sequences. This transition is achieved by *sampling*, that is, extracting values of the continuous-variable function at discrete points in its domain. Our example of far-field 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 $[-\frac{\omega}{c}, \frac{\omega}{c}]$.

Exercise 3.2 *Show that 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*

$$\frac{\omega}{c}\cos\alpha = \frac{\omega}{c}\cos\theta + \frac{2\pi}{\Delta}m,$$

where m is an integer.

To avoid the ambiguity described in Exercise 3.2, we must select $\Delta > 0$ so that

$$-\frac{\omega}{c} + \frac{2\pi}{\Delta} \geq \frac{\omega}{c},$$

or

$$\Delta \leq \frac{\pi c}{\omega} = \frac{\lambda}{2}.$$

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

$$\Delta_s = \frac{\pi}{K}.$$

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 (see [56]).

3.13.1 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 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.

3.14 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.

3.15 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 far-field propagation model. As we noted earlier, the solar emission problem is inherently a two-dimensional problem.

3.15.1 Two-Dimensional Far-field 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).

3.15.2 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

$$H(\alpha, \beta) = 4AB\text{sinc}(A\alpha)\text{sinc}(B\beta).$$

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.

3.16 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.

3.17 The Laplace Transform and the Ozone Layer

In the far-field propagation examples just considered, we found the measured data to be related to the desired object function by a Fourier transformation. The image reconstruction problem then became one of estimating a function from finitely many noisy values of its Fourier transform. In this section we consider an inverse problem involving the Laplace transform. The example is taken from Twomey's book [184].

3.17.1 The Laplace Transform

The Laplace transform of the function $f(x)$ defined for $0 \leq x < +\infty$ is the function

$$\mathcal{F}(s) = \int_0^{+\infty} f(x) e^{-sx} dx.$$

3.17.2 Scattering of Ultraviolet Radiation

The sun emits ultraviolet (UV) radiation that enters the Earth's atmosphere at an angle θ_0 that depends on the sun's position, and with intensity $I(0)$. Let the x-axis be vertical, with $x = 0$ at the top of the atmosphere and x increasing as we move down to the Earth's surface, at $x = X$. The intensity at x is given by

$$I(x) = I(0) e^{-kx/\cos\theta_0}.$$

Within the ozone layer, the amount of UV radiation scattered in the direction θ is given by

$$S(\theta, \theta_0) I(0) e^{-kx/\cos\theta_0} \Delta p,$$

where $S(\theta, \theta_0)$ is a known parameter, and Δp is the change in the pressure of the ozone within the infinitesimal layer $[x, x+\Delta x]$, and so is proportional to the concentration of ozone within that layer.

3.17.3 Measuring the Scattered Intensity

The radiation scattered at the angle θ then travels to the ground, a distance of $X - x$, weakened along the way, and reaches the ground with intensity

$$S(\theta, \theta_0) I(0) e^{-kx/\cos\theta_0} e^{-k(X-x)/\cos\theta} \Delta p.$$

The total scattered intensity at angle θ is then a superposition of the intensities due to scattering at each of the thin layers, and is then

$$S(\theta, \theta_0) I(0) e^{-kX/\cos\theta_0} \int_0^X e^{-x\beta} dp,$$

where

$$\beta = k[\frac{1}{\cos\theta_0} - \frac{1}{\cos\theta}].$$

This superposition of intensity can then be written as

$$S(\theta, \theta_0) I(0) e^{-kX/\cos\theta_0} \int_0^X e^{-x\beta} p'(x) dx.$$

3.17.4 The Laplace Transform Data

Using integration by parts, we get

$$\int_0^X e^{-x\beta} p'(x)dx = p(X)e^{-\beta X} - p(0) + \beta \int_0^X e^{-\beta x} p(x)dx.$$

Since $p(0) = 0$ and $p(X)$ can be measured, our data is then the Laplace transform value

$$\int_0^{+\infty} e^{-\beta x} p(x)dx;$$

note that we can replace the upper limit X with $+\infty$ if we extend $p(x)$ as zero beyond $x = X$.

The variable β depends on the two angles θ and θ_0. We can alter θ as we measure and θ_0 changes as the sun moves relative to the earth. In this way we get values of the Laplace transform of $p(x)$ for various values of β. The problem then is to recover $p(x)$ from these values. Because the Laplace transform involves a smoothing of the function $p(x)$, recovering $p(x)$ from its Laplace transform is more ill-conditioned than is the Fourier transform inversion problem.

3.18 The Laplace Transform and Energy Spectral Estimation

In x-ray transmission tomography, x-ray beams are sent through the object and the drop in intensity is measured. These measurements are then used to estimate the distribution of attenuating material within the object. A typical x-ray beam contains components with different energy levels. Because components at different energy levels will be attenuated differently, it is important to know the relative contribution of each energy level to the entering beam. The energy spectrum is the function $f(E)$ that describes the intensity of the components at each energy level $E > 0$.

3.18.1 The attenuation coefficient function

Each specific material, say aluminum, for example, is associated with attenuation coefficients, which is a function of energy, which we shall denote by $\mu(E)$. A beam with the single energy E passing through a thickness x of the material will be weakened by the factor $e^{-\mu(E)x}$. By passing the beam through various thicknesses x of aluminum and registering the intensity drops, one obtains values of the absorption function

$$R(x) = \int_0^\infty f(E)e^{-\mu(E)x}dE. \tag{3.4}$$

Using a change of variable, we can write $R(x)$ as a Laplace transform.

3.18.2 The absorption function as a Laplace transform

For each material, the attenuation function $\mu(E)$ is a strictly decreasing function of E, so $\mu(E)$ has an inverse, which we denote by g; that is, $g(t) = E$, for $t = \mu(E)$. Equation (3.4) can then be rewritten as

$$R(x) = \int_0^\infty f(g(t))e^{-tx}g'(t)dt. \tag{3.5}$$

We see then that $R(x)$ is the Laplace transform of the function $r(t) = f(g(t))g'(t)$. Our measurements of the intensity drops provide values of $R(x)$, for various values of x, from which we must estimate the functions $r(t)$, and, ultimately, $f(E)$.

Chapter 4

Reconstruction from Line-Integral Data

In many tomographic reconstruction problems, the data we have are not Fourier transform values, but are reasonably well modeled as line integrals associated with the function of interest. However, such data can, in principle, be used to obtain Fourier transform values, so that reconstruction can be achieved by Fourier inversion. For reasons that we shall explore, this approach is not usually practical. However, it does suggest approximate solution methods, involving convolution filtering and backprojection, that lead to useful algorithms.

We saw earlier that the solar emission problem was solved by formulating it as a problem of reconstruction from line-integral data. We begin here with several other signal-processing problems that require reconstruction of a function from its line integrals, including ocean acoustic tomography, x-ray transmission tomography, and positron- and single-photon emission tomography. Then we establish the connection between the tomography problem and Fourier-transform inversion. Finally, we consider several approaches to Fourier inversion that lead to practical algorithms.

4.1 Ocean Acoustic Tomography

Sound travels in the ocean at approximately $c = 1500$ mps, with deviations from this figure due to water temperature, depth at which the sound is traveling, salinity of the water, and so on. If c is constant, sound emitted at point A at time t will reach point B at time $t + d/c$, where d is the distance from A to B. If we know d and measure the delay in receiving the signal, we can find c. The sound speed is not truly constant, however, but is a function $c(x, y, z)$ of position. In fact, it may depend on time, as well, due, for example, to changing seasons of the year; because temporal changes are much slower to occur, we usually ignore time-dependence. Determining the spatial sound-speed profile, the function $c(x, y, z)$, is the objective of ocean acoustic tomography.

4.1.1 Obtaining Line-Integral Data

Since the sound speed is not constant, the sound traveling from point A to point B can now take a curved path; the shortest-time route may not be the shortest-distance route. To keep things from getting too complicated in this example, we consider the situation in which the sound still moves from A to B along the straight line segment joining them, but does not travel at a constant speed. We parameterize this line segment with the variable s, with $s = 0$ corresponding to the point A and $s = d$ the point B. We denote by $c(s)$ the sound speed at the point along the line having parameter value s. The time required for the sound to travel from s to $s+\Delta s$ is approximately $\Delta t = \frac{\Delta s}{c(s)}$, so that the signal reaches point B after a delay of $\int_0^d \frac{1}{c(s)} ds$ seconds. Ocean acoustic tomography has as its goal the estimation of the sound speed profile $c(x, y, z)$ from finitely many such line integrals. Because the sound speed is closely related to ocean temperature, ocean acoustic tomography has important applications in weather prediction, as well as in sonar imaging and active and passive sonar detection and surveillance.

4.1.2 The Difficulties

Now let's consider the various obstacles that we face as we try to solve this problem. First of all, we need to design a signal to be transmitted. It must be one from which we can easily and unambiguously determine the delays. When the delayed signal is received, it will not be the only sound in the ocean and must be clearly distinguished from the acoustic background. The processing of the received signals will be performed digitally, which means that we will have to convert the analog functions of the continuous time variable into discrete samples. These vectors of discrete samples will then be processed mathematically to obtain estimates of the line integrals. Once we have determined the line integrals, we must estimate the function $c(x, y, z)$ from these line integrals. We will know the line integrals only approximately and will have only finitely many of them, so the best we can hope to do is to approximate the function $c(x, y, z)$. How well we do will depend on which pairs of sources and receivers we have chosen to use. On the bright side, we have good prior information about the behavior of the sound speed in the ocean, and can specify *a priori* upper and lower bounds on the possible deviations from the nominal speed of 1500 mps. Even so, we need good algorithms that incorporate our prior information.

As we shall see later, the Fourier transform will provide an important tool for solving these problems.

4.1.3 Why "Tomography"?

Although the sound-speed profile $c(x, y, z)$ is a function of the three spatial variables, accurate reconstruction of such a three-dimensional function from line integrals would require a large number of lines. In ocean acoustic tomography, as well as in other applications, such as x-ray transmission tomography, the three-dimensional object of interest is studied one slice at a time, so that the function is reduced to a two-dimensional distribution. In fact, the term *tomography*, coming as it does from the Greek word for *part* or *slice*, and thereby related to the word *atom* ("no parts"), is used to describe such problems, because of the early emphasis placed on computationally tractable slice-by-slice reconstruction.

4.1.4 An Algebraic Approach

There is a more algebraic way to reconstruct a function from line integrals. Suppose that we transmit our signal from points A_i, $i = 1, ..., I$ and receive them at points B_j, $j = 1, ..., J$. Then we have $N = IJ$ transmitter-receiver pairs, so we have N line integrals, corresponding to N line segments, which we denote L_n, $n = 1, ..., N$. Imagine the part of the ocean involved to be discretized into M cubes or *voxels*, or, in the slice-by slice approach, two-dimensional squares, or *pixels*, and suppose that within the mth voxel the sound speed is equal to c_m; also let $x_m = 1/c_m$. For each line segment L_n let P_{nm} be the length of the intersection of line segment L_n with the mth voxel. The time it takes for the acoustic signal to traverse line segment L_n is then approximately

$$(P\mathbf{x})_n = \sum_{m=1}^{M} P_{nm} x_m,$$

where P denotes the matrix with entries P_{nm} and $\mathbf{x}$ denotes the vector with entries x_m. Our problem now is to solve the system of linear equations $P\mathbf{x} = \mathbf{t}$, where the entries of the vector $\mathbf{t}$ are the travel times we have measured for each line segment. This system can be solved by any number of well known algorithms. Notice that the entries of P, $\mathbf{x}$ and $\mathbf{t}$ are all nonnegative. This suggests that algorithms designed specifically to deal with nonnegative problems may work better. In many cases, both M and N are large, making some algorithms, such as Gauss elimination, impractical, and iterative algorithms competitive.

Although we have presented tomography within the context of ocean acoustics, most of what we have discussed in this section carries over, nearly unchanged, to a number of medical imaging problems.

4.2 X-ray Transmission Tomography

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

4.2.1 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 4.1 *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. As in the case of ocean acoustic tomography, 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.

4.2.2 Difficulties to be Overcome

Once again, there are hurdles to be overcome. 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. 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.

The mathematical similarities between x-ray transmission tomography and ocean acoustic tomography suggest that the reconstruction algorithms used will be similar, and this is the case. As we shall see later, the Fourier transform and the associated theory of convolution filters play important roles.

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.

4.3 Positron Emission Tomography

In emission tomography (ET), which includes positron emission tomography (PET) and single photon emission tomography (SPECT), the patient inhales, or is injected with, chemicals to which radioactive material has been chemically attached [189]. 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 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.

4.3.1 The Coincidence-Detection Model

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.

4.3.2 Line-Integral Data

Let the LOR be parameterized by the variable s, with $s = 0$ and $s = L$ denoting the two ends, and L 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 $s = L$ and be detected is then

$$P(L) = e^{-\int_{s_0}^{L} \mu(t)dt}.$$

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 $s = L$ are independent events, their probabilities multiply, so that the probability of a coincident detection along the LOR, due to an emission at s_0, is

$$e^{-\int_0^L \mu(t)dt}.$$

The expected number of coincident detections along the LOR is then proportional to

$$\int_0^L f(s)e^{-\int_0^L \mu(t)dt}ds = e^{-\int_0^L \mu(t)dt}\int_0^L f(s)ds,$$

where $f(s)$ is the intensity of radionuclide at s. Assuming we know the attenuation function $\mu(s)$, we can estimate the line integral $\int_0^L f(s)ds$ from the number of coincident detections recorded along the LOR. So, once again, we have line-integral data pertaining to the function of interest.

4.4 Single-Photon Emission Tomography

Single-photon 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. Lead collimators are used 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.

4.4.1 The Line-Integral Model

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 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 backprojection* (FBP) algorithm is a commonly used method for medical imaging in clinical settings.

4.4.2 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 backprojection 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 [132]: attenuation, scatter, and spatially dependent resolution.

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 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.

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 $s = L$ and be detected is then

$$P(s_0) = e^{-\int_{s_0}^{L} \mu(t)dt}.$$

If $f(s)$ is the expected number of photons emitted from point s during the scanning, then the expected number of photons detected at L is proportional to

$$\int_0^L f(s) e^{-\int_s^L \mu(t)dt} ds.$$

This quantity 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.*

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 we think they come from. The scattered photons often have reduced energy, compared to *primary*, or unscattered, photons, and scatter-correction can be based on this energy difference; see [132].

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. There are mathematical ways to correct for both spatially varying resolution and uniform attenuation [179]. Correcting for the more realistic non-uniform and patient-specific attenuation is more difficult and is the subject of on-going research.

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 [189]). 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 [165].

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.

4.4.3 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. 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 $\mathbf{y}$.

To achieve our goal we must construct a model that relates $\mathbf{y}$ to $\mathbf{x}$. The standard way to do this 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 modelling radioactivity, we also assume that the Z_{ij} are Poisson-distributed.

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. 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.

4.4.4 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.

4.5 Reconstruction from Line Integrals

As we have just seen, a wide variety of applications involve the determination of a function of several variables from knowledge of line integrals of that function. We turn now to the underlying problem of reconstructing such functions from line-integral data.

4.5.1 The Radon Transform

Our goal is to reconstruct the function $f(x, y)$ 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. 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

$$\int f(x, y)ds = \int f(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta)ds$$

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.

4.5.2 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.

The Fourier-transform inversion formula for two-dimensional functions tells us that the function $f(x,y)$ can be obtained as

$$f(x,y)=\frac{1}{4\pi^2}\int\int F(u,v)e^{-i(xu+yv)}dudv. \tag{4.1}$$

We now derive alternative inversion formulas.

4.5.3 Ramp Filter, then Backproject

Expressing the double integral in Equation (4.1) 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

$$g_f(\theta, x\cos\theta + y\sin\theta) = \frac{1}{2\pi}\int_{-\infty}^{\infty} R_f(\theta,\omega)|\omega|e^{-i\omega t}d\omega \tag{4.2}$$

is the result of a linear filtering of $r_f(\theta,t)$ using a *ramp filter* with transfer function $H(\omega) = |\omega|$. Then,

$$f(x,y) = \frac{1}{2\pi}\int_0^{\pi} g_f(\theta, x\cos\theta + y\sin\theta)d\theta \tag{4.3}$$

gives $f(x,y)$ as the result of a *backprojection 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 backprojection* (FBP) of the line-integral data.

Knowing that $f(x,y)$ is related to the complete set of line integrals by filtered backprojection suggests that, when only finitely many line integrals are available, a similar ramp filtering and backprojection can be used to estimate $f(x,y)$; in the clinic this is the most widely used method for the reconstruction of tomographic images.

4.5.4 Backproject, then Ramp Filter

There is a second way to recover $f(x,y)$ using backprojection and filtering, this time in the reverse order; that is, we backproject 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

$$f(x,y) = \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{\infty} \frac{F(u,v)}{\sqrt{u^2+v^2}}\sqrt{u^2+v^2}e^{-i(xu+yv)}\omega d\omega d\theta$$

$$= \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{\infty} G(u,v)\sqrt{u^2+v^2}e^{-i(xu+yv)}\omega d\omega d\theta, \tag{4.4}$$

using

$$G(u,v) = \frac{F(u,v)}{\sqrt{u^2+v^2}}$$

for $(u,v) \neq (0,0)$. Equation (4.4) 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 backprojection of the function $r_f(\omega,t)$; 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.

4.5.5 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 (4.2) 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

$$f(x,y) = \frac{1}{2\pi} \int_0^\pi HT(\frac{\partial}{\partial t} r_f(\theta, t))d\theta.$$

4.5.6 Practical Issues

Of course, 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 varable 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 backprojection 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.

4.6 Summary

We have seen how the problem of reconstructing a function from line integrals arises in a number of applications. The Central Slice Theorem connects the line integrals and the Radon transform to the Fourier transform of the desired distribution. Various approaches to implementing the Fourier Inversion Formula lead to filtered backprojection algorithms for the reconstruction. In x-ray tomography and 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 5

Discrete Signal Processing

Although we usually model real-world distributions as functions of continuous variables, while the data we actually obtain are finite, it is standard practice to develop signal processing fundamentals within the context of infinite sequences, or functions of discrete variables. Infinite sequences arise when we sample functions of continuous variables, or when we extend finite data. Within the context of discrete signal processing, Fourier series replace Fourier transforms as the key mathematical tool. The Shannon sampling theorem provides the link between these two branches of Fourier analysis.

5.1 Discrete Signals

A discrete signal is a function $x = \{x(n)\}$ defined for all integers n. In signal processing, such discrete signals are often the result of *sampling* a function of a continuous variable. In our discussion of farfield propagation, we saw that the data gathered at each sensor effected a sampling of the Fourier transform, $F(\gamma)$, of the distant distribution $f(x)$. In the theoretical situation in which we had available an infinite discrete set of sensors, we would have an infinite sequence, obtained by sampling the function $F(\gamma)$. In many applications, the function that is being sampled is a function of time, say $f(t)$; we shall use this example in our discussion here.

In the most common case, that of equispaced sampling, we have $x(n) = f(n\Delta)$, where $\Delta > 0$ is the sampling interval. Generally, such discrete signals are neither a realistic model of the physical situation nor an accurate description of what we have actually obtained through measurement. Nevertheless, discrete signals provide the most convenient framework within which to study the the basic tools of signal processing coming from Fourier analysis.

5.2 Notation

It is common practice to denote functions of a discrete variable by the letters x, y or z, as well as f, g or h. So we speak of the discrete signals $x = \{x(n) = 2n - 1, -\infty < n < \infty\}$ or $y = \{y(n) = -n^3 + n, -\infty < n < \infty\}$. For convenience, we often just say $x(n) = 2n - 1$ or $y(n) = n^3 + n$ when we mean the whole function x or y. However, if k is regarded as a fixed, but unspecified, integer, $x(k)$ means the value of the function x at k. This is really the same thing that we do in calculus, when we define a function $f(x) = ax^2 + bx + c$; the x is a variable, while the a, b, and c are parameters that do not change during the discussion of this function. Now n is a variable, while k is a parameter.

There are two special discrete signals with *reserved names*, δ and u: $\delta(0) = 1$ and $\delta(n) = 0$, for $n \neq 0$; $u(n) = 1$, for $n \geq 0$ and $u(n) = 0$ for $n < 0$. When we say that their names are reserved we mean that whenever you see these names you can (usually) assume that they refer to the same functions as just defined; in calculus e^x and $\sin x$ are reserved names, while in signal processing δ and u are reserved names.

5.3 Operations on Discrete Signals

Because discrete signals are functions, we can perform on them many of the operations we perform on functions of a continuous variable. For instance, we can add discrete signals x and y, to get the discrete signal $x+y$, we can multiply x by a real number c to get the discrete signal cx, we can multiply x and y to get xy, and so on. We can *shift* x to the right k units to get y with $y(n) = x(n-k)$. Notice that, if we shift $x = \delta$ to the right k units, we have y with $y(k) = 1$ and $y(n) = 0$ for $n \neq k$; we call this function δ_k, so we sometimes say that $\delta = \delta_0$.

In general, an operation, or, to use the official word, an *operator*, T works on a discrete signal x to produce another discrete signal y; we describe this situation by writing $y = T(x)$. For example, the operator $T = S_k$ shifts any x to the right by k units; for example, $S_3(\delta) = \delta_3$. We are particularly interested in operators that possess certain nice properties.

5.3.1 Linear Operators

An operator T is called *linear* if, for any x and z and numbers a and b we have $T(ax + bz) = aT(x) + bT(z)$; for example, the operator $T = S_k$ is linear.

Exercise 5.1 *Which of the following operators are linear?*

a. $T(x)(n) = x(n-1) + x(n)$*;*

b. $T(x)(n) = nx(n)$;

c. $T(x)(n) = x(n)^2$.

5.3.2 Shift-invariant Operators

Notice that operators are also functions, although not the sort that we usually study; their domains and ranges consist of functions. We have seen such operator-type functions in calculus class- the operator that transforms a function into its derivative is an operator-type function. Therefore we can combine operators using composition, in the same way we compose functions. The composition of operators T and S is the operator that first performs S and then performs T on the result; that is, the composition of T and S begins with x and ends with $y = T(S(x))$. Notice that, just as with ordinary functions, the order of the operators in the composition matters; $T(S(x))$ and $S(T(x))$ need not be the same discrete signal. We say that operators T and S *commute* if $T(S(x)) = S(T(x))$, for all x; in that case we write $TS = ST$.

An operator T is said to be *shift-invariant* if $TS_k = S_kT$ for all integers k. This means that if y is the output of the system described by T when the input is x, then when we shift the input by k, from x to S_kx, all that happens to the output is that the y is also shifted by k, from y to S_ky. For example, suppose that T is the squaring operator, defined by $T(x) = y$ with $y(n) = x(n)^2$. Then T is shift-invariant. On the other hand, the operator T with $y = T(x)$ such that $y(n) = x(-n)$ is not shift-invariant.

Exercise 5.2 *Which of the following operators are shift-invariant?*

a. $T(x)(n) = x(0) + x(n)$;

b. $T(x)(n) = x(n) + x(-n)$;

c. $T(x)(n) = \sum_{k=-2}^{2} x(n+k)$.

We are most interested in operators T that are both linear and shift-invariant; these are called LSI operators. An LSI operator T is often viewed as a linear system having inputs called x and outputs called y, where $y = T(x)$, and we speak of a LSI system.

5.3.3 Convolution Operators

Let h be a fixed discrete signal. For any discrete signal x define $y = T(x)$ by

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k),$$

for any integer n. We then say that y is the *convolution* of x with h and write $y = x * h$. Notice that $x * h = h * x$; that is,

$$\sum_{k=-\infty}^{\infty} h(k)x(n-k) = \sum_{k=-\infty}^{\infty} x(k)h(n-k).$$

The operator T is then the *convolution with* h operator. Any such T is linear.

5.3.4 LSI Filters are Convolutions

The operator T that is convolution with h is linear and shift-invariant. The most important fact in signal processing is that every T that is *linear and shift-invariant* (LSI) must be convolution with h, for some fixed discrete signal h.

Because of the importance of this result we give a proof now. First, we must find the h. To do this we let $x = \delta$; the h we seek is then the output $h = T(\delta)$. Now we must show that, for any other input x, we have $T(x) = x * h$. Note that for any k we have $\delta_k = S_k(\delta)$, so that

$$T(\delta_k) = T(S_k(\delta)) = S_k(T(\delta)) = S_k(h),$$

and so

$$T(\delta_k)(n) = S_k(h)(n) = h(n-k).$$

We can write an arbitrary x in terms of the δ_k as

$$x = \sum_{k=-\infty}^{\infty} x(k)\delta_k.$$

Then

$$T(x)(n) = T(\sum_{k=-\infty}^{\infty} x(k)\delta_k)(n) = \sum_{k=-\infty}^{\infty} x(k)T(\delta_k)(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k).$$

Therefore, $T(x) = x * h$, as we claimed. Because the h associated with the operator T is $h = T(\delta)$, the discrete signal h is called the *impulse-response function* of the system.

5.4 Special Types of Discrete Signals

Some of our calculations, such as convolution, involve infinite sums. In order for these sums to make sense we would need to impose certain restrictions on the discrete signals involved. Some books consider only discrete signals x that are *absolutely summable*, that is, for which

$$\sum_{n=-\infty}^{\infty} |x(n)| < \infty,$$

or, at least, x that are *bounded*, which means that there is a positive constant $b > 0$ with $|x(n)| \leq b$ for all n. Sometimes the condition of absolute summability is imposed only on discrete functions h that are to be associated with LSI operators. Operators T whose h is absolutely summable have the desirable property of *stability*; that is, if the input function x is bounded, so is the output function $y = T(x)$. This property is also called the *bounded in, bounded out* (BIBO) property.

Exercise 5.3 *Show that the operator T is a stable operator if and only if its associated h is absolutely summable. Hint: If h is not absolutely summable, consider the input sequence with $x(n) = \overline{h(-n)}/|h(n)|$.*

In order to make use of the full power of Fourier methods some texts require that discrete signals x be *absolutely square-summable*, that is,

$$\sum_{n=-\infty}^{\infty} |x(n)|^2 < \infty.$$

Exercise 5.4 *Show that the discrete signal $x(n) = \frac{1}{|n|+1}$ is absolutely square-summable, but not absolutely summable.*

Our approach will be to avoid discussing specific requirements, with the understanding that some requirements will usually be needed to make the mathematics rigorous.

5.5 The Frequency-Response Function

Just as sine and cosine functions play important roles in calculus, so do their discrete counterparts in signal processing. The discrete sine function with frequency ω is the discrete signal $\sin_\omega$ with

$$\sin_\omega(n) = \sin(\omega n),$$

for each integer n. Similarly, the discrete cosine function with frequency ω is $\cos_\omega$ with

$$\cos_\omega(n) = \cos(\omega n).$$

It is convenient to include in the discussion the complex exponential e_ω defined by

$$e_\omega(n) = \cos_\omega(n) + i \sin_\omega(n) = e^{i\omega n}.$$

Since these discrete signals are the same for ω and $\omega + 2\pi$ we assume that ω lies in the interval $[-\pi, \pi)$.

5.5.1 The Response of a LSI System to $x = e_\omega$

Let T denote a LSI system and let ω be fixed. We show now that

$$T(e_\omega) = He_\omega,$$

for some constant H. Since the H can vary as we change ω it is really a function of ω, so we denote it $H = H(\omega)$.

Let $v = \{v(n)\}$ be the signal $v = e_\omega - S_1(e_\omega)$. Then we have

$$v(n) = e^{in\omega} - e^{i(n-1)\omega} = (1 - e^{-i\omega})e^{in\omega}.$$

Therefore, we can write

$$v = (1 - e^{-i\omega})e_\omega,$$

from which it follows that

$$T(v) = (1 - e^{-i\omega})T(e_\omega). \tag{5.1}$$

But we also have

$$T(v) = T(e_\omega - S_1(e_\omega)) = T(e_\omega) - TS_1(e_\omega),$$

and, since T is shift-invariant, $TS_1 = S_1T$, we know that

$$T(v) = T(e_\omega) - S_1T(e_\omega). \tag{5.2}$$

Combining Equations (5.1) and (5.2), we get

$$(1 - e^{-i\omega})T(e_\omega) = T(e_\omega) - S_1T(e_\omega).$$

Therefore,

$$S_1T(e_\omega) = e^{-i\omega}T(e_\omega),$$

or

$$T(e_\omega)(n-1) = S_1T(e_\omega)(n) = e^{-i\omega}T(e_\omega)(n).$$

We conclude from this that

$$e^{in\omega}T(e_\omega)(0) = T(e_\omega)(n),$$

for all n. Finally, we let $H(\omega) = T(e_\omega)(0)$.

It is useful to note that we did not use here the fact that T is a convolution operator. However, since we do know that $T(x) = x * h$, for $h = T(\delta)$, we can relate the function $H(\omega)$ to h.

5.5.2 Relating $H(\omega)$ to $h = T(\delta)$

Since T is a LSI operator, T operates by convolving with $h = T(\delta)$. Consider what happens when we select for the input the discrete signal $x = e_\omega$. Then the output is $y = T(e_\omega)$ with

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)e^{i\omega(n-k)} = H(e^{i\omega})e^{i\omega n},$$

where

$$H(e^{i\omega}) = \sum_{k=-\infty}^{\infty} h(k)e^{-i\omega k} \tag{5.3}$$

is the value, at ω, of the *frequency-response function* of T. The point here is that when the input is $x = e_\omega$ the output is a multiple of e_ω, the multiplier being the (possibly complex) number $H(e^{i\omega})$. Linear, shift-invariant systems T do not alter the frequency of the input, but just change its amplitude and/or phase. The constant $H(e^{i\omega})$ is the same as $H(\omega)$ obtained earlier; having two different notations for the same function is an unfortunate, but common, occurrence in the signal-processing literature.

It is important to note that the infinite sum in Equation (5.3) need not converge for arbitrary $h = \{h(k)\}$. It does converge, obviously, whenever h is finitely nonzero; it will also converge for infinitely nonzero sequences that are suitably restricted.

A common problem in signal processing is to design a LSI filter with a desired frequency-response function $H(e^{i\omega})$. To determine $h(m)$, given $H(e^{i\omega})$, we multiply both sides of Equation (5.3) by $e^{i\omega m}$, multiply by $\frac{1}{2\pi}$, integrate over the interval $[-\pi, \pi]$, and use the helpful fact that

$$\int_{-\pi}^{\pi} e^{i(m-k)\omega} d\omega = 0,$$

for $m \neq k$. The result is

$$h(m) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{i\omega})e^{i\omega m} d\omega. \tag{5.4}$$

It is useful to extend the definition of $H(e^{i\omega})$ to permit $e^{i\omega}$ to be replaced by any complex number z. Then we get the *z-transform* of h, given by

$$H(z) = \sum_{k=-\infty}^{\infty} h(k)z^{-k}.$$

We can study the working of the system T on more general inputs x by representing x as a sum of complex-exponential discrete signals e_ω.

The representation, in Equation (5.4), of the infinite sequence $h = \{h(k)\}$ as a superposition of complex-exponential discrete signals suggests the possibility that such a representation is available for general infinite discrete signals, a notion we take up in the next section.

5.6 The Discrete Fourier Transform

A common theme running through mathematics is the representation of complicated objects in terms of simpler ones. Taylor-series expansion enables us to view quite general functions as infinite versions of polynomials by representing them as infinite sums of the power functions. Fourier-series expansions give representations of quite general functions as infinite sums of sines and cosines. Here we obtain similar representation for discrete signals, as infinite sums of the complex exponentials, e_ω, for ω in $[-\pi, \pi)$.

Our goal is to represent a general discrete signal x as a sum of the e_ω, for ω in the interval $[-\pi, \pi)$. Such a sum is, in general, an integral over ω. So we seek to represent x as

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{i\omega n} d\omega, \tag{5.5}$$

where $X(\omega)$ is a function to be determined. As we shall see, the function we seek is the *discrete Fourier transform* (DFT) of x, defined by

$$X(\omega) = \sum_{m=-\infty}^{\infty} x(m) e^{-i\omega m}. \tag{5.6}$$

This follows from the discussion leading up to Equation (5.4). Notice that in the case $x = h$ the function $H(\omega)$ is the same as the frequency-response function $H(e^{i\omega})$ defined earlier. For this reason the notation $X(\omega)$ and $X(e^{i\omega})$ are used interchangably. The DFT of the discrete signal x is sometimes called the *discrete-time Fourier transform* (DTFT).

The sum in Equation (5.6) is the *Fourier-series expansion* for the function $X(\omega)$, over the interval $[-\pi, \pi)$; the $x(n)$ are its *Fourier coefficients*.

The infinite series in Equation (5.4) that is used to define $X(\omega)$ may not converge. For example, suppose that x is an exponential signal, with $x(n) = e^{i\omega_0 n}$. Then the infinite sum would be

$$\sum_{m=-\infty}^{\infty} e^{i(\omega_0 - \omega)m},$$

which obviously does not converge, at least in any ordinary sense. Consider, though, what happens when we put this sum inside an integral and reverse the order of integration and summation. Specifically, consider

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} F(\omega)\sum_{m=-\infty}^{\infty} e^{i(\omega_0-\omega)m}d\omega,$$

$$=\sum_{m=-\infty}^{\infty}(\frac{1}{2\pi}\int_{-\pi}^{\pi} F(\omega)e^{i(\omega_0-\omega)m}d\omega),$$

$$=\sum_{m=-\infty}^{\infty} e^{i\omega_0 m}f(m)=F(\omega_0).$$

So, the infinite sum acts like the Dirac delta $\delta(\omega-\omega_0)$. This motivates the following definition of the infinite sum:

$$\sum_{m=-\infty}^{\infty} e^{i(\omega_0-\omega)m}=\delta(\omega-\omega_0). \tag{5.7}$$

A different approach to the infinite sum is to consider

$$\lim_{N\to+\infty}\frac{1}{2N+1}\sum_{m=-N}^{N} e^{i(\omega_0-\omega)m}.$$

According to Equation (18.4), we have

$$\sum_{n=-N}^{N} e^{i\omega n}=\frac{\sin(\omega(N+\frac{1}{2}))}{\sin(\frac{\omega}{2})}.$$

Therefore,

$$\lim_{N\to+\infty}\frac{1}{2N+1}\sum_{m=-N}^{N} e^{i(\omega_0-\omega)m}=1, \tag{5.8}$$

for $\omega=\omega_0$, and zero, otherwise.

5.7 The Convolution Theorem

Once again, let $y=T(x)$, where T is a LSI operator with associated filter $h=\{h(k)\}$. Because we can write

$$x(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e_\omega(n)d\omega,$$

or, in shorthand, leaving out the n, as

$$x=\frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e_\omega d\omega,$$

we have

$$y = T(x) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)T(e_\omega)d\omega,$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)H(\omega)e_\omega d\omega,$$

or

$$y(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)H(\omega)e_\omega(n)d\omega.$$

But we also have

$$y(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} Y(\omega)e_\omega(n)d\omega,$$

from which we conclude that

$$Y(\omega) = X(\omega)H(\omega), \tag{5.9}$$

for each ω in $[-\pi, \pi)$.

Equation (5.9) is the most important equation in signal processing. It describes the activity of an LSI system by telling us that the system simply multiplies the DFT of the input x by the DFT of the h, the frequency-response function of the system, to produce the DFT of the output y. Since $y = x * h$ it also tells us that whenever y is formed by convolving two discrete signals x and h, its DFT is the product of the DFT of x and the DFT of h.

5.8 Sampling and Aliasing

The term *sampling* refers to the transition from a function $f(t)$ of a continuous variable to a discrete signal x, defined by $x(n) = f(n\Delta)$, where $\Delta > 0$ is the *sample spacing*. For example, suppose that $f(t) = \sin(\gamma t)$ for some frequency $\gamma > 0$. Then $x(n) = \sin(\gamma n\Delta) = \sin(\omega n)$, where $\omega = \gamma\Delta$. We define $X(\omega)$, the DFT of the discrete signal x, for $|\omega| \leq \pi$, so we need $|\gamma|\Delta \leq \pi$. This means we must select Δ so that $\Delta \leq \pi/|\gamma|$. In general, if the function $f(t)$ has sinusoidal components with frequencies γ such that $|\gamma| \leq \Gamma$ then we should select $\Delta \leq \pi/\Gamma$.

If we select Δ too large, then a frequency component of $f(t)$ corresponding to $|\gamma| > \pi/\Delta$ will be mistaken for a frequency with smaller magnitude. This is *aliasing*. For example, if $f(t) = \sin(3t)$, but $\Delta = \pi/2$, then the frequency $\gamma = 3$ is mistaken for the frequency $\gamma = -1$, which lies in $[-2, 2]$. When we sample we get

$$x(n) = \sin(3\Delta n) = \sin(-\Delta n + 4\Delta n) = \sin(-\Delta n + 2\pi n) = \sin(-\Delta n),$$

for each n.

5.9 Important Problems in Discrete Signal Processing

A number of important problems in signal processing involve the relation between a discrete signal and its DFT. One problem is the design of a system to achieve a certain desired result, such as low-pass filtering. A second problem is to estimate the $X(\omega)$ when we do not have all the values $x(n)$, but only finitely many of them.

5.9.1 Low-pass Filtering

When we represent a discrete signal x using

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e_{\omega}(n)d\omega,$$

we take the point of view that the function x is made up of the various discrete sinusoids, the functions e_{ω}, each contributing in the amount $\frac{1}{2\pi}X(\omega)$. Since $X(\omega)$ is usually complex we must interpret this in terms of both an amplitude modulation and a phase change. Suppose that, for some fixed Ω in the interval $(0, \pi)$, we wish to design a system that will leave $X(\omega)$ unchanged for those ω in the interval $[-\Omega, \Omega]$ and change $X(\omega)$ to zero otherwise; such a system is called the (ideal) Ω-*low-pass filter*. To achieve this result we need to take $H(\omega)$ to be $\chi_{\Omega}(\omega)$, the characteristic function of the interval $[-\Omega, \Omega]$, with $\chi_{\Omega}(\omega) = 1$, for $|\omega| \leq \Omega$, and $\chi_{\Omega}(\omega) = 0$, otherwise. We find the $h(k)$ using

$$h(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \chi_{\Omega}(\omega)e^{i\omega k}d\omega.$$

Performing the integration, we find that $h(0) = \Omega/\pi$ and, for $k \neq 0$,

$$h(k) = \frac{\sin \Omega k}{\pi k}.$$

To calculate the low-pass output

$$y(n) = \sum_{k=-\infty}^{\infty} \frac{\sin \Omega k}{\pi k} x(n-k)$$

we need infinitely many values $x(m)$ for $m > n$, as well as infinitely many values for $m < n$. If we think of n as time, then to calculate the value of y at time n we need to know the values of x for the entire infinite past before time n, as well as the values for the entire infinite future after time n. Clearly, this is inconvenient if we wish to perform the filtering in real-time. One goal of signal processing is to approximate such filters with ones that are more convenient, using, say, only finitely many past and future values of the input.

5.9.2 The Finite-Data Problem

In practice we have finite data obtained from measurements. We view these data as values $x(n)$ for finitely many values of n, say $n = 0, 1, ..., N - 1$. The function $X(\omega)$ often is an important object in the problem and must be estimated from the data. One possible estimate is

$$\hat{X}(\omega) = \sum_{n=0}^{N-1} x(n)e^{-i\omega n}.$$

To distinguish this from the DFT, which involves the infinite sum, we shall call $\hat{X}(\omega)$ the DFT of the vector $\mathbf{x} = (x(0), ..., x(N-1))^T$. If N is large, the DFT of $\mathbf{x}$ will usually be a satisfactory approximation of $X(\omega)$. However, in many applications N is not large and the DFT of $\mathbf{x}$ is not adequate. The *finite-data problem* is how to find better estimates of $X(\omega)$ from the limited data we have.

Because the finite-data problem involves approximating one function of a continuous variable by another, we need some way to measure how far apart two such functions are. The way most commonly used in signal processing is the so-called *Hilbert-space distance*, given by

$$||X(\omega) - Y(\omega)|| = \sqrt{\int_{-\pi}^{\pi} |X(\omega) - Y(\omega)|^2 d\omega.}$$

We shall return later to the problem of describing best approximations in Hilbert space.

5.9.3 The Extrapolation Problem

If $x(n)$ is obtained from $f(t)$ by sampling, that is, $x(n) = f(n\Delta)$, we have

$$f(n\Delta) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega)e^{in\omega} d\omega. \tag{5.10}$$

Changing to the variable $\gamma = \omega/\Delta$, and defining $\Gamma = \pi/\Delta$, we can write

$$f(n\Delta) = \frac{\Delta}{2\pi} \int_{-\Gamma}^{\Gamma} X(\gamma\Delta)e^{i(n\Delta)\gamma} d\gamma, \tag{5.11}$$

which makes clearer the use of the sampling time $t = n\Delta$.

The representation in Equation (5.11) is suggestive! Let us define $g(t)$ for all t by the formula

$$g(t) = \frac{\Delta}{2\pi} \int_{-\Gamma}^{\Gamma} X(\gamma\Delta)e^{it\gamma} d\gamma. \tag{5.12}$$

Do we have $g(t) = f(t)$ for all t? On the face of it, it would seem that the answer is clearly no. How could a function of a continuous variable be completely determined by such a sequence of its values? Can we capture all of a function $f(t)$ from discrete samples? It fact, under certain conditions, the answer is yes. Let us investigate what those conditions might be.

Let $\epsilon > 0$ and let $h_\epsilon(t) = \sin((\Gamma+\epsilon)t) - \sin((-\Gamma+\epsilon)t)$. Then $h_\epsilon(n\Delta) = 0$ for each integer n. From the data we have, we cannot decide if $f(t) = g(t)$ or $f(t) = g(t) + h_\epsilon(t)$, or, perhaps, $f(t) = g(t) + h_\epsilon(t)$ for some other ϵ. Notice that, in order to construct $h_\epsilon(t)$ we need a sine function with a frequency outside the interval $[-\Gamma, \Gamma]$.

On the other hand, if $F(\gamma)$, the Fourier transform of $f(t)$, is zero outside $[-\Gamma, \Gamma]$, then $f(t) = g(t)$. This is because the function $F(\gamma)$ has a Fourier-series representation

$$F(\gamma) = \sum_{n=-\infty}^{\infty} a_n e^{i\gamma n\Delta},$$

where, as in our discussion of the DFT, we have

$$a_n = \frac{1}{2\Gamma}\int_{-\Gamma}^{\Gamma} F(\gamma)e^{-i\gamma n\Delta}d\gamma.$$

But the expression on the right side of this equation equals $\Delta f(n\Delta)$, according to the Fourier Inversion Formula. Therefore

$$F(\gamma) = \Delta\sum_{n=-\infty}^{\infty} f(n\Delta)e^{i\gamma n\Delta}$$

$$= \Delta\sum_{n=-\infty}^{\infty} x(n)e^{i\gamma n\Delta}$$

$$= \Delta\sum_{n=-\infty}^{\infty} x(n)e^{i\omega n} = \Delta X(-\gamma\Delta).$$

So, we can write

$$f(t) = \frac{1}{2\pi}\int_{-\Gamma}^{\Gamma} F(\gamma)e^{-it\gamma}d\gamma,$$

$$= \frac{\Delta}{2\pi}\int_{-\Gamma}^{\Gamma} X(\gamma\Delta)e^{it\gamma}d\gamma = g(t).$$

For an arbitrary function $f(t)$ we seek a representation of $f(t)$ as a superposition of complex exponential functions, that is,

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} A(\gamma)e^{it\gamma}d\gamma, \tag{5.13}$$

for some function $A(\gamma)$. The function $A(\gamma)$ that does the job is $A(\gamma) = F(-\gamma)$, where $F(\gamma)$ is the *Fourier transform* of $f(t)$. If $F(\gamma) = 0$ for $|\gamma| > \Gamma$, then $f(t)$ is said to be Γ-*bandlimited*; in this case $F(\gamma) = \Delta X(-\gamma\Delta)$, as discussed previously.

It is important to note that we cannot tell from the samples $x(n) = f(n\Delta)$ whether or not $f(t)$ is Γ-bandlimited. If $f(t)$ is not Γ-bandlimited, but we assume that it is, there will be components of $f(t)$ with frequencies outside the band $[-\Gamma, \Gamma]$ that will be mistaken for sinusoids having frequencies inside the band; this is aliasing.

5.10 Discrete Signals from Finite Data

In problems involving actual data obtained from measurements we may have a vector $\mathbf{x} = (x_1, ..., x_N)^T$ that we wish to associate with a discrete function x. There are, of course, any number of ways to do this. Two of the most commonly used ways employ *zero extension* and *periodic extension.*

5.10.1 Zero-extending the Data

We define $x(n)$ to be x_{n+1}, for $n = 0, ..., N-1$ and $x(n) = 0$ otherwise. Then x is a discrete function that extends the data. The DFT of x is now

$$X(\omega) = \sum_{n=0}^{N-1} x(n)e^{-in\omega}, \tag{5.14}$$

for $|\omega| \leq \pi$ and, from the fact that

$$0 = \int_{-\pi}^{\pi} e^{i(m-n)\omega} d\omega$$

for $m \neq n$, we have

$$x(m) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e^{im\omega} d\omega,$$

for all integers m.

The DFT of x obtained by zero-extending the data provides a way to represent the data as a (continuous) sum, or integral, of the discrete exponential functions e_ω:

$$x_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e^{i(n-1)\omega} d\omega,$$

for $n = 1, ..., N$.

5.10.2 Periodically Extending the Data

Another way to associate a discrete function $\tilde{x}$ with the data vector $\mathbf{x}$ is by extending the data periodically. For $n = 0, ..., N-1$ let $\tilde{x}(n) = x_{n+1}$ and for any integer n define $\tilde{x}(n) = \tilde{x}(n \bmod N)$. Then $\tilde{x}$ extends the data and is N-periodic; that is, $\tilde{x}(n+N) = \tilde{x}(n)$ for all integers n.

Now we want to represent the N-periodic $\tilde{x}$ as a sum of the discrete exponential functions e_ω. Notice, however, that most of the e_ω are not N-periodic; in order for $e^{i(n+N)\omega} = e^{in\omega}$ for all integers n we need $e^{iN\omega} = 1$. This means that $\omega = 2\pi k/N$, for some integer k. Therefore, we shall seek to represent $\tilde{x}$ as a sum of the discrete exponential functions e_ω only for $\omega = 2\pi k/N$. Let us denote such functions as e_k. Notice also that e_{k+N} and e_k are the same function, for any integer k. Therefore, we seek to represent $\tilde{x}$ as a sum of the discrete exponential functions e_k, for $k = 0, 1, ..., N-1$; that is, we want

$$\tilde{x}(n) = \sum_{k=0}^{N-1} X_k e^{2\pi ikn/N}, \tag{5.15}$$

for some choice of numbers X_k.

To determine the X_k we multiply both sides of Equation (5.15) by $e^{-2\pi ijn/N}$ and sum over n. Using the fact that

$$\sum_{n=0}^{N-1} e^{2\pi i(k-j)n/N} = 0,$$

if $k \neq j$, it follows that

$$X_j = \frac{1}{N}\sum_{n=0}^{N-1} x(n) e^{-2\pi ijn/N}, \tag{5.16}$$

for $j = 0, ..., N-1$.

We began with a finite vector $\mathbf{x} = (x_1, ..., x_N)^T$, which we chose to write as $\mathbf{x} = (x(0), ..., x(N-1))^T$, and ended with a finite set of numbers X_j, $j = 0, ..., N-1$, which we used to form the vector $\mathbf{X} = (X_0, ..., X_{N-1})^T$. It is common practice to call the vector $\mathbf{X}$ the DFT of the vector $\mathbf{x}$, but to avoid confusion, we shall refer to the vector $\mathbf{X}$ as the *vector* DFT (vDFT) of the vector $\mathbf{x}$, leaving the terminology DFT of $\mathbf{x}$ to refer to the DFT of the zero-extended discrete function x in equation (5.14). Notice, though, that the vDFT and the DFT are related; for $0 \leq k \leq N/2$ we have $X_k = X(2\pi k/N)$ and for $N/2 < k \leq N-1$ we have $X_k = X(-\pi + 2\pi k/N)$. The vector DFT plays an important role in signal processing because, as we shall see later, there is a fast algorithm for calculating it from the data, called the *fast Fourier transform* (FFT).

5.10.3 A Third Way to Extend the Data

Another way to extend the data vector to a discrete function is to *zero-pad* and then to perform periodic extension. Given the data $x(n)$, $n = 0, ..., N-1$, let $x(n) = 0$, $n = N, N+1, ..., M-1$. Then extend these M numbers M-periodically, so that $\tilde{x}(n) = x(n \bmod M)$, for each integer n. Then $\tilde{x}(n+M) = \tilde{x}(n)$, for all n.

Now, when we represent $\tilde{x}$ as a sum of sinusoids we have

$$\tilde{x}(n) = \sum_{k=0}^{M-1} X_k e^{2\pi ikn/M}, \tag{5.17}$$

for some choice of numbers X_k. Arguing as before, we find that now we have

$$X_k = \frac{1}{M} \sum_{n=0}^{M-1} x(n) e^{-2\pi ikn/M}, \tag{5.18}$$

for $k = 0, ..., M-1$.

5.10.4 A Fourth Way: Bandlimited Extrapolation

Suppose that $f(t)$ is Γ-bandlimited, so that

$$f(t) = \frac{\Delta}{2\pi} \int_{-\Gamma}^{\Gamma} X(\gamma\Delta) e^{it\gamma} d\gamma. \tag{5.19}$$

Inserting $X(\gamma\Delta)$ as in Equation (5.6) into Equation (5.19) and performing the indicated integration, we obtain

$$f(t) = \Delta \sum_{n=-\infty}^{\infty} f(n\Delta) \frac{\sin \Gamma(t - n\Delta)}{\pi(t - n\Delta)}. \tag{5.20}$$

This formula illustrates Shannon's sampling theorem, by showing how to reconstruct the Γ-bandlimited function $f(t)$ from the infinite sequence of samples $\{f(n\Delta)\}$, for any $\Delta < \frac{\pi}{\Gamma}$. We shall use this formula to extend our finite data to obtain a Γ-bandlimited function that is consistent with the finite data. It is not required that the data be equispaced.

Arbitrarily Spaced Data

Now suppose that our data are the values $f(t_m)$, $m = 1, ..., N$, where the t_m are arbitrary. From Equation (5.20) we have

$$f(t_m) = \Delta \sum_{n=-\infty}^{\infty} f(n\Delta) \frac{\sin \Gamma(t_m - n\Delta)}{\pi(t_m - n\Delta)}, \tag{5.21}$$

for each t_m. In this case, however, we do not know the $f(n\Delta)$. Can we find a sequence $\{f(n\Delta)\}$ for which Equation (5.21) is satisfied for each m? The answer is yes; in fact, there are infinitely many ways to do this, as we shall see shortly. But, first, we need a useful identity concerning Γ-bandlimited functions.

A Useful Identity

The function $G(\gamma) = \chi_\Gamma(\gamma)$ that is one for $|\gamma| \leq \Gamma$ and is zero otherwise is the Fourier transform of the function $g(x) = \frac{\sin \Gamma x}{\pi x}$. Therefore, its sequence of Fourier coefficients is $\{\Delta g(n\Delta) = \Delta \frac{\sin \Gamma n\Delta}{\pi n\Delta}\}$. For any fixed t, the function $H_t(\gamma) = G(\gamma)e^{i\gamma t}$ has, for its sequence of Fourier coefficients, $h_t = \{\Delta \frac{\sin \Gamma(n\Delta - t)}{\pi(n\Delta - t)}\}$. Since $H_t(\gamma)H_{-s}(\gamma) = H_{t-s}(\gamma)$, we have $h_t * h_{-s} = h_{t-s}$. Writing this out, we get

$$\frac{\sin \Gamma(n\Delta - t + s)}{\pi(n\Delta - t + s)} =$$

$$\Delta \sum_{k=-\infty}^{\infty} \frac{\sin \Gamma(k\Delta - t)}{\pi(k\Delta - t)} \frac{\sin \Gamma((n-k)\Delta + s)}{\pi((n-k)\Delta + s)}. \tag{5.22}$$

Minimum-Norm Extrapolation

One possibility is to provide a finite-parameter model for the desired sequence $\{f(n\Delta)\}$, as

$$f(n\Delta) = \sum_{j=1}^{N} z_j \frac{\sin \Gamma(t_j - n\Delta)}{\pi(t_j - n\Delta)}. \tag{5.23}$$

Inserting this $f(n\Delta)$ into Equation (5.21), reversing the order of summation, and using the identity in Equation (5.22), we obtain

$$f(t_m) = \Delta \sum_{j=1}^{N} z_j \frac{\sin \Gamma(t_j - t_m)}{\pi(t_j - t_m)}. \tag{5.24}$$

This system of N equations in N unknowns can be solved uniquely for the z_j. Placing these z_j into Equation (5.23) to get the $f(n\Delta)$ and then using these $f(n\Delta)$ in Equation (5.20), we obtain a Γ-bandlimited function $\hat{f}(t)$ that extrapolates the finite data. The function $\hat{f}(t)$ can be written explicitly as

$$\hat{f}(t) = \Delta \sum_{j=1}^{N} z_j \frac{\sin \Gamma(t_j - t)}{\pi(t_j - t)}. \tag{5.25}$$

It can be shown that this choice of $\hat{f}(t)$ is the Γ-bandlimited function extrapolating the data for which the energy $\sum_{n=-\infty}^{\infty} |\hat{f}(n\Delta)|^2$ is the smallest.

Estimating the Fourier Transform

We take the Fourier transform of $\hat{f}(t)$ in Equation (5.25), to obtain an explicit formula for $\hat{F}(\gamma)$, our estimate of the Fourier transform of $f(t)$:

$$\hat{F}(\gamma) = \Delta\chi_\Gamma(\gamma)\sum_{j=1}^{J} z_j e^{it_j\gamma}.$$

When $t_j = j\Delta$, with $\Delta = \frac{\pi}{\Gamma}$, we find that $\Delta z_j = f(j\Delta)$, so that our estimate of $F(\gamma)$ becomes

$$\hat{F}(\gamma) = \sum_{j=1}^{J} f(j\Delta)e^{ij\Delta\gamma}.$$

So our estimate of $X(\omega)$ is

$$\hat{X}(\omega) = \hat{F}(-\frac{\omega}{\Delta}) = \sum_{j=1}^{J} f(j\Delta)e^{-ij\omega},$$

which is the DFT we get when we zero-extend the finite data.

Note that if $f(t)$ is known to be Γ-bandlimited, then $f(t)$ is $(\Gamma+\epsilon)$-bandlimited, for any $\epsilon > 0$. Therefore, we can use $\Gamma + \epsilon$ in place of Γ, in the calculations above, to achieve a bandlimited extrapolation of the finite data. So there are infinitely many different ways to extend the finite data as samples of a bandlimited function. Each of these ways leads to a different estimate for the Fourier transform.

5.11 Is this Analysis or Representation?

As we just saw, we can represent the finite data $x(n)$, $n = 0, ..., N-1$, in any number of different ways as sums of discrete exponential functions. In the first way we have

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e^{in\omega}d\omega, \qquad (5.26)$$

in the second way

$$x(n) = \sum_{k=0}^{N-1} X_k e^{2\pi ikn/N}, \qquad (5.27)$$

and in yet a third way

$$x(n) = \sum_{k=0}^{M-1} X_k e^{2\pi ikn/M}. \tag{5.28}$$

Using the bandlimited extrapolation approach, we can also write

$$x(n) = \frac{1}{2\pi} \int_{-\Gamma}^{\Gamma} \hat{F}(\gamma) e^{-in\Delta\gamma} d\gamma. \tag{5.29}$$

In each of these cases it would appear that the data contains certain sinusoidal components, and yet in each of these ways the sinusoidal frequencies involved are different. How can this be?

By *analysis* we mean the identification of the components of the data, in this case, the complex-exponential components or complex sinusoids, that are really there in the data. When we have at least two different ways to represent the data as a sum of such complex exponentials, can either of these be said to provide true analysis of the data? Equation (5.26) seems to say that the data is made up of complex exponentials whose frequencies encompass the entire interval $[-\pi, \pi)$, while Equation (5.27) exhibits the same data as consisting only of N complex exponentials, with frequencies equispaced through the interval $[-\pi, \pi)$, and Equation (5.28) employs a whole new set of M frequencies, equispaced through the interval $[-\pi, \pi)$. Equation (5.29) says the frequencies are spread over the interval $[-\Gamma, \Gamma]$. Which one is correct? This is not really the right question to ask. The proper response depends on the context, that is, on what the problem is that we are trying to solve.

If the goal is to perform some operation on the data, it may not matter greatly how it is represented. However, as we saw in our discussion of farfield propagation, the data can be finitely many samples of an underlying continuous-variable function $f(t)$ or a discrete function x, for which the frequency-space representation has real physical significance. In the discrete case, the DFT of x can have physical significance beyond simply providing a way to represent the x as a sum of exponential functions. For example, in sonar and radar array processing, the arguments ω may correspond to a direction of a distant object of interest, and ω may take on any value in $[-\pi, \pi)$. In such cases we would like to have all of x, but must settle for the finite data vector $\mathbf{x}$. The goal then is to use the finite data to approximate or estimate $X(\omega)$, the DFT of x. The DFT of the data is then a finite Fourier-series approximation of the infinite Fourier series that is $X(\omega)$. The vector DFT $\mathbf{X}$ of the data gives us N equispaced values of this approximation, which can be calculated efficiently using the FFT.

There is an added twist to the story, however. Given only the data, we have no way of knowing the complete x; there are infinitely many x

that extend the data. Which one is the correct one? In most applications we have some prior information about the nature of the function $X(\omega)$ that we seek to estimate from the data. Effective estimation procedures make use of this additional information to obtain better estimates when the data, by itself, is insufficient. Our fourth way to extend the finite data includes, in the extrapolation process, the prior knowledge that $f(t)$ is Γ-bandlimited. Later, we shall consider other ways to employ prior knowledge to extrapolate the data.

5.12 Oversampling

In many applications, we are essentially free to take as many samples as we wish, but are required to take those samples from within some finite region. In the model of farfield propagation, for example, there may be physical limitations on length of our array of sensors, but within that length, we may place as many sensors as seems reasonable. In synthetic-aperture radar, the array of sensors is moving, simulating a longer array, the length of which is limited, in practice, by the need to correct for time differences in the receipt of the signals. In sampling a function of time, the signal being sampled may only last for a short while, but while it lasts, we may take as many samples as we wish; this is the case in seismic exploration, magnetic resonance imaging, and speech processing. In our discussion previously, we saw that if the function $f(t)$ is Γ-bandlimited, then we must sample at a spacing $\Delta \leq \frac{\pi}{\Gamma}$. If we are required to take all our samples from within the time interval $[0,T]$, and if we use $\Delta = \frac{\pi}{\Gamma}$, we may not be able to take a large number of samples. Would it be better, under these circumstances, to *oversample*, that is, to use, say $\frac{\Delta}{2}$, in order to generate more data? Is there any limit on how small the spacing should be?

Suppose we begin with the samples $f(n\Delta)$, for $n = 0, 1, ..., N-1$, $\Delta = \frac{\pi}{\Gamma}$, and $T = N\Delta$. The DFT of the zero-extended data,

$$\hat{F}(\gamma) = \Delta \sum_{n=0}^{N-1} f(n\Delta) e^{in\Delta\gamma},$$

for $|\gamma| \leq \Gamma$, is then a reasonable estimate of the Fourier transform, $F(\gamma)$. Now let us take samples at spacing $\frac{\Delta}{2}$; that is, we take $f(\frac{m\Delta}{2})$, for $m = 0, ..., 2N-1$. The DFT of the zero-extension of this data is

$$\tilde{F}(\gamma) = \frac{\Delta}{2} \sum_{m=0}^{2N-1} f(m\frac{\Delta}{2}) e^{im\frac{\Delta}{2}\gamma}.$$

But now the interval outside of which the sum repeats itself is no longer $[-\Gamma, \Gamma]$, but $[-2\Gamma, 2\Gamma]$; $\tilde{F}(\gamma)$ is an estimate of $F(\gamma)$ for γ in this larger interval. If we consider $\tilde{F}(\gamma)$ only for γ within the smaller interval $[-\Gamma, \Gamma]$, we find that $\tilde{F}(\gamma)$ is not much different from $\hat{F}(\gamma)$ for those values of γ. What has happened is that, when we chose to sample faster, the DFT estimation "believes" that our function $f(t)$ is 2Γ-bandlimited, which is true, but not precise. We do get twice as many data points, but we then are forced to use them to estimate the Fourier transform over an interval that is twice as wide as before.

There is a way out of this predicament, however. The bandlimited extrapolation method discussed earlier permits us to use any finite set of samples, t_j, $j = 1, ..., J$. Therefore, we can take $t_j = (j-1)\frac{\Delta}{2}$, $j = 1, ..., J = 2N$. Then our estimate of $F(\gamma)$ has the form

$$\hat{F}(\gamma) = \Delta \chi_\Gamma(\gamma) \sum_{m=0}^{2N-1} z_{m+1} e^{im\frac{\Delta}{2}\gamma},$$

but, unlike for $\tilde{F}(\gamma)$, the z_{m+1} are not $\frac{1}{\Delta} f(m\frac{\Delta}{2})$.

Simulation experiments show that this method of estimating the Fourier transform from oversampled data does lead to improved estimates, but becomes increasingly sensitive to noise in the data, as the sample spacing gets smaller. The signal-to-noise ratio in the data provides the ultimate limitation on how small we can make the sample spacing.

5.13 Finite Data and the Fast Fourier Transform

Given the finite measurements $x_1, ..., x_N$, we chose to write these as samples of a function $x(t)$, so that $x_n = x(n-1)$, for $n = 1, ..., N$. We then analyzed the vector $\mathbf{x} = (x(0), ..., x(N-1))^T$ in an attempt to uncover interesting components of the function $x(t)$. One approach involved estimating the Fourier transform $X(\omega)$ of $x(t)$ by means of the DFT,

$$\hat{X}(\omega) = \sum_{n=0}^{N-1} x(n) e^{-in\omega},$$

for $|\omega| \leq \pi$. As we noted previously, the Fast Fourier Transform algorithm can be used to calculate finitely many equi-spaced values of $\hat{X}(\omega)$.

There is another way to view the problem. Our data consists of the vector $\mathbf{x}$ and we choose to write $\mathbf{x}$ as a linear combination of other vectors, in the hope of discovering information that lies within the data. There are infinitely many ways to do this, however.

One way is to select N arbitrary distinct frequencies ω_m, $m = 0, 1, ..., N-1$ in $[-\pi, \pi)$ and define the vectors $\mathbf{e}_{\omega_m}$ by

$$\mathbf{e}_{\omega_m}(n) = e^{in\omega_m},$$

for $n = 0, ..., N-1$. We then write

$$\mathbf{x} = \sum_{m=0}^{N-1} a_m \mathbf{e}_{\omega_m},$$

where the coefficients a_m are found by solving the system of linear equations

$$x(n) = \sum_{m=0}^{N-1} a_m \mathbf{e}_{\omega_m}(n),$$

$n = 0, ..., N-1$.

We write the system of linear equations in matrix form as

$$\mathbf{x} = E\mathbf{a}, \tag{5.30}$$

for $\mathbf{a} = (a_0, ..., a_{N-1})^T$ and E the N by N matrix with the entries

$$E_{nm} = e^{in\omega_m}.$$

Such a system will have a unique solution, and we will always be able to write the data vector as a finite sum of N arbitrarily chosen sinusoidal vectors $\mathbf{e}_{\omega_m}$.

In general, the matrix E is invertible, but solving the system in Equation (5.30) when N is large can be computationally expensive. Since we are choosing the frequencies ω_m arbitrarily, why not select them so that the matrix E is easily inverted. This is motivation for the vector DFT of the data.

We now select the frequencies ω_m more carefully. We take

$$\omega_m = -\pi + \frac{2\pi}{N} m,$$

for $m = 0, ..., N-1$.

Exercise 5.5 *Show that, for this choice of the ω_m, the inverse of the matrix E is*

$$E^{-1} = \frac{1}{N} E^{\dagger}.$$

Using the result of this exercise, we find that the coefficient vector $\mathbf{a}$ has entries

$$a_m = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-2\pi imn/N},$$

for $m = 0, ..., N-1$. These are the entries of the vector DFT, $\mathbf{X}$, as given in Equation (5.16). These a_m are what the FFT calculates.

When we consider the problem from this viewpoint, we see that the representation of the data vector $\mathbf{x}$ as a superposition of sinusoidal vectors involves a completely arbitrary selection of the frequencies ω_m to be used, and yet, once the a_m are found, the data vector is completely described as that superposition. The equispaced frequencies used in the previous paragraph were chosen merely to facilitate the inversion of E. What does it mean to say that the data actually contains the components with frequencies ω_m, when we are free to select whichever ones we wish? What does it mean to say that the function $x(t)$ that was sampled to get the data actually contains sinusoids at these frequencies?

Chapter 6

Randomness in Signal Processing

We treat noise in our data using the probabilistic concept of *random variable*. The term is not self-explanatory, so we begin by explaining what a random variable is.

6.1 Random Variables as Models

When we use mathematical tools, such as differential equations, probability, or systems of linear equations, to describe a real-world situation, we say that we are employing a *mathematical model*. Such models must be sufficiently sophisticated to capture the essential features of the situation, while remaining computationally manageable. In this chapter we are interested in one particular type of mathematical model, the *random variable*.

Imagine that you are holding a baseball four feet off the ground. If you drop it, it will land on the ground directly below where you held it. The height of the ball at any time during the fall is described by the function $h(t)$ satisfying the ordinary differential equation $h''(t) = -32\frac{\text{ft}}{\text{sec}^2}$. Solving this differential equation with the initial conditions $h(0) = 4$ ft , $h'(0) = 0\frac{\text{ft}}{\text{sec}}$, we find that $h(t) = 4 - 16t^2$. Solving $h(T) = 0$ for T we find the elapsed time T until impact is $T = 0.5$ sec.. The velocity of the ball at impact is $h'(T) = -32T = -16\frac{\text{ft}}{\text{sec}}$.

Now imagine that, instead of a baseball, you are holding a feather. The feather and the baseball are both subject to the same laws of gravity, but now other aspects of the situation, which we could safely ignore in the case of the baseball, become important in the case of the feather. Like the baseball, the feather is subjected to air resistance and to whatever fluctuations in air currents may be present during its fall. Unlike the baseball, however, the effects of the air matter to the flight of the feather; in fact, they become the dominant factors. When we designed our differential-equation model for the falling baseball we performed no experiments to help us understand its behavior. We simply ignored all other aspects of the situation, and included only gravity in our mathematical model. Even the modeling of gravity was slightly simplified, in that we assumed a constant gravitational acceleration, even though Newton's Laws tell us that it increases as we approach the center of the earth. When we drop the ball and find that our model is accurate we feel no need to change it. When we drop the feather we discover immediately that a new model is needed; but what?

The first thing we observe is that the feather falls in a manner that is impossible to predict with accuracy. Dropping it once again, we notice that it behaves differently this time, landing in a different place and, perhaps, taking longer to land. How are we to model such a situation, in which repeated experiments produce different results? Can we say nothing useful about what will happen when we drop the feather the next time?

As we continue to drop the feather, we notice that, while the feather usually does not fall directly beneath the point of release, it does not fall too far away. Suppose we draw a grid of horizontal and vertical lines on the ground, dividing the ground into a pattern of squares of equal area. Now we repeatedly drop the feather and record the proportion of times the feather is (mainly) contained within each square; we also record the elapsed time. As we are about to drop the feather the next time, we may well assume that the outcome will be consistent with the behavior we have observed during the previous drops. While we cannot say for certain where the feather will fall, nor what the elapsed time will be, we feel comfortable making a *probabilistic statement* about the likelihood that the feather will land in any given square and about the elapsed time.

The squares into which the feather may land are finite, or, if we insist on creating an infinite grid, discretely infinite, while the elapsed time can be any positive real number. Let us number the squares as $n = 1, 2, 3, ...$ and let p_n be the proportion of drops that resulted in the feather landing mainly in square n. Then $p_n \geq 0$ and $\sum_{n=1}^{\infty} p_n = 1$. The sequence $p = \{p_n | n = 1, 2, ...\}$ is then a *discrete probability sequence* (dps), or a *probability sequence*, or a *discrete probability.* Now let N be the number of the square that will contain the feather on the next drop. All we can say about N is that, according to our model, the probability that N will equal n is p_n. We call N a *discrete random variable* with *probability sequence p.*

It is difficult to be more precise about what probability really means. When we say that the probability is p_n that the feather will land in square n on the next drop, where does that probability reside? Do we believe that the numbers p_n are *in the feather* somehow? Do these numbers simply describe our own ignorance, so are *in our heads*? Are they a combination of the two, in our heads as a result of our having experienced what the feather did previously? Perhaps it is best simply to view probablity as a type of mathematical model that we choose to adopt in certain situations.

Now let T be the elapsed time for the next feather to hit the ground. What can we say about T? Based on our prior experience, we are willing to say that, for any interval $[a, b]$ within $(0, \infty)$, the probability that T will take on a value within $[a, b]$ is the proportion of prior drops in which the elapsed time was between a and b. Then T is a *continuous random variable*, in that the values it may take on (in theory, at least) lie in a continuum. To help us calculate the probabilities associated with T we use our prior experience to specify a function $f_T(t)$, called the *probability density function* (pdf) of T, having the property that the probability that T will lie between a and b can be calculated as $\int_a^b f_T(t)dt$. Such $f_T(t)$ will have the properties $f_T(t) \geq 0$ for all positive t and $\int_0^\infty f_T(t)dt = 1$.

In the case of the falling feather we had to perform experiments to determine appropriate ps p and pdf $f_T(t)$. In practice, we often describe our random variables using a ps or pdf from a well-studied parametric family of such mathematical objects. Popular examples of such ps and pdf, such as Poisson probabilities and Gaussian pdf, are discussed early in most courses in probability theory.

It is simplest to discuss the main points of random signal processing within the context of discrete signals, so we return there now.

6.2 Discrete Random Signal Processing

Previously, we have encountered specific discrete functions, such as δ_k, u, e_ω, whose values at each integer n are given by an exact formula. In signal processing we must also concern ourselves with discrete functions whose values are not given by such formulas, but rather, seem to obey only probabilistic laws. We shall need such discrete functions to model noise. For example, imagine that, at each time n, a fair coin is tossed and $x(n) = 1$ if the coin shows heads, $x(n) = -1$ if the coin shows tails. We cannot determine the value of $x(n)$ from any formula; we must simply toss the coins. Given any discrete function x with values $x(n)$ that are either 1 or -1, we cannot say if x was generated by such a coin-flipping manner. In fact, any such x could have been the result of coin flips. All we can say is how likely it is that a particular x was so generated. For example, if $x(n) = 1$ for n even and $x(n) = -1$ for n odd, we feel, intuitively, that it is highly unlikely that such an x came from random coin tossing. What bothers us, of course, is that the values $x(n)$ seem so predictable; randomness seems to require some degree of unpredictability. If we were given two such sequences, the first being the one described above, with 1 and -1 alternating, and the second exhibiting no obvious pattern, and asked to select the one generated by independent random coin tossing, we would clearly choose the second one. There is a subtle point here, however. When we say that we are “given an infinite sequence” what do we really mean? Because the issue here is not the infinite nature of the sequences, let us reformulate the discussion in terms of finite vectors of length, say, 100, with entries 1 or -1. If we are shown a print-out of two such vectors, the first with alternating 1 and -1, and the second vector exhibiting no obvious pattern, we would immediately say that it was the second one that was generated by the coin-flipping procedure, even though the two vectors are equally likely to have been so generated. The point is that we associate randomness with the absence of a pattern, more than with probability. When there is a pattern, the vector can be described in a way that is significantly shorter than simply listing its entries. Indeed, it has been suggested that a vector is random if it cannot be described in a manner shorter than simply listing its members.

6.2.1 The Simplest Random Sequence

We say that a sequence $x = \{x(n)\}$ is a *random sequence* or a *discrete random process* if $x(n)$ is a random variable for each integer n. A simple, yet remarkably useful, example is the random-coin-flip sequence, which we shall denote by $c = \{c(n)\}$. In this model a coin is flipped for each n and $c(n) = 1$ if the coin comes up heads, with $c(n) = -1$ if the coin comes up tails. It will be convenient to allow for the coin to be *biased*, that is, for the probabilities of heads and tails to be unequal. We denote by p the probability that heads occurs and $1 - p$ the probability of tails; the coin is called *unbiased* or *fair* if $p = 1/2$. To find the *expected value* of $c(n)$, written $E(c(n))$, we multiply each possible value of $c(n)$ by its probability and sum; that is,

$$E(c(n)) = (+1)p + (-1)(1 - p) = 2p - 1.$$

If the coin is fair then $E(c(n)) = 0$. The variance of the random variable $c(n)$, measuring its tendency to deviate from its expected value, is $var(c(n)) = E([c(n) - E(c(n))]^2)$. We have

$$var(c(n)) = [+1 - (2p - 1)]^2 p + [-1 - (2p - 1)]^2 (1 - p) = 4p - 4p^2.$$

If the coin is fair then $var(c(n)) = 1$. It is important to note that we do not change the coin at any time during the generation of the random sequence c; in particular, the p does not depend on n.

The random-coin-flip sequence c is the simplest example of a discrete random process or a random discrete function. It is important to remember that a random discrete function is not any one particular discrete function, but rather a probabilistic model chosen to allow us to talk about the probabilities associated with the values of the $x(n)$. In the next section we shall use this discrete random process to generate a wide class of discrete random processes, obtained by viewing $c = c(n)$ as the input into a linear, shift-invariant (LSI) filter.

6.3 Random Discrete Functions or Discrete Random Processes

A linear, shift-invariant (LSI) operator T with impulse response function $h = \{h(k)\}$ operates on any input sequence $x = \{x(n)\}$ to produce the output sequence $y = \{y(n)\}$ according to the convolution formula

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) = \sum_{k=-\infty}^{\infty} x(k)h(n-k). \tag{6.1}$$

We learn more about the system that T represents when we select as input sinusoids at fixed frequencies. Let ω be a fixed frequency in the interval $[-\pi, \pi)$ and let $x = e_\omega$, so that $x(n) = e^{in\omega}$ for each integer n. Then Equation (6.1) shows us that the output is

$$y(n) = H(e^{i\omega})x(n),$$

where

$$H(e^{i\omega}) = \sum_{k=-\infty}^{\infty} h(k)e^{-ik\omega}. \tag{6.2}$$

This function of ω is called the *frequency-response function* of the system. We can learn even more about the system by selecting as input the sequence $x(n) = z^n$, where z is an arbitrary complex number. Then Equation (6.1) gives the output as

$$y(n) = H(z)x(n),$$

where

$$H(z) = \sum_{k=-\infty}^{\infty} h(k)z^{-k}. \tag{6.3}$$

Note that if we select $z = e^{i\omega}$ then $H(z) = H(e^{i\omega})$ as given by Equation (6.2). The function $H(z)$ of the complex variable z is the z-transform of the sequence h and also the *transfer function* of the system determined by h.

Now we take this approach one step further. Let us select as our input $x = \{x(n)\}$ the random-coin-flip sequence $c = \{c(n)\}$, with $p = 0.5$. It is important to note that such an x is not one specific discrete function, but a random model for such functions. The output $y = \{y(n)\}$ is again a random sequence, with

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)c(n-k). \tag{6.4}$$

Clearly, in order for the infinite sum to converge we would need to place restrictions on the sequence h; if $h(k)$ is zero except for finitely many values of k then we have no problem. We shall put off discussion of convergence issues and focus on statistical properties of the output random sequence y.

Let u and v be (possibly complex-valued) random variables with expected values $E(u)$ and $E(v)$, respectively. The covariance between u and v is defined to be

$$cov(u,v) = E([u - E(u))\overline{(v - E(v))}]),$$

and the cross-correlation between u and v is

$$corr(u,v) = E(u\overline{v}).$$

It is easily shown that $cov(u,v) = corr(u,v) - E(u)\overline{E(v)}$. When $u = v$ we get $cov(u,u) = var(u)$ and $corr(u,u) = E(|u|^2)$. If $E(u) = E(v) = 0$ then $cov(u,v) = corr(u,v)$.

To illustrate, let $u = c(n)$ and $v = c(n-m)$. Then, since the coin is fair, $E(c(n)) = E(c(n-m)) = 0$ and

$$cov(c(n), c(n-m)) = corr(c(n), c(n-m)) = E(c(n)\overline{c(n-m)}).$$

Because the $c(n)$ are independent, $E(c(n)\overline{c(n-m)}) = 0$ for m not equal to 0, and $E(|c(n)|^2) = var(c(n)) = 1$. Therefore

$$cov(c(n), c(n-m)) = corr(c(n), c(n-m)) = 0, \mathrm{for}\, m \neq 0,$$

and

$$cov(c(n), c(n)) = corr(c(n), c(n)) = 1.$$

Returning now to the output sequence $y = \{y(n)\}$ we compute the correlation $corr(y(n), y(n-m)) = E(y(n)\overline{y(n-m)})$. Using the convolution formula Equation (6.4) we find that

$$corr(y(n), y(n-m)) = \sum_{k=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} h(k)\overline{h(j)}corr(c(n-k), c(n-m-j)).$$

Since

$$corr(c(n-k), c(n-m-j)) = 0, \mathrm{for}\, k \neq m+j,$$

we have

$$corr(y(n), y(n-m)) = \sum_{k=-\infty}^{\infty} h(k)\overline{h(k-m)}. \tag{6.5}$$

The expression of the right side of Equation (6.5) is the definition of the *autocorrelaton* of the sequence h, denoted $\rho_h(m)$; that is,

$$\rho_h(m) = \sum_{k=-\infty}^{\infty} h(k)\overline{h(k-m)}. \tag{6.6}$$

It is important to note that the expected value of $y(n)$ is

$$E(y(n)) = \sum_{k=-\infty}^{\infty} h(k)E(c(n-k)) = 0$$

and the correlation $corr(y(n), y(n-m))$ depends only on m; neither quantity depends on n and the sequence y is therefore called *weak-sense stationary.* Let's consider an example.

Take $h(0) = h(1) = 0.5$ and $h(k) = 0$ otherwise. Then the system is the two-point moving-average, with

$$y(n) = 0.5x(n) + 0.5x(n-1).$$

With $x(n) = c(n)$ we have

$$y(n) = 0.5c(n) + 0.5c(n-1).$$

In the case of the random-coin-flip sequence c each $c(n)$ is unrelated to any other $c(m)$; the coin flips are independent. This is no longer the case for the $y(n)$; one effect of the filter h is to introduce correlation into the output. To illustrate, since $y(0)$ and $y(1)$ both depend, to some degree, on the value $c(0)$, they are related. Using Equation (6.6) we have

$$\rho_h(0) = h(0)h(0) + h(1)h(1) = 0.25 + 0.25 = 0.5,$$

$$\rho_h(-1) = h(0)h(1) = 0.25,$$

$$\rho_h(+1) = h(1)h(0) = 0.25,$$

and

$$\rho_h(m) = 0, \text{otherwise}.$$

So we see that $y(n)$ and $y(n-m)$ are related, for $m = -1, 0, +1$, but not otherwise.

6.4 Correlation Functions and Power Spectra

As we have seen, any nonrandom sequence $h = \{h(k)\}$ has its autocorrelation function defined, for each integer m, by

$$\rho_h(m) = \sum_{k=-\infty}^{\infty} h(k)\overline{h(k-m)}.$$

For a random sequence $y(n)$ that is wide-sense stationary, its correlation function is defined to be

$$\rho_y(m) = E(y(n)\overline{y(n-m)}).$$

The *power spectrum* of h is defined for ω in $[-\pi, \pi]$ by

$$S_h(\omega) = \sum_{m=-\infty}^{\infty} \rho_h(m)e^{-im\omega}.$$

It is easy to see that

$$S_h(\omega) = |H(e^{i\omega})|^2,$$

so that $S_h(\omega) \geq 0$. The power spectrum of the random sequence $y = \{y(n)\}$ is defined as

$$S_y(\omega) = \sum_{m=-\infty}^{\infty} \rho_y(m)e^{-im\omega}.$$

Although it is not immediately obvious, we also have $S_y(\omega) \geq 0$. One way to see this is to consider

$$Y(e^{i\omega}) = \sum_{n=-\infty}^{\infty} y(n)e^{-in\omega}$$

and to calculate

$$E(|Y(e^{i\omega})|^2) = \sum_{m=-\infty}^{\infty} E(y(n)\overline{y(n-m)})e^{-im\omega} = S_y(\omega).$$

Given any power spectrum $S_y(\omega)$ we can construct $H(e^{i\omega})$ by selecting an arbitrary phase angle θ and letting

$$H(e^{i\omega}) = \sqrt{S_y(\omega)}e^{i\theta}.$$

We then obtain the nonrandom sequence h associated with $H(e^{i\omega})$ using

$$h(n) = \int_{-\pi}^{\pi} H(e^{i\omega})e^{in\omega}d\omega/2\pi.$$

It follows that $\rho_h(m) = \rho_y(m)$ for each m and $S_h(\omega) = S_y(\omega)$ for each ω.

What we have discovered is that, when the input to the system is the random-coin-flip sequence c, the output sequence y has a correlation function $\rho_y(m)$ that is equal to the autocorrelation of the sequence h. As we just saw, for any weak-sense stationary random sequence y with expected value $E(y(n))$ constant and correlation function $corr(y(n), y(n-m))$ independent of n, there is a LSI system h with $\rho_h(m) = \rho_y(m)$ for each m. Therefore, any weak-sense stationary random sequence y can be viewed as the output of an LSI system, when the input is the random-coin-flip sequence $c = \{c(n)\}$.

6.5 Random Sinusoidal Sequences

If $A = |A|e^{i\theta}$, with amplitude $|A|$ a positive-valued random variable and phase angle θ a random variable taking values in the interval $[-\pi, \pi]$ then A is a complex-valued random variable. For a fixed frequency ω_0 we define a random sinusoidal sequence $s = \{s(n)\}$ by $s(n) = Ae^{in\omega_0}$. We assume that θ has the uniform distribution over $[-\pi, \pi]$ so that the expected value of $s(n)$ is zero. The correlation function for s is

$$\rho_s(m) = E(s(n)\overline{s(n-m)}) = E(|A|^2)e^{im\omega_0}$$

and the power spectrum of s is

$$S_s(\omega) = E(|A|^2) \sum_{m=-\infty}^{\infty} e^{im(\omega_0 - \omega)},$$

so that, by Equation (5.7), we have

$$S_s(\omega) = E(|A|^2)\delta(\omega - \omega_0).$$

We generalize this example to the case of multiple independent sinusoids. Suppose that, for $j = 1, ..., J$, we have fixed frequencies ω_j and independent complex-valued random variables A_j. We let our random sequence be defined by

$$s(n) = \sum_{j=1}^{J} A_j e^{in\omega_j}.$$

Then the correlation function for x is

$$\rho_s(m) = \sum_{j=1}^{J} E(|A_j|^2)e^{im\omega_j}$$

and the power spectrum for s is

$$S_s(\omega) = \sum_{j=1}^{J} E(|A_j|^2)\delta(\omega - \omega_j).$$

A commonly used model in signal processing is that of independent sinusoids in additive noise.

Let $q = \{q(n)\}$ be an arbitrary weak-sense stationary discrete random sequence, with correlation function $\rho_q(m)$ and power spectrum $S_q(\omega)$. We say that q is white noise if $\rho_q(m) = 0$ for m not equal to zero, or, equivalently, if the power spectrum $S_q(\omega)$ is constant over the interval $[-\pi, \pi]$. The *independent sinusoids in additive noise* model is a random sequence of the form

$$x(n) = \sum_{j=1}^{J} A_j e^{in\omega_j} + q(n).$$

The *signal power* is defined to be $\rho_s(0)$, which is the sum of the $E(|A_j|^2)$, while the noise power is $\rho_q(0)$. The *signal-to-noise ratio* (SNR) is the ratio of signal power to noise power.

It is often the case that the SNR is quite low and it is desirable to process the x to enhance this ratio. The data we have is typically finitely many values of $x(n)$, say for $n = 1, 2, ..., N$. One way to process the data is to estimate $\rho_x(m)$ for some small number of integers m around zero, using, for example, the *lag products* estimate

$$\hat{\rho}_x(m) = \frac{1}{N-m} \sum_{n=1}^{N-m} x(n)\overline{x(n-m)},$$

for $m = 0, 1, ..., M < N$ and $\hat{\rho}_x(-m) = \overline{\hat{\rho}_x(m)}$. Because $\rho_q(m) = 0$ for m not equal to zero, we will have $\hat{\rho}_x(m)$ approximating $\rho_s(m)$ for nonzero values of m, thereby reducing the effect of the noise.

The additive noise is said to be *correlated* or *non-white* if it is not the case that $\rho_x(m) = 0$ for all nonzero m. In this case the noise power spectrum is not constant, and so may be concentrated in certain regions of the interval $[-\pi, \pi]$.

6.6 Spread-Spectrum Communication

In this section we return to the random-coin-flip model, this time allowing the coin to be biased, that is, p need not be 0.5. Let $s = \{s(n)\}$ be a random sequence, such as $s(n) = Ae^{in\omega_0}$, with $E(s(n)) = \mu$ and correlation function $\rho_s(m)$. Define a second random sequence x by

$$x(n) = s(n)c(n).$$

The random sequence x is generated from the random signal s by randomly changing its signs. We can show that

$$E(x(n)) = \mu(2p-1)$$

and, for m not equal to zero,

$$\rho_x(m) = \rho_s(m)(2p-1)^2,$$

with $\rho_x(0) = \rho_s(0) + 4p(1-p)\mu^2$. Therefore, if $p = 1$ or $p = 0$ we get $\rho_x(m) = \rho_s(m)$ for all m, but for $p = 0.5$ we get $\rho_x(m) = 0$ for m not equal to zero. If the coin is unbiased, then the random sign changes convert the original signal s into white noise. Generally, we have

$$S_x(\omega) = (2p-1)^2 S_s(\omega) + (1-(2p-1)^2)(\mu^2 + \rho_s(0)),$$

which says that the power spectrum of x is a combination of the signal power spectrum and a white-noise power spectrum, approaching the white-noise power spectrum as p approaches 0.5. If the original signal power spectrum is concentrated within a small interval, then the effect of the random sign changes is to spread that spectrum. Once we know what the sequence c is we can recapture the original signal from $s(n) = x(n)c(n)$. The use of such a spread spectrum permits the sending of multiple narrow-band signals, without confusion, as well as protecting against any narrow-band additive interference.

6.7 Stochastic Difference Equations

The ordinary first-order differential equation $y'(t) + ay(t) = f(t)$, with initial condition $y(0) = 0$, has for its solution $y(t) = e^{-at}\int_0^t e^{as} f(s)ds$. One way to look at such differential equations is to consider $f(t)$ to be the input to a system having $y(t)$ as its output. The system determines which terms will occur on the left side of the differential equation. In many applications the input $f(t)$ is viewed as random noise and the output is then a continuous-time random process. Here we want to consider the discrete analog of such differential equations.

We replace the first derivative with the first difference, $y(n+1) - y(n)$ and we replace the input with the random-coin-flip sequence $c = \{c(n)\}$, to obtain the random difference equation

$$y(n+1) - y(n) + ay(n) = c(n). \tag{6.7}$$

With $b = 1 - a$ and $0 < b < 1$ we have

$$y(n+1) - by(n) = c(n). \tag{6.8}$$

The solution is $y = \{y(n)\}$ given by

$$y(n) = b^n \sum_{k=-\infty}^{n} b^{-k} c(k). \tag{6.9}$$

Comparing this with the solution of the differential equation, we see that the term b^n plays the role of $e^{-at} = (e^{-a})^t$, so that $b = 1 - a$ is substituting for e^{-a}. The infinite sum replaces the infinite integral, with $b^{-k}c(k)$ replacing the integrand $e^{as}f(s)$.

The solution sequence y given by Equation (6.9) is a weak-sense stationary random sequence and its correlation function is

$$\rho_y(m) = b^m/(1 - b^2).$$

Since

$$b^n \sum_{k=-\infty}^{n} b^{-k} = 1 - b$$

the random sequence $(1 - b)^{-1}y(n)$ is an infinite *moving-average* random sequence formed from the random sequence c.

We can derive the solution in Equation (6.9) using z-transforms. The expression $y(n) - by(n - 1)$ can be viewed as the output of a LSI system with $h(0) = 1$ and $h(1) = -b$. Then $H(z) = 1 - bz^{-1} = (z - b)/z$ and the inverse $H(z)^{-1} = z/(z - b)$ describes the inverse system. Since

$$H(z)^{-1} = z/(z - b) = 1/(1 - bz^{-1}) = 1 + bz^{-1} + b^2z^{-2} + ...$$

the inverse system applied to input $c = \{c(n)\}$ is

$$y(n) = c(n) + bc(n - 1) + b^2c(n - 2) + ... = b^n \sum_{k=-\infty}^{n} b^{-k} c(k).$$

6.8 Random Vectors and Correlation Matrices

In estimation and detection theory, the task is to distinguish *signal vectors* from *noise vectors*. In order to perform such a task, we need to know how signal vectors differ from noise vectors. Most frequently, what we have is statistical information. The signal vectors of interest, which we denote by $s = (s_1, ..., s_N)^T$, typically exhibit some patterns of behavior among their entries. For example, a constant signal, such as $s = (1, 1, ..., 1)^T$, has all its entries identical. A sinusoidal signal, such as $s = (1, -1, 1, -1, ..., 1, -1)^T$, exhibits a periodicity in its entries. If the signal is a vectorization of a two-dimensional image, then the patterns will be more difficult to describe, but will be there, nevertheless. In contrast, a typical noise vector, denoted $q = (q_1, ..., q_N)^T$, may have entries that are unrelated to each other, as in white noise. Of course, what is signal and what is noise depends on the context; unwanted interference in radio may be viewed as noise, even though it may be a weather report or a song.

To deal with these notions mathematically, we adopt statistical models. The entries of s and q are taken to be random variables, so that s and q are random vectors. Often we assume that the mean values, $E(s)$ and $E(q)$, are zero. Then patterns that may exist among the entries of these vectors are described in terms of *correlations*. The *noise covariance matrix*, which we denote by Q, has for its entries $Q_{mn} = E((q_m - E(q_m))\overline{(q_n - E(q_n))})$, for $m, n = 1, ..., N$. The signal covariance matrix is defined similarly. If $E(q_n) = 0$ and $E(|q_n|^2) = 1$ for each n, then Q is the *noise correlation matrix*. Such matrices Q are Hermitian and non-negative definite, that is, $x^\dagger Qx$ is non-negative, for every vector x. If Q is a positive multiple of the identity matrix, then the noise is said to be *white noise*.

Chapter 7

Estimation, Detection, Discrimination, and Classification

In some applications of remote sensing, our goal is simply to see what is "out there"; in sonar mapping of the sea floor, the data are the acoustic signals as reflected from the bottom, from which the changes in depth can be inferred. Such problems are *estimation* problems.

In other applications, such as sonar target detection or medical diagnostic imaging, we are looking for certain things, evidence of a surface vessel or submarine, in the sonar case, or a tumor or other abnormality in the medical case. These are *detection* problems. In the sonar case, the data may be used directly in the detection task, or may be processed in some way, perhaps frequency-filtered, prior to being used for detection. In the medical case, or in synthetic-aperture radar (SAR), the data is usually used to construct an image, which is then used for the detection task. In estimation, the goal can be to determine how much of something is present; detection is then a special case, in which we want to decide if the amount present is zero or not.

The detection problem is also a special case of *discrimination*, in which the goal is to decide which of two possibilities is true; in detection the possibilities are simply the presence or absence of the sought-for signal.

More generally, in *classification* or *identification*, the objective is to decide, on the basis of measured data, which of several possibilities is true.

7.1 Estimation

We consider only estimates that are linear in the data, that is, estimates of the form

$$\hat{\gamma} = b^\dagger x = \sum_{n=1}^{N} \overline{b_n} x_n, \tag{7.1}$$

where $x = (x_1, ..., x_N)^T$ is the vector of data and $b^\dagger$ denotes the conjugate transpose of the vector $b = (b_1, ..., b_N)^T$. The vector b that we use will be the *best linear unbiased estimator* (BLUE) [56] for the particular estimation problem.

7.1.1 The simplest case: a constant in noise

We begin with the simplest case, estimating the value of a constant, given several instances of the constant in additive noise. Our data are $x_n = \gamma + q_n$, for $n = 1, ..., N$, where γ is the constant to be estimated, and the q_n are noises. For convenience, we write

$$x = \gamma u + q, \tag{7.2}$$

where $x = (x_1, ..., x_N)^T$, $q = (q_1, ..., q_N)^T$, $u = (1, ..., 1)^T$, the expected value of the random vector q is $E(q) = 0$, and the covariance matrix of q is $E(qq^T) = Q$. The BLUE employs the vector

$$b = \frac{1}{u^\dagger Q^{-1} u} Q^{-1} u. \tag{7.3}$$

The BLUE estimate of γ is

$$\hat{\gamma} = \frac{1}{u^\dagger Q^{-1} u} u^\dagger Q^{-1} x. \tag{7.4}$$

If $Q = \sigma^2 I$, for some $\sigma > 0$, with I the identity matrix, then the noise q is said to be *white*. In this case, the BLUE estimate of γ is simply the average of the x_n.

7.1.2 A known signal vector in noise

Generalizing somewhat, we consider the case in which the data vector x has the form

$$x = \gamma s + q, \tag{7.5}$$

where $s = (s_1, ..., s_N)^T$ is a known signal vector. The BLUE estimator is

$$b = \frac{1}{s^\dagger Q^{-1} s} Q^{-1} s \tag{7.6}$$

and the BLUE estimate of γ is now

$$\hat{\gamma} = \frac{1}{s^\dagger Q^{-1} s} s^\dagger Q^{-1} x. \tag{7.7}$$

In numerous applications of signal processing, the signal vectors take the form of sampled sinusoids; that is, $s = e_\theta$, with

$$e_\theta = \frac{1}{\sqrt{N}}(e^{-i\theta}, e^{-2i\theta}, ..., e^{-Ni\theta})^T, \tag{7.8}$$

where θ is a frequency in the interval $[0, 2\pi)$. If the noise is white, then the BLUE estimate of γ is

$$\hat{\gamma} = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} x_n e^{in\theta}, \tag{7.9}$$

which is the *discrete Fourier transform*(DFT) of the data, evaluated at the frequency θ.

7.1.3 Multiple signals in noise

Suppose now that the data values are

$$x_n = \sum_{m=1}^{M} \gamma_m s_n^m + q_n, \tag{7.10}$$

where the signal vectors $s^m = (s_1^m, ..., s_N^m)^T$ are known and we want to estimate the γ_m. We write this in matrix-vector notation as

$$x = Sc + q, \tag{7.11}$$

where S is the matrix with entries $S_{nm} = s_n^m$, and our goal is to find $c = (\gamma_1, ..., \gamma_N)^T$, the vector of coefficients. The BLUE estimate of the vector c is

$$\hat{c} = (S^\dagger Q^{-1} S)^{-1} S^\dagger Q^{-1} x, \tag{7.12}$$

assuming that the matrix $S^\dagger Q^{-1} S$ is invertible, in which case we must have $M \leq N$.

If the signals s^m are mutually orthogonal and have length one, then $S^\dagger S = I$; if, in addition, the noise is white, the BLUE estimate of c is $\hat{c} = S^\dagger x$, so that

$$\hat{c}_m = \sum_{n=1}^{N} x_n \overline{s_n^m}. \tag{7.13}$$

This case arises when the signals are $s^m = e_{\theta_m}$, for $\theta_m = 2\pi m/M$, for $m = 1, ..., M$, in which case the BLUE estimate of c_m is

$$\hat{c}_m = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} x_n e^{2\pi imn/M}, \tag{7.14}$$

the DFT of the data, evaluated at the frequency θ_m. Note that when the frequencies θ_m are not these, the matrix $S^\dagger S$ is not I, and the BLUE estimate is not obtained from the DFT of the data.

7.2 Detection

As we noted previously, the detection problem is a special case of estimation. Detecting the known signal s in noise is equivalent to deciding if the coefficient γ is zero or not. The procedure is to calculate $\hat{\gamma}$, the BLUE estimate of γ, and say that s has been detected if $|\hat{\gamma}|$ exceeds a certain threshold. In the case of multiple known signals, we calculate $\hat{c}$, the BLUE estimate of the coefficient vector c, and base our decisions on the magnitudes of each entry of $\hat{c}$.

7.2.1 Parametrized signal

It is sometimes the case that we know that the signal s we seek to detect is a member of a parametrized family, $\{s_\theta | \theta \in \Theta\}$, of potential signal vectors, but we do not know the value of the parameter θ. For example, we may be trying to detect a sinusoidal signal, $s = e_\theta$, where θ is an unknown frequency in the interval $[0, 2\pi)$. In sonar direction-of-arrival estimation, we seek to detect a farfield point source of acoustic energy, but do not know the direction of the source. The BLUE estimator can be extended to these cases, as well [56]. For each fixed value of the parameter θ, we estimate γ using the BLUE, obtaining the estimate

$$\hat{\gamma}(\theta) = \frac{1}{s_\theta^\dagger Q^{-1} s_\theta} s_\theta^\dagger Q^{-1} x, \tag{7.15}$$

which is then a function of θ. If the maximum of the magnitude of this function exceeds a specified threshold, then we may say that there is a signal present corresponding to that value of θ.

Another approach would be to extend the model of multiple signals to include a continuum of possibilities, replacing the finite sum with an integral. Then the model of the data becomes

$$x = \int_{\theta \in \Theta} \gamma(\theta) s_\theta d\theta + q. \tag{7.16}$$

Let S now denote the integral operator

$$S(\gamma) = \int_{\theta\in\Theta} \gamma(\theta) s_\theta d\theta \tag{7.17}$$

that transforms a function γ of the variable θ into a vector. The adjoint operator, $S^\dagger$, transforms any N-vector v into a function of θ, according to

$$S^\dagger(v)(\theta) = \sum_{n=1}^{N} v_n \overline{(s_\theta)_n} = s_\theta^\dagger v\,. \tag{7.18}$$

Consequently, $S^\dagger Q^{-1} S$ is the function of θ given by

$$g(\theta) = (S^\dagger Q^{-1} S)(\theta) = \sum_{n=1}^{N} \sum_{j=1}^{N} Q_{nj}^{-1} (s_\theta)_j \overline{(s_\theta)_n}, \tag{7.19}$$

so

$$g(\theta) = s_\theta^\dagger Q^{-1} s_\theta. \tag{7.20}$$

The generalized BLUE estimate of $\gamma(\theta)$ is then

$$\hat\gamma(\theta) = \frac{1}{g(\theta)} \sum_{j=1}^{N} a_j \overline{(s_\theta)_j} = \frac{1}{g(\theta)} s_\theta^\dagger a\,, \tag{7.21}$$

where $x = Qa$ or

$$x_n = \sum_{j=1}^{N} a_j Q_{nj}, \tag{7.22}$$

for $j = 1, ..., N$, and so $a = Q^{-1}x$. This is the same estimate we obtained in the previous paragraph. The only difference is that, in the first case, we assume that there is only one signal active, and apply the BLUE for each fixed θ, looking for the one most likely to be active. In the second case, we choose to view the data as a noisy superposition of a continuum of the s_θ, not just one. The resulting estimate of $\gamma(\theta)$ describes how each of the individual signal vectors s_θ contribute to the data vector x. Nevertheless, the calculations we perform are the same.

If the noise is white, we have $a_j = x_j$ for each j. The function $g(\theta)$ becomes

$$g(\theta) = \sum_{n=1}^{N} |(s_\theta)_n|^2, \tag{7.23}$$

which is simply the square of the length of the vector s_θ. If, in addition, the signal vectors all have length one, then the estimate of the function $\gamma(\theta)$ becomes

$$\hat{\gamma}(\theta) = \sum_{n=1}^{N} x_n \overline{(s_\theta)_n} = s_\theta^\dagger x. \tag{7.24}$$

Finally, if the signals are sinusoids $s_\theta = e_\theta$, then

$$\hat{\gamma}(\theta) = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} x_n e^{in\theta}, \tag{7.25}$$

again, the DFT of the data vector.

7.3 Discrimination

The problem now is to decide if the data is $x = s^1 + q$ or $x = s^2 + q$, where s^1 and s^2 are known vectors. This problem can be converted into a detection problem: Do we have $x - s^1 = q$ or $x - s^1 = s^2 - s^1 + q$? Then the BLUE involves the vector $Q^{-1}(s^2 - s^1)$ and the discrimination is made based on the quantity $(s^2 - s^1)^\dagger Q^{-1} x$. If this quantity is near enough to zero we say that the signal is s^1; otherwise, we say that it is s^2. The BLUE in this case is sometimes called the *Hotelling linear discriminant*, and a procedure that uses this method to perform medical diagnostics is called a *Hotelling observer*.

More generally, suppose we want to decide if a given vector x comes from class C_1 or from class C_2. If we can find a vector b such that $b^T x > a$ for every x that comes from C_1, and $b^T x < a$ for every x that comes from C_2, then the vector b is a linear discriminant for deciding between the classes C_1 and C_2.

7.3.1 Channelized Observers

The N by N matrix Q can be quite large, particularly when x and q are vectorizations of two-dimensional images. If, in additional, the matrix Q is obtained from K observed instances of the random vector q, then for Q to be invertible, we need $K \geq N$. To avoid these and other difficulties, the *channelized* Hotelling linear discriminant is often used. The idea here is to replace the data vector x with Ux for an appropriately chosen J by N matrix U, with J much smaller than N; the value $J = 3$ is used in [110], with the channels chosen to capture image information within selected frequency bands.

7.3.2 An Example of Discrimination

Suppose that there are two groups of students, the first group denoted G_1, the second G_2. The math SAT score for the students in G_1 is always above 500, while their verbal scores are always below 500. For the students in G_2 the opposite is true; the math scores are below 500, the verbal above. For each student we create the two-dimensional vector $x = (x_1, x_2)^T$ of SAT scores, with x_1 the math score, x_2 the verbal score. Let $b = (1, -1)^T$. Then for every student in G_1 we have $b^T x > 0$, while for those in G_2, we have $b^T x < 0$. Therefore, the vector b provides a linear discriminant.

Suppose we have a third group, G_3, whose math scores and verbal scores are both below 500. To discriminate between members of G_1 and G_3 we can use the vector $b = (1, 0)^T$ and $a = 500$. To discriminate between the groups G_2 and G_3, we can use the vector $b = (0, 1)^T$ and $a = 500$.

Now suppose that we want to decide from which of the three groups the vector x comes; this is classification.

7.4 Classification

The classification problem is to determine to which of several classes of vectors a given vector x belongs. For simplicity, we assume all vectors are real. The simplest approach to solving this problem is to seek linear discriminant functions; that is, for each class we want to have a vector b with the property that $b^T x > 0$ if and only if x is in the class. If the vectors x are randomly distributed according to one of the parametrized family of probability density functions (pdf) $p(x; \omega)$ and the ith class corresponds to the parameter value ω_i then we can often determine the discriminant vectors b^i from these pdf. In many cases, however, we do not have the pdf and the b^i must be estimated through a learning or training step before they are used on as yet unclassified data vectors. In the discussion that follows we focus on obtaining b for one class, suppressing the index i.

7.4.1 The Training Stage

In the training stage a candidate for b is tested on vectors whose class membership is known, say $\{x^1, ..., x^M\}$. First, we replace each vector x^m that is not in the class with its negative. Then we seek b such that $b^T x^m > 0$ for all m. With A the matrix whose mth row is $(x^m)^T$ we can write the problem as $Ab > 0$. If the b we obtain has some entries very close to zero it might not work well enough on actual data; it is often better, then, to take a vector ϵ with small positive entries and require $Ab \geq \epsilon$. When we have found b for each class we then have the machinery to perform the classification task.

There are several problems to be overcome, obviously. The main one is that there may not be a vector b for each class; the problem $Ab \geq \epsilon$ need not have a solution. In classification this is described by saying that the vectors x^m are not linearly separable [90]. The second problem is finding the b for each class; we need an algorithm to solve $Ab \geq \epsilon$.

One approach to designing an algorithm for finding b is the following: for arbitrary b let $f(b)$ be the number of the x^m misclassified by vector b. Then minimize $f(b)$ with respect to b. Alternatively, we can minimize the function $g(b)$ defined to be the sum of the values $-b^T x^m$, taken over all the x^m that are misclassified; the $g(b)$ has the advantage of being continuously valued. The batch Perceptron algorithm [90] uses gradient descent methods to minimize $g(b)$. Another approach is to use the Agmon-Motzkin-Schoenberg (AMS) algorithm to solve the system of linear inequalities $Ab \geq \epsilon$ [56].

When the training set of vectors is linearly separable, the batch Perceptron and the AMS algorithms converge to a solution, for each class. When the training vectors are not linearly separable there will be a class for which the problem $Ab \geq \epsilon$ will have no solution. Iterative algorithms in this case cannot converge to a solution. Instead, they may converge to an approximate solution or, as with the AMS algorithm, converge subsequentially to a limit cycle of more than one vector.

7.4.2 Our Example Again

We return to the example given earlier, involving the three groups of students and their SAT scores. To be consistent with the conventions of this section, we define $x = (x_1, x_2)^T$ differently now. Let x_1 be the math SAT score, minus 500, and x_2 be the verbal SAT score, minus 500. The vector $b = (1, 0)^T$ has the property that $b^T x > 0$ for each x coming from G_1, but $b^T x < 0$ for each x not coming from G_1. Similarly, the vector $b = (0, 1)^T$ has the property that $b^T x > 0$ for all x coming from G_2, while $b^T x < 0$ for all x not coming from G_2. However, there is no vector b with the property that $b^T x > 0$ for x coming from G_3, but $b^T x < 0$ for all x not coming from G_3; the group G_3 is not linearly separable from the others. Notice, however, that if we perform our classification sequentially, we can employ linear classifiers. First, we use the vector $b = (1, 0)^T$ to decide if the vector x comes from G_1 or not. If it does, fine; if not, then use vector $b = (0, 1)^T$ to decide if it comes from G_2 or G_3.

7.5 More realistic models

In many important estimation and detection problems, the signal vector s is not known precisely. In medical diagnostics, we may be trying to detect a lesion, and may know it when we see it, but may not be able to describe it using a single vector s, which now would be a vectorized image. Similarly, in discrimination or classification problems, we may have several examples of each type we wish to identify, but will be unable to reduce these types to single representative vectors. We now have to derive an analog of the BLUE that is optimal with respect to the examples that have been presented for training. The linear procedure we seek will be one that has performed best, with respect to a training set of examples. The *Fisher linear discriminant* is an example of such a procedure.

7.5.1 The Fisher linear discriminant

Suppose that we have available for training K vectors $x^1, ..., x^K$ in R^N, with vectors $x^1, ..., x^J$ in the class A, and the remaining $K-J$ vectors in the class B. Let w be an arbitrary vector of length one, and for each k let $y_k = w^T x^k$ be the projected data. The numbers y_k, $k = 1, ..., J$, form the set Y_A, the remaining ones the set Y_B. Let

$$\mu_A = \frac{1}{J}\sum_{k=1}^{J} x^k, \tag{7.26}$$

$$\mu_B = \frac{1}{K-J}\sum_{k=J+1}^{K} x^k, \tag{7.27}$$

$$m_A = \frac{1}{J}\sum_{k=1}^{J} y_k = w^T \mu_A, \tag{7.28}$$

and

$$m_B = \frac{1}{K-J}\sum_{k=J+1}^{K} y_k = w^T \mu_B. \tag{7.29}$$

Let

$$\sigma_A^2 = \sum_{k=1}^{J} (y_k - m_A)^2, \tag{7.30}$$

and

$$\sigma_B^2 = \sum_{k=J+1}^{K} (y_k - m_B)^2. \tag{7.31}$$

The quantity $\sigma^2 = \sigma_A^2 + \sigma_B^2$ is the *total within-class scatter* of the projected data. Define the function $F(w)$ to be

$$F(w) = \frac{(m_A - m_B)^2}{\sigma^2}. \tag{7.32}$$

The *Fisher linear discriminant* is the vector w for which $F(w)$ achieves its maximum.

Define the scatter matrices S_A and S_B as follows:

$$S_A = \sum_{k=1}^{J} (x^k - \mu_A)(x^k - \mu_A)^T, \tag{7.33}$$

and

$$S_B = \sum_{k=J+1}^{K} (x^k - \mu_B)(x^k - \mu_B)^T. \tag{7.34}$$

Then

$$S_{within} = S_A + S_B \tag{7.35}$$

is the *within-class scatter matrix* and

$$S_{between} = (\mu_A - \mu_B)(\mu_A - \mu_B)^T \tag{7.36}$$

is the *between-class scatter matrix.* The function $F(w)$ can then be written as

$$F(w) = w^T S_{between} w / w^T S_{within} w. \tag{7.37}$$

The w for which $F(w)$ achieves its maximum value is then

$$w = S_{within}^{-1}(\mu_A - \mu_B). \tag{7.38}$$

This vector w is Fisher linear discriminant. When a new data vector x is obtained, we decide to which of the two classes it belongs by calculating $w^T x$.

7.6 A more general estimation problem

It is often the case, in practice, that the object of interest is a function of one or several continuous variables, and our data consists of finitely many linear functional values. For example, suppose that our object of interest is the function of two real variables $f(u, v)$, and that our data are the values

$$x_n = \int\int f(u, v) h_n(u, v) du dv + q_n, \tag{7.39}$$

for noise q_n and known functions $h_n(u,v)$, $n = 1, ..., N$. Our goal may be to reconstruct the function $f(u,v)$ itself, or, more modestly, to estimate some other linear functional value, $\int\int f(u,v)g(u,v)dudv$, such as the integral of $f(u,v)$ over some two-dimensional set A. We consider only estimates that are linear in the data x. Unfortunately, we can obtain an unbiased estimate of $\int\int f(u,v)g(u,v)dudv$ only if we can calculate $\int\int f(u,v)g(u,v)dudv$ from noise-free data, for any $f(u,v)$, which can be done only if the function $g(u,v)$ has the form

$$g(u,v) = \sum_{n=1}^{N} a_n h_n(u,v), \tag{7.40}$$

for some constants a_n. This rather negative result suggests that the information about $f(u,v)$ that we can expect to extract from the data is quite limited. On the other hand, if we should know, in advance, that $f(u,v)$ is a member of a parametrized family of functions and if the data is sufficient to calculate the parameter, then not only can we estimate $\int\int f(u,v)g(u,v)dudv$ from the data, for every $g(u,v)$, but we can determine $f(u,v)$ itself.

To investigate this problem further, we assume that f and the h_n are members of a Hilbert space X, such as $L^2(R)$ or $L^2(R^2)$. Since the problem of obtaining an unbiased linear estimate is equivalent to that of achieving perfect reconstruction from noise-free data, we assume that the data we have are

$$x_n = \langle f, h_n \rangle, \tag{7.41}$$

where $\langle a, b \rangle$ denotes the inner product in the space X. For $X = L^2(R^2)$ we have

$$\langle a, b \rangle = \int\int a(u,v)\overline{b(u,v)}dudv. \tag{7.42}$$

The goal is to reconstruct the linear functional $\langle f, g \rangle$ as a linear combination of the entries of the data vector x.

Each g in X can be written in the form

$$g = \sum_{n=1}^{N} c_n h_n + z, \tag{7.43}$$

for some choice of constants c_n and some z with the property that

$$\langle z, h_n \rangle = 0, \tag{7.44}$$

for each n. Then we have

$$\langle f, g \rangle = \sum_{n=1}^{N} c_n \langle f, h_n \rangle + \langle f, z \rangle = \sum_{n=1}^{N} c_n x_n + \langle f, z \rangle. \tag{7.45}$$

The problem then is that we cannot determine the quantity $\langle f, z\rangle$ from the data, in general.

However, if it should be the case that f is a linear combination of the h_n, that is, there are constants a_n so that

$$f = \sum_{n=1}^{N} a_n h_n, \tag{7.46}$$

then $\langle f, z\rangle = 0$. But why should it be the case?

Notice that the data we have measured exists prior to the specification of the Hilbert space X. By choosing different Hilbert spaces, the data can be represented in different ways, using different inner products and different h_n. To make this somewhat abstract statement more concrete, consider the example of Fourier-transform data.

7.6.1 An Example: Fourier-Transform Data

Suppose that the object of interest is $f(r)$, a function of the single real variable r. Suppose that our data values are

$$x_n = F(\omega_n) = \int f(r) e^{-i\omega_n r} dr, \tag{7.47}$$

for $n = 1, ..., N$, and ω_n arbitrary frequencies. With $X = L^2(R)$, we can write

$$x_n = F(\omega_n) = \langle f, h_n\rangle, \tag{7.48}$$

for

$$h_n(r) = e^{i\omega_n r}. \tag{7.49}$$

Then we will have f in the span of the h_n if f can be written

$$f(r) = \sum_{n=1}^{N} a_n e^{i\omega_n r}, \tag{7.50}$$

for some constants a_n. However, unless N is very large, or the $h_n(r)$ have been carefully chosen, f will probably not be well described by such a sum.

But we should not give up! We can also write

$$x_n = \int f(r) p(r) e^{-i\omega_n r} p(r)^{-1} dr, \tag{7.51}$$

where $p(r) > 0$. If we define X now to be the Hilbert space with

$$\langle s, t\rangle = \int s(r)\overline{t(r)} p(r)^{-1} dr, \tag{7.52}$$

then

$$h_n(r) = p(r)e^{i\omega_n r}. \tag{7.53}$$

Now we will have f in the span of the h_n if

$$f(r) = p(r)\sum_{n=1}^{N} a_n e^{i\omega_n r}, \tag{7.54}$$

for some a_n. If we have prior knowledge about $f(r)$, or, more precisely, about $|f(r)|$, such as its support, or any prominent components that it may have, we can include them in a prior estimate $p(r)$ of $|f(r)|$, making it much more likely that f lies in the span of the h_n, or, at least, can be well approximately by members of this span.

This approach was developed for image reconstruction from Fourier data in [33, 34, 40]. In those papers it was called the PDFT estimator. See the appendix for more discussion of Fourier-transform estimation.

7.6.2 More Generally

In general, if we want to make it plausible that f lies in the span of the h_n, we can alter the ambient Hilbert space, and its inner product, so that the h_n that represent the data also have a good chance of capturing the desired f within their span. This freedom to tailor the Hilbert space to the f, using prior knowledge of f, is the *way out* that we need to overcome the negative result we saw early on.

7.7 Conclusions

We always have finite data. In the absence of additional knowledge about f, we can say little, unless the data set is large. But, in most reconstruction problems we do have additional information, often qualitiative, about the object f to be recovered. We may, for instance, be willing to say that f is well-approximated by a finite sum of pixels, voxels, or blobs. Finite data, if there is enough of it, will then suffice to recover f, at least approximately, from which we can calculate any desired linear-functional value. The example above, involving Fourier data, shows how we can use prior knowledge to tailor the ambient Hilbert space, to get beyond the negative earlier result. The negative result reinforces the point that there is no *one-size-fits-all* method that will work for all f, but for each individual f, if we have prior knowledge about it, all is not lost. There have been a great many papers stressing the importance of prior information in reconstruction from limited data [37, 81].

Chapter 8

Randomness in Tomography

There seems to be a tradition in physics of using simple models involving urns and marbles to illustrate important principles. In keeping with that tradition, we have here such a model, to illustrate various aspects of remote sensing. We begin with the model itself, and then give several examples to show how the model illustrates randomness in tomography.

8.1 The Urn Model

Although remote-sensing problems differ from one another in many respects, they often share a fundamental aspect that can best be illustrated by a simple model involving urns containing colored marbles.

8.1.1 The Model

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, in that what we have access to is related to, but not equal to, which we are interested in. 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 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.

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, such as the EMML, may be needed to find such solutions.

8.1.2 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.

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.

8.1.3 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 $y_i = \sum_{j=1}^{J} P_{ij}x_j$.

8.1.4 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.

8.2 Transmission Tomography

It is tempting to view tomographic problems as reconstruction from line-integral data, and to reconstruct using the non-iterative filtered back-projection (FBP) method. Although removing the randomness distorts the physics of the problem, this FBP approach is quick and does often lead to reasonable images. Recently, however, as computing speed has improved, iterative reconstruction algorithms have become competitive, permitting more sophisticated stochastic physical models to be used. These iterative algorithms are often optimization methods that maximize or minimize some objective function appropriate for the problem. These stochastic physical models, the associated objective functions, and the iterative algorithms for optimizing these functions are the topics of this chapter.

In our previous discussion of transmission tomography, we concentrated on the deterministic formulation of the problem, in which each data value is used to calculate the drop in intensity along the corresponding line segment. This intensity drop is then used to estimate the integral of the attenuation function along the given line segment. If we know the line integrals along every line segment through the object, then, by the Central Slice Theorem for the Radon transform, we know the Fourier transform of the attenuation function. The problem is solved by Fourier inversion, which can be implemented as filtered back-projection. As we noted earlier, there are several weaknesses in this line-integral data model. In addition, we have estimates of only finitely many line integrals. In practice, the filtered back-projection approach requires us to select a filter that provides good resolution, while not amplifying the noise; the resulting reconstructed image will depend on the filter we choose. Because the deterministic approach slights the physics, in favor of computational simplicity, it is reasonable to hope that, by incorporating more of the physics, one can obtain better reconstructed images. The paper by Peters [164] describes ways in which the line-integral model can be improved.

In 1976 Rockmore and Macovski suggested, in [172], that the stochastic nature of the problem be made part of the model. Following their suggestions, we begin by discretizing the problem, decomposing the slice through the body into pixels, indexed by $j = 1, ..., J$, and denoting by c_i the photon count received at the ith detector, located at the end of the ith line segment. Simplifying somewhat, we can say that the photon count c_i obeys Poisson statistics, with mean value

$$E(c_i) = b_i e^{-(Px)_i} + r_i, \tag{8.1}$$

where b_i is the expected number of photons entering the ith line segment, x_j is the intensity of the attenuation in the jth pixel, P_{ij} is the length of the intersection of the ith line segment with the jth pixel, and

$$(Px)_i = \sum_{j=1}^{J} P_{ij} x_j,$$

for each i. The r_i is the expected number of background counts.

If we ignore the presence of these background counts, replace the expected counts $E(c_i)$ on the left side of Equation (8.1) with the actual counts c_i, and take the logarithm on both sides, we obtain

$$y_i = \log(b_i/c_i)$$

as an estimate of $(Px)_i$, for each i. We can then solve this system of linear equations for the x_j. This is basically what we do in the deterministic case, when we take the y_i as our approximate line-integral data.

The approach of Rockmore and Macovski is different. They suggest that we view the c_i as instances of random variables and treat the unknown x_j as parameters to be determined by maximizing the likelihood function, the standard statistical method for parameter estimation. Using the Poisson formula for each of the c_i, and treating them as independent random variables, we find that the logarithm of the likelihood function is

$$L(x_1, ..., x_J) = \sum_{i=1}^{I} c_i \log[b_i e^{-(Px)_i} + r_i] - b_i e^{-(Px)_i} - r_i - \log(c_i!). \tag{8.2}$$

The maximum likelihood approach is to find the x_j for which this function attains its maximum. Obviously, we cannot solve this problem by simple algebra; we need to employ an iterative optimization method. How we obtain such algorithms is one of the main topics of this chapter.

8.3 Emission Tomography

The views in [172] concerning the potential improvement in image reconstruction through the inclusion of randomness in the physical models apply to emission tomography as well.

In the emission tomography case the photon count y_i at the ith detector (SPECT) or ith LOR (PET) is a Poisson random variable, whose mean value is $(Px)_i$, where P_{ij} is the probability that a photon coming from the jth pixel or voxel will be detected at the ith detector or, in the coincident PET case, that the coincident detections will be associated with the ith LOR, and x_j is the expected number of emissions at the jth pixel. The probabilities P_{ij} depend on the attenuation and, therefore, on the particular patient being scanned, as well as on the geometry of the scanning process.

The log of the likelihood function now takes the form

$$L(x_1, ..., x_J) = \sum_{i=1}^{I} y_i \log(Px)_i - (Px)_i - \log(y_i!).$$

The maximum likelihood method now says that we should maximize this function to obtain our estimate of the radionuclide intensities.

Obtaining a useful iterative algorithm for maximizing likelihood in the emission case is a simpler matter than in the transmission case. Algorithms for the transmission case can be derived by analogy with the emission case, but with a certain amount of approximation.

8.4 An Algorithm for Emission Likelihood Maximization

In their 1982 paper [176] Shepp and Vardi, in discussing the emission problem, suggest that the program of Rockmore and Macovski might be carried out using the iterative method known as the *expectation maximization* (EM) maximum likelihood algorithm. The EM algorithm, discussed in [84], has a rather long history (see [148]). It is not a single algorithm, but rather a framework for developing algorithms to maximize likelihood in a variety of cases. It is a bit unfortunate that the proof of convergence given in [84] for the general algorithm has a flaw, so that convergence of an EM algorithm must be established for each particular application. The paper [176] includes the mathematical formulation of the EM algorithm and a proof of convergence, for the particular case of emission tomography. To distinguish this particular application of the EM approach from the general formulation, I shall refer to the algorithm in the emission case as the EMML algorithm.

In [139], Lange and Carson develop the mathematics for the EM algorithm, for both the transmission and emission problems, and point out that the proof of convergence in [176] has an error. They present a corrected proof, but using a somewhat restrictive condition.

Responding to [139] in [187], the authors remove the restrictive condition of [139], and use a result of Csiszár and Tusnády [80]to prove convergence of the EMML algorithm. The paper [187] appeared in a journal that publishes the comments of other researchers following the original paper. In their discussion of [187], Herman *et al* [120] express the belief that the stochastic approach based on the Poisson statistics and likelihood maximization is not so different from the *algebraic* approach whereby the equations $y_i = (Px)_i$ are solved, approximately, if necessary, for the x_j. They go on to present a concise and useful description of the algebraic approach. They feel that these problems are *reconstructions from projections* in a broad sense, and require finding an approximate solution of the system of linear equations $y_i = (Px)_i$. What is needed, they feel, is the specification of a suitable measure of distance between the y_i and the $(Px)_i$, which can then be minimized. Once an appropriate distance measure is selected, an iterative algorithm that minimizes this distance is required. To be useful, the algorithm must produce accurate reconstructions quickly. Herman *et al* compare the EMML algorithm with some of their methods and find the EMML lacking on several counts.

To improve the entertainment value of the journal, the editors permit the authors of the original paper to respond to the comments of their colleagues, as the authors of [187] chose to do. Vardi *et al* are particularly upset with the comments of Herman *et al.* They deny that they are trying to solve any system of linear equations. They stress the need to adhere to the physics of the situation, which demands a stochastic model. They note that likelihood maximization is a procedure with a proven record in statistics. They dismiss the methods offered by Herman *et al* as *ad hoc*, in contrast to the EMML algorithm, which they view as objective, and easy to interpret. Eventually, it was shown that the EMML algorithm does, in fact, seek an approximate solution of the system of linear equations $y_i = (Px)_i$, just as Herman *et al* had suspected, with the negative of the likelihood function providing an entropy-based distance measure, called the cross-entropy or Kullback-Leibler distance [42].

8.4.1 Cross-Entropy Minimization

The *cross-entropy* or *Kullback-Leibler distance* from the positive number a to the positive number b is

$$KL(a, b) = a \log \frac{a}{b} + b - a = a[\frac{b}{a} - 1 - \log \frac{b}{a}];$$

since $x - 1 - \log x \geq 0$, for all $x > 0$, with equality if and only if $x = 1$, it follows that $KL(a, b) \geq 0$, with equality if and only if $a = b$. We also let $KL(a, 0) = +\infty$ and $KL(0, b) = b$. We define

$$KL(u, v) = \sum_{n=1}^{N} KL(u_n, v_n),$$

for $u = (u_1, ..., u_N)$ and $v = (v_1, ..., v_N)$ vectors with nonnegative entries. It is easy to see that maximizing the likelihood in the emission tomography case is equivalent to minimizing the distance $KL(y, Px)$ over all nonnegative vectors x, with $y = (y_1, ..., y_I)^T$. If there is a nonnegative x with $y = Px$, then such an x maximizes the likelihood and $KL(y, Px) = 0$; we call this the consistent case. In the inconsistent case, in which there is no nonnegative vector x with $y = Px$, the minimum value of $KL(y, Px)$ will be positive and any x for which this minimum value is attained is a maximizer of the likelihood. In either case, it is clear that by maximizing the likelihood function we are seeking a nonnegative x that makes Px as close to y as possible, in the KL sense.

8.4.2 The EMML algorithm

As we just saw, maximizing the likelihood in the emission case is equivalent to minimizing the function $f(x) = KL(y, Px)$ over all x in the nonnegative cone, R^N_+. This is a constrained minimization problem of the sort discussed in the appendix on optimization. If x^* minimizes $f(x)$ over all nonnegative vectors x, then

$$\frac{\partial f}{\partial x_j}(x^*) = 0,$$

for all j such that $x^*_j > 0$, and

$$\frac{\partial f}{\partial x_j}(x^*) \geq 0,$$

for those j for which $x^*_j = 0$. We can compress these two conditions into one by saying

$$x^*_j \frac{\partial f}{\partial x_j}(x^*) = 0,$$

for all indices j; this is the Karush-Kuhn-Tucker (KKT) condition.

For the function $f(x) = KL(y, Px)$ we have

$$\frac{\partial f}{\partial x_j}(x) = \sum_{i=1}^{I} P_{ij}[1 - y_i/(Px)_i],$$

so the steepest descent method takes the form

$$x^{k+1}_j = x^k_j - \alpha_k \sum_{i=1}^{I} P_{ij}[1 - y_i/(Px^k)_i],$$

with $\alpha_k > 0$ chosen as discussed in the appendix on optimization. The EMML algorithm involves a modification of this iteration that leads to an interior-point method and guarantees convergence to a solution. In place of the α_k, the EM uses $x_j^k / \sum_{i=1}^I P_{ij}$, which varies not only with each k, but with each j. The EM iterative step can then be written as

$$x_j^{k+1} = x^k(\sum_{i=1}^I P_{ij}y_i/(Px^k)_i)/(\sum_{i=1}^I P_{ij}).$$

Clearly, if x^0 is a vector with all positive entries, then so is x^k for every k; consequently, the limit, which will exist, will be a nonnegative vector. It can also be shown that each step of the EMML iteration increases the likelihood function.

The modifications used to get the EMML iteration appear to be quite *ad hoc*, but there is a way to motivate the choices, using the KKT conditions. For $f(x) = KL(y, Px)$, the KKT conditions become

$$x_j^* \sum_{i=1}^I P_{ij} = x_j^* \sum_{i=1}^I P_{ij}(y_i/(Px^*)_i),$$

for each j. We can derive an iterative algorithm by replacing the x^* on the right side with the current vector, x^k, and using the left side to define the next vector, x^{k+1}. The true solution, x^*, is a *fixed point* of this iteration, in the sense that if we put x^* on the right side, the left side does not give us anything new.

Simply motivating the EMML algorithm is not enough; we need a firmer foundation if we are to establish useful properties of the algorithm. The *alternating minimization* framework provides such a foundation.

8.5 Alternating Minimization

Suppose that we want to minimize a function $f(x)$ over suitable vectors x. Let $H(x, z)$ be such that, for all suitable x and z,

$$f(z) = H(z, z) \leq H(x, z).$$

Begin with any suitable x^0 and, having found x^k, let x^{k+1} minimize $H(x^k, z)$. Then we have

$$f(x^k) = H(x^k, x^k) \geq H(x^k, x^{k+1}) \geq H(x^{k+1}, x^{k+1}) = f(x^{k+1}).$$

Consequently, the sequence $\{f(x^k)\}$ is decreasing. It does not necessarily follow that the sequence $\{x^k\}$ converges, or, if it does, that the limit minimizes $f(x)$. The idea here is to find $H(x, z)$ so that the minimization with respect to z can be performed easily, and for which we have convergence to a minimizer. To illustrate, we consider the alternating minimization approach for the emission case.

8.5.1 Alternating minimization: the emission case

For each non-negative vector x with $(Px)_i > 0$ for all i, let $r(x)$ be the I by J array with entries

$$r(x)_{ij} = x_j P_{ij}(y_i/(Px)_i),$$

and $q(x)$ the I by J array with entries

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

If there is a non-negative x for which $q(x) = r(x)$ then $y = Px$ and the likelihood is maximized.

The function $H(x, z)$ we use now is

$$H(x, z) = \sum_{i=1}^{I}\sum_{j=1}^{J} KL(r(x)_{ij}, q(z)_{ij}). \tag{8.3}$$

It is easy to see that, having found x^k, the z that minimizes $H(x^k, z)$ has entries

$$x_j^{k+1} = x_j^k \sum_{i=1}^{I} P_{ij}y_i/(Px^k)_i.$$

The sequence generated by the alternating minimization approach is therefore the EMML sequence. For further details about the EMML algorithm and proof of its convergence, see [56].

8.6 Regularizing the EMML algorithm

Maximizing the likelihood seems like a good idea, whether or not it is viewed as solving a system of linear equations. Nevertheless, in practice, the resulting images are often not useful, due to sensitivity to noise in the data. One reason for this was given in [42], where it was shown that, except for certain pathological situations that never occur in practice, when the data is noisy and there is no nonnegative solution to the system $y = Px$, then the maximum likelihood solution will have at most $I - 1$ nonzero entries. Consequently, if we have chosen J larger than I, that is, there are more pixels than data, some of the pixel values must be zero. In practice, these zero values tend to be scattered throughout the image, making the maximum likelihood reconstruction quite noisy.

Maximizing the likelihood can have the effect of making Px too close to y, thereby overfitting the answer to the noisy data. One way out of this is to stop the iteration before it reaches this noisy image. Another way is to use Bayesian maximum *a posteriori* (MAP) methods, as described in [140] (see also [56] and the references given there). The Bayesian formulation adds a second term to the function to be maximized, with the result that the over-fitting to the noisy data is avoided. 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 the 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

$$F(x) = KL(y, Px) - \log f(x). \tag{8.4}$$

Having selected the prior pdf $f(x)$, we want an iterative algorithm to minimize the function $F(x)$ in Equation (26.16). This approach of augmenting the negative likelihood with a penalty function is called *regularization*. It would be a great help if we could mimic the alternating minimization formulation and obtain x^{k+1} by minimizing

$$KL(r(x^k), q(z)) - \log f(z) \tag{8.5}$$

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.

8.6.1 The Gamma prior distribution for x

In [140] 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

$$\gamma(x) = \frac{1}{\Gamma(\alpha)}(\frac{\alpha}{\beta})^{\alpha} x^{\alpha-1} e^{-\alpha x/\beta},$$

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 $\sigma^2 = \beta^2/\alpha$.

Exercise 8.1 *Show that 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 (8.5) with respect to z is equivalent to minimizing the function*

$$KL(r(x^k), q(z)) + \sum_{j=1}^{J} \delta_j KL(\gamma_j, z_j), \tag{8.6}$$

for

$$\delta_j = \frac{\mu_j}{\sigma_j^2},\ \gamma_j = \frac{\mu_j^2 - \sigma_j^2}{\mu_j},$$

under the assumption that the latter term is positive. Show further that the resulting x^{k+1} has entries given in closed form by

$$x_j^{k+1} = \frac{\delta_j}{\delta_j + s_j}\gamma_j + \frac{1}{\delta_j + s_j} x_j^k \sum_{i=1}^{I} P_{ij} y_i / (Px^k)_i, \tag{8.7}$$

where $s_j = \sum_{i=1}^{I} P_{ij}$.

We see from Equation (26.19) 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 in [140]; see also [42].

8.7 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 (8.5), we need some other way to proceed with the alternating minimization. Green [113] has offered the *one-step-late* (OSL) alternative.

When we try to minimize the function in line (8.5) 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 [51] an interior point algorithm (IPA) is presented that avoids the OSL issue. In [153] the IPA is used to regularize transmission tomographic images.

8.8 De Pierro's Surrogate-Function Method

In [85] De Pierro presents a modified EMML algorithm that includes regularization in the form of a penalty function. His objective is 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 [68], 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 (8.3), 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

$$F(x) = KL(y, Px) + g(x),$$

where $g(x) \geq 0$ is some penalty function. De Pierro uses penalty functions $g(x)$ of the form

$$g(x) = \sum_{l=1}^{p} f_l(\langle s_l, x\rangle\ ).$$

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,

$$g(x) = \sum_{l=1}^{p} f_l((Sx)_l).$$

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

$$f_l((Sx)_l) = f_l(\sum_{j=1}^{J} S_{lj}x_j) = f_l(\sum_{j=1}^{J} \lambda_{lj}(S_{lj}/\lambda_{lj})x_j)$$

$$\leq \sum_{j=1}^{J} \lambda_{lj} f_l((S_{lj}/\lambda_{lj})x_j).$$

Therefore,

$$g(x) \leq \sum_{l=1}^{p} \sum_{j=1}^{J} \lambda_{lj} f_l((S_{lj}/\lambda_{lj})x_j).$$

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

$$\sum_{l=1}^{p} \sum_{j=1}^{J} \lambda_{lj} f_l((S_{lj}/\lambda_{lj})z_j).$$

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

$$f_l((S_{lj}/\lambda_{lj})z_j - (S_{lj}/\lambda_{lj})x_j + (Sx)_l).$$

So, De Pierro's function $H(x,z)$ is the sum of the $H(x,z)$ used in the EMML case and the function

$$\sum_{l=1}^{p} \sum_{j=1}^{J} \lambda_{lj} f_l((S_{lj}/\lambda_{lj})z_j - (S_{lj}/\lambda_{lj})x_j + (Sx)_l).$$

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).

8.9 The EM Algorithm: The Transmission Case

Maximizing the likelihood in the transmission case is equivalent to maximizing the log of the likelihood, given by Equation (8.2), which, in turn, is equivalent to minimizing the KL distance

$$g(x) = KL(c, b\exp(-Px) + r),$$

where $c = (c_1, ..., c_I)^T$, and the symbol $b\exp(-Px) + r$ denotes the vector with entries $b_i e^{-(Px)_i} + r_i$. As Fessler *et al* [99] have pointed out, for the transmission problem we are better off if we do not take the logarithm to reduce the problem to linear equations. They feel that it is better to maximize the likelihood in its original form, mainly because the counts at the detectors can be quite small at times.

Because x, the vector of unknowns, appears in the exponent in the transmission likelihood function, developing a suitable algorithm for maximizing the likelihood is more difficult than it was in the emission case. It helps that, in the emission case, the KL distance, $KL(y, Px)$, is a convex function of x; in the transmission case, if we include the background counts, r_i, the function $g(x)$ is no longer convex, which makes it harder to apply optimization theory.

In [99] De Pierro's surrogate-function approach is used to improve earlier algorithms for the transmission case. Parabolic approximations are introduced at a late stage in the algorithm to facilitate the minimization. In [95] parabolic approximations are introduced early, as global surrogates for the function to be minimized. In more recent work, the surrogate-function approach is combined with ordered-subset, or incremental gradient, methods, to accelerate the algorithms [2, 3].

Part III

Systems of Linear Equations

Chapter 9

An Overview of Algorithms

In this chapter we present an overview of iterative algorithms for solving systems of linear equations. In the chapters to follow, we examine each of these algorithms in some detail. We denote by A an arbitrary I by J matrix and by S an N by N square matrix, both with complex entries. For notational convenience, we shall assume throughout this chapter that the rows of A have been rescaled to have Euclidean length one.

9.1 The Algebraic Reconstruction Technique (ART)

The *algebraic reconstruction technique* (ART) applies to an arbitrary system $Ax = b$ of linear equations [112, 121, 128]. For an arbitrary starting point x^0 and $i = k(\bmod I) + 1$, we have

$$x_j^{k+1} = x_j^k + (\sum_{n=1}^{J} |A_{in}|^2)^{-1} \overline{A_{ij}}(b_i - (Ax^k)_i).$$

Since the rows of A have length one, we can write

$$x_j^{k+1} = x_j^k + \overline{A_{ij}}(b_i - (Ax^k)_i). \tag{9.1}$$

In the consistent case, the ART converges to the solution closest to x^0, in the sense of the Euclidean distance. In the inconsistent case, it does not converge, but subsequences associated with the same i converge to distinct vectors, forming a *limit cycle.*

The iterative step in the ART can be written as $x^{k+1} = P_i x^k$, where P_i denotes the orthogonal projection onto the hyperplane associated with the i-th equation. The operator P_i is an affine linear operator.

9.1.1 Relaxed ART

Let $\omega \in (0, 2)$. The *relaxed* ART algorithm has the iterative step

$$x_j^{k+1} = x_j^k + \omega \overline{A_{ij}}(b_i - (Ax^k)_i)). \tag{9.2}$$

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.

9.1.2 Constrained ART

Let C be a closed, nonempty convex subset of C^J and $P_C x$ the orthogonal projection of x onto C. The *constrained* ART algorithm has the iterative step

$$x_j^{k+1} = P_C(x_j^k + \overline{A_{ij}}(b_i - (Ax^k)_i)). \tag{9.3}$$

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

$$x_j^{k+1} = (x_j^k + A_{ij}(b_i - (Ax^k)_i))_+, \tag{9.4}$$

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.

9.1.3 Regularized ART

If the entries of b are noisy but the system $Ax = b$ remains consistent (which can easily happen in the underdetermined 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

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

The solution to this problem is the vector

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

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 [56], while the second one is due to Eggermont, Herman, and Lent [93].

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

$$[\,A^\dagger \quad \gamma I\,]\begin{bmatrix} u \\ v \end{bmatrix} = 0.$$

We begin with $u^0 = b$ and $v^0 = 0$. Then, the lower component of the limit vector is $v^\infty = -\gamma \hat{x}_\epsilon$.

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

$$[\,A \quad \gamma I\,]\begin{bmatrix} x \\ v \end{bmatrix} = b.$$

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

9.2 Cimmino's Algorithm

At each step of the ART algorithm, we perform the orthogonal projection of the current vector x^k onto the i-th hyperplane. Cimmino's method is to project the current vector onto all the hyperplanes and then take the arithmetic mean [71]. The iterative step of Cimmino's algorithm is

$$x_j^{k+1} = x_j^k + \frac{1}{I}\sum_{i=1}^{I} \overline{A_{ij}}(b_i - (Ax^k)_i), \tag{9.5}$$

which can be written as

$$x^{k+1} = x^k + \frac{1}{I}A^\dagger(b - Ax^k). \tag{9.6}$$

As with the ART, Cimmino's method converges to the solution closest to x^0, in the consistent case. Unlike the ART, Cimmino's method converges in the inconsistent case, as well, to the least-squares solution closest to x^0. Note that we can write the iterative step of Cimmino's algorithm as

$$x^{k+1} = \frac{1}{I}\sum_{i=1}^{I} P_i x^k = Tx^k.$$

The operator

$$T = \frac{1}{I}\sum_{i=1}^{I} P_i$$

is an affine linear operator.

9.3 Landweber's Algorithm

Landweber's algorithm [137] has the iterative step

$$x^{k+1} = Tx^k = x^k + \gamma A^\dagger (b - Ax^k), \tag{9.7}$$

which we can write as

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

The operator T with

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

is an affine linear operator, and the linear part,

$$B = I - \gamma A^\dagger A,$$

is Hermitian.

For $\gamma = \frac{1}{I}$ we get Cimmino's method. The Landweber algorithm converges to the solution, or least squares solution, closest to x^0, when $0 < \gamma < 2/\rho(A^\dagger A)$, where $\rho(S)$ denotes the *spectral radius* of S, the maximum of $|\lambda|$, over all eigenvalues λ of S. Since the rows of A have length one, the trace of $AA^\dagger$, which is the sum of its eigenvalues, is I; therefore $\rho(A^\dagger A) = \rho(AA^\dagger) \leq I$. The choice of $\gamma = \frac{1}{I}$ is therefore acceptable in the Landweber algorithm.

The Landweber algorithm minimizes the function $f(x) = \frac{1}{2}||Ax - b||_2^2$. The gradient of $f(x)$ is $\nabla f(x) = A^\dagger (Ax - b)$. Therefore, the iterative step of the Landweber algorithm can be written as

$$x^{k+1} = x^k - \gamma \nabla f(x^k). \tag{9.8}$$

We see from Equation (9.8) that the Landweber algorithm is a special case of *gradient descent* minimization of a function $f(x)$.

9.3.1 SART

The SART algorithm is a special case of the Landweber algorithm. Suppose now that $A_{ij} \geq 0$, for all i and j, and that

$$A_{i+} = \sum_{j=1}^{J} A_{ij} > 0,$$

for each i, and

$$A_{+j} = \sum_{i=1}^{I} A_{ij} > 0,$$

for each j. The SART algorithm [4] has the iterative step

$$x_j^{k+1} = x_j^k + \frac{1}{A_{+j}} \sum_{i=1}^{I} A_{ij}(b_i - (Ax^k)_i)/A_{i+}. \tag{9.9}$$

With

$$B_{ij} = A_{ij}/\sqrt{A_{i+}A_{+j}},$$

$$z_j = x_j\sqrt{A_{+j}},$$

and

$$c_i = b_i/\sqrt{A_{i+}},$$

Equation (9.9) becomes

$$z^{k+1} = z^k + B^T(c - Bz^k), \tag{9.10}$$

which is a special case of the Landweber iteration, with $\gamma = 1$. It can be shown that $\rho(B^TB) = 1$, so the choice of $\gamma = 1$ is acceptable.

9.4 The Projected Landweber Algorithm

For a closed, nonempty convex set C in C^J, the projected Landweber algorithm [15] has the iterative step

$$x^{k+1} = P_C(x^k + \gamma A^\dagger(b - Ax^k)). \tag{9.11}$$

The operator T with

$$Tx = P_C((I - \gamma A^\dagger A)x + \gamma A^\dagger b)$$

is not an affine linear operator. For $\gamma \in (0, 2/\rho(A^\dagger A))$, the projected Landweber algorithm minimizes the function $f(x) = \frac{1}{2}||Ax - b||_2^2$, over $x \in C$, if such a minimizer exists. The projected Landweber iterative step can be written as

$$x^{k+1} = P_C(I - \gamma\nabla f(x^k)),$$

which, for general functions $f(x)$, is the iterative step of the *projected gradient descent* method.

9.5 The CQ Algorithm

The CQ algorithm generalizes the Landweber and projected Landweber methods. Let C and Q denote closed, nonempty convex sets in C^J and C^I, respectively. The function $f(x) = \frac{1}{2}||P_QAx - Ax||_2^2$ has for its gradient

$$\nabla f(x) = A^\dagger(I - P_Q)Ax.$$

The projected gradient descent algorithm now takes the form

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

which is the iterative step of the CQ algorithm [53, 54]. This algorithm minimizes $f(x)$ over x in C, whenever such minimizers exist, provided that γ is in the interval $(0, 2/\rho(A^\dagger A))$.

9.6 Splitting Methods for $Sz = h$

We turn now to square systems of linear equations, denoted $Sz = h$. The *splitting method* involves writing $S = M + K$, where systems of the form $Mx = b$ are easily solved [6]. From

$$Mz = -Kz + h$$

we derive the iteration

$$z^{k+1} = -M^{-1}Kz^k + M^{-1}h. \tag{9.12}$$

The iteration can be written as

$$z^{k+1} = Tz^k = Bz^k + d,$$

where

$$B = -M^{-1}K = I - M^{-1}S,$$

and $d = M^{-1}h$. The operator T is then an affine linear operator, but its linear part B is typically not Hermitian. We consider next some important examples of the splitting method.

9.7 The Jacobi Method

The square matrix S can be written as $S = D + L + U$, where D is its diagonal part, L it lower triangular part, and U its upper triangular part. We assume that D is invertible. The Jacobi method uses $M = D$. The Jacobi iterative step is then

$$z^{k+1} = z^k + D^{-1}(h - Sz^k), \tag{9.13}$$

which we can write as

$$z^{k+1} = Tz^k = Bz^k + d, \tag{9.14}$$

for $B = I - D^{-1}S$ and $d = D^{-1}h$. If S is diagonally dominant, then $\rho(B) < 1$, and there is a vector norm with respect to which T is a strict contraction; the Jacobi method then converges to the unique solution of $Sz = h$. When S is Hermitian, T is then a strict contraction in the Euclidean norm.

9.8 The Jacobi Overrelaxation Method

In order to make this approach applicable to a more general class of problems, the Jacobi *overrelaxation method* (JOR) was introduced. The JOR method uses $M = \frac{1}{\omega}D$. Then $B = I - \omega D^{-1}S$. We are particularly interested in the JOR algorithm for Hermitian, positive-definite S.

9.8.1 When S is Positive-Definite

Suppose that S is Hermitian and positive-definite. Such S arise when we begin with a general system $Ax = b$ and consider the *normal equations* $A^\dagger Ax = A^\dagger b$, or the *Björck-Elfving equations* $AA^\dagger z = b$ [83].Then S has the form $S = R^\dagger R$, for R the N by N Hermitian, positive-definite square root of S. Let $A = RD^{-1/2}$, $x^k = D^{1/2}z^k$, and $b = (R^\dagger)^{-1}h$. Then the JOR iterative step becomes

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

which is the Landweber algorithm, for $Ax = b$. For convergence, we need γ in the interval $(0, 2/\rho(A^\dagger A))$. Note that $\rho(A^\dagger A) = \rho(D^{-1/2}SD^{-1/2})$.

When we apply the JOR to the normal equations $A^\dagger Ax = A^\dagger b$, we find that it is equivalent to the Landweber iteration on the system $AD^{-1/2}z = b$. When we apply the JOR iteration to the Björck-Elfving equations $AA^\dagger z = b$, we find that it is equivalent to the Landweber iteration applied to the system $D^{-1/2}Ax = D^{-1/2}b$.

9.9 The Gauss-Seidel Method

The Gauss-Seidel (GS) method uses the matrix $M = D + L$. The GS iteration can be written as

$$x^{k+1} = Tx^k = Bx^k + d,$$

for

$$B = I - (D + L)^{-1}S$$

and $d = (D+L)^{-1}h$. Once again, the operator T is affine linear; the linear part B is typically not Hermitian.

9.9.1 When S is Nonnegative-Definite

If the matrix S is Hermitian, nonnegative-definite, then it can be shown that $|\lambda| < 1$ for every eigenvalue λ of B that is not equal to one. Consequently, there is a vector norm with respect to which the operator T is paracontractive. The GS iteration then converges to a solution, whenever one exists. If S is positive-definite, then T is a strict contraction, for that same vector norm, and the GS iteration converges to the unique solution of $Sz = h$.

9.10 Successive Overrelaxation

The *successive overrelaxation* (SOR) method uses the matrix $M = \frac{1}{\omega}D + L$; when $\omega = 1$ we have the GS method. The SOR iteration can be written as

$$z^{k+1} = Tz^k = Bz^k + d,$$

for

$$B = (D + \omega L)^{-1}((1 - \omega)D - \omega U).$$

It can be shown that $|\det(B)| = |1 - \omega|^N$, so that $\rho(B) > 1$, for $\omega < 0$ or $\omega > 2$.

9.10.1 When S is Positive-Definite

Suppose that S is positive-definite. Then we can write $S = AA^\dagger$. Let $\{z^k\}$ be the iterative sequence generated by the SOR. Then the sequence $\{x^k = A^\dagger z^k\}$ is the sequence generated by one full cycle of the ART algorithm, applied to the system $Ax = b$.

9.11 Projecting onto Convex Sets

The iterative step of the ART algorithm is $x^{k+1} = P_i x^k$, where P_i denotes the orthogonal projection onto the hyperplane associated with the i-th equation. This suggests a more general algorithm for finding a vector in the nonempty intersection of closed, convex sets $C_1, ..., C_I$. For each k, let $i = k(\bmod I) + 1$ and let

$$x^{k+1} = P_{C_i} x^k,$$

where P_{C_i} denotes the orthogonal projection onto the set C_i. This algorithm is the *successive orthogonal projection* (SOP) method [114]. It converges whenever the intersection is nonempty.

9.11.1 The Agmon-Motzkin-Schoenberg Algorithm

When the convex sets C_i are half-spaces

$$C_i = \{x|(Ax)_i \geq b_i\},$$

the SOP algorithm becomes the Agmon-Motzkin-Schoenberg (AMS) algorithm [1, 152].

9.12 The Multiplicative ART (MART)

We turn now to the case in which the entries of the matrix A and vector x are nonnegative and those of b are positive. We seek a nonnegative solution of the system $Ax = b$. The *multiplicative* ART (MART) algorithm [112] has the iterative step

$$x_j^{k+1} = x_j^k (b_i/(Ax^k))^{A_{ij}/m_i},$$

for $i = k(\bmod I) + 1$ and $m_i = \max\{A_{ij} | j = 1, ..., J\}$. When nonnegative solutions exist, we say that we are in the consistent case. In the consistent case, the MART converges to the nonnegative solution of $Ax = b$ for which the cross-entropy, or Kullback-Leibler distance $KL(x, x^0)$ is minimized.

9.13 The Simultaneous MART (SMART)

The MART algorithm resembles the ART algorithm, in that it uses only a single equation at each step. Analogous to the Cimmino algorithm we have the *simultaneous* MART (SMART) [42, 43, 82, 124, 175]. The SMART method begins with a positive vector x^0; having calculated x^k, we calculate x^{k+1} using

$$\log x_j^{k+1} = \log x_j^k + s_j^{-1} \sum_{i=1}^I A_{ij} \log \frac{b_i}{(Ax^k)_i}, \tag{9.15}$$

where $s_j = \sum_{i=1}^I A_{ij} > 0$.

In the consistent case the SMART converges to the unique nonnegative solution of $b = Ax$ for which the KL distance $KL(x, x^0)$ is minimized. In the inconsistent case it converges to the unique nonnegative minimizer of the distance $KL(Ax, b)$ for which $KL(x, x^0)$ is minimized; if A and every matrix derived from A by deleting columns has full rank then there is a unique nonnegative minimizer of $KL(Ax, b)$ and at most $I - 1$ of its entries are nonzero.

9.14 The Expectation-Maximization Maximum Likelihood (EMML) Method

The iterative tep of the EMML algorithm is

$$x_j^{k+1} = x_j^k s_j^{-1} \sum_{i=1}^I A_{ij} \frac{b_i}{(Ax^k)_i}.$$

In the consistent case the EMML algorithm [42, 43, 84, 139, 140, 176, 187] converges to nonnegative solution of $Ax = b$. In the inconsistent case it converges to a nonnegative minimizer of the distance $KL(b, Ax)$; if A and every matrix derived from A by deleting columns has full rank then there is a unique nonnegative minimizer of $KL(b, Ax)$ and at most $I - 1$ of its entries are nonzero.

9.15 Block-Iterative Algorithms

We begin by selecting subsets S_n, $n = 1, ..., N$ whose union is the set of equation indices $\{i = 1, ..., I\}$; the S_n need not be disjoint. Having found iterate x^k, set $n = k(\bmod N) + 1$. The RBI-EMML [45, 55] algorithm has the following iterative step:

$$x_j^{k+1} = x_j^k(1 - m_n^{-1}s_j^{-1}s_{nj}) + x_j^k m_n^{-1}s_j^{-1}\sum_{i\in S_n} A_{ij}\frac{b_i}{(Ax^k)_i}, \tag{9.16}$$

where

$$m_n = \max\{s_{nj}/s_j \,|j = 1, ..., J\}. \tag{9.17}$$

For any choice of subsets S_n, and any starting vector $x^0 > 0$, the RBI-EMML converges to a nonnegative solution whenever one exists. The acceleration, compared to the EMML, is roughly on the order of N, the number of subsets. As with the ART, the composition of the subsets, as well as their ordering, can affect the rate of convergence.

9.16 Summary

These algorithms fall into three broad categories. The first, involving orthogonal projection operators P_C, affine operators with positive-definite linear parts, or, more generally, operators of the form $I - \gamma\nabla f$, for suitable γ and convex functions $f(x)$, will be show to be *averaged non-expansive* with respect to the Euclidean norm. Convergence of these algorithms will follow from the Krasnoselskii-Mann Theorem 27.1. The second class, involving affine operators whose linear parts are not positive-definite, are shown to be paracontractive, with respect to an appropriately chosen norm, and their convergence will be established using the Elsner-Koltracht-Neumann Theorem 27.2. The third class, those involving operators whose domain is restricted to nonnegative vectors, are shown to be paracontractive in the generalized sense of cross-entropy. Many of these algorithms were obtained by extending algorithms in the other classes to the cross-entropy case. Proofs of convergence for these algorithms are then obtained by mimicking the proofs for the other classes, but changing the notion of distance.

Chapter 10

The Algebraic Reconstruction Technique

The algebraic reconstruction technique (ART) [112] is a sequential iterative algorithm for solving an arbitrary system $Ax = b$ of I real or complex linear equations in J unknowns. For notational simplicity, we shall assume, from now on in this chapter, that the equations have been normalized so that the rows of A have Euclidean length one, that is, for each $i = 1, ..., I$, we have

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

10.1 Calculating the ART

The ART is the following: begin with an arbitrary vector x^0; for each nonnegative integer k, having found x^k, let x^{k+1} be the vector with entries

$$x_j^{k+1} = x_j^k + \gamma\overline{A_{ij}}(b_i - (Ax^k)_i), \tag{10.1}$$

where the parameter γ is chosen in the interval $(0, 2)$. When $\gamma = 1$, we say that the ART is unrelaxed. Because the ART uses only a single equation at each step, it has been called a *row-action* method [60].

10.2 Convergence of the ART

When the system $Ax = b$ has exact solutions the ART converges to the solution closest to x^0. How fast the algorithm converges will depend on the ordering of the equations and on whether or not we use relaxation.

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* [182, 86, 56]. 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?

10.2.1 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

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

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

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

and these need not be the same. A simple example is the following. Consider the system

$$x = 1; x = 2,$$

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

$$2x = 2; x = 2,$$

which has the least-squares solution $x = 1.2$. The so-called *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 [46] (see also [56]) 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 generally true, however.

Open Question: In both the consistent and inconsistent cases, the sequence $\{x^k\}$ of ART iterates is bounded [182, 86, 46, 56]. The proof is easy in the consistent case. Is there an easy proof for the inconsistent case?

10.2.2 Nonnegatively Constrained ART

If we are seeking a nonnegative solution for the real system $Ax = b$, we can modify the ART by replacing the x^{k+1} given by Equation (10.1) with $(x^{k+1})_+$. This version of ART will converge to a nonnegative solution, whenever one exists, but will produce a limit cycle otherwise.

10.3 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) [50]:

10.3.1 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 $Ax = y$.

10.3.2 Strongly Underrelaxed ART

Another method for avoiding the LC is *strong underrelaxation* [61]. Let $t > 0$. Replace the iterative step in ART with

$$x_j^{k+1} = x_j^k + t\overline{A_{ij}}(b_i - (Ax^k)_i). \tag{10.2}$$

In [61] 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 [46]. Bertsekas [17] uses strong underrelaxation to obtain convergence of more general incremental methods.

10.4 Approximate Solutions and the Nonnegativity Constraint

For the real system $Ax = b$, consider the *nonnegatively constrained least-squares* problem of minimizing the function $||Ax - b||_2$, subject to the constraints $x_j \geq 0$ for all j; this is a nonnegatively constrained least-squares approximate solution. As noted previously, we can solve this problem using a slight modification of the ART. Although there may be multiple solutions $\hat{x}$, we know, at least, that $A\hat{x}$ is the same for all solutions.

According to the Karush-Kuhn-Tucker theorem [163], the vector $A\hat{x}$ must satisfy the condition

$$\sum_{i=1}^{I} A_{ij}((A\hat{x})_i - b_i) = 0 \tag{10.3}$$

for all j for which $\hat{x}_j > 0$ for some solution $\hat{x}$. Let S be the set of all indices j for which there exists a solution $\hat{x}$ with $\hat{x}_j > 0$. Then Equation (10.3) must hold for all j in S. Let Q be the matrix obtained from A by deleting those columns whose index j is not in S. Then $Q^T(A\hat{x} - b) = 0$. If Q has full rank and the cardinality of S is greater than or equal to I, then Q^T is one-to-one and $A\hat{x} = b$. We have proven the following result.

Theorem 10.1 *Suppose that A has the full-rank property, that is, A and every matrix Q obtained from A by deleting columns has full rank. Suppose there is no nonnegative solution of 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.*

When $\hat{x}$ is a vectorized two-dimensional image and $J > I$, the presence of at most $I - 1$ positive pixels makes the resulting image resemble stars in the sky; for that reason this theorem and the related result for the EMML algorithm ([42]) are sometimes called *night sky* theorems. The zero-valued pixels typically appear scattered throughout the image. This behavior occurs with all the algorithms discussed so far that impose nonnegativity, whenever the real system $Ax = b$ has no nonnegative solutions.

This result leads to the following open question:

Open Question: How does the set S defined above vary with the choice of algorithm, with the choice of x^0 for a given algorithm, and for the choice of subsets in the block-iterative algorithms?

Chapter 11

The Multiplicative ART (MART)

The *multiplicative* ART (MART) [112] is an iterative algorithm closely related to the ART. It applies to systems of linear equations $Ax = b$ for which the b_i are positive and the A_{ij} are nonnegative; the solution x we seek will 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, in which the relation is most clearly visible.

11.1 A Special Case of ART and MART

We begin by considering the application of ART 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.

Exercise 11.1 *With A defined as above, multiplying an I by 1 vector c by the transpose, A^T, is backprojection. Examine the effect of backprojection by considering the individual entries of A^Tc.*

With $i = k(\bmod I) + 1$, the iterative step of the ART algorithm is

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

for j in L_i, and

$$x_j^{k+1} = x_j^k,$$

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

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

for those j in L_i, and

$$x_j^{k+1} = x_j^k,$$

otherwise. Therefore, we can write the iterative step as

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

11.2 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. The MART begins with a positive vector x^0. Having found x^k for nonnegative integer k, we let $i = k(\mathrm{mod}\, I) + 1$ and define x^{k+1} by

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i} \Big)^{m_i^{-1} A_{ij}}, \tag{11.1}$$

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.

The MART can be accelerated by relaxation, as well. The relaxed MART has the iterative step

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i} \Big)^{\gamma_i m_i^{-1} A_{ij}}, \tag{11.2}$$

where γ_i is in the interval $(0, 1)$. As with ART, finding the best relaxation parameters is a bit of an art.

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

Theorem 11.1 *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,

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

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.

11.3 ART and MART as Sequential Projection Methods

The iterative ART step can be viewed as the orthogonal projection of the current vector, x^k, onto

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

the hyperplane associated with the i-th equation. Can we view MART in a similar way? Yes, but we need to consider a different measure of closeness between nonnegative vectors.

11.3.1 Cross-Entropy or the Kullback-Leibler Distance

For positive numbers u and v, the Kullback-Leibler distance [136] from u to v is

$$KL(u,v) = u \log \frac{u}{v} + v - u. \tag{11.3}$$

We also define $KL(0,0) = 0$, $KL(0,v) = v$ and $KL(u,0) = +\infty$. The KL distance is extended to nonnegative vectors component-wise, so that for nonnegative vectors x and z we have

$$KL(x,z) = \sum_{j=1}^{J} KL(x_j, z_j). \tag{11.4}$$

Exercise 11.2 *One of the most useful facts about the KL distance is that, for all nonnegative vectors x and z, with $z_+ = \sum_{j=1}^{J} z_j > 0$, we have*

$$KL(x,z) = KL(x_+, z_+) + KL(x, \frac{x_+}{z_+} z). \tag{11.5}$$

Prove this.

Given the vector x^k, we find the vector z in H_i for which the KL distance $f(z) = KL(x^k, z)$ is minimized; this z will be the KL projection of x^k onto H_i. Using a Lagrange multiplier, we find that

$$0 = \frac{\partial f}{\partial z_j}(z) - \lambda_i A_{ij},$$

for some constant λ_i, so that

$$0 = -\frac{x_j^k}{z_j} + 1 - \lambda_i A_{ij},$$

for each j. Multiplying by z_j, we get

$$z_j - x_j = z_j A_{ij} \lambda_i. \tag{11.6}$$

For the special case in which the entries of A_{ij} are zero or one, we can solve Equation (11.6) for z_j. We have

$$z_j - x_j^k = z_j A_{ij} \lambda_i,$$

for each $j \in L_i$, and $z_j = x_j^k$, otherwise. Multiply both sides by A_{ij} and sum on j to get

$$b_i(1 - \lambda_i) = (Ax^k)_i.$$

Therefore,

$$z_j = x_j^k \frac{b_i}{(Ax^k)_i},$$

which is clearly x_j^{k+1}. So, at least in the special case we have been discussing, MART consists of projecting, in the KL sense, onto each of the hyperplanes in succession.

11.3.2 Weighted KL Projections

For the more general case in which the entries A_{ij} are arbitrary nonnegative numbers, we cannot directly solve for z_j in Equation (11.6). There is an alternative, though. Instead of minimizing $KL(x, z)$, subject to $(Az)_i = b_i$, we minimize the weighted KL distance

$$\sum_{j=1}^{J} A_{ij} KL(x_j, z_j),$$

subject to the same constraint on z. We shall denote the optimal z by $Q_i x$. Again using a Lagrange multiplier approach, we find that

$$0 = -A_{ij}(\frac{x_j}{z_j} + 1) - A_{ij}\lambda_i,$$

for some constant λ_i. Multiplying by z_j, we have

$$A_{ij}z_j - A_{ij}x_j = A_{ij}z_j\lambda_i. \tag{11.7}$$

Summing over the index j, we get

$$b_i - (Ax)_i = b_i\lambda_i,$$

from which it follows that

$$1 - \lambda_i = (Ax)_i/b_i.$$

Substituting for λ_i in equation (11.7), we obtain

$$z_j = (Q_i x)_j = x_j \frac{b_i}{(Ax)_i}, \tag{11.8}$$

for all j for which $A_{ij} \neq 0$.

Note that the MART step does not define x^{k+1} to be this weighted KL projection of x^k onto the hyperplane H_i; that is,

$$x_j^{k+1} \neq (Q_i x^k)_j,$$

except for those j for which $\frac{A_{ij}}{m_i} = 1$. What is true is that the MART step involves relaxation. Writing

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

we see that x_j^{k+1} is a weighted geometric mean of x_j^k and $(Q_i x^k)_j$.

11.4 Proof of Convergence for MART

We assume throughout this proof that x is a nonnegative solution of $Ax = b$. For $i = 1, 2, ..., I$, let

$$G_i(x,z) = KL(x,z) + m_i^{-1} KL((Ax)_i, b_i) - m_i^{-1} KL((Ax)_i, (Az)_i).$$

Exercise 11.3 *Use Equation (26.5) to prove that $G_i(x,z) \geq 0$ for all x and z.*

Exercise 11.4 *Show that $G_i(x,z)$, viewed as a function of z, is minimized by $z = x$, by showing that*

$$G_i(x,z) = G_i(x,x) + KL(x,z) - m_i^{-1} KL((Ax)_i, (Az)_i). \tag{11.9}$$

Exercise 11.5 *Show that $G_i(x,z)$, viewed as a function of x, is minimized by $x = z'$, where*

$$z'_j = z_j \Big(\frac{b_i}{(Az)_i} \Big)^{m_i^{-1} A_{ij}},$$

by showing that

$$G_i(x,z) = G_i(z',z) + KL(x,z'). \tag{11.10}$$

We note that $x^{k+1} = (x^k)'$.

Now we calculate $G_i(x, x^k)$ in two ways, using, first, the definition, and, second, Equation (11.10). From the definition, we have

$$G_i(x,x^k) = KL(x,x^k) - m_i^{-1} KL(b_i, (Ax^k)_i).$$

From Equation (11.10), we have

$$G_i(x,x^k) = G_i(x^{k+1}, x^k) + KL(x, x^{k+1}).$$

Therefore,

$$KL(x,x^k) - KL(x,x^{k+1}) = G_i(x^{k+1},x^k) + m_i^{-1} KL(b_i, (Ax^k)_i). \tag{11.11}$$

From Equation (11.11) we can conclude several things:

1) the sequence $\{KL(x,x^k)\}$ is decreasing;

2) the sequence $\{x^k\}$ is bounded, and therefore has a cluster point, x^*; and 3) the sequences $\{G_i(x^{k+1},x^k)\}$ and $\{m_i^{-1} KL(b_i, (Ax^k)_i)\}$ converge decreasingly to zero, and so $b_i = (Ax^*)_i$ for all i.

Since $b = Ax^*$, we can use x^* in place of the arbitrary solution x to conclude that the sequence $\{KL(x^*, x^k)\}$ is decreasing. But, a subsequence converges to zero, so the entire sequence must converge to zero, and therefore $\{x^k\}$ converges to x^*. Finally, since the right side of Equation (11.11) is independent of which solution x we have used, so is the left side. Summing over k on the left side, we find that

$$KL(x, x^0) - KL(x, x^*)$$

is independent of which x we use. We can conclude then that minimizing $KL(x, x^0)$ over all solutions x has the same answer as minimizing $KL(x, x^*)$ over all such x; but the solution to the latter problem is obviously $x = x^*$. This concludes the proof. ∎

11.5 Comments on the Rate of Convergence of MART

We can see from Equation (11.11),

$$KL(x, x^k) - KL(x, x^{k+1}) = G_i(x^{k+1}, x^k) + m_i^{-1} KL(b_i, (Ax^k)_i),$$

that the decrease in distance to a solution that occurs with each step of MART depends on m_i^{-1} and on $KL(b_i, (Ax^k)_i)$; the latter measures the extent to which the current vector x^k solves the current equation. We see then that it is reasonable to select m_i as we have done, namely, as the smallest positive number c_i for which $A_{ij}/c_i \leq 1$ for all j. We also see that it is helpful if the equations are ordered in such a way that $KL(b_i, (Ax^k)_i)$ is fairly large, for each k. It is not usually necessary to determine an optimal ordering of the equations; the important thing is to avoid ordering the equations so that successive hyperplanes have nearly parallel normal vectors.

Chapter 12

Rescaled Block-Iterative (RBI) 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.

12.1 Block-Iterative Methods

Methods based on cross-entropy, such as the *multiplicative ART* (MART), its simultaneous version, SMART, the expectation maximization maximum likelihood method (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 nonnegative 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.

12.2 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. Both begin with an arbitrary positive vector x^0. Having found x^k the iterative step for the SMART is

SMART:

$$x_j^{k+1} = x_j^k \exp\Big(s_j^{-1} \sum_{i=1}^{I} A_{ij} \log \frac{b_i}{(Ax^k)_i}\Big) \tag{12.1}$$

while that for the EMML method is

EMML:

$$x_j^{k+1} = x_j^k s_j^{-1} \sum_{i=1}^{I} A_{ij} \frac{b_i}{(Ax^k)_i}. \tag{12.2}$$

The main results concerning the SMART is given by the following theorem.

Theorem 12.1 *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 12.2 *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

$$x_j = x_j^0 \exp\Big(\sum_{i=1}^{I} A_{ij}\lambda_i\Big) \tag{12.3}$$

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 12.1 *Minimizing $KL(d, x)$ over x as in Equation (12.3) is equivalent to minimizing $KL(x, x^0)$, subject to $Ax = Pd$.*

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

$$L(\lambda) = \prod_{j=1}^{J} p_j^{d_j} \tag{12.4}$$

so that the log-likelihood function is

$$LL(\lambda) = \sum_{j=1}^{J} d_j \log p_j. \tag{12.5}$$

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.

12.3 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 $n = k(\text{mod}\, N) + 1$, having found x^k let

OSEM:

$$x_j^{k+1} = x_j^k s_{nj}^{-1} \sum_{i \in B_n} A_{ij} \frac{b_i}{(Ax^k)_i}. \tag{12.6}$$

The OSSMART has the following iterative step:

OSSMART

$$x_j^{k+1} = x_j^k \exp\Big(s_{nj}^{-1} \sum_{i \in B_n} A_{ij} \log \frac{b_i}{(Ax^k)_i} \Big). \tag{12.7}$$

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.

12.4 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. The SMART was discovered in 1972, independently, by Darroch and Ratcliff, working in statistics, [82] and by Schmidlin [175] in medical imaging. Block-iterative versions of SMART are also treated in [82], but they also insist on subset balance. The inconsistent case was not considered.

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

$$x_j^{k+1} = x_j^k \exp\Big(\beta_{nj} \sum_{i \in B_n} \alpha_{ni} A_{ij} \log\Big(\frac{b_i}{(Ax^k)_i}\Big)\Big), \qquad (12.8)$$

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

$$\gamma_j \delta_n \sigma_{nj} \leq 1, \qquad (12.9)$$

for $\sigma_{nj} = \sum_{i \in B_n} \alpha_{ni} A_{ij}$. With these conditions satisfied we have the following result.

Theorem 12.3 *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 (12.8) 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 12.2 *Let $b = Ax$ for some nonnegative x. Then for $\{x^k\}$ as in Equation (12.8) we have*

$$\sum_{j=1}^J \gamma_j^{-1} KL(x_j, x_j^k) - \sum_{j=1}^J \gamma_j^{-1} KL(x_j, x_j^{k+1}) \geq$$

$$\delta_n \sum_{i \in B_n} \alpha_{ni} KL(b_i, (Ax^k)_i). \tag{12.10}$$

Proof: First note that

$$x_j^{k+1} = x_j^k \exp\Big(\gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \log\Big(\frac{b_i}{(Ax^k)_i}\Big)\Big), \tag{12.11}$$

and

$$\exp\Big(\gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \log\Big(\frac{b_i}{(Ax^k)_i}\Big)\Big)$$

can be written as

$$\exp\Big((1 - \gamma_j \delta_n \sigma_{nj}) \log 1 + \gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \log\Big(\frac{b_i}{(Ax^k)_i}\Big)\Big),$$

which, by the convexity of the exponential function, is not greater than

$$(1 - \gamma_j \delta_n \sigma_{nj}) + \gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \frac{b_i}{(Ax^k)_i}.$$

It follows that

$$\sum_{j=1}^J \gamma_j^{-1} (x_j^k - x_j^{k+1}) \geq \delta_n \sum_{i \in B_n} \alpha_{ni} ((Ax^k)_i - b_i).$$

We also have

$$\log(x_j^{k+1}/x_j^k) = \gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \log \frac{b_i}{(Ax^k)_i}.$$

Therefore

$$\sum_{j=1}^{J}\gamma_j^{-1}KL(x_j,x_j^k)-\sum_{j=1}^{J}\gamma_j^{-1}KL(x_j,x_j^{k+1})$$

$$=\sum_{j=1}^{J}\gamma_j^{-1}(x_j\log(x_j^{k+1}/x_j^k)+x_j^k-x_j^{k+1})$$

$$=\sum_{j=1}^{J}x_j\delta_n\sum_{i\in B_n}\alpha_{ni}A_{ij}\log\frac{b_i}{(Ax^k)_i}+\sum_{j=1}^{J}\gamma_j^{-1}(x_j^k-x_j^{k+1})$$

$$=\delta_n\sum_{i\in B_n}\alpha_{ni}(\sum_{j=1}^{J}x_jA_{ij})\log\frac{b_i}{(Ax^k)_i}+\sum_{j=1}^{J}\gamma_j^{-1}(x_j^k-x_j^{k+1})$$

$$\geq\delta_n\Big(\sum_{i\in B_n}\alpha_{ni}(b_i\log\frac{b_i}{(Ax^k)_i}+(Ax^k)_i-b_i)\Big)=\delta_n\sum_{i\in B_n}\alpha_{ni}KL(b_i,(Ax^k)_i).$$

This completes the proof of the lemma. ∎

From the inequality (12.10) we conclude 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 $\{x^k\}$. 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 (12.10) 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 (12.10) 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

$$\delta_n = 1/\min\{\sigma_{nj}\gamma_j,\, j = 1, ..., J\}. \tag{12.12}$$

The *rescaled* BI-SMART (RBI-SMART) as presented in [**?**, 46, 47] uses this choice, but with $\alpha_{ni} = 1$ for each n and i. 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 (12.7) 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 [66] 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 (26.3) uses $N = 1$, $\gamma_j = s_j^{-1}$ and $\alpha_{ni} = \alpha_i = 1$. Clearly the inequality (12.9) 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 [112], we take $N = I$ and $B_n = B_i = \{i\}$ for $i = 1, ..., I$. The MART begins with a strictly positive vector x^0 and has the iterative step

The MART:

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i}\Big)^{m_i^{-1} A_{ij}}, \tag{12.13}$$

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 [**?**].

Darroch and Ratcliff included a discussion of a block-iterative version of SMART in their 1972 paper [82]. 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 [125], 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.

12.5 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

$$x_j^{k+1} = x_j^k(1 - \beta_{nj}\sigma_{nj}) + x_j^k \beta_{nj} \sum_{i \in B_n} \alpha_{ni} A_{ij} \frac{b_i}{(Ax^k)_i}, \tag{12.14}$$

for $j = 1, 2, ..., J$, $n = k(\text{mod } N)+1$ 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,

$$b_{nj} = \gamma_j \delta_n$$

for each j and n and that the inequality (12.9) hold. With these conditions satisfied we have the following result.

Theorem 12.4 *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 (12.8) 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 12.3 *Let $b = Ax$ for some nonnegative x. Then for $\{x^k\}$ as in Equation (12.14) we have*

$$\sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^k) - \sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^{k+1}) \geq$$

$$\delta_n \sum_{i\in B_n} \alpha_{ni} KL(b_i, (Ax^k)_i). \tag{12.15}$$

Proof: From the iterative step

$$x_j^{k+1} = x_j^k(1-\gamma_j\delta_n\sigma_{nj}) + x_j^k\gamma_j\delta_n \sum_{i\in B_n} \alpha_{ni}A_{ij}\frac{b_i}{(Ax^k)_i}$$

we have

$$\log(x_j^{k+1}/x_j^k) = \log\Big((1-\gamma_j\delta_n\sigma_{nj}) + \gamma_j\delta_n \sum_{i\in B_n} \alpha_{ni}A_{ij}\frac{b_i}{(Ax^k)_i}\Big).$$

By the concavity of the logarithm we obtain the inequality

$$\log(x_j^{k+1}/x_j^k) \geq \Big((1-\gamma_j\delta_n\sigma_{nj})\log 1 + \gamma_j\delta_n \sum_{i\in B_n} \alpha_{ni}A_{ij}\log\frac{b_i}{(Ax^k)_i}\Big),$$

or

$$\log(x_j^{k+1}/x_j^k) \geq \gamma_j\delta_n \sum_{i\in B_n} \alpha_{ni}A_{ij}\log\frac{b_i}{(Ax^k)_i}.$$

Therefore

$$\sum_{j=1}^J \gamma_j^{-1}x_j\log(x_j^{k+1}/x_j^k) \geq \delta_n \sum_{i\in B_n} \alpha_{ni}(\sum_{j=1}^J x_jA_{ij})\log\frac{b_i}{(Ax^k)_i}.$$

Note that it is at this step that we used the separability of the β_{nj}. Also

$$\sum_{j=1}^J \gamma_j^{-1}(x_j^{k+1}-x_j^k) = \delta_n \sum_{i\in B_n}((Ax^k)_i - b_i).$$

This concludes the proof of the lemma. ∎

From the inequality (12.15) 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 $b = Ax$. We are now unable to replace the inequality (12.15) 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 (12.15) 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 once again we take

$$\delta_n = 1/\min\{\sigma_{nj}\gamma_j,\, j = 1, ..., J\}. \tag{12.16}$$

The *rescaled* BI-EMML (RBI-EMML) as presented in [**?**, 46, 47] uses this choice, but with $\alpha_{ni} = 1$ for each n and i. 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 (12.6) 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 (26.4) uses $N = 1$, $\gamma_j = s_j^{-1}$ and $\alpha_{ni} = \alpha_i = 1$. Clearly the inequality (12.9) 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, has the iterative step

EM-MART:

$$x_j^{k+1} = (1 - \delta_i \gamma_j \alpha_{ii} A_{ij}) x_j^k + \delta_i \gamma_j \alpha_{ii} A_{ij} \frac{b_i}{(Ax^k)_i}, \tag{12.17}$$

with

$$\gamma_j \delta_i \alpha_{ii} A_{ij} \leq 1$$

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 [27]. 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 has the following iterative step:
RAMLA:

$$x_j^{k+1} = (1 - \lambda_k \sum\nolimits^n A_{ij})x_j^k + \lambda_k x_j^k \sum\nolimits^n A_{ij}\Big(\frac{b_i}{(Ax^k)_i}\Big), \tag{12.18}$$

where the positive relaxation parameters λ_k are chosen to converge to zero and $\sum_{k=0}^{+\infty} \lambda_k = +\infty$.

12.6 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 (12.11) 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 12.4 *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*

$$KL(a, b) = KL(a_+, b_+) + KL(a_+, \frac{a_+}{b_+} b). \tag{12.19}$$

For nonnegative vectors x and z let

$$G_n(x, z) = \sum_{j=1}^J \gamma_j^{-1} KL(x_j, z_j)$$

$$+\delta_n \sum_{i \in B_n} \alpha_{ni}[KL((Ax)_i, b_i) - KL((Ax)_i, (Pz)_i)]. \tag{12.20}$$

It follows from Lemma 12.19 and the inequality

$$\gamma_j^{-1} - \delta_n \sigma_{nj} \geq 1$$

that $G_n(x, z) \geq 0$ in all cases.

Lemma 12.5 *For every x we have*

$$G_n(x, x) = \delta_n \sum_{i \in B_n} \alpha_{ni} KL((Ax)_i, b_i) \tag{12.21}$$

so that

$$G_n(x,z) = G_n(x,x) + \sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, z_j)$$

$$-\delta_n \sum_{i \in B_n} \alpha_{ni} KL((Ax)_i, (Pz)_i). \tag{12.22}$$

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

$$x_j = z'_j = z_j \exp\Big(\gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \log \frac{b_i}{(Pz)_i}\Big). \tag{12.23}$$

Lemma 12.6 *For each x and z we have*

$$G_n(x,z) = G_n(z',z) + \sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, z'_j). \tag{12.24}$$

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

$$G_n(u,x^k) = \sum_{j=1}^{J} \gamma_j^{-1} KL(u_j, x_j^k) - \delta_n \sum_{i \in B_n} \alpha_{ni} KL(b_i, (Ax^k)_i),$$

while using Lemma 12.24 we find that

$$G_n(u,x^k) = G_n(x^{k+1}, x^k) + \sum_{j=1}^{J} \gamma_j^{-1} KL(u_j, x_j^{k+1}).$$

Therefore

$$\sum_{j=1}^{J} \gamma_j^{-1} KL(u_j, x_j^k) - \sum_{j=1}^{J} \gamma_j^{-1} KL(u_j, x_j^{k+1})$$

$$= G_n(x^{k+1}, x^k) + \delta_n \sum_{i \in B_n} \alpha_{ni} KL(b_i, (Ax^k)_i). \tag{12.25}$$

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 (12.25) does not depend on the particular choice of solution we made, neither does the left side. By *telescoping* we conclude that

$$\sum_{j=1}^J \gamma_j^{-1} KL(u_j, x_j^0) - \sum_{j=1}^J \gamma_j^{-1} KL(u_j, x_j^*)$$

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 13

The Split Feasibility Problem

The *split feasibility problem* (SFP) [62] 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.

13.1 The CQ Algorithm

In [53] the CQ algorithm for solving the SFP was presented, for the real case. It has the iterative step

$$x^{k+1} = P_C(x^k - \gamma A^T(I - P_Q)Ax^k), \tag{13.1}$$

where I is the identity operator and $\gamma \in (0, 2/\rho(A^TA))$, for $\rho(A^TA)$ the spectral radius of the matrix A^TA, 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

$$x^{k+1} = P_C(x^k - \gamma A^\dagger(I - P_Q)Ax^k). \tag{13.2}$$

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

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

over the set C, provided such constrained minimizers exist [54]. The CQ algorithm employs the relaxation parameter γ in the interval $(0, 2/L)$, where L is the largest eigenvalue of the matrix $A^T A$. 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 $A^T A$ does not exceed the maximum number of nonzero elements in any column of A. A similar upper bound on $\rho(A^T A)$ can be obtained for non-normalized, ϵ-sparse A.

13.2 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 [137] and projected Landweber methods (see [15]), are particular cases of the CQ algorithm.

13.2.1 The Landweber algorithm

With x^0 arbitrary and $k = 0, 1, ...$ let

$$x^{k+1} = x^k + \gamma A^T(b - Ax^k). \tag{13.3}$$

This is the Landweber algorithm.

13.2.2 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

$$x^{k+1} = P_C(x^k + \gamma A^T(b - Ax^k)). \tag{13.4}$$

13.2.3 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.

13.2.4 The Simultaneous ART (SART)

Another example of the CQ algorithm is the *simultaneous algebraic reconstruction technique* (SART) [4] for solving $Ax = b$, for nonnegative matrix A. Let A be an I by J matrix with nonnegative entries. Let $A_{i+} > 0$ be the sum of the entries in the ith row of A and $A_{+j} > 0$ be the sum of the entries in the jth column of A. Consider the (possibly inconsistent) system $Ax = b$. The SART algorithm has the following iterative step:

$$x_j^{k+1} = x_j^k + \frac{1}{A_{+j}} \sum\nolimits_{i=1}^{I} A_{ij}(b_i - (Ax^k)_i)/A_{i+}.$$

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 Theorem 27.1, once we know that the largest eigenvalue of $B^T B$ is less than two; in fact, we show that it is one [53].

If $B^T B$ 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 $B^T B$ also had an eigenvalue greater than one. Therefore, we assume, without loss of generality, that A has all positive entries. Since the new $B^T B$ also has only positive entries, this matrix is irreducible and the Perron-Frobenius theorem applies. We shall use this to complete the proof.

Let $u = (u_1, ..., u_J)^T$ with $u_j = (A_{+j})^{1/2}$ and $v = (v_1, ..., v_I)^T$, with $v_i = (A_{i+})^{1/2}$. Then we have $Bu = v$ and $B^T v = u$; that is, u is an eigenvector of $B^T B$ with associated eigenvalue equal to one, and all the entries of u are positive, by assumption. The Perron-Frobenius theorem applies and tells us that the eigenvector associated with the largest eigenvalue has all positive entries. Since the matrix $B^T B$ is symmetric its eigenvectors are orthogonal; therefore u itself must be an eigenvector associated with the largest eigenvalue of $B^T B$. The convergence of SART follows.

13.2.5 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 [97] 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.

13.2.6 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 [193].

In a recent paper [63] Censor *et al* discuss the application of the CQ algorithm to the problem of intensity-modulated radiation therapy treatment planning. Details concerning this application are in a later chapter.

Chapter 14

Conjugate-Direction Methods in Optimization

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.

14.1 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

$$x^k = x^{k-1} + \alpha_k d^k. \tag{14.1}$$

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 14.1 *Differentiate the function $f(x^{k-1} + \alpha d^k)$ with respect to the variable α to show that*

$$\nabla f(x^k) \cdot d^k = 0. \tag{14.2}$$

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* [146]:

$$x^{k+1} = x^k - \alpha_{k+1} \nabla f(x^k).$$

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 [154] 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.

14.2 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

$$f(x) = \frac{1}{2}||b - Ax||_2^2. \tag{14.3}$$

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 (14.1) for $f(x)$ as in Equation (14.3). 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 14.2 *Show that the optimal α_k is*

$$\alpha_k = \frac{r^k \cdot d^k}{d^k \cdot Qd^k}, \tag{14.4}$$

where $r^k = c - Qx^{k-1}$.

Exercise 14.3 *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 14.1 *Let* $\{d^1, ..., d^J\}$ *be any finite set whose span is all of* R^J*. Let* α_k *be chosen according to Equation (14.4). 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 14.2 *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. ∎

14.3 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 $\hat{x}$.

14.3.1 Conjugate Directions

From Equation (14.2) we have

$$(c - Qx^{k+1}) \cdot d^k = 0,$$

which can be expressed as

$$(\hat{x} - x^{k+1}) \cdot Qd^k = (\hat{x} - x^{k+1})^T Q d^k = 0.$$

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 14.4 *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.

14.3.2 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,

$$p^j = v^j - \frac{p^1 \cdot Qv^j}{p^1 \cdot Qp^1} p^1 - ... - \frac{p^{j-1} \cdot Qv^j}{p^{j-1} \cdot Qp^{j-1}} p^{j-1}. \tag{14.5}$$

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 [146]. 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 14.3 *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

$$p^{n+1} = Qp^n - \frac{Qp^n \cdot Qp^n}{p^n \cdot Qp^n} p^n - \frac{Qp^{n-1} \cdot Qp^n}{p^{n-1} \cdot Qp^{n-1}} p^{n-1}. \tag{14.6}$$

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 (14.6) 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 (14.6) also occur in the Gram-Schmidt approach in Equation (14.5); the point is that Equation (14.6) uses only the first three terms, in every case.

14.4 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

$$x^n = x^{n-1} + \alpha_n p^n. \tag{14.7}$$

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

$$p^{n+1} = r^{n+1} - \frac{r^{n+1} \cdot Qp^n}{p^n \cdot Qp^n} p^n. \tag{14.8}$$

Since the α_n is the optimal choice and

$$r^{n+1} = -\nabla f(x^n),$$

we have, according to Equation (14.2),

$$r^{n+1} \cdot p^n = 0.$$

Exercise 14.5 *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 14.3 establishes that the set $\{p^1, ..., p^J\}$ is a conjugate set [146, 154]. In fact, we can say more.

Theorem 14.4 *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 [154].

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 14.6 *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 14.7 *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 14.8 *Show that $r^{n+1} \cdot r^j = 0$, for $j = 1, ..., n - 1$. Hint: use the induction hypothesis.*

Exercise 14.9 *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 14.10 *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 14.11 *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 [6]).

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 [154].

Part IV

More Applications

Chapter 15

Magnetic-Resonance Imaging

Fourier-transform estimation and extrapolation techniques play a major role in the rapidly expanding field of *magnetic-resonance imaging* (MRI)[115].

15.1 An Overview of MRI

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 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. 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.

15.2 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.

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.

15.3 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.*

15.4 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 apply 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.

15.5 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.

15.5.1 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.

15.5.2 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

$$e^{i\omega_0 t}e^{i\gamma G_y yt},$$

so that, after T seconds,we have

$$e^{i\omega_0 T}e^{i\gamma G_y yT}.$$

For $t \geq T$, the harmonic $e^{i\omega_0 t}$ returns, but now it is

$$e^{i\omega_0 t}e^{i\gamma G_y yT}.$$

The effect is to introduce a phase shift of $\gamma G_y yT$. 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 $e^{i\omega_0 t}e^{i\gamma G_y yT}$ is transformed into

$$e^{i\omega_0 t}e^{i\gamma G_y yT}e^{i\gamma G_x xt}.$$

The received signal is now

$$S(t) = e^{i\omega_0 t}\int\int \rho(x,y)e^{i\gamma G_y yT}e^{i\gamma G_x xt}dxdy,$$

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

$$\int\int \rho(x,y)e^{i\gamma G_y yT}e^{i\gamma G_x xt}dxdy,$$

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.

15.6 The General Formulation

The external magnetic field generated in the MRI scanner is generally described by

$$H(r,t) = (H_0 + \mathbf{G}(t)\cdot\mathbf{r})\mathbf{k} + H_1(t)(\cos(\omega t)\mathbf{i} + \sin(\omega t)\mathbf{j}). \qquad (15.1)$$

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 $\mathbf{G}(t)$.

15.7 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

$$S(t) = \int_{R^3} \rho(\mathbf{r}) \exp(-i\gamma(\int_0^t \mathbf{G}(s)ds) \cdot \mathbf{r}) \exp(-t/T_2) d\mathbf{r}, \qquad (15.2)$$

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 $x \int_0^t G_x(s)ds$.

15.7.1 An Example of $\mathbf{G}(t)$

Suppose now that $g > 0$ and θ is an arbitrary direction vector. Let

$$\mathbf{G}(t) = g\theta, \text{ for } \tau \leq t, \tag{15.3}$$

and $\mathbf{G}(t) = 0$ otherwise. Then the received signal $S(t)$ is

$$S(t) = \int_{R^3} \rho(\mathbf{r}) \exp(-i\gamma g(t-\tau)\theta \cdot \mathbf{r}) d\mathbf{r}$$

$$= (2\pi)^{3/2} \hat{\rho}(\gamma g(t-\tau)\theta), \tag{15.4}$$

for $\tau \leq t << T_2$, where $\hat{\rho}$ denotes the three-dimensional Fourier transform of the function $\rho(\mathbf{r})$.

From Equation (15.4) 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.

15.7.2 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

$$t_n = n\Delta t + \tau, \mathrm{for}\, n = 1, ..., N,$$

$$g_{2k} = k\Delta g,$$

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.

Chapter 16

Intensity-Modulated Radiation Therapy

In [63] Censor *et al.* extend the CQ algorithm to solve what they call the *multiple-set split feasibility problem* (MSSFP) . In the sequel [64] this extended CQ algorithm is used to determine dose intensities for *intensity-modulated radiation therapy* (IMRT) that satisfy both dose constraints and radiation-source constraints.

16.1 The Extended CQ Algorithm

For $n = 1, ..., N$, let C_n be a nonempty, closed convex subset of R^J. For $m = 1, ..., M$, let Q_m be a nonempty, closed convex subset of R^I. Let D be a real I by J matrix. The MSSFP is to find a member x of $C = \cap_{n=1}^N C_n$ for which $h = Dx$ is a member of $Q = \cap_{m=1}^M Q_m$. A somewhat more general problem is to find a minimizer of the proximity function

$$p(x) = \frac{1}{2}\sum_{n=1}^N \alpha_n ||P_{C_n}x - x||_2^2 + \frac{1}{2}\sum_{m=1}^M \beta_m ||P_{Q_m}Dx - Dx||_2^2, \tag{16.1}$$

with respect to the nonempty, closed convex set $\Omega \subseteq R^N$, where α_n and β_m are positive and

$$\sum_{n=1}^N \alpha_n + \sum_{m=1}^M \beta_m = 1.$$

They show that $\nabla p(x)$ is L-Lipschitz, for

$$L = \sum_{n=1}^N \alpha_n + \rho(D^T D)\sum_{m=1}^M \beta_m.$$

The algorithm given in [63] has the iterative step

$$x^{k+1}=P_\Omega\Bigg(x^k+s\Big(\sum_{n=1}^N\alpha_n(P_{C_n}x^k-x^k)+\sum_{m=1}^M\beta_m D^T(P_{Q_m}Dx^k-Dx^k)\Big)\Bigg),\tag{16.2}$$

for $0<s<2/L$. This algorithm converges to a minimizer of $p(x)$ over Ω, whenever such a minimizer exists, and to a solution, within Ω, of the MSSFP, whenever such solutions exist.

16.2 Intensity-Modulated Radiation Therapy

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. 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 target volumes* (PTV), and an upper bound on the dosage delivered to those regions designated as *organs at risk* (OAR).

16.3 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* (EUD) function $e_t(h)$ by

$$e_t(h)=(\frac{1}{N_t}\sum_{i\in S_t}(h_i)^\alpha)^{1/\alpha},\tag{16.3}$$

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.

16.4 The Algorithm

The constraint sets are convex sets of the form $\{x|f(x) \leq 0\}$, for particular convex functions f. Therefore, the cyclic subgradient projection (CSP) method is used to find the solution to the MSSFP.

Part V

Appendices

Chapter 17

Basic Concepts

In iterative methods, we begin with an initial vector, say x^0, and, for each nonnegative integer k, we calculate the next vector, x^{k+1}, from the current vector x^k. The limit of such a sequence of vectors $\{x^k\}$, when the limit exists, is the desired solution to our problem. The fundamental tools we need to understand iterative algorithms are the geometric concepts of distance between vectors and mutual orthogonality of vectors, the algebraic concept of transformation or operator on vectors, and the vector-space notions of subspaces and convex sets.

17.1 The Geometry of Euclidean Space

We denote by R^J the real Euclidean space consisting of all J-dimensional column vectors $x = (x_1, ..., x_J)^T$ with real entries x_j; here the superscript T denotes the transpose of the 1 by J matrix (or, row vector) $(x_1, ..., x_J)$. We denote by C^J the collection of all J-dimensional column vectors $x = (x_1, ..., x_J)^\dagger$ with complex entries x_j; here the superscript $\dagger$ denotes the conjugate transpose of the 1 by J matrix (or, row vector) $(x_1, ..., x_J)$. When discussing matters that apply to both R^J and C^J we denote the underlying space simply as $\mathcal{X}$.

17.1.1 Inner Products

For $x = (x_1, ..., x_J)^T$ and $y = (y_1, ..., y_J)^T$ in R^J, the dot product $x \cdot y$ is defined to be

$$x \cdot y = \sum_{j=1}^{J} x_j y_j.$$

Note that we can write

$$x \cdot y = y^T x = x^T y,$$

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

$$||x||_2 = \sqrt{x \cdot x} = \sqrt{x^T x}.$$

The *Euclidean distance* between two vectors x and y in R^J is $||x - y||_2$. As we discuss in the chapter on metric spaces, there are other norms on $\mathcal{X}$; nevertheless, in this chapter we focus on the 2-norm of x.

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

$$x \cdot y = \sum_{j=1}^{J} x_j \overline{y_j}.$$

Note that we can write

$$x \cdot y = y^\dagger x.$$

The norm, or Euclidean length, of x is

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

As in the real case, the distance between vectors x and y is $||x - y||_2$.

Both of the spaces R^J and C^J, along with their dot products, are examples of finite-dimensional Hilbert space. Much of what follows in this chapter applies to both R^J and C^J. In such cases, we shall simply refer to the underlying space as $\mathcal{X}$ and refer to the associated dot product using the *inner product* notation $\langle x, y \rangle$.

17.1.2 Cauchy's Inequality

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

$$|\langle x, y \rangle| \leq ||x||_2 ||y||_2,$$

with equality if and only if $y = \alpha x$, for some scalar α.

Proof of Cauchy's inequality: To prove Cauchy's inequality for the complex vector dot product, we write $x \cdot y = |x \cdot y| e^{i\theta}$. Let t be a real variable and consider

$$0 \leq ||e^{-i\theta} x - ty||_2^2 = (e^{-i\theta} x - ty) \cdot (e^{-i\theta} x - ty)$$

$$= ||x||_2^2 - t[(e^{-i\theta} x) \cdot y + y \cdot (e^{-i\theta} x)] + t^2 ||y||_2^2$$

$$= ||x||_2^2 - t[(e^{-i\theta}x)\cdot y + \overline{(e^{-i\theta}x)\cdot y}] + t^2||y||_2^2$$

$$= ||x||_2^2 - 2Re(te^{-i\theta}(x\cdot y)) + t^2||y||_2^2$$

$$= ||x||_2^2 - 2Re(t|x\cdot y|) + t^2||y||_2^2 = ||x||_2^2 - 2t|x\cdot y| + t^2||y||_2^2.$$

This is a nonnegative quadratic polynomial in the variable t, so it cannot have two distinct real roots. Therefore, the discriminant $4|x\cdot y|^2 - 4||y||_2^2||x||_2^2$ must be nonpositive; that is, $|x\cdot y|^2 \leq ||x||_2^2||y||_2^2$. This is Cauchy's inequality. ▮

Exercise 17.1 *Use Cauchy's inequality to show that*

$$||x+y||_2 \leq ||x||_2 + ||y||_2;$$

this is called the triangle inequality.

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

Exercise 17.2 *Prove the Parallelogram Law:*

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

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

17.2 Hyperplanes in Euclidean Space

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 $H(a,\gamma) = \{z|\langle a, z\rangle = \gamma\}$.

Exercise 17.3 *Show that the vector a is orthogonal to the hyperplane $H = H(a,\gamma)$; that is, if u and v are in H, then a is orthogonal to $u - v$.*

For an arbitrary vector x in $\mathcal{X}$ and arbitrary hyperplane $H = H(a,\gamma)$, the *orthogonal projection* of x onto H is the member $z = P_Hx$ of H that is closest to x.

Exercise 17.4 *Show that, for $H = H(a,\gamma)$, $z = P_Hx$ is the vector*

$$z = P_Hx = x + (\gamma - \langle a, x\rangle)a. \tag{17.1}$$

For $\gamma = 0$, the hyperplane $H = H(a, 0)$ is also a *subspace* of $\mathcal{X}$, meaning that, for every x and y in H and scalars α and β, the linear combination $\alpha x + \beta y$ is again in H; in particular, the zero vector 0 is in $H(a, 0)$.

17.3 Convex Sets in Euclidean Space

A subset C of $\mathcal{X}$ is said to be *convex* if, for every pair of members x and y of C, and for every α in the open interval $(0,1)$, the vector $\alpha x + (1-\alpha)y$ is also in C.

Exercise 17.5 *Show that the unit ball U in $\mathcal{X}$, 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.*

A convex set C is said to be *closed* if it contains all the vectors that lie on its boundary. We say that $d \geq 0$ is the distance from the point x to the set C if, for every $\epsilon > 0$, there is c_ϵ in C, with $||x - c_\epsilon||_2 < d + \epsilon$, and no c in C with $||x - c||_2 < d$.

Exercise 17.6 *Show that, if C is closed and $d = 0$, then x is in C.*

Proposition 17.1 *Given any nonempty closed convex set C and an arbitrary vector x in $\mathcal{X}$, 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 $\max(x_j, 0)$.

17.4 Basic Linear Algebra

In this section we discuss systems of linear equations, Gaussian elimination, basic and non-basic variables, the fundamental subspaces of linear algebra and eigenvalues and norms of square matrices.

17.4.1 Bases

A subset S of $\mathcal{X}$ is a *subspace* if, for every x and y in S, and every scalars α and β, the vector $\alpha x + \beta y$ is again in S. A collection of vectors $\{u^1, ..., u^N\}$

in $\mathcal{X}$ is *linearly independent* if there is no collection of scalars $\alpha_1, ..., \alpha_N$, not all zero, such that

$$0 = \alpha_1 u^1 + ... + \alpha_n u^N.$$

The *span* of a collection of vectors $\{u^1, ..., u^N\}$ in $\mathcal{X}$ is the set of all vectors x that can be written as linear combinations of the u^n; that is, there are scalars $c_1, ..., c_N$, such that

$$x = c_1 u^1 + ... + c_N u^N.$$

A collection of vectors $\{u^1, ..., u^N\}$ in $\mathcal{X}$ is called a *basis* for a subspace S if the collection is linearly independent and S is their span. A collection $\{u^1, ..., u^N\}$ is called *orthonormal* if $||u^n||_2 = 1$, for all n, and $(u^m)^\dagger u^n = 0$, for $m \neq n$.

17.4.2 Systems of Linear Equations

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

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

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

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

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:

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

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

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

and N is the matrix

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

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

$$Ax = Bx_B + Nx_N = 0,$$

so that

$$x_B = -B^{-1}Nx_N. \tag{17.2}$$

17.4.3 Real and Complex Systems

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. 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. 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

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

by $\tilde{x}$ the real vector

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

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

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

Exercise 17.7 *Show that x satisfies the system $Ax = b$ if and only if $\tilde{x}$ satisfies the system $\tilde{A}\tilde{x} = \tilde{b}$.*

Exercise 17.8 *Show that the eigenvalues of the Hermitian matrix*

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

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

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

and

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

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

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

and

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

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

Exercise 17.9 *Show that B is Hermitian if and only if the real matrix $\tilde{B}$ is symmetric.*

Exercise 17.10 *Let B be Hermitian. For any $x = x^1 + ix^2$, let $\tilde{x}' = (-x^2, x^1)^T$. Show that the following are equivalent: 1) $Bx = \lambda x$; 2) $\tilde{B}\tilde{x} = \lambda\tilde{x}$; 3) $\tilde{B}\tilde{x}' = \lambda\tilde{x}'$.*

Exercise 17.11 *Show that $B^\dagger Bx = c$ if and only if $\tilde{B}^T\tilde{B}\tilde{x} = \tilde{c}$.*

Exercise 17.12 *Say that the complex square matrix N is non-expansive (with respect to the Euclidean norm) if $||Nx||_2 \leq ||x||_2$, for all x. Show that N is non-expansive if and only if $\tilde{N}$ is non-expansive.*

Exercise 17.13 *Say that the complex square matrix A is averaged if there is a non-expansive N and scalar α in the interval $(0, 1)$, with $A = (1 - \alpha)I + \alpha N$, where I is the identity matrix. Show that A is averaged if and only if $\tilde{A}$ is averaged.*

17.4.4 The Fundamental Subspaces

We begin with some definitions. Let S be a subspace of finite-dimensional Euclidean space C^J. We denote by $S^\perp$ the set of vectors u that are orthogonal to every member of S; that is,

$$S^\perp = \{u | u^\dagger s = 0, \text{for every } s \in S\}.$$

Let A be an I by J matrix. Then $CS(A)$, the column space of A, is the subspace of R^I consisting of all the linear combinations of the columns of A; we also say that $CS(A)$ is the *range* of A. The null space of $A^\dagger$, denoted $NS(A^\dagger)$, is the subspace of C^I containing all the vectors w for which $A^\dagger w = 0$.

Exercise 17.14 *Show that $CS(A)^\perp = NS(A^\dagger)$. Hint: If $v \in CS(A)^\perp$, then $v^\dagger Ax = 0$ for all x, including $x = A^\dagger v$.*

Exercise 17.15 *Show that $CS(A) \cap NS(A^\dagger) = \{0\}$. Hint: If $y = Ax \in NS(A^\dagger)$ consider $||y||_2^2 = y^\dagger y$.*

The *four fundamental subspaces* of linear algebra are $CS(A), NS(A^\dagger), CS(A^\dagger)$ and $NS(A)$.

Exercise 17.16 *Show that $Ax = b$ has solutions if and only if the associated Björck-Elfving equations $AA^\dagger z = b$ has solutions.*

Let Q be a I by I matrix. We denote by $Q(S)$ the set

$$Q(S) = \{t | \text{there exists}\, s \in S \,\text{with}\, t = Qs\}$$

and by $Q^{-1}(S)$ the set

$$Q^{-1}(S) = \{u | Qu \in S\}.$$

Note that the set $Q^{-1}(S)$ is defined whether or not Q is invertible.

Exercise 17.17 *Let S be any subspace of C^I. Show that if Q is invertible and $Q(S) = S$ then $Q^{-1}(S) = S$. Hint: If $Qt = Qs$ then $t = s$.*

Exercise 17.18 *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 Hermitian Q for which $Q^{-1}(S)^\perp$ and $Q(S^\perp)$ are different.*

We assume, now, that Q is Hermitian and invertible and that the matrix $A^\dagger A$ is invertible. Note that the matrix $A^\dagger Q^{-1} A$ need not be invertible under these assumptions. We shall denote by S an arbitrary subspace of R^J.

Exercise 17.19 *Show that $Q(S) = S$ if and only if $Q(S^\perp) = S^\perp$. Hint: Use Exercise 17.18.*

Exercise 17.20 *Show that if $Q(CS(A)) = CS(A)$ then $A^\dagger Q^{-1} A$ is invertible. Hint: Show that $A^\dagger Q^{-1} Ax = 0$ if and only if $x = 0$. Recall that $Q^{-1} Ax \in CS(A)$, by Exercise 17.17. Then use Exercise 17.15.*

17.5 Linear and Nonlinear Operators

In our study of iterative algorithms we shall be concerned with sequences of vectors $\{x^k|k = 0, 1, ...\}$. The core of an iterative algorithm is the transition from the current vector x^k to the next one x^{k+1}. To understand the algorithm, we must understand the operation (or operator) T by which x^k is transformed into $x^{k+1} = Tx^k$. An *operator* is any function T defined on $\mathcal{X}$ with values again in $\mathcal{X}$.

Exercise 17.21 *Prove the following identity relating an arbitrary operator T on $\mathcal{X}$ to its complement $G = I - T$:*

$$||x - y||_2^2 - ||Tx - Ty||_2^2 = 2Re(\langle Gx - Gy, x - y\rangle) - ||Gx - Gy||_2^2. \tag{17.3}$$

Exercise 17.22 *Use the previous exercise to prove that*

$$Re(\langle Tx - Ty, x - y\rangle) - ||Tx - Ty||_2^2 = Re(\langle Gx - Gy, x - y\rangle) - ||Gx - Gy||_2^2. \tag{17.4}$$

17.5.1 Linear and Affine Linear Operators

For example, if $\mathcal{X} = C^J$ and A is a J by J complex matrix, then we can define an operator T by setting $Tx = Ax$, for each x in C^J; here Ax denotes the multiplicaton of the matrix A and the column vector x. Such operators are *linear operators*:

$$T(\alpha x + \beta y) = \alpha Tx + \beta Ty,$$

for each pair of vectors x and y and each pair of scalars α and β.

Exercise 17.23 *Show that, for $H = H(a, \gamma)$, $H_0 = H(a, 0)$, and any x and y in $\mathcal{X}$,*

$$P_H(x + y) = P_Hx + P_Hy - P_H0,$$

so that

$$P_{H_0}(x + y) = P_{H_0}x + P_{H_0}y,$$

that is, the operator P_{H_0} is an additive operator. Also, show that

$$P_{H_0}(\alpha x) = \alpha P_{H_0}x,$$

so that P_{H_0} is a linear operator. Show that we can write P_{H_0} as a matrix multiplication:

$$P_{H_0}x = (I - aa^\dagger)x.$$

If d is a fixed nonzero vector in C^J, the operator defined by $Tx = Ax+d$ is not a linear operator; it is called an *affine linear operator*.

Exercise 17.24 *Show that, for any hyperplane $H = H(a, \gamma)$ and $H_0 = H(a, 0)$,*

$$P_H x = P_{H_0} x + P_H 0,$$

so P_H is an affine linear operator.

Exercise 17.25 *For $i = 1, ..., I$ let H_i be the hyperplane $H_i = H(a^i, \gamma_i)$, $H_{i0} = H(a^i, 0)$, and P_i and P_{i0} the orthogonal projections onto H_i and H_{i0}, respectively. Let T be the operator $T = P_I P_{I-1} \cdots P_2 P_1$. Show that T is an affine linear operator, that is, T has the form*

$$Tx = Bx + d,$$

for some matrix B and some vector d. Hint: Use the previous exercise and the fact that P_{i0} is linear to show that

$$B = (I - a^I (a^I)^\dagger) \cdots (I - a^1 (a^1)^\dagger).$$

Exercise 17.26 *Let A be a complex I by J matrix with $I < J$, b a fixed vector in C^I, and S the affine subspace of C^J consisting of all vectors x with $Ax = b$. Denote by $P_S z$ the orthogonal projection of vector z onto S. Assume that A has rank I, so that the matrix $AA^\dagger$ is invertible. Show that*

$$P_S z = (I - A^\dagger (AA^\dagger)^{-1} A) z + A^\dagger (AA^\dagger)^{-1} b.$$

Hint: note that, if $z = 0$, then $P_S z$ is the minimum-norm solution of the system $Ax = b$.

17.5.2 Orthogonal Projection onto Convex Sets

For an arbitrary nonempty closed convex set C in $\mathcal{X}$, 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 17.2 *For a given x, a vector z in C is $P_C x$ if and only if*

$$Re(\langle c - z, z - x \rangle) \geq 0,$$

for all c in the set C.

Proof: For simplicity, we consider only the real case, $\mathcal{X} = R^J$. Let c be arbitrary in C and α in $(0, 1)$. Then

$$||x - P_C x||_2^2 \leq ||x - (1 - \alpha) P_C x - \alpha c||_2^2 = ||x - P_C x + \alpha (P_C x - c)||_2^2$$

$$= ||x - P_C x||_2^2 - 2\alpha\langle x - P_C x, c - P_C x\rangle + \alpha^2||P_C x - c||_2^2.$$

Therefore,

$$-2\alpha\langle x - P_C x, c - P_C x\rangle + \alpha^2||P_C x - c||_2^2 \geq 0,$$

so that

$$2\langle x - P_C x, c - P_C x\rangle \leq \alpha||P_C x - c||_2^2.$$

Taking the limit, as $\alpha \to 0$, we conclude that

$$\langle c - P_C x, P_C x - x\rangle \geq 0.$$

If z is a member of C that also has the property

$$\langle c - z, z - x\rangle \geq 0,$$

for all c in C, then we have both

$$\langle z - P_C x, P_C x - x\rangle \geq 0,$$

and

$$\langle z - P_C x, x - z\rangle \geq 0.$$

Adding on both sides of these two inequalities lead to

$$\langle z - P_C x, P_C x - z\rangle \geq 0.$$

But,

$$\langle z - P_C x, P_C x - z\rangle = -||z - P_C x||_2^2,$$

so it must be the case that $z = P_C x$. This completes the proof. ▮

Exercise 17.27 *Let C be a fixed, non-empty, closed convex subset of $\mathcal{X}$, and x not in C. Where are the vectors z for which $P_C z = P_C x$? Prove your conjecture.*

Corollary 17.1 *Let S be any subspace of $\mathcal{X}$. Then, for any x in $\mathcal{X}$ and s in S, we have*

$$\langle P_S x - x, s\rangle = 0.$$

Exercise 17.28 *Prove Corollary 17.1. Hints: since S is a subspace, $s + P_S x$ is again in S, for all s, as is cs, for every scalar c.*

Corollary 17.2 *Let S be any subspace of $\mathcal{X}$, d a fixed vector, and V the affine subspace $V = S + d = \{v = s + d | s \in S\}$, obtained by translating the members of S by the vector d. Then, for every x in $\mathcal{X}$ and every v in V, we have*

$$\langle P_V x - x, v - P_V x\rangle = 0.$$

Exercise 17.29 *Prove Corollary 17.2. Hints: 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 17.3 *Let H be the hyperplane $H(a, \gamma)$. Then, for every x, and every h in H, we have*

$$\langle P_H x - x, h - P_H x \rangle = 0.$$

Corollary 17.4 *Let S be a subspace of $\mathcal{X}$. Then, every x in $\mathcal{X}$ can be written as $x = s + u$, for a unique s in S and a unique u in $S^\perp$.*

Exercise 17.30 *Prove Corollary 17.4. Hint: the vector $P_S x - x$ is in $S^\perp$.*

Corollary 17.5 *Let S be a subspace of $\mathcal{X}$. Then $(S^\perp)^\perp = S$.*

Exercise 17.31 *Prove Corollary 17.5. Hint: every x in $\mathcal{X}$ has the form $x = s + u$, with s in S and u in $S^\perp$. Suppose x is in $(S^\perp)^\perp$. Show $u = 0$.*

17.5.3 Gradient Operators

Another important example of a nonlinear operator is the gradient of a real-valued function of several variables. Let $f(x) = f(x_i, ..., x_J)$ be a real number for each vector x in R^J. The *gradient* of f at the point x is the vector whose entries are the partial derivatives of f; that is,

$$\nabla f(x) = (\frac{\partial f}{\partial x_1}(x), ..., \frac{\partial f}{\partial x_J}(x))^T.$$

The operator $Tx = \nabla f(x)$ is linear only if the function $f(x)$ is quadratic; that is, $f(x) = x^T A x$ for some square matrix x, in which case the gradient of f is $\nabla f(x) = \frac{1}{2}(A + A^T)x$.

If u is any vector in $\mathcal{X}$ with $||u||_2 = 1$, then u is said to be a *direction vector*. Let $f : R^J \to R$. The *directional derivative* of f, at the point x, in the direction of u, is

$$D_u f(x) = \lim_{t \to 0} (1/t)(f(x + tu) - f(x)),$$

if this limit exists. If the partial derivatives of f are continuous, then

$$D_u f(x) = u_1 \frac{\partial f}{\partial x_1}(x) + ... + u_J \frac{\partial f}{\partial x_J}(x).$$

It follows from the Cauchy Inequality that $|D_u f(x)| \leq ||\nabla f(x)||_2$, with equality if and only if u is parallel to the gradient vector, $\nabla f(x)$. The gradient points in the direction of the greatest increase in $f(x)$.

Chapter 18

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

$$f(t) = \cos \omega t + i \sin \omega t, \tag{18.1}$$

for some real frequency ω. Complex sinusoids are also called *complex exponential functions*.

18.1 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 (18.1) 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.

18.2 Taylor-series expansions

The Taylor series expansion for the exponential function $g(t) = e^t$ is

$$e^t = 1 + t + \frac{1}{2!}t^2 + \frac{1}{3!}t^3 + ... \tag{18.2}$$

If we replace t with $i\omega$, where $i = \sqrt{-1}$, we obtain

$$e^{i\omega} = (1 - \frac{1}{2!}\omega^2 + \frac{1}{4!}\omega^4 - ...) + i(\omega - \frac{1}{3!}\omega^3 + \frac{1}{5!}\omega^5 - ...). \tag{18.3}$$

We recognize the two series in Equation (18.3) 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 (18.1) 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.

18.3 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 18.1 *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 18.2 *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

$$\sum_{n=M}^{N} e^{i\omega n} = e^{i\frac{M+N}{2}} \frac{\sin(\omega \frac{N-M+1}{2})}{\sin(\frac{\omega}{2})}. \tag{18.4}$$

Exercise 18.3 *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

$$\sum_{n=M}^{N} \sin(\omega n) = \sin(\frac{M+N}{2}) \frac{\sin(\omega \frac{N-M+1}{2})}{\sin(\frac{\omega}{2})}.$$

Part VI

Appendices

Chapter 19

The Fourier Transform

In this chapter we review the basic properties of the Fourier transform.

19.1 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

$$F(\gamma) = \int_{-\infty}^{\infty} f(x) e^{i\gamma x} dx. \tag{19.1}$$

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, [106]).

19.1.1 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*:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\gamma) e^{-i\gamma x} d\gamma. \tag{19.2}$$

The functions $f(x)$ and $F(\gamma)$ are called a *Fourier-transform pair*. Once again, the proper interpretation of Equation (19.2) 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. To illustrate some of the issues involved, we consider the functions in the Schwartz class [106]

19.1.2 Functions in the Schwartz class

A function $f(x)$ is said to be in the *Schwartz class*, or to be a *Schwartz function* if $f(x)$ is infinitely differentiable and

$$|x|^m f^{(n)}(x) \to 0$$

as x goes to $-\infty$ and $+\infty$. Here $f^{(n)}(x)$ denotes the nth derivative of $f(x)$. An example of a Schwartz function is $f(x) = e^{-x^2}$, with Fourier transform $F(\gamma) = \sqrt{\pi}e^{-\gamma^2/4}$. If $f(x)$ is a Schwartz function, then so is its Fourier transform. To prove the Fourier Inversion Formula it is sufficient to show that

$$f(0) = \int_{-\infty}^{\infty} F(\gamma)d\gamma/2\pi.$$

Write

$$f(x) = f(0)e^{-x^2} + (f(x) - f(0)e^{-x^2}) = f(0)e^{-x^2} + g(x). \qquad (19.3)$$

Then $g(0) = 0$, so $g(x) = xh(x)$. Then the Fourier transform of $g(x)$ is the derivative of the Fourier transform of $h(x)$; that is,

$$G(\gamma) = H'(\gamma).$$

The function $H(\gamma)$ is a Schwartz function, so it goes to zero at the infinities. Computing the Fourier transform of both sides of Equation (19.3), we obtain

$$F(\gamma) = f(0)\sqrt{\pi}e^{-\gamma^2/4} + H'(\gamma). \qquad (19.4)$$

Therefore,

$$\int_{-\infty}^{\infty} F(\gamma)d\gamma = 2\pi f(0) + H(+\infty) - H(-\infty) = 2\pi f(0).$$

To prove the Fourier Inversion Formula, we let $K(\gamma) = F(\gamma)e^{-ix_0\gamma}$, for fixed x_0. Then the inverse Fourier transform of $K(\gamma)$ is $k(x) = f(x + x_0)$, and therefore

$$\int_{-\infty}^{\infty} K(\gamma)d\gamma = 2\pi k(0) = 2\pi f(x_0).$$

In the next subsection we consider a discontinuous $f(x)$.

19.1.3 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.

19.1.4 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.

19.2 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

$$\delta(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\gamma x} d\gamma. \tag{19.5}$$

Obviously, this integral cannot be understood in the usual way. The integral in Equation (19.5) is a symbolic way of saying that

$$\int h(x)(\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\gamma x} d\gamma)dx = \int h(x)\delta(x)dx = h(0), \tag{19.6}$$

for all $h(x)$ that are continuous at $x = 0$; that is, the integral in Equation (19.5) has the sifting property, so it acts like $\delta(x)$. Interchanging the order of integration in Equation (19.6), 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.

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, the magnitude of the sum of the signals received from various parts of the object is much smaller than we might expect, especially when A is large. For smaller A the signals received at a sensor are much more *in phase* with one another, and so the magnitude of the sum remains large. A more quantitative statement of this phenomenon is provided by the *uncertainty principle* (see [55]).

19.3 Practical Limitations

In actual remote-sensing problems, antennas 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; in the direct transmission problem we measure $F(\gamma)$ only for $\gamma = k$, with $|k| \leq \frac{\omega}{c}$. 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.

19.3.1 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

$$g(x) = \frac{1}{2\pi}\int F(\gamma)H(\gamma)e^{-i\gamma x}dx. \tag{19.7}$$

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 magnitude of $H(\gamma)$, $|H(\gamma)|$, 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 (19.7), replacing $F(\gamma)$ and $H(\gamma)$ with their definitions, as given by Equation (19.1). Then we have

$$g(x) = \int (\int f(t)e^{i\gamma t}dt)(\int h(s)e^{i\gamma s}ds)e^{-i\gamma x}d\gamma.$$

Interchanging the order of integration, we get

$$g(x) = \int\int f(t)h(s)(\int e^{i\gamma(t+s-x)}d\gamma)dsdt.$$

Now using Equation (19.5) to replace the inner integral with $\delta(t+s-x)$, the next integral becomes

$$\int h(s)\delta(t+s-x)ds = h(x-t).$$

Finally, we have

$$g(x) = \int f(t)h(x-t)dt; \tag{19.8}$$

this is the definition of the convolution of the functions f and h.

19.3.2 Low-Pass Filtering

A major problem in image reconstruction is the removal of blurring, which is often modelled 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.

19.4 Two-Dimensional Fourier Transforms

More generally, we consider a function $f(x,z)$ of two real variables. Its Fourier transformation is

$$F(\alpha,\beta) = \int\int f(x,z)e^{i(x\alpha+z\beta)}dxdz. \tag{19.9}$$

For example, suppose that $f(x,z) = 1$ for $\sqrt{x^2+z^2} \leq R$, and zero, otherwise. Then we have

$$F(\alpha,\beta)=\int_{-\pi}^{\pi}\int_0^R e^{-i(\alpha r\cos\theta+\beta r\sin\theta)}rdrd\theta.$$

In polar coordinates, with $\alpha=\rho\cos\phi$ and $\beta=\rho\sin\phi$, we have

$$F(\rho,\phi)=\int_0^R\int_{-\pi}^{\pi}e^{ir\rho\cos(\theta-\phi)}d\theta rdr.$$

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

$$\int_0^z t^nJ_{n-1}(t)dt=z^nJ_n(z),$$

we have

$$F(\rho,\phi)=\frac{2\pi R}{\rho}J_1(\rho R).$$

Notice that, since $f(x,z)$ is a radial function, that is, dependent only on the distance from $(0,0)$ to (x,z), 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 [134].

19.4.1 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

$$f(x,y)=\frac{1}{4\pi^2}\int\int F(\alpha,\beta)e^{-i(\alpha x+\beta y)}d\alpha d\beta. \tag{19.10}$$

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.

Chapter 20

The Fast Fourier Transform (FFT)

A fundamental problem in signal processing is to estimate finitely many values of the function $F(\omega)$ from finitely many values of its (inverse) Fourier transform, $f(t)$. As we have seen, 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 [77]. 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.

20.1 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

$$P(x) = a_0 + a_1x + a_2x^2 + ... + a_{2K}x^{2K},$$

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

$$P(x) = (a_0 + a_2x^2 + ... + a_{2K}x^{2K}) + x(a_1 + a_3x^2 + ... + a_{2K-1}x^{2K-2})$$

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.

20.2 The DFT and Vector DFT

Suppose that the data are the samples are $\{f(n\Delta), n = 1, ..., N\}$, where $\Delta > 0$ is the sampling increment or sampling spacing.

The DFT estimate of $F(\omega)$ is the function $F_{DFT}(\omega)$, defined for ω in $[-\pi/\Delta, \pi/\Delta]$, and given by

$$F_{DFT}(\omega) = \Delta \sum_{n=1}^{N} f(n\Delta) e^{in\Delta\omega}.$$

The DFT estimate $F_{DFT}(\omega)$ is data consistent; its inverse Fourier-transform value at $t = n\Delta$ 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 $f(t)$.

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 $DFT_{\mathbf{f}}(\omega)$, defined for ω in $[0, 2\pi)$, given by

$$DFT_{\mathbf{f}}(\omega) = \sum_{n=0}^{N-1} f_n e^{in\omega}.$$

Let $\mathbf{F}$ be the complex N-dimensional vector $\mathbf{F} = (F_0, F_1, ..., F_{N-1})^T$, where $F_k = DFT_{\mathbf{f}}(2\pi k/N), k = 0, 1, ..., N-1$. So the vector $\mathbf{F}$ consists of N values of the function $DFT_{\mathbf{f}}$, taken at N equispaced points $2\pi/N$ apart in $[0, 2\pi)$.

From the formula for $DFT_{\mathbf{f}}$ we have, for $k = 0, 1, ..., N-1$,

$$F_k = F(2\pi k/N) = \sum_{n=0}^{N-1} f_n e^{2\pi i n k/N}. \tag{20.1}$$

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.

20.3 Exploiting Redundancy

Suppose that $N = 2M$ is even. We can rewrite Equation (20.1) as follows:

$$F_k = \sum_{m=0}^{M-1} f_{2m} e^{2\pi i(2m)k/N} + \sum_{m=0}^{M-1} f_{2m+1} e^{2\pi i(2m+1)k/N},$$

or, equivalently,

$$F_k = \sum_{m=0}^{M-1} f_{2m} e^{2\pi imk/M} + e^{2\pi ik/N} \sum_{m=0}^{M-1} f_{2m+1} e^{2\pi imk/M}. \tag{20.2}$$

Note that if $0 \leq k \leq M-1$ then

$$F_{k+M} = \sum_{m=0}^{M-1} f_{2m} e^{2\pi imk/M} - e^{2\pi ik/N} \sum_{m=0}^{M-1} f_{2m+1} e^{2\pi imk/M}, \tag{20.3}$$

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 (20.2) and (20.3) 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 (20.2) we see that $C(N) = 2C(N/2) + (N/2)$. Applying the same reasoning to get $C(N/2) = 2C(N/4) + (N/4)$, we obtain

$$C(N) = 2C(N/2) + (N/2) = 4C(N/4) + 2(N/2) = ...$$

$$= 2^L C(N/2^L) + L(N/2) = N + L(N/2).$$

Therefore, the cost required to calculate $\mathbf{F}$ is approximately $N \log_2 N$.

From our earlier discussion of discrete linear filters and convolution, we see that the FFT can be used to calculate the periodic convolution (or even the nonperiodic convolution) of finite length vectors.

Finally, let's return to the original context of estimating the Fourier transform $F(\omega)$ of function $f(t)$ from finitely many samples of $f(t)$. If we have N equispaced samples, we can use them to form the vector $\mathbf{f}$ and perform the FFT algorithm to get vector $\mathbf{F}$ consisting of N values of the DFT estimate of $F(\omega)$. It may happen that we wish to calculate more than N values of the DFT estimate, perhaps to produce a smooth looking graph. We can still use the FFT, but we must trick it into thinking we have more data that the N samples we really have. We do this by *zero-padding*. Instead of creating the N-dimensional vector $\mathbf{f}$, we make a longer vector by appending, say, J zeros to the data, to make a vector that has dimension $N+J$. The DFT estimate is still the same function of ω, since we have only included new zero coefficients as fake data; but, the FFT thinks we have $N+J$ data values, so it returns $N+J$ values of the DFT, at $N+J$ equispaced values of ω in $[0,2\pi)$.

20.4 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 equispaced 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 21

Fourier Transform Estimation

In many remote-sensing problems, the measured data is related to the function to be imaged by Fourier transformation. In the *Fourier* approach to tomography, the data are often viewed as line integrals through the object of interest. These line integrals can then be converted into values of the Fourier transform of the object function. In magnetic-resonance imaging (MRI), adjustments to the external magnetic field cause the measured data to be Fourier-related to the desired proton-density function. In such applications, the imaging problem becomes a problem of estimating a function from finitely many noisy values of its Fourier transform. To overcome these limitations, one can use iterative and non-iterative methods for incorporating prior knowledge and regularization; data-extrapolation algorithms form one class of such methods. We focus on the use of iterative algorithms for improving resolution through extrapolation of Fourier-transform data.

21.1 The Limited-Fourier-Data Problem

For notational convenience, we shall discuss only the one-dimensional case, involving the estimation of the (possibly complex-valued) function $f(x)$ of the real variable x, from finitely many values $F(\omega_n)$, $n = 1, ..., N$ of its Fourier transform. Here we adopt the definitions

$$F(\omega) = \int f(x)e^{ix\omega}dx,$$

and

$$f(x) = \frac{1}{2\pi}\int F(\omega)e^{-ix\omega}d\omega.$$

Because it is the case in the applications of interest to us here, we shall assume that the object function has bounded support, that is, there is $A > 0$, such that $f(x) = 0$ for $|x| > A$.

The values $\omega = \omega_n$ at which we have measured the function $F(\omega)$ may be structured in some way; they may be equi-spaced along a line, or, in the higher-dimensional case, arranged in a cartesian grid pattern, as in MRI. According to the Central Slice Theorem, the Fourier data in tomography lie along rays through the origin. Nevertheless, in what follows, we shall not assume any special arrangement of these data points.

Because the data are finite, there are infinitely many functions $f(x)$ consistent with the data. We need some guidelines to follow in selecting a best estimate of the true $f(x)$. First, we must remember that the data values are noisy, so we want to avoid overfitting the estimate to noisy data. This means that we should include regularization in whatever method we adopt. Second, the limited data is often insufficient to provide the desired resolution, so we need to incorporate additional prior knowledge about $f(x)$, such as non-negativity, upper and lower bounds on its values, its support, its overall shape, and so on. Third, once we have selected prior information to include, we should be conservative in choosing an estimate consistent with that information. This may involve the use of constrained minimum-norm solutions. Fourth, we should not expect our prior information to be perfectly accurate, so our estimate should not be overly sensitive to slight changes in the prior information. Finally, the estimate we use will be one for which there are good algorithms for its calculation.

21.2 Minimum-Norm Estimation

To illustrate the notion of minimum-norm estimation, we begin with the finite-dimensional problem of solving an underdetermined system of linear equations, $Ax = b$, where A is a rea I by J matrix with $J > I$ and AA^T is invertible.

21.2.1 The Minimum-Norm Solution of $Ax = b$

Each equation can be written as

$$b_i = (a^i)^T x = \langle x, a^i \rangle,$$

where the vector a^i is the ith column of the matrix A^T and $\langle u, v \rangle$ denoted the inner, or dot product of the vectors u and v.

Exercise 21.1 *Show that every vector x in R^J can be written as*

$$x = A^T z + w, \tag{21.1}$$

with $Aw = 0$ and

$$||x||_2^2 = ||A^T z||_2^2 + ||w||_2^2.$$

Consequently, $Ax = b$ if and only if $A(A^T z) = b$ and $A^T z$ is the solution having the smallest norm. This minimum-norm solution $\hat{x} = A^T z$ can be found explicitly; it is

$$\hat{x} = A^T z = A^T (AA^T)^{-1} b. \tag{21.2}$$

Hint: multiply both sides of Equation (21.1) by A and solve for z.

It follows from this exercise that the minimum-norm solution $\hat{x}$ of $Ax = b$ has the form $\hat{x} = A^T z$, which means that $\hat{x}$ is a linear combination of the a^i:

$$\hat{x} = \sum_{i=1}^{I} z_i a^i.$$

21.2.2 Minimum-Weighted-Norm Solution of $Ax = b$

As we shall see later, it is sometimes convenient to introduce a new norm for the vectors. Let Q be a J by J symmetric positive-definite matrix and define

$$||x||_Q^2 = x^T Q x.$$

With $Q = C^T C$, where C is the positive-definite symmetric square-root of Q, we can write

$$||x||_Q^2 = ||y||_2^2,$$

for $y = Cx$. Now suppose that we want to find the solution of $Ax = b$ for which $||x||_Q^2$ is minimum. We write

$$Ax = b$$

as

$$AC^{-1} y = b,$$

so that, from Equation (21.2), we find that the solution y with minimum norm is

$$\hat{y} = (AC^{-1})^T (AC^{-1}(AC^{-1})^T)^{-1} b,$$

or

$$\hat{y} = (AC^{-1})^T (AQ^{-1}A^T)^{-1} b,$$

so that the $\hat{x}_Q$ with minimum weighted norm is

$$\hat{x}_Q = C^{-1}\hat{y} = Q^{-1}A^T (AQ^{-1}A^T)^{-1} b, \tag{21.3}$$

Notice that, writing

$$\langle u, v \rangle_Q = u^T Q v,$$

we find that

$$b_i = \langle Q^{-1}a^i, \hat{x}_Q\rangle_Q,$$

and the minimum-weighted-norm solution of $Ax = b$ is a linear combination of the columns g^i of $Q^{-1}A^T$, that is,

$$\hat{x}_Q = \sum_{i=1}^{I} d_i g^i,$$

where

$$d_i = ((AQ^{-1}A^T)^{-1}b)_i,$$

for each $i = 1, ..., I$.

21.3 Fourier-Transform Data

Returning now to the case in which we have finitely many values of the Fourier transform of $f(x)$, we write

$$F(\omega) = \int f(x)e^{ix\omega}dx = \langle e_\omega, f\rangle\,,$$

where $e_\omega(x) = e^{-ix\omega}$ and

$$\langle g, h\rangle = \int g(x)h(x)dx.$$

The norm of a function $f(x)$ is then

$$||f||_2 = \sqrt{\langle f, f\rangle} = \sqrt{\int |f(x)|^2 dx}.$$

21.3.1 The Minimum-Norm Estimate

Arguing as we did in the finite-dimensional case, we conclude that the minimum-norm solution of the data-consistency equations

$$F(\omega_n) = \langle e_{\omega_n}, f\rangle\,, n = 1, ..., N,$$

has the form

$$\hat{f}(x) = \sum_{n=1}^{N} a_n e^{-ix\omega_n}.$$

If the integration assumed to extend over the whole real line, the functions $e_\omega(x)$ are mutually orthogonal and so

$$a_n = \frac{1}{2\pi}F(\omega_n). \tag{21.4}$$

In most applications, however, the function $f(x)$ is known to have finite support.

Exercise 21.2 *Show that, if $f(x) = 0$ for x outside the interval $[a, b]$, then the coefficients a_n satisfy the system of linear equations*

$$F(\omega_n) = \sum_{m=1}^{N} G_{nm} a_m,$$

with

$$G_{nm} = \int_a^b e^{ix(\omega_n - \omega_m)} dx.$$

For example, suppose that $[a, b] = [-\pi, \pi]$ and

$$\omega_n = -\pi + \frac{2\pi}{N} n,$$

for $n = 1, ..., N$

Exercise 21.3 *Show that, in this example, $G_{nn} = 2\pi$ and $G_{nm} = 0$, for $n \neq m$. Therefore, for this special case, we again have*

$$a_n = \frac{1}{2\pi} F(\omega_n).$$

21.3.2 Minimum-Weighted-Norm Estimates

Let $p(x) \geq 0$ be a weight function. Let

$$\langle g, h \rangle_p = \int g(x)h(x)p(x)^{-1} dx,$$

with the understanding that $p(x)^{-1} = 0$ outside of the support of $p(x)$. The associated weighted norm is then

$$||f||_p = \sqrt{\int |f(x)|^2 p(x)^{-1} dx}.$$

We can then write

$$F(\omega_n) = \langle pe_\omega, f \rangle_p = \int (p(x)e^{-ix\omega}) f(x)p(x)^{-1} dx.$$

It follows that the function consistent with the data and having the minimum weighted norm has the form

$$\hat{f}_p(x) = p(x) \sum_{n=1}^{N} b_n e^{-ix\omega_n}. \tag{21.5}$$

Exercise 21.4 *Show that the coefficients b_n satisfy the system of linear equations*

$$F(\omega_n) = \sum_{m=1}^{N} b_m P_{nm}, \tag{21.6}$$

with

$$P_{nm} = \int p(x) e^{ix(\omega_n - \omega_m)} dx,$$

for $m, n = 1, ..., N$.

Whenever we have prior information about the support of $f(x)$, or about the shape of $|f(x)|$, we can incorporate this information through our choice of the weight function $p(x)$. In this way, the prior information becomes part of the estimate, through the first factor in Equation (21.5), with the second factor providing information gathered from the measurement data. This minimum-weighted-norm estimate of $f(x)$ is called the PDFT, and is discussed in more detail in [56].

Once we have $\hat{f}_p(x)$, we can take its Fourier transform, $\hat{F}_p(\omega)$, which is then an estimate of $F(\omega)$. Because the coefficients b_n satisfy Equations (21.6), we know that

$$\hat{F}_p(\omega_n) = F(\omega_n),$$

for $n = 1, ..., N$. For other values of ω, the estimate $\hat{F}_p(\omega)$ provides an extrapolation of the data. For this reason, methods such as the PDFT are sometimes called *data-extrapolation methods.* If $f(x)$ is supported on an interval $[a, b]$, then the function $F(\omega)$ is said to be *band-limited.* If $[c, d]$ is an interval containing $[a, b]$ and $p(x) = 1$, for x in $[c, d]$, and $p(x) = 0$ otherwise, then the PDFT estimate is a non-iterative version of the Gerchberg-Papoulis band-limited extrapolation estimate of $f(x)$ (see [56]).

21.3.3 Implementing the PDFT

The PDFT can be extended easily to the estimation of functions of several variables. However, there are several difficult steps that can be avoided by iterative implementation. Even in the one-dimensional case, when the values ω_n are not equispaced, the calculation of the matrix P can be messy. In the case of higher dimensions, both calculating P and solving for the coefficients can be expensive. In the next section we consider an iterative implementation that solves both of these problems.

21.4 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,

$$F(\omega_n) = \sum_{j=1}^{J} f_j E_{nj},$$

where $E_{nj} = e^{ix_j\omega_n}$. 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 by letting the positive-definite matrix Q be the diagonal matrix having for its jth diagonal entry

$$Q_{jj} = 1/p(x_j),$$

if $p(x_j) > 0$, and zero, otherwise.

21.4.1 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 underdetermined case, the ART provides the the solution closest to the starting vector, in the sense of the Eucliean 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.

21.4.2 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, implemented using ART, we regularize the ART algorthm, as we discussed earlier.

For recent work on the PDFT and DPDFT, the reader should consult the papers by Shieh, et al, available on my website.

Chapter 22

Using Prior Knowledge in Remote Sensing

The problem is to reconstruct a (possibly complex-valued) function $f : R^D \to C$ 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.

22.1 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 multi-variable 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.

22.2 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

$$g_n = \int_S f(r)\overline{h_n(r)}dr, \tag{22.1}$$

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 (22.1) in the context of a Hilbert space, and to write

$$g_n = \langle f, h_n \rangle, \tag{22.2}$$

where the usual Hilbert space inner product is defined by

$$\langle f, h \rangle_2 = \int_S f(r)\overline{h(r)}dr, \tag{22.3}$$

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

$$||f||_2 = \sqrt{\int_S |f(r)|^2 dr} \tag{22.4}$$

is finite.

22.2.1 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

$$\mathcal{H}^\dagger a(r) = a_1 h_1(r) + ... + a_N h_N(r). \tag{22.5}$$

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

$$g = \mathcal{H}\mathcal{H}^\dagger a \tag{22.6}$$

for the vector a. Then the function

$$\hat{f}(r) = \mathcal{H}^\dagger a(r) \tag{22.7}$$

is consistent with the measured data and is the function in $L^2(S)$ of least norm for which this is true. The function $w(r) = f(r) - \hat{f}(r)$ has the property $\mathcal{H}w = 0$.

Exercise 22.1 *Show that* $||f||_2^2 = ||\hat{f}||_2^2 + ||w||_2^2$

The estimate $\hat{f}(r)$ is the *minimum-norm solution*, with respect to the norm defined in Equation (22.4). 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

$$\langle \mathcal{T}f, h\rangle_2 = \langle f, \mathcal{T}^\dagger h\rangle_2.$$

The adjoint operator will change when we change the inner product.

22.3 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

$$\langle f, h\rangle_{\mathcal{T}} = \langle \mathcal{T}^{-1}f, \mathcal{T}^{-1}h\rangle_2. \tag{22.8}$$

We can then use this inner product to define the problem to be solved. We now say that

$$g_n = \langle f, t^n\rangle_{\mathcal{T}}, \tag{22.9}$$

for known functions $t^n(x)$. Using the definition of the $\mathcal{T}$ inner product, we find that

$$g_n = \langle f, h^n\rangle_2 = \langle \mathcal{T}f, \mathcal{T}h^n\rangle_{\mathcal{T}}.$$

The adjoint operator for $\mathcal{T}$, with respect to the $\mathcal{T}$-norm, is denoted $\mathcal{T}^*$, and is defined by

$$\langle \mathcal{T}f, h\rangle_{\mathcal{T}} = \langle f, \mathcal{T}^*h\rangle_{\mathcal{T}}.$$

Therefore,

$$g_n = \langle f, \mathcal{T}^*\mathcal{T}h^n\rangle_{\mathcal{T}}.$$

Exercise 22.2 *Show that* $\mathcal{T}^*\mathcal{T} = \mathcal{T}\mathcal{T}^\dagger$.

Consequently, we have

$$g_n = \langle f, \mathcal{T}\mathcal{T}^\dagger h^n\rangle_{\mathcal{T}}. \tag{22.10}$$

22.4 Minimum-$\mathcal{T}$-Norm Solutions

The function $\tilde{f}$ consistent with the data and having the smallest $\mathcal{T}$-norm has the algebraic form

$$\hat{f} = \sum_{m=1}^{N} a_m \mathcal{T}\mathcal{T}^\dagger h^m. \tag{22.11}$$

Applying the $\mathcal{T}$-inner product to both sides of Equation (22.11), we get

$$g_n = \langle \hat{f}, \mathcal{T}\mathcal{T}^\dagger h^n\rangle_{\mathcal{T}}$$

$$= \sum_{m=1}^{N} a_m \langle \mathcal{T}\mathcal{T}^\dagger h^m, \mathcal{T}\mathcal{T}^\dagger h^n\rangle_{\mathcal{T}}.$$

Therefore,

$$g_n = \sum_{m=1}^{N} a_m \langle \mathcal{T}^\dagger h^m, \mathcal{T}^\dagger h^n\rangle_2. \tag{22.12}$$

We solve this system for the a_m and insert them into Equation (22.11) to get our reconstruction. The Gram matrix that appears in Equation (22.12) is positive-definite, but is often ill-conditioned; increasing the main diagonal by a percent or so usually is sufficient regularization.

22.5 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

$$g_n = \int_S f(x)e^{-i\omega_n x}dx,$$

for arbitrary values ω_n.

22.5.1 The $L^2(-\pi, \pi)$ Case

Assume that $f(x) = 0$, for $|x| > \pi$. The minimum-2-norm solution has the form

$$\hat{f}(x) = \sum_{m=1}^{N} a_m e^{i\omega_m x}, \tag{22.13}$$

with

$$g_n = \sum_{m=1}^{N} a_m \int_{-\pi}^{\pi} e^{i(\omega_m - \omega_n)x}dx.$$

For the equispaced values $\omega_n = n$ we find that $a_m = g_m$ and the minimum-norm solution is

$$\hat{f}(x) = \sum_{n=1}^{N} g_n e^{inx}. \tag{22.14}$$

22.5.2 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 equispaced data at $\omega_n = n$, we have

$$g_n = \int_{-\pi}^{\pi} f(x)\chi_A(x)e^{-inx}dx,$$

so that the minimum-norm solution has the form

$$\hat{f}(x) = \chi_A(x) \sum_{m=1}^{N} a_m e^{imx},$$

with

$$g_n = 2\sum_{m=1}^{N} a_m \frac{\sin A(m-n)}{m-n}.$$

The minimum-norm solution is support-limited to $[-A, A]$ and consistent with the Fourier-transform data.

22.5.3 Using a Prior Estimate of f

Suppose that $f(x) = 0$ for $|x| > \pi$ again, and that $p(x)$ satisfies

$$0 < \epsilon \leq p(x) \leq E < +\infty,$$

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

$$\langle f, h\rangle_{\mathcal{T}} = \int_{-\pi}^{\pi} f(x)\overline{h(x)}p(x)^{-1}dx.$$

It follows that

$$g_n = \int_{-\pi}^{\pi} f(x)p(x)e^{-inx}p(x)^{-1}dx,$$

so that the minimum $\mathcal{T}$-norm solution is

$$\hat{f}(x) = \sum_{m=1}^{N} a_m p(x)e^{imx} = p(x)\sum_{m=1}^{N} a_m e^{imx}, \tag{22.15}$$

where

$$g_n = \sum_{m=1}^{N} a_m \int_{-\pi}^{\pi} p(x)e^{i(m-n)x}dx.$$

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 (22.15) was presented in [33], where it was called the PDFT method. The PDFT was based on an earlier non-iterative version of the Gerchberg-Papoulis bandlimited extrapolation procedure [32]. The PDFT was then applied to image reconstruction problems in [34]. An application of the PDFT was presented in [37]. In [36] 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 [39] and in [40] 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 [56].

Chapter 23

Iterative Optimization

Optimization means finding a maximum or minimum value of a real-valued function of one or several variables. Constrained optimization means that the acceptable solutions must satisfy some additional restrictions, such as being nonnegative. Even if we know equations that optimal points must satisfy, solving these equations is often difficult and usually cannot be done algebraically. In this chapter we sketch the conditions that must hold in order for a point to be an optimum point, and then use those conditions to motivate iterative algorithms for finding the optimum points. We shall consider only minimization problems, since any maximization problem can be converted into a minimization problem by changing the sign of the function involved.

23.1 Functions of a Single Real Variable

If $f(x)$ is a continuous, real-valued function of a real variable x and we want to find an x for which the function takes on its minimum value, then we need only examine those places where the derivative, $f'(x)$, is zero, and those places where $f'(x)$ does not exist; of course, without further assumptions, there is no guarantee that a minimum exists. Therefore, if $f(x)$ is differentiable at all x, and if its minimum value occurs at x^*, then $f'(x^*) = 0$. If the problem is a *constrained minimization*, that is, if the allowable x lie within some interval, say, $[a, b]$, then we must also examine the end-points, $x = a$ and $x = b$. If the constrained minimum occurs at $x^* = a$ and $f'(a)$ exists, then $f'(a)$ need not be zero; however, we must have $f'(a) \geq 0$, since, if $f'(a) < 0$, we could select $x = c$ slightly to the right of $x = a$ with $f(c) < f(a)$. Similarly, if the minimum occurs at $x = b$, and $f'(b)$ exists, we must have $f'(b) \leq 0$. We can combine these end-point conditions by saying that if the minimum occurs at one of the two end-points, moving away from the minimizing point into the interval $[a, b]$ cannot result in the function growing smaller. For functions of several variables similar conditions hold, involving the partial derivatives of the function.

23.2 Functions of Several Real Variables

Suppose, from now on, that $f(x) = f(x_1, ..., x_N)$ is a continuous, real-valued function of the N real variables $x_1, ..., x_N$ and that $x = (x_1, ..., x_N)^T$ is the column vector of unknowns, lying in the N-dimensional space R^N. When the problem is to find a minimum (or a maximum) of $f(x)$, we call $f(x)$ the *objective function.* As in the case of one variable, without additional assumptions, there is no guarantee that a minimum (or a maximum) exists.

23.2.1 Cauchy's Inequality for the Dot Product

For any two vectors v and w in R^N the dot product is defined to be

$$v \cdot w = \sum_{n=1}^{N} v_n w_n.$$

Cauchy's inequality tells us that $|v \cdot w| \leq ||v||_2 ||w||_2$, with equality if and only if $w = \alpha v$ for some real number α. In the multi-variable case we speak of the derivative of a function at a point, in the direction of a given vector; these are the *directional derivatives* and their definition involves the dot product.

23.2.2 Directional Derivatives

If $\frac{\partial f}{\partial x_n}(z)$, the partial derivative of f, with respect to the variable x_n, at the point z, is defined for all z, and $u = (u_1, ..., u_N)^T$ is a vector of length one, that is, its norm,

$$||u||_2 = \sqrt{u_1^2 + ... + u_N^2},$$

equals one, then the derivative of $f(x)$, at a point $x = z$, in the direction of u, is

$$\frac{\partial f}{\partial x_1}(z)u_1 + ... + \frac{\partial f}{\partial x_N}(z)u_N.$$

Notice that this directional derivative is the dot product of u with the gradient of $f(x)$ at $x = z$, defined by

$$\nabla f(z) = (\frac{\partial f}{\partial x_1}(z), ..., \frac{\partial f}{\partial x_N}(z))^T.$$

According to Cauchy's inequality, the dot product $\nabla f(z) \cdot u$ will take on its maximum value when u is a positive multiple of $\nabla f(z)$, and therefore, its minimum value when u is a negative multiple of $\nabla f(z)$. Consequently, the gradient of $f(x)$ at $x = z$ points in the direction, from $x = z$, of the greatest increase in the function $f(x)$. This suggests that, if we are trying to minimize $f(x)$, and we are currently at $x = z$, we should consider moving in the direction of $-\nabla f(z)$; this leads to Cauchy's iterative method of *steepest descent*, which we shall discuss in more detail later.

If the minimum value of $f(x)$ occurs at $x = x^*$, then either all the directional derivatives are zero at $x = x^*$, in which case $\nabla f(z) = 0$, or at least one directional derivative does not exist. But, what happens when the problem is a constrained minimization?

23.2.3 Constrained Minimization

Unlike the single-variable case, in which constraining the variable simply meant requiring that it lie within some interval, in the multi-variable case constraints can take many forms. For example, we can require that each of the entries x_n be nonnegative, or that each x_n lie within an interval $[a_n, b_n]$ that depends on n, or that the norm of x, defined by $||x||_2 = \sqrt{x_1^2 + ... + x_N^2}$, which measures the distance from x to the origin, does not exceed some bound. In fact, for any set C in N-dimensional space, we can pose the problem of minimizing $f(x)$, subject to the restriction that x be a member of the set C. In place of end-points, we have what are called boundary-points of C, which are those points in C that are not entirely surrounded by other points in C. For example, in the one-dimensional case, the points $x = a$ and $x = b$ are the boundary-points of the set $C = [a, b]$. If $C = R^N_+$ is the subset of N-dimensional space consisting of all the vectors x whose entries are nonnegative, then the boundary-points of C are all nonnegative vectors x having at least one zero entry.

Suppose that C is arbitrary in R^N and the point $x = x^*$ is the solution to the problem of minimizing $f(x)$ over all x in the set C. Assume also that all the directional derivatives of $f(x)$ exist at each x. If x^* is not a boundary-point of C, then all the directional derivatives of $f(x)$, at the point $x = x^*$, must be nonnegative, in which case they must all be zero, so that we must have $\nabla f(z) = 0$. On the other hand, speaking somewhat loosely, if x^* is a boundary-point of C, then it is necessary only that the directional derivatives of $f(x)$, at the point $x = x^*$, in directions that point back into the set C, be nonnegative.

23.2.4 An Example

To illustrate these concepts, consider the problem of minimizing the function of two variables, $f(x_1, x_2) = x_1 + 3x_2$, subject to the constraint that

$x = (x_1, x_2)$ lie within the unit ball $C = \{x = (x_1, x_2)|x_1^2 + x_2^2 \leq 1\}$. With the help of simple diagrams we discover that the minimizing point $x^* = (x_1^*, x_2^*)$ is a boundary-point of C, and that the line $x_1+3x_2 = x_1^*+3x_2^*$ is tangent to the unit circle at x^*. The gradient of $f(x)$, at $x = z$, is $\nabla f(z) = (1, 3)^T$, for all z, and is perpendicular to this tangent line. But, since the point x^* lies on the unit circle, the vector $(x_1^*, x_2^*)^T$ is also perpendicular to the line tangent to the circle at x^*. Consequently, we know that $(x_1^*, x_2^*)^T = \alpha(1, 3)^T$, for some real α. From $x_1^2 + x_2^2 = 1$, it follows that $|\alpha| = \sqrt{10}$. This gives us two choices for x^*: either $x^* = (\sqrt{10}, 3\sqrt{10})$, or $x^* = (-\sqrt{10}, -3\sqrt{10})$. Evaluating $f(x)$ at both points reveals that $f(x)$ attains its maximum at the first, and its minimum at the second.

Every direction vector u can be written in the form $u = \beta(1, 3)^T + \gamma(-3, 1)^T$, for some β and γ. The directional derivative of $f(x)$, at $x = x^*$, in any direction that points from $x = x^*$ back into C, must be nonnegative. Such directions must have a nonnegative dot product with the vector $(-x_1^*, -x_2^*)^T$, which tells us that

$$0 \leq \beta(1, 3)^T \cdot (-x_1^*, -x_2^*)^T + \gamma(-3, 1)^T \cdot (-x_1^*, x_2^*)^T,$$

or

$$0 \leq (3\gamma - \beta)x_1^* + (-3\beta - \gamma)x_2^*.$$

Consequently, the gradient $(1, 3)^T$ must have a nonnegative dot product with every direction vector u that has a nonnegative dot product with $(-x_1^*, -x_2^*)^T$. For the dot product of $(1, 3)^T$ with any u to be nonnegative we need $\beta \geq 0$. So we conclude that $\beta \geq 0$ for all β and γ for which

$$0 \leq (3\gamma - \beta)x_1^* + (-3\beta - \gamma)x_2^*.$$

Saying this another way, if $\beta < 0$ then

$$(3\gamma - \beta)x_1^* + (-3\beta - \gamma)x_2^* < 0,$$

for all γ. Taking the limit, as $\beta \to 0$ from the left, it follows that

$$3\gamma x_1^* - \gamma x_2^* \leq 0,$$

for all γ. The only way this can happen is if $3x_1^* - x_2^* = 0$. Therefore, our optimum point must satisfy the equation $x_2^* = 3x_1^*$, which is what we found previously.

We have just seen the conditions necessary for x^* to minimize $f(x)$, subject to constraints, be used to determine the point x^* algebraically. In more complicated problems we will not be able to solve for x^* merely by performing simple algebra. But we may still be able to find x^* using iterative optimization methods.

23.3 Gradient Descent Optimization

Suppose that we want to minimize $f(x)$, over all x, without constraints. Begin with an arbitrary initial guess, $x = x^0$. Having proceeded to x^k, we show how to move to x^{k+1}. At the point $x = x^k$, the direction of greatest rate of decrease of $f(x)$ is $u = -\nabla f(x^k)$. Therefore, it makes sense to move from x^k in the direction of $-\nabla f(x^k)$, and to continue in that direction until the function stops decreasing. In other words, we let

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k),$$

where $\alpha_k \geq 0$ is the *step size*, determined by the condition

$$f(x^k - \alpha_k \nabla f(x^k)) \leq f(x^k - \alpha \nabla f(x^k)),$$

for all $\alpha \geq 0$. This iterative procedure is Cauchy's *steepest descent* method. To establish the convergence of this algorithm to a solution requires additional restrictions on the function f; we shall not consider these issues further. Our purpose here is merely to illustrate an iterative minimization philosophy that we shall recall in various contexts.

If the problem is a constrained minimization, then we must proceed more carefully. One method, known as *interior-point* iteration, begins with x^0 within the constraint set C and each subsequent step is designed to produce another member of C; if the algorithm converges, the limit is then guaranteed to be in C. For example, if $C = R^N_+$, the nonnegative cone in R^N, we could modify the steepest descent method so that, first, x^0 is a nonnegative vector, and second, the step from x^k in C is restricted so that we stop before x^{k+1} ceases to be nonnegative. A somewhat different modification of the steepest descent method would be to take the full step from x^k to x^{k+1}, but then to take as the true x^{k+1} that vector in C nearest to what would have been x^{k+1}, according to the original steepest descent algorithm; this new iterative scheme is the *projected steepest descent* algorithm. It is not necessary, of course, that every intermediate vector x^k be in C; all we want is that the limit be in C. However, in applications, iterative methods must always be stopped before reaching their limit point, so, if we must have a member of C for our (approximate) answer, then we would need x^k in C when we stop the iteration.

23.4 The Newton-Raphson Approach

The Newton-Raphson approach to minimizing a real-valued function $f : R^J \to R$ involves finding x^* such that $\nabla f(x^*) = 0$.

23.4.1 Functions of a Single Variable

We begin with the problem of finding a root of a function $g : R \to R$. If x^0 is not a root, compute the line tangent to the graph of g at $x = x^0$ and let x^1 be the point at which this line intersects the horizontal axis; that is,

$$x^1 = x^0 - g(x^0)/g'(x^0).$$

Continuing in this fashion, we have

$$x^{k+1} = x^k - g(x^k)/g'(x^k).$$

This is the *Newton-Raphson algorithm* for finding roots. Convergence, when it occurs, is more rapid than gradient descent, but requires that x^0 be sufficiently close to the solution.

Now suppose that $f : R \to R$ is a real-valued function that we wish to minimize by solving $f'(x) = 0$. Letting $g(x) = f'(x)$ and applying the Newton-Raphson algorithm to $g(x)$ gives the iterative step

$$x^{k+1} = x^k - f'(x^k)/f''(x^k).$$

This is the Newton-Raphson optimization algorithm. Now we extend these results to functions of several variables.

23.4.2 Functions of Several Variables

The Newton-Raphson algorithm for finding roots of functions $g : R^J \to R^J$ has the iterative step

$$x^{k+1} = x^k - [\mathcal{J}(g)(x^k)]^{-1} g(x^k),$$

where $\mathcal{J}(g)(x)$ is the Jacobian matrix of first partial derivatives, $\frac{\partial g_m}{\partial x_j}(x^k)$, for $g(x) = (g_1(x), ..., g_J(x))^T$.

To minimize a function $f : R^J \to R$, we let $g(x) = \nabla f(x)$ and find a root of g. Then the Newton-Raphson iterative step becomes

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k),$$

where $\nabla^2 f(x) = \mathcal{J}(g)(x)$ is the Hessian matrix of second partial derivatives of f.

23.5 Other Approaches

Choosing the negative of the gradient as the next direction makes good sense in minimization problems, but it is not the only, or even the best, way to proceed. For least squares problems the method of conjugate directions is a popular choice (see [56]). Other modifications of the gradient can also be used, as, for example, in the EMML algorithm.

Chapter 24

Convex Sets and Convex Functions

In this chapter we consider several algorithms pertaining to convex sets and convex functions, whose convergence is a consequence of the KM theorem.

24.1 Optimizing Functions of a Single Real Variable

Let $f : R \rightarrow R$ be a differentiable function. From the Mean-Value Theorem we know that

$$f(b) = f(a) + f'(c)(b - a),$$

for some c between a and b. If there is a constant L with $|f'(x)| \leq L$ for all x, that is, the derivative is bounded, then we have

$$|f(b) - f(a)| \leq L|b - a|, \tag{24.1}$$

for all a and b; functions that satisfy Equation (24.1) are said to be *L-Lipschitz.*

Suppose $g : R \rightarrow R$ is differentiable and attains its minimum value. We want to minimize the function $g(x)$. Solving $g'(x) = 0$ to find the optimal $x = x^*$ may not be easy, so we may turn to an iterative algorithm for finding roots of $g'(x)$, or one that minimizes $g(x)$ directly. In the latter case, we may consider a steepest descent algorithm of the form

$$x^{k+1} = x^k - \gamma g'(x^k),$$

for some $\gamma > 0$. We denote by T the operator

$$Tx = x - \gamma g'(x).$$

Then, using $g'(x^*) = 0$, we find that

$$|x^* - x^{k+1}| = |Tx^* - Tx^k|.$$

We would like to know if there are choices for γ that make T an av operator. For functions $g(x)$ that are *convex*, the answer is yes.

24.1.1 The Convex Case

A function $g : R \to R$ is called *convex* if, for each pair of distinct real numbers a and b, the line segment connecting the two points $A = (a, g(a))$ and $B = (b, g(b))$ is on or above the graph of $g(x)$. The function $g(x) = x^2$ is a simple example of a convex function.

Proposition 24.1 *The following are equivalent:*
1) $g(x)$ is convex;
2) for all points $a < x < b$

$$g(x) \leq \frac{g(b) - g(a)}{b - a}(x - a) + g(a); \tag{24.2}$$

3) for all points $a < x < b$

$$g(x) \leq \frac{g(b) - g(a)}{b - a}(x - b) + g(b); \tag{24.3}$$

4) for all points a and b and for all α in the interval $(0, 1)$

$$g((1 - \alpha)a + \alpha b) \leq (1 - \alpha)g(a) + \alpha g(b). \tag{24.4}$$

Exercise 24.1 *Prove Proposition 24.1.*

Exercise 24.2 *Use Proposition 24.1 to show that, if $g(x)$ is convex, then, for every triple of points $a < x < b$, we have*

$$\frac{g(x) - g(a)}{x - a} \leq \frac{g(b) - g(a)}{b - a} \leq \frac{g(b) - g(x)}{b - x}. \tag{24.5}$$

If $g(x)$ is a differentiable function, then convexity can be expressed in terms of properties of the derivative, $g'(x)$.

Exercise 24.3 *Show that, if $g(x)$ is differentiable, then, for every triple of points $a < x < b$, we have*

$$g'(a) \leq \frac{g(b) - g(a)}{b - a} \leq g'(b). \tag{24.6}$$

We see from this exercise that, if $g(x)$ is differentiable and convex, then $g'(x)$ is an increasing function. In fact, the converse is also true, as we shall see shortly.

Recall that the line tangent to the graph of $g(x)$ at the point $x = a$ has the equation

$$y = g'(a)(x - a) + g(a).$$

Theorem 24.1 *For the differentiable function $g(x)$, the following are equivalent:*
1) $g(x)$ is convex;
2) for all a and x we have

$$g(x) \geq g(a) + g'(a)(x - a); \tag{24.7}$$

3) the derivative, $g'(x)$, is an increasing function, or, equivalently,

$$(g'(x) - g'(a))(x - a) \geq 0, \tag{24.8}$$

for all a and x.

Proof: Assume that $g(x)$ is convex. If $x > a$, then

$$g'(a) \leq \frac{g(x) - g(a)}{x - a},$$

while, if $x < a$, then

$$\frac{g(a) - g(x)}{a - x} \leq g'(a).$$

In either case, the inequality in (24.7) holds. Now, assume that the inequality in (24.7) holds. Then

$$g(x) \geq g'(a)(x - a) + g(a),$$

and

$$g(a) \geq g'(x)(a - x) + g(x).$$

Adding the two inequalities, we obtain

$$g(a) + g(x) \geq (g'(x) - g(a))(a - x) + g(a) + g(x),$$

from which we conclude that

$$(g(x) - g(a))(x - a) \geq 0.$$

So $g'(x)$ is increasing. Finally, we assume the derivative is increasing and show that $g(x)$ is convex. If $g(x)$ is not convex, then there are points $a < b$ such that, for all x in (a, b),

$$\frac{g(x) - g(a)}{x - a} > \frac{g(b) - g(a)}{b - a}.$$

By the Mean Value Theorem there is c in (a,b) with

$$g'(c) = \frac{g(b) - g(a)}{b - a}.$$

Select x in the interval (a,c). Then there is d in (a,x) with

$$g'(d) = \frac{g(x) - g(a)}{x - a}.$$

Then $g'(d) > g'(c)$, which contradicts the assumption that $g'(x)$ is increasing. This concludes the proof. ▮

If $g(x)$ is twice differentiable, we can say more.

Theorem 24.2 *If $g(x)$ is twice differentiable, then $g(x)$ is convex if and only if $g''(x) \geq 0$, for all x.*

Proof: According to the Mean Value Theorem, as applied to the function $g'(x)$, for any points $a < b$ there is c in (a,b) with $g'(b) - g'(a) = g''(c)(b-a)$. If $g''(x) \geq 0$, the right side of this equation is nonnegative, so the left side is also. Now assume that $g(x)$ is convex, which implies that $g'(x)$ is an increasing function. Since $g'(x+h) - g'(x) \geq 0$ for all $h > 0$, it follows that $g''(x) \geq 0$. ▮

Suppose that $g(x)$ is convex and the function $f(x) = g'(x)$ is L-Lipschitz. If $g(x)$ is twice differentiable, this would be the case if

$$0 \leq g''(x) \leq L,$$

for all x. As we shall see, if γ is in the interval $(0, \frac{2}{L})$, then T is an av operator and the iterative sequence converges to a minimizer of $g(x)$. In this regard, we have the following result.

Theorem 24.3 *Let $h(x)$ be convex and differentiable and $h'(x)$ non-expansive, that is,*

$$|h'(b) - h'(a)| \leq |b - a|,$$

for all a and b. Then $h'(x)$ is firmly non-expansive, which means that

$$(h'(b) - h'(a))(b - a) \geq (h'(b) - h'(a))^2.$$

Proof: Since $h(x)$ is convex and differentiable, the derivative, $h'(x)$, must be increasing. Therefore, if $b > a$, then $|b - a| = b - a$ and

$$|h'(b) - h(a)| = h'(b) - h'(a).$$

▮

If $g(x)$ is convex and $f(x) = g'(x)$ is L-Lipschitz, then $\frac{1}{L}g'(x)$ is ne, so that $\frac{1}{L}g'(x)$ is fne and $g'(x)$ is $\frac{1}{L}$-ism. Then, for $\gamma > 0$, $\gamma g'(x)$ is $\frac{1}{\gamma L}$-ism, which tells us that the operator

$$Tx = x - \gamma g'(x)$$

is av whenever $0 < \gamma < \frac{2}{L}$. It follows from the KM Theorem that the iterative sequence $x^{k+1} = Tx^k = x^k - \gamma g'(x^k)$ converges to a minimizer of $g(x)$.

In the next section we extend these results to functions of several variables.

24.2 Optimizing Functions of Several Real Variables

Let $F : R^J \to R^N$ be a R^N-valued function of J real variables. The function $F(x)$ is said to be *differentiable* at the point x^0 if there is an N by J matrix $F'(x^0)$ such that

$$\lim_{h \to 0} \frac{1}{||h||_2}[F(x^0 + h) - F(x^0) - F'(x^0)h] = 0.$$

It can be shown that, if F is differentiable at $x = x^0$, then F is continuous there as well [104].

If $f : R^J \to R$ is differentiable, then $f'(x^0) = \nabla f(x^0)$, the gradient of f at x^0. The function $f(x)$ is differentiable if each of its first partial derivatives is continuous. If the derivative $f' : R^J \to R^J$ is, itself, differentiable, then $f'' : R^J \to R^J$, and $f''(x) = H(x) = \nabla^2 f(x)$, the Hessian matrix whose entries are the second partial derivatives of f. The function $f(x)$ will be twice differentiable if each of the second partial derivatives is continuous. In that case, the mixed second partial derivatives are independent of the order of the variables, the Hessian matrix is symmetric, and the chain rule applies.

Let $f : R^J \to R$ be a differentiable function. From the Mean-Value Theorem ([104], p. 41) we know that, for any two points a and b, there is α in $(0, 1)$ such that

$$f(b) = f(a) + \langle \nabla f((1 - \alpha)a + \alpha b), b - a\rangle.$$

If there is a constant L with $||\nabla f(x)||_2 \leq L$ for all x, that is, the gradient is bounded in norm, then we have

$$|f(b) - f(a)| \leq L||b - a||_2, \tag{24.9}$$

for all a and b; functions that satisfy Equation (24.9) are said to be *L-Lipschitz.*

In addition to real-valued functions $f : R^J \to R$, we shall also be interested in functions $F : R^J \to R^J$, such as $F(x) = \nabla f(x)$, whose range is R^J, not R. We say that $F : R^J \to R^J$ is L-Lipschitz if there is $L > 0$ such that

$$||F(b) - F(a)||_2 \leq L||b - a||_2,$$

for all a and b.

Suppose $g : R^J \to R$ is differentiable and attains its minimum value. We want to minimize the function $g(x)$. Solving $\nabla g(x) = 0$ to find the optimal $x = x^*$ may not be easy, so we may turn to an iterative algorithm for finding roots of $\nabla g(x)$, or one that minimizes $g(x)$ directly. In the latter case, we may again consider a steepest descent algorithm of the form

$$x^{k+1} = x^k - \gamma\nabla g(x^k),$$

for some $\gamma > 0$. We denote by T the operator

$$Tx = x - \gamma\nabla g(x).$$

Then, using $\nabla g(x^*) = 0$, we find that

$$||x^* - x^{k+1}||_2 = ||Tx^* - Tx^k||_2.$$

We would like to know if there are choices for γ that make T an av operator. As in the case of functions of a single variable, for functions $g(x)$ that are *convex*, the answer is yes.

24.2.1 The Convex Case

The function $g(x) : R^J \to R$ is said to be *convex* if, for each pair of distinct vectors a and b and for every α in the interval $(0, 1)$ we have

$$g((1 - \alpha)a + \alpha b) \leq (1 - \alpha)g(a) + \alpha g(b).$$

The function $g(x)$ is convex if and only if, for every x and z in R^J and real t, the function $f(t) = g(x + tz)$ is a convex function of t. Therefore, the theorems for the multi-variable case can also be obtained from previous results for the single-variable case.

If $g(x)$ is a differentiable function, then convexity can be expressed in terms of properties of the derivative, $\nabla g(x)$. Note that, by the chain rule, $f'(t) = \nabla g(x + tz) \cdot z$.

Theorem 24.4 *For the differentiable function $g(x)$, the following are equivalent:*
1) $g(x)$ is convex;
2) for all a and b we have

$$g(b) \geq g(a) + \langle\nabla g(a), b - a\rangle\,; \tag{24.10}$$

3) for all a and b we have

$$\langle \nabla g(b) - \nabla g(a), b - a \rangle \geq 0. \tag{24.11}$$

As in the case of functions of a single variable, we can say more when the function $g(x)$ is twice differentiable. Note that, by the chain rule again, $f''(t) = z^T \nabla^2 g(x + tz)z$.

Theorem 24.5 *Let each of the second partial derivatives of $g(x)$ be continuous, so that $g(x)$ is twice continuously differentiable. Then $g(x)$ is convex if and only if the second derivative matrix $\nabla^2 g(x)$ is non-negative definite, for each x.*

Suppose that $g(x) : R^J \to R$ is convex and the function $F(x) = \nabla g(x)$ is L-Lipschitz. As we shall see, if γ is in the interval $(0, \frac{2}{L})$, then the operator $T = I - \gamma F$ defined by

$$Tx = x - \gamma \nabla g(x),$$

is an av operator and the iterative sequence converges to a minimizer of $g(x)$. In this regard, we have the following analog of Theorem 24.3.

Theorem 24.6 *Let $h(x)$ be convex and differentiable and its derivative, $\nabla h(x)$, non-expansive, that is,*

$$||\nabla h(b) - \nabla h(a)||_2 \leq ||b - a||_2,$$

for all a and b. Then $\nabla h(x)$ is firmly non-expansive, which means that

$$\langle \nabla h(b) - \nabla h(a), b - a \rangle \geq ||\nabla h(b) - \nabla h(a)||_2^2.$$

Unlike the proof of Theorem 24.3, the proof of this theorem is not trivial. In [111] Golshtein and Tretyakov prove the following theorem, from which Theorem 24.6 follows immediately.

Theorem 24.7 *Let $g : R^J \to R$ be convex and differentiable. The following are equivalent:*

$$||\nabla g(x) - \nabla g(y)||_2 \leq ||x - y||_2; \tag{24.12}$$

$$g(x) \geq g(y) + \langle \nabla g(y), x - y \rangle + \frac{1}{2}||\nabla g(x) - \nabla g(y)||_2^2; \tag{24.13}$$

and

$$\langle \nabla g(x) - \nabla g(y), x - y \rangle \geq ||\nabla g(x) - \nabla g(y)||_2^2. \tag{24.14}$$

Proof: The only difficult step in the proof is showing that Inequality (24.12) implies Inequality (24.13). To prove this part, let $x(t) = (1-t)y+tx$, for $0 \leq t \leq 1$. Then

$$g'(x(t)) = \langle \nabla g(x(t)), x - y\rangle,$$

so that

$$\int_0^1 \langle \nabla g(x(t)) - \nabla g(y), x - y\rangle dt = g(x) - g(y) - \langle \nabla g(y), x - y\rangle.$$

Therefore,

$$g(x) - g(y) - \langle \nabla g(y), x - y\rangle \leq \int_0^1 ||\nabla g(x(t)) - \nabla g(y)||_2 ||x(t) - y||_2 dt$$

$$\leq \int_0^1 ||x(t) - y||_2^2 dt = \int_0^1 ||t(x - y)||_2^2 dt = \frac{1}{2}||x - y||_2^2,$$

according to Inequality (24.12). Therefore,

$$g(x) \leq g(y) + \langle \nabla g(y), x - y\rangle + \frac{1}{2}||x - y||_2^2.$$

Now let $x = y - \nabla g(y)$, so that

$$g(y - \nabla g(y)) \leq g(y) + \langle \nabla g(y), \nabla g(y)\rangle + \frac{1}{2}||\nabla g(y)||_2^2.$$

Consequently,

$$g(y - \nabla g(y)) \leq g(y) - \frac{1}{2}||\nabla g(y)||_2^2.$$

Therefore,

$$\inf g(x) \leq g(y) - \frac{1}{2}||\nabla g(y)||_2^2,$$

or

$$g(y) \geq \inf g(x) + \frac{1}{2}||\nabla g(y)||_2^2. \tag{24.15}$$

Now fix y and define the function $h(x)$ by

$$h(x) = g(x) - g(y) - \langle \nabla g(y), x - y\rangle.$$

Then $h(x)$ is convex, differentiable, and non-negative,

$$\nabla h(x) = \nabla g(x) - \nabla g(y),$$

and $h(y) = 0$, so that $h(x)$ attains its minimum at $x = y$. Applying Inequality (24.15) to the function $h(x)$, with z in the role of x and x in the role of y, we find that

$$\inf h(z) = 0 \leq h(x) - \frac{1}{2}||\nabla h(x)||_2^2.$$

From the definition of $h(x)$, it follows that

$$0 \leq g(x) - g(y) - \langle \nabla g(y), x - y\rangle - \frac{1}{2}||\nabla g(x) - \nabla g(y)||_2^2.$$

This completes the proof of the implication. ▮

If $g(x)$ is convex and $f(x) = \nabla g(x)$ is L-Lipschitz, then $\frac{1}{L}\nabla g(x)$ is ne, so that $\frac{1}{L}\nabla g(x)$ is fne and $\nabla g(x)$ is $\frac{1}{L}$-ism. Then for $\gamma > 0$, $\gamma\nabla g(x)$ is $\frac{1}{\gamma L}$-ism, which tells us that the operator

$$Tx = x - \gamma\nabla g(x)$$

is av whenever $0 < \gamma < \frac{2}{L}$. It follows from the KM Theorem that the iterative sequence $x^{k+1} = Tx^k = x^k - \gamma\nabla g(x^k)$ converges to a minimizer of $g(x)$, whenever minimizers exist.

24.3 Convex Feasibility

The *convex feasibility problem* (CFP) is to find a point in the non-empty intersection C of finitely many closed, convex sets C_i in R^J. The *successive orthogonal projections* (SOP) method [114] is the following. Begin with an arbitrary x^0. For $k = 0, 1, ...,$ and $i = k(\text{mod}\, I) + 1$, let

$$x^{k+1} = P_i x^k,$$

where $P_i x$ denotes the orthogonal projection of x onto the set C_i. Since each of the operators P_i is firmly non-expansive, the product

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

is averaged. Since C is not empty, T has fixed points. By the KM Theorem, the sequence $\{x^k\}$ converges to a member of C. It is useful to note that the limit of this sequence will not generally be the point in C closest to x^0; it is if the C_i are hyperplanes, however.

24.3.1 The SOP for Hyperplanes

For any x, $P_i x$, the orthogonal projection of x onto the closed, convex set C_i, is the unique member of C_i for which

$$\langle P_i x - x, y - P_i x\rangle \geq 0,$$

for every y in C_i.

Exercise 24.4 *Show that*

$$||y - P_i x||_2^2 + ||P_i x - x||_2^2 \leq ||y - x||_2^2,$$

for all x and for all y in C_i.

When the C_i are hyperplanes, we can say more.

Exercise 24.5 *Show that, if C_i is a hyperplane, then*

$$\langle P_i x - x, y - P_i x \rangle = 0,$$

for all y in C_i. Use this result to show that

$$||y - P_i x||_2^2 + ||P_i x - x||_2^2 = ||y - x||_2^2,$$

for every y in the hyperplane C_i. Hint: since both $P_i x$ and y are in C_i, so is $P_i x + t(y - P_i x)$, for every real t.

Let the C_i be hyperplanes with C their non-empty intersection. Let $\hat{c}$ be in C.

Exercise 24.6 *Show that, for $x^{k+1} = P_i x^k$, where $i = k (\bmod I) + 1$,*

$$||\hat{c} - x^k||_2^2 - ||\hat{c} - x^{k+1}||_2^2 = ||x^k - x^{k+1}||_2^2. \tag{24.16}$$

It follows from this exercise that the sequence $\{||\hat{c} - x^k||_2\}$ is decreasing and that the sequence $\{||x^k - x^{k+1}||_2^2\}$ converges to zero. Therefore, the sequence $\{x^k\}$ is bounded, so has a cluster point, x^*, and the cluster point must be in C. Therefore, replacing $\hat{c}$ with x^*, we find that the sequence $\{||x^* - x^k||_2^2\}$ converges to zero, which means that $\{x^k\}$ converges to x^*. Summing over k on both sides of Equation (24.16), we get

$$||\hat{c} - x^*||_2^2 - ||\hat{c} - x^0||_2^2$$

on the left side, while on the right side we get a quantity that does not depend on which $\hat{c}$ in C we have selected. It follows that minimizing $||\hat{c} - x^0||_2^2$ over $\hat{c}$ in C is equivalent to minimizing $||\hat{c} - x^*||_2^2$ over $\hat{c}$ in C; the minimizer of the latter problem is clearly $\hat{c} = x^*$. So, when the C_i are hyperplanes, the SOP algorithm does converge to the member of the intersection that is closest to x^0. Note that the SOP is the ART algorithm, for the case of hyperplanes.

24.3.2 The SOP for Half-Spaces

If the C_i are half-spaces, that is, there is some I by J matrix A and vector b so that

$$C_i = \{x | (Ax)_i \geq b_i\},$$

then the SOP becomes the Agmon-Motzkin-Schoenberg algorithm. When the intersection is non-empty, the algorithm converges, by the KM Theorem, to a member of that intersection. When the intersection is empty, we get subsequential convergence to a limit cycle.

24.3.3 The SOP when C is empty

When the intersection C of the sets C_i, $i = 1, ..., I$ is empty, the SOP cannot converge. Drawing on our experience with two special cases of the SOP, the ART and the AMS algorithms, we conjecture that, for each $i = 1, ..., I$, the subsequences $\{x^{nI+i}\}$ converge to $c^{*,i}$ in C_i, with $P_i c^{*,i-1} = c^{*,i}$ for $i = 2, 3, ..., I$, and $P_1 c^{*,I} = c^{*,1}$; see [86]. The set $\{c^{*,i}\}$ is then a limit cycle. For the special case of $I = 2$ we can prove this.

Theorem 24.8 *Let C_1 and C_2 be nonempty, closed convex sets in $\mathcal{X}$, with $C_1 \cap C_2 = \emptyset$. Assume that there is a unique $\hat{c}_2$ in C_2 minimizing the function $f(x) = ||c_2 - P_1 c_2||_2$, over all c_2 in C_2. Let $\hat{c}_1 = P_1 \hat{c}_2$. Then $P_2 \hat{c}_1 = \hat{c}_2$. Let z^0 be arbitrary and, for $n = 0, 1, ...$, let*

$$z^{2n+1} = P_1 z^{2n},$$

and

$$z^{2n+2} = P_2 z^{2n+1}.$$

Then

$$\{z^{2n+1}\} \to \hat{c}_1,$$

and

$$\{z^{2n}\} \to \hat{c}_2.$$

Proof: We apply the CQ algorithm, with the iterative step given by Equation (27.2), with $C = C_2$, $Q = C_1$, and the matrix $A = I$, the identity matrix. The CQ iterative step is now

$$x^{k+1} = P_2(x^k + \gamma(P_1 - I)x^k).$$

Using the acceptable choice of $\gamma = 1$, we have

$$x^{k+1} = P_2 P_1 x^k.$$

This CQ iterative sequence then converges to $\hat{c}_2$, the minimizer of the function $f(x)$. Since $z^{2n} = x^n$, we have $\{z^{2n}\} \to \hat{c}_2$. Because

$$||P_2 \hat{c}_1 - \hat{c}_1||_2 \leq ||\hat{c}_2 - \hat{c}_1||_2,$$

it follows from the uniqueness of $\hat{c}_2$ that $P_2 \hat{c}_1 = \hat{c}_2$. This completes the proof. ∎

24.4 Optimization over a Convex Set

Suppose now that $g : R^J \to R$ is a convex, differentiable function and we want to find a minimizer of $g(x)$ over a closed, convex set C, if such minimizers exists. We saw earlier that, if $\nabla g(x)$ is L-Lipschitz, and γ is in the interval $(0, 2/L)$, then the operator $Tx = x - \gamma\nabla g(x)$ is averaged. Since P_C, the orthogonal projection onto C, is also averaged, their product, $S = P_C T$, is averaged. Therefore, by the KM Theorem, the sequence $\{x^{k+1} = Sx^k\}$ converges to a fixed point of S, whenever such fixed points exist.

Exercise 24.7 *Show that $\hat{x}$ is a fixed point of S if and only if $\hat{x}$ minimizes $g(x)$ over x in C.*

24.4.1 Linear Optimization over a Convex Set

Suppose we take $g(x) = d^T x$, for some fixed vector d. Then $\nabla g(x) = d$ for all x, and $\nabla g(x)$ is L-Lipschitz for every $L > 0$. Therefore, the operator $Tx - x - \gamma d$ is averaged, for any positive γ. Since P_C is also averaged, the product, $S = P_C T$ is averaged and the iterative sequence $x^{k+1} = Sx^k$ converges to a minimizer of $g(x) = d^T x$ over C, whenever minimizers exist.

For example, suppose that C is the closed, convex region in the plane bounded by the coordinate axes and the line $x + y = 1$. Let $d^T = (1, -1)$. The problem then is to minimize the function $g(x, y) = x - y$ over C. Let $\gamma = 1$ and begin with $x^0 = (1, 1)^T$. Then $x^0 - d = (0, 2)^T$ and $x^1 = P_C(0, 2)^T = (0, 1)^T$, which is the solution.

For this algorithm to be practical, $P_C x$ must be easy to calculate. In those cases in which the set C is more complicated than in the example, other algorithms, such as the simplex algorithm, will be preferred. We consider these ideas further, when we discuss the linear programming problem.

24.5 Geometry of Convex Sets

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 cannot be written as

$$x = (1 - \alpha)y + \alpha z,$$

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 $\mathrm{Ext}(C)$.

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.

The fundamental problem in linear programming is to minimize the function

$$f(x) = c^T x,$$

over the *feasible set* F, that is, the convex set of all $x \geq 0$ with$Ax = b$. In the next chapter we present an algebraic description of the extreme points of the feasible set F, in terms of *basic feasible solutions*, show that there are at most finitely many extreme points of F and that every member of F can be written as a convex combination of the extreme points, plus a direction of unboundedness. These results will be used to prove the basic theorems about the primal and dual linear programming problems and to describe the simplex algorithm.

24.6 Projecting onto Convex Level Sets

Suppose that $f : R^J \to R$ is a convex function and $C = \{x|f(x) \leq 0\}$. Then C is a convex set. A vector t is said to be a *subgradient* of f at x if, for all z, we have

$$f(z) - f(x) \geq \langle t, z - x\rangle.$$

Such subgradients always exist, for convex functions. If f is differentiable at x, then f has a unique subgradient, namely, its gradient, $t = \nabla f(x)$.

Unless f is a linear function, calculating the orthogonal projection, $P_C z$, of z onto C requires the solution of an optimization problem. For that reason, closed-form approximations of $P_C z$ are often used. One such approximation occurs in the *cyclic subgradient projection* (CSP) method. Given x not in C, let

$$\Pi_C x = x - \alpha t,$$

where t is any subgradient of f at x and $\alpha = \frac{f(x)}{||t||^2} > 0$.

Proposition 24.2 *For any c in C, $||c - \Pi_C x||_2^2 < ||c - x||_2^2$.*

Proof: Since x is not in C, we know that $f(x) > 0$. Then,

$$||c - \Pi_C x||_2^2 = ||c - x + \alpha t||_2^2$$

$$= ||c - x||_2^2 + 2\alpha\langle c - x, t\rangle + \alpha f(x).$$

Since t is a subgradient, we know that

$$\langle c - x, t\rangle \leq f(c) - f(x),$$

so that

$$||c - \Pi_C x||_2^2 - ||c - x||_2^2 \leq 2\alpha(f(c) - f(x)) + \alpha f(x) < 0.$$

The CSP method is a variant of the SOP method, in which P_{C_i} is replaced with Π_{C_i}.

24.7 Projecting onto the Intersection of Convex Sets

As we saw previously, the SOP algorithm need not converge to the point in the intersection closest to the starting point. To obtain the point closest to x^0 in the intersection of the convex sets C_i, we can use *Dykstra's algorithm*, a modification of the SOP method [92]. For simplicity, we shall discuss only the case of $C = A \cap B$, the intersection of two closed, convex sets.

24.7.1 A Motivating Lemma

The following lemma will help to motivate Dykstra's algorithm.

Lemma 24.1 *If $x = c + p + q$, where $c = P_A(c + p)$ and $c = P_B(c + q)$, then $c = P_C x$.*

Proof: Let d be arbitrary in C. Then

$$\langle c - (c + p), d - c\rangle \geq 0,$$

since d is in A, and

$$\langle c - (c + q), d - c\rangle \geq 0,$$

since d is in B. Adding the two inequalities, we get

$$\langle -p - q, d - c\rangle \geq 0.$$

But

$$-p - q = c - x,$$

so

$$\langle c - x, d - c\rangle \geq 0,$$

for all d in C. Therefore, $c = P_C x$. ■

24.7.2 Dykstra's Algorithm

Dykstra's algorithm begins with $b_0 = x$, $p_0 = q_0 = 0$. It involves the construction of two sequences, $\{a_n\}$ and $\{b_n\}$, both converging to $c = P_C x$, along with two other sequences, $\{p_n\}$ and $\{q_n\}$ designed so that

$$a_n = P_A(b_{n-1} + p_{n-1}),$$

$$b_n = P_B(a_n + q_{n-1}),$$

and

$$x = a_n + p_n + q_{n-1} = b_n + p_n + q_n.$$

Both $\{a_n\}$ and $\{b_n\}$ converge to $c = P_C x$. Usually, but not always, $\{p_n\}$ converges to p and $\{q_n\}$ converges to q, so that

$$x = c + p + q,$$

with

$$c = P_A(c + p) = P_B(c + q).$$

Generally, however, $\{p_n + q_n\}$ converges to $x - c$.

In [24], Bregman considers the problem of minimizing a convex function $f : R^J \to R$ over the intersection of half-spaces, that is, over the set of points x for which $Ax =\geq b$. His approach is a *primal-dual* algorithm involving the notion of projecting onto a convex set, with respect to a generalized distance constructed from f. Such generalized projections have come to be called *Bregman projections*. In [65], Censor and Reich extend Dykstra's algorithm to Bregman projections, and, in [25], the three show that the extended Dykstra algorithm of [65] is the natural extension of Bregman's primal-dual algorithm to the case of intersecting convex sets. We shall consider these results in more detail in a subsequent chapter.

24.7.3 The Halpern-Lions-Wittmann-Bauschke Algorithm

There is yet another approach to finding the orthogonal projection of the vector x onto the nonempty intersection C of finitely many closed, convex sets C_i, $i = 1, ..., I$. The algorithm has the following iterative step:

$$x^{k+1} = t_k x + (1 - t_k) P_{C_i} x^k,$$

where P_{C_i} denotes the orthogonal projection onto C_i, t_k is in the interval $(0, 1)$, and $i = k (\mathrm{mod}\, I) + 1$. Several authors have proved convergence of the sequence $\{x^k\}$ to $P_C x$, with various conditions imposed on the parameters $\{t_k\}$. As a result, the algorithm is known as the Halpern-Lions-Wittmann-Bauschke (HLWB) algorithm, after the names of several who have contributed to the evolution of the theorem. The conditions imposed by Bauschke [9] are $\{t_k\} \to 0$, $\sum t_k = \infty$, and $\sum |t_k - t_{k+I}| < +\infty$. The HLWB algorithm has been extended by Deutsch and Yamada [88] to minimize certain (possibly non-quadratic) functions over the intersection of fixed point sets of operators more general than P_{C_i}.

Chapter 25

Sensitivity to Noise

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, [56]).

25.1 Where Does Sensitivity Come From?

We illustrate the sensitivity problem that can arise when the inconsistent system $Ax = b$ has more equations than unknowns. Let A be I by J. We calculate the least-squares solution,

$$x_{LS} = (A^\dagger A)^{-1}A^\dagger b,$$

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

$$Q = \lambda_1 u_1 u_1^\dagger + \cdots + \lambda_J u_J u_J^\dagger,$$

where the (necessarily positive) eigenvalues of Q are

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_J > 0,$$

and the vectors u_j are the corresponding orthonormal eigenvectors.

25.1.1 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

$$A = \sqrt{\lambda_1}u_1v_1^\dagger + \cdots + \sqrt{\lambda_J}u_Jv_J^\dagger,$$

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 $u_jv_j^\dagger$.

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.

25.1.2 The Inverse of $Q = A^\dagger A$

The inverse of Q can then be written

$$Q^{-1} = \lambda_1^{-1}u_1u_1^\dagger + \cdots + \lambda_J^{-1}u_Ju_J^\dagger,$$

so that, with $A^\dagger b = c$, we have

$$x_{LS} = \lambda_1^{-1}(u_1^\dagger c)u_1 + \cdots + \lambda_J^{-1}(u_J^\dagger c)u_J.$$

Because the eigenvectors are orthonormal, we can express $||A^\dagger b||_2^2 = ||c||_2^2$ as

$$||c||_2^2 = |u_1^\dagger c|^2 + \cdots + |u_J^\dagger c|^2,$$

and $||x_{LS}||_2^2$ as

$$||x_{LS}||_2^2 = \lambda_1^{-1}|u_1^\dagger c|^2 + \cdots + \lambda_J^{-1}|u_J^\dagger c|^2.$$

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.

25.1.3 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

$$F_\epsilon(x) = (1-\epsilon)\|Ax - b\|_2^2 + \epsilon\|x - p\|_2^2. \tag{25.1}$$

Exercise 25.1 *Show that F_ϵ always has a unique minimizer $\hat{x}_\epsilon$, given by*

$$\hat{x}_\epsilon = ((1-\epsilon)A^\dagger A + \epsilon I)^{-1}((1-\epsilon)A^\dagger b + \epsilon p);$$

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

$$\hat{x}_\epsilon = (A^\dagger A + \gamma^2 I)^{-1} A^\dagger b, \tag{25.2}$$

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.

Exercise 25.2 *Let ϵ be in $(0,1)$, and let I be the identity matrix whose dimensions are understood from the context. Show that*

$$((1-\epsilon)AA^\dagger + \epsilon I)^{-1}A = A((1-\epsilon)A^\dagger A + \epsilon I)^{-1},$$

and, taking conjugate transposes,

$$A^\dagger((1-\epsilon)AA^\dagger + \epsilon I)^{-1} = ((1-\epsilon)A^\dagger A + \epsilon I)^{-1}A^\dagger.$$

Hint: use the identity

$$A((1-\epsilon)A^\dagger A + \epsilon I) = ((1-\epsilon)AA^\dagger + \epsilon I)A.$$

Exercise 25.3 *Show that 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.

Exercise 25.4 *Show that, 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

$$\hat{x}_\epsilon = A^\dagger((1-\epsilon)AA^\dagger + \epsilon I)^{-1}((1-\epsilon)b + \epsilon q) + ((1-\epsilon)A^\dagger A + \epsilon I)^{-1}(\epsilon r).$$

Exercise 25.5 (a) *Show that*

$$((1-\epsilon)A^\dagger A + \epsilon I)^{-1}(\epsilon r) = r,$$

for all $\epsilon \in (0,1)$. **(b)** *Now take the limit of $\hat{x}_\epsilon$, as $\epsilon \to 0$, to get $\hat{x}_\epsilon \to A^\dagger(AA^\dagger)^{-1}b + r$. Show that this is the solution of $Ax = b$ closest to p. Hints: For part (a) let*

$$t_\epsilon = ((1-\epsilon)A^\dagger A + \epsilon I)^{-1}(\epsilon r).$$

Then, multiplying by A gives

$$At_\epsilon = A((1-\epsilon)A^\dagger A + \epsilon I)^{-1}(\epsilon r).$$

Now show that $At_\epsilon = 0$. For part (b) draw a diagram for the case of one equation in two unknowns.

25.2 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 underdetermined 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 (25.1). For the case of $p = 0$, the solution to this problem is the vector $\hat{x}_\epsilon$ in Equation (25.2). However, we do not want to calculate $A^\dagger A + \gamma^2 I$, in order to solve

$$(A^\dagger A + \gamma^2 I)x = A^\dagger b,$$

when the matrix A is large. Fortunately, there are ways to find $\hat{x}_\epsilon$, using only the matrix A and the Landweber or ART algorithms.

25.2.1 Iterative Regularization with Landweber's Algorithm

Our goal is to minimize the function in Equation (25.1). Notice that this function can be written as

$$F_\epsilon(x) = ||Bx - c||_2^2,$$

for

$$B = \begin{bmatrix} A \\ \gamma^2 I \end{bmatrix},$$

and

$$c = \begin{bmatrix} b \\ 0 \end{bmatrix},$$

where 0 denotes a column vector with all entries equal to zero. The Landweber iteration for the problem $Bx = c$ is

$$x^{k+1} = x^k + \alpha B^T(c - Bx^k), \tag{25.3}$$

for $0 < \alpha < \rho(B^T B)$, where $\rho(B^T B)$ is the spectral radius of $B^T B$. Equation (25.3) can be written as

$$x^{k+1} = (1 - \alpha\gamma^2)x^k + \alpha A^T(b - Ax^k). \tag{25.4}$$

We see from Equation (25.4) that the Landweber algorithm for solving the regularized least squares problem amounts to a relaxed version of the Landweber algorithm applied to the original least squares problem.

25.2.2 Iterative Regularization with ART

We discuss two methods for using ART to obtain regularized solutions of $Ax = b$. The first one is presented in [56], while the second one is due to Eggermont, Herman, and Lent [93].

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

$$[\, A^\dagger \quad \gamma I \,] \begin{bmatrix} u \\ v \end{bmatrix} = 0.$$

We begin with $u^0 = b$ and $v^0 = 0$.

Exercise 25.6 *Show that the lower component of the limit vector is* $v^\infty = -\gamma \hat{x}_\epsilon$.

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

$$[\, A \quad \gamma I \,] \begin{bmatrix} x \\ v \end{bmatrix} = b.$$

We begin at $x^0 = 0$ and $v^0 = 0$.

Exercise 25.7 *Show that the limit vector has for its upper component* $x^\infty = \hat{x}_\epsilon$ *as before, and that* $\gamma v^\infty = b - A\hat{x}_\epsilon$.

Chapter 26

The EMML and SMART Algorithms

How we develop algorithms for tomographic image reconstruction depends, to some extent, on how we view the problem. The filtered backprojection (FBP) approach to tomographic image reconstruction is based on a continuous model and the idea that the data are line integrals. The Central Slice Theorem relates these line integrals to the two-dimensional Fourier transform of the function we seek. Reconstruction algorithms are then methods for performing (approximately, in practice) the Fourier-transform inversion. The resulting FBP methods are non-iterative.

When the problem is discretized, it naturally becomes one of solving a large, noisy system of linear equations, subject to constraints, such as non-negativity. This is the approach that led to the iterative ART, MART, simultaneous MART (SMART), and related methods.

A different point of view emerges by focusing on the statistical nature of the data, and treating the unknowns as parameters to be estimated. Likelihood maximization is the natural choice for determining the unknown parameters and reconstruction algorithms become methods for performing this maximization. This approach led to the EMML algorithm.

In the (typical) case of noisy data, the reconstructions provided by the EMML and SMART methods are usually inadequate. Both methods attempt to match the measured data to a theoretical model, and, for noisy data, too close a match can need to poor reconstructions. In such cases, the objectives of the EMML and SMART algorithms must be augmented to reduce sensitivity to noise; this ia called *regularization*.

Although the EMML and SMART algorithms have quite different histories and are not typically considered together they are closely related [42, 43]. In this chapter we examine these two algorithms in tandem, following [44]. The method of *alternating minimization* that we use here is important in its own right, but fundamental for understanding a number of regularization methods, such as De Pierro's *surrogate-function* approach. 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 [42] was based on the analogous proof for the EMML [187], while discovery of the faster version of the EMML, the *rescaled block-iterative* EMML (RBI-EMML) [45] came from studying the analogous block-iterative version of SMART [66]. The proofs we give here are elementary and rely mainly on easily established properties of the cross-entropy or Kullback-Leibler distance.

26.1 The SMART and the EMML method

In the stochastic model used in single-photon emission tomography (SPECT), the data are b_i, $i = 1, ..., I$, the number of photons detected at each of the I detectors. These quantities are viewed as realizations of independent Poisson random variables having expected values $(Ax)_i = \sum_{j=1}^{J} A_{ij}x_j$. The x_j are the unknown intensities at each of the j pixels, and are the quantities we seek to estimate. The A_{ij} are the probabilities that a photon emitted at pixel j will be detected at detector i. The likelihood function to be maximized is then

$$L(x) = \prod_{i=1}^{I} e^{-(Ax)_i}(Ax)^{b_i}/b_i!. \tag{26.1}$$

Taking logs, we get the log likelihood function

$$LL(x) = \sum_{i=1}^{I} b_i \log(Ax)_i - (Ax)_i - \log(b_i!). \tag{26.2}$$

Maximizing $LL(x)$ is equivalent to minimizing

$$\sum_{i=1}^{I} b_i \log b_i - b_i \log(Ax)_i + (Ax)_i - b_i,$$

which, as we shall see shortly, is the Kullback-Leibler distance between the vectors $b = (b_1, ..., b_I)^T$ and $Ax = ((Ax)_1, ..., (Ax)_I)^T$. Maximizing the likelihood over $x \geq 0$ is equivalent to finding an $x \geq 0$ that makes Ax as close as possible to the data vector b. When the data are noisy, this may not be a good thing to do.

Both the SMART and the EMML method provide a solution of $b = Ax$ when such exist and (distinct) approximate solutions in the inconsistent case. Both begin with an arbitrary positive vector x^0. Having found x^k the iterative step for the SMART is

SMART:

$$x_j^{k+1} = x_j^k \exp\Big(s_j^{-1} \sum_{i=1}^{I} A_{ij} \log \frac{b_i}{(Ax^k)_i}\Big) \tag{26.3}$$

while that for the EMML method is

EMML:

$$x_j^{k+1} = x_j^k s_j^{-1} \sum_{i=1}^{I} A_{ij} \frac{b_i}{(Ax^k)_i}. \tag{26.4}$$

The main results concerning the SMART is given by the following theorem.

Theorem 26.1 *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 26.2 *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 elsewhere.

26.2 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 backprojection, 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 [112]. 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) [82, 175] is a variant of MART that uses all the data at each step of the iteration.

26.3 The Kullback-Leibler Distance

For $a > 0$ and $b > 0$, we define

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

$$KL(a,0) = +\infty,$$

$$KL(0,b) = b,$$

and

$$KL(0,0) = 0.$$

The Kullback-Leibler distance $KL(x,z)$ is defined for nonnegative vectors x and z by

$$KL(x,z) = \sum_{n=1}^{N} KL(x_n, z_n).$$

Clearly, the KL distance has the property $KL(cx,cz) = cKL(x,z)$ for all positive scalars c.

Exercise 26.1 *Let* $z_+ = \sum_{j=1}^{J} z_j > 0$. *Then*

$$KL(x,z) = KL(x_+, z_+) + KL(x, (x_+/z_+)z). \tag{26.5}$$

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

26.4 The Alternating Minimization Paradigm

Let A be an I by J matrix with entries $A_{ij} \geq 0$, such that, for each $j = 1, ..., J$, we have $s_j = \sum_{i=1}^{I} A_{ij} > 0$. Let $b = (b_1, ..., b_I)^T$ with $b_i > 0$ 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 A_{ij} with A_{ij}/s_j; the quantities $(Ax)_i$ are unchanged.

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

$$r(x)_{ij} = x_j A_{ij} \frac{b_i}{(Ax)_i}$$

and

$$q(x)_{ij} = x_j A_{ij}.$$

The KL distances

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

and

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

will play important roles in the discussion that follows. Note that if there is nonnegative x with $r(x) = q(x)$ then $b = Ax$.

26.4.1 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 $KL(r(z), q(x))$ and $KL(q(x), r(z))$ are minimized, first with respect to the variable x and then with respect to the variable 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 26.2 *Establish the following Pythagorean identities:*

$$KL(r(x),q(z)) = KL(r(z),q(z)) + KL(r(x),r(z)); \tag{26.6}$$

$$KL(r(x),q(z)) = KL(r(x),q(x')) + KL(x',z), \tag{26.7}$$

for

$$x'_j = x_j \sum_{i=1}^{I} A_{ij} \frac{b_i}{(Ax)_i}; \tag{26.8}$$

$$KL(q(x),r(z)) = KL(q(x),r(x)) + KL(x,z) - KL(Ax,Az); \tag{26.9}$$

$$KL(q(x),r(z)) = KL(q(z''),r(z)) + KL(x,z''), \tag{26.10}$$

for

$$z''_j = z_j \exp(\sum_{i=1}^{I} A_{ij} \log \frac{b_i}{(Az)_i}). \tag{26.11}$$

Note that it follows from Equation (26.5) that $KL(\mathbf{x},z) - KL(Ax,Az) \geq 0$.

26.4.2 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 $b = Ax$ has nonnegative solutions, both algorithms produce such a solution. In general, the EMML gives a nonnegative minimizer of $KL(b,Ax)$, while the SMART minimizes $KL(Ax,b)$ over nonnegative x.

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

$$x_j^{k+1} = (x^k)'_j = x_j^k \sum_{i=1}^{I} A_{ij} \frac{b_i}{(Ax^k)_i}. \tag{26.12}$$

The iterative step for the SMART is

$$x_j^{m+1} = (x^m)''_j = x_j^m \exp\Big(\sum_{i=1}^{I} A_{ij} \log \frac{b_i}{(Ax^m)_i}\Big). \tag{26.13}$$

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

Exercise 26.3 *Show that, for $\{x^k\}$ given by Equation (26.12), $\{KL(\mathbf{y}, Ax^k)\}$ is decreasing and $\{KL(x^{k+1}, x^k)\} \to 0$. Show that, for $\{x^m\}$ given by Equation (26.13), $\{KL(Ax^m, b)\}$ is decreasing and $\{KL(x^m, x^{m+1})\} \to 0$.*

Hint: Use $KL(r(x), q(x)) = KL(b, Ax)$, $KL(q(x), r(x)) = KL(Ax, b)$, and the Pythagorean identities.

Exercise 26.4 *Show that the EMML sequence $\{x^k\}$ is bounded by showing*

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

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

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

Exercise 26.5 *Show that $(x^*)' = x^*$ for any cluster point x^* of the EMML sequence $\{x^k\}$ and that $(x^*)'' = x^*$ for any cluster point x^* of the SMART sequence $\{x^m\}$.*

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

Exercise 26.6 *Let $\hat{x}$ and $\tilde{x}$ minimize $KL(b, Ax)$ and $KL(Ax, b)$, respectively, over all $x \geq \mathbf{0}$. Then, $(\hat{x})' = \hat{x}$ and $(\tilde{x})'' = \tilde{\mathbf{x}}$.*

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

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

Exercise 26.7 *For the EMML sequence $\{x^k\}$ with cluster point x^* and $\hat{x}$ as defined previously, we have the double inequality*

$$KL(\hat{x}, x^k) \geq KL(r(\hat{x}), r(x^k)) \geq KL(\hat{x}, x^{k+1}), \tag{26.14}$$

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

Hint: For the first inequality calculate $KL(r(\hat{x}), q(x^k))$ in two ways. For the second one, use $(\mathbf{x})'_j = \sum_{i=1}^{I} r(x)_{ij}$ and Exercise 26.1.

Exercise 26.8 *Show that, for the SMART sequence* $\{x^m\}$ *with cluster point* x^* *and* $\tilde{x}$ *as defined previously, we have*

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

$$KL(P\tilde{x}, Ax^m) + KL(x^{m+1}, x^m) - KL(Ax^{m+1}, Ax^m), \qquad (26.15)$$

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

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

Exercise 26.9 *For* x^* *a cluster point of the EMML sequence* $\{x^k\}$ *we have* $KL(b, Ax^*) = KL(b, P\hat{x})$. *Therefore,* x^* *is a nonnegative minimizer of* $KL(b, Ax)$. *Consequently, the sequence* $\{KL(x^*, x^k)\}$ *converges to zero, and so* $\{\mathbf{x}^k\} \to x^*$.

Hint: Use the double inequality of Equation (26.14) and $KL(r(\hat{x}), q(x^*))$.

Exercise 26.10 *For* x^* *a cluster point of the SMART sequence* $\{x^m\}$ *we have* $KL(Ax^*, b) = KL(P\tilde{x}, b)$. *Therefore,* x^* *is a nonnegative minimizer of* $KL(Ax, b)$. *Consequently, the sequence* $\{KL(x^*, x^m)\}$ *converges to zero, and so* $\{x^m\} \to x^*$. *Moreover,*

$$KL(\tilde{x}, x^0) \geq KL(x^*, x^0)$$

for all $\tilde{x}$ *as before.*

Hints: Use Exercise 26.8. For the final assertion use the fact that the difference $KL(\tilde{x}, \mathbf{x}^m) - KL(\tilde{x}, x^{m+1})$ is independent of the choice of $\tilde{x}$, since it depends only on $Ax^* = P\tilde{x}$. 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.

26.5 Bayesian Regularization

As we noted previously, when the data are noisy, both the EMML and SMART typically lead to unacceptable images. One way to remedy this problem is simply to halt the algorithm after a few iterations, to avoid over-fitting the x to the noisy data. A more mathematically sophisticated remedy is to add a penalty function to the function being minimized. The penalty function can usually be related to a prior probability distribution on the vector x, so these methods are often called Bayesian methods. In Bayesian methods we 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

$$F(x) = KL(b, Ax) - \log f(x). \tag{26.16}$$

26.6 Penalized EMML

The EMML algorithm minimizes the function $KL(b, Ax)$, over $x \geq 0$. A penalized EMML algorithm with penalty function $g(x) \geq 0$ will minimize $F(x) = KL(b, Ax) + g(x)$. Each step of the EMML algorithm arises by minimizing $KL(r(x^k), q(x))$, as a function of x, to get x^{k+1}. Suppose we attempt to find an iterative algorithm to minimize $F(x)$ by minimizing $KL(r(x^k), q(x)) + g(x)$ to get x^{k+1}. Setting the partial derivative of this function with respect to x_j equal to zero gives

$$x_j^{k+1} = [x_j^k \sum_{i=1}^{I} A_{ij} \frac{b_i}{(Ax^k)_i}]/[1 + \frac{\partial g}{\partial x_j}(x^{k+1})]. \tag{26.17}$$

Obviously, this poses a problem; we have not succeeded in isolating x_j^{k+1} on the left side. In [113] Green suggests replacing x^{k+1} with x^k on the right side; this approach is called the *one-step-late*(OSL) algorithm. Then, we can solve for x_j^{k+1} in closed form. 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 [**?**] an interior point algorithm (IPA) is presented that avoids the OSL issue. In [153] the IPA is used to regularize transmission tomographic images.

A different approach is to select $g(x)$ more carefully, to insure that we can solve for x_j^{k+1} at each step.

26.6.1 Using a Norm Constraint

For example, suppose that

$$g(x) = \frac{1}{2}||x||^2 = \frac{1}{2}\sum_{j=1}^{J} x_j^2.$$

Then

$$\frac{\partial g}{\partial x_j}(x^{k+1}) = x_j^{k+1},$$

and we have

$$x_j^{k+1}[1 + x_j^{k+1}] = x_j^k \sum_{i=1}^{I} A_{ij} \frac{b_i}{(Ax^k)_i},$$

which is a quadratic equation in the unknown x_j^{k+1}.

26.6.2 The Gamma Prior Distribution for x

In [140] 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

$$\gamma(x) = \frac{1}{\Gamma(\alpha)}(\frac{\alpha}{\beta})^{\alpha} x^{\alpha-1} e^{-\alpha x/\beta},$$

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 $\sigma^2 = \beta^2/\alpha$.

Exercise 26.11 *Show that 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 (8.5) with respect to z is equivalent to minimizing the function*

$$KL(r(x^k), q(z)) + \sum_{j=1}^{J} \delta_j KL(\gamma_j, z_j), \tag{26.18}$$

for

$$\delta_j = \frac{\mu_j}{\sigma_j^2}, \; \gamma_j = \frac{\mu_j^2 - \sigma_j^2}{\mu_j},$$

under the assumption that the latter term is positive. Show further that the resulting x^{k+1} *has entries given in closed form by*

$$x_j^{k+1} = \frac{\delta_j}{\delta_j + s_j}\gamma_j + \frac{1}{\delta_j + s_j} x_j^k \sum_{i=1}^{I} A_{ij} b_i/(Ax^k)_i, \tag{26.19}$$

where $s_j = \sum_{i=1}^{I} A_{ij}$.

We see from Equation (26.19) 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 in [140]; see also [42].

More simply, suppose that $g(x) = KL(p, x)$, where $p \geq 0$ is a prior estimate of the solution we seek. Then

$$\frac{\partial g}{\partial x_j}(x^{k+1}) = -\frac{p_j}{x_j^{k+1}} + 1,$$

and assuming again that $s_j = 1$ for all j, we have

$$x_j^{k+1} = \frac{1}{2}[x_j^k \sum_{i=1}^{I} A_{ij} \frac{b_i}{(Ax^k)_i}] + \frac{1}{2} p_j.$$

26.7 Penalized SMART

In order to minimize $KL(Ax, b) + g(x)$ using alternating minimization, we want to select $g(x) \geq 0$ so that minimizing $KL(q(x), r(x^k)) + g(x)$, with respect to x, to get x_j^{k+1} is easy. By analogy with the gamma-prior method for EMML, we try $g(x) = KL(x, p)$. The resulting algorithm has the iterative step

$$x_j^{k+1} = [x_j^k \exp(\sum_{i=1}^{I} A_{ij} \log \frac{b_i}{(Ax^k)_i})]^{1/2} (p_j)^{1/2}. \tag{26.20}$$

It was shown in [42] that this algorithm converges to a non-negative minimizer of $KL(Ax, b)) + Kl(x, p)$.

26.8 The Surrogate-Function Approach

The EMML and SMART algorithms are examples of an optimization method based on alternating minimization of a function $H(x, z) > 0$ of two vector variables. For the EMML, we have

$$H(x, z) = KL(r(z), q(x)),$$

while for the SMART we have

$$H(x,z) = KL(q(x), r(z)).$$

In both cases, holding $z = x^k$ fixed and minimizing with respect to x gives the next iterate, x^{k+1}; holding $x = x^{k+1}$ fixed and minimizing with respect to z leads to $z = x^{k+1}$ again. For the EMML we have $H(x,x) = KL(b, Ax)$, while for the SMART we have $H(x,x) = KL(Ax, b)$. In both cases we have

$$H(x^k, x^k) \geq H(x^{k+1}, x^k) \geq H(x^{k+1}, x^{k+1}),$$

which tells us that both algorithms are reducing the functions they seek to minimize. De Pierro's approach to regularization [85] is to find a new function $H(x,z)$ that includes the penalty function, while behaving like the two choices for $H(x,z)$ just discussed. Because his *surrogate function* method has been used subsequently by others to obtain penalized likelihood algorithms [68], we consider his approach in some detail. For clarity, we consider the penalized EMML case.

Let x and z be vector variables and $H(x,z) > 0$. Mimicking the behavior of the functions $H(x,z)$ used in EMML and SMART, we require that if we fix x and minimize $H(x,z)$ with respect to z, the solution should be $x = z$, the vector we fixed; that is, $H(x,z) \geq H(x,x)$ always. If we fix z and minimize $H(x,z)$ with respect to x, 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) $H(x,z) \geq H(x,x)$ 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 x for fixed z is easy.

The function to be minimized is

$$F(x) = KL(b, Ax) + g(x),$$

where $g(x) \geq 0$ is some penalty function. De Pierro uses penalty functions $g(x)$ of the form

$$g(x) = \sum_{l=1}^{p} f_l(\langle s_l, x\rangle\ ).$$

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,

$$g(x) = \sum_{l=1}^{p} f_l((Sx)_l).$$

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

$$f_l((Sx)_l) = f_l(\sum_{j=1}^{J} S_{lj}x_j) = f_l(\sum_{j=1}^{J} \lambda_{lj}(S_{lj}/\lambda_{lj})x_j)$$

$$\leq \sum_{j=1}^{J} \lambda_{lj} f_l((S_{lj}/\lambda_{lj})x_j).$$

Therefore,

$$g(x) \leq \sum_{l=1}^{p}\sum_{j=1}^{J} \lambda_{lj} f_l((S_{lj}/\lambda_{lj})x_j).$$

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

$$\sum_{l=1}^{p}\sum_{j=1}^{J} \lambda_{lj} f_l((S_{lj}/\lambda_{lj})z_j).$$

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

$$f_l((S_{lj}/\lambda_{lj})z_j - (S_{lj}/\lambda_{lj})x_j + (Sx)_l).$$

So, De Pierro's function $H(x,z)$ is the sum of the $H(x,z)$ used in the EMML case and the function

$$\sum_{l=1}^{p}\sum_{j=1}^{J} \lambda_{lj} f_l((S_{lj}/\lambda_{lj})z_j - (S_{lj}/\lambda_{lj})x_j + (Sx)_l).$$

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).

26.9 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

$$KL(Ax, b) + \epsilon KL(x, p)?$$

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

$$KL(b, Ax) + \epsilon KL(p, x)?$$

There have been various attempts to include regularization in block-iterative methods, to reduce noise sensitivity and avoid limit cycles, but all 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 (12.18) 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 [87] RAMLA is extended to a regularized version, but with no guarantee of convergence.

Chapter 27

Iterative Algorithms: An Overview

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 the 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, in practice sometimes only a few iterative steps are used.

27.1 Algorithms and Operators

For most of the iterative algorithms we shall consider, the iterative step is

$$x^{k+1} = Tx^k,$$

for some operator T. The behavior of the algorithm will then depend on the properties of the 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.

27.1.1 Steepest Descent Minimization

Suppose that we want to minimize a real-valued function $f : R^J \to R$. At each x the direction of greatest decrease of f is the negative of the gradient, $-\nabla f(x)$. The *steepest descent* method has the iterative step

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k),$$

where, ideally, the *step-length parameter* α_k would be chosen so as to minimize $f(x)$ in the chosen direction, that is, the choice of $\alpha = \alpha_k$ would minimize

$$f(x^k - \alpha \nabla f(x^k)).$$

In practice, it is difficult, if not impossible, to determine the optimal value of α_k at each step. Therefore, a line search is usually performed to find a suitable α_k, meaning that values of $f(x^k - \alpha \nabla f(x^k))$ are calculated, for some finite number of α values, to determine a suitable choice for α_k.

For practical reasons, we are often interested in iterative algorithms that avoid line searches. Some of the minimization algorithms we shall study take the form

$$x^{k+1} = x^k - \alpha \nabla f(x^k),$$

where the α is a constant, selected at the beginning of the iteration. Such iterative algorithms have the form $x^{k+1} = Tx^k$, for T the operator defined by

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

When properly chosen, the α will not be the optimal step-length parameter for every step of the iteration, but will be sufficient to guarantee convergence. In addition, the resulting iterative sequence is often *monotonically decreasing*, which means that

$$f(x^{k+1}) < f(x^k),$$

for each k.

We shall discuss other iterative monotone methods, such as the EMML and SMART algorithms, that can be viewed as generalized steepest descent methods taking the form

$$x_j^{k+1} = x_j^k - \alpha_{k,j} \nabla f(x^k)_j.$$

In these cases, the step-length parameter α_k is replaced by ones that also vary with the entry index j. While this may seem even more complicated to implement, for the algorithms mentioned, these $\alpha_{k,j}$ are automatically calculated as part of the algorithm, with no line searches involved.

27.1.2 Selecting the Operator

Although any iterative algorithm involves a transformation of the current vector x^k into the next one, x^{k+1}, it may be difficult, if not impossible, and perhaps useless, to represent that transformation simply as $x^{k+1} = Tx^k$. The transformation that occurs in the bisection method for root-finding is not naturally represented using an operator T. Nevertheless, many algorithms do take the form $x^{k+1} = Tx^k$, as we shall see, and investigating the properties of such operators is an important part of the study of iterative algorithms.

27.2 Operators on Finite-Dimensional Space

Much of our attention will be devoted to operators on the finite-dimensional spaces R^J, the space of real J-vectors, and C^J, the space of complex J-vectors; we call the space $\mathcal{X}$ when what we are saying applies to either of these spaces. The notation $||x||_2$ will denote the *Euclidean norm* or *Euclidean length* of the vector x, given by

$$||x||_2 = \sqrt{\sum_{j=1}^{J} |x_j|^2}.$$

The *Euclidean distance* between vectors x and y is $||x-y||_2$. There are other norms that we shall use, but the Euclidean norm is the most convenient for our purposes.

An operator T on $\mathcal{X}$ can be written in terms of its scalar-valued component functions $T_j(x)$ as

$$Tx = (T_1(x), ..., T_J(x))^T.$$

We say that T is continuous if each of the functions T_j is continuous, as a scalar-valued function on $\mathcal{X}$. Continuity of T, by itself, will not guarantee the convergence of the iterative scheme $x^{k+1} = Tx^k$, even when Fix(T), the set of fixed points of T, is non-empty.

27.2.1 Lipschitz Continuity

An operator T on $\mathcal{X}$ is *Lipschitz continuous*, with respect to a norm $|| \cdot ||$, if there is a positive constant λ such that

$$||Tx - Ty|| \leq \lambda ||x - y||,$$

for all x and y in $\mathcal{X}$.

27.2.2 Non-Expansive Operators

We shall focus on operators T that are *non-expansive* (ne), with respect to some norm $||\cdot||$, which means that, for all vectors x and y in $\mathcal{X}$,

$$||Tx - Ty|| \leq ||x - y||.$$

Clearly, any ne operator is Lipschitz continuous, for $\lambda = 1$. Even being ne is not enough for convergence, as the example $T = -I$, I the identity operator, shows.

27.2.3 Strict Contractions

To guarantee convergence of $\{T^k x^0\}$ to a fixed point of T, it is sufficient to assume that T is a *strict contraction* (sc), with respect to some norm $||\cdot||$, which means that there is r in the interval $(0, 1)$ such that, for all x and y in $\mathcal{X}$,

$$||Tx - Ty|| \leq r||x - y||.$$

As we shall see later, if T is sc, then T has a unique fixed point, say $\hat{x}$, and the sequence $\{T^k x^0\}$ converges to $\hat{x}$, for every starting vector x^0. But being a strict contraction is too strong for our purposes.

27.2.4 Averaged Operators

Many of the operators we need to study have multiple fixed points. For example, the orthogonal projection onto a hyperplane in $\mathcal{X}$ has the entire hyperplane for its fixed-point set. We need a class of operators between the ne operators and the sc ones. The Krasnoselskii-Mann (KM) Theorem shows us how to select this class:

Theorem 27.1 *Let $T = (1 - \alpha)I + \alpha N$, for some α in the interval $(0, 1)$ and operator N that is ne, with respect to $||\cdot||_2$. Then the sequence $\{T^k x^0\}$ converges to a fixed point of T, whenever Fix(T) is non-empty.*

This theorem suggests that the appropriate class is that of the *averaged* (av) operators, that is, those T as described in the KM Theorem. The class of averaged operators is quite broad, and includes many of the operators we need to study, in the Euclidean case. Products of av operators are av operators, which is quite helpful in designing algorithms for constrained optimization.

Note that we could have defined av operators more generally, by requiring that N be ne, with respect to some norm, not necessarily the Euclidean norm. For any operator T on $\mathcal{X}$, we have the following identity relating T to its complement operator, $G = I - T$:

$$||x - y||_2^2 - ||Tx - Ty||_2^2 = 2Re(\langle Gx - Gy, x - y\rangle) - ||Gx - Gy||_2^2. \quad (27.1)$$

This identity, which allows us to transform properties of T into properties of G that might be easier to work with, and which is valid only for the Euclidean norm, is a key ingredient in the proof of the KM Theorem.

We encounter averaged operators in two different ways. In the first way, we are interested in a particular operator N that is ne, with respect to the Euclidean norm, has fixed points, and we wish to calculate one of them. We know that the iteration $x^{k+1} = Nx^k$ may not converge. Therefore, we select some α in $(0, 1)$ and use instead the iteration $x^{k+1} = Tx^k$, for $T = (1 - \alpha)I + \alpha N$. The fixed points of T are those of N and convergence is guaranteed by the KM Theorem. In the second way, we have some operator T and we want to know if T is av, so that we can use the KM Theorem. Deciding if a given operator is av is not always easy, and it is sometimes more convenient to consider the corresponding properties of the complement operator, $G = I - T$.

27.2.5 Affine Linear and Linear Operators

An operator B is *linear* if, for all scalars α and β and vectors x and y,

$$B(\alpha x + \beta y) = \alpha Bx + \beta By.$$

An operator T is *affine linear*, or just *affine*, if there is a linear operator B and a vector d, such that, for all vectors x,

$$Tx = Bx + d.$$

We can see that an affine linear operator T will be ne, sc, or av precisely when its linear component, B, is ne, sc, or av, respectively.

A linear operator B, which we shall view as multiplication by the matrix B, is said to be *Hermitian* if $B = B^\dagger$; this means that $B^\dagger$, the conjugate transpose of B, is equal to B. The eigenvalues of such linear operators are real and we have the following: B is ne, with respect to the Euclidean norm, if and only if all its eigenvalues lie in the interval $[-1, 1]$; B is av if and only if all its eigenvalues lie in the interval $(-1, 1]$; and B is sc, with respect to the Euclidean norm, if and only if all its eigenvalues lie in the interval $(-1, 1)$.

When B is not Hermitian, we cannot determine if B is av from its eigenvalues, which need not be real. An alternative approach is to ask if B, and therefore T, is a paracontraction for some vector norm, as discussed below.

27.2.6 Projection Operators

Several of the problems of interest to us here involve finding a vector that satisfies certain constraints, such as optimizing a function, or lying within certain convex sets. If C is a closed, non-empty convex set in $\mathcal{X}$, and x is any vector, then there is a unique point P_Cx in C closest to x. 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_Cx in terms of x; for general convex sets C, however, we will not be able to express P_Cx explicitly.

As we shall see, the orthogonal projection operators $T = P_C$ are *firmly non-expansive* (fne) operators. The fne operators, which are defined by the inequality

$$Re(\langle Tx - Ty, x - y\rangle) \geq ||Tx - Ty||_2^2,$$

form a class of operators within the class of av operators. It follows from Cauchy's Inequality and the fact that P_C is fne that

$$||P_Cx - P_Cy||_2 \leq ||x - y||_2,$$

with equality if and only if

$$P_Cx - P_Cy = \alpha(x - y),$$

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

$$0 \leq Re(\langle P_Cx - P_Cy, x - y\rangle) = \alpha||x - y||_2^2,$$

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

$$P_Cx - x = P_Cy - y.$$

This leads to the definition of *paracontractive* operators.

27.2.7 Paracontractive Operators

A (possibly nonlinear) operator T is said to be a *paracontraction* (pc), or a *paracontractive operator*, with respect to a vector norm $||\cdot||$, if, for every fixed point y of T, and for every x,

$$||Tx - y|| < ||x - y||,$$

or $Tx = x$ [94]. If T has no fixed points, then T is trivially pc. Being pc does not imply being ne. An operator T is said to be *strictly non-expansive* (sne), with respect to some vector norm $||\cdot||$, if

$$||Tx - Ty|| < ||x - y||,$$

or $x - y = Tx - Ty$ [133]. Every T that is sne is pc. We have the following Elsner/Koltracht/Neumann (EKN) convergence theorem from [94]:

Theorem 27.2 *If T is pc with respect to some vector norm, and T has fixed points, then the iterative sequence $\{T^kx^0\}$ converges to a fixed point of T, for every starting vector x^0.*

Unlike av operators, the product of two or more pc operators may not be pc; the product is pc if the operators share at least one fixed point.

27.2.8 Linear and Affine Paracontractions

Say that the linear operator B is *diagonalizable* if $\mathcal{X}$ has a basis of eigenvectors of B. In that case let the columns of V be such an eigenvector basis. Then we have $V^{-1}BV = L$, where L is the diagonal matrix having the eigenvalues of B along its diagonal.

Exercise 27.1 *Show that B is diagonalizable if all its eigenvalues are distinct.*

We see from the exercise that almost all 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

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

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.

Suppose that T is 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. Let $\{u^1, ..., u^J\}$ be linearly independent eigenvectors of B. For each x, we have

$$x = \sum_{j=1}^{J} a_j u^j,$$

for some coefficients a_j. Define

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

Then, T is pc with respect to this norm.

Exercise 27.2 *Show that, if B is a linear av operator, then $|\lambda| < 1$ for all eigenvalues λ of B that are not equal to one.*

We see from the exercise 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.

27.2.9 Operators Related to a Gradient

The *gradient descent* method for minimizing a function $g : R^J \to R$ has the iterative step

$$x^{k+1} = x^k - \gamma_k \nabla g(x^k),$$

where the *step-length parameter* γ_k is adjusted at each step. If we hold $\gamma_k = \gamma$ fixed, then we have $x^{k+1} = Tx^k$, for

$$Tx = x - \gamma \nabla g(x).$$

We shall seek conditions on g and γ under which the operator T is av, which will then lead to iterative algorithms for minimizing g, with convergence a consequence of the KM Theorem.

27.2.10 Constrained Minimization

If our goal is to minimize $g(x)$ over only those x that are in the closed, convex set C, then we may consider a *projected gradient descent* method, having the iterative step

$$x^{k+1} = P_C(x^k - \gamma \nabla g(x^k)).$$

When the operator $Tx = x - \gamma \nabla g(x)$ is av, so is $P_C T$, so the KM Theorem will apply once again.

27.3 Systems of Linear Equations

In remote-sensing problems, including magnetic-resonance imaging, transmission and emission tomography, acoustic and radar array processing, and elsewhere, the data we have measured is related to the object we wish to recover by linear transformation, often involving the Fourier transform. In the vector case, in which the object of interest is discretized, the vector b of measured data is related to the vector x we seek by linear equations that we write as $Ax = b$. The matrix A need not be square, there can be infinitely many solutions, or no solutions at all. We may want to calculate a minimum-norm solution, in the under-determined case, or a least-squares solution, in the over-determined case. The vector x may be the vectorization of a two-dimensional image, in which case I, the number of rows, and J, the number of columns of A, can be in the thousands, precluding the use of non-iterative solution techniques. We may have additional prior knowledge about x, such as its entries are non-negative, which we want to impose as constraints. There is usually noise in measured data, so we may not want an exact solution of $Ax = b$, even if such solutions exist, but prefer a regularized approximate solution. What we need then are iterative algorithms to solve these problems involving linear constraints.

27.3.1 Exact Solutions

When $J \geq I$, the system $Ax = b$ typically has exact solutions, and we want to calculate one of these, we can choose among many iterative algorithms. The *algebraic reconstruction technique* (ART) associates the ith equation in the system with the hyperplane

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

With P_i the orthogonal projection onto H_i, and $i = k(\bmod I) + 1$, the ART iterative step is

$$x^{k+1} = P_i x^k.$$

The operators P_i are av, so the product

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

is also av and convergence of the ART follows from the KM Theorem. The ART is also an optimization method, in the sense that it minimizes $||x - x^0||_2$ over all x with $Ax = b$.

We can also use the operators P_i in a simultaneous manner, taking the iterative step to be

$$x^{k+1} = \frac{1}{I} \sum_{i=1}^{I} P_i x^k.$$

This algorithm is the *Cimmino algorithm* [71]. Once again, convergence follows from the KM Theorem, since the operator

$$T = \frac{1}{I} \sum_{i=1}^{I} P_i$$

is av. Cimmino's algorithm also minimizes $||x - x^0||_2$ over all x with $Ax = b$, but tends to converge more slowly than ART. One advantage Cimmino's algorithm has over the ART is that, in the inconsistent case, in which $Ax = b$ has no solutions, Cimmino's algorithm converges to a least-squares solution of $Ax = b$, while the ART produces a limit cycle of multiple vectors.

Note that $Ax = b$ has solutions precisely when the square system $AA^\dagger z = b$ has a solution; for $J \geq I$, if A has full rank I (which is most of the time) the matrix $AA^\dagger$ will be invertible and the latter system will have a unique solution $z = (AA^\dagger)^{-1}b$. Then $x = A^\dagger z$ is the *minimum-norm solution* of the system $Ax = b$.

If we require a solution of $Ax = b$ that lies in the closed, convex set C, we can modify both the ART and Cimmino's algorithm to achieve this end; all we need to do is to replace x^{k+1} with $P_C x^{k+1}$, the orthogonal projection of x^{k+1} onto C. We call these modified algorithms the *projected ART* and *projected Cimmino algorithm*, respectively. Convergence is again the result of the KM Theorem.

27.3.2 Optimization and Approximate Solutions

When $I > J$ and the system $Ax = b$ has no exact solutions, we can calculate the least-squares solution closest to x^0 using Cimmino's algorithm. When all the rows of A are normalized to have Euclidean length one, the iterative step of Cimmino's algorithm can be written as

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

This is a special case of the *Landweber algorithm*, which has the iterative step

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

Landweber's algorithm converges to the least-squares solution closest to x^0, if the parameter γ is in the interval $(0, 2/L)$, where L is the largest eigenvalue of the matrix $A^\dagger A$. Landweber's algorithm can be written as $x^{k+1} = Tx^k$, for the operator T defined by

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

This operator is affine linear and is an av operator, since its linear part, the matrix $B = I - \gamma A^\dagger A$, is av for any γ in $(0, 2/L)$. Convergence then follows from the KM Theorem. When the rows of A have Euclidean length one, the trace of $AA^\dagger$ is I, the number of rows in A, so $L \leq I$. Therefore, the choice of $\gamma = \frac{1}{I}$ used in Cimmino's algorithm is permissible, but usually much smaller than the optimal choice.

To minimize $||Ax - b||_2$ over x in the closed, convex set C we can use the *projected Landweber algorithm*, with the iterative step

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

Since P_C is an av operator, the operator

$$Tx = P_C(x + \gamma A^\dagger (b - Ax))$$

is av for all γ in $(0, 2/L)$. Convergence again follows from the KM algorithm, whenever minimizers exist. Note that when $Ax = b$ has solutions in C, the projected Landweber algorithm converges to such a solution.

27.3.3 Splitting Methods

As we noted previously, the system $Ax = b$ has solutions if and only if the square system $AA^\dagger z = b$ has solutions. The *splitting methods* apply to square systems $Sz = h$. The idea is to decompose S into $S = M - K$, where M is easily inverted. Then

$$Sz = Mz - Kz = h.$$

The operator T given by

$$Tz = M^{-1}Kz + M^{-1}h$$

is affine linear and is av whenever the matrix $M^{-1}K$ is av. When $M^{-1}K$ is not Hermitian, if $M^{-1}K$ is a paracontraction, with respect to some norm, we can use Theorem 27.2.

Particular choices of M and K lead to Jacobi's method, the Gauss-Seidel method, and the more general Jacobi and Gauss-Seidel overrelaxation methods (JOR and SOR). For the case of S non-negative-definite, the JOR algorithm is equivalent to the Landweber algorithm and the SOR is closely related to the relaxed ART method. Convergence of both JOR and SOR in this case follows from the KM Theorem.

27.4 Positive Solutions of Linear Equations

Suppose now that the entries of the matrix A are non-negative, those of b are positive, and we seek a solution x with non-negative entries. We can, of course, use projected algorithms discussed in the previous section. Alternatively, we can use algorithms designed specifically for non-negative problems and based on cross-entropy, rather than on the Euclidean distance between vectors.

27.4.1 Cross-Entropy

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

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

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

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

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 non-negative x.

27.4.2 The EMML and SMART algorithms

The *expectation maximization maximum likelihood* (EMML) algorithm minimizes $KL(b, Ax)$, while the *simultaneous multiplicative* ART (SMART) minimizes $KL(Ax, b)$. These methods were developed for application to tomographic image reconstruction, although they have much more general uses. Whenever there are nonnegative solutions of $Ax = b$, SMART converges to the nonnegative solution that minimizes $KL(x, x^0)$; the EMML also converges to a non-negative solution, but no explicit description of that solution is known.

27.4.3 Acceleration

Both the EMML and SMART algorithms are simultaneous, like Cimmino's algorithm, using all the equations in each step of the iterative. Like Cimmino's algorithm, they are slow to converge. In the consistent case, the ART converges much faster than Cimmino's algorithm, and analogous successive- and block-projection methods for accelerating the EMML and SMART methods have been developed; including the *multiplicative* ART (MART), the *rescaled block-iterative* SMART (RBI-SMART) and the *rescaled block-iterative* EMML (RBI-EMML). These methods can be viewed as involving projections onto hyperplanes, but the projections are entropic, not orthogonal, projections.

27.4.4 Entropic Projections onto Hyperplanes

Let H_i be the hyperplane

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

For any non-negative z, denote by $x = P_i^e z$ the non-negative vector in H_i that minimizes the entropic distance $KL(x, z)$. Generally, we cannot express $P_i^e z$ in closed form. On the other hand, if we ask for the non-negative vector $x = Q_i^e z$ in H_i for which the weighted entropic distance

$$\sum_{j=1}^{J} A_{ij} KL(x_j, z_j)$$

is minimized, we find that $x = Q_i^e z$ can be written explicitly:

$$x_j = z_j \frac{b_i}{(Az)_i}.$$

We can use these weighted entropic projection operators Q_i^e to derive the MART, the SMART, the EMML, the RBI-SMART, and the RBI-EMML methods.

27.5 Sensitivity to Noise

In many applications of these iterative methods, the vector b consists of measurements, and therefore, is noisy. Even though exact solutions of

$Ax = b$ may exist, they may not be useful, because they are the result of over-fitting the answer to noisy data. It is important to know where sensitivity to noise can come from, and how modify the algorithms to lessen the sensitivity. Ill-conditioning in the matrix A can lead to sensitivity to noise and *regularization* can help to make the solution less sensitive to noise and other errors.

27.5.1 Norm Constraints

For example, in the inconsistent case, when we seek a least-squares solution of $Ax = b$, we minimize $||Ax - b||_2$. To avoid over-fitting to noisy data we can minimize

$$||Ax - b||_2^2 + \epsilon^2 ||x||_2^2,$$

for some small ϵ. In the consistent case, instead of calculating the exact solution that minimizes $||x - x^0||_2$, we can calculate the minimizer of

$$||Ax - b||_2^2 + \epsilon^2 ||x - x^0||_2^2.$$

These approaches to regularization involve the additional of a penalty term to the function being minimized. Such regularization can often be obtained through a Bayesian *maximum a posteriori probability* (MAP) approach.

Noise in the data can manifest itself in a variety of ways. For example, consider 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. We have the following result:

Theorem 27.3 *Suppose that A and every matrix Q obtained from A by deleting columns has full rank. Suppose there is no nonnegative solution of 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.*

This theorem 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.

27.6 Constrained Optimization

In image reconstruction, we often have prior constraints that we wish to impose on the vectorized image x, as well as measured data, with which a suitable x should be in reasonable agreement. Taken together, these constraints are usually insufficient to specify a unique solution; we obtain our desired solution by optimizing some cost function over all the x satisfying our constraints. This is constrained optimization.

27.6.1 Convex Feasibility and Split Feasibility

The constraints we wish to impose on x can often be formulated as requiring that x be a member of closed, convex sets C_i, $i = 1, ..., I$. In some cases, there are sufficiently many C_i so that any member of C, their intersection, will be a satisfactory answer to our problem. Finding a member of C is the *convex feasibility problem* (CFP). When the intersection C is empty, we can minimize a proximity function, such as

$$F(x) = \sum_{i=1}^{I} ||P_{C_i}x - x||_2^2.$$

When the intersection C is quite large, we may want to minimize a cost function $f(x)$ over the members of C. For example, we may want the member of C that is closest to x^0; that is, we want to minimize $||x - x^0||_2$ over C.

Let A be an I by J real matrix. The *split feasibility problem* (SFP) [62] is to find a member of a closed, convex set C in R^J for which Ax is a member of a second closed, convex set Q in R^I. When there is no such x, we can minimize the proximity function

$$G(x) = ||P_Q Ax - Ax||_2,$$

over all x in C, whenever such minimizers exist.

27.6.2 Algorithms

The CFP can be solved using the *successive orthogonal projections* (SOP) method. The iterative step of the SOP is

$$x^{k+1} = P_I P_{I-1} \cdots P_2 P_1 x^k,$$

where $P_i = P_{C_i}$ is the orthogonal projection onto C_i. The operator

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

is averaged and convergence of the SOP follows from the KM Theorem. The SOP is useful when the sets C_i are easily described and the P_i are easily calculated, but P_C is not. The SOP converges to the member of C closest to x^0 when the C_i are hyperplanes, but not in general.

When $C = \cap_{i=1}^{I} C_i$ is empty and we seek to minimize the proximity function $F(x)$, the relevant iteration is

$$x^{k+1} = \frac{1}{I}\sum_{i=1}^{I} P_i x^k.$$

The operator

$$T = \frac{1}{I}\sum_{i=1}^{I} P_i$$

is averaged, so this iteration converges, by the KM Theorem, whenever $F(x)$ has a minimizer.

The CQ algorithm for the SFP has the iterative step

$$x^{k+1} = P_C(x^k - \gamma A^T(I - P_Q)Ax^k). \tag{27.2}$$

The operator

$$T = P_C(I - \gamma A^T(I - P_Q)A)$$

is averaged whenever γ is in the interval $(0, 2/L)$, where L is the largest eigenvalue of $A^T A$, and so the CQ algorithm converges to a fixed point of T, whenever such fixed points exist. When the SFP has a solution, the CQ algorithm converges to a solution; when it does not, the CQ algorithm converges to a minimizer, over C, of the proximity function $||P_Q Ax - Ax||_2$, whenever such minimizers exist.

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

$$x^{k+1} = P_C(x^k - \gamma A^\dagger(I - P_Q)Ax^k). \tag{27.3}$$

When the intersection $C = \cap_{i=1}^{I} C_i$ is large, and just finding any member of C is not sufficient for our purposes, we may want to calculate the orthogonal projection of x^0 onto C using the operators P_{C_i}. We cannot use the SOP unless the C_i are hyperplanes; instead we can use Dykstra's algorithm or the Halpern-Lions-Wittmann-Bauschke (HLWB) algorithm. Dykstra's algorithm employs the projections P_{C_i}, but not directly on x^k, but on translations of x^k. It is motivated by the following lemma:

Lemma 27.1 *If $x = c + \sum_{i=1}^{I} p_i$, where, for each i, $c = P_{C_i}(c + p_i)$, then $c = P_C x$.*

Bregman discovered an iterative algorithm for minimizing a more general convex function $f(x)$ over x with $Ax = b$ and also x with $Ax \geq b$ [24]. These algorithms are based on his extension of the SOP to include projections with respect to generalized distances, such as entropic distances.

27.7 Bregman Projections and the SGP

If $f : R^J \rightarrow R$ is convex and differentiable, then, for all x and y, we have

$$D_f(x,y) = f(x) - f(y) - \langle \nabla f(y), x - y \rangle \geq 0.$$

If $\hat{x}$ minimizes $f(x)$ over x with $Ax = b$, then

$$\nabla f(\hat{x}) + A^\dagger c = 0,$$

for some vector c. Bregman's idea is to use $D_f(x,y)$ to define generalized projections, and then to mimic the SOP to solve for $\hat{x}$. Simply requiring that $f(x)$ be convex and differentiable is not sufficient for a complete theory and additional requirements are necessary; see the appendix on Bregman-Legendre functions and Bregman projections.

For each i, let $P_i^f z$ be the point in the hyperplane

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

that minimizes $D_f(x,z)$. Then $P_i^f z$ is the *Bregman projection* of z onto H_i and

$$\nabla f(P_i^f z) - \nabla f(z) = \lambda_i a^i,$$

for some λ_i, where a_i is the ith column of $A^\dagger$. Bregman's *successive generalized projection* (SGP) method has the iterative step

$$x^{k+1} = \nabla f^{-1}(\nabla f(x^k) + \lambda_k a^i),$$

for some scalar λ_k and $i = k(\text{mod}\, I) + 1$. The sequence $\{x^k\}$ will converge to x with $Ax = b$, provided solutions exist, and when x^0 is chosen so that $x^0 = A^\dagger d$, for some d, the sequence will converge to the solution that minimizes $f(x)$. Bregman also uses Bregman distances to obtain a primal-dual algorithm for minimizing $f(x)$ over all x with $Ax \geq b$. Dykstra's algorithm can be extended to include Bregman projections; this extended algorithm is then equivalent to the generalization of Bregman's primal-dual algorithm to minimize $f(x)$ over the intersection of closed , convex sets.

27.8 The Multiple-Distance SGP (MSGP)

As we noted earlier, both the EMML and SMART algorithms can be viewed in terms of weighted entropic projections onto hyperplanes. Unlike the SGP, the weighted entropic distances used vary with the hyperplane, suggesting that it may be possible to extend the SGP algorithm to include Bregman projections in which the function f is replaced by f_i that depends on the set C_i. It is known, however, that merely replacing the single

Bregman function f with f_i that varies with the i is not enough to guarantee convergence. The *multiple-distance* SGP (MSGP) algorithm achieves convergence by using a dominating Bregman distance $D_h(x,y)$ with

$$D_h(x,y) \geq D_{f_i}(x,y),$$

for each i, and a generalized notion of relaxation. The MSGP leads to an interior-point method, the IPA, for minimizing certain convex functions over convex sets.

27.9 Linear Programming

Bregman's primal-dual algorithm suggests a method for approximating the solution of the basic problem in linear programming, to minimize a linear function $c^T x$, over all x with $Ax \geq b$. Other solution methods exist for this problem, as well. Associated with the basic *primary* problem is a *dual* problem. Both the primary and dual problems can be stated in their *canonical forms* or their *standard forms*. The primary and dual problems are connected by the Weak Duality and Strong Duality theorems. The simplex method is the best known solution procedure.

27.10 Applications

Iterative algorithms are necessary in many areas of applications. Transmission and emission tomography involve the solving of large-scale systems of linear equations, or optimizing convex functions of thousands of variables. Magnetic-resonance imaging produces data that is related to the object of interest by means of the Fourier transform or the Radon transform. Hyperspectral imaging leads to several problems involving limited Fourier-transform data. Iterative data-extrapolation algorithms can be used to incorporate prior knowledge about the object being reconstructed, as well as to improve resolution. Entropy-based iterative methods are used to solve the mixture problems common to remote-sensing, as illustrated by sonar and radar array processing, as well as hyperspectral imaging.

Chapter 28

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 [153] as a method for including in transmission tomographic reconstruction spatially varying upper and lower bounds on the x-ray attenuation.

28.1 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 [48], 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 $n(k) = k(\text{mod}\, N) + 1$.

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.

28.2 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

$$x_j^{k+1} = \alpha_j^k v_j + (1 - \alpha_j^k) u_j, \tag{28.1}$$

with $n = n(k)$,

$$\alpha_j^k = \frac{c_j^k \prod^n (d_i^k)^{A_{ij}}}{1 + c_j^k \prod^n (d_i^k)^{A_{ij}}}, \tag{28.2}$$

$$c_j^k = \frac{(x_j^k - u_j)}{(v_j - x_j^k)}, \tag{28.3}$$

and

$$d_j^k = \frac{(b_i - (Au)_i)((Av)_i - (Ax^k)_i)}{((Av)_i - b_i)((Ax^k)_i - (Au)_i)}, \tag{28.4}$$

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 28.1 *If there is a soluton 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 [48].

28.3 The ABEMML Algorithm

We make the same assumptions as in the previous section. The iterative step of the ABEMML algorithm is

$$x_j^{k+1} = \alpha_j^k v_j + (1-\alpha_j^k)u_j, \tag{28.5}$$

where

$$\alpha_j^k = \gamma_j^k / d_j^k, \tag{28.6}$$

$$\gamma_j^k = (x_j^k - u_j)e_j^k, \tag{28.7}$$

$$\beta_j^k = (v_j - x_j^k)f_j^k, \tag{28.8}$$

$$d_j^k = \gamma_j^k + \beta_j^k, \tag{28.9}$$

$$e_j^k = \left(1 - \sum_{i \in B_n} A_{ij}\right) + \sum_{i \in B_n} A_{ij}\left(\frac{b_i - (Au)_i}{(Ax^k)_i - (Au)_i}\right), \tag{28.10}$$

and

$$f_j^k = \left(1 - \sum_{i \in B_n} A_{ij}\right) + \sum_{i \in B_n} A_{ij}\left(\frac{(Av)_i - b_i}{(Av)_i - (Ax^k)_i}\right). \tag{28.11}$$

We have the following theorem concerning the convergence of the ABEMML algorithm:

Theorem 28.2 *If there is a soluton 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 [48]. 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?

Chapter 29

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.

29.1 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

$$\hat{c} = \sum_{i=1}^J z_i \sigma_i^{-2}/(\sum_{j=1}^J \sigma_j^{-2}).$$

29.2 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 29.1 *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.

Exercise 29.2 *Show that for the estimator to be unbiased we need $K^\dagger H = I$, the identity matrix.*

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 29.3 *Show that for $\epsilon > 0$ we have*

$$(H^\dagger Q^{-1}H + \epsilon I)^{-1}H^\dagger Q^{-1} = H^\dagger(HH^\dagger + \epsilon Q)^{-1}. \tag{29.1}$$

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 (29.1) that

$$\mathbf{x}_{BLUE} = \lim_{\epsilon \to 0} H^\dagger(HH^\dagger + \epsilon Q)^{-1}\mathbf{z}. \tag{29.2}$$

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 $\mathbf{x} = H^\dagger \mathbf{a}$.

29.3 Some Useful Matrix Identities

In the exercise that follows we consider several matrix identities that are useful in developing the Kalman filter.

Exercise 29.4 *Establish the following identities, assuming that all the products and inverses involved are defined:*

$$CDA^{-1}B(C^{-1}-DA^{-1}B)^{-1}=(C^{-1}-DA^{-1}B)^{-1}-C; \tag{29.3}$$

$$(A-BCD)^{-1}=A^{-1}+A^{-1}B(C^{-1}-DA^{-1}B)^{-1}DA^{-1}; \tag{29.4}$$

$$A^{-1}B(C^{-1}-DA^{-1}B)^{-1}=(A-BCD)^{-1}BC; \tag{29.5}$$

$$(A-BCD)^{-1}=(I+GD)A^{-1}, \tag{29.6}$$

for

$$G=A^{-1}B(C^{-1}-DA^{-1}B)^{-1}.$$

Hints: To get Equation (29.3) use

$$C(C^{-1}-DA^{-1}B)=I-CDA^{-1}B.$$

For the second identity, multiply both sides of Equation (29.4) on the left by $A-BCD$ and at the appropriate step use Equation (29.3). For Equation (29.5) show that

$$BC(C^{-1}-DA^{-1}B)=B-BCDA^{-1}B=(A-BCD)A^{-1}B.$$

For Equation (29.6), substitute what G is and use Equation (29.4).

29.4 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

$$\mathbf{z} = H\mathbf{x} + \mathbf{v} \text{ and}$$

$$\mathbf{y} = \mathbf{x} + \mathbf{w}.$$

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 (29.4) and (29.5) we can rewrite this estimate in the more useful form

$$\hat{\mathbf{x}} = \mathbf{y} + G(\mathbf{z} - H\mathbf{y}),$$

for

$$G = RH^\dagger (Q + HRH^\dagger)^{-1}. \tag{29.7}$$

The covariance matrix of the optimal estimator is $K^\dagger PK$, which can be written as

$$K^\dagger PK = (R^{-1} + H^\dagger Q^{-1} H)^{-1} = (I - GH)R.$$

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 $K^\dagger PK$ is the posterior covariance of the current state. The algorithm proceeds recursively from one state to the next in time.

29.5 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 (29.7).

29.6 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 29.5 *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* $E(\mathbf{w}_k) = \mathbf{0}$.

29.7 Kalman Filtering and the BLUE

The *Kalman filter* [130, 107, 70] 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

$$\hat{\mathbf{x}}_k = C_k^\dagger \mathbf{z}_k + D_k^\dagger \mathbf{y}_k, \tag{29.8}$$

for matrices C_k and D_k to be determined. As we shall see, this estimate can also be written as

$$\hat{\mathbf{x}}_k = \mathbf{y}_k + G_k(\mathbf{z}_k - H_k \mathbf{y}_k), \tag{29.9}$$

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 [70].

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.

29.8 Adaptive Kalman Filtering

As in [70] 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

$$\hat{S} = \frac{1}{k-1}\sum_{j=1}^{k}(\mathbf{z}_j - H_j\mathbf{y}_j)(\mathbf{z}_j - H_j\mathbf{y}_j)^{\dagger}.$$

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, A. and Kak, A. (1984) "Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm."*Ultrasonic Imaging* **6**, pp. 81–94.

[5] Ash, R. and Gardner, M. (1975) *Topics in Stochastic Processes* Boston: Academic Press.

[6] Axelsson, O. (1994) *Iterative Solution Methods.* Cambridge, UK: Cambridge University Press.

[7] Baillet, S., Mosher, J., and Leahy, R. (2001) "Electromagnetic Brain Mapping" , *IEEE Signal Processing Magazine*, **18 (6)**, pp. 14–30.

[8] Barrett, H., White, T., and Parra, L. (1997) "List-mode likelihood."*J. Opt. Soc. Am. A* **14**, pp. 2914–2923.

[9] 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.

[10] 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.

[11] Bauschke, H. and Borwein, J. (1996) “On projection algorithms for solving convex feasibility problems.”*SIAM Review* **38 (3)**, pp. 367–426.

[12] 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.

[13] Bauschke, H., and Lewis, A. (2000) “Dykstra’s algorithm with Bregman projections: a convergence proof.” *Optimization*, **48**, pp. 409–427.

[14] Bertero, M. (1992) “Sampling theory, resolution limits and inversion methods.”in [16], pp. 71–94.

[15] Bertero, M., and Boccacci, P. (1998) *Introduction to Inverse Problems in Imaging* Bristol, UK: Institute of Physics Publishing.

[16] Bertero, M. and Pike, E.R., editors (1992) *Inverse Problems in Scattering and Imaging* Malvern Physics Series, Adam Hilger, IOP Publishing, London.

[17] Bertsekas, D.P. (1997) “A new class of incremental gradient methods for least squares problems.”*SIAM J. Optim.* **7**, pp. 913–926.

[18] Blackman, R. and Tukey, J. (1959) *The Measurement of Power Spectra.* New York: Dover Publications.

[19] 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.

[20] Born, M. and Wolf, E. (1999) *Principles of Optics: 7th edition.* Cambridge, UK: Cambridge University Press.

[21] Bochner, S. and Chandrasekharan, K. (1949) *Fourier Transforms*, Annals of Mathematical Studies, No. 19. Princeton, NJ: Princeton University Press.

[22] Borwein, J. and Lewis, A. (2000) *Convex Analysis and Nonlinear Optimization.* Canadian Mathematical Society Books in Mathematics, New York: Springer-Verlag.

[23] Bracewell, R.C. (1979) Image Reconstruction in Radio Astronomy, in [118], pp. 81–104.

[24] 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.

[25] 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.

[26] Brooks, D., and MacLeod, R. (1997) "Electrical Imaging of the Heart" *IEEE Signal Processing Magazine*, **14 (1)**, pp. 24–42.

[27] 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.

[28] 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.

[29] Burg, J. (1967) "Maximum entropy spectral analysis."*paper presented at the 37th Annual SEG meeting, Oklahoma City, OK.*

[30] Burg, J. (1972) "The relationship between maximum entropy spectra and maximum likelihood spectra."*Geophysics* **37**, pp. 375–376.

[31] Burg, J. (1975) *Maximum Entropy Spectral Analysis*, Ph.D. dissertation, Stanford University.

[32] 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.

[33] Byrne, C. and Fitzgerald, R. (1982) "Reconstruction from partial information, with applications to tomography."*SIAM J. Applied Math.* **42(4)**, pp. 933–940.

[34] 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.

[35] Byrne, C., and Wells, D. (1983) "Limit of continuous and discrete finite-band Gerchberg iterative spectrum extrapolation."*Optics Letters* **8 (10)**, pp. 526–527.

[36] 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.

[37] 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.

[38] 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.

[39] Byrne, C. and Fiddy, M. (1987) “Estimation of continuous object distributions from Fourier magnitude measurements.” *JOSA A* **4**, pp. 412–417.

[40] Byrne, C. and Fiddy, M. (1988) “Images as power spectra; reconstruction as Wiener filter approximation.” *Inverse Problems* **4**, pp. 399–409.

[41] Byrne, C., Haughton, D., and Jiang, T. (1993) “High-resolution inversion of the discrete Poisson and binomial transformations.” *Inverse Problems* **9**, pp. 39–56.

[42] Byrne, C. (1993) “Iterative image reconstruction algorithms based on cross-entropy minimization.” *IEEE Transactions on Image Processing* **IP-2**, pp. 96–103.

[43] 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.

[44] 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.

[45] Byrne, C. (1996) “Block-iterative methods for image reconstruction from projections.” *IEEE Transactions on Image Processing* **IP-5**, pp. 792–794.

[46] Byrne, C. (1997) “Convergent block-iterative algorithms for image reconstruction from inconsistent data.” *IEEE Transactions on Image Processing* **IP-6**, pp. 1296–1304.

[47] 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.

[48] Byrne, C. (1998) “Iterative deconvolution and deblurring with constraints” , *Inverse Problems*, **14**, pp. 1455-1467.

[49] Byrne, C. (1999) “Iterative projection onto convex sets using multiple Bregman distances.”*Inverse Problems* **15**, pp. 1295–1313.

[50] Byrne, C. (2000) “Block-iterative interior point optimization methods for image reconstruction from limited data.”*Inverse Problems* **16**, pp. 1405–1419.

[51] 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.,

[52] Byrne, C. (2001) “Likelihood maximization for list-mode emission tomographic image reconstruction.”*IEEE Transactions on Medical Imaging* **20(10)**, pp. 1084–1092.

[53] Byrne, C. (2002) “Iterative oblique projection onto convex sets and the split feasibility problem.”*Inverse Problems* **18**, pp. 441–453.

[54] Byrne, C. (2004) “A unified treatment of some iterative algorithms in signal processing and image reconstruction.”*Inverse Problems* **20**, pp. 103–120.

[55] Byrne, C. (2005) Choosing parameters in block-iterative or ordered-subset reconstruction algorithms, *IEEE Transactions on Image Processing*, **14 (3)**, pp. 321–327.

[56] Byrne, C. (2005) “Signal Processing: A Mathematical Approach” , AK Peters, Publ., Wellesley, MA.

[57] 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.

[58] Candy, J. (1988) *Signal Processing: The Modern Approach* New York: McGraw-Hill Publ.

[59] 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.

[60] Censor, Y. (1981) “Row-action methods for huge and sparse systems and their applications.”*SIAM Review*, **23**: 444–464.

[61] 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.

[62] Censor, Y. and Elfving, T. (1994) "A multiprojection algorithm using Bregman projections in a product space."*Numerical Algorithms* **8**, pp. 221–239.

[63] Censor, Y., Elfving, T., Kopf, N., and Bortfeld, T. (2006) "The multiple-sets split feasibility problem and its application for inverse problems." *Inverse Problems*, to appear.

[64] Censor, Y., Bortfeld, T., Martin, B., and Trofimov, A. (2006) "A unified approach for inversion problems in intensity-modulated radiation therapy." , to appear.

[65] Censor, Y., and Reich, S. (1998) "The Dykstra algorithm for Bregman projections." *Communications in Applied Analysis*, **2**, pp. 323–339.

[66] Censor, Y. and Segman, J. (1987) "On block-iterative maximization."*J. of Information and Optimization Sciences* **8**, pp. 275–291.

[67] Censor, Y. and Zenios, S.A. (1997) *Parallel Optimization: Theory, Algorithms and Applications*. New York: Oxford University Press.

[68] 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.

[69] Childers, D., editor (1978) *Modern Spectral Analysis*. New York:IEEE Press.

[70] Chui, C. and Chen, G. (1991) *Kalman Filtering*, second edition. Berlin: Springer-Verlag.

[71] Cimmino, G. (1938) "Calcolo approssimato per soluzioni die sistemi di equazioni lineari."*La Ricerca Scientifica XVI, Series II, Anno IX* **1**, pp. 326–333.

[72] Combettes, P. (1993) "The foundations of set theoretic estimation."*Proceedings of the IEEE* **81 (2)**, pp. 182–208.

[73] Combettes, P. (1996) "The convex feasibility problem in image recovery."*Advances in Imaging and Electron Physics* **95**, pp. 155–270.

[74] Combettes, P. (2000) "Fejér monotonicity in convex optimization."in *Encyclopedia of Optimization*, C.A. Floudas and P. M. Pardalos, editors, Boston: Kluwer Publ.

[75] 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.

[76] Combettes, P., and Wajs, V. (2005) Signal recovery by proximal forward-backward splitting, *Multiscale Modeling and Simulation*, **4(4)**, pp. 1168–1200.

[77] Cooley, J. and Tukey, J. (1965) "An algorithm for the machine calculation of complex Fourier series." *Math. Comp.*, **19**, pp. 297–301.

[78] Csiszár, I. (1989) "A geometric interpretation of Darroch and Ratcliff's generalized iterative scaling." *The Annals of Statistics* **17 (3)**, pp. 1409–1413.

[79] 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.

[80] Csiszár, I. and Tusnády, G. (1984) "Information geometry and alternating minimization procedures." *Statistics and Decisions* **Supp. 1**, pp. 205–237.

[81] Dainty, J. C. and Fiddy, M. (1984) "The essential role of prior knowledge in phase retrieval." *Optica Acta* **31**, pp. 325–330.

[82] Darroch, J. and Ratcliff, D. (1972) "Generalized iterative scaling for log-linear models." *Annals of Mathematical Statistics* **43**, pp. 1470–1480.

[83] Dax, A. (1990) "The convergence of linear stationary iterative processes for solving singular unstructured systems of linear equations," *SIAM Review*, **32**, pp. 611–635.

[84] 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.

[85] 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.

[86] 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.

[87] 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)**.

[88] 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.

[89] 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.

[90] Duda, R., Hart, P., and Stork, D. (2001) *Pattern Classification*, Wiley.

[91] Dugundji, J. (1970) *Topology* Boston: Allyn and Bacon, Inc.

[92] Dykstra, R. (1983) “An algorithm for restricted least squares regression” *J. Amer. Statist. Assoc.*, **78 (384)**, pp. 837–842.

[93] 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.

[94] Elsner, L., Koltracht, L., and Neumann, M. (1992) “Convergence of sequential and asynchronous nonlinear paracontractions.” *Numerische Mathematik*, **62**, pp. 305–319.

[95] Erdogan, H., and Fessler, J. (1999) “Fast monotonic algorithms for transmission tomography” *IEEE Transactions on Medical Imaging*, **18(9)**, pp. 801–814.

[96] Everitt, B. and Hand, D. (1981) *Finite Mixture Distributions* London: Chapman and Hall.

[97] Farncombe, T. (2000) “Functional dynamic SPECT imaging using a single slow camera rotation” , *Ph.D. thesis, Dept. of Physics, University of British Columbia.*

[98] 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.

[99] 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.

[100] Feynman, R., Leighton, R., and Sands, M. (1963) *The Feynman Lectures on Physics, Vol. 1.* Boston: Addison-Wesley.

[101] Fiddy, M. (1983) “The phase retrieval problem.”in *Inverse Optics*, SPIE Proceedings 413 (A.J. Devaney, editor), pp. 176–181.

[102] Fienup, J. (1979) “Space object imaging through the turbulent atmosphere.” *Optical Engineering* **18**, pp. 529–534.

[103] 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.

[104] Fleming, W. (1965) *Functions of Several Variables*, Addison-Wesley Publ., Reading, MA.

[105] Frieden, B. R. (1982) *Probability, Statistical Optics and Data Testing*. Berlin: Springer-Verlag.

[106] Gasquet, C. and Witomski, F. (1998) *Fourier Analysis and Applications*. Berlin: Springer-Verlag.

[107] Gelb, A., editor, (1974) *Applied Optimal Estimation*, written by the technical staff of The Analytic Sciences Corporation, MIT Press, Cambridge, MA.

[108] 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.

[109] Gerchberg, R. W. (1974) “Super-restoration through error energy reduction.” *Optica Acta* **21**, pp. 709–720.

[110] 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.

[111] Golshtein, E., and Tretyakov, N. (1996) *Modified Lagrangians and Monotone Maps in Optimization*. New York: John Wiley and Sons, Inc.

[112] 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.

[113] Green, P. (1990) “Bayesian reconstructions from emission tomography data using a modified EM algorithm.” *IEEE Transactions on Medical Imaging* **9**, pp. 84–93.

[114] 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.

[115] Haacke, E., Brown, R., Thompson, M., and Venkatesan, R. (1999) *Magnetic Resonance Imaging.* New York: Wiley-Liss.

[116] Haykin, S. (1985) *Array Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall.

[117] 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.

[118] Herman, G.T. (ed.) (1979) "Image Reconstruction from Projections" , *Topics in Applied Physics, Vol. 32*, Springer-Verlag, Berlin.

[119] Herman, G.T., and Natterer, F. (eds.) (1981) "Mathematical Aspects of Computerized Tomography" , *Lecture Notes in Medical Informatics, Vol. 8*, Springer-Verlag, Berlin.

[120] Herman, G.T., Censor, Y., Gordon, D., and Lewitt, R. (1985) Comment (on the paper [187]), *Journal of the American Statistical Association* **80**, pp. 22–25.

[121] Herman, G. T. and Meyer, L. (1993) "Algebraic reconstruction techniques can be made computationally efficient."*IEEE Transactions on Medical Imaging* **12**, pp. 600–609.

[122] Hildreth, C. (1957) "A quadratic programming procedure."*Naval Research Logistics Quarterly* **4**, pp. 79–85. Erratum, p. 361.

[123] Hogg, R. and Craig, A. (1978) *Introduction to Mathematical Statistics* MacMillan, New York.

[124] 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.

[125] 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.

[126] 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.

[127] 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.

[128] Kaczmarz, S. (1937) “Angenäherte Auflösung von Systemen linearer Gleichungen.” *Bulletin de l’Academie Polonaise des Sciences et Lettres* **A35**, pp. 355–357.

[129] Kak, A., and Slaney, M. (2001) “Principles of Computerized Tomographic Imaging” , SIAM, Philadelphia, PA.

[130] Kalman, R. (1960) “A new approach to linear filtering and prediction problems.” *Trans. ASME, J. Basic Eng.* **82**, pp. 35–45.

[131] Katznelson, Y. (1983) *An Introduction to Harmonic Analysis.* New York: John Wiley and Sons, Inc.

[132] 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 [189], pp. 473–498.

[133] Koltracht, L., and Lancaster, P. (1990) “Constraining strategies for linear iterative processes.” *IMA J. Numer. Anal.*, **10**, pp. 555–567.

[134] Körner, T. (1988) *Fourier Analysis.* Cambridge, UK: Cambridge University Press.

[135] Körner, T. (1996) *The Pleasures of Counting.* Cambridge, UK: Cambridge University Press.

[136] Kullback, S. and Leibler, R. (1951) “On information and sufficiency.” *Annals of Mathematical Statistics* **22**, pp. 79–86.

[137] Landweber, L. (1951) “An iterative formula for Fredholm integral equations of the first kind.” *Amer. J. of Math.* **73**, pp. 615–624.

[138] Lane, R. (1987) “Recovery of complex images from Fourier magnitude.” *Optics Communications* **63(1)**, pp. 6–10.

[139] Lange, K. and Carson, R. (1984) “EM reconstruction algorithms for emission and transmission tomography.” *Journal of Computer Assisted Tomography* **8**, pp. 306–316.

[140] 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.

[141] 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.

[142] 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.

[143] 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.

[144] 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.

[145] 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.

[146] Luenberger, D. (1969) *Optimization by Vector Space Methods*. New York: John Wiley and Sons, Inc.

[147] Mann, W. (1953) "Mean value methods in iteration."*Proc. Amer. Math. Soc.* **4**, pp. 506–510.

[148] McLachlan, G.J. and Krishnan, T. (1997) *The EM Algorithm and Extensions*. New York: John Wiley and Sons, Inc.

[149] McVeigh, E., and Ozturk, C. (2001) "Imaging Myocardial Strain" , *IEEE Signal Processing Magazine*, **18 (6)**, pp. 44–56.

[150] 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.

[151] Meijering, E., Smal, I., and Danuser, G. (2006) "Tracking in Molecular Bioimaging" , *IEEE Signal Processing Magazine*, **23 (3)**, pp. 46–53.

[152] Motzkin, T. and Schoenberg, I. (1954) "The relaxation method for linear inequalities."*Canadian Journal of Mathematics* **6**, pp. 393–404.

[153] 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.

[154] Nash, S. and Sofer, A. (1996) *Linear and Nonlinear Programming*. New York: McGraw-Hill.

[155] Natterer, F. (1986) *Mathematics of Computed Tomography*. New York: John Wiley and Sons, Inc.

[156] Natterer, F., and Wübbeling, F. (2001) *Mathematical Methods in Image Reconstruction*. Philadelphia, PA: SIAM Publ.

[157] Ollinger, J., and Fessler, J. (1997) "Positron-Emission Tomography" , *IEEE Signal Processing Magazine*, **14 (1)**, pp. 43–55.

[158] Oppenheim, A. and Schafer, R. (1975) *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall.

[159] Papoulis, A. (1975) "A new algorithm in spectral analysis and band-limited extrapolation."*IEEE Transactions on Circuits and Systems* **22**, pp. 735–742.

[160] Papoulis, A. (1977) *Signal Analysis*. New York: McGraw-Hill.

[161] 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.

[162] Paulraj, A., Roy, R., and Kailath, T. (1986) "A subspace rotation approach to signal parameter estimation."*Proceedings of the IEEE* **74**, pp. 1044–1045.

[163] Peressini, A., Sullivan, F., and Uhl, J. (1988) *The Mathematics of Nonlinear Programming*. Berlin: Springer-Verlag.

[164] Peters, T. (1981) "Resolution improvement to CT systems using aperture-function correction" , in [119], pp. 241–251.

[165] 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.

[166] 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.

[167] Poggio, T. and Smale, S. (2003) "The mathematics of learning: dealing with data."*Notices of the American Mathematical Society* **50 (5)**, pp. 537–544.

[168] Priestley, M. B. (1981) *Spectral Analysis and Time Series.* Boston: Academic Press.

[169] Qian, H. (1990) “Inverse Poisson transformation and shot noise filtering.”*Rev. Sci. Instrum.* **61**, pp. 2088–2091.

[170] Quistgaard, J. (1997) “Signal Acquisition and Processing in Medical Diagnostic Ultrasound” , *IEEE Signal processing Magazine*, **14 (1)**, pp. 67–74.

[171] Rockafellar, R. (1970) *Convex Analysis.* Princeton, NJ: Princeton University Press.

[172] 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.

[173] Sarder, P., and Nehorai, A. (2006) “Deconvolution Methods for 3-D Fluoresence Microscopy Images” , *IEEE Signal Processing Magazine*, **23 (3)**, pp. 32–45.

[174] Saulnier, G., Blue, R., Newell, J., Isaacson, D., and Edic, P. (2001) “Electrical Impedance Tomography” , *IEEE Signal Processing Magazine*, **18 (6)**, pp. 31–43.

[175] Schmidlin, P. (1972) “Iterative separation of sections in tomographic scintigrams.”*Nucl. Med.* **15(1)**.

[176] Shepp, L., and Vardi, Y. (1982) Maximum likelihood reconstruction for emission tomography, *IEEE Transactions on Medical Imaging*, **MI-1**, pp. 113–122.

[177] Smith, C. Ray and Grandy, W.T., editors (1985) *Maximum-Entropy and Bayesian Methods in Inverse Problems.* Dordrecht: Reidel Publ.

[178] Smith, C. Ray and Erickson, G., editors (1987) *Maximum-Entropy and Bayesian Spectral Analysis and Estimation Problems.* Dordrecht: Reidel Publ.

[179] 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.

[180] 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.

[181] Strang, G. (1980) *Linear Algebra and its Applications*. New York: Academic Press.

[182] Tanabe, K. (1971) "Projection method for solving a singular system of linear equations and its applications."*Numer. Math.* **17**, pp. 203–214.

[183] Therrien, C. (1992) *Discrete Random Signals and Statistical Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall.

[184] Twomey, S. (1996) *Introduction to the Mathematics of Inversion in Remote Sensing and Indirect Measurement*. New York: Dover Publ.

[185] 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.

[186] Van Trees, H. (1968) *Detection, Estimation and Modulation Theory*. New York: John Wiley and Sons, Inc.

[187] 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.

[188] Vonesch, C., Aguet, F., Vonesch, J-L, and Unser, M. (2006) "The Colored Revolution in BioImaging" , *IEEE Signal Processing Magazine*, **23 (3)**, pp. 20–31.

[189] Wernick, M. and Aarsvold, J., editors (2004) *Emission Tomography: The Fundamentals of PET and SPECT*. San Diego: Elsevier Academic Press.

[190] Wiener, N. (1949) *Time Series*. Cambridge, MA: MIT Press.

[191] Wright, G.A. (1997) "Magnetic Resonance Imaging" , *IEEE Signal Processing Magazine*, **14 (1)**, pp. 56–66.

[192] Wright, W., Pridham, R., and Kay, S. (1981) "Digital signal processing for sonar."*Proc. IEEE* **69**, pp. 1451–1506.

[193] Yang, Q. (2004) "The relaxed CQ algorithm solving the split feasibility problem." *Inverse Problems*, **20**, pp. 1261–1266.

[194] Youla, D. (1978) "Generalized image restoration by the method of alternating projections."*IEEE Transactions on Circuits and Systems* **CAS-25 (9)**, pp. 694–702.

[195] 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.

[196] Young, R. (1980) *An Introduction to Nonharmonic Fourier Analysis*. Boston: Academic Press.

[197] Zhou, X., and Wong, S. (2006) “Informatics challenges of high-throughput microscopy” , *IEEE Signal Processing Magazine*, **23 (3)**, pp. 63–72.

[198] 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