Iterative Algorithms in Inverse Problems

Charles L. Byrne

April 25, 2006

Contents

Part I

Preliminaries

Chapter 1

Preface

VALENTINE: What she's doing is, every time she works out a value for y, she's using *that* as her next value for x. And so on. Like a feedback. She's feeding the solution into the equation, and then solving it again. Iteration, you see. ... This thing works for any phenomenon which eats its own numbers.

HANNAH: What I don't understand is... why nobody did this feedback thing before- it's not like relativity, you don't have to be Einstein.

VALENTINE: You couldn't see to look before. The electronic calculator was what the telescope was for Galileo.

HANNAH: Calculator?

VALENTINE: There wasn't enough time before. There weren't enough *pencils*. ... Now she'd only have to press a button, the same button, over and over. Iteration. ... And so boring!

HANNAH: Do you mean that was the only problem? Enough time? And paper? And the boredom?

VALENTINE: Well, the other thing is, you'd have to be insane.

Arcadia (Act 1, Scene 4), by Tom Stoppard

The well known formula for solving a quadratic equation produces the answer in a finite number of calculations; it is a non-iterative method, if we are willing to accept a square-root symbol in our answer. Similarly, Gauss elimination gives the solution to a system of linear equations, if there is one, in a finite number of steps; it, too, is a non-iterative method. A typical iterative algorithm (the name comes from the Latin word *iterum*, meaning "again"), involves a relatively simple calculation, performed repeatedly. An iterative method produces a sequence of approximate answers that, in the best case, converges to the solution. The characters in Stoppard's play are discussing the apparent anticipation, by a (fictional) teenage girl in 1809, of the essential role of iterative algorithms in chaos theory and fractal geometry. A good example of an iterative algorithm is the *bi-section method* for finding a root of a real-valued continuous function $f(x)$ of the real variable x: begin with an interval $[a, b]$ such that $f(a)f(b) < 0$ and then replace one of the endpoints with the average $\frac{a+b}{2}$, maintaining the negative product. The length of each interval so constructed is half the length of the previous interval and each interval contains a root. In the limit, the two sequences defined by the left endpoints and right endpoints converge to the same root.

Iterative algorithms are used to solve problems for which there is no non-iterative solution method, as well as problems for which non-iterative methods are impractical, such as using Gauss elimination to solve a system of thousands of linear equations in thousands of unknowns. We may want to find a root of $f(x) = x^2 - 2$ in order to approximate $\sqrt{2}$, or to solve an algebraic equation, such as $x = \tan x$, by writing the equation as $f(x) = x - \tan x = 0$. On the other hand, we may want a root of $f(x)$ because $f(x)$ is the derivative of another function, say $F(x)$, that we wish to optimize. If our goal is to minimize $F(x)$, we may choose, instead, to generate an iterative sequence $\{x^k\}$, $k = 0, 1, ...$, that converges to a minimizer of $F(x)$.

Iterative algorithms are often formulated as *fixed-point* methods: the equation $f(x) = 0$ is equivalent to $x = f(x) + x = g(x)$, so we may try to find a fixed point of $g(x)$, that is, an x for which $g(x) = x$.

The idea of using iterative procedures for solving problems is an ancient one. Archimedes' use of the areas of inscribed and circumscribed regular polygons to estimate the area of a circle is a famous instance of an iterative procedure, as is his method of exhaustion for finding the area of a section of a parabola.

It is not our aim here to describe all the various problems that can be solved by iterative methods. We shall focus on iterative methods currently being used in inverse problems, with special attention to remote-sensing applications, such as image reconstruction from tomographic data in medical diagnostics and acoustic array signal processing. Such methods include those for solving large systems of linear equations, with and without constraints, optimization techniques, such as likelihood and entropy maximiza-

tion, data-extrapolation procedures, and algorithms for convex feasibility problems.

Throughout these discussions we shall be concerned with the speed of the algorithms, as well as their sensitivity to noise or errors in the data; methods for accelerating and regularizing the algorithms will be treated in detail.

The iterative algorithms we discuss take the form $x^{k+1} = Tx^k$, where T is some (usually nonlinear) continuous operator on the space R^J of J-dimensional real vectors, or C^J, the space of J-dimensional complex vectors. If the sequence $\{T^k x^0\}$ converges to x^*, then $Tx^* = x^*$, that is, x^* is a *fixed point* of T. To be sure that the sequence $\{T^k x^0\}$ converges, we need to know that T has fixed points, but we need more than that.

We shall focus on two broad classes of operators, those that are *averaged, non-expansive* with respect to the Euclidean vector norm, and those that are *paracontractive* with respect to some vector norm. Convergence for the first class of operators is a consequence of the Krasnoselskii/Mann (KM) Theorem, and the Elsner/Koltracht/Neumann (EKN) Theorem establishes convergence for the second class. The definitions of these classes are derived from basic properties of orthogonal projection operators, which are members of both classes.

In many remote-sensing applications, the (discretized) object sought is naturally represented as a vector with nonnegative entries. For such problems, we can incorporate nonnegativity in the algorithms through the use of projections with respect to entropy-based distances. These algorithms are often developed by analogy with those methods using orthogonal projections. As we shall see, this analogy can often be further exploited to derive convergence theorems.

The cross-entropy distance is just one example of a Bregman distance. The notion of an operator being paracontractive, with respect to a norm, can be extended to being paracontractive, with respect to a Bregman distance. Bregman projections onto convex sets are paracontractive in this generalized sense, as are many of the operators of interest. The EKN Theorem and many of its corollaries can be extended to operators that are paracontractive, with respect to Bregman distances.

We begin with an overview of the algorithms and their applications.

Chapter 2

Introduction

Because the field of iterative algorithms is vast, any set of lecture notes must involve selection of a few topics that the author wishes to treat in some detail. Here the goal is to discuss those methods most relevant to image reconstruction and signal processing, especially medical tomographic image reconstruction.

2.1 Overview

Although our chosen subject may sound narrow, it includes many of the topics found in standard texts on procedures for iterative solution of linear equations and iterative optimization. Our goal will require us to discuss classes of linear and nonlinear operators on finite-dimensional real and complex Euclidean space and the fixed-point algorithms associated with these operators, eigenvalues and eigenvectors of matrices, cross-entropy distance between nonnegative vectors, Fourier analysis, statistical likelihood maximization and Bayesian methods, regularization to decrease sensitivity to noise, and acceleration techniques.

2.1.1 Image Reconstruction in Tomography

Image reconstruction from tomographic data is a fairly recent, and increasingly important, area of applied numerical linear algebra, particularly for medical diagnosis [74, 78, 89, 107, 108, 120, 121] . In the so-called *algebraic* approach, the problem is to solve, at least approximately, a large system of linear equations, $Ax = b$. The vector x is large because it is usually a vectorization of a discrete approximation of a function of two or three continuous spatial variables. The size of the system necessitates the use of iterative solution methods [95]. Because the entries of x usually represent intensity levels, of beam attenuation in transmission tomography, and

of radionuclide concentration in emission tomography, we require x to be nonnegative; the physics of the situation may impose additional constraints on the entries of x. In practice, we often have prior knowledge about the function represented, in discrete form, by the vector x and we may wish to include this knowledge in the reconstruction. In tomography the entries of A and b are also nonnegative. Iterative algorithms tailored to find solutions to these special, constrained problems may out-perform general iterative solution methods [105]. To be medically useful in the clinic, the algorithms need to produce acceptable reconstructions early in the iterative process.

2.1.2 Systems of Linear Equations

Exact solutions of $Ax = b$ may not exist, so we need appropriate measures of distance between vectors to obtain suitable approximate solutions. In tomography and other forms of remote sensing, the entries of the vector b are data obtained by measurements, and so are noisy. Consequently, exact solutions of $Ax = b$, even when available, may be too noisy to be useful. Bayesian or penalized optimization algorithms are used to obtain reconstructions displaying the desired smoothness [59, 71, 75, 77, 96, 98].

2.1.3 Iterative Methods

The basic idea in iterative algorithms is to begin with an initial vector x^0 and to transform that vector to get x^1, and continue in this way to generate a sequence of vectors $\{x^k\}$, each obtained from the previous one by some transformation, which we denote by T. The iterative step is $x^{k+1} = Tx^k$. In the limit, further transformaton should result in no change; that is, we have a fixed point of T. If there is a unique solution $\hat{x}$ of the problem, we often require that we get closer to $\hat{x}$ with each step of the iteration; that is,

$$||\hat{x} - x^{k+1}|| < ||\hat{x} - x^k||.$$

It is sensible, then, that we focus on operators T that are *non-expansive*(ne), which means that

$$||Tx - Tz|| \leq ||x - z||,$$

for all vectors x and z, where $||x||$ denotes the Euclidean length or the 2-norm of the vector x. Being ne is not enough, in most cases, and we shall require T to have additional properties that guarantee convergence of the sequence $\{x^k\}$. Although the 2-norm and the Euclidean distance between vectors will play a prominent role in what follows, we shall also be interested in other notions of distance, such as cross-entropy, along with operators that are well-behaved with respect to these other distances.

Certain iterative algorithms require that we select a parameter that governs the size of the steps taken at each iteration. For the Landweber

and projected Landweber methods [12], this parameter is dependent on the largest eigenvalue, λ_{max}, of the matrix $A^\dagger A$. Because the system is large, calculating $A^\dagger A$, let alone computing λ_{max}, is impractical. If we overestimate λ_{max}, the step lengths become too small and the algorithm is too slow to be practical; tight upper bounds for λ_{max} that can be obtained from A itself help to accelerate these algorithms. Upper bounds exist that are particularly useful for the common case in which A is sparse, that is, most of its entries are zero [31]. These upper bounds are shown to become tighter as the size of the system increases [36].

The *Fourier* approach to tomographic image reconstruction maintains, at least initially, the continuous model for the attenuation function. The data are taken to be line integrals through the attenuator, that is, values of its so-called *x-ray transform*, which, in the two-dimensional case, is the Radon transform. The Central Slice Theorem then relates the Radon-transform values to values of the Fourier transform of the attenuation function. Image reconstruction then becomes estimation of the (inverse) Fourier transform. In magnetic-resonance imaging (MRI), we again have the measured data related to the function we wish to image, the proton density function, by a Fourier relation.

In the transmission and emission tomography, the data are photon counts, so it is natural to adopt a statistical model and to convert the image reconstruction problem into a statistical parameter-estimation problem. The estimation can be done using maximum likelihood (ML) or maximum *a posteriori* (MAP) Bayesian methods, which then require iterative optimization algorithms.

2.2 Tomography

These days, the term *tomography* is used by lay people and practitioners alike to describe any sort of scan, from ultrasound to magnetic resonance. It has apparently lost its association with the idea of slicing, as in the expression *three-dimensional tomography*. In this paper we focus on two important modalities, transmission tomography and emission tomography. An x-ray CAT scan is an example of the first, a positron-emission (PET) scan is an example of the second. Although there is some flexibility in the mathematical description of the image reconstruction problem posed by these methods, we shall concentrate here on the algebraic formulation of the problem. In this formulation, the problem is to solve, at least approximately, a large system of linear equations, $Ax = b$. What the entries of the matrix A and the vectors x and b represent will vary from one modality to another; for our purposes, the main point is simply that all of these entries are nonnegative.

In both modalities the vector x that we seek is a vectorization, that

is, a one-dimensional encoding, of an unknown two- or three-dimensional discrete function. It is this transition from higher dimensions to a single dimension that causes x to be large. The quantity x_j, the j-th entry of the vector x, represents the value of the function at the *pixel* or *voxel* corresponding to the index j. The quantity b_i, the i-th entry of the vector b, is measured data, the discrete line integral of x along the i-th line segment, in the transmission case, and photon counts at the i-th detector in the emission case. The entries of the matrix A describe the relationship that holds between the various pixels and the various detectors, that is, they describe the scanning process whereby the information about the unknown function is translated into measured data. In the transmission case, the entries of A describe the geometric relationship between the patient and the scanner, as well as the paths taken by the beams. In the emission case, the entries of A are the probabilities of a photon being detected at the various detectors, given that it was emitted at a particular pixel. In both cases, there is a certain amount of simplification and guesswork that goes into the choice of these entries. In the emission case, the probabilities depend, in part, on the attenuation encountered as the photons pass from within the body to the exterior, and so will depend on the anatomy of the particular patient being scanned.

2.2.1 Transmission Tomography

When an x-ray beam travels along a line segment through the body it becomes progressively weakened by the material it encounters. By comparing the initial strength of the beam as it enters the body with its final strength as it exits the body, we can estimate the integral of the attenuation function, along that line segment. The data in transmission tomography are these line integrals, corresponding to thousands of lines along which the beams have been sent. The image reconstruction problem is to create a discrete approximation of the attenuation function. The inherently three-dimensional problem is usually solved one two-dimensional plane, or slice, at a time, hence the name *tomography* [78].

The beam attenuation at a given point in the body will depend on the material present at that point; estimating and imaging the attenuation as a function of spatial location will give us a picture of the material within the body. A bone fracture will show up as a place where significant attenuation should be present, but is not.

The attenuation function is discretized, in the two-dimensional case, by imagining the body to consist of finitely many squares, or *pixels*, within which the function has a constant, but unknown, value. This value at the j-th pixel is denoted x_j. In the three-dimensional formulation, the body is viewed as consisting of finitely many cubes, or *voxels*. The beam is sent through the body along various lines and both initial and final

beam strength is measured. From that data we can calculate a discrete line integral along each line. For $i = 1, ..., I$ we denote by L_i the i-th line segment through the body and by b_i its associated line integral. Denote by A_{ij} the length of the intersection of the j-th pixel with L_i; therefore, A_{ij} is nonnegative. Most of the pixels do not intersect line L_i, so A is quite sparse. Then the data value b_i can be described, at least approximately, as

$$b_i = \sum_{j=1}^{J} A_{ij} x_j. \tag{2.1}$$

Both I, the number of lines, and J, the number of pixels or voxels, are quite large, although they certainly need not be equal, and are typically unrelated.

The matrix A is large and rectangular. The system $Ax = b$ may or may not have exact solutions. We are always free to select J, the number of pixels, as large as we wish, limited only by computation costs. We may also have some choice as to the number I of lines, but within the constraints posed by the scanning machine and the desired duration and dosage of the scan. When the system is underdetermined ($J > I$), there may be infinitely many exact solutions; in such cases we usually impose constraints and prior knowledge to select an appropriate solution. As we mentioned earlier, noise in the data, as well as error in our model of the physics of the scanning procedure, may make an exact solution undesirable, anyway. When the system is overdetermined ($J < I$), we may seek a least-squares approximate solution, or some other approximate solution. We may have prior knowledge about the physics of the materials present in the body that can provide us with upper bounds for x_j, as well as information about body shape and structure that may tell where $x_j = 0$. Incorporating such information in the reconstruction algorithms can often lead to improved images [105].

2.2.2 Emission Tomography

In *single-photon emission tomography* (SPECT) and *positron emission tomography* (PET) the patient is injected with, or inhales, a chemical to which a radioactive substance has been attached [121]. The chemical is designed to become concentrated in the particular region of the body under study. Once there, the radioactivity results in photons that travel through the body and, at least some of the time, are detected by the scanner. The function of interest is the actual concentration of the radioactive material at each spatial location within the region of interest. Learning what the concentrations are will tell us about the functioning of the body at the various spatial locations. Tumors may take up the chemical (and its radioactive passenger) more avidly than normal tissue, or less avidly, perhaps. Mal-

functioning portions of the brain may not receive the normal amount of the chemical and will, therefore, exhibit an abnormal amount of radioactivity.

As in the transmission tomography case, this nonnegative function is discretized and represented as the vector x. The quantity b_i, the i-th entry of the vector b, is the photon count at the i-th detector; in coincidence-detection PET a detection is actually a nearly simultaneous detection of a photon at two different detectors. The entry A_{ij} of the matrix A is the probability that a photon emitted at the j-th pixel or voxel will be detected at the i-th detector.

In the emission tomography case it is common to take a statistical view [94, 93, 112, 115, 120], in which the quantity x_j is the expected number of emissions at the j-th pixel during the scanning time, so that the expected count at the i-th detector is

$$E(b_i) = \sum_{j=1}^{J} A_{ij} x_j. \tag{2.2}$$

The system of equations $Ax = b$ is obtained by replacing the expected count, $E(b_i)$, with the actual count, b_i; obviously, an exact solution of the system is not needed in this case. As in the transmission case, we seek an approximate, and nonnegative, solution of $Ax = b$, where, once again, all the entries of the system are nonnegative.

2.2.3 Maximum-Likelihood Parameter Estimation

The measured data in tomography are values of random variables. The probabilities associated with these random variables are used in formulating the image reconstruction problem as one of solving a large system of linear equations. We can also use the stochastic model of the data to formulate the problem as a statistical parameter-estimation problem, which suggests the image be estimated using likelihood maximization. When formulated that way, the problem becomes a constrained optimization problem. The desired image can then be calculated using general-purpose iterative optimization algorithms, or iterative algorithms designed specifically to solve the particular problem.

Part II

Fixed-Point Iterative Algorithms

Chapter 3

Convergence Theorems

In this chapter we consider three fundamental convergence theorems that will play important roles in much of what follows.

3.1 Fixed Points of Iterative Algorithms

The iterative methods we shall consider can be formulated as

$$x^{k+1} = Tx^k, \tag{3.1}$$

for $k = 0, 1, ...$, where T is a linear or nonlinear continuous operator on (all or some of) the space $\mathcal{X}$ of real or complex J-dimensional vectors and x^0 is an arbitrary starting vector. For any such operator T on $\mathcal{X}$ the *fixed point set* of T is

$$\text{Fix}(T) = \{z | \, Tz = z\}.$$

Exercise 3.1 *Show that, if the iterative sequence defined by Equation (3.1) converges, then the limit is a member of Fix(T).*

A wide variety of problems can be solved by finding a fixed point of a particular operator and algorithms for finding such points play a prominent role in a number of applications. The paper [124] is an excellent source of background on these topics, particularly as they apply to signal and image processing. The more recent article by Bauschke and Borwein [8] is also quite helpful. The book by Borwein and Lewis [14] is an important reference.

In the algorithms of interest here the operator T is selected so that the set Fix(T) contains those vectors z that possess the properties we desire in a solution to the original signal processing or image reconstruction problem; finding a fixed point of the iteration leads to a solution of our problem.

3.2 Convergence Theorems for Iterative Algorithms

In general, a sequence of the form $\{T^k x^0\}$ need not converge, even when T has fixed points. The Newton-Raphson iteration, for example, may converge only when the starting vector x^0 is sufficiently close to a solution. We shall be concerned mainly with classes of operators T for which convergence holds for all starting vectors, whenever T has fixed points. The class of *strict contractions* provides a good example.

3.2.1 Strict Contractions

An operator T on $\mathcal{X}$ is *Lipschitz continuous*, with respect to a vector norm $||\cdot||$, if there is a positive constant λ such that

$$||Tx - Ty|| \leq \lambda ||x - y||,$$

for all x and y in $\mathcal{X}$.

An operator T on $\mathcal{X}$ is a *strict contraction* (sc), with respect to a vector norm $||\cdot||$, if there is $r \in (0, 1)$ such that

$$||Tx - Ty|| \leq r||x - y||,$$

for all vectors x and y.

Exercise 3.2 *Show that a strict contraction can have at most one fixed point.*

For strict contractions, we have the Banach-Picard theorem [64]:

Theorem 3.1 *Let T be sc. Then, there is a unique fixed point and, for any starting vector x^0, the sequence $\{T^k x^0\}$ converges to the fixed point.*

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

Exercise 3.3 *Show that the sequence $\{x^k\}$ is a Cauchy sequence. Hint: consider*

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

and use

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

Exercise 3.4 *Since $\{x^k\}$ is a Cauchy sequence, it has a limit, say $\hat{x}$. Let $e^k = \hat{x} - x^k$. Show that $\{e^k\} \to 0$, as $k \to +\infty$, so that $\{x^k\} \to \hat{x}$. Finally, show that $T\hat{x} = \hat{x}$.*

Exercise 3.5 *Suppose that we want to solve the equation*

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

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

Exercise 3.6 *Let T be an affine operator, that is, T has the form $Tx = Bx + d$, where B is a linear operator, and d is a fixed vector. Show that T is a strict contraction if and only if $||B||$, the induced matrix norm of B, is less than one.*

The spectral radius of B, written $\rho(B)$, is the maximum of $|\lambda|$, over all eigenvalues λ of B. Since $\rho(B) \leq ||B||$ for every norm on B induced by a vector norm, B is sc implies that $\rho(B) < 1$. When B is Hermitian, the matrix norm of B induced by the Euclidean vector norm is $||B||_2 = \rho(B)$, so if $\rho(B) < 1$, then B is sc with respect to the Euclidean norm.

When B is not Hermitian, it is not as easy to determine if the affine operator T is sc with respect to a given norm. Instead, we often tailor the norm to the operator T.

To illustrate, suppose that B is a diagonalizable matrix, that is, there is a basis for $\mathcal{X}$ consisting of eigenvectors of B. Let $\{u^1, ..., u^J\}$ be such a basis, and let $Bu^j = \lambda_j u^j$, for each $j = 1, ..., J$. For each x in $\mathcal{X}$, there are unique coefficients a_j so that

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

Then let

$$||x|| = \sum_{j=1}^{J} |a_j|. \tag{3.2}$$

Exercise 3.7 *Show that $||\cdot||$ defines a norm on $\mathcal{X}$.*

Exercise 3.8 *Suppose that $\rho(B) < 1$. Show that the affine operator T is sc, with respect to the norm defined by Equation (3.2).*

Actually, this result holds for any square matrix B, even if B is not diagonalizable. According to Lemma 35.1, for any square matrix B and any $\epsilon > 0$, there is a vector norm for which the induced matrix norm satisfies $||B|| \leq \rho(B) + \epsilon$.

In many of the applications of interest to us, there will be multiple fixed points of T. Therefore, T will not be sc for any vector norm, and the Banach-Picard fixed-point theorem will not apply. We need to consider other classes of operators.

The first class we consider are the *paracontractive* (pc) operators. This class is particularly important for the study of affine operators, since T being pc can be related to the behavior of the eigenvalues of B.

For the (possibly) non-affine case, we shall begin with operators that are *non-expansive* (ne) with respect to the Euclidean norm, and then focus on an important sub-class, the *averaged* operators.

3.3 Paracontractive Operators

An operator T on $\mathcal{X}$ is a *paracontraction* (pc), with respect to a vector norm $||\cdot||$, if, for every fixed point y of T, and every x, we have

$$||Tx - y|| < ||x - y||,$$

unless $Tx = x$. If T has no fixed points, then T is trivially pc. An operator T is *strictly non-expansive* (sne) if

$$||Tx - Ty|| < ||x - y||,$$

unless $Tx - Ty = x - y$. Clearly, if T is sc, then T is sne.

Exercise 3.9 *Show that, if T is sne, then T is pc.*

Exercise 3.10 *Let $H(a, \gamma) = \{x | \langle x, a\rangle = \gamma\}$. Show that P, the orthogonal projection onto $H(a, \gamma)$, is given by*

$$Px = x + \frac{\gamma - \langle x, a\rangle}{\langle a, a\rangle} a.$$

Then show that P is pc, but not sc, with respect to the Euclidean norm.

To illustrate, suppose, once again, that B is a diagonalizable matrix, that is, there is a basis for $\mathcal{X}$ consisting of eigenvectors of B. Let $\{u^1, ..., u^J\}$ be such a basis, and let $Bu^j = \lambda_j u^j$, for each $j = 1, ..., J$.

Exercise 3.11 *Suppose that $|\lambda_j| < 1$, for all eigenvalues λ_j that are not equal to one. Show that the affine operator T, given by $Tx = Bx + d$, is pc, with respect to the norm defined by Equation (3.2).*

Our interest in paracontractions is due to the Elsner/Koltracht/Neumann (EKN) Theorem [67]:

Theorem 3.2 *Let T be pc with respect to some vector norm. If T has fixed points, then the sequence $\{T^k x^0\}$ converges to a fixed point of T, for all starting vectors x^0.*

The product of two or more sne operators is again sne. The product of two or more pc operators will be pc if the operators share at least one fixed point, but not generally.

3.4 Averaged Non-expansive Operators

An operator T on $\mathcal{X}$ is *non-expansive* (ne), with respect to some vector norm, if, for every x and y, we have

$$||Tx - Ty|| \leq ||x - y||.$$

The identity map $Ix = x$ for all x is clearly ne; more generally, for any fixed vector w in $\mathcal{X}$, the maps $Nx = x + w$ and $Nx = -x + w$ are ne. If T is pc, then T is ne. Being ne is not enough to guarantee convergence of the iterative sequence $\{T^k x^0\}$, as the example $T = -I$ illustrates.

An operator T is *averaged* (av) if there is $\alpha \in (0, 1)$ and a non-expansive operator N, such that

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

where I is the identity operator. We also say that T is α-av.

Exercise 3.12 *Show that an av operator is ne.*

Although this defines the av operators for any vector norm, the notion of av operators is most useful in the context of the Euclidean norm, that is, the operator N in the definition is ne, with respect to the Euclidean norm. The main reason for this is the following identity, relating an operator T to its complement $G = I - T$, which holds only for the Euclidean norm:

$$||x - y||_2^2 - ||Tx - Ty||_2^2 = 2Re(\langle Gx - Gy, x - y\rangle) - ||Gx - Gy||_2^2. \quad (3.3)$$

Our interest in averaged operators is due to the Krasnoselskii/Mann Theorem [100]:

Theorem 3.3 *Let T be averaged, with respect to the Euclidean norm. If T has fixed points, then the iterative sequence $\{T^k x^0\}$ converges to a fixed point of T, for every starting vector, x^0.*

To make use of the KM Theorem, we shall assume, from now on, that all av operators are averaged with respect to the Euclidean norm.

The product of two or more av operators is again av, which makes the class of av operators important for the development of convergent iterative algorithms.

3.5 Projection onto Convex Sets

Let C be a nonempty, closed convex subset of $\mathcal{X}$. It is a basic result in Hilbert space theory that, for every x in $\mathcal{X}$, there is a unique point in C closest to x, in the Euclidean distance; this point is denoted $P_C x$ and the operator P_C is the orthogonal projection onto C. For most sets C we will not be able to describe $P_C x$ explicitly. We can, however, characterize $P_C x$ as the unique member of C for which

$$Re(\langle P_C x - x, c - P_C x\rangle) \geq 0, \tag{3.4}$$

for all c in C; see Proposition 34.2.

Exercise 3.13 *Show that the orthogonal projection operator $T = P_C$ is nonexpansive, with respect to the Euclidean norm. Hint: use Inequality (3.4) to get*

$$Re(\langle P_C y - P_C x, P_C x - x\rangle) \geq 0,$$

and

$$Re(\langle P_C x - P_C y, P_C y - y\rangle) \geq 0.$$

Add the two inequalities and use the Cauchy inequality.

In fact, this exercise shows that

$$Re(\langle P_C x - P_C y, x - y\rangle) \geq ||P_C x - P_C y||_2^2,$$

which says that the operator $T = P_C$ is not simply ne, but is *firmly nonexpansive* (fne). As we shall see later, being fne implies being av, so the P_C operators are av. If C_i, $i = 1, ..., I$ are convex sets, and P_i the orthogonal projection onto C_i, then the operator

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

is again av. When the intersection of the C_i is non-empty, the sequence $\{x^k\}$ will converge to a member of that intersection.

Proposition 3.1 *For any closed, convex set C, the operator P_C is pc, with respect to the Euclidean norm.*

Proof: It follows from Cauchy's Inequality that

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

with equality if and only if

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

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

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

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

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

This shows that the P_C operators are pc. ∎

3.6 Generalized Projections

So far, we have been discussing algorithms that apply to any vectors in $\mathcal{X}$. In a number of applications, the vectors of interest will naturally have non-negative entries. For such problems, it is reasonable to consider distances that apply only to non-negative vectors, such as the cross-entropy, or Kullback-Leibler, distance. Associated with such distances are generalized projections. Algorithms that are based on orthogonal projection operators can then be extended to employ these generalized projections. Of course, new proofs of convergence will be needed, but even there, aspects of earlier proofs are often helpful.

The orthogonal projection operators lead us to both the averaged operators and the paracontractive operators, as well as to generalized projections and Bregman paracontractions, and the algorithms built from them.

Chapter 4

Averaged Non-expansive Operators

Many well known algorithms in optimization, signal processing, and image reconstruction are iterative in nature. The Jacobi, Gauss-Seidel, and successive overrelaxation (SOR) procedures for solving large systems of linear equations, *projection onto convex sets* (POCS) methods and iterative optimization procedures, such as entropy and likelihood maximization, are the primary examples. The editorial [95] provides a brief introduction to many of the recent efforts in medical imaging. It is a pleasant fact that convergence of many of these algorithms is a consequence of the Krasnoselskii/Mann (KM) Theorem for averaged operators or the Elsner/Koltracht/Neumann (EKN) Theorem for paracontractions. In this chapter we take a closer look at averaged non-expansive operators and the Krasnoselskii/Mann Theorem. In the following chapter, we turn to paracontractive non-expansive operators and the results of Elsner, Koltracht and Neumann.

4.1 Convex Feasibility

Recall that an operator T on $\mathcal{X}$ is averaged (av) if there is an α in the interval $(0, 1)$ and an operator N, non-expansive with respect to the Euclidean norm, for which $T = (1 - \alpha)I + \alpha N$. For such T, the sequence $\{T^k x^0\}$ converges to a fixed point of T, whenever fixed points exist; this is the content of the KM Theorem.

To illustrate, suppose that C is a closed convex set in $\mathcal{X}$, such as the nonnegative vectors in R^J. The orthogonal projection operator P_C associates with every x in $\mathcal{X}$ the point $P_C x$ in C that is nearest to x, in the Euclidean distance. If C_1 and C_2 are two such sets the fixed points of the

operator $T = P_{C_2}P_{C_1}$ are the vectors in the intersection $C = C_1 \cap C_2$. Finding points in the intersection of convex sets is called the *convex feasibility problem* (CFP). If C is nonempty; then the sequence $\{x^k\}$ generated by Equation (3.1) converges to a member of C. This is a consequence of the KM Theorem, since the operator T is av.

4.2 Constrained Optimizaton

Some applications involve constrained optimization, in which we seek a vector x in a given convex set C that minimizes a certain function f. For suitable $\gamma > 0$ the operator $T = P_C(I - \gamma\nabla f)$ will be av and the sequence $\{T^k x^0\}$ will converge to a solution.

4.3 Solving Linear Systems

An important class of operators are the *affine linear* ones, having the form

$$Tx = Bx + h,$$

where B is linear, so that Bx is the multiplication of the vector x by the matrix B, and h is a fixed vector. Affine linear operators occur in iterative methods for solving linear systems of equations.

4.3.1 The Landweber Algorithm

The Landweber algorithm for solving the system $Ax = b$ is

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

where γ is a selected parameter. We can write the Landweber iteration as

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

for

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

The Landweber algorithm actually solves the square linear system $A^\dagger A = A^\dagger b$ for a least-squares solution of $Ax = b$. When there is a unique solution or unique least-squares solution of $Ax = b$, say $\hat{x}$, then the error at the k-th step is $e^k = \hat{x} - x^k$ and we see that

$$Be^k = e^{k+1}.$$

We want $e^k \to 0$, and so we want $||B||_2 < 1$; this means that both T and B are Euclidean strict contractions. Since B is Hermitian, B will be sc if

and only $||B||_2 < 1$, where $||B||_2 = \rho(B)$ is the matrix norm induced by the Euclidean vector norm.

On the other hand, when there are multiple solutions of $Ax = b$, the solution found by the Landweber algorithm will be the one closest to the starting vector. In this case, we cannot define e^k and we do not want $||B||_2 < 1$; that is, we do not need that B be a strict contraction, but something weaker. As we shall see, since B is Hermitian, B will be av whenever γ lies in the interval $(0, 2/\rho(B))$.

4.3.2 Splitting Algorithms

Affine linear operators also occur in splitting algorithms for solving a square system of linear equations, $Sx = b$. We write $S = M - K$, with M invertible. Then, the iteration is

$$x^{k+1} = M^{-1}Kx^k + M^{-1}b,$$

which can be written as

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

for the affine linear operator

$$Tx = M^{-1}Kx + M^{-1}b = Bx + h.$$

When S is invertible, there is a unique solution of $Sx = b$, say $\hat{x}$, and we can define the error $e^k = \hat{x} - x^k$. Then $e^{k+1} = Be^k$, and again we want $||B||_2 < 1$, that is, B is a strict contraction. However, if S is not invertible and there are multiple solutions, then we do not want B to be sc. Since B is usually not Hermitian, deciding if B is av may be difficult. Therefore, we may instead ask if there is a vector norm with respect to which B is pc.

We begin, in the next section, a detailed discussion of averaged operators, followed by an examination of the proof of the Krasnoselskii/Mann theorem.

4.4 Averaged Non-expansive Operators

As we have seen, the fact that a ne operator N has fixed points is not sufficient to guarantee convergence of the orbit sequence $\{N^k x^0\}$; additional conditions are needed. Requiring the operator to be a strict contraction is quite restrictive; most of the operators we are interested in here have multiple fixed points, so are not sc, in any norm. For example, if $T = P_C$, then $C = \text{Fix}(T)$. Motivated by the KM Theorem, we concentrate on averaged operators, by which we shall always mean with respect to the Euclidean norm.

4.4.1 Properties of Averaged Operators

As we shall see now, in seeking fixed points for an operator T it is helpful to consider properties of its complement, $G = I - T$. An operator G on $\mathcal{X}$ is called *ν-inverse strongly monotone* (ν-ism) [73] (also called *co-coercive* in [52]) if there is $\nu > 0$ such that

$$Re(\langle Gx - Gy, x - y\rangle) \geq \nu ||Gx - Gy||_2^2.$$

Exercise 4.1 *Show that N is ne if and only if its complement $G = I - N$ is $\frac{1}{2}$-ism. If G is ν-ism and $\gamma > 0$ then the operator γG is $\frac{\nu}{\gamma}$-ism.*

Lemma 4.1 *An operator A is av if and only if its complement $G = I - A$ is ν-ism for some $\nu > \frac{1}{2}$.*

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

Exercise 4.2 *Show that, if the operator A is α-av and $1 > \beta > \alpha$, then A is β-av.*

Exercise 4.3 *Note that we can establish that a given operator is av by showing that there is an α in the interval $(0, 1)$ such that the operator*

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

is ne. Use this approach to show that if T is sc, then T is av.

Lemma 4.2 *Let $T = (1 - \alpha)A + \alpha N$ for some $\alpha \in (0, 1)$. If A is averaged and N is non-expansive then T is averaged.*

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

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

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

Corollary 4.1 *If A and B are av and α is in the interval $[0, 1]$, then the operator $T = (1 - \alpha)A + \alpha B$ formed by taking the convex combination of A and B is av.*

An operator F on $\mathcal{X}$ is called *firmly non-expansive* (fne), with respect to the Euclidean norm, if it is 1-ism [124], [8].

Lemma 4.3 *An operator F is fne if and only if its complement $I - F$ is fne. If F is fne then F is av.*

Proof: By Equation (34.4), we know that, for any operator F with $G = I - F$, we have

$$Re(\langle Fx - Fy, x - y\rangle) - ||Fx - Fy||_2^2 = Re(\langle Gx - Gy, x - y\rangle) - ||Gx - Gy||_2^2.$$

The left side is nonnegative if and only if the right side is. Finally, if F is fne then $I - F$ is fne, so $I - F$ is ν-ism for $\nu = 1$. Therefore F is av by Lemma 4.1. ∎

Corollary 4.2 *Let $T = (1 - \alpha)F + \alpha N$ for some $\alpha \in (0, 1)$. If F is fne and N is Euclidean-ne then T is averaged.*

Proposition 4.1 *For any closed, convex set C, the operator P_C is fne, and, therefore, is av.*

Proof: Since the orthogonal projection of x onto C is characterized by the inequalities

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

for all $c \in C$, we have

$$Re(\langle P_C y - P_C x, P_C x - x\rangle) \geq 0$$

and

$$Re(\langle P_C x - P_C y, P_C y - y\rangle) \geq 0.$$

Adding, we find that

$$Re(\langle P_C x - P_C y, x - y\rangle) \geq ||P_C x - P_C y||_2^2;$$

the operator P_C is fne, and therefore also av.

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

$$T = (1 - \omega)I + \omega P_H,$$

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

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

for each x.

Exercise 4.4 *Show that R_H is an isometry; that is,*

$$||R_H x - R_H y||_2 = ||x - y||_2,$$

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

Exercise 4.5 *Show that, for $\omega = 1 + \gamma$ in the interval $[1, 2)$, we have*

$$(1 - \omega)I + \omega P_H = \alpha I + (1 - \alpha)R_H,$$

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

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

If $A = (1 - \alpha)I + \alpha N$ is averaged and B is averaged then $T = AB$ has the form $T = (1 - \alpha)B + \alpha NB$. Since B is av and NB is ne, it follows from Lemma 4.1 that T is averaged. Summarizing, we have

Proposition 4.2 *If A and B are averaged, then $T = AB$ is averaged.*

It is possible for $\mathrm{Fix}(AB)$ to be nonempty while $\mathrm{Fix}(A) \cap \mathrm{Fix}(B)$ is empty; however, if the latter is nonempty, it must coincide with $\mathrm{Fix}(AB)$ [8]:

Proposition 4.3 *Let A and B be averaged operators and suppose that Fix(A)∩Fix(B) is nonempty. Then Fix(A)∩Fix(B) =Fix(AB)=Fix(BA).*

Proof: Let $I - A$ be ν_A-ism and $I - B$ be ν_B-ism, where both ν_A and ν_B are taken greater than $\frac{1}{2}$. Let z be in $\mathrm{Fix}(A) \cap \mathrm{Fix}(B)$ and x in $\mathrm{Fix}(BA)$. Then

$$||z - x||_2^2 \geq ||z - Ax||_2^2 + (2\nu_A - 1)||Ax - x||_2^2$$

$$\geq ||z - BAx||_2^2 + (2\nu_B - 1)||BAx - Ax||_2^2 + (2\nu_A - 1)||Ax - x||_2^2$$

$$= ||z - x||_2^2 + (2\nu_B - 1)||BAx - Ax||_2^2 + (2\nu_A - 1)||Ax - x||_2^2.$$

Therefore $||Ax - x||_2 = 0$ and $||BAx - Ax||_2 = ||Bx - x||_2 = 0$. ∎

4.4.2 Averaged Linear Operators

Affine linear operators have the form $Tx = Bx + d$, where B is a matrix. The operator T is av if and only if B is av. It is useful, then, to consider conditions under which B is av.

When B is averaged, there is a positive α in $(0, 1)$ and a Euclidean ne operator N, with

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

Therefore

$$N = \frac{1}{\alpha}B + (1 - \frac{1}{\alpha})I \tag{4.1}$$

is non-expansive. Clearly, N is a linear operator; that is, N is multiplication by a matrix, which we also denote N. When is such an operator N ne?

Exercise 4.6 *Show that a linear operator N is ne, in the Euclidean norm, if and only if $||N||_2 = \sqrt{\rho(N^\dagger N)}$, the matrix norm induced by the Euclidean vector norm, does not exceed one.*

We know that B is av if and only if its complement, $I - B$, is ν-ism for some $\nu > \frac{1}{2}$. Therefore,

$$Re(\langle (I - B)x, x\rangle) \geq \nu ||(I - B)x||_2^2,$$

for all x. This implies that $x^\dagger(I - B)x \geq 0$, for all x. Since this quadratic form can be written as

$$x^\dagger(I - B)x = x^\dagger(I - Q)x,$$

for $Q = \frac{1}{2}(B + B^\dagger)$, it follows that $I - Q$ must be non-negative definite. Moreover, if B is av, then B is ne, so that $||B||_2 \leq 1$. Since $||B||_2 = ||B^\dagger||_2$, and $||Q||_2 \leq \frac{1}{2}(||B||_2 + ||B^\dagger||_2)$, it follows that Q must be Euclidean ne. In fact, since N is Euclidean ne if and only if $N^\dagger$ is, B is av if and only if $B^\dagger$ is av. Consequently, if the linear operator B is av, then so is the Hermitian operator Q, and so the eigenvalues of Q must lie in the interval $(-1, 1]$. We also know from Exercise **??** that, if B is av, then $|\lambda| < 1$, unless $\lambda = 1$, for every eigenvalue λ of B.

In later chapters we shall be particularly interested in linear operators B that are Hermitian, in which case N will also be Hermitian. Therefore, we shall assume, for the remainder of this subsection, that B is Hermitian, so that all of its eigenvalues are real. It follows from our discussion relating matrix norms to spectral radii that a Hermitian N is ne if and only if $\rho(N) \leq 1$. We now derive conditions on the eigenvalues of B that are equivalent to B being an av linear operator.

For any (necessarily real) eigenvalue λ of B, the corresponding eigenvalue of N is

$$\nu = \frac{1}{\alpha}\lambda + (1 - \frac{1}{\alpha}).$$

Exercise 4.7 *Show that $|\nu| \leq 1$ if and only if*

$$1 - 2\alpha \leq \lambda \leq 1.$$

From the exercise, we see that the Hermitian linear operator B is av if and only if there is α in $(0, 1)$ such that

$$-1 < 1 - 2\alpha \leq \lambda \leq 1,$$

for all eigenvalues λ of B. This is equivalent to saying that

$$-1 < \lambda \leq 1,$$

for all eigenvalues λ of B. The choice

$$\alpha_0 = \frac{1 - \lambda_{min}}{2}$$

is the smallest α for which

$$N = \frac{1}{\alpha}B + (1 - \frac{1}{\alpha})I$$

will be non-expansive; here λ_{min} denotes the smallest eigenvalue of B. So, α_0 is the smallest α for which B is α-av.

The linear operator B will be fne if and only if it is $\frac{1}{2}$-av. Therefore, B will be fne if and only if $0 \leq \lambda \leq 1$, for all eigenvalues λ of B. Since B is Hermitian, we can say that B is fne if and only if B and $I - B$ are non-negative definite. We summarize the situation for Hermitian B as follows. Let λ be any eigenvalue of B. Then

B is non-expansive if and only if $-1 \leq \lambda \leq 1$, for all λ;

B is averaged if and only if $-1 < \lambda \leq 1$, for all λ;

B is a strict contraction if and only if $-1 < \lambda < 1$, for all λ;

B is firmly non-expansive if and only if $0 \leq \lambda \leq 1$, for all λ.

4.5 The KM Theorem

The Krasnoselskii/Mann Theorem is the following:

Theorem 4.1 *Let T be an av operator on $\mathcal{X}$ and let Fix(T) be nonempty. Then the orbit sequence $\{T^k x\}$ converges to a member of Fix(T), for any x.*

As we shall see, many of the iterative methods used in signal and image processing are special cases of the KM approach.

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

$$x^{k+1} = Tx^k = (1-\alpha)x^k + \alpha N x^k. \tag{4.2}$$

The identity in Equation (34.3) is the key to proving Theorem 4.1.

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

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

Since, by Lemma 4.1, G is $\frac{1}{2\alpha}$-ism, we have

$$||z - x^k||_2^2 - ||z - x^{k+1}||_2^2 \geq (\frac{1}{\alpha} - 1)||x^k - x^{k+1}||_2^2. \tag{4.3}$$

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

For those cases in which N is the operator of interest, and we form T only to apply the KM Theorem, it might appear that Equation 4.3 is telling us to select α small, so as to make the term $\frac{1}{\alpha} - 1$, and therefore, the left side, quite large. However, a small α will tend to make $||x^* - x^k||_2$ small as well. Selecting the best α is not a simple matter.

As we outlined in the Introduction, a wide variety of operators T can be shown to be av. The convergence of the iterative fixed-point algorithms associated with these operators then follows as a consequence of this theorem.

4.6 The De Pierro-Iusem Approach

As we have seen, the class of non-expansive operators is too broad, and the class of strict contractions too narrow, for our purposes. The KM Theorem encourages us to focus on the intermediate class of averaged operators. While this is certainly a fruitful approach, it is not the only possible one. In [60] De Pierro and Iusem take a somewhat different approach, basing their class of operators on properties of orthogonal projections onto convex sets.

Exercise 4.8 *Use the Cauchy-Schwarz Inequality and the fact that $T = P_C$ is firmly non-expansive to show that*

$$||Tx - Ty||_2 = ||x - y||_2 \tag{4.4}$$

implies that

$$Tx - Ty = x - y, \tag{4.5}$$

and

$$\langle Tx - x, x - y \rangle = 0. \tag{4.6}$$

De Pierro and Iusem consider operators $Q : R^J \to R^J$ that are non-expansive and for which the property in Equation (4.4) implies both Equations (4.5) and (4.6). They then show that this class is closed to finite products and convex combinations.

Chapter 5

Paracontractive Operators

An affine linear operator $Tx = Bx + d$ is an averaged non-expansive operator if and only if its linear part, B, is also averaged. A Hermitian B is av if and only if $-1 < \lambda \leq 1$, for each eigenvalue λ of B. When B is not Hermitian, deciding if B is av is harder. In such cases, we can ask if there is some vector norm, with respect to which B is paracontractive (pc). As we shall see, if B is diagonalizable, then B is pc if $|\lambda| < 1$, for every eigenvalue λ of B that is not equal to one. Then we can use the results of Elsner, Koltracht and Neumann to establish convergence of the iterative algorithm given by Equation (3.1).

5.1 Paracontractions and Convex Feasibility

An operator T on $\mathcal{X}$ is *paracontractive* (pc), with respect to some vector norm $||\cdot||$, if, for every fixed point y of T and for every x, we have

$$||Tx - y|| < ||x - y||,$$

unless $Tx = x$. Note that T can be pc without being continuous, hence without being ne. We shall restrict our attention here to those pc operators that are continuous.

Let C_i, $i = 1, ..., I$, be non-empty, closed convex sets in $\mathcal{X}$, with non-empty intersection C. The orthogonal projection $P_i = P_{C_i}$ onto C_i is pc, with respect to the Euclidean norm, for each i. The product $T = P_I P_{I-1} \cdots P_1$ is also pc, since C is non-empty. The SOP algorithm converges to a member of C, for any starting vector x^0, as a consequence of the EKN Theorem. For the SOP to be a practical procedure, we need to be able to calculate easily the orthogonal projection onto each C_i. The *cyclic subgradient projection* method (CSP) (see [45]) provides a practical

alternative to the SOP, for sets C_i of the form

$$C_i = \{x | g_i(x) \leq b_i\},$$

where g_i is a convex function on $\mathcal{X}$. In the case in which g is differentiable, for each i, let

$$T_i x = x - \omega \alpha_i(x) \nabla g_i(x),$$

for

$$\alpha_i(x) = \max(g_i(x) - b_i, 0)/||\nabla g_i(x)||^2.$$

From [67] we have

Theorem 5.1 *For $0 < \omega < 2$, the operators T_i are pc, with respect to the Euclidean norm.*

Proof: A vector y is a fixed point of T_i if and only if $g_i(y) \leq 0$, so if and only if $y \in C_i$. Let x be a vector outside of C_i, and let $\alpha = \alpha_i(x)$. Since g_i has no relative minimum outside of C_i, $T_i x$ is well defined. We want to show that $||T_i x - y|| < ||x - y||$. This is equivalent to showing that

$$\omega^2 \alpha^2 ||\nabla g_i(x)||^2 \leq 2\omega\alpha \langle \nabla g_i(x), x - y \rangle,$$

which, in turn, is equivalent to showing that

$$\omega(g_i(x) - b_i) \leq \langle \nabla g_i(x), x - y \rangle. \tag{5.1}$$

Since $g_i(y) \leq b_i$ and g_i is convex, we have

$$(g_i(x) - \beta) \leq (g_i(x) - g_i(y)) \leq \langle \nabla g_i(x), x - y \rangle.$$

Inequality (5.1) follows immediately. ∎

The CSP algorithm has the iterative step

$$x^{k+1} = T_{i(k)} x^k,$$

where $i(k) = k(\mathrm{mod}\, I)+1$. Since each of the operators T_i is pc, the sequence converges to a member of C, whenever C is non-empty, as a consequence of the EKN Theorem.

Let A be an I by J real matrix, and for each i let $g_i(x) = (Ax)_i$. Then the gradient of g_i is $\nabla g_i(x) = a^i$, the ith column of A^T. The set C_i is the half-space $C = \{x | \, (Ax)_i \leq b_i\}$, and the operator T_i is the orthogonal projection onto C_i. The CSP algorithm in this case becomes the AMS algorithm for finding x with $Ax \leq b$.

5.2 The EKN Theorem

We have the Elsner/Koltracht/Neumann Theorem and its corollaries from [67]:

Theorem 5.2 *Suppose that there is a vector norm on $\mathcal{X}$, with respect to which each T_i is a pc operator, for $i = 1, ..., I$, and that $F = \cap_{i=1}^{I} \mathrm{Fix}(T_i)$ is not empty. For $k = 0, 1, ...$, let $i(k) = k(\mathrm{mod}\, I) + 1$, and $x^{k+1} = T_{i(k)}x^k$. The sequence $\{x^k\}$ converges to a member of F, for every starting vector x^0.*

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

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

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

$$||T_i x^* - y|| = ||x^* - y||,$$

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

Corollary 5.1 *If T is pc with respect to some vector norm, and T has fixed points, then the iterative sequence $\{x^k\}$ generated by Equation (3.1) converges to a fixed point of T, for every starting vector x^0.*

Corollary 5.2 *If $T = T_I T_{I-1} \cdots T_2 T_1$, and $F = \cap_{i=1}^{I} \mathrm{Fix}\,(T_i)$ is not empty, then $F = \mathrm{Fix}\,(T)$.*

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

Corollary 5.3 *The product T of two or more pc operators T_i, $i = 1, ..., I$ is again a pc operator, if $F = \cap_{i=1}^{I} \mathrm{Fix}\,(T_i)$ is not empty.*

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

$$||Tx - y|| = ||x - y||.$$

Then, since

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

it follows that

$$||T_i x - y|| = ||x - y||,$$

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

5.3 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.

5.3.1 Back-propagation-of-error Methods

Suppose that A is I by J, with $J > I$ and that $Ax = b$ has infinitely many solutions. A *backpropagation-of-error* approach leads to an algorithm with the iterative step

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

where C is some I by J matrix. The algorithm can then be written in the form $x^{k+1} = T^k x^0$, for T the affine operator given by

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

Since $Ax = b$ has multiple solutions, A has a non-trivial null space, so that some of the eigenvalues of $B = (I - \gamma C^\dagger A)$ are equal to one. As we shall see, if γ is chosen so that $|\lambda| < 1$, for all the remaining eigenvalues of B, and B is diagonalizable, then T will be pc, with respect to some vector norm, and the iterative sequence $\{x^k\}$ will converge to a solution. For such a γ to exist, it is necessary that, for all nonzero eigenvalues $\mu = a + bi$ of the matrix $C^\dagger A$, the real parts a be nonzero and have the same sign, which we may, without loss of generality, assume to be positive. Then we need to select γ in the intersection of the intervals $(0, 2a/(a^2+b^2))$, taken over every eigenvalue μ. When $C = A$, all the nonzero eigenvalues of $C^\dagger A = A^\dagger A$ are positive, so such a γ exists. As C deviates from A, the eigenvalues of $C^\dagger A$ begin to change. We are asking that the C not deviate from A enough to cause the real part of an eigenvalue to become negative.

5.3.2 Defining the Norm

Suppose that $Tx = Bx + d$ 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

$$||x|| = \sum_{j=1}^{J} |a_j|,$$

We know from a previous exercise that T is pc with respect to this norm. It follows from Theorem 3.2 that the iterative sequence $\{x^k\}$ will converge to a fixed point of T, whenever T has fixed points.

5.3.3 Proof of Convergence

It is not difficult to prove convergence directly, as we now show.

Proof of convergence: Let the eigenvalues of B be λ_j, for $j = 1, ..., J$, with associated linearly independent eigenvectors u^j. Define a norm on vectors x by

$$||x|| = \sum_{j=1}^{J} |a_j|,$$

for

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

Assume that $\lambda_j = 1$, for $j = K+1, ..., J$, and that $|\lambda_j| < 1$, for $j = 1, ..., K$. Let

$$d = \sum_{j=1}^{J} d_j u^j.$$

Let $\hat{x}$ be an arbitrary fixed point of T, with

$$\hat{x} = \sum_{j=1}^{J} \hat{a}_j u^j.$$

From $T\hat{x} = \hat{x}$ we have

$$\sum_{j=1}^{J} \hat{a}_j u^j = \sum_{j=1}^{J} (\lambda_j \hat{a}_j + d_j) u^j.$$

Then with

$$x^k = \sum_{j=1}^{J} a_{jk} u^j,$$

and

$$x^{k+1} = Bx^k + h = \sum_{j=1}^{J} (\lambda_j a_{jk} + d_j) u^j,$$

we have

$$x^k - \hat{x} = \sum_{j=1}^{J} (a_{jk} - \hat{a}_j) u^j,$$

and

$$x^{k+1} - \hat{x} = \sum_{j=1}^{K} \lambda_j (a_{jk} - \hat{a}_j) u^j + \sum_{j=K+1}^{J} (a_{jk} - \hat{a}_j) u^j.$$

Therefore,

$$||x^k - \hat{x}|| = \sum_{j=1}^{K} |a_{jk} - \hat{a}| + \sum_{j=K+1}^{J} |a_{jk} - \hat{a}_j|,$$

while

$$||x^{k+1} - \hat{x}|| = \sum_{j=1}^{K} |\lambda_j||a_{jk} - \hat{a}| + \sum_{j=K+1}^{J} |a_{jk} - \hat{a}_j|.$$

Consequently,

$$||x^k - \hat{x}|| - ||x^{k+1} - \hat{x}|| = \sum_{j=1}^{K} (1 - |\lambda_j|)|a_{jk} - \hat{a}_j|.$$

It follows that the sequence $\{||x^k - \hat{x}||\}$ is decreasing, and that the sequences $\{|a_{jk} - \hat{a}_j|\}$ converge to zero, for each $j = 1, ..., K$.

Since the sequence $\{x^k\}$ is then bounded, select a cluster point, x^*, with

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

Then we must have

$$\{|a_{jk} - a_j^*|\} \to 0,$$

for $j = 1, ..., K$. It follows that $\hat{a}_j = a_j^*$, for $j = 1, ..., K$. Therefore,

$$\hat{x} - x^* = \sum_{j=K+1}^{J} c_j u^j,$$

for $c_j = \hat{a}_j - a_j^*$. We can conclude, therefore, that

$$\hat{x} - B\hat{x} = x^* - Bx^*,$$

so that x^* is another solution of the system $(I - B)x = d$. Therefore, the sequence $\{||x^k - x^*||\}$ is decreasing; but a subsequence converges to zero, so the entire sequence must converge to zero. We conclude that $\{x^k\}$ converges to the solution x^*. ∎

It is worth noting that the condition that B be diagonalizable cannot be omitted. Consider the non-diagonalizable matrix

$$B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},$$

and the affine operator

$$Tx = Bx + (1, 0)^T.$$

The fixed points of T are the solutions of $(I - B)x = (1, 0)^T$, which are the vectors of the form $x = (a, -1)^T$. With starting vector $x^0 = (1, 0)^T$, we find that $x^k = (k - 1)x^0$, so that the sequence $\{x^k\}$ does not converge to a fixed point of T. There is no vector norm for which T is pc.

If T is an affine linear operator with diagonalizable linear part, then T is pc whenever T is av, as we know from Exercise **??**. We see from that 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.

Chapter 6

Bregman-Paracontractive Operators

In the previous chapter, we considered operators that are paracontractive, with respect to some norm. In this chapter, we extend that discussion to operators that are paracontractive, with respect to some Bregman distance. Our objective here is to examine the extent to which the EKN Theorem and its consequences can be extended to the broader class of Bregman paracontractions. Typically, these operators are not defined on all of $\mathcal{X}$, but on a restricted subset, such as the non-negative vectors, in the case of entropy. For details concerning Bregman distances and related notions, see the appendix.

6.1 Bregman Paracontractions

Let f be a closed proper convex function that is differentiable on the nonempty set int$D$. The corresponding *Bregman distance* $D_f(x, z)$ is defined for $x \in R^J$ and $z \in \text{int}D$ by

$$D_f(x, z) = f(x) - f(z) - \langle \nabla f(z), x - z \rangle,$$

where $D = \{x \,|f(x) < +\infty\}$ is the essential domain of f. When the domain of f is not all of R^J, we define $f(x) = +\infty$, for x outside its domain. Note that $D_f(x, z) \geq 0$ always and that $D_f(x, z) = +\infty$ is possible. If f is essentially strictly convex then $D_f(x, z) = 0$ implies that $x = z$.

Let C be a nonempty closed convex set with $C \cap \text{int}D \neq \emptyset$. Pick $z \in \text{int}D$. The *Bregman projection* of z onto C, with respect to f, is

$$P_C^f(z) = \text{argmin}_{x \in C \cap D} D_f(x, z).$$

If f is essentially strictly convex, then $P_C^f(z)$ exists. If f is strictly convex on D then $P_C^f(z)$ is unique. We assume that f is Legendre, so that $P_C^f(z)$ is uniquely defined and is in int$D$; this last condition is sometimes called *zone consistency*.

We shall make much use of the *Bregman Inequality* (37.1):

$$D_f(c,z) \geq D_f(c,P_C^f z) + D_f(P_C^f z, z). \tag{6.1}$$

A continuous operator $T : \text{int}D \to \text{int}D$ is called a *Bregman paracontraction* (bpc) if, for every fixed point z of T, and for every x, we have

$$D_f(z,Tx) < D_f(z,x),$$

unless $Tx = x$. In order for the Bregman distances $D_f(z,x)$ and $D_f(z,Tx)$ to be defined, it is necessary that $\nabla f(x)$ and $\nabla f(Tx)$ be defined, and so we need to restrict the domain and range of T in the manner above. This can sometimes pose a problem, when the iterative sequence $\{x^{k+1} = Tx^k\}$ converges to a point on the boundary of the domain of f. This happens, for example, in the EMML and SMART methods, in which each x^k is a positive vector, but the limit can have entries that are zero. One way around this problem is to extend the notion of a fixed point: say that z is an *asymptotic fixed point* of T if (z,z) is in the closure of the graph of T, that is, (z,z) is the limit of points of the form (x,Tx). Theorems for iterative methods involving Bregman paracontractions can then be formulated to involve convergence to an asymptotic fixed point [27]. In our discussion here, however, we shall not consider this more general situation.

6.1.1 Entropic Projections

As an example of a Bregman distance and Bregman paracontractions, consider the function $g(t) = t\log(t) - t$, with $g(0) = 0$, and the associated Bregman-Legendre function

$$f(x) = \sum_{j=1}^{J} g(x_j),$$

defined for vectors x in the non-negative cone R_+^J. The corresponding Bregman distance is the Kullback-Leibler, or cross-entropy, distance

$$D_f(x,z) = f(x) - f(z) - \langle \nabla f(z), x - z\rangle = KL(x,z).$$

For any non-empty, closed, convex set C, the *entropic projection* operator P_C^e is defined by $P_C^e z$ is the member x of $C \cap R_+^J$ for which $KL(x,z)$ is minimized.

Theorem 6.1 *The operator $T = P_C^e$ is bpc, with respect to the cross-entropy distance.*

Proof: The fixed points of $T = P_C^e$ are the vectors c in $C \cap R_+^J$. From the Bregman Inequality (6.1) we have

$$D_f(c, x) - D_f(c, P_C^e x) \geq D_f(P_C^e x, x) \geq 0,$$

with equality if and only if $D_f(P_C^e x, x) = 0$, in which case $Tx = x$. ∎

6.1.2 Weighted Entropic Projections

Generally, we cannot exhibit the entropic projection onto a closed, convex set C in closed form. When we consider the EMML and SMART algorithms, we shall focus on non-negative systems $Ax = b$, in which the entries of A are non-negative, those of b are positive, and we seek a non-negative solution. For each $i = 1, ..., I$, let

$$H_i = \{x \geq 0 | (Ax)_i = b_i\}.$$

We cannot write the entropic projection of z onto H_i in closed form, but, for each positive vector z, the member of H_i that minimizes the weighted cross-entropy,

$$\sum_{j=1}^{J} A_{ij} KL(x_j, z_j) \tag{6.2}$$

is

$$x_j = (Q_i^e z)_j = z_j \frac{b_i}{(Az)_i}.$$

Exercise 6.1 *Show that the operator Q_i^e is bpc, with respect to the Bregman distance in Equation (6.2). Hint: show that, for each x in H_i,*

$$\sum_{j=1}^{J} A_{ij} KL(x_j, z_j) - \sum_{j=1}^{J} A_{ij} KL(x_j, (Q_i^e z)_j) = KL(b_i, (Az)_i).$$

With $\sum_{i=1}^{I} A_{ij} = 1$, for each j, the iterative step of the EMML algorithm can be written as $x^{k+1} = Tx^k$, for

$$(Tx)_j = \sum_{i=1}^{I} A_{ij} (Q_i^e x)_j,$$

and that of the SMART is $x^{k+1} = Tx^k$, for

$$(Tx)_j = \prod_{i=1}^{I} [(Q_i^e x)_j]^{A_{ij}}.$$

It follows from the theory of these two algorithms that, in both cases, T is bpc, with respect to the cross-entropy distance.

6.2 Extending the EKN Theorem

Now we present a generalization of the EKN Theorem.

Theorem 6.2 *For $i = 1, ..., I$, let T_i be bpc, for the Bregman distance D_f. Let $F = \cap_{i=1}^{I} \text{Fix}(T_i)$ be non-empty. Let $i(k) = k(\text{mod}\, I) + 1$ and $x^{k+1} = T_{i(k)}x^k$. Then the sequence $\{x^k\}$ converges to a member of F.*

Proof: Let z be a member of F. We know that

$$D_f(z, x^k) - D_f(z, x^{k+1}) \geq 0,$$

so that the sequence $\{D_f(z, x^k\}$ is decreasing, with limit $d \geq 0$. Then the sequence $\{x_k\}$ is bounded; select a cluster point, x^*. Then T_1x^* is also a cluster point, so we have

$$D_f(z, x) - D_f(z, T_1x) = 0,$$

from which we conclude that $T_1x = x$. Similarly, $T_2T_1x^* = T_2x^*$ is a cluster point, and $T_2x^* = x^*$. Continuing in this manner, we show that x^* is in F. Then $\{D_f(x^*, x^k)\} \to 0$, so that $\{x^k\} \to x^*$. ∎

We have the following generalization of Corollary 5.3:

Corollary 6.1 *For $i = 1, ..., I$, let T_i be bpc, for the Bregman distance D_f. Let $F = \cap_{i=1}^{I} \text{Fix}(T_i)$ be non-empty. Let $T = T_I T_{I-1} \cdots T_2 T_1$. Then the sequence $\{x^{k+1} = Tx^k\}$ converges to a member of F.*

Proof: Let z be in F. Since $D_f(z, T_ix) \leq D_f(z, x)$, for each i, it follows that

$$D_f(z, x) - D_f(z, Tx) \geq 0.$$

If equality holds, then

$$D_f(z, (T_I T_{I-1} \cdots T_1)x) = D_f(z, (T_{I-1} \cdots T_1)x)$$

$$... = D_f(z, T_1x) = D_f(z, x),$$

from which we can conclude that $T_ix = x$, for each i. Therefore, $Tx = x$, and T is bpc. ∎

Corollary 6.2 *If F is not empty, then $F = \text{Fix}(T)$.*

Exercise 6.2 *Prove this corollary.*

6.3 Multiple Bregman Distances

We saw earlier that both the EMML and the SMART algorithms involve Bregman projections with respect to distances that vary with the sets $C_i = H_i$. This suggests that Theorem 6.2 could be extended to include continuous operators T_i that are bpc, with respect to Bregman distances D_{f_i} that vary with i. However, there is a counter-example in [32] that shows that the sequence $\{x^{k+1} = T_{i(k)}x^k\}$ need not converge to a fixed point of T. The problem is that we need some Bregman distance D_h that is independent of i, with $\{D_h(z, x^k\}$ decreasing. The result we present now is closely related to the MSGP algorithm.

6.3.1 Assumptions and Notation

We make the following assumptions throughout this section. The function h is super-coercive and Bregman-Legendre with essential domain $D = \operatorname{dom} h$. For $i = 1, 2, ..., I$ the function f_i is also Bregman-Legendre, with $D \subseteq \operatorname{dom} f_i$, so that $\operatorname{int} D \subseteq \operatorname{int} \operatorname{dom} f_i$. For all $x \in \operatorname{dom} h$ and $z \in \operatorname{int} \operatorname{dom} h$ we have $D_h(x, z) \geq D_{f_i}(x, z)$, for each i.

6.3.2 The Algorithm

The *multi-distance* extension of Theorem 6.2 concerns the algorithm with the following iterative step:

$$x^{k+1} = \nabla h^{-1}\Big(\nabla h(x^k) - \nabla f_{i(k)}(x^k) + \nabla f_{i(k)}(T_{i(k)}(x^k))\Big). \tag{6.3}$$

6.3.3 A Preliminary Result

For each $k = 0, 1, ...$ define the function $G^k(\cdot) : \operatorname{dom} h \to [0, +\infty)$ by

$$G^k(x) = D_h(x, x^k) - D_{f_{i(k)}}(x, x^k) + D_{f_{i(k)}}(x, T_{i(k)}(x^k)). \tag{6.4}$$

The next proposition provides a useful identity, which can be viewed as an analogue of Pythagoras' theorem. The proof is not difficult and we omit it.

Proposition 6.1 *For each $x \in \operatorname{dom} h$, each $k = 0, 1, ...$, and x^{k+1} given by Equation (6.3) we have*

$$G^k(x) = G^k(x^{k+1}) + D_h(x, x^{k+1}). \tag{6.5}$$

Consequently, x^{k+1} is the unique minimizer of the function $G^k(\cdot)$.

This identity (6.5) is the key ingredient in the proof of convergence of the algorithm.

6.3.4 Convergence of the Algorithm

We shall prove the following convergence theorem:

Theorem 6.3 *Let F be non-empty. Let $x^0 \in \operatorname{int}\operatorname{dom} h$ be arbitrary. Any sequence x^k obtained from the iterative scheme given by Equation (6.3) converges to $x^\infty \in F \cap \operatorname{dom} h$.*

Proof: Let z be in F. Then it can be shown that

$$D_h(z, x^k) - D_h(z, x^{k+1}) = G^k(x^{k+1}) + D_{f_i}(z, x^k) - D_{f_i}(z, T_{i(k)}x^k).$$

Therefore, the sequence $\{D_h(z, x^k)\}$ is decreasing, and the non-negative sequences $\{G^k(x^{k+1})\}$ and $\{D_{f_i}(z, x^k) - D_{f_i}(z, T_{i(k)}x^k)\}$ converge to zero. The sequence $\{x^{mI}\}$ is then bounded and we can select a subsequence $\{x^{m_nI}\}$ with limit point $x^{*,0}$. Since the sequence $\{x^{m_nI+1}\}$ is bounded, it has a subsequence with limit $x^{*,1}$. But, since

$$D_{f_1}(z, x^{m_nI}) - D_{f_1}(z, x^{m_nI+1}) \to 0,$$

we conclude that $T_1x^{*,0} = x^{*,0}$. Continuing in this way, we eventually establish that $T_ix^{*,0} = x^{*,0}$, for each i. So, $x^{*,0}$ is in F. Using $x^{*,0}$ in place of z, we find that $\{D_h(x^{*,0}, x^k)\}$ is decreasing; but a subsequence converges to zero, so the entire sequence converges to zero, and $\{x^k\} \to x^{*,0}$. ∎

Part III

Systems of Linear Equations

Chapter 7

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.

7.1 The Algebraic Reconstruction Technique (ART)

The *algebraic reconstruction technique* (ART) applies to an arbitrary system $Ax = b$ of linear equations [74, 81, 88]. For an arbitrary starting point x^0 and $i = k(\mathrm{mod}\, I) + 1$, 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

$$x_j^{k+1} = x_j^k + \overline{A_{ij}} (b_i - (Ax^k)_i). \tag{7.1}$$

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.

7.1.1 Relaxed ART

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

$$x_j^{k+1} = x_j^k + \omega \overline{A_{ij}}(b_i - (Ax^k)_i)). \tag{7.2}$$

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.

7.1.2 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

$$x_j^{k+1} = P_C(x_j^k + \overline{A_{ij}}(b_i - (Ax^k)_i)). \tag{7.3}$$

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

$$x_j^{k+1} = (x_j^k + A_{ij}(b_i - (Ax^k)_i))_+, \tag{7.4}$$

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.

7.1.3 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 [34], while the second one is due to Eggermont, Herman, and Lent [66].

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

7.2 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 [48]. The iterative step of Cimmino's algorithm is

$$x_j^{k+1} = x_j^k + \frac{1}{I} \sum_{i=1}^{I} \overline{A_{ij}} (b_i - (Ax^k)_i), \tag{7.5}$$

which can be written as

$$x^{k+1} = x^k + \frac{1}{I} A^\dagger (b - Ax^k). \tag{7.6}$$

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.

7.3 Landweber's Algorithm

Landweber's algorithm [92] has the iterative step

$$x^{k+1} = Tx^k = x^k + \gamma A^\dagger(b - Ax^k), \tag{7.7}$$

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

$$x^{k+1} = x^k - \gamma \nabla f(x^k). \tag{7.8}$$

We see from Equation (7.8) that the Landweber algorithm is a special case of *gradient descent* minimization of a function $f(x)$.

7.3.1 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 [2] has the iterative step

$$x_j^{k+1} = x_j^k + \frac{1}{A_{+j}} \sum_{i=1}^{I} A_{ij}(b_i - (Ax^k)_i)/A_{i+}. \tag{7.9}$$

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 (7.9) becomes

$$z^{k+1} = z^k + B^T(c - Bz^k), \tag{7.10}$$

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.

7.4 The Projected Landweber Algorithm

For a closed, nonempty convex set C in C^J, the projected Landweber algorithm [12] has the iterative step

$$x^{k+1} = P_C(x^k + \gamma A^\dagger(b - Ax^k)). \tag{7.11}$$

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.

7.5 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 algorthm 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 [31, 32]. This algorithm minimizes $f(x)$ over x in C, whenever such minimizers exist, provided that γ is in the interval $(0, 2/\rho(A^\dagger A))$.

7.6 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 [4]. From

$$Mz = -Kz + h$$

we derive the iteration

$$z^{k+1} = -M^{-1}Kz^k + M^{-1}h. \tag{7.12}$$

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.

7.7 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

$$z^{k+1} = z^k + D^{-1}(h - Sz^k), \tag{7.13}$$

which we can write as

$$z^{k+1} = Tz^k = Bz^k + d, \tag{7.14}$$

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.

7.8 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.

7.8.1 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* $AA^\dagger z = b$ [57].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 $D^{-1/2}Ax = D^{-1/2}b$.

7.9 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.

7.9.1 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 $Sz = h$.

7.10 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 $\omega > 2$.

7.10.1 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 $Ax = b$.

7.11 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 $i = k(\mathrm{mod}\, I) + 1$ 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 ortogonal projection* (SOP) method [76]. It converges whenever the intersection is nonempty.

7.11.1 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, 104].

7.12 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 [74] has the iterative step

$$x_j^{k+1} = x_j^k (b_i/(Ax^k))^{A_{ij}/m_i},$$

for $i = k(\mathrm{mod}\, I) + 1$ 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.

7.13 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) [20, 21, 56, 84, 113]. The SMART method begins with a positive vector x^0; having calculated x^k, we calculate x^{k+1} using

$$\log x_j^{k+1} = \log x_j^k + s_j^{-1} \sum_{i=1}^{I} A_{ij} \log \frac{b_i}{(Ax^k)_i}, \tag{7.15}$$

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.

7.14 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 [20, 21, 58, 93, 94, 115, 120] 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.

7.15 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 $n = k(\mathrm{mod}\, N) + 1$. The RBI-EMML [23, 33] algorithm has the following iterative step:

$$x_j^{k+1} = x_j^k(1 - m_n^{-1}s_j^{-1}s_{nj}) + x_j^k m_n^{-1}s_j^{-1}\sum_{i\in S_n} A_{ij}\frac{b_i}{(Ax^k)_i}, \tag{7.16}$$

where

$$m_n = \max\{s_{nj}/s_j \,|j = 1, ..., J\}. \tag{7.17}$$

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.

7.16 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 4.1. 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 5.2. 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 8

The Algebraic Reconstruction Technique

The algebraic reconstruction technique (ART) [74] 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.

8.1 The ART

For each index value i let H_i be the hyperplane of J-dimensional vectors given by

$$H_i = \{x | (Ax)_i = b_i\}, \tag{8.1}$$

and P_i the orthogonal projection operator onto H_i. Let x^0 be arbitrary and, for each nonnegative integer k, let $i(k) = k(\mathrm{mod}\, I) + 1$. The iterative step of the ART is

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

Because the ART uses only a single equation at each step, it has been called a *row-action* method [38].

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

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

As we saw previously, the operators P_i are averaged (av), so that the operator T is av. According to the KM theorem, the sequence $\{T^k x\}$ will converge to a fixed point of T, for any x, whenever such fixed points exist.

When the system $Ax = b$ has solutions, the fixed points of T are solutions. When there are no solutions of $Ax = b$, the operator T will still have fixed points, but they will no longer be exact solutions.

The ART can also include relaxation. For ω in the interval $(0, 2)$, let

$$Q_i = (1 - \omega)I + \omega P_i.$$

As we have seen, the operators Q_i are also av, as is their product.

8.2 Calculating the ART

Given any vector z the vector in H_i closest to z, in the sense of the Euclidean distance, has the entries

$$x_j = z_j + \overline{A_{ij}}(b_i - (Az)_i)/\sum_{m=1}^{J} |A_{im}|^2 = z_j + \overline{A_{ij}}(b_i - (Az)_i). \tag{8.2}$$

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 in H_i closest to x^k. We can use Equation (8.2) to write

$$x_j^{k+1} = x_j^k + \overline{A_{ij}}(b_i - (Ax^k)_i). \tag{8.3}$$

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. In selecting the equation ordering, the important thing is to avoid particularly bad orderings, in which the hyperplanes H_i and H_{i+1} are nearly parallel. Relaxed ART has the iterative step

$$x_j^{k+1} = x_j^k + \gamma\overline{A_{ij}}(b_i - (Ax^k)_i), \tag{8.4}$$

where $\gamma \in (0, 2)$.

8.3 When $Ax = b$ Has Solutions

When the system $Ax = b$ is consistent, that is, has solutions, the convergence of the full-cycle ART sequence

$$z^{k+1} = P_I P_{I-1} \cdots P_2 P_1 z^k$$

to a solution is a consequence of the KM theorem. In fact, as we shall show now, the ART sequence $\{x^{k+1} = P_{i(k)}x^k\}$ also converges, and to the solution closest to the initial vector x^0.

Exercise 8.1 *Let x^0 and y^0 be arbitrary and $\{x^k\}$ and $\{y^k\}$ be the sequences generated by applying the ART algorithm, beginning with x^0 and y^0, respectively; that is, $y^{k+1} = P_{i(k)}y^k$. Show that*

$$||x^0 - y^0||_2^2 - ||x^I - y^I||_2^2 = \sum_{i=1}^{I} |(Ax^{i-1})_i - (Ay^{i-1})_i|^2. \tag{8.5}$$

We give a proof of the following result.

Theorem 8.1 *Let $A\hat{x} = b$ and let x^0 be arbitrary. Let $\{x^k\}$ be generated by Equation (8.3). Then the sequence $\{||\hat{x} - x^k||_2\}$ is decreasing and $\{x^k\}$ converges to the solution of $Ax = b$ closest to x^0.*

Proof: Let $A\hat{x} = b$. Let $v_i^r = (Ax^{rI+i-1})_i$ and $v^r = (v_1^r, ..., v_I^r)^T$, for $r = 0, 1, ...$ It follows from Equation (8.5) that the sequence $\{||\hat{x} - x^{rI}||_2\}$ is decreasing and the sequence $\{v^r - b\} \to 0$. So $\{x^{rI}\}$ is bounded; let $x^{*,0}$ be a cluster point. Then, for $i = 1, 2, ..., I$, let $x^{*,i}$ be the successor of $x^{*,i-1}$ using the ART algorithm. It follows that $(Ax^{*,i-1})_i = b_i$ for each i, from which we conclude that $x^{*,0} = x^{*,i}$ for all i and that $Ax^{*,0} = b$. Using $x^{*,0}$ in place of the arbitrary solution $\hat{x}$, we have that the sequence $\{||x^{*,0} - x^k||_2\}$ is decreasing. But a subsequence converges to zero, so $\{x^k\}$ converges to $x^{*,0}$. By Equation (8.5), the difference $||\hat{x} - x^k||_2^2 - ||\hat{x} - x^{k+1}||_2^2$ is independent of which solution $\hat{x}$ we pick; consequently, so is $||\hat{x} - x^0||_2^2 - ||\hat{x} - x^{*,0}||_2^2$. It follows that $x^{*,0}$ is the solution closest to x^0. This completes the proof. ∎

8.4 When $Ax = b$ Has No Solutions

When there are no exact solutions, the ART does not converge to a single vector, but, for each fixed i, the subsequence $\{x^{nI+i}, n = 0, 1, ...\}$ converges to a vector z^i and the collection $\{z^i \,|i = 1, ..., I\}$ is called the *limit cycle* [118, 60, 34]. For simplicity, we assume that $I > J$, and that the matrix A has full rank, which implies that $Ax = 0$ if and only if $x = 0$. Because the operator $T = P_I P_{i-1} \cdots P_2 P_1$ is av, this subsequential convergence to a limit cycle will follow from the KM theorem, once we have established that T has fixed points.

8.4.1 Subsequential Convergence of ART

We know from Exercise (34.25) that the operator T is affine linear and has the form

$$Tx = Bx + d,$$

where B is the matrix

$$B = (I - a^I(a^I)^\dagger) \cdots (I - a^1(a^1)^\dagger),$$

and d a vector.

The matrix $I - B$ is invertible, since if $(I - B)x = 0$, then $Bx = x$. It follows that x is in H_{i0} for each i, which means that $\langle a^i, x\rangle = 0$ for each i. Therefore $Ax = 0$, and so $x = 0$.

Exercise 8.2 *Show that the operator T is strictly nonexpansive, meaning that*

$$||x - y||_2 \geq ||Tx - Ty||_2,$$

with equality if and only if $x = Tx$ and $y = Ty$. Hint: Write $Tx - Ty = Bx - By = B(x - y)$ Since B is the product of orthogonal projections, B is av. Therefore, there is $\alpha > 0$ with

$$||x - y||_2^2 - ||Bx - By||_2^2 \geq (\frac{1}{\alpha} - 1)||(I - B)x - (I - B)y||_2^2.$$

The function $||x-Tx||_2$ has minimizers, since $||x-Tx||_2^2 = ||x-Bx-d||_2^2$ is quadratic in x. For any such minimizer z we will have

$$||z - Tz||_2 = ||Tz - T^2z||_2.$$

Since T is strictly ne, it follows that $z = Tz$.

Exercise 8.3 *Let $AA^\dagger = L + D + L^\dagger$, for diagonal matrix D and lower triangular matrix L. Show that, for the operator T above, Tx can be written as*

$$Tx = (I - A^\dagger(L + D)^{-1})x + A^\dagger(L + D)^{-1}b.$$

As we shall see, this formulation of the operator T provides a connection between the full-cycle ART for $Ax = b$ and the Gauss-Seidel method, as applied to the system $AA^\dagger z = b$ [57].

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?

8.4.2 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

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

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 [24] (see also [34]) 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 [118, 60, 24, 34]. The proof is easy in the consistent case. Is there an easy proof for the inconsistent case?

8.4.3 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 (8.3) with $(x^{k+1})_+$. This version of ART will converge to a nonnegative solution, whenever one exists, but will produce a limit cycle otherwise.

8.5 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) [28]:

8.5.1 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 $Ax = y$.

8.5.2 Strongly Underrelaxed ART

Another method for avoiding the LC is *strong underrelaxation* [39]. Let $t > 0$. Replace the iterative step in ART with

$$x_j^{k+1} = x_j^k + t\overline{A_{ij}}(b_i - (Ax^k)_i). \tag{8.6}$$

In [39] 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 [24]. Bertsekas [13] uses strong underrelaxation to obtain convergence of more general incremental methods.

8.6 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 [109], the vector $A\hat{x}$ must satisfy the condition

$$\sum_{i=1}^{I} A_{ij}((A\hat{x})_i - b_i) = 0 \tag{8.7}$$

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 (8.7) 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 8.2 *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 ([20]) 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 9

Simultaneous ART

The ART is a sequential algorithm, using only a single equation from the system $Ax = b$ at each step of the iteration. In this chapter we consider iterative procedures for solving $Ax = b$ in which all of the equations are used at each step. Such methods are called *simultaneous* algorithms. As before, we shall assume that the equations have been normalized so that the rows of A have Euclidean length one.

9.1 Cimmino's Algorithm

The ART seeks a solution of $Ax = b$ by projecting the current vector x^k orthogonally onto the next hyperplane $H(a^{i(k)}, b_{i(k)})$ to get x^{k+1}. In Cimmino's algorithm, we project the current vector x^k onto each of the hyperplanes and then average the result to get x^{k+1}. The algorithm begins with an arbitrary x^0; the iterative step is then

$$x^{k+1} = \frac{1}{I} \sum_{i=1}^{I} P_i x^k, \tag{9.1}$$

where P_i is the orthogonal projection onto $H(a^i, b_i)$.

Exercise 9.1 *Show that the iterative step can then be written as*

$$x^{k+1} = x^k + \frac{1}{I} A^\dagger (b - Ax^k). \tag{9.2}$$

As we saw in our discussion of the ART, when the system $Ax = b$ has no solutions, the ART does not converge to a single vector, but to a limit cycle. One advantage of many simultaneous algorithms, such as Cimmino's, is that they do converge to the least squares solution in the inconsistent case.

Cimmino's algorithm has the form $x^{k+1} = Tx^k$, for the operator T given by

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

Experience with Cimmino's algorithm shows that it is slow to converge. In the next section we consider how we might accelerate the algorithm.

9.2 The Landweber Algorithms

The Landweber algorithm [92, 12], with the iterative step

$$x^{k+1} = x^k + \gamma A^\dagger(b - Ax^k), \tag{9.3}$$

converges to the least squares solution closest to the starting vector x^0, provided that $0 < \gamma < 2/\lambda_{max}$, where λ_{max} is the largest eigenvalue of the nonnegative-definite matrix $A^\dagger A$. Loosely speaking, the larger γ is, the faster the convergence. However, precisely because A is large, calculating the matrix $A^\dagger A$, not to mention finding its largest eigenvalue, can be prohibitively expensive. The matrix A is said to be sparse if most of its entries are zero. In [31] upper bounds for λ_{max} were obtained in terms of the degree of sparseness of the matrix A; we discuss these bounds in the final section of this chapter.

9.2.1 Finding the Optimum γ

The operator

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

is affine linear and is av if and only if its linear part, the Hermitian matrix

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

is av. To guarantee this we need $0 \leq \gamma < 2/\lambda_{max}$. Should we always try to take γ near its upper bound, or is there an optimum value of γ? To answer this question we consider the eigenvalues of B for various values of γ.

Exercise 9.2 *Show that, if $\gamma < 0$, then none of the eigenvalues of B is less than one.*

Exercise 9.3 *Show that, for*

$$0 \leq \gamma \leq \frac{2}{\lambda_{max} + \lambda_{min}},$$

we have

$$\rho(B) = 1 - \gamma\lambda_{min};$$

the smallest value of $\rho(B)$ *occurs when*

$$\gamma = \frac{2}{\lambda_{max} + \lambda_{min}},$$

and equals

$$\frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}}.$$

Similarly, show that, for

$$\gamma \geq \frac{2}{\lambda_{max} + \lambda_{min}},$$

we have

$$\rho(B) = \gamma\lambda_{max} - 1;$$

the smallest value of $\rho(B)$ *occurs when*

$$\gamma = \frac{2}{\lambda_{max} + \lambda_{min}},$$

and equals

$$\frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}}.$$

We see from this exercise that, if $0 \leq \gamma < 2/\lambda_{max}$, and $\lambda_{min} > 0$, then $||B||_2 = \rho(B) < 1$, so that B is sc. We minimize $||B||_2$ by taking

$$\gamma = \frac{2}{\lambda_{max} + \lambda_{min}},$$

in which case we have

$$||B||_2 = \frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}} = \frac{c-1}{c+1},$$

for $c = \lambda_{max}/\lambda_{min}$, the *condition number* of the positive-definite matrix $A^\dagger A$. The closer c is to one, the smaller the norm $||B||_2$, and the faster the convergence.

On the other hand, if $\lambda_{min} = 0$, then $\rho(B) = 1$ for all γ in the interval $(0, 2/\lambda_{max})$. The matrix B is still av, but it is no longer sc. For example, consider the orthogonal projection P_0 onto the hyperplane $H_0 = H(a, 0)$, where $||a||_2 = 1$. This operator can be written

$$P_0 = I - aa^\dagger.$$

The largest eigenvalue of $aa^\dagger$ is $\lambda_{max} = 1$; the remaining ones are zero. The relaxed projection operator

$$B = I - \gamma aa^\dagger$$

has $\rho(B) = 1 - \gamma > 1$, if $\gamma < 0$, and for $\gamma \geq 0$, we have $\rho(B) = 1$. The operator B is av, in fact, it is fne, but it is not sc.

It is worth noting that the definition of the condition number given above applies only to positive-definite matrices. For general square, invertible matrices S, the condition number depends on the particular induced matrix norm and is defined as

$$c = ||S||\,||S^{-1}||.$$

To motivate this definition of the condition number, suppose that $x = S^{-1}h$ is the solution of $Sx = h$, and that h is perturbed to $h + \delta_h$. Then let δ_x be such that $x + \delta_x = S^{-1}(h + \delta_h)$. The relative change in the solution, $||\delta_x||/||x||$, is related to the relative change in h, $||\delta_h||/||h||$, by

$$\frac{||\delta_x||}{||x||} \leq ||S||\,||S^{-1}||\frac{||\delta_h||}{||h||}.$$

9.2.2 The Projected Landweber Algorithm

When we require a nonnegative approximate solution x for the real system $Ax = b$ we can use a modified version of the Landweber algorithm, called the projected Landweber algorithm [12], in this case having the iterative step

$$x^{k+1} = (x^k + \gamma A^\dagger(b - Ax^k))_+, \tag{9.4}$$

where, for any real vector a, we denote by $(a)_+$ the nonnegative vector whose entries are those of a, for those that are nonnegative, and are zero otherwise. The projected Landweber algorithm converges to a vector that minimizes $||Ax - b||_2$ over all nonnegative vectors x, for the same values of γ.

The projected Landweber algorithm is actually more general. For any closed, nonempty convex set C in X, define the iterative sequence

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

This sequence converges to a minimizer of the function $||Ax - b||_2$ over all x in C, whenever such minimizers exist.

Both the Landweber and projected Landweber algorithms are special cases of the CQ algorithm [31], which, in turn, is a special case of the more general iterative fixed point algorithm, the Krasnoselskii/Mann (KM) method.

9.3 An Upper Bound for the Maximum Eigenvalue of $A^\dagger A$

The upper bounds for λ_{max} we present here apply to any matrix A, but will be particularly helpful when A is sparse.

9.3.1 The Normalized Case

We assume now that the matrix A has been normalized so that each of its rows has Euclidean length one. Denote by s_j the number of nonzero entries in the jth column of A, and let s be the maximum of the s_j. Our first result is the following [31]:

Theorem 9.1 *For normalized A, λ_{max}, the largest eigenvalue of the matrix $A^\dagger A$, does not exceed s.*

Proof: For notational simplicity, we consider only the case of real matrices and vectors. Let $A^T Av = cv$ for some nonzero vector v. We show that $c \leq s$. We have $AA^T Av = cAv$ and so $w^T AA^T w = v^T A^T AA^T Av = cv^T A^T Av = cw^T w$, for $w = Av$. Then, with $e_{ij} = 1$ if $A_{ij} \neq 0$ and $e_{ij} = 0$ otherwise, we have

$$(\sum_{i=1}^{I} A_{ij} w_i)^2 = (\sum_{i=1}^{I} A_{ij} e_{ij} w_i)^2$$

$$\leq (\sum_{i=1}^{I} A_{ij}^2 w_i^2)(\sum_{i=1}^{I} e_{ij}^2) =$$

$$(\sum_{i=1}^{I} A_{ij}^2 w_i^2) s_j \leq (\sum_{i=1}^{I} A_{ij}^2 w_i^2) s.$$

Therefore,

$$w^T AA^T w = \sum_{j=1}^{J} (\sum_{i=1}^{I} A_{ij} w_i)^2 \leq \sum_{j=1}^{J} (\sum_{i=1}^{I} A_{ij}^2 w_i^2) s,$$

and

$$w^T AA^T w = c \sum_{i=1}^{I} w_i^2 = c \sum_{i=1}^{I} w_i^2 (\sum_{j=1}^{J} A_{ij}^2)$$

$$= c \sum_{i=1}^{I} \sum_{j=1}^{J} w_i^2 A_{ij}^2.$$

The result follows immediately. ∎

When A is normalized the trace of AA^T, that is, the sum of its diagonal entries, is M. Since the trace is also the sum of the eigenvalues of both AA^T and A^TA, we have $\lambda_{max} \leq M$. When A is sparse, s is much smaller than M, so provides a much tighter upper bound for λ_{max}.

9.3.2 The General Case

A similar upper bound for λ_{max} is given for the case in which A is not normalized.

Theorem 9.2 *For each $i = 1, ..., I$ let $\nu_i = \sum_{j=1}^{J} |A_{ij}|^2 > 0$. For each $j = 1, ..., J$, let $\sigma_j = \sum_{i=1}^{I} e_{ij}\nu_i$, where $e_{ij} = 1$ if $A_{ij} \neq 0$ and $e_{ij} = 0$ otherwise. Let σ denote the maximum of the σ_j. Then the eigenvalues of the matrix $A^\dagger A$ do not exceed σ.*

The proof of Theorem 9.2 is similar to that of Theorem 9.1; the details are in [31].

9.3.3 Upper Bounds for ϵ-Sparse Matrices

If A is not sparse, but most of its entries have magnitude not exceeding $\epsilon > 0$ we say that A is ϵ-sparse. We can extend the results for the sparse case to the ϵ-sparse case.

Given a matrix A, define the entries of the matrix B to be $B_{ij} = A_{ij}$ if $|A_{ij}| > \epsilon$, and $B_{ij} = 0$, otherwise. Let $C = A - B$; then $|C_{ij}| \leq \epsilon$, for all i and j. If A is ϵ-sparse, then B is sparse. The 2-norm of the matrix A, written $||A||_2$, is defined to be the square root of the largest eigenvalue of the matrix $A^\dagger A$, that is, $||A||_2 = \sqrt{\lambda_{max}}$. From Theorem 9.2 we know that $||B||_2 \leq \sigma$. The trace of the matrix $C^\dagger C$ does not exceed $IJ\epsilon^2$. Therefore

$$\sqrt{\lambda_{max}} = ||A||_2 = ||B + C||_2 \leq ||B||_2 + ||C||_2 \leq \sqrt{\sigma} + \sqrt{IJ}\epsilon, \qquad (9.5)$$

so that

$$\lambda_{max} \leq \sigma + 2\sqrt{\sigma IJ}\epsilon + IJ\epsilon^2. \qquad (9.6)$$

Simulation studies have shown that these upper bounds become tighter as the size of the matrix A increases. In hundreds of runs, with I and J in the hundreds, we found that the relative error of the upper bound was around one percent [36].

Chapter 10

Block-Iterative Variants of ART

As we have seen, the ART uses one equation at a time, while the simultaneous Cimmino and Landweber algorithms use all the equations at each step of the iteration. Block-iterative ART is more general, in that it allows us to use some, but perhaps not all, of the equations at each step.

10.1 The Block-Iterative ART

We consider the system of linear equations $Ax = b$, where A is a complex I by J matrix. For notational simplicity, we shall assume that the equations have been rescaled so that each row of A has Euclidean length one. Let the index set $\{i = 1, ..., I\}$ be partitioned into N subsets, or blocks, B_1,...,B_N, for some positive integer N, with $1 \leq N \leq I$. Let I_n be the cardinality of B_n. Let A_n be the I_n by J matrix obtained from A by discarding all rows except those whose index is in B_n. Similarly, let b^n be the I_n by 1 vector obtained from b. For $k = 0, 1, ...$, let $n = k(\mathrm{mod}\, N) + 1$. The *block-iterative ART* (BI-ART) has the iterative step

$$x^{k+1} = x^k + \frac{1}{I_n} A_n^\dagger (b^n - A_n x^k). \tag{10.1}$$

10.2 The Rescaled Block-Iterative ART

More generally, the *rescaled* BI-ART (RE-BI-ART) has the iterative step

$$x^{k+1} = x^k + \gamma_n A_n^\dagger (b^n - A_n x^k), \tag{10.2}$$

for $0 < \gamma_n < 2/L_n$, where L_n is the largest eigenvalue of the matrix $A_n^\dagger A_n$. How we select the blocks and the parameters γ_n will determine the speed of convergence of RE-BI-ART

10.3 Convergence of the RE-BI-ART

Suppose now that the system is consistent and that $A\hat{x} = b$. Then

$$||\hat{x} - x^k||_2^2 - ||\hat{x} - x^{k+1}||_2^2$$

$$= 2\gamma_n Re\langle \hat{x} - x^k, A_n^\dagger(b^n - A_n x^k)\rangle - \gamma_n^2 ||A_n^\dagger(b^n - A_n x^k)||_2^2$$

$$= 2\gamma_n ||b^n - A_n x^k||_2^2 - \gamma_n^2 ||A_n^\dagger(b^n - A_n x^k)||_2^2.$$

Therefore, we have

$$||\hat{x} - x^k||_2^2 - ||\hat{x} - x^{k+1}||_2^2 \geq (2\gamma_n - \gamma_n^2 L_n)||b^n - A_n x^k||_2^2. \tag{10.3}$$

It follows that the sequence $\{||\hat{x} - x^k||_2^2\}$ is decreasing and that the sequence $\{||b^n - A_n x^k||_2^2\}$ converges to 0. The sequence $\{x^k\}$ is then bounded; let x^* be any cluster point of the subsequence $\{x^{mN}\}$. Then let

$$x^{*,n} = x^{*,n-1} + \gamma_n A_n^\dagger(b^n - A_n x^{*,n-1}),$$

for $n = 1, 2, ..., N$. It follows that $x^{*,n} = x^*$ for all n and that $Ax^* = b$. Replacing the arbitrary solution $\hat{x}$ with x^*, we find that the sequence $\{||x^* - x^k||_2^2\}$ is decreasing; but a subsequence converges to zero. Consequently, the sequence $\{||x^* - x^k||_2^2\}$ converges to zero. We can therefore conclude that the RE-BI-ART converges to a solution, whenever the system is consistent. In fact, since we have shown that the difference $||\hat{x} - x^k||_2^2 - ||\hat{x} - x^{k+1}||_2^2$ is nonnegative and independent of the solution $\hat{x}$ that we choose, we known that the difference $||\hat{x} - x^0||_2^2 - ||\hat{x} - x^*||_2^2$ is also nonnegative and independent of $\hat{x}$. It follows that x^* is the solution closest to x^0.

From the Inequality (10.3) we see that we make progress toward a solution to the extent that the right side of the inequality,

$$(2\gamma_n - \gamma_n^2 L_n)||b^n - A_n x^k||_2^2$$

is large. One conclusion we draw from this is that we want to avoid ordering the blocks so that the quantity $||b^n - A_n x^k||_2^2$ is small. We also want to select γ_n reasonably large, subject to the bound $\gamma_n < 2/L_n$; the maximum of $2\gamma_n - \gamma_n^2 L_n$ is at $\gamma_n = L_n$. Because the rows of A_n have length one, the trace of $A_n^\dagger A_n$ is I_n, the number of rows in A_n. Since L_n is not greater than this trace, we have $L_n \leq I_n$, so the choice of $\gamma_n = 1/I_n$ used in BI-ART is acceptable, but possibly far from optimal, particularly if A_n is sparse.

Inequality (10.3) can be used to give a rough measure of the speed of convergence of RE-BI-ART. The term $||b^n - A_n x^k||_2^2$ is on the order of I_n,

while the term $2\gamma_n - \gamma_n^2 L_n$ has $1/L_n$ for its maximum, so, very roughly, is on the order of $1/I_n$. Consequently, the improvement made in one step of BI-ART is on the order of one. One complete cycle of BI-ART, that is, one complete pass through all the blocks, then corresponds to an improvement on the order of N, the number of blocks. It is a "rule of thumb" that block-iterative methods are capable of improving the speed of convergence by a factor of the number of blocks, if unfortunate ordering of the blocks and selection of the equations within the blocks are avoided, and the parameters are well chosen.

To obtain good choices for the γ_n , we need to have a good estimate of L_n. As we have seen, such estimates are available for sparse matrices.

10.4 Using Sparseness

Let s_{nj} be the number of non-zero elements in the j-th column of A_n, and let s_n be the maximum of the s_{nj}. We know then that $L_n \leq s_n$. Therefore, we can choose $\gamma_n < 2/s_n$.

Suppose, for the sake of illustration, that each column of A has s non-zero elements, for some $s < I$, and we let $r = s/I$. Suppose also that $I_n = I/N$ and that N is not too large. Then s_n is approximately equal to $rI_n = s/N$. On the other hand, unless A_n has only zero entries, we know that $s_n \geq 1$. Therefore, it is no help to select N for which $s/N < 1$. For a given degree of sparseness s we need not select N greater than s. The more sparse the matrix A, the fewer blocks we need to gain the maximum advantage from the rescaling, and the more we can benefit from parallelizability in the calculations at each step of the RE-BI-ART.

Chapter 11

Jacobi and Gauss-Seidel Methods

Linear systems $Ax = b$ need not be square but can be associated with two square systems, $A^\dagger Ax = A^\dagger b$, the so-called *normal equations*, and $AA^\dagger z = b$, sometimes called the *Björck-Elfving equations* [57]. In this chapter we consider two well known iterative algorithms for solving square systems of linear equations, the Jacobi method and the Gauss-Seidel method. Both these algorithms are easy to describe and to motivate. They both require not only that the system be square, that is, have the same number of unknowns as equations, but satisfy additional constraints needed for convergence.

Both the Jacobi and the Gauss-Seidel algorithms can be modified to apply to any square system of linear equations, $Sz = h$. The resulting algorithms, the Jacobi overrelaxation (JOR) and successive overrelaxation (SOR) methods, involve the choice of a parameter. The JOR and SOR will converge for more general classes of matrices, provided that the parameter is appropriately chosen.

When we say that an iterative method is convergent, or converges, under certain conditions, we mean that it converges for any consistent system of the appropriate type, and for any starting vector; any iterative method will converge if we begin at the right answer.

11.1 The Jacobi and Gauss-Seidel Methods: An Example

Suppose we wish to solve the 3 by 3 system

$$S_{11}z_1 + S_{12}z_2 + S_{13}z_3 = h_1$$

$$S_{21}z_1 + S_{22}z_2 + S_{23}z_3 = h_2$$
$$S_{31}z_1 + S_{32}z_2 + S_{33}z_3 = h_3,$$

which we can rewrite as

$$z_1 = S_{11}^{-1}[h_1 - S_{12}z_2 - S_{13}z_3]$$
$$z_2 = S_{22}^{-1}[h_2 - S_{21}z_1 - S_{23}z_3]$$
$$z_3 = S_{33}^{-1}[h_3 - S_{31}z_1 - S_{32}z_2],$$

assuming that the diagonal terms S_{mm} are not zero. Let $z^0 = (z_1^0, z_2^0, z_3^0)^T$ be an initial guess for the solution. We then insert the entries of z^0 on the right sides and use the left sides to define the entries of the next guess z^1. This is one full cycle of *Jacobi's method*.

The Gauss-Seidel method is similar. Let $z^0 = (z_1^0, z_2^0, z_3^0)^T$ be an initial guess for the solution. We then insert z_2^0 and z_3^0 on the right side of the first equation, obtaining a new value z_1^1 on the left side. We then insert z_3^0 and z_1^1 on the right side of the second equation, obtaining a new value z_2^1 on the left. Finally, we insert z_1^1 and z_2^1 into the right side of the third equation, obtaining a new z_3^1 on the left side. This is one full cycle of the *Gauss-Seidel* (GS) method.

11.2 Splitting Methods

The Jacobi and the Gauss-Seidel methods are particular cases of a more general approach, known as splitting methods. Splitting methods apply to square systems of linear equations. Let S be an arbitrary N by N square matrix, written as $S = M - K$. Then the linear system of equations $Sz = h$ is equivalent to $Mz = Kz + h$. If M is invertible, then we can also write $z = M^{-1}Kz + M^{-1}h$. This last equation suggests a class of iterative methods for solving $Sz = h$ known as *splitting methods*. The idea is to select a matrix M so that the equation

$$Mz^{k+1} = Kz^k + h$$

can be easily solved to get z^{k+1}; in the Jacobi method M is diagonal, and in the Gauss-Seidel method, M is triangular. Then we write

$$z^{k+1} = M^{-1}Kz^k + M^{-1}h. \tag{11.1}$$

From $K = M - S$, we can write Equation (17.8) as

$$z^{k+1} = z^k + M^{-1}(h - Sz^k). \tag{11.2}$$

Suppose that S is invertible and $\hat{z}$ is the unique solution of $Sz = h$. The error we make at the k-th step is $e^k = \hat{z} - z^k$.

Exercise 11.1 *Show that* $e^{k+1} = M^{-1}Ke^k$

We want the error to decrease with each step, which means that we should seek M and K so that $||M^{-1}K|| < 1$. If S is not invertible and there are multiple solutions of $Sz = h$, then we do not want $M^{-1}K$ to be a strict contraction, but only av or pc. The operator T defined by

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

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

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

$$-1 < \lambda \leq 1,$$

for all (necessarily real) eigenvalues λ of B.

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

Alternatively, we can use Theorem 5.2. According to that theorem, if B has a basis of eigenvectors, and $|\lambda| < 1$ for all eigenvalues λ of B that are not equal to one, then $\{z^k\}$ will converge to a solution of $Sz = h$, whenever solutions exist.

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

$$S = L + D + U,$$

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

$$S = L + D + L^\dagger.$$

We list now several examples of iterative algorithms obtained by the splitting method. In the remainder of the chapter we discuss these methods in more detail.

11.3 Some Examples of Splitting Methods

As we shall now see, the Jacobi and Gauss-Seidel methods, as well as their overrelaxed versions, JOR and SOR, are splitting methods.

Jacobi's Method: Jacobi's method uses $M = D$ and $K = -L-U$, under the assumption that D is invertible. The matrix B is

$$B = M^{-1}K = -D^{-1}(L+U). \tag{11.3}$$

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

$$B = I - (D+L)^{-1}S. \tag{11.4}$$

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

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

and

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

The matrix B is

$$B = M^{-1}K = (I - \omega D^{-1}S). \tag{11.5}$$

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

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

or

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

or

$$= (I + \omega D^{-1}L)^{-1}[(1-\omega)I - \omega D^{-1}U]. \tag{11.6}$$

11.4 Jacobi's Algorithm and JOR

The matrix B in Equation (11.3) is not generally av and the Jacobi iterative scheme will not converge, in general. Additional conditions need to be imposed on S in order to guarantee convergence. One such condition is that S be strictly diagonally dominant. In that case, all the eigenvalues of $B = M^{-1}K$ can be shown to lie inside the unit circle of the complex plane, so that $\rho(B) < 1$. It follows from Lemma 35.1 that B is sc with respect to some vector norm, and the Jacobi iteration converges. If, in addition, S is Hermitian, the eigenvalues of B are in the interval $(-1, 1)$, and so B is sc with respect to the Euclidean norm.

Alternatively, one has the *Jacobi overrelaxation* (JOR) method, which is essentially a special case of the Landweber algorithm and involves an arbitrary parameter.

For S an N by N matrix, Jacobi's method can be written as

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

for $m = 1, ..., N$. With D the invertible diagonal matrix with entries $D_{mm} = S_{mm}$ we can write one cycle of Jacobi's method as

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

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

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

choosing $\omega = 1$ we get the Jacobi method. Convergence of the JOR iteration will depend, of course, on properties of S and on the choice of ω. When S is Hermitian, nonnegative-definite, for example, $S = A^\dagger A$ or $S = AA^\dagger$, we can say more.

11.4.1 The JOR in the Nonnegative-definite Case

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

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

Exercise 11.2 *Show that, if $\{z^k\}$ is the sequence obtained from the JOR, then the sequence $\{A^\dagger z^k\}$ is the sequence obtained by applying the Landweber algorithm to the system $D^{-1/2}Ax = D^{-1/2}b$, where D is the diagonal part of the matrix $S = AA^\dagger$.*

If we select $\omega = 1/I$ we obtain the Cimmino method. Since the trace of the matrix $D^{-1/2}SD^{-1/2}$ equals I we know that $\omega = 1/I$ is not greater than the largest eigenvalue of the matrix $D^{-1/2}SD^{-1/2}$ and so this choice of ω is acceptable and the Cimmino algorithm converges whenever there are solutions of $Ax = b$. In fact, it can be shown that Cimmino's method converges to a least squares approximate solution generally.

Similarly, the JOR method applied to the system $A^\dagger Ax = A^\dagger b$ is equivalent to the Landweber algorithm, applied to the system $Ax = b$.

Exercise 11.3 *Show that, if $\{z^k\}$ is the sequence obtained from the JOR, then the sequence $\{D^{1/2}z^k\}$ is the sequence obtained by applying the Landweber algorithm to the system $AD^{-1/2}x = b$, where D is the diagonal part of the matrix $S = A^\dagger A$.*

11.5 The Gauss-Seidel Algorithm and SOR

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

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

where $S = D + L + U$ is the decomposition of the square matrix S into its diagonal, lower triangular and upper triangular diagonal parts. The GS method does not converge without restrictions on the matrix S. As with the Jacobi method, strict diagonal dominance is a sufficient condition.

11.5.1 The Nonnegative-Definite Case

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

Exercise 11.4 *Show that all the entries of D are nonnegative.*

We assume that all the diagonal entries of D are positive, so that $D + L$ is invertible. The Gauss-Seidel iterative step is $z^{k+1} = Tz^k$, where T is the affine linear operator given by $Tz = Bz + d$, for $B = -(D + L)^{-1}L^\dagger$ and $d = (D + L)^{-1}h$.

Proposition 11.1 *Let λ be an eigenvalue of B that is not equal to one. Then $|\lambda| < 1$.*

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

Proof of Proposition (11.1): Let $Bv = \lambda v$, for v nonzero. Then $-Bv = (D + L)^{-1}L^\dagger v = -\lambda v$, so that

$$L^\dagger v = -\lambda(D + L)v,$$

and

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

Therefore,

$$v^\dagger L^\dagger v = -\lambda v^\dagger (D + L)v.$$

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

$$v^\dagger Sv = (1 - \lambda)v^\dagger(D + L)v.$$

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

$$(1 - \overline{\lambda})(D + L)^\dagger v = (1 - \lambda)v^\dagger(D + L)v$$

$$= (1-\lambda)v^\dagger Dv + (1-\lambda)v^\dagger Lv$$
$$= (1-\lambda)v^\dagger Dv - (1-\lambda)\overline{\lambda}v^\dagger(D+L)^\dagger v.$$

So we have

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

or

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

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

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

which is equal to

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

and, on the right side, we get

$$|1-\lambda|^2 v^\dagger Dv.$$

Consequently, we have

$$(1-|\lambda|^2)v^\dagger Sv = |1-\lambda|^2 v^\dagger Dv.$$

Since $v^\dagger Sv \geq 0$ and $v^\dagger Dv > 0$, it follows that $1-|\lambda|^2 \geq 0$. If $|\lambda| = 1$, then $|1-\lambda|^2 = 0$, so that $\lambda = 1$. This completes the proof. ∎

Note that $\lambda = 1$ if and only if $Sv = 0$. Therefore, if S is invertible, the affine linear operator T is a strict contraction, and the GS iteration converges to the unique solution of $Sz = h$.

11.5.2 Successive Overrelaxation

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

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

the choice of $\omega = 1$ gives the GS method. Convergence of the SOR iteration will depend, of course, on properties of S and on the choice of ω.

Exercise 11.5 *Use the form*

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

to show that

$$|\det(B)| = |1-\omega|^N.$$

Conclude from this and the fact that the determinant of B is the product of its eigenvalues that $\rho(B) > 1$ if $\omega < 0$ or $\omega > 2$.

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

11.5.3 The SOR for Nonnegative-Definite S

When S is nonnegative-definite and the system $Sz = h$ is consistent the SOR converges to a solution for any $\omega \in (0, 2)$. This follows from the convergence of the ART algorithm, since, for such S, the SOR is equivalent to the ART.

Now we consider the SOR method applied to the Björck-Elfving equations. Rather than count a full cycle as one iteration, we now count as a single step the calculation of a single new entry. Therefore, for $k = 0, 1, ...$ the $k + 1$-st step replaces the value z_i^k only, where $i = k(\mathrm{mod}\, I) + 1$. We have

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

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

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

This is one step of the relaxed *algebraic reconstruction technique* (ART) applied to the original system of equations $Ax = b$. The relaxed ART converges to a solution, when solutions exist, for any $\omega \in (0, 2)$.

When $Ax = b$ is consistent, so is $AA^\dagger z = b$. We consider now the case in which $S = AA^\dagger$ is invertible. Since the relaxed ART sequence $\{x^k = A^\dagger z^k\}$ converges to a solution x^∞, for any $\omega \in (0, 2)$, the sequence $\{AA^\dagger z^k\}$ converges to b. Since $S = AA^\dagger$ is invertible, the SOR sequence $\{z^k\}$ then converges to $S^{-1}b$.

Chapter 12

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.

12.1 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

$$x^k = x^{k-1} + \alpha_k d^k. \tag{12.1}$$

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 12.1 *Differentiate the function $f(x^{k-1} + \alpha d^k)$ with respect to the variable α to show that*

$$\nabla f(x^k) \cdot d^k = 0. \tag{12.2}$$

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* [99]:

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

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 [106] 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.

12.2 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

$$f(x) = \frac{1}{2} ||b - Ax||_2^2. \tag{12.3}$$

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 (12.1) for $f(x)$ as in Equation (12.3). 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 12.2 *Show that the optimal α_k is*

$$\alpha_k = \frac{r^k \cdot d^k}{d^k \cdot Q d^k}, \tag{12.4}$$

where $r^k = c - Qx^{k-1}$.

Exercise 12.3 *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 12.1 *Let $\{d^1, ..., d^J\}$ be any finite set whose span is all of R^J. Let α_k be chosen according to Equation (12.4). Then, for $k = 0, 1, ...$, $j = k(\text{mod}\, J)$, and any x^0, the sequence defined by*

$$x^k = x^{k-1} + \alpha_k d^j$$

converges to the least squares solution.

Proof: The sequence $\{||\hat{x} - x^k||_Q^2\}$ is decreasing and, therefore, the sequence $\{(r^k \cdot d^k)^2 / d^k \cdot Qd^k$ must converge to zero. Therefore, the vectors x^k are bounded, and for each $j = 1, ..., J$, the subsequences $\{x^{mJ+j}$, $m = 0, 1, ...\}$ have cluster points, say $x^{*,j}$ with

$$x^{*,j} = x^{*,j-1} + \frac{(c - Qx^{*,j-1}) \cdot d^j}{d^j \cdot Qd^j} d^j.$$

Since

$$r^{mJ+j} \cdot d^j \to 0,$$

it follows that, for each $j = 1, ..., J$,

$$(c - Qx^{*,j}) \cdot d^j = 0.$$

Therefore,

$$x^{*,1} = ... = x^{*,J} = x^*$$

with $Qx^* = c$. Consequently, x^* is the least squares solution and the sequence $\{||x^* - x^k||_Q\}$ is decreasing. But a subsequence converges to zero; therefore, $\{||x^* - x^k||_Q\} \to 0$. This completes the proof. ∎

There is an interesting corollary to this theorem that pertains to a modified version of the ART algorithm. For $k = 0, 1, ...$ and $i = k(\text{mod}\, M) + 1$ 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 12.2 *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. ∎

12.3 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, in addition, the set $\{u^1, ..., u^M\}$ is an orthogonal basis, then $u^j \cdot u^m = 0$, unless $j = m$. 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 $\hat{x}$.

12.3.1 Conjugate Directions

From Equation (12.2) we have

$$(c - Qx^{k+1}) \cdot d^k = 0,$$

which can be expressed as

$$(\hat{x} - x^{k+1}) \cdot Qd^k = (\hat{x} - x^{k+1})^T Qd^k = 0.$$

Two vectors x and y are said to be *Q-orthogonal* (or *Q-conjugate*, or just *conjugate*), if $x \cdot Qy = 0$. So, the least-squares solution that we seek lies in a direction from x^{k+1} that is Q-orthogonal to d^k. This suggests that we can do better than steepest descent if we take the next direction to be Q-orthogonal to the previous one, rather than just orthogonal. This leads us to *conjugate direction methods.*

Exercise 12.4 *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.

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.

12.3.2 The Gram-Schmidt Method

Let $\{v^1, ..., v^J\}$ be an arbitrary basis for R^J. The Gram-Schmidt method uses the v^j to create an orthogonal basis $\{u^1, ..., u^J\}$ for R^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,

$$p^j = v^j - \frac{p^1 \cdot Qv^j}{p^1 \cdot Qp^1} p^1 - ... - \frac{p^{j-1} \cdot Qv^j}{p^{j-1} \cdot Qp^{j-1}} p^{j-1}. \tag{12.5}$$

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 $v^j = Q^j p^1$, we have a much more efficient mechanism for generating a conjugate basis, namely a three-term recursion formula [99]. The set $\{v^1, Qv^1, ..., Q^{J-1}v^1\}$ 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 12.3 *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

$$p^{n+1} = Qp^n - \frac{Qp^n \cdot Qp^n}{p^n \cdot Qp^n} p^n - \frac{Qp^n \cdot Qp^{n-1}}{p^{n-1} \cdot Qp^{n-1}} p^{n-1}. \tag{12.6}$$

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 that

$$p^{n+1} \cdot Qp^n = p^{n+1} \cdot Qp^{n-1} = 0.$$

For $j \leq n - 1$, 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 (12.6) also occur in the Gram-Schmidt approach in Equation (12.5); the point is that Equation (12.6) uses only the first three terms, in every case.

12.4 The Conjugate Gradient Method

The *conjugate gradient method* (CGM) combines the use of the negative gradient directions from the steepest descent method with the use of a conjugate basis of directions. Since, in the quadratic case, we have

$$-\nabla f(x^k) = r^k = (c - Qx^k),$$

the CGM constructs a conjugate basis of directions from the residuals r^k. The iterative step for the CGM is the following:

$$x^{n+1} = x^n + \frac{r^n \cdot p^n}{p^n \cdot Qp^n} p^n.$$

As before, there is an efficient recursion formula that provides the next direction: let $p^1 = r^1 = (c - Qx^0)$ and

$$p^{n+1} = r^{n+1} - \frac{r^{n+1} \cdot Qp^n}{p^n \cdot Qp^n} p^n. \tag{12.7}$$

Since the α_n is the optimal choice and

$$r^{n+1} = -\nabla f(x^{n+1}),$$

we have, according to Equation (12.2),

$$r^{n+1} \cdot p^n = 0.$$

Consequently, if $p^{n+1} = 0$ then $r^{n+1} = 0$ also, which tells us that $Qx^{n+1} = c$. In theory the CGM converges to the least squares solution in finitely many steps. 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 12.3 establishes that the set $\{p^1, ..., p^J\}$ is a conjugate set [99]. Assume that the set $\{p^1, ..., p^n\}$ is a conjugate set, for $n < J$ and show that the same is true for $\{p^1, ..., p^{n+1}\}$. The key steps in the proof are contained in the following exercises.

Exercise 12.5 *Use the fact that*

$$r^{j+1} = r^j - \alpha_j Q p^j,$$

to show that Qp^j is in the span of the vectors r^j and r^{j+1}.

Exercise 12.6 *Use Equation (12.7) and $p^1 = r^1$ to show that the spans of the sets $\{p^1, ..., p^j\}$ and $\{r^1, ..., r^j\}$ are the same.*

Exercise 12.7 *Show that, for $1 \leq j \leq n$, $p^j \cdot r^{n+1} = 0$. Hints: recall that $p^j \cdot r^{j+1} = 0$ because of the optimality of α_j. Then*

$$p^j \cdot r^{n+1} = p^j \cdot r^n - \alpha_n p^j \cdot Q p^n$$

$$= ... = p^j \cdot r^{j+1} - \alpha_{j+1} p^j \cdot Q p^{j+1} - ... - \alpha_n p^j \cdot Q p^n.$$

We know that the first term on the right side is zero. Now use the induction hypothesis.

Exercise 12.8 *Show that $r^j \cdot r^{n+1} = 0$, for $j = 1, ..., n$. Hint: use the fact that $p^j \cdot r^{n+1} = 0$ for $j = 1, ..., n$.*

Exercise 12.9 *Use the fact that Qp^j is in the span of r^j and r^{j+1} to show that $r^{n+1} \cdot Qp^j = 0$, for $j = 1, ..., n-1$.*

For $j = 1, ..., n-1$ we have

$$p^{n+1} \cdot Q p^j = r^{n+1} \cdot Q p^j - \frac{r^{n+1} \cdot Q p^n}{p^n \cdot Q p^n} p^n \cdot Q p^j.$$

Both terms on the right side are zero, so $p^{n+1} \cdot Qp^j = 0$. This concludes the induction proof.

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 $P^{-1/2} Q P^{-1/2}$, for some positive-definite approximation P of Q (see [4]).

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 [106].

Part IV

Positivity in Linear Systems

Chapter 13

The Multiplicative ART (MART)

The *multiplicative* ART (MART) [74] 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.

13.1 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. With $i = k(\mathrm{mod}\, I) + 1$, the iterative step of the ART algorithm is

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

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

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

13.2 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

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i} \Big)^{m_i^{-1} A_{ij}}, \tag{13.1}$$

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

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i} \Big)^{\gamma_i m_i^{-1} A_{ij}}, \tag{13.2}$$

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 13.1 *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.

13.3 ART and MART as Sequential Projection Methods

We know from our discussion of the ART that the iterative ART step can be viewed as the orthogonal projection of the current vector, x^k, onto H_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.

13.3.1 Cross-Entropy or the Kullback-Leibler Distance

For positive numbers u and v, the Kullback-Leibler distance [91] from u to v is

$$KL(u, v) = u \log \frac{u}{v} + v - u. \tag{13.3}$$

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

$$KL(x, z) = \sum_{j=1}^{J} KL(x_j, z_j). \tag{13.4}$$

Exercise 13.1 *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*

$$KL(x, z) = KL(x_+, z_+) + KL(x, \frac{x_+}{z_+}z). \tag{13.5}$$

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

$$z_j - x_j = z_j A_{ij} \lambda_i. \tag{13.6}$$

For the special case in which the entries of A_{ij} are zero or one, we can solve Equation (13.6) 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.

13.3.2 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 (13.6). 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

$$A_{ij}z_j - A_{ij}x_j = A_{ij}z_j\lambda_i. \tag{13.7}$$

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 (13.7), we obtain

$$z_j = (Q_i x)_j = x_j \frac{b_i}{(Ax)_i}, \tag{13.8}$$

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 $(Q_i x^k)_j$.

13.4 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 13.2 *Use Equation (13.5) to prove that $G_i(x, z) \geq 0$ for all x and z.*

Exercise 13.3 *Show that $G_i(x,z)$, viewed as a function of z, is minimized by $z = x$, by showing that*

$$G_i(x,z) = G_i(x,x) + KL(x,z) - m_i^{-1} KL((Ax)_i, (Az)_i). \tag{13.9}$$

Exercise 13.4 *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

$$G_i(x,z) = G_i(z',z) + KL(x,z'). \tag{13.10}$$

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 (13.10). 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 (13.10), we have

$$G_i(x,x^k) = G_i(x^{k+1},x^k) + KL(x,x^{k+1}).$$

Therefore,

$$KL(x,x^k) - KL(x,x^{k+1}) = G_i(x^{k+1},x^k) + m_i^{-1} KL(b_i,(Ax^k)_i). \tag{13.11}$$

From Equation (13.11) 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 (13.11) 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. ∎

13.5 Comments on the Rate of Convergence of MART

We can see from Equation (13.11),

$$KL(x, x^k) - KL(x, x^{k+1}) = G_i(x^{k+1}, x^k) + m_i^{-1} KL(b_i, (Ax^k)_i),$$

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 14

The Simultaneous MART (SMART)

There is a simultaneous version of MART, called the SMART [44, 56, 113]. As with MART, the SMART applies only to nonnegative systems. Unlike MART, SMART uses all equations in each step of the iteration.

14.1 The SMART Iteration

It begins with a positive vector x^0; having calculated x^k, we calculate x^{k+1} using

$$\log x_j^{k+1} = \log x_j^k + s_j^{-1} \sum_{i=1}^{I} A_{ij} \log \frac{b_i}{(Ax^k)_i}, \tag{14.1}$$

where $s_j = \sum_{i=1}^{I} A_{ij} > 0$.

The following theorem describes what we know concerning the SMART.

Theorem 14.1 *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 $KL(Ax, b)$ 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 $KL(Ax, b)$ and at most $I - 1$ of its entries are nonzero.*

When there are nonnegative solutions of $Ax = b$, both MART and SMART converge to the nonnegative solution minimizing the Kullback-Leibler distance $KL(x, x^0)$; if x^0 is the vector whose entries are all one,

then the solution minimizes the Shannon entropy, $SE(x)$, given by

$$SE(x) = \sum_{j=1}^{J} x_j \log x_j - x_j. \tag{14.2}$$

One advantage that SMART has over MART is that, if the nonnegative system $Ax = b$ has no nonnegative solutions, the SMART converges to the nonnegative minimizer of the function $KL(Ax, b)$ for which $KL(x, x^0)$ is minimized. One disadvantage of SMART, compared to MART, is that it is slow.

14.2 The SMART as a Generalized Projection Method

As we saw previously, the MART algorithm can be viewed as a sequential, relaxed generalized projection method that involves the weighted KL projections Q_i. In this section we show that the SMART iteration can be viewed in this way also.

Recall that, for any nonnegative vector x, the nonnegative vector $z = Q_i x$ given by

$$z_j = (Q_i x)_j = x_j \frac{b_i}{(Ax)_i}$$

minimizes the weighted KL distance

$$\sum_{j=1}^{J} A_{ij} KL(x_j, z_j),$$

over all nonnegative z with $(Az)_i = b_i$. Given x^k, we take as x^{k+1} the vector whose entries x_j^{k+1} are weighted geometric means of the $(Q_i x^k)_j$; that is,

$$\log x_j^{k+1} = \sum_{i=1}^{I} s_j^{-1} A_{ij} \log(Q_i x^k)_j,$$

with $s_j = \sum_{i=1}^{I} A_{ij} > 0$. We then have

$$x_j^{k+1} = x_j^k \exp(\sum_{i=1}^{I} s_j^{-1} A_{ij} \log \frac{b_i}{(Ax^k)_i}),$$

or

$$x_j^{k+1} = x_j^k \prod_{i=1}^{I} (\frac{b_i}{(Ax^k)_i})^{s_j^{-1} A_{ij}}.$$

This is the SMART iterative step.

14.3 Proof of Convergence of the SMART

For the consistent case, in which there are nonnegative solutions of $A = b$, the proof of convergence of SMART is almost the same as that for MART given previously. To simplify the notation, we shall assume that we have normalize the problem so that the sums of the entries in each column of A is one. That means we replace each A_{ij} with $s_j^{-1}A_{ij}$ and each x_j with $s_j x_j$. Instead of $G_i(x, z)$, use

$$G(x, z) = KL(x, z) - KL(Ax, Az) + KL(Ax, b).$$

It follows from our assumption about normalization and Equation (13.5) that

$$KL(x, z) - KL(Ax, Az) \geq 0,$$

so $G(x, z) \geq 0$ for all nonnegative x and z. Notice that

$$G(x, x) = KL(Ax, b), \tag{14.3}$$

so that

$$G(x, z) = G(x, x) + KL(x, z) - KL(Ax, Az),$$

and $G(x, z)$ is minimized, as a function of z, by the choice $z = x$. Minimizing $G(x, z)$ with respect to x, for fixed z, as we did for MART, we find that

$$G(x, z) = G(z', z) + KL(x, z'), \tag{14.4}$$

for z' given by

$$z'_j = z_j \prod_{i=1}^{I} (\frac{b_i}{(Az)_i})^{A_{ij}}.$$

Notice that the SMART iteration, in the normalized case, is

$$x^{k+1} = (x^k)'.$$

We complete the convergence proof through several exercises. In completing these exercises, it will be helpful to study the related results used in the convergence proof of MART.

Exercise 14.1 *Show that the sequence* $\{KL(Ax^k, b)\}$ *is decreasing and the sequence* $\{KL(x^k, x^{k+1})\}$ *converges to zero. Hint: use Equations (14.3) and (14.4).*

Exercise 14.2 *Show that the sequence* $\{x^k\}$ *is bounded, by showing that*

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

Exercise 14.3 *From the previous exercise, we know that the sequence* $\{x^k\}$ *has cluster points; let* x^* *be one of them. Show that* $(x^*)' = x^*$. *Hint: use the fact that* $\{KL(x^k, x^{k+1})\}$ *converges to zero.*

Exercise 14.4 *Let* $x = \hat{x} \geq 0$ *minimize* $KL(Ax, b)$, *over all nonnegative vectors* x. *Show that* $(\hat{x})' = \hat{x}$.

Exercise 14.5 *Show that, for the SMART sequence* $\{x^k\}$ *with cluster point* x^* *and* $\hat{x}$ *as defined previously, we have*

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

$$KL(A\hat{x}, Ax^k) + KL(x^{k+1}, x^k) - KL(Ax^{k+1}, Ax^k), \tag{14.5}$$

and so $KL(A\hat{x}, Ax^*) = 0$, *the sequence* $\{KL(\hat{x}, x^k)\}$ *is decreasing and* $KL(\hat{x}, x^*) < +\infty$.

Exercise 14.6 *Show that, for any cluster point* x^* *of the sequence* $\{x^k\}$, *we have*

$$KL(A\hat{x}, b) = KL(Ax^*, b),$$

so that x^* *is a nonnegative minimizer of* $KL(Ax, b)$. *Consequently, the sequence* $\{KL(x^*, x^k)\}$ *converges to zero, the sequence* $\{x^k\}$ *converges to* x^*, *and*

$$KL(\hat{x}, x^0) \geq KL(x^*, x^0).$$

14.4 Remarks on the Rate of Convergence of the SMART

In the consistent case, the progress we make toward a solution, using the SMART, is described by Equation (14.5), which now says

$$KL(\hat{x}, x^k) - KL(\hat{x}, x^{k+1})$$

$$= KL(Ax^{k+1}, b) + KL(b, Ax^k) + KL(x^{k+1}, x^k) - KL(Ax^{k+1}, Ax^k).$$

It follows that

$$KL(\hat{x}, x^k) - KL(\hat{x}, x^{k+1}) \geq KL(b, Ax^k).$$

While this is not an equality, it suggests that the improvement we make with each step is on the order of $KL(A\hat{x}, Ax^k)$. In the MART case, the improvement we make with each step is

$$KL(\hat{x}, x^k) - KL(\hat{x}, x^{k+1}) \geq m_i^{-1} KL(b_i, (Ax^k)_i).$$

Since we are assuming that the columns of A sum to one, the individual entries will be on the order of $\frac{1}{I}$, if all the entries are roughly the same size, so that m_i is then on the order of $\frac{1}{I}$. This indicates that the MART makes about as much progress toward a solution in one step (which means using a single equation), as SMART makes using one step (which means using all the equations). Said another way, the progress made in one pass through all the data using MART is about I times better than in one iteration of SMART, and yet involves about the same amount of calculation. Of course, this is a rough estimate, but it does correspond to what we typically observe in practice. If, however, the matrix A is sparse and has, say, only about $\sqrt{I}$ non-zero entries per column, then each entry is roughly $\frac{1}{\sqrt{I}}$, and m_i^{-1} is on the order of $\sqrt{I}$. In such cases, the progress made in one pass through all the data using MART is about $\sqrt{I}$ times better than in one iteration of SMART, and yet involves about the same amount of calculation.

14.5 Block-Iterative SMART

As we just argued, there is good empirical, as well as theoretical, justification for the claim that MART converges, in the consistent case, significantly faster than SMART. On the other hand, the SMART can be implemented in parallel, which will accelerate the computation time. Because the MART uses only a single equation at each step, it does not take advantage of the computer architecture. A compromise between being purely sequential and being purely simultaneous might provide the best solution. Such a method is a *block-iterative method*.

Block-iterative methods involve a partition of the index set $\{i = 1, ..., I\}$ into nonempty subsets B_n, $n = 1, 2, ..., N$. For $k = 0, 1, 2, ...$, and $n(k) = k(\mathrm{mod}\, N) + 1$, only the equations corresponding to i in the set B_n are used to calculate x^{k+1} from x^k. The ART and MART are extreme examples of block-iterative algorithms, in which $N = I$ and $B_n = B_i = \{i\}$, for each i.

The SMART algorithm involves a summation over $i = 1, ..., I$ at each step. Block-iterative SMART algorithms replace this sum with a sum only over those i in the current block.

14.5.1 The Rescaled Block-Iterative SMART

Both the MART and SMART involve weighted geometric means of the generalized projections Q_i; MART involves relaxation, as well, while SMART does not. The block-iterative SMART algorithms can also be written in terms of such relaxed weighted geometric means. The *rescaled block-iterative* SMART (RBI-SMART) also uses a particular choice of a parameter designed to accelerate the convergence in the consistent case.

The vector x^{k+1} determined by the RBI-SMART is the following:

$$x_j^{k+1} = (x_j^k)^{1-m_n^{-1}s_j^{-1}s_{nj}} \prod_{i \in B_n} [x_j^k \frac{b_i}{(Ax^k)_i}]^{m_n^{-1}s_j^{-1}A_{ij}},$$

where

$$s_{nj} = \sum\nolimits_{i \in B_n} A_{ij},$$

and

$$m_n = \max\{s_{nj}s_j^{-1} | \, j = 1, ..., J\}.$$

Consequently, x_j^{k+1} is a weighted geometric mean of x_j^k and the $(Q_i x^k)_j$ for i in the block B_n.

The RBI-SMART converges, in the consistent case, to the same solution as MART and SMART, for all choices of blocks. The proof is similar to that for MART and SMART and we leave it as an exercise for the reader. There are variants of the RBI-SMART that involve other parameters [33].

As with ART and MART, the RBI-SMART does not converge to a single vector in the inconsistent case. What is always observed is that RBI-SMART exhibits subsequential convergence to a limit cycle. There is no proof of this, however.

Chapter 15

Expectation Maximization Maximum Likelihood (EMML)

For nonnegative systems $Ax = b$ in which the column sums of A and the entries of b are positive, the expectation maximization maximum likelihood (EMML) method produces a nonnegative solution of $Ax = b$, whenever one exists [20, 21, 33, 53, 101, 115, 93, 120, 94] . If not, the EMML converges to a nonnegative approximate solution that minimizes the function $KL(b, Ax)$ [20, 22, 33, 53, 120].

15.1 The EMML Iteration

As we saw previously, the iterative step in the SMART involves a weighted geometric mean of the weighted KL projections $Q_i x^k$: for the SMART we have

$$\log x_j^{k+1} = s_j^{-1} \sum_{i=1}^{I} A_{ij} \log(Q_i x^k)_j.$$

It would be nice if we could avoid the exponentiation required in the SMART iterative step. This suggests the algorithm in which the entries x_j^{k+1} are weighted arithmetic means of the $(Q_i x^k)_j$; that is, the iterative step should be

$$x_j^{k+1} = s_j^{-1} \sum_{i=1}^{I} A_{ij} (Q_i x^k)_j,$$

which can be written as

$$x_j^{k+1} = x_j^k s_j^{-1} \sum_{i=1}^{I} A_{ij} \frac{b_i}{(Ax^k)_i}. \tag{15.1}$$

This is the iterative step of the EMML algorithm.

The EMML algorithm was not originally derived from the SMART algorithm, but from a general method for likelihood maximization in statistics, the *expectation maximization* (EM) approach [58]. The EMML algorithm we study here is the EM method, as it applies to the case in which the data b_i are instances of independent Poisson random variables with mean values $(Ax)_i$; here the entries of x are the parameters to be estimated.

For the EMML algorithm the main results are the following.

Theorem 15.1 *In the consistent case the EMML algorithm 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.*

An open question about the EMML algorithm is the following:

Open Question: How does the EMML limit depend on the starting vector x^0? In particular, when there are nonnegative exact solutions of $Ax = b$, which one does the EMML produce and how does it depend on x^0?

15.2 Proof of Convergence of the EMML Algorithm

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 section 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 distance

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

will play an important role in the proof that follows. Note that if there is nonnegative x with $r(x) = q(x)$ then $b = Ax$.

15.2.1 Some Pythagorean Identities Involving the KL Distance

The EMML iterative algorithm is derived using the principle of *alternating minimization*, according to which the distance $KL(r(x), q(z))$ is 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 15.1 *Establish the following Pythagorean identities:*

$$KL(r(x), q(z)) = KL(r(z), q(z)) + KL(r(x), r(z)); \tag{15.2}$$

$$KL(r(x), q(z)) = KL(r(x), q(x')) + KL(x', z), \tag{15.3}$$

for

$$x'_j = x_j \sum_{i=1}^{I} A_{ij} \frac{b_i}{(Ax)_i}. \tag{15.4}$$

Note that it follows from normalization and Equation (13.5) that $KL(x, z) - KL(Ax, Az) \geq 0$.

Exercise 15.2 *Show that, for $\{x^k\}$ given by Equation (15.1), $\{KL(b, Ax^k)\}$ is decreasing and $\{KL(x^{k+1}, x^k)\} \to 0$. Hint: Use $KL(r(x), q(x)) = KL(b, Ax)$, and the Pythagorean identities.*

Exercise 15.3 *Show that the EMML sequence $\{x^k\}$ is bounded by showing*

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

Exercise 15.4 *Show that $(x^*)' = x^*$ for any cluster point x^* of the EMML sequence $\{x^k\}$. Hint: Use the fact that $\{KL(x^{k+1}, x^k)\} \to 0$.*

Exercise 15.5 *Let $\hat{x}$ minimize $KL(b, Ax)$ over all $x \geq \mathbf{0}$. Then, $(\hat{x})' = \hat{x}$. Hint: Apply Pythagorean identities to $KL(r(\hat{x}), q(\hat{x}))$.*

Note that, because of convexity properties of the KL distance, even if the minimizer $\hat{x}$ is not unique, the vector $A\hat{x}$ is unique.

Exercise 15.6 *Show that, for the EMML sequence $\{x^k\}$ with cluster point x^* and $\hat{x}$ as defined previously, we have the double inequality*

$$KL(\hat{x}, x^k) \geq KL(r(\hat{x}), r(x^k)) \geq KL(\hat{x}, x^{k+1}), \tag{15.5}$$

from which we conclude that the sequence $\{KL(\hat{x}, x^k)\}$ is decreasing and $KL(\hat{x}, x^*) < +\infty$. Hints: For the first inequality calculate $KL(r(\hat{x}), q(x^k))$ in two ways. For the second one, use $(x)'_j = \sum_{i=1}^{I} r(x)_{ij}$ and Exercise 13.1.

Exercise 15.7 *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 $\{x^k\} \to x^*$. Hint: Use the double inequality of Equation (15.5) and $KL(r(\hat{x}), q(x^*))$.*

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.

15.3 Block-Iterative EMML Iteration

Block-iterative versions of ART and SMART have been known for decades. In contrast, the first block-iterative variant of the EMML algorithm, the *ordered-subset* EM (OSEM) [85], was discovered in 1994. The main idea in the OSEM is simply to replace all the sums over all the indices i with sums only over those i in the current block. This is not quite right; it ignores the relaxation that we have seen in the MART and RBI-SMART. The OSEM was shown to converge, in the consistent case, only when the matrix A satisfies a quite restrictive condition, *subset balance*. This means that the sums

$$s_{nj} = \sum_{i \in B_n} A_{ij}$$

depend only on n, and not on j.

The *rescaled block-iterative* EMML (RBI-EMML) corrects this omission. It has the iterative step

$$x_j^{k+1} = (1 - m_n^{-1} s_j^{-1} s_{nj}) x_j^k + m_n^{-1} s_j^{-1} x_j^k \sum_{i \in B_n} A_{ij} \frac{b_i}{(Ax^k)_i}. \tag{15.6}$$

The RBI-EMML converges, in the consistent case, for any choice of blocks.

Open Question: When there are multiple nonnegative solutions of $Ax = b$, the RBI-EMML solution will depend on the starting vector, x^0, but precisely how is unknown. Simulations seem to show that the solution may also vary with the choice of blocks, as well as with their ordering. How?

15.3.1 A Row-Action Variant of EMML

The MART is the row-action, or sequential, variant of RBI-SMART. There is also a row-action variant of EMML, obtained by selecting $N = I$ and taking $B_n = B_i = \{i\}$ as the blocks. This row-action variant has been called the EM-MART [33]. The EM-MART has the iterative step

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

for $m_i = \max\{A_{ij} s_j^{-1}\}$. Note that another version of EM-MART has the iterative step

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

for $m_i = \max\{A_{ij}\}$. The second convergent version looks more like MART, while the first follows directly from the RBI-EMML formula.

Chapter 16

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.

16.1 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.

16.2 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:

$$x_j^{k+1} = x_j^k \exp\left(s_j^{-1} \sum_{i=1}^{I} A_{ij} \log \frac{b_i}{(Ax^k)_i}\right) \tag{16.1}$$

while that for the EMML method is

EMML:

$$x_j^{k+1} = x_j^k s_j^{-1} \sum_{i=1}^{I} A_{ij} \frac{b_i}{(Ax^k)_i}. \tag{16.2}$$

The main results concerning the SMART is given by the following theorem.

Theorem 16.1 *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 16.2 *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

$$x_j = x_j^0 \exp\Big(\sum_{i=1}^{I} A_{ij}\lambda_i\Big) \tag{16.3}$$

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 16.1 *Minimizing $KL(d, x)$ over x as in Equation (16.3) 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

$$L(\lambda) = \prod_{j=1}^{J} p_j^{d_j} \tag{16.4}$$

so that the log-likelihood function is

$$LL(\lambda) = \sum_{j=1}^{J} d_j \log p_j. \tag{16.5}$$

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.

16.3 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 $n = k(\mathrm{mod}\, N)+1$, having found x^k let

OSEM:

$$x_j^{k+1} = x_j^k s_{nj}^{-1} \sum_{i \in B_n} A_{ij} \frac{b_i}{(Ax^k)_i}. \tag{16.6}$$

The OSSMART has the following iterative step:

OSSMART

$$x_j^{k+1} = x_j^k \exp\Big(s_{nj}^{-1} \sum_{i \in B_n} A_{ij} \log \frac{b_i}{(Ax^k)_i}\Big). \tag{16.7}$$

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.

16.4 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, [56] and by Schmidlin [113] in medical imaging. Block-iterative versions of SMART are also treated in [56], 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

$$x_j^{k+1} = x_j^k \exp\Big(\beta_{nj} \sum_{i \in B_n} \alpha_{ni} A_{ij} \log\Big(\frac{b_i}{(Ax^k)_i}\Big)\Big), \tag{16.8}$$

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

$$\gamma_j \delta_n \sigma_{nj} \leq 1, \tag{16.9}$$

for $\sigma_{nj} = \sum_{i \in B_n} \alpha_{ni} A_{ij}$. With these conditions satisfied we have the following result.

Theorem 16.3 *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 (16.8) 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 16.2 *Let $b = Ax$ for some nonnegative x. Then for $\{x^k\}$ as in Equation (16.8) we have*

$$\sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^k) - \sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^{k+1}) \geq$$

$$\delta_n \sum_{i \in B_n} \alpha_{ni} KL(b_i, (Ax^k)_i). \tag{16.10}$$

Proof: First note that

$$x_j^{k+1} = x_j^k \exp\Big(\gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \log\Big(\frac{b_i}{(Ax^k)_i}\Big)\Big), \tag{16.11}$$

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

$$\sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^k) - \sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^{k+1})$$

$$= \sum_{j=1}^{J} \gamma_j^{-1} (x_j \log(x_j^{k+1}/x_j^k) + x_j^k - x_j^{k+1})$$

$$= \sum_{j=1}^{J} x_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \log \frac{b_i}{(Ax^k)_i} + \sum_{j=1}^{J} \gamma_j^{-1} (x_j^k - x_j^{k+1})$$

$$= \delta_n \sum_{i \in B_n} \alpha_{ni} (\sum_{j=1}^{J} x_j A_{ij}) \log \frac{b_i}{(Ax^k)_i} + \sum_{j=1}^{J} \gamma_j^{-1} (x_j^k - x_j^{k+1})$$

$$\geq \delta_n \Big(\sum_{i \in B_n} \alpha_{ni} (b_i \log \frac{b_i}{(Ax^k)_i} + (Ax^k)_i - b_i) = \delta_n \sum_{i \in B_n} \alpha_{ni} KL(b_i, (Ax^k)_i).$$

This completes the proof of the lemma. ∎

From the inequality (16.10) 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 (16.10) 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 (16.10) 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

$$\delta_n = 1/\min\{\sigma_{nj}\gamma_j,\, j = 1, ..., J\}. \tag{16.12}$$

The *rescaled* BI-SMART (RBI-SMART) as presented in [22, 24, 25] 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 (16.7) 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 [44] 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 (16.1) uses $N = 1$, $\gamma_j = s_j^{-1}$ and $\alpha_{ni} = \alpha_i = 1$. Clearly the inequality (16.9) 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 [74], 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:

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i}\Big)^{m_i^{-1}A_{ij}}, \tag{16.13}$$

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 [82].

Darroch and Ratcliff included a discussion of a block-iterative version of SMART in their 1972 paper [56]. 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 [85], 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.

16.5 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

$$x_j^{k+1} = x_j^k(1 - \beta_{nj}\sigma_{nj}) + x_j^k\beta_{nj}\sum_{i \in B_n} \alpha_{ni}A_{ij}\frac{b_i}{(Ax^k)_i}, \tag{16.14}$$

for $j = 1, 2, ..., J$, $n = k(\mathrm{mod}\, N)+1$ and β_{nj} and α_{ni} positive. As in the case of BI-SMART, our convergence proof will require that β_{nj} be separable, that is,

$$b_{nj} = \gamma_j\delta_n$$

for each j and n and that the inequality (16.9) hold. With these conditions satisfied we have the following result.

Theorem 16.4 *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 (16.8) 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 16.3 *Let $b = Ax$ for some nonnegative x. Then for $\{x^k\}$ as in Equation (16.14) we have*

$$\sum_{j=1}^J \gamma_j^{-1}KL(x_j, x_j^k) - \sum_{j=1}^J \gamma_j^{-1}KL(x_j, x_j^{k+1}) \geq$$

$$\delta_n \sum_{i \in B_n} \alpha_{ni} KL(b_i, (Ax^k)_i). \tag{16.15}$$

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

$$\sum_{j=1}^J \gamma_j^{-1} x_j \log(x_j^{k+1}/x_j^k) \geq \delta_n \sum_{i \in B_n} \alpha_{ni} (\sum_{j=1}^J x_j A_{ij}) \log \frac{b_i}{(Ax^k)_i}.$$

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 (16.15) 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 (16.15) 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 (16.15) 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

$$\delta_n = 1/\min\{\sigma_{nj}\gamma_j,\, j = 1, ..., J\}. \tag{16.16}$$

The *rescaled* BI-EMML (RBI-EMML) as presented in [22, 24, 25] 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 (16.6) 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 (16.2) uses $N = 1$, $\gamma_j = s_j^{-1}$ and $\alpha_{ni} = \alpha_i = 1$. Clearly the inequality (16.9) 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:

$$x_j^{k+1} = (1 - \delta_i \gamma_j \alpha_{ii} A_{ij}) x_j^k + \delta_i \gamma_j \alpha_{ii} A_{ij} \frac{b_i}{(Ax^k)_i}, \tag{16.17}$$

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* $\text{LC} = \{x^{\infty,i} | i = 1, ..., I\}$ 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 [19]. 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:

$$x_j^{k+1} = (1 - \lambda_k \sum^n A_{ij}) x_j^k + \lambda_k x_j^k \sum^n A_{ij} \Big(\frac{b_i}{(Ax^k)_i} \Big), \tag{16.18}$$

where the positive relaxation parameters λ_k are chosen to converge to zero and $\sum_{k=0}^{+\infty} \lambda_k = +\infty$.

16.6 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 (16.11) 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 16.4 *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*

$$KL(a, b) = KL(a_+, b_+) + KL(a_+, \frac{a_+}{b_+} b). \tag{16.19}$$

For nonnegative vectors x and z let

$$G_n(x, z) = \sum_{j=1}^J \gamma_j^{-1} KL(x_j, z_j)$$

$$+\delta_n \sum_{i i B_n} \alpha_{ni} [KL((Ax)_i, b_i) - KL((Ax)_i, (Pz)_i)]. \tag{16.20}$$

It follows from Lemma 16.19 and the inequality

$$\gamma_j^{-1} - \delta_n \sigma_{nj} \geq 1$$

that $G_n(x, z) \geq 0$ in all cases.

Lemma 16.5 *For every x we have*

$$G_n(x, x) = \delta_n \sum_{i \in B_n} \alpha_{ni} KL((Ax)_i, b_i) \tag{16.21}$$

so that

$$G_n(x,z) = G_n(x,x) + \sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, z_j)$$

$$-\delta_n \sum_{i \in B_n} \alpha_{ni} KL((Ax)_i, (Pz)_i). \tag{16.22}$$

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

$$x_j = z'_j = z_j \exp\Big(\gamma_j \delta_n \sum_{i \in B_n} \alpha_{ni} A_{ij} \log \frac{b_i}{(Pz)_i}\Big). \tag{16.23}$$

Lemma 16.6 *For each x and z we have*

$$G_n(x,z) = G_n(z',z) + \sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, z'_j). \tag{16.24}$$

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 16.24 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

$$\sum_{j=1}^{J} \gamma_j^{-1} KL(u_j, x_j^k) - \sum_{j=1}^{J} \gamma_j^{-1} KL(u_j, x_j^{k+1})$$

$$= G_n(x^{k+1}, x^k) + \delta_n \sum_{i \in B_n} \alpha_{ni} KL(b_i, (Ax^k)_i). \tag{16.25}$$

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 (16.25) 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.

Part V

Stability

Chapter 17

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, [34]).

17.1 Where Does Sensitivity Come From?

We illustrate the sensitivity problem that can arise when the inconsistent system $Ax = b$ has more equations than unknowns and we calculate the least-squares solution,

$$x_{LS} = (A^\dagger A)^{-1} A^\dagger b,$$

assuming that the Hermitian, nonnegative-definite matrix $Q = (A^\dagger A)$ is invertible, and therefore positive-definite.

The matrix Q has the eigenvalue/eigenvector decomposition

$$Q = \lambda_1 u_1 u_1^\dagger + \cdots + \lambda_I u_I u_I^\dagger,$$

where the (necessarily positive) eigenvalues of Q are

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_I > 0,$$

and the vectors u_i are the corresponding orthogonal eigenvectors.

17.1.1 The Singular-Value Decomposition of A

The square roots $\sqrt{\lambda_i}$ are called the *singular values* of A. The *singular-value decomposition* (SVD) of A is similar to the eigenvalue/eigenvector decomposition of Q: we have

$$A = \sqrt{\lambda_1} u_1 v_1^\dagger + \cdots + \sqrt{\lambda_I} u_I v_I^\dagger,$$

where the v_i are particular eigenvectors of $AA^\dagger$. We see from the SVD that the quantitites $\sqrt{\lambda_i}$ determine the relative importance of each term $u_i v_i^\dagger$.

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 th 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.

17.1.2 The Inverse of $Q = A^\dagger A$

The inverse of Q can then be written

$$Q^{-1} = \lambda_1^{-1} u_1 u_1^\dagger + \cdots + \lambda_I^{-1} u_I u_I^\dagger,$$

so that, with $A^\dagger b = c$, we have

$$x_{LS} = \lambda_1^{-1} (u_1^\dagger c) u_1 + \cdots + \lambda_I^{-1} (u_I^\dagger c) u_I.$$

Because the eigenvectors are orthogonal, we can express $||A^\dagger b||_2^2 = ||c||_2^2$ as

$$||c||_2^2 = |u_1^\dagger c|^2 + \cdots + |u_I^\dagger c|^2,$$

and $||x_{LS}||_2^2$ as

$$||x_{LS}||_2^2 = \lambda_1^{-1} |u_1^\dagger c|^2 + \cdots + \lambda_I^{-1} |u_I^\dagger c|^2.$$

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 $|u_i^\dagger c|$ relatively small, for the indices

near I, keeping the product $\lambda_i^{-1}|u_i^\dagger c|^2$ 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.

17.1.3 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

$$F_\epsilon(x) = (1 - \epsilon)\|Ax - b\|_2^2 + \epsilon\|x - p\|_2^2. \tag{17.1}$$

Exercise 17.1 *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, A is a prior estimate of the desired solution. Note that the inverse above always exists.

Note that, if $p = 0$, then

$$\hat{x}_\epsilon = (A^\dagger A + \gamma^2 I)^{-1} A^\dagger b, \tag{17.2}$$

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 17.2 *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 17.3 *Show that any vector A in R^J can be written as $A = A^\dagger q + r$, 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: $N \leq M, A^\dagger A$ is invertible; or

Case 2: $N > M, AA^\dagger$ is invertible.

Exercise 17.4 *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 $A = A^\dagger q + r$, 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 17.5 (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 A. 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.

17.2 Iterative Regularization in ART

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 (17.1). For the case of $p = 0$, the solution to this problem is the vector $\hat{x}_\epsilon$ in Equation (17.2). 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 ART algorithm.

We discuss two methods for using ART to obtain regularized solutions of $Ax = b$. The first one is presented in [34], while the second one is due to Eggermont, Herman, and Lent [66].

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

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

We begin with $u^0 = b$ and $v^0 = 0$.

Exercise 17.6 *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

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

We begin at $x^0 = 0$ and $v^0 = 0$.

Exercise 17.7 *Show that the limit vector has for its upper component* $x^\infty = \hat{x}_\epsilon$ *as before, and that* $\gamma v^\infty = b - A\hat{x}_\epsilon$.

17.3 A Bayesian View of Reconstruction

The EMML iterative algorithm maximizes the likelihood function for the case in which the entries of the data vector $b = (b_1, ..., b_I)^T$ are assumed to be samples of independent Poisson random variables with mean values $(Ax)_i$; here, A is an I by J matrix with nonnegative entries and

$x = (x_1, ..., x_J)^T$ is the vector of nonnegative parameters to be estimated. Equivalently, it minimizes the Kullback-Leibler distance $KL(b, Ax)$. This situation arises in single photon emission tomography, where the b_i are the number of photons counted at each detector i, x is the vectorized image to be reconstructed and its entries x_j are (proportional to) the radionuclide intensity levels at each voxel j. When the signal-to-noise ratio is low, which is almost always the case in medical applications, maximizing likelihood can lead to unacceptably noisy reconstructions, particularly when J is larger than I. One way to remedy this problem is simply to halt the EMML algorithm after a few iterations, to avoid over-fitting the x to the noisy data. A more mathematically sophisticated remedy is to employ a Bayesian approach and seek a maximum *a posteriori* (MAP) estimate of x.

In the Bayesian approach we view x as an instance of a random vector having a probability density function $f(x)$. Instead of maximizing the likelihood given the data, we now maximize the posterior likelihood, given both the data and the prior distribution for x. This is equivalent to minimizing

$$F(x) = KL(b, Ax) - \log f(x). \tag{17.3}$$

The EMML algorithm is an example of an optimization method based on alternating minimization of a function $H(x, z) > 0$ of two vector variables. The alternating minimization works this way: let x and z be vector variables and $H(x, z) > 0$. If we fix z and minimize $H(x, z)$ with respect to x, we find that the solution is $x = z$, the vector we fixed; that is,

$$H(x, z) \geq H(z, z)$$

always. If we fix x and minimize $H(x, z)$ with respect to z, we get something new; call it Tx. The EMML algorithm has the iterative step $x^{k+1} = Tx^k$.

Obviously, we can't use an arbitrary function H; it must be related to the function $KL(b, Ax)$ that we wish to minimize, and we must be able to obtain each intermediate optimizer in closed form. The clever step is to select $H(x, z)$ so that $H(x, x) = KL(b, Ax)$, for any x. Now see what we have so far:

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

That tells us that the algorithm makes $KL(b, Ax^k)$ decrease with each iteration. The proof doesn't stop here, but at least it is now plausible that the EMML iteration could minimize $KL(b, Ax)$.

The function $H(x, z)$ used in the EMML case is the KL distance

$$H(x, z) = KL(r(x), q(z)) = \sum_{i=1}^{I} \sum_{j=i}^{J} KL(r(x)_{ij}, q(z)_{ij}); \tag{17.4}$$

we define, for each nonnegative vector x for which $(Ax)_i = \sum_{j=1}^{J} A_{ij}x_j > 0$, the arrays $r(x) = \{r(x)_{ij}\}$ and $q(x) = \{q(x)_{ij}\}$ with entries

$$r(x)_{ij} = x_j A_{ij} \frac{b_i}{(Ax)_i}$$

and

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

With $x = x^k$ fixed, we minimize with respect to z to obtain the next EMML iterate x^{k+1}. Having selected the prior pdf $f(x)$, we want an iterative algorithm to minimize the function $F(x)$ in Equation (17.3). It would be a great help if we could mimic the alternating minimization formulation and obtain x^{k+1} by minimizing

$$KL(r(x^k), q(z)) - \log f(z) \tag{17.5}$$

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.

17.4 The Gamma Prior Distribution for x

In [94] 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 17.8 *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 (17.5) with respect to z is equivalent to minimizing the function*

$$KL(r(x^k), q(z)) + \sum_{j=1}^{J} \delta_j KL(\gamma_j, z_j), \tag{17.6}$$

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

$$x_j^{k+1} = \frac{\delta_j}{\delta_j + s_j}\gamma_j + \frac{1}{\delta_j + s_j} x_j^k \sum_{i=1}^{I} A_{ij} b_i / (Ax^k)_i, \tag{17.7}$$

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

We see from Equation (17.7) 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 [94]; see also [20].

17.5 The One-Step-Late Alternative

It may well happen that we do not wish to use the gamma priors model and prefer some other $f(x)$. Because we will not be able to find a closed form expression for the z minimizing the function in line (17.5), we need some other way to proceed with the alternating minimization. Green [75] has offered the *one-step-late* (OSL) alternative.

When we try to minimize the function in line (17.5) 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 [29] an interior point algorithm (IPA) is presented that avoids the OSL issue. In [105] the IPA is used to regularize transmission tomographic images.

17.6 Regularizing the SMART

The SMART algorithm is not derived as a maximum likelihood method, so regularized versions do not take the form of MAP algorithms. Nevertheless, in the presence of noisy data, the SMART algorithm suffers from the same problem that afflicts the EMML, overfitting to noisy data resulting in an unacceptably noisy image. As we saw earlier, there is a close connection between the EMML and SMART algorithms. This suggests that a regularization method for SMART can be developed along the lines of the MAP with gamma priors used for EMML. Since the SMART is obtained by minimizing the function $KL(q(z), r(x^k))$ with respect to z to obtain x^{k+1},

it seems reasonable to attempt to derive a regularized SMART iterative scheme by minimizing

$$KL(q(z), r(x^k)) + \sum_{j=1}^{J} \delta_j KL(z_j, \gamma_j), \tag{17.8}$$

for selected positive parameters δ_j and γ_j.

Exercise 17.9 *Show that the z_j minimizing the function in line (17.8) can be expressed in closed form and that the resulting x^{k+1} has entries that satisfy*

$$\log x_j^{k+1} = \frac{\delta_j}{\delta_j + s_j} \log \gamma_j + \frac{1}{\delta_j + s_j} x_j^k \sum_{i=1}^{I} A_{ij} \log[b_i/(Ax^k)_i]. \tag{17.9}$$

In [20] it was shown that this iterative sequence converges to a minimizer of the function

$$KL(Ax, y) + \sum_{j=1}^{J} \delta_j KL(x_j, \gamma_j).$$

It is useful to note that, although it may be possible to rederive this minimization problem within the framework of Bayesian MAP estimation by carefully selecting a prior pdf for the vector x, we have not done so. The MAP approach is a special case of regularization through the use of penalty functions. These penalty functions need not arise through a Bayesian formulation of the parameter-estimation problem.

17.7 De Pierro's Surrogate-Function Method

In [59] De Pierro presents a modified EMML algorithm that includes regularization in the form of a penalty function. His objective is the same as ours was in the case of regularized SMART: to embed the penalty term in the alternating minimization framework in such a way as to make it possible to obtain the next iterate in closed form. Because his *surrogate function* method has been used subsequently by others to obtain penalized likelihood algorithms [46], 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 (17.4), we require that if we fix z and minimize $H(x, z)$ with respect to x, the solution should be $x = z$, the vector we fixed; that is, $H(x, z) \geq H(z, z)$ always. If we fix x and minimize $H(x, z)$ with respect to z, we should get something new; call it Tx. As with the EMML, the algorithm will have the iterative step $x^{k+1} = Tx^k$.

Summarizing, we see that we need a function $H(x,z)$ with the properties (1) $H(x,z) \geq H(z,z)$ for all x and z; (2) $H(x,x)$ is the function $F(x)$ we wish to minimize; and (3) minimizing $H(x,z)$ with respect to z for fixed x is easy.

The function to be minimized is

$$F(x) = KL(b, Ax) + g(x),$$

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).

17.8 Block-Iterative Regularization

We saw previously that it is possible to obtain a regularized least-squares solution $\hat{x}_\epsilon$, and thereby avoid the limit cycle, using only the matrix A and the ART algorithm. This prompts us to ask if it is possible to find regularized SMART solutions using block-iterative variants of SMART. Similarly, we wonder if it is possible to do the same for EMML.

Open Question: Can we use the MART to find the minimizer of the function

$$KL(Ax,b) + \epsilon KL(x,p)?$$

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 (16.18) 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 [61] RAMLA is extended to a regularized version, but with no guarantee of convergence.

Chapter 18

Feedback in Block-Iterative Reconstruction

When the nonnegative system of linear equations $Ax = b$ has no nonnegative solutions we say that we are in the *inconsistent case*. In this case the SMART and EMML algorithms still converge, to a nonnegative minimizer of $KL(Ax, b)$ and $KL(b, Ax)$, respectively. On the other hand, the rescaled block-iterative versions of these algorithms, RBI-SMART and RBI-EMML, do not converge. Instead they exhibit *cyclic subsequential convergence*; for each fixed $n = 1, ..., N$, with N the number of blocks, the subsequence $\{x^{mN+n}\}$ converges to their own limits. These limit vectors then constitute the *limit cycle* (LC). The LC for RBI-SMART is not the same as for RBI-EMML, generally, and the LC varies with the choice of blocks. Our problem is to find a way to calculate the SMART and EMML limit vectors using the RBI methods. More specifically, how can we calculate the SMART and EMML limit vectors from their associated RBI limit cycles?

As is often the case with the algorithms based on the KL distance, we can turn to the ART algorithm for guidance. What happens with the ART algorithm in the inconsistent case is often closely related to what happens with RBI-SMART and RBI-EMML, although proofs for the latter methods are more difficult to obtain. For example, when the system $Ax = b$ has no solution we can prove that ART exhibits cyclic subsequential convergence to a limit cycle. The same behavior is seen with the RBI methods, but no one knows how to prove this. When the system $Ax = b$ has no solution we usually want to calculate the least squares (LS) approximate solution. The problem then is to use the ART to find the LS solution. There are several ways to do this, as discussed in [24, 34]. We would like to be able

to borrow some of these methods and apply them to the RBI problem. In this section we focus on one specific method that works for ART and we try to make it work for RBI; it is the *feedback* approach.

18.1 Feedback in ART

Suppose that the system $Ax = b$ has no solution. We apply the ART and get the limit cycle $\{z^1, z^2, ..., z^I\}$, where I is the number of equations and $z^0 = z^I$. We assume that the rows of A have been normalized so that their lengths are equal to one. Then the ART iterative step gives

$$z_j^i = z_j^{i-1} + \overline{A_{ij}}(b_i - (Az^{i-1})_j)$$

or

$$z_j^i - z_j^{i-1} = \overline{A_{ij}}(b_i - (Az^{i-1})_j).$$

Summing over the index i and using $z^0 = z^I$ we obtain zero on the left side, for each j. Consequently $A^\dagger b = A^\dagger c$, where c is the vector with entries $c_i = (Az^{i-1})_i$. It follows that the systems $Ax = b$ and $Ax = c$ have the same LS solutions and that it may help to use both b and c to find the LS solution from the limit cycle. The article [24] contains several results along these lines. One approach is to apply the ART again to the system $Ax = c$, obtaining a new LC and a new candidate for the right side of the system of equations. If we repeat this *feedback* procedure, each time using the LC to define a new right side vector, does it help us find the LS solution? Yes, as Theorem 4 of [24] shows. Our goal in this section is to explore the possibility of using the same sort of feedback in the RBI methods. Some results in this direction are in [24]; we review those now.

18.2 Feedback in RBI methods

One issue that makes the KL methods more complicated than the ART is the support of the limit vectors, meaning the set of indices j for which the entries of the vector are positive. In [20] it was shown that when the system $Ax = b$ has no nonnegative solutions and A has the *full rank property* there is a subset S of $\{j = 1, ..., J\}$ with cardinality at most $I-1$, such that every nonnegative minimizer of $KL(Ax, b)$ has zero for its j-th entry whenever j is not in S. It follows that the minimizer is unique. The same result holds for the EMML, although it has not been proven that the set S is the same set as in the SMART case. The same result holds for the vectors of the LC for both RBI-SMART and RBI-EMML.

A simple, yet helpful, example to refer to as we proceed is the following.

$$A = \begin{bmatrix} 1 & .5 \\ 0 & .5 \end{bmatrix}, b = \begin{bmatrix} .5 \\ 1 \end{bmatrix}.$$

There is no nonnegative solution to this system of equations and the support set S for SMART, EMML and the RBI methods is $S = \{j = 2\}$.

18.2.1 The RBI-SMART

Our analysis of the SMART and EMML methods has shown that the theory for SMART is somewhat nicer than that for EMML and the resulting theorems for SMART are a bit stronger. The same is true for RBI-SMART, compared to RBI-EMML. For that reason we begin with RBI-SMART.

Recall that the iterative step for RBI-SMART is

$$x_j^{k+1} = x_j^k \exp(m_n^{-1} s_j^{-1} \sum_{i \in B_n} A_{ij} \log(b_i/(Ax^k)_i)),$$

where $n = k(\mathrm{mod}\, N) + 1$, $s_j = \sum_{i=1}^I A_{ij}$, $s_{nj} = \sum_{i \in B_n} A_{ij}$ and $m_n = \max\{s_{nj}/s_j,\, j = 1, ..., J\}$.

For each n let
$G_n(x, z) =$

$$\sum_{j=1}^J s_j KL(x_j, z_j) - m_n^{-1} \sum_{i \in B_n} KL((Ax)_i, (Az)_i) + m_n^{-1} \sum_{i \in B_n} KL((Ax)_i, b_i).$$

Exercise 18.1 *Show that*

$$\sum_{j=1}^J s_j KL(x_j, z_j) - m_n^{-1} \sum_{i \in B_n} KL((Ax)_i, (Az)_i) \geq 0,$$

so that $G_n(x, z) \geq 0$.

Exercise 18.2 *Show that*

$$G_n(x, z) = G_n(z', z) + \sum_{j=1}^J s_j KL(x_j, z'_j),$$

where

$$z'_j = z_j \exp(m_n^{-1} s_j^{-1} \sum_{i \in B_n} A_{ij} \log(b_i/(Az)_i).$$

We assume that there are no nonnegative solutions to the nonnegative system $Ax = b$. We apply the RBI-SMART and get the limit cycle $\{z^1, ..., z^N\}$, where N is the number of blocks. We also let $z^0 = z^N$ and for each i let $c_i = (Az^{n-1})_i$ where $i \in B_n$, the n-th block. Prompted by what we learned concerning the ART, we ask if the nonnegative minimizers of $KL(Ax, b)$ and $KL(Ax, c)$ are the same. This would be the correct question to ask if

we were using the slower unrescaled block-iterative SMART, in which the m_n are replaced by one. For the rescaled case it turns out that the proper question to ask is: Are the nonnegative minimizers of the functions

$$\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} KL((Ax)_i, b_i)$$

and

$$\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} KL((Ax)_i, c_i)$$

the same? The answer is "Yes, probably." The difficulty has to do with the support of these minimizers; specifically: Are the supports of both minimizers the same as the support of the LC vectors? If so, then we can prove that the two minimizers are identical. This is our motivation for the feedback approach.

The *feedback* approach is the following: beginning with $b^0 = b$ we apply the RBI-SMART and obtain the LC, from which we extract the vector c, which we also call c^0. We then let $b^1 = c^0$ and apply the RBI-SMART to the system $b^1 = Ax$. From the resulting LC we extract $c^1 = b^2$, and so on. In this way we obtain an infinite sequence of *data vectors* $\{b^k\}$. We denote by $\{z^{k,1}, ..., z^{k,N}\}$ the LC we obtain from the system $b^k = Ax$, so that

$$b_i^{k+1} = (Az^{k,n})_i, \text{ for } i \in B_n.$$

One issue we must confront is how we use the support sets. At the first step of feedback we apply RBI-SMART to the system $b = b^0 = Ax$, beginning with a positive vector x^0. The resulting limit cycle vectors are supported on a set S^0 with cardinality less than I. At the next step we apply the RBI-SMART to the system $b^1 = Ax$. Should we begin with a positive vector (not necessarily the same x^0 as before) or should our starting vector be supported on S^0?

Exercise 18.3 *Show that the RBI-SMART sequence $\{x^k\}$ is bounded. Hints: For each j let $M_j = \max\{b_i/A_{ij}\,, |A_{ij} > 0\}$ and let $C_j = \max\{x_j^0, M_j\}$. Show that $x_j^k \leq C_j$ for all k.*

Exercise 18.4 *Let S be the support of the LC vectors. Show that*

$$\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} A_{ij} \log(b_i/c_i) \leq 0 \tag{18.1}$$

for all j, with equality for those $j \in S$. Conclude from this that

$$\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} KL((Ax)_i, b_i) - \sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} KL((Ax)_i, c_i) \geq$$

$$\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} (b_i - c_i),$$

with equality if the support of the vector x lies within the set S. Hints: For $j \in S$ consider $\log(z_j^n / z_j^{n-1})$ and sum over the index n, using the fact that $z^N = z^0$. For general j assume there is a j for which the inequality does not hold. Show that there is M and $\epsilon > 0$ such that for $m \geq M$

$$\log(x_j^{(m+1)N} / x_j^{mN}) \geq \epsilon.$$

Conclude that the sequence $\{x_j^{mN}\}$ is unbounded.

Exercise 18.5 *Show that*

$$\sum_{n=1}^{N} G_n(z^{k,n}, z^{k,n-1}) = \sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} (b_i^k - b_i^{k+1}),$$

and conclude that the sequence $\{\sum_{n=1}^{N} m_n^{-1} (\sum_{i \in B_n} b_i^k)\}$ is decreasing and that the sequence $\{\sum_{n=1}^{N} G_n(z^{k,n}, z^{k,n-1})\} \to 0$ as $k \to \infty$. Hints: Calculate $G_n(z^{k,n}, z^{k,n-1})$ using Exercise (18.2).

Exercise 18.6 *Show that for all vectors $x \geq 0$ the sequence*

$$\{\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} KL((Ax)_i, b_i^k)\}$$

is decreasing and the sequence

$$\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} (b_i^k - b_i^{k+1}) \to 0,$$

as $k \to \infty$. Hints: Calculate

$$\{\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} KL((Ax)_i, b_i^k)\} - \{\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} KL((Ax)_i, b_i^{k+1})\}$$

and use the previous exercise.

Exercise 18.7 *Extend the boundedness result obtained earlier to conclude that for each fixed n the sequence $\{z^{k,n}\}$ is bounded.*

Since the sequence $\{z^{k,0}\}$ is bounded there is a subsequence $\{z^{k_t,0}\}$ converging to a limit vector $z^{*,0}$. Since the sequence $\{z^{k_t,1}\}$ is bounded there is subsequence converging to some vector $z^{*,1}$. Proceeding in this

way we find subsequences $\{z^{k_m,n}\}$ converging to $z^{*,n}$ for each fixed n. Our goal is to show that, with certain restrictions on A, $z^{*,n} = z^*$ for each n. We then show that the sequence $\{b^k\}$ converges to Az^* and that z^* minimizes

$$\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} KL((Ax)_i, b_i).$$

It follows from Exercise (18.5) that

$$\{\sum_{n=1}^{N} G_n(z^{*,n}, z^{*,n-1})\} = 0.$$

Exercise 18.8 *Find suitable restrictions on the matrix A that permit us to conclude from above that $z^{*,n} = z^{*,n-1} = z^*$ for each n.*

Exercise 18.9 *Show that the sequence $\{b^k\}$ converges to Az^*. Hints: Since the sequence $\{\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} KL((Az^*)_i, b_i^k)\}$ is decreasing and a subsequence converges to zero, it follows that the whole sequence converges to zero.*

Exercise 18.10 *Use Exercise (18.4) to obtain conditions that permit us to conclude that the vector z^* is a nonnegative minimizer of the function*

$$\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} KL((Ax)_i, b_i).$$

18.2.2 The RBI-EMML

We turn now to the RBI-EMML method, having the iterative step

$$x_j^{k+1} = (1 - m_n^{-1} s_j^{-1} s_{nj}) x_j^k + m_n^{-1} s_j^{-1} x_j^k \sum_{i \in B_n} A_{ij} b_i / (Ax^k)_i,$$

with $n = k(\mathrm{mod}\, N) + 1$. As we warned earlier, developing the theory for feedback with respect to the RBI-EMML algorithm appears to be more difficult than in the RBI-SMART case.

Applying the RBI-EMML algorithm to the system of equations $Ax = b$ having no nonnegative solution, we obtain the LC $\{z^1, ..., z^N\}$. As before, for each i we let $c_i = (Az^{n-1})_i$ where $i \in B_n$. There is a subset S of $\{j = 1, ..., J\}$ with cardinality less than I such that for all n we have $z_j^n = 0$ if j is not in S.

The first question that we ask is: Are the nonnegative minimizers of the functions

$$\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} KL(b_i, (Ax)_i)$$

and

$$\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} KL(c_i, (Ax)_i)$$

the same?

As before, the feedback approach involves setting $b^0 = b$, $c^0 = c = b^1$ and for each k defining $b^{k+1} = c^k$, where c^k is extracted from the limit cycle

$$LC(k) = \{z^{k,1}, ..., z^{k,N} = z^{k,0}\}$$

obtained from the system $b^k = Ax$ as $c_i^k = (Az^{k,n-1})_i$ where n is such that $i \in B_n$. Again, we must confront the issue of how we use the support sets. At the first step of feedback we apply RBI-EMML to the system $b = b^0 = Ax$, beginning with a positive vector x^0. The resulting limit cycle vectors are supported on a set S^0 with cardinality less than I. At the next step we apply the RBI-EMML to the system $b^1 = Ax$. Should we begin with a positive vector (not necessarily the same x^0 as before) or should our starting vector be supported on S^0? One approach could be to assume first that $J < I$ and that $S = \{j = 1, ..., J\}$ always and then see what can be discovered.

Our conjectures, subject to restrictions involving the support sets, are as follows:
1: The sequence $\{b^k\}$ converges to a limit vector b^∞;
2: The system $b^\infty = Ax$ has a nonnegative solution, say x^∞;
3: The LC obtained for each k converge to the singleton x^∞;
4: The vector x^∞ minimizes the function

$$\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} KL(b_i, (Ax)_i)$$

over nonnegative x.

Some results concerning feedback for RBI-EMML were presented in [24]. We sketch those results now.

Exercise 18.11 *Show that the quantity*

$$\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} b_i^k$$

is the same for $k = 0, 1, ...$ *Hints: Show that*

$$\sum_{j=1}^{J} s_j \sum_{n=1}^{N} (z_j^{k,n} - z_j^{k,n-1}) = 0$$

and rewrite it in terms of b^k *and* b^{k+1}.

Exercise 18.12 *Show that there is a constant $B > 0$ such that $z_j^{k,n} \leq B$ for all k, n and j.*

Exercise 18.13 *Show that*

$$s_j \log(z_j^{k,n-1}/z_j^{k,n}) \leq m_n^{-1} \sum_{i \in B_n} A_{ij} \log(b_i^{k+1}/b_i^k).$$

Hints: Use the convexity of the log function and the fact that the terms $1 - m_n^{-1} s_{nj}$ and $m_n^{-1} A_{ij}$, $i \in B_n$ sum to one.

Exercise 18.14 *Use the previous exercise to prove that the sequence*

$$\{\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} KL((Ax)_i, b_i^k)\}$$

is decreasing for each nonnegative vector x and the sequence

$$\{\sum_{n=1}^{N} m_n^{-1} \sum_{i \in B_n} A_{ij} \log(b_i^k)\}$$

is increasing.

Part VI

Optimization

Chapter 19

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.

19.1 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.

19.2 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.

19.2.1 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.

19.2.2 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?

19.2.3 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.

19.2.4 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.

19.3 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.

19.4 The Newton-Raphson Approach

The Newton-Raphson approach to minimizing a real-valued function $f : R^J \to R$ involves finding x^* such that $\nabla f(x^*) = 0$.

19.4.1 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.

19.4.2 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.

19.5 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 [34]). Other modifications of the gradient can also be used, as, for example, in the EMML algorithm.

Chapter 20

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.

20.1 Optimizing Functions of a Single Real Variable

Let $f : R \to R$ 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

$$|f(b) - f(a)| \leq L|b - a|, \tag{20.1}$$

for all a and b; functions that satisfy Equation (20.1) are said to be *L-Lipschitz*.

Suppose $g : R \to R$ 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.

20.1.1 The Convex Case

The function $g(x)$ is said to be *convex* if, for each pair of distinct real numbers a and b and for every α in the interval $(0, 1)$, we have

$$g((1-\alpha)a + \alpha b) \leq (1-\alpha)g(a) + \alpha g(b).$$

If $g(x)$ is a differentiable function, then convexity can be expressed in terms of properties of the derivative, $g'(x)$.

Theorem 20.1 *For the differentiable function $g(x)$, the following are equivalent:*
1) $g(x)$ is convex;
2) for all a and b we have

$$g(b) \geq g(a) + g'(a)(b-a); \tag{20.2}$$

3) the derivative, $g'(x)$, is an increasing function, or, equivalently,

$$(g'(b) - g'(a))(b-a) \geq 0, \tag{20.3}$$

for all a and b.

Proof of the Theorem: Assume that $g(x)$ is convex. Then, for any a and b and α in $(0, 1)$, we have

$$g(a + \alpha(b-a)) = g((1-\alpha)a + \alpha b) \leq (1-\alpha)g(a) + \alpha g(b).$$

Then,

$$[g(a + \alpha(b-a)) - g(a)]/[\alpha(b-a)] \leq [g(b) - g(a)]/[b-a].$$

The limit on the left, as $\alpha \to 0$, is $g'(a)$. It follows that

$$g'(a) \leq [g(b) - g(a)]/[b-a],$$

which is Inequality (20.2).

Assume now that Inequality (20.2) holds, for all a and b. Therefore, we also have

$$g(a) - g(b) \geq g'(b)(a-b),$$

or

$$g(a) - g(b) \geq -g'(b)(b-a). \tag{20.4}$$

Adding Inequalities (20.3) and (20.4), we obtain

$$0 \geq (g'(a) - g'(b))(b-a),$$

from which we easily conclude that $g'(x)$ is increasing.

Finally, assume that $g'(x)$ is an increasing function, so that Inequality (20.3) holds. We show that $g(x)$ is convex. Let $a < b$ and let $f(\alpha)$ be defined by

$$f(\alpha) = [(1-\alpha)g(a) + \alpha g(b)] - g((1-\alpha)a + \alpha b).$$

Then $f(0) = f(1) = 0$, and

$$f'(\alpha) = g(b) - g(a) - g'((1-\alpha)a + \alpha b)(b-a). \tag{20.5}$$

If $f(\alpha) < 0$ for some α, then there must be a minimum at $\alpha = \hat{\alpha}$ with $f'(\hat{\alpha}) = 0$. But, if $f(\alpha)$ had a relative minimum, then $f'(\alpha)$ would be increasing nearby. We conclude by showing that the function

$$g'((1-\alpha)a + \alpha b)(b-a)$$

is an increasing function of α. To see this, note that, for $\beta > \alpha$,

$$(\beta - \alpha)[g'((1-\beta)a + \beta b) - g'((1-\alpha)a + \alpha b)(b-a)$$

$$= [g'((1-\beta)a + \beta b) - g'((1-\alpha)a + \alpha b)][((1-\beta)a + \beta b) - ((1-\alpha)a + \alpha b)],$$

which is non-negative, according to Inequality (20.3). It follows that $f'(\alpha)$ is a decreasing function of α, so cannot have a relative minimum. This concludes the proof. ∎

Theorem 20.2 *If $g(x)$ is twice differentiable and $g''(x) \geq 0$ for all x, then $g(x)$ is convex.*

Proof: We have $g''(x) \geq 0$ for all x, so that

$$f''(\alpha) = -g''((1-\alpha)a + \alpha b)(b-a)^2 \leq 0,$$

where $f(\alpha)$ is as in the proof of the previous theorem. Therefore $f(\alpha)$ cannot have a relative minimum. This completes the proof. ∎

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 20.3 *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.

20.2 Optimizing Functions of Several Real Variables

Let $f : R^J \to R$ be a real-valued function of J real variables. The function $f(x)$ is said to be *differentiable* at the point x^0 if the partial derivatives, $\frac{\partial f}{\partial x_j}(x^0)$, exist for $j = 1, ..., J$ and

$$\lim_{h \to 0} \frac{1}{||h||_2}[f(x^0 + h) - f(x^0) - \langle \nabla f(x^0), h \rangle] = 0.$$

It can be shown that, if f is differentiable at $x = x^0$, then f is continuous there as well [70].

Let $f : R^J \to R$ be a differentiable function. From the Mean-Value Theorem ([70], 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

$$|f(b) - f(a)| \leq L||b - a||_2, \tag{20.6}$$

for all a and b; functions that satisfy Equation (20.6) 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.

20.2.1 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

$$g((1 - \alpha)a + \alpha b) \leq (1 - \alpha)g(a) + \alpha g(b).$$

If $g(x)$ is a differentiable function, then convexity can be expressed in terms of properties of the derivative, $\nabla g(x)$.

Theorem 20.4 *For the differentiable function $g(x)$, the following are equivalent:*
1) $g(x)$ is convex;

2) for all a and b we have

$$g(b) \geq g(a) + \langle \nabla g(a), b - a \rangle ; \tag{20.7}$$

3) for all a and b we have

$$\langle \nabla g(b) - \nabla g(a), b - a \rangle \geq 0. \tag{20.8}$$

Proof: Assume that $g(x)$ is convex. Then, for any a and b and α in $(0, 1)$, we have

$$g(a + \alpha(b - a)) = g((1 - \alpha)a + \alpha b) \leq (1 - \alpha)g(a) + \alpha g(b).$$

Then,

$$g(a + \alpha(b - a)) - g(a) \leq g(b) - g(a).$$

The limit on the left, as $\alpha \to 0$, is

$$\langle \nabla g(a), b - a \rangle.$$

It follows that

$$\langle \nabla g(a), b - a \rangle \leq g(b) - g(a).$$

which is Inequality (20.7).

Assume now that Inequality (20.7) holds, for all a and b. Therefore, we also have

$$g(a) - g(b) \geq \langle \nabla g(b), a - b \rangle,$$

or

$$g(a) - g(b) \geq -\langle \nabla g(b), b - a \rangle. \tag{20.9}$$

Adding Inequalities (20.7) and (20.9), we obtain Inequality (20.8).

Finally, assume that Inequality (20.8) holds. We show that $g(x)$ is convex. Let $a < b$ and let $f(\alpha)$ be defined by

$$f(\alpha) = [(1 - \alpha)g(a) + \alpha g(b)] - g((1 - \alpha)a + \alpha b).$$

Then $f(0) = f(1) = 0$, and

$$f'(\alpha) = g(b) - g(a) - \langle \nabla g((1 - \alpha)a + \alpha b), b - a \rangle. \tag{20.10}$$

If $f(\alpha) < 0$ for some α, then there must be a minimum at $\alpha = \hat{\alpha}$ with $f'(\hat{\alpha}) = 0$. But, if $f(\alpha)$ had a relative minimum, then $f'(\alpha)$ would be increasing nearby. We conclude by showing that the function

$$\langle \nabla g((1 - \alpha)a + \alpha b), b - a \rangle$$

is an increasing function of α. To see this, note that, for $\beta > \alpha$,

$$(\beta - \alpha)[\langle \nabla g((1-\beta)a + \beta b) - \nabla g((1-\alpha)a + \alpha b), b - a\rangle]$$

$$= \langle \nabla g((1-\beta)a+\beta b) - \nabla g((1-\alpha)a+\alpha b), ((1-\beta)a+\beta b) - ((1-\alpha)a+\alpha b)\rangle,$$

which is non-negative, according to Inequality (20.3). It follows that $f'(\alpha)$ is a decreasing function of α, so cannot have a relative minimum. This concludes the proof. ∎

As in the case of functions of a single variable, we can say more when the function $g(x)$ is twice differentiable.

Theorem 20.5 *If $g(x)$ is twice differentiable and the second derivative matrix is non-negative definite, that is, $\nabla^2 g(x) \geq 0$ for all x, then $g(x)$ is convex.*

Proof: Now we have

$$f''(\alpha) = -(b-a)^T \nabla^2 g((1-\alpha)a + \alpha b)(b-a) \leq 0,$$

where $f(\alpha)$ is as in the proof of the previous theorem. Therefore $f(\alpha)$ cannot have a relative minimum. This completes the proof. ∎

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 20.3.

Theorem 20.6 *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 20.3, the proof of this theorem is not trivial. In [73] Golshtein and Tretyakov prove the following theorem, from which Theorem 20.6 follows immediately.

Theorem 20.7 *Let $g : R^J \to R$ be convex and differentiable. The following are equivalent:*

$$||\nabla g(x) - \nabla g(y)||_2 \leq ||x - y||_2; \tag{20.11}$$

$$g(x) \geq g(y) + \langle \nabla g(y), x - y \rangle + \frac{1}{2}||\nabla g(x) - \nabla g(y)||_2^2; \tag{20.12}$$

and

$$\langle \nabla g(x) - \nabla g(y), x - y \rangle \geq ||\nabla g(x) - \nabla g(y)||_2^2. \tag{20.13}$$

Proof: The only difficult step in the proof is showing that Inequality (20.11) implies Inequality (20.12). 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 (20.11). 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

$$g(y) \geq \inf g(x) + \frac{1}{2}||\nabla g(y)||_2^2. \tag{20.14}$$

Now fix y and define the function $f(x)$ 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 (20.14) 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.

20.3 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 [76] is the following. Begin with an arbitrary x^0. For $k = 0, 1, ...$, and $i = k(\mathrm{mod}\, I) + 1$, 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.

20.3.1 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 20.1 *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 20.2 *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 20.3 *Show that, for $x^{k+1} = P_i x^k$, where $i = k(\mathrm{mod}\, I) + 1$,*

$$||\hat{c} - x^k||_2^2 - ||\hat{c} - x^{k+1}||_2^2 = ||x^k - x^{k+1}||_2^2. \tag{20.15}$$

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 (20.15), 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.

20.3.2 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.

20.3.3 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 $P_1 c^{*,I} = c^{*,1}$. The set $\{c^{*,i}\}$ is then a limit cycle. For the special case of $I = 2$ we can prove this.

Theorem 20.8 *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 (**??**), 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. ∎

20.4 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 20.4 *Show that $\hat{x}$ is a fixed point of S if and only if $\hat{x}$ minimizes $g(x)$ over x in C.*

20.4.1 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.

20.5 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 $\text{Ext}(C)$.

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.

20.6 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 20.1 *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}.

20.7 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 [65]. For simplicity, we shall discuss only the case of $C = A \cap B$, the intersection of two closed, convex sets.

20.7.1 A Motivating Lemma

The following lemma will help to motivate Dykstra's algorithm.

Lemma 20.1 *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, $c = P_C x$. ∎

20.7.2 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 [16], 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 [43], Censor and Reich extend Dykstra's algorithm to Bregman projections, and, in [17], the three show that the extended Dykstra algorithm of [43] 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.

20.7.3 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 [6] 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 [62] to minimize certain (possibly non-quadratic) functions over the intersection of fixed point sets of operators more general than P_{C_i}.

Chapter 21

Generalized Projections onto Convex Sets

The *convex feasibility problem* (CFP) is to find a member of the nonempty set $C = \bigcap_{i=1}^{I} C_i$, where the C_i are closed convex subsets of R^J. In most applications the sets C_i are more easily described than the set C and algorithms are sought whereby a member of C is obtained as the limit of an iterative procedure involving (exact or approximate) orthogonal or generalized projections onto the individual sets C_i.

In his often cited paper [16] Bregman generalizes the SOP algorithm for the convex feasibility problem to include projections with respect to a generalized distance, and uses this *successive generalized projections* (SGP) method to obtain a *primal-dual algorithm* to minimize a convex function $f : R^J \to R$ over the intersection of half-spaces, that is, over x with $Ax \geq b$. The generalized distance is built from the function f, which then must exhibit additional properties, beyond convexity, to guarantee convergence of the algorithm

21.1 Bregman Functions and Bregman Distances

The class of functions f that are used to define the generalized distance have come to be called *Bregman functions*; the associated generalized distances are then *Bregman distances*, which are used to define generalized projections onto closed convex sets (see the book by Censor and Zenios [45] for details). In [9] Bauschke and Borwein introduce the related class of *Bregman-Legendre functions* and show that these functions provide an appropriate setting in which to study Bregman distances and generalized

projections associated with such distances. For further details concerning Bregman and Bregman-Legendre functions, see the appendix.

Bregman's *successive generalized projection* (SGP) method uses projections with respect to Bregman distances to solve the convex feasibility problem. Let $f : R^J \to (-\infty, +\infty]$ be a closed, proper convex function, with essential domain $D = \mathrm{dom} f = \{x | f(x) < +\infty\}$ and $\emptyset \neq \mathrm{int}\, D$. Denote by $D_f(\cdot, \cdot) : D \times \mathrm{int}\, D \to [0, +\infty)$ the Bregman distance, given by

$$D_f(x, z) = f(x) - f(z) - \langle \nabla f(z), x - z \rangle \tag{21.1}$$

and by $P^f_{C_i}$ the Bregman projection operator associated with the convex function f and the convex set C_i; that is

$$P^f_{C_i} z = \arg\min_{x \in C_i \cap D} D_f(x, z). \tag{21.2}$$

The Bregman projection of x onto C is characterized by *Bregman's Inequality*:

$$\langle \nabla f(P^f_C x) - \nabla f(x), c - P^f_C \rangle \geq 0, \tag{21.3}$$

for all c in C.

21.2 The Successive Generalized Projections Algorithm

Bregman considers the following generalization of the SOP algorithm:

Algorithm 21.1 Bregman's method of Successive Generalized Projections (SGP): *Beginning with* $x^0 \in \mathrm{int}\, \mathrm{dom} f$, *for* $k = 0, 1, ...,$ *let* $i = i(k) := k(\mathrm{mod}\, \mathrm{I}) + 1$ *and*

$$x^{k+1} = P^f_{C_{i(k)}}(x^k). \tag{21.4}$$

He proves that the sequence $\{x^k\}$ given by (21.4) converges to a member of $C \cap \mathrm{dom} f$, whenever this set is nonempty and the function f is what came to be called a Bregman function ([16]). Bauschke and Borwein [9] prove that Bregman's SGP method converges to a member of C provided that one of the following holds: 1) f is Bregman-Legendre; 2) $C \cap \mathrm{int} D \neq \emptyset$ and $\mathrm{dom}\, f^*$ is open; or 3) dom f and dom f^* are both open, with f^* the function conjugate to f.

In [16] Bregman goes on to use the SGP to find a minimizer of a Bregman function $f(x)$ over the set of x such that $Ax = b$. Each hyperplane associated with a single equation is a closed, convex set. The SGP finds the Bregman projection of the starting vector onto the intersection of the hyperplanes. If the starting vector has the form $x^0 = A^T d$, for some vector d, then this Bregman projection also minimizes $f(x)$ over x in the intersection.

21.3 Bregman's Primal-Dual Algorithm

The problem is to minimize $f : R^J \to R$ over the set of all x for which $Ax \geq b$. Begin with x^0 such that $x^0 = A^T u^0$, for some $u^0 \geq 0$. For $k = 0, 1, ...$, let $i = k(\mathrm{mod}\, I) + 1$. Having calculated x^k, there are three possibilities:

a) if $(Ax^k)_i < b_i$, then let x^{k+1} be the Bregman projection onto the hyperplane $H_i = \{x|(Ax)_i = b_i\}$, so that

$$\nabla f(x^{k+1}) = \nabla f(x^k) + \lambda_k a^i,$$

where a^i is the ith column of A^T. With $\nabla f(x^k) = A^T u^k$, for $u^k \geq 0$, update u^k by

$$u_i^{k+1} = u_i^k + \lambda_k,$$

and

$$u_m^{k+1} = u_m^k,$$

for $m \neq i$.

b) if $(Ax^k)_i = b_i$, or $(Ax^k)_i > b_i$ and $u_i^k = 0$, then $x^{k+1} = x^k$, and $u^{k+1} = u^k$.

c) if $(Ax^k)_i > b_i$ and $u_i^k > 0$, then let μ_k be the smaller of the numbers μ_k' and μ_k'', where

$$\nabla f(y) = \nabla f(x^k) - \mu_k' a^i$$

puts y in H_i, and

$$\mu_k'' = u_i^k.$$

Then take x^{k+1} with

$$\nabla f(x^{k+1}) = \nabla f(x^k) - \mu_k a^i.$$

With appropriate assumptions made about the function f, the sequence $\{x^k\}$ so defined converges to a minimizer of $f(x)$ over the set of x with $Ax \geq b$. For a detailed proof of this result, see [45].

Bregman also suggests that this primal-dual algorithm be used to find approximate solutions for linear programming problems, where the problem is to minimize a linear function $c^T x$, subject to constraints. His idea is to replace the function $c^T x$ with $h(x) = c^T x + \epsilon f(x)$, and then apply his primal-dual method to $h(x)$.

21.4 Dykstra's Algorithm for Bregman Projections

We are concerned now with finding the Bregman projection of x onto the intersection C of finitely many closed convex sets, C_i. The problem can be solved by extending Dykstra's algorithm to include Bregman projections.

21.4.1 A Helpful Lemma

The following lemma helps to motivate the extension of Dykstra's algorithm.

Lemma 21.1 *Suppose that*

$$\nabla f(c) - \nabla f(x) = \nabla f(c) - \nabla f(c+p) + \nabla f(c) - \nabla f(c+q),$$

with $c = P_A^f(c+p)$ *and* $c = P_B^f(c+q)$. *Then* $c = P_C^f x$.

Proof: Let d be arbitrary in C. We have

$$\langle \nabla f(c) - \nabla f(c+p), d-c \rangle \geq 0,$$

and

$$\langle \nabla f(c) - \nabla f(c+q), d-c \rangle \geq 0.$$

Adding, we obtain

$$\langle \nabla f(c) - \nabla f(x), d-c \rangle \geq 0.$$

∎

This suggests the following algorithm for finding $c = P_C^f x$, which turns out to be the extension of Dykstra's algorithm to Bregman projections.

Begin with $b^0 = x$, $p_0 = q_0 = 0$. Define

$$b_{n-1} + p_{n-1} = \nabla f^{-1}(\nabla f(b_{n-1}) + r_{n-1}),$$

$$a_n = P_A^f(b_{n-1} + p_{n-1}),$$

$$r_n = \nabla f(b_{n-1}) + r_{n-1} - \nabla f(a_n),$$

$$\nabla f(a_n + q_{n-1}) = \nabla f(a_n) + s_{n-1},$$

$$b_n = P_B^f(a_n + q_{n-1}),$$

and

$$s_n = \nabla f(a_n) + s_{n-1} - \nabla f(b_n).$$

In place of

$$\nabla f(c+p) - \nabla f(c) + \nabla f(c+q) - \nabla f(c),$$

we have

$$[\nabla f(b_{n-1}) + r_{n-1}] - \nabla f(b_{n-1}) + [\nabla f(a_n) + s_{n-1}] - \nabla f(a_n) = r_{n-1} + s_{n-1},$$

and also

$$[\nabla f(a_n) + s_{n-1}] - \nabla f(a_n) + [\nabla f(b_n) + r_n] - \nabla f(b_n) = r_n + s_{n-1}.$$

But we also have

$$r_{n-1} + s_{n-1} = \nabla f(x) - \nabla f(b_{n-1}),$$

and

$$r_n + s_{n-1} = \nabla f(x) - \nabla f(a_n).$$

Then the sequences $\{a_n\}$ and $\{b_n\}$ converge to c. For further details, see [43] and [11].

In [17] the authors show that the extension of Dykstra's algorithm to Bregman projections can be viewed as an extension of Bregman's primal-dual algorithm to the case in which the intersection of half-spaces is replaced by the intersection of closed convex sets.

Chapter 22

An Interior-Point Optimization Method

Investigations in [23] into several well known iterative algorithms, including the 'expectation maximization maximum likelihood' (EMML) method, the 'multiplicative algebraic reconstruction technique' (MART) as well as block-iterative and simultaneous versions of MART, revealed that the iterative step of each algorithm involved weighted arithmetic or geometric means of Bregman projections onto hyperplanes; interestingly, the projections involved were associated with Bregman distances that differed from one hyperplane to the next. This representation of the EMML algorithm as a weighted arithmetic mean of Bregman projections provided the key step in obtaining block-iterative and row-action versions of EMML. Because it is well known that convergence is not guaranteed if one simply extends Bregman's algorithm to multiple distances by replacing the single distance D_f in (21.4) with multiple distances D_{f_i}, the appearance of distinct distances in these algorithms suggested that a somewhat more sophisticated algorithm employing multiple Bregman distances might be possible.

22.1 The Multiprojection Successive Generalized Projection Method

In [27] such an iterative multiprojection method for solving the CFP, called the *multidistance successive generalized projection* method (MSGP), was presented in the context of Bregman functions, and subsequently, in the framework of Bregman-Legendre functions [29]; see the Appendix on Bregman functions for definitions and details concerning these functions. The MSGP extends Bregman's SGP method by allowing the Breg-

man projection onto each set C_i to be performed with respect to a Bregman distance D_{f_i} derived from a Bregman-Legendre function f_i. The MSGP method depends on the selection of a super-coercive Bregman-Legendre function h whose Bregman distance D_h satisfies the inequality $D_h(x,z) \geq D_{f_i}(x,z)$ for all $x \in \operatorname{dom} h \subseteq \bigcap_{i=1}^{I} \operatorname{dom} f_i$ and all $z \in \operatorname{int} \operatorname{dom} h$, where $\operatorname{dom} h = \{x | h(x) < +\infty\}$. By using different Bregman distances for different convex sets, we found that we can sometimes calculate the desired Bregman projections in closed form, thereby obtaining computationally tractable iterative algorithms (see [23]).

22.2 An Interior-Point Algorithm (IPA)

Consideration of a special case of the MSGP, involving only a single convex set C_1, leads us to an interior point optimization method. If $I = 1$ and $f := f_1$ has a unique minimizer $\hat{x}$ in $\operatorname{int} \operatorname{dom} h$, then the MSGP iteration using $C_1 = \{\hat{x}\}$ is

$$\nabla h(x^{k+1}) = \nabla h(x^k) - \nabla f(x^k). \tag{22.1}$$

This suggests an *interior-point algorithm* (IPA) that could be applied more broadly to minimize a convex function f over the closure of $\operatorname{dom} h$.

First, we present the MSGP method and prove convergence, in the context of Bregman-Legendre functions. Then we investigate the IPA suggested by the MSGP algorithm.

22.3 The MSGP Algorithm

We begin by setting out the assumptions we shall make and the notation we shall use in this section.

22.3.1 Assumptions and Notation

We make the following assumptions throughout this section. Let $C = \cap_{i=1}^{I} C_i$ be the nonempty intersection of closed convex sets C_i. The function h is super-coercive and Bregman-Legendre with essential domain $D = \operatorname{dom} h$ and $C \cap \operatorname{dom} h \neq \emptyset$. For $i = 1, 2, ..., I$ the function f_i is also Bregman-Legendre, with $D \subseteq \operatorname{dom} f_i$, so that $\operatorname{int} D \subseteq \operatorname{int} \operatorname{dom} f_i$; also $C_i \cap \operatorname{int} \operatorname{dom} f_i \neq \emptyset$. For all $x \in \operatorname{dom} h$ and $z \in \operatorname{int} \operatorname{dom} h$ we have $D_h(x,z) \geq D_{f_i}(x,z)$, for each i.

22.3.2 The MSGP Algorithm

Algorithm 22.1 The MSGP algorithm: *Let $x^0 \in \operatorname{int}\operatorname{dom} h$ be arbitrary. For $k = 0, 1, ...$ and $i(k) := k(\operatorname{mod} I) + 1$ let*

$$x^{k+1} = \nabla h^{-1}\Big(\nabla h(x^k) - \nabla f_{i(k)}(x^k) + \nabla f_{i(k)}(P_{C_{i(k)}}^{f_{i(k)}}(x^k))\Big). \tag{22.2}$$

22.3.3 A Preliminary Result

For each $k = 0, 1, ...$ define the function $G^k(\cdot) : \operatorname{dom} h \to [0, +\infty)$ by

$$G^k(x) = D_h(x, x^k) - D_{f_{i(k)}}(x, x^k) + D_{f_{i(k)}}(x, P_{C_{i(k)}}^{f_{i(k)}}(x^k)). \tag{22.3}$$

The next proposition provides a useful identity, which can be viewed as an analogue of Pythagoras' theorem. The proof is not difficult and we omit it.

Proposition 22.1 *For each $x \in \operatorname{dom} h$, each $k = 0, 1, ...$, and x^{k+1} given by (22.2) we have*

$$G^k(x) = G^k(x^{k+1}) + D_h(x, x^{k+1}). \tag{22.4}$$

Consequently, x^{k+1} is the unique minimizer of the function $G^k(\cdot)$.

This identity (22.4) is the key ingredient in the convergence proof for the MSGP algorithm.

22.3.4 The MSGP Convergence Theorem

We shall prove the following convergence theorem:

Theorem 22.1 *Let $x^0 \in \operatorname{int}\operatorname{dom} h$ be arbitrary. Any sequence x^k obtained from the iterative scheme given by Algorithm 22.1 converges to $x^\infty \in C \cap \operatorname{dom} h$. If the sets C_i are hyperplanes, then x^∞ minimizes the function $D_h(x, x^0)$ over all $x \in C \cap \operatorname{dom} h$; if, in addition, x^0 is the global minimizer of h, then x^∞ minimizes $h(x)$ over all $x \in C \cap \operatorname{dom} h$.*

Proof: All details concerning Bregman functions are in the Appendix. Let c be a member of $C \cap \operatorname{dom} h$. From the Pythagorean identity (22.4) it follows that

$$G^k(c) = G^k(x^{k+1}) + D_h(c, x^{k+1}). \tag{22.5}$$

Using the definition of $G^k(\cdot)$, we write

$$G^k(c) = D_h(c, x^k) - D_{f_{i(k)}}(c, x^k) + D_{f_{i(k)}}(c, P_{C_{i(k)}}^{f_{i(k)}}(x^k)). \tag{22.6}$$

From Bregman's Inequality (21.3) we have that

$$D_{f_{i(k)}}(c, x^k) - D_{f_{i(k)}}(c, P_{C_{i(k)}}^{f_{i(k)}}(x^k)) \geq D_{f_{i(k)}}(P_{C_{i(k)}}^{f_{i(k)}}(x^k), x^k). \quad (22.7)$$

Consequently, we know that

$$D_h(c, x^k) - D_h(c, x^{k+1}) \geq G^k(x^{k+1}) + D_{f_{i(k)}}(P_{C_{i(k)}}^{f_{i(k)}}(x^k), x^k) \geq 0. \quad (22.8)$$

It follows that $\{D_h(c, x^k)\}$ is decreasing and finite and the sequence $\{x^k\}$ is bounded. Therefore, $\{D_{f_{i(k)}}(P_{C_{i(k)}}^{f_{i(k)}}(x^k), x^k)\} \to 0$ and $\{G^k(x^{k+1})\} \to 0$; from the definition of $G^k(x)$ it follows that $\{D_{f_{i(k)}}(x^{k+1}, P_{C_{i(k)}}^{f_{i(k)}}(x^k))\} \to 0$ as well. Using the Bregman inequality we obtain the inequality

$$D_h(c, x^k) \geq D_{f_{i(k)}}(c, x^k) \geq D_{f_{i(k)}}(c, P_{C_{i(k)}}^{f_{i(k)}}(x^k)), \quad (22.9)$$

which tells us that the sequence $\{P_{C_{i(k)}}^{f_{i(k)}}(x^k)\}$ is also bounded. Let x^* be an arbitrary cluster point of the sequence $\{x^k\}$ and let $\{x^{k_n}\}$ be a subsequence of the sequence $\{x^k\}$ converging to x^*.

We first show that $x^* \in \operatorname{dom} h$ and $\{D_h(x^*, x^k)\} \to 0$. If x^* is in $\operatorname{int}\operatorname{dom} h$ then our claim is verified, so suppose that x^* is in $\operatorname{bdry}\operatorname{dom} h$. If c is in $\operatorname{dom} h$ but not in $\operatorname{int}\operatorname{dom} h$, then, applying B2 of the Appendix on Bregman functions, we conclude that $x^* \in \operatorname{dom} h$ and $\{D_h(x^*, x^k)\} \to 0$. If, on the other hand, c is in $\operatorname{int}\operatorname{dom} h$ then by R2 x^* would have to be in $\operatorname{int}\operatorname{dom} h$ also. It follows that $x^* \in \operatorname{dom} h$ and $\{D_h(x^*, x^k)\} \to 0$. Now we show that x^* is in C.

Label $x^* = x_0^*$. Since there must be at least one index i that occurs infinitely often as $i(k)$, we assume, without loss of generality, that the subsequence $\{x^{k_n}\}$ has been selected so that $i(k) = 1$ for all $n = 1, 2, ...$ Passing to subsequences as needed, we assume that, for each $m = 0, 1, 2, ..., I-1$, the subsequence $\{x^{k_n+m}\}$ converges to a cluster point x_m^*, which is in $\operatorname{dom} h$, according to the same argument we used in the previous paragraph. For each m the sequence $\{D_{f_m}(c, P_{C_m}^{f_m}(x^{k_n+m-1}))\}$ is bounded, so, again, by passing to subsequences as needed, we assume that the subsequence $\{P_{C_m}^{f_m}(x^{k_n+m-1})\}$ converges to $c_m^* \in C_m \cap \overline{\operatorname{dom} f_m}$.

Since the sequence $\{D_{f_m}(c, P_{C_m}^{f_m}(x^{k_n+m-1})\}$ is bounded and $c \in \operatorname{dom} f_m$, it follows, from either B2 or R2, that $c_m^* \in \operatorname{dom} f_m$. We know that

$$\{D_{f_m}(P_{C_m}^{f_m}(x^{k_n+m-1}), x^{k_n+m-1})\} \to 0 \quad (22.10)$$

and both $P_{C_m}^{f_m}(x^{k_n+m-1})$ and x^{k_n+m-1} are in $\operatorname{int}\operatorname{dom} f_m$. Applying R1, B3 or R3, depending on the assumed locations of c_m^* and x_{m-1}^*, we conclude that $c_m^* = x_{m-1}^*$.

We also know that

$$\{D_{f_m}(x^{k_n+m}, P_{C_m}^{f_m}(x^{k_n+m-1}))\} \to 0, \tag{22.11}$$

from which it follows, using the same arguments, that $x_m^* = c_m^*$. Therefore, we have $x^* = x_m^* = c_m^*$ for all m; so $x^* \in C$.

Since $x^* \in C \cap \operatorname{dom} h$, we may now use x^* in place of the generic c, to obtain that the sequence $\{D_h(x^*, x^k)\}$ is decreasing. However, we also know that the sequence $\{D_h(x^*, x^{k_n})\} \to 0$. So we have $\{D_h(x^*, x^k)\} \to 0$. Applying R5, we conclude that $\{x^k\} \to x^*$.

If the sets C_i are hyperplanes, then we get equality in Bregman's inequality (21.3)and so

$$D_h(c, x^k) - D_h(c, x^{k+1}) = G^k(x^{k+1}) + D_{f_{i(k)}}(P_{C_{i(k)}}^{f_{i(k)}}(x^k), x^k). \tag{22.12}$$

Since the right side of this equation is independent of which c we have chosen in the set $C \cap \operatorname{dom} h$, the left side is also independent of this choice. This implies that

$$D_h(c, x^0) - D_h(c, x^M) = D_h(x^*, x^0) - D_h(x^*, x^M), \tag{22.13}$$

for any positive integer M and any $c \in C \cap \operatorname{dom} h$. Therefore

$$D_h(c, x^0) - D_h(x^*, x^0) = D_h(c, x^M) - D_h(x^*, x^M). \tag{22.14}$$

Since $\{D_h(x^*, x^M)\} \to 0$ as $M \to +\infty$ and $\{D_h(c, x^M)\} \to \alpha \geq 0$, we have that $D_h(c, x^0) - D_h(x^*, x^0) \geq 0$. This completes the proof. ∎

22.4 An Interior-Point Algorithm for Iterative Optimization

We consider now an interior point algorithm (IPA) for iterative optimization. This algorithm was first presented in [28] and applied to transmission tomography in [105]. The IPA is suggested by a special case of the MSGP, involving functions h and $f := f_1$.

22.4.1 Assumptions

We assume, for the remainder of this section, that h is a super-coercive Legendre function with essential domain $D = \operatorname{dom} h$. We also assume that f is continuous on the set $\overline{D}$, takes the value $+\infty$ outside this set and is differentiable in $\operatorname{int} D$. Thus, f is a closed, proper convex function on R^J. We assume also that $\hat{x} = \operatorname{argmin}_{x \in \overline{D}} f(x)$ exists, but not that it is unique. As in the previous section, we assume that $D_h(x, z) \geq D_f(x, z)$ for all $x \in \operatorname{dom} h$ and $z \in \operatorname{int} \operatorname{dom} h$. As before, we denote by h^* the function conjugate to h.

22.4.2 The IPA

The IPA is an iterative procedure that, under conditions to be described shortly, minimizes the function f over the closure of the essential domain of h, provided that such a minimizer exists.

Algorithm 22.2 *Let* x^0 *be chosen arbitrarily in* $\operatorname{int} D$. *For* $k = 0, 1, ...$ *let* x^{k+1} *be the unique solution of the equation*

$$\nabla h(x^{k+1}) = \nabla h(x^k) - \nabla f(x^k). \tag{22.15}$$

Note that equation (22.15) can also be written as

$$x^{k+1} = \nabla h^{-1}(\nabla h(x^k) - \nabla f(x^k)) = \nabla h^*(\nabla h(x^k) - \nabla f(x^k)). \tag{22.16}$$

22.4.3 Motivating the IPA

As already noted, the IPA was originally suggested by consideration of a special case of the MSGP. Suppose that $\overline{x} \in \operatorname{dom} h$ is the unique global minimizer of the function f, and that $\nabla f(\overline{x}) = 0$. Take $I = 1$ and $C = C_1 = \{\overline{x}\}$. Then $P^f_{C_1}(x^k) = \overline{x}$ always and the iterative MSGP step becomes that of the IPA. Since we are assuming that $\overline{x}$ is in dom h, the convergence theorem for the MSGP tells us that the iterative sequence $\{x^k\}$ converges to $\overline{x}$.

In most cases, the global minimizer of f will not lie within the essential domain of the function h and we are interested in the minimum value of f on the set $\overline{D}$, where $D = \operatorname{dom} h$; that is, we want $\hat{x} = \operatorname{argmin}_{x \in \overline{D}} f(x)$, whenever such a minimum exists. As we shall see, the IPA can be used to advantage even when the specific conditions of the MSGP do not hold.

22.4.4 Preliminary results for the IPA

Two aspects of the IPA suggest strongly that it may converge under more general conditions than those required for convergence of the MSGP. The sequence $\{x^k\}$ defined by (22.15) is entirely within the interior of $\operatorname{dom} h$. In addition, as we now show, the sequence $\{f(x^k)\}$ is decreasing. Adding both sides of the inequalities $D_h(x^{k+1}, x^k) - D_f(x^{k+1}, x^k) \geq 0$ and $D_h(x^k, x^{k+1}) - D_f(x^k, x^{k+1}) \geq 0$ gives

$$\langle \nabla h(x^k) - \nabla h(x^{k+1}) - \nabla f(x^k) + \nabla f(x^{k+1}), x^k - x^{k+1} \rangle \geq 0. \tag{22.17}$$

Substituting according to equation (22.15) and using the convexity of the function f, we obtain

$$f(x^k) - f(x^{k+1}) \geq \langle \nabla f(x^{k+1}), x^k - x^{k+1} \rangle \geq 0. \tag{22.18}$$

Therefore, the sequence $\{f(x^k)\}$ is decreasing; since it is bounded below by $f(\hat{x})$, it has a limit, $\hat{f} \geq f(\hat{x})$. We have the following result (see [28], Prop. 3.1).

Lemma 22.1 $\hat{f} = f(\hat{x})$.

Proof: Suppose, to the contrary, that $0 < \delta = \hat{f} - f(\hat{x})$. Select $z \in D$ with $f(z) \leq f(\hat{x}) + \delta/2$. Then $f(x^k) - f(z) \geq \delta/2$ for all k. Writing $H_k = D_h(z, x^k) - D_f(z, x^k)$ for each k, we have

$$H_k - H_{k+1} = D_h(x^{k+1}, x^k) - D_f(x^{k+1}, x^k) + \langle \nabla f(x^{k+1}), x^{k+1} - z \rangle \quad (22.19)$$

Since $\langle \nabla f(x^{k+1}), x^{k+1} - z \rangle \geq f(x^{k+1}) - f(z) \geq \delta/2 > 0$ and $D_h(x^{k+1}, x^k) - D_f(x^{k+1}, x^k) \geq 0$, it follows that $\{H_k\}$ is a decreasing sequence of positive numbers, so that the successive differences converge to zero. This is a contradiction; we conclude that $\hat{f} = f(\hat{x})$. ∎

Convergence of the IPA

We prove the following convergence result for the IPA (see also [28]).

Theorem 22.2 *If $\hat{x} = \text{argmin}_{x \in \overline{D}} f(x)$ is unique, then the sequence $\{x^k\}$ generated by the IPA according to equation (22.15) converges to $\hat{x}$. If $\hat{x}$ is not unique, but can be chosen in D, then the sequence $\{D_h(\hat{x}, x^k)\}$ is decreasing. If, in addition, the function $D_h(\hat{x}, \cdot)$ has bounded level sets, then the sequence $\{x^k\}$ is bounded and so has cluster points $x^* \in \overline{D}$ with $f(x^*) = f(\hat{x})$. Finally, if h is a Bregman-Legendre function, then $x^* \in D$ and the sequence $\{x^k\}$ converges to x^*.*

Proof: According to Corollary 8.7.1 of [111], if G is a closed, proper convex function on R^J and if the level set $L_\alpha = \{x | G(x) \leq \alpha\}$ is nonempty and bounded for at least one value of α, then L_α is bounded for all values of α. If the constrained minimizer $\hat{x}$ is unique, then, by the continuity of f on $\overline{D}$ and Rockafellar's corollary, we can conclude that the sequence $\{x^k\}$ converges to $\hat{x}$. If $\hat{x}$ is not unique, but can be chosen in D, then, with additional assumptions, convergence can still be established.

Suppose now that $\hat{x}$ is not necessarily unique, but can be chosen in D. Assuming $\hat{x} \in D$, we show that the sequence $\{D_h(\hat{x}, x^k)\}$ is decreasing. Using Equation (22.15) we have

$$\begin{aligned} D_h(\hat{x}, x^k) - D_h(\hat{x}, x^{k+1}) &= D_h(x^{k+1}, x^k) + \langle \nabla h(x^{k+1}) - \nabla h(x^k), \hat{x} - x^{k+1} \rangle \\ &= D_h(x^{k+1}, x^k) - D_f(x^{k+1}, x^k) + D_f(x^{k+1}, x^k) + \langle \nabla f(x^k), x^{k+1} - \hat{x} \rangle \\ &= D_h(x^{k+1}, x^k) - D_f(x^{k+1}, x^k) + f(x^{k+1}) - f(x^k) - \langle \nabla f(x^k), \hat{x} - x^k \rangle \\ &\geq D_h(x^{k+1}, x^k) - D_f(x^{k+1}, x^k) + f(x^{k+1}) - f(x^k) + f(x^k) - f(\hat{x}); \end{aligned}$$

the final inequality follows from the convexity of f. Since $D_h(x^{k+1}, x^k) - D_f(x^{k+1}, x^k) \geq 0$ and $f(x^{k+1}) - f(\hat{x}) \geq 0$, it follows that the sequence $\{D_h(\hat{x}, x^k)\}$ is decreasing.

If h has bounded level sets, then the sequence $\{x^k\}$ is bounded and we can extract a subsequence $\{x^{k_n}\}$ converging to some x^* in the closure of D.

Finally, assume that h is a Bregman-Legendre function. If $\hat{x}$ is in D but not in $\operatorname{int} D$, then, by B2, $x^* \in \operatorname{bdry} D$ implies that x^* is in D and $\{D_h(x^*, x^{k_n})\} \to 0$. If $\hat{x}$ is in $\operatorname{int} D$, then we conclude, from R2, that x^* is also in $\operatorname{int} D$. Then, by R1, we have $\{D_h(x^*, x^{k_n})\} \to 0$. We can then replace the generic $\hat{x}$ with x^*, to conclude that $\{D_h(x^*, x^k)\}$ is decreasing. But, $\{D_h(x^*, x^{k_n})\}$ converges to zero; therefore, the entire sequence $\{D_h(x^*, x^k)\}$ converges to zero. Applying R5, we conclude that $\{x^k\}$ converges to x^*. This completes the proof. ∎

Chapter 23

Linear and Convex Programming

The term *linear programming* (LP) refers to the problem of optimizing a linear function of several variables over linear equality or inequality constraints. In this chapter we present the problem and establish the basic facts. For a much more detailed discussion, consult [106].

23.1 Primal and Dual Problems

Associated with the basic problem in LP, called the *primary problem*, there is a second problem, the *dual problem*. Both of these problems can be written in two equivalent ways, the canonical form and the standard form.

23.1.1 Canonical and Standard Forms

Let b and c be fixed vectors and A a fixed matrix. The problem

$$\text{minimize } z = c^T x, \text{ subject to } Ax \geq b, x \geq 0 \text{ (PC)} \tag{23.1}$$

is the so-called *primary problem* of LP, in *canonical form*. The *dual problem* in canonical form is

$$\text{maximize } w = b^T y, \text{ subject to } A^T y \leq c, y \geq 0. \text{ (DC)} \tag{23.2}$$

The primary problem, in *standard form*, is

$$\text{minimize } z = c^T x, \text{ subject to } Ax = b, x \geq 0 \text{ (PS)} \tag{23.3}$$

with the dual problem in standard form given by

$$\text{maximize } w = b^T y, \text{ subject to } A^T y \leq c. \text{ (DS)} \tag{23.4}$$

Notice that the dual problem in standard form does not require that y be nonnegative. Note also that the standard problems make sense only if the system $Ax = b$ is underdetermined and has infinitely many solutions. For that reason, we shall assume, for the standard problems, that the I by J matrix A has more columns than rows, so $J > I$, and has full row rank.

If we are given the primary problem in canonical form, we can convert it to standard form by augmenting the variables, that is, by defining

$$u_i = (Ax)_i - b_i,$$

for $i = 1, ..., I$, and rewriting $Ax \geq b$ as

$$\tilde{A}\tilde{x} = b,$$

for $\tilde{A} = [\, A \quad -I \,]$ and $\tilde{x} = [x^T u^T]^T$.

23.1.2 Weak Duality

Consider the problems (PS) and (DS). Say that x is *feasible* if $x \geq 0$ and $Ax = b$. Let F be the set of feasible x. Say that y is *feasible* if $A^T y \leq c$. The *Weak Duality Theorem* is the following:

Theorem 23.1 *Let x and y be feasible vectors. Then*

$$z = c^T x \geq b^T y = w.$$

Corollary 23.1 *If z is not bounded below, then there are no feasible y.*

Corollary 23.2 *If x and y are both feasible, and $z = w$, then both x and y are optimal for their respective problems.*

Exercise 23.1 *Prove the theorem and its corollaries.*

The nonnegative quantity $c^T x - b^T y$ is called the *duality gap*. The *complementary slackness condition* says that, for optimal x and y, we have

$$x_j(c_j - (A^T y)_j) = 0,$$

for each j, which says that the duality gap is zero. Primal-dual algorithms for solving linear programming problems are based on finding sequences $\{x^k\}$ and $\{y^k\}$ that drive the duality gap down to zero [106].

23.1.3 Strong Duality

The *Strong Duality Theorem* makes a stronger statement.

Theorem 23.2 *If one of the problems (PS) or (DS) has an optimal solution, then so does the other and $z = w$ for the optimal vectors.*

Before we consider the proof of the theorem, we need a few preliminary results.

A point x in F is said to be a *basic feasible solution* if the columns of A corresponding to positive entries of x are linearly independent; denote by B an invertible matrix obtained by deleting from A columns associated with zero entries of x. The entries of an arbitrary x corresponding to the columns not deleted are called the *basic variables*. Then, assuming that the columns of B are the first I columns of A, we write $x^T = (x_B^T, x_N^T)$, and

$$A = [\, B \quad N \,],$$

so that $Ax = Bx_B = b$, and $x_B = B^{-1}b$. The following theorems are taken from [106].

Theorem 23.3 *A point x is in Ext(F) if and only if x is a basic feasible solution.*

Proof: Suppose that x is a basic feasible solution, and we write $x^T = (x_B^T, 0^T)$, $A = [\, B \quad N \,]$. If x is not an extreme point of F, then there are $y \neq x$ and $z \neq x$ in F, and α in $(0, 1)$, with

$$x = (1 - \alpha)y + \alpha z.$$

Then $y^T = (y_B^T, y_N^T)$, $z^T = (z_B^T, z_N^T)$, and $y_N \geq 0$, $z_N \geq 0$. From

$$0 = x_N = (1 - \alpha)y_N + (\alpha)z_N$$

it follows that

$$y_N = z_N = 0,$$

and $b = By_B = Bz_B = Bx_B$. But, since B is invertible, we have $x_B = y_B = z_B$. This is a contradiction, so x must be in $\text{Ext}(F)$.

Conversely, suppose that x is in $\text{Ext}(F)$. Since it is in F, we know that $Ax = b$ and $x \geq 0$. By reordering the variables if necessary, we may assume that $x^T = (x_B^T, x_N^T)$, with $x_B > 0$ and $x_N = 0$; we do not know that x_B is a vector of length I, however, so when we write $A = [\, B \quad N \,]$, we do not know that B is square. If B is invertible, then x is a basic feasible solution. If not, we shall construct $y \neq x$ and $z \neq x$ in F, such that

$$x = \frac{1}{2}y + \frac{1}{2}z.$$

If $\{B_1, B_2, ..., B_K\}$ are the columns of B and are linearly dependent, then there are constants $p_1, p_2, ..., p_K$, not all zero, with

$$p_1 B_1 + ... + p_K B_K = 0.$$

With $p^T = (p_1, ..., p_K)$, we have

$$B(x_B + \alpha p) = B(x_B - \alpha p) = Bx_B = b,$$

for all $\alpha \in (0, 1)$. We then select α so small that both $x_B + \alpha p > 0$ and $x_B - \alpha p > 0$. Let

$$y^T = (x_B^T + \alpha p^T, x_N^T)$$

and

$$z^T = (x_B^T - \alpha p^T, x_N^T).$$

This completes the proof. ∎

Exercise 23.2 *Show that there are at most finitely many basic feasible solutions, so there are at most finitely many members of Ext(F).*

Theorem 23.4 *If F is not empty, then Ext(F) is not empty. In that case, let $\{v^1, ..., v^K\}$ be the members of Ext(F). Every x in F can be written as*

$$x = d + \alpha_1 v^1 + ... + \alpha_K v^K,$$

for some $\alpha_k \geq 0$, with $\sum_{k=1}^K \alpha_k = 1$, and some direction of unboundedness, d.

Proof: We consider only the case in which F is bounded, so there is no direction of unboundedness; the unbounded case is similar. Let x be a feasible point. If x is an extreme point, fine. If not, then x is not a basic feasible solution. The columns of A that correspond to the positive entries of x are not linearly independent. Then we can find a vector p such that $Ap = 0$ and $p_j = 0$ if $x_j = 0$. If $|\epsilon|$ is small, $x + \epsilon p \geq 0$ and $(x + \epsilon p)_j = 0$ if $x_j = 0$, then $x + \epsilon p$ is in F. We can alter ϵ in such a way that eventually $y = x + \epsilon p$ has one more zero entry than x has, and so does $z = x - \epsilon p$. Both y and z are in F and x is the average of these points. If y and z are not basic, repeat the argument on y and z, each time reducing the number of positive entries. Eventually, we will arrive at the case where the number of non-zero entries is I, and so will have a basic feasible solution. ∎

Proof of the Strong Duality Theorem: Suppose now that x_* is a solution of the problem (PS) and $z_* = c^T x_*$. Without loss of generality, we may assume that x_* is a basic feasible solution, hence an extreme point of F. Then we can write

$$x_*^T = ((B^{-1}b)^T, 0^T),$$

$$c^T = (c_B^T, c_N^T),$$

and $A = [\, B \quad N \,]$. Every feasible solution has the form

$$x^T = ((B^{-1}b)^T, 0^T) + ((B^{-1}Nv)^T, v^T),$$

for some $v \geq 0$. From $c^T x \geq c^T x_*$ we find that

$$(c_N^T - c_B^T B^{-1} N)(v) \geq 0,$$

for all $v \geq 0$. It follows that

$$c_N^T - c_B^T B^{-1} N = 0.$$

Nw let $y_* = (B^{-1})^T c_B$, or $y_*^T = c_B^T B^{-1}$. We show that y_* is feasible for (DS); that is, we show that

$$A^T y_* \leq c^T.$$

Since

$$y_*^T A = (y_*^T B, y_*^T N) = (c_B^T, y_*^T N) = (c_B^T, c_B^T B^{-1} N)$$

and

$$c_N^T \geq c_B^T B^{-1} N,$$

we have

$$y_*^T A \leq c^T,$$

so y_* is feasible for (DS). Finally, we show that

$$c^T x_* = y_*^T b.$$

We have

$$y_*^T b = c_B^T B^{-1} b = c^T x_*.$$

This completes the proof. ∎

23.2 The Simplex Method

In this section we sketch the main ideas of the simplex method. For further details see [106].

Begin with a basic feasible solution of (PS), say

$$x^T = (\hat{b}^T, 0^T) = ((B^{-1}b)^T, 0^T).$$

Compute the vector $y^T = c_B^T B^{-1}$. If

$$\hat{c}_N^T = c_N^T - y^T N \geq 0,$$

then x is optimal. Otherwise, select a *entering varable* x_j such that

$$(\hat{c}_N)_j < 0.$$

Compute $\hat{a}^j = B^{-1}a^j$, where a^j is the jth column of A. Find an index s such that

$$\frac{\hat{b}_s}{(\hat{a}_j)_s} = \min_{1 \leq i \leq I} \{ \frac{\hat{b}_i}{(\hat{a}_j)_i} : (\hat{a}_j)_i > 0 \}.$$

If there are no such positive denominators, the problem is unbounded. Then x_s is the *leaving variable*, replacing x_j. Redefine B and the basic variables x_B accordingly.

23.3 Convex Programming

Let f and g_i, $i = 1, ..., I$, be convex functions defined on C, a non-empty closed, convex subset of R^J. The *primal problem* in *convex programming* is the following:

$$\text{minimize } f(x), \text{subject to} g_i(x) \leq 0, \text{ for } i = 1, ..., I. \text{ (P)} \tag{23.5}$$

The Lagrangian is

$$L(x, \lambda) = f(x) + \sum_{i=1}^{I} \lambda_i g_i(x).$$

The corresponding dual problem is

$$\text{maximize } h(\lambda) = \inf_{x \in C} L(x, \lambda), \text{ for } \lambda \geq 0. \tag{23.6}$$

23.3.1 An Example

Let $f(x) = \frac{1}{2}||x||_2^2$. The primary problem is to minimize $f(x)$ over all x for which $Ax \geq b$. Then $g_i = b_i - (Ax)_i$, for $i = 1, ..., I$, and the set C is all of R^J. The Lagrangian is then

$$L(x, \lambda) = \frac{1}{2}||x||_2^2 - \lambda^T Ax + \lambda^T b.$$

The infimum over x occurs when $x = A^T\lambda$ and so

$$h(\lambda) = \lambda^T b - \frac{1}{2}||A^T\lambda||_2^2.$$

For any x satisfying $Ax \geq b$ and any $\lambda \geq 0$ we have $h(\lambda) \leq f(x)$. If x^* is the unique solution of the primal problem and λ^* any solution of the dual problem, we have $f(x^*) = h(\lambda^*)$. The point here is that the constraints in the dual problem are easier to implement in an iterative algorithm, so solving the dual problem is the simpler task.

23.3.2 An Iterative Algorithm for the Dual Problem

In [97] Lent and Censor present the following sequential iterative algorithm for solving the dual problem above. At each step only one entry of the current λ is altered. Let a_i denote the i-th row of the matrix A. Having calculated x^k and $\lambda^k > 0$, let $i = k(\mathrm{mod}\, I) + 1$. Then let

$$\theta = (b_i - (a_i)^T x^k)/a_i^T a_i,$$

$$\delta = \max\{-\lambda_i^k, \omega\theta\},$$

and set

$$\lambda_i^{k+1} = \lambda_i^k + \delta,$$

and

$$x^{k+1} = x^k + \delta a_i.$$

Chapter 24

Systems of Linear Inequalities

Designing linear discriminants for pattern classification involves the problem of solving a system of linear inequalities $Ax \geq b$. In this chapter we discuss the iterative Agmon-Motzkin-Schoenberg (AMS) algorithm [1, 104] for solving such problems. We prove convergence of the AMS algorithm, for both the consistent and inconsistent cases, by mimicking the proof for the ART algorithm. Both algorithms are examples of the method of projection onto convex sets. The AMS algorithm is a special case of the cyclic subgradient projection (CSP) method, so that convergence of the AMS, in the consistent case, follows from the convergence theorem for the CSP algorithm.

24.1 Projection onto Convex Sets

In [125] Youla suggests that problems in image restoration might be viewed geometrically and the method of projection onto convex sets (POCS) employed to solve such inverse problems. In the survey paper [124] he examines the POCS method as a particular case of iterative algorithms for finding fixed points of nonexpansive mappings. This point of view is increasingly important in applications such as medical imaging and a number of recent papers have addressed the theoretical and practical issues involved [8], [10], [7], [27], [31], [37], [49], [50], [51].

In this geometric approach the restored image is a solution of the *convex feasibility problem* (CFP), that is, it lies within the intersection of finitely many closed nonempty convex sets $C_i, i = 1, ..., I$, in R^J (or sometimes, in infinite dimensional Hilbert space). For any nonempty closed convex set C, the *metric projection* of x onto C, denoted $P_C x$, is the unique member

of C closest to x. The iterative methods used to solve the CFP employ these metric projections. Algorithms for solving the CFP are discussed in the papers cited above, as well as in the books by Censor and Zenios [45], Stark and Yang [117] and Borwein and Lewis [14].

The simplest example of the CFP is the solving of a system of linear equations $Ax = b$. Let A be an I by J real matrix and for $i = 1, ..., I$ let $B_i = \{x|(Ax)_i = b_i\}$, where b_i denotes the i-th entry of the vector b. Now let $C_i = B_i$. Any solution of $Ax = b$ lies in the intersection of the C_i; if the system is inconsistent then the intersection is empty. The Kaczmarz algorithm [88] for solving the system of linear equations $Ax = b$ has the iterative step

$$x_j^{k+1} = x_j^k + A_{i(k)j}(b_{i(k)} - (Ax^k)_{i(k)}), \tag{24.1}$$

for $j = 1, ..., J$, $k = 0, 1, ...$ and $i(k) = k(\mathrm{mod}\, I) + 1$. This algorithm was rediscovered by Gordon, Bender and Herman [74], who called it the *algebraic reconstruction technique* (ART). This algorithm is an example of the method of *successive orthogonal projections* (SOP) [76] whereby we generate the sequence $\{x^k\}$ by taking x^{k+1} to be the point in $C_{i(k)}$ closest to x^k. Kaczmarz's algorithm can also be viewed as a method for constrained optimization: whenever $Ax = b$ has solutions, the limit of the sequence generated by equation (24.1) minimizes the function $||x - x^0||_2$ over all solutions of $Ax = b$.

In the example just discussed the sets C_i are hyperplanes in R^J; suppose now that we take the C_i to be half-spaces and consider the problem of finding x such that $Ax \geq b$. For each i let H_i be the half-space $H_i = \{x|(Ax)_i \geq b_i\}$. Then x will be in the intersection of the sets $C_i = H_i$ if and only if $Ax \geq b$. Methods for solving this CFP, such as Hildreth's algorithm, are discussed in [45]. Of particular interest for us here is the behavior of the Agmon-Motzkin-Schoenberg (AMS) algorithm (AMS) algorithm [1] [104] for solving such systems of inequalities $Ax \geq b$. The AMS algorithm has the iterative step

$$x_j^{k+1} = x_j^k + A_{i(k)j}(b_{i(k)} - (Ax^k)_{i(k)})_+. \tag{24.2}$$

The AMS algorithm converges to a solution of $Ax \geq b$, if there are solutions. If there are no solutions the AMS algorithm converges cyclically, that is, subsequences associated with the same m converge [60],[10]. We present an elementary proof of this result in this chapter.

Algorithms for solving the CFP fall into two classes: those that employ all the sets C_i at each step of the iteration (the so-called *simultaneous methods*) and those that do not (the *row-action algorithms* or, more generally, *block-iterative methods*).

In the consistent case, in which the intersection of the convex sets C_i is nonempty, all reasonable algorithms are expected to converge to a mem-

ber of that intersection; the limit may or may not be the member of the intersection closest to the starting vector x^0.

In the inconsistent case, in which the intersection of the C_i is empty, simultaneous methods typically converge to a minimizer of a *proximity function* [37], such as

$$f(x) = \sum_{i=1}^{I} ||x - P_{C_i}x||_2^2,$$

if a minimizer exists.

Methods that are not simultaneous cannot converge in the inconsistent case, since the limit would then be a member of the (empty) intersection. Such methods often exhibit what is called *cyclic convergence*; that is, subsequences converge to finitely many distinct limits comprising a limit cycle. Once a member of this limit cycle is reached, further application of the algorithm results in passing from one member of the limit cycle to the next. Proving the existence of these limit cycles seems to be a difficult problem.

Tanabe [118] showed the existence of a limit cycle for Kaczmarz's algorithm (see also [57]), in which the convex sets are hyperplanes. The SOP method may fail to have a limit cycle for certain choices of the convex sets. For example, if, in R^2, we take C_1 to be the lower half-plane and $C_2 = \{(x, y)|x > 0, y \geq 1/x\}$, then the SOP algorithm fails to produce a limit cycle. However, Gubin, Polyak and Riak [76] prove weak convergence to a limit cycle for the method of SOP in Hilbert space, under the assumption that at least one of the C_i is bounded, hence weakly compact. In [10] Bauschke, Borwein and Lewis present a wide variety of results on the existence of limit cycles. In particular, they prove that if each of the convex sets C_i in Hilbert space is a convex polyhedron, that is, the intersection of finitely many half-spaces, then there is a limit cycle and the subsequential convergence is in norm. This result includes the case in which each C_i is a half-space, so implies the existence of a limit cycle for the AMS algorithm. In this paper we give a proof of existence of a limit cycle for the AMS algorithm using a modification of our proof for the ART.

In the next section we consider the behavior of the ART for solving $Ax = b$. The proofs given by Tanabe and Dax of the existence of a limit cycle for this algorithm rely heavily on aspects of the theory of linear algebra, as did the proof given in an earlier chapter here. Our goal now is to obtain a more direct proof that can be easily modified to apply to the AMS algorithm.

We assume throughout this chapter that the real I by J matrix A has full rank and its rows have Euclidean length one.

24.2 Solving $Ax = b$

For $i = 1, 2, ..., I$ let $K_i = \{x|(Ax)_i = 0\}$, $B_i = \{x|(Ax)_i = b_i\}$ and p^i be the metric projection of $x = 0$ onto B_i. Let $v_i^r = (Ax^{rI+i-1})_i$

and $v^r = (v_1^r, ..., v_I^r)^T$, for $r = 0, 1, ...$ We begin with some basic facts concerning the ART.
Fact 1:

$$||x^k||_2^2 - ||x^{k+1}||_2^2 = (A(x^k)_{i(k)})^2 - (b_{i(k)})^2.$$

Fact 2:

$$||x^{rI}||_2^2 - ||x^{(r+1)I}||_2^2 = ||v^r||_2^2 - ||b||_2^2.$$

Fact 3:

$$||x^k - x^{k+1}||_2^2 = ((Ax^k)_{i(k)} - b_{i(k)})^2.$$

Fact 4: There exists $B > 0$ such that, for all $r = 0, 1, ...,$ if $||v^r||_2 \leq ||b||_2$ then $||x^{rI}||_2 \geq ||x^{(r+1)I}||_2 - B$.

Fact 5: Let x^0 and y^0 be arbitrary and $\{x^k\}$ and $\{y^k\}$ the sequences generated by applying the ART. Then

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

24.2.1 When the System $Ax = b$ is Consistent

In this subsection we give a proof of the following result.

Theorem 24.1 *Let $A\hat{x} = b$ and let x^0 be arbitrary. Let $\{x^k\}$ be generated by Equation (24.1). Then the sequence $\{||\hat{x} - x^k||_2\}$ is decreasing and $\{x^k\}$ converges to the solution of $Ax = b$ closest to x^0.*

Proof: Let $A\hat{x} = b$. It follows from Fact 5 that the sequence $\{||\hat{x} - x^{rI}||_2\}$ is decreasing and the sequence $\{v^r - b\} \to 0$. So $\{x^{rI}\}$ is bounded; let $x^{*,0}$ be a cluster point. Then, for $i = 1, 2, ..., I$ let $x^{*,i}$ be the successor of $x^{*,i-1}$ using the ART. It follows that $(Ax^{*,i-1})_i = b_i$ for each i, from which we conclude that $x^{*,0} = x^{*,i}$ for all i and that $Ax^{*,0} = b$. Using $x^{*,0}$ in place of $\hat{x}$, we have that $\{||x^{*,0} - x^k||_2\}$ is decreasing. But a subsequence converges to zero, so $\{x^k\}$ converges to $x^{*,0}$. By Fact 5 the difference $||\hat{x} - x^k||_2^2 - ||\hat{x} - x^{k+1}||_2^2$ is independent of which solution $\hat{x}$ we pick; consequently, so is $||\hat{x} - x^0||_2^2 - ||\hat{x} - x^{*,0}||_2^2$. It follows that $x^{*,0}$ is the solution closest to x^0. This completes the proof. ∎

24.2.2 When the System $Ax = b$ is Inconsistent

In the inconsistent case the sequence $\{x^k\}$ will not converge, since any limit would be a solution. However, for each fixed $i \in \{1, 2, ..., I\}$, the subsequence $\{x^{rI+i}\}$ converges [118], [57]; in this subsection we prove this result and then, in the next section, we extend the proof to get cyclic convergence for the AMS algorithm. We start by showing that the sequence $\{x^{rI}\}$ is bounded. We assume that $I > J$ and A has full rank.

Proposition 24.1 *The sequence $\{x^{rI}\}$ is bounded.*

Proof: Assume that the sequence $\{x^{rI}\}$ is unbounded. We first show that we can select a subsequence $\{x^{r_tI}\}$ with the properties $||x^{r_tI}||_2 \geq t$ and $||v^{r_t}||_2 < ||b||_2$, for $t = 1, 2, ...$

Assume that we have selected x^{r_tI}, with the properties $||x^{r_tI}||_2 \geq t$ and $||v^{r_t}||_2 < ||b||_2$; we show how to select $x^{r_{t+1}I}$. Pick integer $s > 0$ such that

$$||x^{sI}||_2 \geq ||x^{r_tI}||_2 + B + 1,$$

where $B > 0$ is as in Fact 4. With $n + r_t = s$ let $m \geq 0$ be the smallest integer for which

$$||x^{(r_t+n-m-1)I}||_2 < ||x^{sI}||_2 \leq ||x^{(r_t+n-i)I}||_2.$$

Then $||v^{r_t+n-m-1}||_2 < ||b||_2$. Let $x^{r_{t+1}I} = x^{(r_t+n-m-1)I}$. Then we have

$$||x^{r_{t+1}I}||_2 \geq ||x^{(r_t+n-m)I}||_2 - B \geq ||x^{sI}||_2 - B \geq ||x^{r_tI}||_2 + B + 1 - B \geq t + 1.$$

This gives us the desired subsequence.

For every $k = 0, 1, ...$ let $z^{k+1} = x^{k+1} - p^{i(k)}$. Then $z^{k+1} \in K_{i(k)}$. For $z^{k+1} \neq 0$ let $u^{k+1} = z^{k+1}/||z^{k+1}||_2$. Since the subsequence $\{x^{r_tI}\}$ is unbounded, so is $\{z^{r_tI}\}$, so for sufficiently large t the vectors u^{r_tI} are defined and on the unit sphere. Let $u^{*,0}$ be a cluster point of $\{u^{r_tI}\}$; replacing $\{x^{r_tI}\}$ with a subsequence if necessary, assume that the sequence $\{u^{r_tI}\}$ converges to $u^{*,0}$. Then let $u^{*,1}$ be a subsequence of $u^{r_tI+1}\}$; again, assume the sequence $\{u^{r_tI+1}\}$ converges to $u^{*,1}$. Continuing in this manner, we have $\{u^{r_tI+\tau}\}$ converging to $u^{*,\tau}$ for $\tau = 0, 1, 2, ...$ We know that $\{z^{r_tI}\}$ is unbounded and since $||v^{r_t}||_2 < ||b||_2$, we have, by Fact 3, that $\{z^{r_tI+i-1} - z^{r_tI+i}\}$ is bounded for each i. Consequently $\{z^{r_tI+i}\}$ is unbounded for each i.

Now we have

$$||z^{r_tI+i-1} - z^{r_tI+i}||_2 \geq ||z^{r_tI+i-1}||_2 \, ||u^{r_tI+i-1} - \langle u^{r_tI+i-1}, u^{r_tI+i}\rangle u^{r_tI+i}||_2.$$

Since the left side is bounded and $||z^{r_tI+i-1}||_2$ has no infinite bounded subsequence, we conclude that

$$||u^{r_tI+i-1} - \langle u^{r_tI+i-1}, u^{r_jI+i}\rangle u^{r_tI+i}||_2 \to 0.$$

It follows that $u^{*,0} = u^{*,i}$ or $u^{*,0} = -u^{*,i}$ for each $i = 1, 2, ..., I$. Therefore $u^{*,0}$ is in K_i for each i; but, since the null space of A contains only zero, this is a contradiction. This completes the proof of the proposition. ∎

Now we give a proof of the following result.

Theorem 24.2 *Let A be I by J, with $I > J$ and A with full rank. If $Ax = b$ has no solutions, then, for any x^0 and each fixed $i \in \{0, 1, ..., I\}$, the subsequence $\{x^{rI+i}\}$ converges to a limit $x^{*,i}$. Beginning the iteration in Equation (24.1) at $x^{*,0}$, we generate the $x^{*,i}$ in turn, with $x^{*,I} = x^{*,0}$.*

Proof: Let $x^{*,0}$ be a cluster point of $\{x^{rI}\}$. Beginning the ART at $x^{*,0}$ we obtain $x^{*,n}$, for $n = 0, 1, 2, ...$ It is easily seen that

$$||x^{(r-1)I} - x^{rI}||_2^2 - ||x^{rI} - x^{(r+1)I}||_2^2 = \sum_{i=1}^{I} ((Ax^{(r-1)I+i-1})_i - (Ax^{rI+i-1})_i)^2.$$

Therefore the sequence $\{||x^{(r-1)I} - x^{rI}||_2\}$ is decreasing and

$$\{\sum_{i=1}^{I} ((Ax^{(r-1)I+i-1})_i - (Ax^{rI+i-1})_i)^2\} \to 0.$$

Therefore $(Ax^{*,i-1})_i = (Ax^{*,I+i-1})_i$ for each i.

For arbitrary x we have

$$||x - x^{*,0}||^2 - ||x - x^{*,I}||_2^2 = \sum_{i=1}^{I} ((Ax)_i - (Ax^{*,i-1})_i)^2 - \sum_{i=1}^{I} ((Ax)_i - b_i)^2,$$

so that

$$||x - x^{*,0}||_2^2 - ||x - x^{*,I}||_2^2 = ||x - x^{*,I}||_2^2 - ||x - x^{*,2I}||_2^2.$$

Using $x = x^{*,I}$ we have

$$||x^{*,I} - x^{*,0}||_2 = -||x^{*,I} - x^{*,2I}||_2,$$

from which we conclude that $x^{*,0} = x^{*,I}$. From Fact 5 it follows that the sequence $\{||x^{*,0} - x^{rI}||_2\}$ is decreasing; but a subsequence converges to zero, so the entire sequence converges to zero and $\{x^{rI}\}$ converges to $x^{*,0}$. This completes the proof. ∎

Now we turn to the problem $Ax \geq b$.

24.3 The Agmon-Motzkin-Schoenberg algorithm

In this section we are concerned with the behavior of the AMS algorithm for finding x such that $Ax \geq b$, if such x exist. We begin with some basic facts concerning the AMS algorithm.

Let $w_i^r = \min\{(Ax^{rI+i-1})_i, b_i\}$ and $w^r = (w_1^r, ..., w_I^r)^T$, for $r = 0, 1, ...$ The following facts are easily established.

Fact 1a:

$$||x^{rI+i-1}||_2^2 - ||x^{rI+i}||_2^2 = (w_i^r)^2 - (b_i)^2.$$

Fact 2a:

$$||x^{rI}||_2^2 - ||x^{(r+1)I}||_2^2 = ||w^r||_2^2 - ||b||_2^2.$$

Fact 3a:

$$||x^{rI+i-1} - x^{rI+i}||_2^2 = (w_i^r - b_i)^2.$$

Fact 4a: There exists $B > 0$ such that, for all $r = 0, 1, ...$, if $||w^r||_2 \leq ||b||_2$ then $||x^{rI}||_2 \geq ||x^{(r+1)I}||_2 - B$.

Fact 5a: Let x^0 and y^0 be arbitrary and $\{x^k\}$ and $\{y^k\}$ the sequences generated by applying the AMS algorithm. Then
$||x^0 - y^0||_2^2 - ||x^I - y^I||_2^2 =$

$$\sum_{i=1}^{I}((Ax^{i-1})_i - (Ay^{i-1})_i)^2 -$$

$$\sum_{i=1}^{I}(((Ax^{i-1})_i - b_i)_+ - ((Ay^{i-1})_i - b_i)_+)^2 \geq 0.$$

Consider for a moment the elements of the second sum in the inequality above. There are four possibilities:
1) both $(Ax^{i-1})_i - b_i$ and $(Ay^{i-1})_i - b_i$ are nonnegative, in which case this term becomes $((Ax^{i-1})_i - (Ay^{i-1})_i)^2$ and cancels with the same term in the previous sum;
2) neither $(Ax^{i-1})_i - b_i$ nor $(Ay^{i-1})_i - b_i$ is nonnegative, in which case this term is zero;
3) precisely one of $(Ax^{i-1})_i - b_i$ and $(Ay^{i-1})_i - b_i$ is nonnegative; say it is $(Ax^{i-1})_i - b_i$, in which case the term becomes $((Ax^{i-1})_i - b_i)^2$.
Since we then have

$$(Ay^{i-1})_i \leq b_i < (Ax^{i-1})_i$$

it follows that

$$((Ax^{i-1})_i - (Ay^{i-1})_i)^2 \geq ((Ax^{i-1})_i - b_i)^2.$$

We conclude that the right side of the equation in Fact 5a is nonnegative, as claimed.

It will be important in subsequent discussions to know under what conditions the right side of this equation is zero, so we consider that now. We then have

$$((Ax^{i-1})_i - (Ay^{i-1})_i)^2 - (((Ax^{i-1})_i - b_i)_+ - ((Ay^{i-1})_i - b_i)_+)^2 = 0$$

for each m separately, since each of these terms is nonnegative, as we have just seen.

In case 1) above this difference is already zero, as we just saw. In case 2) this difference reduces to $((Ax^{i-1})_i - (Ay^{i-1})_i)^2$, which then is zero precisely when $(Ax^{i-1})_i = (Ay^{i-1})_i$. In case 3) the difference becomes

$$((Ax^{i-1})_i - (Ay^{i-1})_i)^2 - ((Ax^{i-1})_i - b_i)^2,$$

which equals

$$((Ax^{i-1})_i - (Ay^{i-1})_i + (Ax^{i-1})_i - b_i)(b_i - (Ay^{i-1})_i).$$

Since this is zero, it follows that $(Ay^{i-1})_i = b_i$, which contradicts our assumptions in this case. We conclude therefore that the difference of sums in Fact 5a is zero if and only if, for all i, either both $(Ax^{i-1})_i \geq b_i$ and $(Ay^{i-1})_i \geq b_i$ or $(Ax^{i-1})_i = (Ay^{i-1})_i < b_i$.

24.3.1 When $Ax \geq b$ is Consistent

We now prove the following result.

Theorem 24.3 *Let $A\hat{x} \geq b$. Let x^0 be arbitrary and let $\{x^k\}$ be generated by equation (24.2). Then the sequence $\{||\hat{x} - x^k||_2\}$ is decreasing and the sequence $\{x^k\}$ converges to a solution of $Ax \geq b$.*

Proof: Let $A\hat{x} \geq b$. When we apply the AMS algorithm beginning at $\hat{x}$ we obtain $\hat{x}$ again at each step. Therefore, by Fact 5a and the discussion that followed, with $y^0 = \hat{x}$, we have
$||x^k - \hat{x}||_2^2 - ||x^{k+1} - \hat{x}||_2^2 =$

$$((Ax^k)_i - (A\hat{x})_i)^2 - (((Ax^k)_i - b_i)_+ - (A\hat{x})_i + b_i)^2 \geq 0. \tag{24.3}$$

Therefore the sequence $\{||x^k - \hat{x}||_2\}$ is decreasing and so $\{x^k\}$ is bounded; let $x^{*,0}$ be a cluster point.

The sequence defined by the right side of Equation (24.3) above converges to zero. It follows from the discussion following Fact 5a that $Ax^{*,0} \geq b$. Continuing as in the case of $Ax = b$, we have that the sequence $\{x^k\}$ converges to $x^{*,0}$. In general it is not the case that $x^{*,0}$ is the solution of $Ax \geq b$ closest to x^0. ∎

Now we turn to the inconsistent case.

24.3.2 When $Ax \geq b$ is Inconsistent

In the inconsistent case the sequence $\{x^k\}$ will not converge, since any limit would be a solution. However, we do have the following result.

Theorem 24.4 *Let A be I by J, with $I > J$ and A with full rank. Let x^0 be arbitrary. The sequence $\{x^{rI}\}$ converges to a limit $x^{*,0}$. Beginning the AMS algorithm at $x^{*,0}$ we obtain $x^{*,k}$, for $k = 1, 2, ...$. For each fixed $i \in \{0, 1, 2, ..., I\}$, the subsequence $\{x^{rI+i}\}$ converges to $x^{*,i}$ and $x^{*,I} = x^{*,0}$.*

We start by showing that the sequence $\{x^{rI}\}$ is bounded.

Proposition 24.2 *The sequence $\{x^{rI}\}$ is bounded.*

Proof: Assume that the sequence $\{x^{rI}\}$ is unbounded. We first show that we can select a subsequence $\{x^{r_tI}\}$ with the properties $||x^{r_tI}||_2 \geq t$ and $||w^{r_t}||_2 < ||b||_2$, for $t = 1, 2, ...$

Assume that we have selected $x^{r_t I}$, with the properties $||x^{r_t I}||_2 \geq t$ and $||w^{r_t}||_2 < ||b||_2$; we show how to select $x^{r_{t+1} I}$. Pick integer $s > 0$ such that

$$||x^{sI}||_2 \geq ||x^{r_t I}||_2 + B + 1,$$

where $B > 0$ is as in Fact 4a. With $n + r_t = s$ let $m \geq 0$ be the smallest integer for which

$$||x^{(r_t+n-m-1)I}||_2 < ||x^{sI}||_2 \leq ||x^{(r_t+n-m)I}||_2.$$

Then $||w^{r_t+n-m-1}||_2 < ||b||_2$. Let $x^{r_{t+1} I} = x^{(r_t+n-m-1)I}$. Then we have

$$||x^{r_{t+1} I}||_2 \geq ||x^{(r_t+n-m)I}||_2 - B \geq ||x^{sI}||_2 - B \geq ||x^{r_t I}||_2 + B + 1 - B \geq t + 1.$$

This gives us the desired subsequence.

For every $k = 0, 1, ...$ let z^{k+1} be the metric projection of x^{k+1} onto the hyperplane $K_{i(k)}$. Then $z^{k+1} = x^{k+1} - p^{i(k)}$ if $(Ax^k)_i \leq b_i$ and $z^{k+1} = x^{k+1} - (Ax^k)_i A^i$ if not; here A^i is the i-th column of A^T. Then $z^{k+1} \in K_{i(k)}$. For $z^{k+1} \neq 0$ let $u^{k+1} = z^{k+1}/||z^{k+1}||_2$. Let $u^{*,0}$ be a cluster point of $\{u^{r_t I}\}$; replacing $\{x^{r_t I}\}$ with a subsequence if necessary, assume that the sequence $\{u^{r_t I}\}$ converges to $u^{*,0}$. Then let $u^{*,1}$ be a subsequence of $\{u^{r_t I+1}\}$; again, assume the sequence $\{u^{r_t I+1}\}$ converges to $u^{*,1}$. Continuing in this manner, we have $\{u^{r_t I+m}\}$ converging to $u^{*,m}$ for $m = 0, 1, 2, ...$ Since $||w^{r_t}||_2 < ||b||_2$, we have, by Fact 3a, that $\{z^{r_t I+i-1} - z^{r_t I+i}\}$ is bounded for each i. Now we have

$$||z^{r_t I+i-1} - z^{r_t I+i}||_2 \geq ||z^{r_t I+i-1}||_2 \, ||u^{r_t I+i-1} - \langle u^{r_t I+i-1}, u^{r_t+I+i}\rangle u^{r_t I+i}||_2.$$

The left side is bounded. We consider the sequence $||z^{r_t I+i-1}||_2$ in two cases: 1) the sequence is unbounded; 2) the sequence is bounded.

In the first case, it follows, as in the case of $Ax = b$, that $u^{*,i-1} = u^{*,i}$ or $u^{*,i-1} = -u^{*,i}$. In the second case we must have $(Ax^{r_t I+i-1})_i > b_i$ for t sufficiently large, so that, from some point on, we have $x^{r_t I+i-1} = x^{r_t I+i}$, in which case we have $u^{*,i-1} = u^{*,i}$. So we conclude that $u^{*,0}$ is in the null space of A, which is a contradiction. This concludes the proof of the proposition. ∎

Proof of Theorem 24.4: Let $x^{*,0}$ be a cluster point of $\{x^{rI}\}$. Beginning the AMS iteration (24.2) at $x^{*,0}$ we obtain $x^{*,m}$, for $m = 0, 1, 2, ...$ From Fact 5a it is easily seen that the sequence $\{||x^{rI} - x^{(r+1)I}||_2\}$ is decreasing and that the sequence

$$\{\sum\nolimits_{i=1}^{I} ((Ax^{(r-1)I+i-1})_i - (Ax^{rI+i-1})_i)^2 -$$

$$\sum\nolimits_{i=1}^{I} (((Ax^{(r-1)I+i-1})_i - b_i)_+ - ((Ax^{rI+i-1})_i - b_i)_+)^2\} \to 0.$$

Again, by the discussion following Fact 5a, we conclude one of two things: either Case (1): $(Ax^{*,i-1})_i = (Ax^{*,jI+i-1})_i$ for each $j = 1, 2, ...$ or Case (2): $(Ax^{*,i-1})_i > b_i$ and, for each $j = 1, 2, ...$, $(Ax^{*,jI+i-1})_i > b_i$. Let A^i denote the i-th column of A^T. As the AMS iteration proceeds from $x^{*,0}$ to $x^{*,I}$, from $x^{*,I}$ to $x^{*,2I}$ and, in general, from $x^{*,jI}$ to $x^{*,(j+1)I}$ we have either $x^{*,i-1} - x^{*,i} = 0$ and $x^{*,jI+i-1} - x^{*,jI+i} = 0$, for each $j = 1, 2, ...$, which happens in Case (2), or $x^{*,i-1} - x^{*,i} = x^{*,jI+i-1} - x^{*,jI+i} = (b_i - (Ax^{*,i-1})_i)A^i$, for $j = 1, 2, ...$, which happens in Case (1). It follows, therefore, that

$$x^{*,0} - x^{*,I} = x^{*,jI} - x^{*,(j+1)I}$$

for $j = 1, 2, ...$. Since the original sequence $\{x^{rI}\}$ is bounded, we have

$$||x^{*,0} - x^{*,jI}||_2 \leq ||x^{*,0}||_2 + ||x^{*,jI}||_2 \leq K$$

for some K and all $j = 1, 2, ...$. But we also have

$$||x^{*,0} - x^{*,jI}||_2 = j||x^{*,0} - x^{*,I}||_2.$$

We conclude that $||x^{*,0} - x^{*,I}||_2 = 0$ or $x^{*,0} = x^{*,I}$.

From Fact 5a, using $y^0 = x^{*,0}$, it follows that the sequence $\{||x^{*,0} - x^{rI}||_2\}$ is decreasing; but a subsequence converges to zero, so the entire sequence converges to zero and $\{x^{rI}\}$ converges to $x^{*,0}$. This completes the proof of Theorem 24.4. ∎

Chapter 25

The Split Feasibility Problem

The *split feasibility problem* (SFP) [40] 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.

25.1 The CQ Algorithm

In [31] the CQ algorithm for solving the SFP was presented, for the real case. It has the iterative step

$$x^{k+1} = P_C(x^k - \gamma A^T(I - P_Q)Ax^k), \tag{25.1}$$

where I is the identity operator and $\gamma \in (0, 2/\rho(A^T A))$, for $\rho(A^T A)$ the spectral radius of the matrix $A^T A$, which is also its largest eigenvalue. The CQ algorithm can be extended to the complex case, in which the matrix A has complex entries, and the sets C and Q are in C^J and C^I, respectively. The iterative step of the extended CQ algorithm is then

$$x^{k+1} = P_C(x^k - \gamma A^\dagger(I - P_Q)Ax^k). \tag{25.2}$$

The CQ algorithm converges to a solution of the SFP, for any starting vector x^0, whenever the SFP has solutions. When the SFP has no solutions, the CQ algorithm converges to a minimizer of the function

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

over the set C, provided such constrained minimizers exist. Therefore the CQ algorithm is an iterative constrained optimization method. As shown in [32], convergence of the CQ algorithm is a consequence of Theorem 4.1.

The function $f(x)$ is convex and differentiable on R^J and its derivative is the operator

$$\nabla f(x) = A^T(I - P_Q)Ax;$$

see [3].

Lemma 25.1 *The derivative operator ∇f is λ-Lipschitz continuous for $\lambda = \rho(A^TA)$, therefore it is ν-ism for $\nu = \frac{1}{\lambda}$.*

Proof: We have

$$||\nabla f(x) - \nabla f(y)||_2^2 = ||A^T(I - P_Q)Ax - A^T(I - P_Q)Ay||_2^2$$
$$\leq \lambda||(I - P_Q)Ax - (I - P_Q)Ay||_2^2.$$

Also

$$||(I - P_Q)Ax - (I - P_Q)Ay||_2^2 = ||Ax - Ay||_2^2$$
$$+||P_QAx - P_QAy||_2^2 - 2\langle P_QAx - P_QAy, Ax - Ay\rangle$$

and, since P_Q is fne,

$$\langle P_QAx - P_QAy, Ax - Ay\rangle \geq ||P_QAx - P_QAy||_2^2.$$

Therefore,

$$||\nabla f(x) - \nabla f(y)||_2^2 \leq \lambda(||Ax - Ay||_2^2 - ||P_QAx - P_QAy||_2^2)$$
$$\leq \lambda||Ax - Ay||_2^2 \leq \lambda^2||x - y||_2^2.$$

This completes the proof. ∎

If $\gamma \in (0, 2/\lambda)$ then $B = P_C(I - \gamma A^T(I - P_Q)A)$ is av and, by Theorem 4.1, the orbit sequence $\{B^kx\}$ converges to a fixed point of B, whenever such points exist. If z is a fixed point of B, then $z = P_C(z - \gamma A^T(I - P_Q)Az)$. Therefore, for any c in C we have

$$\langle c - z, z - (z - \gamma A^T(I - P_Q)Az)\rangle \geq 0.$$

This tells us that

$$\langle c - z, A^T(I - P_Q)Az\rangle \geq 0,$$

which means that z minimizes $f(x)$ relative to the set C.

The CQ algorithm employs the relaxation parameter γ in the interval $(0, 2/L)$, where L is the largest eigenvalue of the matrix A^TA. Choosing the best relaxation parameter in any algorithm is a nontrivial procedure. Generally speaking, we want to select γ near to $1/L$. We saw a simple estimate for L in our discussion of singular values of sparse matrices: if A is normalized so that each row has length one, then the spectral radius of A^TA does not exceed the maximum number of nonzero elements in any column of A. A similar upper bound on $\rho(A^TA)$ was obtained for non-normalized, ϵ-sparse A.

25.2 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 [92] and projected Landweber methods (see [12]), are particular cases of the CQ algorithm.

25.2.1 The Landweber algorithm

With x^0 arbitrary and $k = 0, 1, ...$ let

$$x^{k+1} = x^k + \gamma A^T(b - Ax^k). \tag{25.1}$$

This is the Landweber algorithm.

25.2.2 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

$$x^{k+1} = P_C(x^k + \gamma A^T(b - Ax^k)). \tag{25.2}$$

25.2.3 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.

25.2.4 The Simultaneous ART (SART)

Another example of the CQ algorithm is the *simultaneous algebraic reconstruction technique* (SART) [2] 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 4.1, once we know that the largest eigenvalue of B^TB is less than two; in fact, we show that it is one [31].

If B^TB had an eigenvalue greater than one and some of the entries of A are zero, then, replacing these zero entries with very small positive entries, we could obtain a new A whose associated B^TB also had an eigenvalue greater than one. Therefore, we assume, without loss of generality, that A has all positive entries. Since the new B^TB also has only positive entries, this matrix is irreducible and the Perron-Frobenius theorem applies. We shall use this to complete the proof.

Let $u = (u_1, ..., u_J)^T$ with $u_j = (A_{+j})^{1/2}$ and $v = (v_1, ..., v_I)^T$, with $v_i = (A_{i+})^{1/2}$. Then we have $Bu = v$ and $B^Tv = u$; that is, u is an eigenvector of B^TB with associated eigenvalue equal to one, and all the entries of u are positive, by assumption. The Perron-Frobenius theorem applies and tells us that the eigenvector associated with the largest eigenvalue has all positive entries. Since the matrix B^TB is symmetric its eigenvectors are orthogonal; therefore u itself must be an eigenvector associated with the largest eigenvalue of B^TB. The convergence of SART follows.

25.2.5 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 [68] 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

$$x^{t+1}_j - x^t_j \geq 0,$$

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:

$$(x^{t+1}_j - x^t_j) - (x^{t+2}_j - x^{t+1}_j) \geq 0.$$

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.

25.2.6 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 [123].

In a recent paper [41] 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 26

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.

26.1 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 [26], in which the

vectors u and v were called a and b, hence the names of the algorithms. Throughout this chapter we shall assume that the entries of the matrix A are nonnegative. We shall denote by B_n, $n = 1, ..., N$ a partition of the index set $\{i = 1, ..., I\}$ into blocks. For $k = 0, 1, ...$ let $n(k) = k(\mathrm{mod}\, N)+1$.

The projected Landweber algorithm can also be used to impose the restrictions $u_j \leq x_j \leq v_j$; however, the projection step in that algorithm is implemented by clipping, or setting equal to u_j or v_j values of x_j that would otherwise fall outside the desired range. The result is that the values u_j and v_j can occur more frequently than may be desired. One advantage of the AB methods is that the values u_j and v_j represent barriers that can only be reached in the limit and are never taken on at any step of the iteration.

26.2 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

$$x_j^{k+1} = \alpha_j^k v_j + (1 - \alpha_j^k) u_j, \tag{26.1}$$

with $n = n(k)$,

$$\alpha_j^k = \frac{c_j^k \prod^n (d_i^k)^{A_{ij}}}{1 + c_j^k \prod^n (d_i^k)^{A_{ij}}}, \tag{26.2}$$

$$c_j^k = \frac{(x_j^k - u_j)}{(v_j - x_j^k)}, \tag{26.3}$$

and

$$d_j^k = \frac{(b_i - (Au)_i)((Av)_i - (Ax^k)_i)}{((Av)_i - b_i)((Ax^k)_i - (Au)_i)}, \tag{26.4}$$

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 26.1 *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 [26].

26.3 The ABEMML Algorithm

We make the same assumptions as in the previous section. The iterative step of the ABEMML algorithm is

$$x_j^{k+1} = \alpha_j^k v_j + (1 - \alpha_j^k) u_j, \tag{26.5}$$

where

$$\alpha_j^k = \gamma_j^k / d_j^k, \tag{26.6}$$

$$\gamma_j^k = (x_j^k - u_j) e_j^k, \tag{26.7}$$

$$\beta_j^k = (v_j - x_j^k) f_j^k, \tag{26.8}$$

$$d_j^k = \gamma_j^k + \beta_j^k, \tag{26.9}$$

$$e_j^k = \left(1 - \sum_{i \in B_n} A_{ij}\right) + \sum_{i \in B_n} A_{ij} \left(\frac{b_i - (Au)_i}{(Ax^k)_i - (Au)_i}\right), \tag{26.10}$$

and

$$f_j^k = \left(1 - \sum_{i \in B_n} A_{ij}\right) + \sum_{i \in B_n} A_{ij} \left(\frac{(Av)_i - b_i}{(Av)_i - (Ax^k)_i}\right). \tag{26.11}$$

We have the following theorem concerning the convergence of the ABEMML algorithm:

Theorem 26.2 *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 [26]. 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 27

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. The reader should consult the appendices for brief discussion of some of the applications of these methods.

27.1 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.

27.2 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.

27.2.1 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 27.1 *Show that every vector x in R^J can be written as*

$$x = A^T z + w, \tag{27.1}$$

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

$$\hat{x} = A^T z = A^T (AA^T)^{-1} b. \tag{27.2}$$

Hint: multiply both sides of Equation (27.1) 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:

$$\hat{x} = \sum_{i=1}^{I} z_i a^i.$$

27.2.2 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 (27.2), 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

$$\hat{x}_Q = C^{-1}\hat{y} = Q^{-1}A^T(AQ^{-1}A^T)^{-1}b, \tag{27.3}$$

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 $i = 1, ..., I$.

27.3 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

$$||f||_2 = \sqrt{\langle f, f\rangle} = \sqrt{\int |f(x)|^2 dx}.$$

27.3.1 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

$$a_n = \frac{1}{2\pi} F(\omega_n). \tag{27.4}$$

In most applications, however, the function $f(x)$ is known to have finite support.

Exercise 27.2 *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 27.3 *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*

$$a_n = \frac{1}{2\pi} F(\omega_n).$$

27.3.2 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

$$\hat{f}_p(x) = p(x) \sum_{n=1}^{N} b_n e^{-ix\omega_n}. \tag{27.5}$$

Exercise 27.4 *Show that the coefficients b_n satisfy the system of linear equations*

$$F(\omega_n) = \sum_{m=1}^{N} b_m P_{nm}, \tag{27.6}$$

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 (27.5), 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 [34].

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 (27.6), 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 [34]).

27.3.3 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.

27.4 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.

27.4.1 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.

27.4.2 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 algorithm, as we discussed earlier.

Part VII

Applications

Chapter 28

Detection 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.

28.1 Estimation

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

$$\hat{\gamma} = b^\dagger x = \sum_{n=1}^{N} \overline{b_n} x_n, \tag{28.1}$$

where $b^\dagger$ 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) [34] for the particular estimation problem.

28.1.1 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

$$x = \gamma u + q, \tag{28.2}$$

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

$$b = \frac{1}{u^\dagger Q^{-1} u} Q^{-1} u. \tag{28.3}$$

The BLUE estimate of γ is

$$\hat{\gamma} = \frac{1}{u^\dagger Q^{-1} u} u^\dagger Q^{-1} x. \tag{28.4}$$

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.

28.1.2 A known signal vector in noise

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

$$x = \gamma s + q, \tag{28.5}$$

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

$$b = \frac{1}{s^\dagger Q^{-1} s} Q^{-1} s \tag{28.6}$$

and the BLUE estimate of γ is now

$$\hat{\gamma} = \frac{1}{s^\dagger Q^{-1} s} s^\dagger Q^{-1} x. \tag{28.7}$$

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

$$e_\theta = \frac{1}{\sqrt{N}}(e^{-i\theta}, e^{-2i\theta}, ..., e^{-Ni\theta})^T, \tag{28.8}$$

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

$$\hat{\gamma} = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} x_n e^{in\theta}, \tag{28.9}$$

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

28.1.3 Multiple signals in noise

Suppose now that the data values are

$$x_n = \sum_{m=1}^{M} \gamma_m s_n^m + q_n, \tag{28.10}$$

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

$$x = Sc + q, \tag{28.11}$$

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

$$\hat{c} = (S^\dagger Q^{-1} S)^{-1} S^\dagger Q^{-1} x, \tag{28.12}$$

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

$$\hat{c}_m = \sum_{n=1}^{N} x_n \overline{s_n^m}. \tag{28.13}$$

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

$$\hat{c}_m = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} x_n e^{2\pi i m n/M}, \tag{28.14}$$

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.

28.2 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 $\hat{c}$.

28.2.1 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 [34]. For each fixed value of the parameter θ, we estimate γ using the BLUE, obtaining the estimate

$$\hat{\gamma}(\theta) = \frac{1}{s_\theta^\dagger Q^{-1} s_\theta} s_\theta^\dagger Q^{-1} x, \tag{28.15}$$

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

$$x = \int_{\theta \in \Theta} \gamma(\theta) s_\theta d\theta + q. \tag{28.16}$$

Let S now denote the integral operator

$$S(\gamma) = \int_{\theta \in \Theta} \gamma(\theta) s_\theta d\theta \tag{28.17}$$

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

$$S^\dagger(v)(\theta) = \sum_{n=1}^{N} v_n \overline{(s_\theta)_n} = s_\theta^\dagger v \,. \tag{28.18}$$

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

$$g(\theta) = (S^\dagger Q^{-1} S)(\theta) = \sum_{n=1}^{N} \sum_{j=1}^{N} Q_{nj}^{-1} (s_\theta)_j \overline{(s_\theta)_n}, \tag{28.19}$$

so

$$g(\theta) = s_\theta^\dagger Q^{-1} s_\theta. \tag{28.20}$$

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

$$\hat{\gamma}(\theta) = \frac{1}{g(\theta)} \sum_{j=1}^{N} a_j \overline{(s_\theta)_j} = \frac{1}{g(\theta)} s_\theta^\dagger a \,, \tag{28.21}$$

where $x = Qa$ or

$$x_n = \sum_{j=1}^{N} a_j Q_{nj}, \tag{28.22}$$

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

$$g(\theta) = \sum_{n=1}^{N} |(s_\theta)_n|^2, \tag{28.23}$$

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

$$\hat{\gamma}(\theta) = \sum_{n=1}^{N} x_n \overline{(s_\theta)_n} = s_\theta^\dagger x. \tag{28.24}$$

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

$$\hat{\gamma}(\theta) = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} x_n e^{in\theta}, \tag{28.25}$$

again, the DFT of the data vector.

28.3 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.

28.3.1 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 [72], with the channels chosen to capture image information within selected frequency bands.

28.3.2 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.

28.4 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.

28.4.1 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 [63]. 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 [63] 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 $Ab \geq \epsilon$ [34].

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.

28.4.2 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.

28.5 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.

28.5.1 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

$$\mu_A = \frac{1}{J} \sum_{k=1}^{J} x^k, \tag{28.26}$$

$$\mu_B = \frac{1}{K-J} \sum_{k=J+1}^{K} x^k, \tag{28.27}$$

$$m_A = \frac{1}{J} \sum_{k=1}^{J} y_k = w^T \mu_A, \tag{28.28}$$

and

$$m_B = \frac{1}{K-J} \sum_{k=J+1}^{K} y_k = w^T \mu_B. \tag{28.29}$$

Let

$$\sigma_A^2 = \sum_{k=1}^{J} (y_k - m_A)^2, \tag{28.30}$$

and

$$\sigma_B^2 = \sum_{k=J+1}^{K} (y_k - m_B)^2. \tag{28.31}$$

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

$$F(w) = \frac{(m_A - m_B)^2}{\sigma^2}. \tag{28.32}$$

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:

$$S_A = \sum_{k=1}^{J} (x^k - \mu_A)(x^k - \mu_A)^T, \tag{28.33}$$

and

$$S_B = \sum_{k=J+1}^{K} (x^k - \mu_B)(x^k - \mu_B)^T. \tag{28.34}$$

Then

$$S_{within} = S_A + S_B \tag{28.35}$$

is the *within-class scatter matrix* and

$$S_{between} = (\mu_A - \mu_B)(\mu_A - \mu_B)^T \tag{28.36}$$

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

$$F(w) = w^T S_{between} w / w^T S_{within} w. \tag{28.37}$$

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

$$w = S_{within}^{-1}(\mu_A - \mu_B). \tag{28.38}$$

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

Chapter 29

Tomography

In this chapter we present a brief overview of transmission and emission tomography. These days, the term *tomography* is used by lay people and practitioners alike to describe any sort of scan, from ultrasound to magnetic resonance. It has apparently lost its association with the idea of slicing, as in the expression *three-dimensional tomography*. In this chapter we focus on two important modalities, transmission tomography and emission tomography. An x-ray CAT scan is an example of the first, a positron-emission (PET) scan is an example of the second.

29.1 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. In the continuous formulation, the data are line integrals of the function of interest.

When an x-ray beam travels along a line segment through the body it becomes progressively weakened by the material it encounters. By comparing the initial strength of the beam as it enters the body with its final strength as it exits the body, we can estimate the integral of the attenuation function, along that line segment. The data in transmission tomography are these line integrals, corresponding to thousands of lines along which the beams have been sent. The image reconstruction problem is to create a discrete approximation of the attenuation function. The inherently three-dimensional problem is usually solved one two-dimensional plane, or slice, at a time, hence the name *tomography* [78].

The beam attenuation at a given point in the body will depend on the material present at that point; estimating and imaging the attenuation as a

function of spatial location will give us a picture of the material within the body. A bone fracture will show up as a place where significant attenuation should be present, but is not.

29.1.1 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

$$\frac{dI}{ds} = -f(s)I(s).$$

Exercise 29.1 *Show that the solution to this differential equation is*

$$I(s) = I(0) \exp(-\int_{u=0}^{u=s} f(u)du).$$

Hint: Use an integrating factor.

From knowledge of I_{in} and I_{out}, we can determine $\int_L f$. If we know $\int_L f$ for every line in the x, y-plane we can reconstruct the attenuation function f. In the real world we know line integrals only approximately and only for finitely many lines. The goal in x-ray transmission tomography is to estimate the attenuation function $f(x, y)$ in the slice, from finitely many noisy measurements of the line integrals. We usually have prior information about the values that $f(x, y)$ can take on. We also expect to find sharp boundaries separating regions where the function $f(x, y)$ varies only slightly. Therefore, we need algorithms capable of providing such images. As we shall see, the line-integral data can be viewed as values of the Fourier transform of the attenuation function.

29.1.2 Reconstruction from Line Integrals

We turn now to the underlying problem of reconstructing such functions from line-integral data. Our goal is to reconstruct the function $f(x, y)$ from line-integral data. Let θ be a fixed angle in the interval $[0, \pi)$, and consider the rotation of the x, y -coordinate axes to produce the t, s-axis system, where

$$t = x\cos\theta + y\sin\theta,$$

and

$$s = -x\sin\theta + y\cos\theta.$$

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 $\int f(x, y)ds$, obtaining the integral of $f(x, y) = f(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta)$ 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.

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,

$$R_f(\theta, \omega) = \int (\int f(x, y)ds)e^{i\omega t}dt,$$

which we can write as

$$R_f(\theta, \omega) = \int\int f(x, y)e^{i\omega(x\cos\theta + y\sin\theta)}dxdy = F(\omega\cos\theta, \omega\sin\theta),$$

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

$$f(x, y) = \frac{1}{4\pi^2}\int\int F(u, v)e^{-i(xu+yv)}dudv. \tag{29.1}$$

The *filtered backprojection* methods commonly used in the clinic are derived from different ways of calculating the double integral in Equation (29.1).

29.1.3 The Algebraic Approach

Although there is some flexibility in the mathematical description of the image reconstruction problem in transmission tomography, one popular approach is the algebraic formulation of the problem. In this formulation, the problem is to solve, at least approximately, a large system of linear equations, $Ax = b$.

The attenuation function is discretized, in the two-dimensional case, by imagining the body to consist of finitely many squares, or *pixels*, within which the function has a constant, but unknown, value. This value at the j-th pixel is denoted x_j. In the three-dimensional formulation, the body is viewed as consisting of finitely many cubes, or *voxels*. The beam is sent through the body along various lines and both initial and final beam strength is measured. From that data we can calculate a discrete line integral along each line. For $i = 1, ..., I$ we denote by L_i the i-th line segment through the body and by b_i its associated line integral. Denote by A_{ij} the length of the intersection of the j-th pixel with L_i; therefore, A_{ij} is nonnegative. Most of the pixels do not intersect line L_i, so A is quite sparse. Then the data value b_i can be described, at least approximately, as

$$b_i = \sum_{j=1}^{J} A_{ij} x_j. \tag{29.2}$$

Both I, the number of lines, and J, the number of pixels or voxels, are quite large, although they certainly need not be equal, and are typically unrelated.

The matrix A is large and rectangular. The system $Ax = b$ may or may not have exact solutions. We are always free to select J, the number of pixels, as large as we wish, limited only by computation costs. We may also have some choice as to the number I of lines, but within the constraints posed by the scanning machine and the desired duration and dosage of the scan. When the system is underdetermined ($J > I$), there may be infinitely many exact solutions; in such cases we usually impose constraints and prior knowledge to select an appropriate solution. As we mentioned earlier, noise in the data, as well as error in our model of the physics of the scanning procedure, may make an exact solution undesirable, anyway. When the system is overdetermined ($J < I$), we may seek a least-squares approximate solution, or some other approximate solution. We may have prior knowledge about the physics of the materials present in the body that can provide us with upper bounds for x_j, as well as information about body shape and structure that may tell where $x_j = 0$. Incorporating such information in the reconstruction algorithms can often lead to improved images [105].

29.2 Emission Tomography

In *single-photon emission tomography* (SPECT) and *positron emission tomography* (PET) the patient is injected with, or inhales, a chemical to which a radioactive substance has been attached [121]. The chemical is designed to become concentrated in the particular region of the body under study. Once there, the radioactivity results in photons that travel through the body and, at least some of the time, are detected by the scanner. The function of interest is the actual concentration of the radioactive material at each spatial location within the region of interest. Learning what the concentrations are will tell us about the functioning of the body at the various spatial locations. Tumors may take up the chemical (and its radioactive passenger) more avidly than normal tissue, or less avidly, perhaps. Malfunctioning portions of the brain may not receive the normal amount of the chemical and will, therefore, exhibit an abnormal amount of radioactivity.

As in the transmission tomography case, this nonnegative function is discretized and represented as the vector x. The quantity b_i, the i-th entry of the vector b, is the photon count at the i-th detector; in coincidence-detection PET a detection is actually a nearly simultaneous detection of a photon at two different detectors. The entry A_{ij} of the matrix A is the probability that a photon emitted at the j-th pixel or voxel will be detected at the i-th detector.

In the emission tomography case it is common to take a statistical view [94, 93, 112, 115, 120], in which the quantity x_j is the expected number of emissions at the j-th pixel during the scanning time, so that the expected count at the i-th detector is

$$E(b_i) = \sum_{j=1}^{J} A_{ij} x_j. \tag{29.3}$$

The system of equations $Ax = b$ is obtained by replacing the expected count, $E(b_i)$, with the actual count, b_i; obviously, an exact solution of the system is not needed in this case. As in the transmission case, we seek an approximate, and nonnegative, solution of $Ax = b$, where, once again, all the entries of the system are nonnegative.

29.2.1 Maximum-Likelihood Parameter Estimation

The measured data in tomography are values of random variables. The probabilities associated with these random variables are used in formulating the image reconstruction problem as one of solving a large system of linear equations. We can also use the stochastic model of the data to formulate the problem as a statistical parameter-estimation problem, which suggests the image be estimated using likelihood maximization. When formulated

that way, the problem becomes a constrained optimization problem. The desired image can then be calculated using general-purpose iterative optimization algorithms, or iterative algorithms designed specifically to solve the particular problem.

29.3 Image Reconstruction in Tomography

Image reconstruction from tomographic data is an increasingly important area of applied numerical linear algebra, particularly for medical diagnosis [74, 78, 89, 107, 108, 120, 121] . In the algebraic approach, the problem is to solve, at least approximately, a large system of linear equations, $Ax = b$. The vector x is large because it is usually a vectorization of a discrete approximation of a function of two or three continuous spatial variables. The size of the system necessitates the use of iterative solution methods [95]. Because the entries of x usually represent intensity levels, of beam attenuation in transmission tomography, and of radionuclide concentration in emission tomography, we require x to be nonnegative; the physics of the situation may impose additional constraints on the entries of x. In practice, we often have prior knowledge about the function represented, in discrete form, by the vector x and we may wish to include this knowledge in the reconstruction. In tomography the entries of A and b are also nonnegative. Iterative algorithms tailored to find solutions to these special, constrained problems may out-perform general iterative solution methods [105]. To be medically useful in the clinic, the algorithms need to produce acceptable reconstructions early in the iterative process.

The *Fourier* approach to tomographic image reconstruction maintains, at least initially, the continuous model for the attenuation function. The data are taken to be line integrals through the attenuator, that is, values of its so-called *x-ray transform*, which, in the two-dimensional case, is the Radon transform. The Central Slice Theorem then relates the Radon-transform values to values of the Fourier transform of the attenuation function. Image reconstruction then becomes estimation of the (inverse) Fourier transform. In magnetic-resonance imaging (MRI), we again have the measured data related to the function we wish to image, the proton density function, by a Fourier relation.

In the transmission and emission tomography, the data are photon counts, so it is natural to adopt a statistical model and to convert the image reconstruction problem into a statistical parameter-estimation problem. The estimation can be done using maximum likelihood (ML) or maximum *a posteriori* (MAP) Bayesian methods, which then require iterative optimization algorithms.

Chapter 30

Intensity-Modulated Radiation Therapy

In [41] Censor *et al.* extend the CQ algorithm to solve what they call the *multiple-set split feasibility problem* (MSSFP) . In the sequel [42] 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.

30.1 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

$$p(x) = \frac{1}{2}\sum_{n=1}^{N} \alpha_n ||P_{C_n}x - x||_2^2 + \frac{1}{2}\sum_{m=1}^{M} \beta_m ||P_{Q_m}Dx - Dx||_2^2, \quad (30.1)$$

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 [41] has the iterative step

$$x^{k+1} = P_\Omega\left(x^k + s\Big(\sum_{n=1}^{N}\alpha_n(P_{C_n}x^k - x^k) + \sum_{m=1}^{M}\beta_m D^T(P_{Q_m}Dx^k - Dx^k)\Big)\right), \quad (30.2)$$

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.

30.2 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).

30.3 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

$$e_t(h) = (\frac{1}{N_t}\sum_{i\in S_t}(h_i)^\alpha)^{1/\alpha}, \quad (30.3)$$

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.

30.4 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.

Chapter 31

Magnetic-Resonance Imaging

Fourier-transform estimation and extrapolation techniques play a major role in the rapidly expanding field of *magnetic-resonance imaging* (MRI).

31.1 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. Within a single volume element of the body, there will be many protons. In elements with an odd number of protons, the nucleus itself will have a net magnetic moment. In much of *magnetic-resonance imaging* (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.

In the absence of an external magnetic field, the axes of these magnetic dipole moments have random orientation, dictated mainly by thermal effects. When a magnetic field is introduced, it induces a small fraction of the dipole moments to begin to align their axes with that of the 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.

The axes of the magnetic dipole moments precess 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. When the body is

probed with an electromagnetic field at a given frequency, a resonance signal is produced by those protons whose spin axes are precessing at that frequency. The strength of the signal is proportional to the proton density within the targeted volume. The received signal is then processed to obtain information about that proton density.

As we shall see, when the external magnetic field is appropriately chosen, a Fourier relationship is established between the information extracted from the received signal and the proton density.

31.2 The External Magnetic Field

The external magnetic field generated in the MRI scanner is

$$H(r,t) = (H_0 + \mathbf{G}(t) \cdot \mathbf{r})\mathbf{k} + H_1(t)(\cos(\omega_0 t)\mathbf{i} + \sin(\omega_0 t)\mathbf{j}), \tag{31.1}$$

where $\mathbf{r} = (x, y, z)$ is the spatial position vector, and ω_0 is the Larmor frequency associated with the static field intensity H_0, that is,

$$\omega_0 = \gamma H_0,$$

with γ the gyromagnetic ratio. The vectors $\mathbf{i}, \mathbf{j}$, and $\mathbf{k}$ are the unit vectors along the coordinate axes. The vector-valued function $\mathbf{G}(t)$ 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. If $\mathbf{G}(t) = \theta$, for some direction vector θ, then the Larmor frequency is constant on planes normal to θ. In that case, when the body is probed with an electromagnetic field of frequency

$$\omega = \gamma(H_0 + s),$$

there is a resonance signal received from the locations $\mathbf{r}$ lying in the plane $\theta \cdot \mathbf{r} = s$. The strength of the received signal is proportional to the integral, over that 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 $\mathbf{G}(t)$.

31.3 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

signal produced by the probed precessing magnetic dipole moments is approximately

$$S(t) = \int_{R^3} M_0(\mathbf{r}) \exp(-i\gamma(\int_0^t \mathbf{G}(s)ds) \cdot \mathbf{r}) \exp(-t/T_2)d\mathbf{r}, \tag{31.2}$$

where $M_0(\mathbf{r})$ is the local magnetization, which is proportional to the proton density function, and T_2 is the *transverse* or *spin-spin* relaxation time.

31.3.1 An Example of $\mathbf{G}(t)$

Suppose now that $g > 0$ and θ is an arbitrary direction vector. Let

$$\mathbf{G}(t) = g\theta, \text{ for } \tau \leq t, \tag{31.3}$$

and $\mathbf{G}(t) = 0$ otherwise. Then the received signal $S(t)$ is

$$S(t) = \int_{R^3} M_0(\mathbf{r}) \exp(-i\gamma g(t-\tau)\theta \cdot \mathbf{r})d\mathbf{r}$$

$$= (2\pi)^{3/2} \hat{M}_0(\gamma g(t-\tau)\theta), \tag{31.4}$$

for $\tau \leq t << T_2$, where $\hat{M}_0$ denotes the three-dimensional Fourier transform of the function $M_0(\mathbf{r})$.

From Equation (31.4) 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 M_0 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 $M_0(\mathbf{r})$ 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 $M_0(\mathbf{r})$ from finitely many noisy values of its Fourier transform. We can do this by selecting the gradient field $\mathbf{G}(t)$ differently.

31.3.2 Another Example of $\mathbf{G}(t)$

The vector-valued function $\mathbf{G}(t)$ can be written as

$$\mathbf{G}(t) = (G_1(t), G_2(t), G_3(t)).$$

Now we let

$$G_2(t) = g_2,$$

and

$$G_3(t) = g_3,$$

for $0 \leq t \leq \tau$, and zero otherwise, and

$$G_1(t) = g_1,$$

for $\tau \leq t$, and zero otherwise. This means that only $H_0\mathbf{k}$ and the rf field are present up to time τ, and then the rf field is shut off and the gradient field is turned on. Then, for $t \geq \tau$, we have

$$S(t) = (2\pi)^{3/2}\hat{M}_0(\gamma(t-\tau)g_1, \gamma\tau g_2, \gamma\tau g_3).$$

By selecting

$$t_n = n\Delta t + \tau, \text{for } n = 1, ..., N,$$

$$g_{2k} = k\Delta g,$$

and

$$g_{3i} = i\Delta g,$$

for $i, k = -m, ..., m$ we have values of the Fourier transform, $\hat{M}_0$, on a Cartesian grid in three-dimensional k-space. The local magnetization function, M_0, can then be approximated using the fast Fourier transform.

Chapter 32

Hyperspectral Imaging

Hyperspectral image processing provides an excellent example of the need for estimating Fourier transform values from limited data. In this chapter we describe one novel approach, due to Mooney et al. [103]; the presentation here follows [18].

32.1 Spectral Component Dispersion

In this hyperspectral-imaging problem the electromagnetic energy reflected or emitted by a point, such as light reflected from a location on the earth's surface, is passed through a prism to separate the components as to their wavelengths. Due to the dispersion of the different frequency components caused by the prism, these components are recorded in the image plane not at a single spatial location, but at distinct points along a line. Since the received energy comes from a region of points, not a single point, what is received in the image plane is a superposition of different wavelength components associated with different points within the object. The first task is to reorganize the data so that each location in the image plane is associated with all the components of a single point of the object being imaged; this is a Fourier-transform estimation problem, which we can solve using band-limited extrapolation.

The points of the image plane are in one-to-one correspondence with points of the object. These spatial locations in the image plane and in the object are discretized into finite two-dimensional grids. Once we have reorganized the data we have, for each grid point in the image plane, a function of wavelength, describing the intensity of each component of the energy from the corresponding grid point on the object. Practical considerations limit the fineness of the grid in the image plane; the resulting discretization of the object is into pixels. In some applications, such as

satellite imaging, a single pixel may cover an area several meters on a side. Achieving subpixel resolution is one goal of hyperspectral imaging; capturing other subtleties of the scene is another.

Within a single pixel of the object, there may well be a variety of object types, each reflecting or emitting energy differently. The data we now have corresponding to a single pixel are therefore a mixture of the energies associated with each of the subobjects within the pixel. With prior knowledge of the possible types and their reflective or emissive properties, we can separate the mixture to determine which object types are present within the pixel and to what extent. This mixture problem can be solved using the RBI-EMML method.

32.2 A Single Point Source

From an abstract perspective the problem is the following: F and f are a Fourier-transform pair, as are G and g; F and G have finite support. We measure G and want F; g determines some, but not all, of the values of f. We will have, of course, only finitely many measurements of G from which to estimate values of g. Having estimated finitely many values of g, we have the corresponding estimates of f. We apply band-limited extrapolation of these finitely many values of f to estimate F. In fact, once we have estimated values of F, we may not be finished; each value of F is a mixture whose individual components may be what we really want. For this unmixing step we use the RBI-EMML algorithm.

The region of the object that we wish to image is described by the two-dimensional spatial coordinate $\mathbf{x} = (x_1, x_2)$. For simplicity, we take these coordinates to be continuous, leaving until the end the issue of discretization. We shall also denote by $\mathbf{x}$ the point in the image plane corresponding to the point $\mathbf{x}$ on the object; the units of distance between two such points in one plane and their corresponding points in the other plane may, of course, be quite different. For each $\mathbf{x}$ we let $F(\mathbf{x}, \lambda)$ denote the intensity of the component at wavelength λ of the electromagnetic energy that is reflected from or emitted by location $\mathbf{x}$. We shall assume that $F(\mathbf{x}, \lambda) = 0$ for $(\mathbf{x}, \lambda)$ outside some bounded portion of three-dimensional space.

Consider, for a moment, the case in which the energy sensed by the imaging system comes from a single point $\mathbf{x}$. If the dispersion axis of the prism is oriented according to the unit vector $\mathbf{p}_\theta$, for some $\theta \in [0, 2\pi)$, then the component at wavelength λ of the energy from $\mathbf{x}$ on the object is recorded not at $\mathbf{x}$ in the image plane but at the point $\mathbf{x} + \mu(\lambda - \lambda_0)\mathbf{p}_\theta$. Here, $\mu > 0$ is a constant and λ_0 is the wavelength for which the component from point $\mathbf{x}$ of the object is recorded at $\mathbf{x}$ in the image plane.

32.3 Multiple Point Sources

Now imagine energy coming to the imaging system for all the points within the imaged region of the object. Let $G(\mathbf{x}, \theta)$ be the intensity of the energy received at location $\mathbf{x}$ in the image plane when the prism orientation is θ. It follows from the description of the sensing that

$$G(\mathbf{x}, \theta) = \int_{-\infty}^{+\infty} F(\mathbf{x} - \mu(\lambda - \lambda_0)\mathbf{p}_\theta, \lambda) d\lambda. \tag{32.1}$$

The limits of integration are not really infinite due to the finiteness of the aperture and the focal plane of the imaging system. Our data will consist of finitely many values of $G(\mathbf{x}, \theta)$, as $\mathbf{x}$ varies over the grid points of the image plane and θ varies over some finite discretized set of angles.

We begin the image processing by taking the two-dimensional inverse Fourier transform of $G(\mathbf{x}, \theta)$ with respect to the spatial variable $\mathbf{x}$ to get

$$g(\mathbf{y}, \theta) = \frac{1}{(2\pi)^2} \int G(\mathbf{x}, \theta) \exp(-i\mathbf{x} \cdot \mathbf{y}) d\mathbf{x}. \tag{32.2}$$

Inserting the expression for G in Equation (32.1) into Equation (32.2), we obtain

$$g(\mathbf{y}, \theta) = \exp(i\mu\lambda_0 \mathbf{p}_\theta \cdot \mathbf{y}) \int \exp(-i\mu\lambda \mathbf{p}_\theta \cdot \mathbf{y}) f(\mathbf{y}, \lambda) d\lambda, \tag{32.3}$$

where $f(\mathbf{y}, \lambda)$ is the two-dimensional inverse Fourier transform of $F(\mathbf{x}, \lambda)$ with respect to the spatial variable $\mathbf{x}$. Therefore,

$$g(\mathbf{y}, \theta) = \exp(i\mu\lambda_0 \mathbf{p}_\theta \cdot \mathbf{y}) \mathcal{F}(\mathbf{y}, \gamma_\theta), \tag{32.4}$$

where $\mathcal{F}(\mathbf{y}, \gamma)$ denotes the three-dimensional inverse Fourier transform of $F(\mathbf{x}, \lambda)$ and $\gamma_\theta = \mu \mathbf{p}_\theta \cdot \mathbf{y}$. We see then that each value of $g(\mathbf{y}, \theta)$ that we estimate from our measurements provides us with a single estimated value of $\mathcal{F}$.

We use the measured values of $G(\mathbf{x}, \theta)$ to estimate values of $g(\mathbf{y}, \theta)$ guided by the discussion in our earlier chapter on discretization. Having obtained finitely many estimated values of $\mathcal{F}$, we use the support of the function $F(\mathbf{x}, \lambda)$ in three-dimensional space to perform a band-limited extrapolation estimate of the function F.

Alternatively, for each fixed $\mathbf{y}$ for which we have values of $g(\mathbf{y}, \theta)$ we use the PDFT or MDFT to solve Equation (32.3), obtaining an estimate of $f(\mathbf{y}, \lambda)$ as a function of the continuous variable λ. Then, for each fixed λ, we again use the PDFT or MDFT to estimate $F(\mathbf{x}, \lambda)$ from the values of $f(\mathbf{y}, \lambda)$ previously obtained.

32.4 Solving the Mixture Problem

Once we have the estimated function $F(\mathbf{x}, \lambda)$ on a finite grid in three-dimensional space, we can use the RBI-EMML method, as in [102], to solve the mixture problem and identify the individual object types contained within the single pixel denoted $\mathbf{x}$. For each fixed $\mathbf{x}$ corresponding to a pixel, denote by $\mathbf{b} = (b_1, ..., b_I)^T$ the column vector with entries $b_i = F(\mathbf{x}, \lambda_i)$, where λ_i, $i = 1, ..., I$ constitute a discretization of the wavelength space of those λ for which $F(\mathbf{x}, \lambda) > 0$. We assume that this energy intensity distribution vector $\mathbf{b}$ is a superposition of those vectors corresponding to a number of different object types; that is, we assume that

$$\mathbf{b} = \sum_{j=1}^{J} a_j \mathbf{q}_j, \tag{32.5}$$

for some $a_j \geq 0$ and intensity distribution vectors $\mathbf{q}_j$, $j = 1, ..., J$. Each column vector $\mathbf{q}_j$ is a model for what $\mathbf{b}$ would be if there had been only one object type filling the entire pixel. These $\mathbf{q}_j$ are assumed to be known *a priori*. Our objective is to find the a_j.

With Q the I by J matrix whose jth column is $\mathbf{q}_j$ and $\mathbf{a}$ the column vector with entries a_j we write Equation (32.5) as $\mathbf{b} = Q\mathbf{a}$. Since the entries of Q are nonnegative, the entries of $\mathbf{b}$ are positive, and we seek a nonnegative solution a, we can use any of the entropy-based iterative algorithms discussed earlier. Because of its simplicity of form and speed of convergence our preference is the RBI-EMML algorithm. The recent master's thesis of E. Meidunas [102] discusses just such an application.

Chapter 33

Planewave Propagation

In this chapter we demonstrate how the Fourier transform arises naturally as we study the signals received in the farfield from an array of tranmitters or reflectors. We restrict our attention to single-frequency, or narrowband, signals.

33.1 Transmission and Remote-Sensing

For pedagogical reasons, we shall discuss separately what we shall call the transmission and the remote-sensing problems, although the two problems are opposite sides of the same coin, in a sense. In the one-dimensional transmission problem, it is convenient to imagine the transmitters located at points $(x, 0)$ within a bounded interval $[-A, A]$ of the x-axis, and the measurements taken at points P lying on a circle of radius D, centered at the origin. The radius D is large, with respect to A. It may well be the case that no actual sensing is to be performed, but rather, we are simply interested in what the received signal pattern is at points P distant from the transmitters. Such would be the case, for example, if we were analyzing or constructing a transmission pattern of radio broadcasts. In the remote-sensing problem, in contrast, we imagine, in the one-dimensional case, that our sensors occupy a bounded interval of the x-axis, and the transmitters or reflectors are points of a circle whose radius is large, with respect to the size of the bounded interval. The actual size of the radius does not matter and we are interested in determining the amplitudes of the transmitted or reflected signals, as a function of angle only. Such is the case in astronomy, farfield sonar or radar, and the like. Both the transmission and remote-sensing problems illustrate the important role played by the Fourier transform.

33.2 The Transmission Problem

We identify two distinct transmission problems: the direct problem and the inverse problem. In the direct transmission problem, we wish to determine the farfield pattern, given the complex amplitudes of the transmitted signals. In the inverse transmission problem, the array of transmitters or reflectors is the object of interest; we are given, or we measure, the farfield pattern and wish to determine the amplitudes. For simplicity, we consider only single-frequency signals.

We suppose that each point x in the interval $[-A, A]$ transmits the signal $f(x)e^{i\omega t}$, where $f(x)$ is the complex amplitude of the signal and $\omega > 0$ is the common fixed frequency of the signals. Let $D > 0$ be large, with respect to A, and consider the signal received at each point P given in polar coordinates by $P = (D, \theta)$. The distance from $(x, 0)$ to P is approximately $D - x\cos\theta$, so that, at time t, the point P receives from $(x, 0)$ the signal $f(x)e^{i\omega(t-(D-x\cos\theta)/c)}$, where c is the propagation speed. Therefore, the combined signal received at P is

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

The integral term, which gives the farfield pattern of the tranmission, 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 [34].

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

$$F(\gamma) = 2A\frac{\sin(A\gamma)}{A\gamma},$$

for $\gamma \neq 0$, and $F(0) = 2A$. 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.

33.3 Reciprocity

For certain remote-sensing applications, such as sonar and radar array processing and astronomy, it is convenient to switch the roles of sender and receiver. Imagine that superimposed planewave fields are sensed at points within some bounded region of the interior of the sphere, having been transmitted or reflected from the points P on the surface of a sphere whose radius D is large with respect to the bounded region. The *reciprocity principle* tells us that the same mathematical relation holds between points P and $(x, 0)$, regardless of which is the sender and which the receiver. Consequently, the data obtained at the points $(x, 0)$ are then values of the inverse Fourier transform of the function describing the amplitude of the signal sent from each point P.

33.4 Remote Sensing

A basic problem in remote sensing is to determine the nature of a distant object by measuring signals transmitted by or reflected from that object. If the object of interest is sufficiently remote, that is, is in the *farfield*, the data we obtain by sampling the propagating spatio-temporal field is related, approximately, to what we want by *Fourier transformation*. The problem is then to estimate a function from finitely many (usually noisy) values of its *Fourier transform*. The application we consider here is a common one of remote-sensing of transmitted or reflected waves propagating from distant sources. Examples include optical imaging of planets and asteroids using reflected sunlight, radio-astronomy imaging of distant sources of radio waves, active and passive sonar, and radar imaging.

33.5 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,

$$f''(t) + \omega^2 f(t) = 0, \tag{33.1}$$

and

$$\nabla^2 g(x, y, z) + \frac{\omega^2}{c^2} g(x, y, z) = 0. \tag{33.2}$$

Equation (33.2) is the *Helmholtz equation*.

Equation (33.1) 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.

33.6 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 33.1 *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.

33.7 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

$$f(\mathbf{s}) = \int F(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{s}} d\mathbf{k}. \tag{33.3}$$

The data are then inverse Fourier transform values of the complex function $F(\mathbf{k})$; $F(\mathbf{k})$ is defined for all three-dimensional real vectors $\mathbf{k}$, but is zero, in theory, at least, for those $\mathbf{k}$ whose squared length $||\mathbf{k}||^2$ is not equal to ω^2/c^2. Our goal is then to estimate $F(\mathbf{k})$ from measured values of its inverse 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 $\mathbf{k}$.

33.7.1 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 $\mathbf{k}$.

33.8 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.

33.8.1 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 (33.3) 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 inverse 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 inverse 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.

33.8.2 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 (33.3) 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 inverse 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 inverse 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.

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

33.9 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.

33.9.1 The Solar-Emission Problem

In [15] 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. 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.

33.10 Sampling

In the one-dimensional case, the signal received at the point $(x, 0, 0)$ is essentially the inverse Fourier transform $f(x)$ of the function $F(k_1)$; for notational simpicity, 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.

33.11 The Limited-Aperture Problem

In the remote-sensing problem, our measurements at points $(x, 0)$ in the farfield give us the values $f(x)$. Suppose now that we are able to take measurements only for limited values of x, say for $|x| \leq A$; then $2A$ is the *aperture* of our antenna or array of sensors. We describe this by saying that we have available measurements of $f(x)h(x)$, where $h(x) = \chi_A(x) = 1$, for $|x| \leq A$, and zero otherwise. So, in addition to describing blurring and low-pass filtering, the convolution-filter model can also be used to model the limited-aperture problem. As in the low-pass case, the limited-aperture problem can be attacked using extrapolation, but with the same sort of risks described for the low-pass case. A much different approach is to increase the aperture by physically moving the array of sensors, as in *synthetic aperture radar* (SAR).

Returning to the farfield remote-sensing model, if we have inverse 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.

33.12 Resolution

If $F(k) = \delta(k)$ and $h(x) = \chi_A(x)$ describes the aperture-limitation of the imaging system, then the point-spread function is $H(\gamma) = 2A\frac{\sin A\gamma}{\pi\gamma}$. 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

$$R(k) = 2A\frac{\sin A(k-a)}{\pi(k-a)} + \frac{\sin A(k-b)}{\pi(x-b)}.$$

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

$$R(k) = H(k-a) = \frac{\sin A(k-a)}{\pi(k-a)}$$

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.

33.12.1 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 $(x, 0)$, 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.

33.13 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 farfield propagation can be used to explore some of the issues involved in sampling.

Imagine an infinite *uniform line array* of sensors formed by placing receivers at the points $(n\Delta, 0)$, 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 33.2 *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 33.2, 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 [34]).

33.13.1 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.

33.14 The Finite-Data Problem

Suppose that we build a *uniform line array* of sensors by placing receivers at the points $(n\Delta, 0)$, 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.

33.15 Functions of Several Variables

Fourier transformation applies, as well, to functions of several variables. As in the one-dimensional case, we can motivate the multi-dimensional Fourier transform using the farfield propagation model. As we noted earlier, the solar emission problem is inherently a two-dimensional problem.

33.15.1 Two-Dimensional Farfield Object

Assume that our sensors are located at points $\mathbf{s} = (x, y, 0)$ in the x,y-plane. As discussed previously, we assume that the function $F(\mathbf{k})$ can be viewed as a function $F(k_1, k_2)$. Since, in most applications, the distant object has a small angular diameter when viewed from a great distance - the sun's is only 30 minutes of arc - the function $F(k_1, k_2)$ will be supported on a small subset of vectors (k_1, k_2).

33.15.2 Limited Apertures in Two Dimensions

Suppose we have the values of the inverse 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

$$H(\alpha, \beta) = 4AB \frac{\sin A\alpha}{\pi\alpha} \frac{\sin B\beta}{\pi\beta}.$$

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 inverse 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 $H(\alpha, \beta) = \frac{A}{2\pi r} J_1(rA)$, 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.

33.16 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.

33.17 The Laplace Transform and the Ozone Layer

In the farfield 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 fro 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 [119].

33.17.1 The Laplace Transform

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

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

33.17.2 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.

33.17.3 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

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

33.17.4 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.

Part VIII

Appendices

Chapter 34

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.

34.1 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 $\mathcal{X}$.

34.1.1 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 appendix 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 these notes 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 $\langle x, y \rangle$.

34.1.2 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 34.1 *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 34.2 *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.

34.2 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 34.3 *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 $z = P_H x$ of H that is closest to x.

Exercise 34.4 *Show that, for $H = H(a,\gamma)$, $z = P_H x$ is the vector*

$$z = P_H x = x + (\gamma - \langle a, x\rangle)a. \tag{34.1}$$

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 $H(a,0)$.

34.3 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 34.5 *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 34.6 *Show that, if C is closed and $d = 0$, then x is in C.*

Proposition 34.1 *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 $\max(x_j, 0)$.

34.4 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.

34.4.1 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 $m \neq n$.

34.4.2 Systems of Linear Equations

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

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

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:

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

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

$$x_B = -B^{-1}Nx_N. \tag{34.2}$$

34.4.3 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 34.7 *Show that x satisfies the system $Ax = b$ if and only if $\tilde{x}$ satisfies the system $\tilde{A}\tilde{x} = \tilde{b}$.*

Exercise 34.8 *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 34.9 *Show that* B *is Hermitian if and only if the real matrix* $\tilde{B}$ *is symmetric.*

Exercise 34.10 *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 34.11 *Show that* $B^\dagger Bx = c$ *if and only if* $\tilde{B}^T\tilde{B}\tilde{x} = \tilde{c}$*.*

Exercise 34.12 *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 34.13 *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.*

34.4.4 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 34.14 *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 34.15 *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 34.16 *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 34.17 *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 34.18 *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 34.19 *Show that $Q(S) = S$ if and only if $Q(S^\perp) = S^\perp$. Hint: Use Exercise 34.18.*

Exercise 34.20 *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 34.17. Then use Exercise 34.15.*

34.5 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 34.21 *Prove the following identity relating an arbitrary operator T on $\mathcal{X}$ to its complement $G = I - T$:*

$$||x - y||_2^2 - ||Tx - Ty||_2^2 = 2Re(\langle Gx - Gy, x - y\rangle) - ||Gx - Gy||_2^2. \quad (34.3)$$

Exercise 34.22 *Use the previous exercise to prove that*

$$Re(\langle Tx - Ty, x - y\rangle) - ||Tx - Ty||_2^2 = Re(\langle Gx - Gy, x - y\rangle) - ||Gx - Gy||_2^2. \quad (34.4)$$

34.5.1 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 34.23 *Show that, for $H = H(a, \gamma)$, $H_0 = H(a, 0)$, and any x and y in $\mathcal{X}$,*

$$P_H(x + y) = P_H x + P_H y - P_H 0,$$

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 34.24 *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 34.25 *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

$$B = (I - a^I (a^I)^\dagger) \cdots (I - a^1 (a^1)^\dagger).$$

34.5.2 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, $C = H(a, 0)$ for some vector a. We may not be able to describe $P_C x$ explicitly, but we do know a useful property of $P_C x$.

Proposition 34.2 *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. ∎

Corollary 34.1 *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 34.26 *Prove Corollary 34.1. 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 34.2 *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 34.27 *Prove Corollary 34.2. 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 34.3 *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 34.4 *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 34.28 *Prove Corollary 34.4. Hint: the vector $P_S x - x$ is in $S^{\perp}$.*

Corollary 34.5 *Let S be a subspace of $\mathcal{X}$. Then $(S^{\perp})^{\perp} = S$.*

Exercise 34.29 *Prove Corollary 34.5. 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 $u = 0$.*

34.5.3 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*. The *directional derivative* of $f(x)$, at the point x, in the direction of u, written $D_u f(x)$, is

$$D_u f(x) = u_1 \frac{\partial f}{\partial x_1}(x) + ... + u_J \frac{\partial f}{\partial x_J}(x).$$

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 35

Metric Spaces and Norms

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

35.1 Metric Spaces

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

$$d(s, t) \geq 0, \tag{35.1}$$

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

$$d(s, t) = 0 \tag{35.2}$$

if and only if $s = t$;

$$d(s, t) = d(t, s), \tag{35.3}$$

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

$$d(s, t) \leq d(s, u) + d(u, t) \tag{35.4}$$

The last inequality is the *triangle inequality*.

35.2 Analysis in Metric Space

A sequence $\{s^k\}$ in the metric space $(\mathcal{S}, d)$ is said to have limit s^* if

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

Any sequence with a limit is said to be *convergent*.

Exercise 35.1 *Show that a sequence can have at most one limit.*

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

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

Exercise 35.2 *Show that every convergent sequence is a Cauchy sequence.*

The metric space $(\mathcal{S}, d)$ is said to be *complete* if every Cauchy sequence is a convergent sequence. The finite-dimensional Euclidean spaces $R^J$and C^J are complete.

Exercise 35.3 *Let $\mathcal{S}$ be the set of rational numbers, with $d(s,t) = |s-t|$. Show that $(\mathcal{S}, d)$ is a metric space, but not a complete metric space.*

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

Exercise 35.4 *Show that any convergent sequence in a metric space is bounded. Find a bounded sequence of real numbers that is not convergent.*

Exercise 35.5 *Show that, if $\{s^k\}$ is bounded, then, for any element c in the metric space, there is a constant $r > 0$, with $d(c, s^k) \leq r$, for all k.*

A subset K of the metric space is said to be *closed* if, for every convergent sequence $\{s^k\}$ of elements in K, the limit point is again in K. For example, in $\mathcal{X} = R$, the set $K = (0, 1]$ is not closed, because it does not contain the point $s = 0$, which is the limit of the sequence $\{s^k = \frac{1}{k}\}$; the set $K = [0, 1]$ is closed and is the *closure* of the set $(0, 1]$, that is, it is the smallest closed set containing $(0, 1]$.

For any bounded sequence $\{x^k\}$ in $\mathcal{X}$, there is at least one subsequence, often denoted $\{x^{k_n}\}$, that is convergent; the notation implies that the positive integers k_n are ordered, so that $k_1 < k_2 < ....$ The limit of such a subsequence is then said to be a *cluster point* of the original sequence. When we investigate iterative algorithms, we will want to know if the sequence $\{x^k\}$ generated by the algorithm converges. As a first step, we will usually ask if the sequence is bounded? If it is bounded, then it will have at least one cluster point. We then try to discover if that cluster point is really the limit of the sequence.

Exercise 35.6 *Show that your bounded, but not convergent, sequence found in Exercise 35.4 has a cluster point.*

Exercise 35.7 *Show that, if x is a cluster point of the sequence $\{x^k\}$, and if $d(x, x^k) \geq d(x, x^{k+1})$, for all k, then x is the limit of the sequence.*

We turn now to metrics that come from norms.

35.3 Norms

Let $\mathcal{X}$ denote either R^J or C^J. We say that $||x||$ defines a *norm* on $\mathcal{X}$ if

$$||x|| \geq 0, \tag{35.5}$$

for all x,

$$||x|| = 0 \tag{35.6}$$

if and only if $x = 0$,

$$||\gamma x|| = |\gamma| \, ||x||, \tag{35.7}$$

for all x and scalars γ, and

$$||x + y|| \leq ||x|| + ||y||, \tag{35.8}$$

for all vectors x and y.

Exercise 35.8 *Show that $d(x, y) = ||x - y||$ defines a metric on $\mathcal{X}$.*

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

35.3.1 The 1-norm

The 1-norm on $\mathcal{X}$ is defined by

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

Exercise 35.9 *Show that the 1-norm is a norm.*

35.3.2 The ∞-norm

The ∞-norm on $\mathcal{X}$ is defined by

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

Exercise 35.10 *Show that the ∞-norm is a norm.*

35.3.3 The 2-norm

The 2-norm, also called the Euclidean norm, is the most commonly used norm on $\mathcal{X}$. It is the one that comes from the inner product:

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

Exercise 35.11 *Show that the 2-norm is a norm. Hint: for the triangle inequality, use the Cauchy Inequality.*

It is this close relationship between the 2-norm and the inner product that makes the 2-norm so useful.

35.3.4 Weighted 2-norms

Let Q be a positive-definite Hermitian matrix. Define

$$||x||_Q = \sqrt{x^\dagger Q x},$$

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

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

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

Exercise 35.12 *Show that the Q-weighted 2-norm is a norm.*

35.4 Eigenvalues and Eigenvectors

Let S be a complex, square matrix. We say that λ is an eigenvalue of S if λ is a root of the complex polynomial $\det(\lambda I - S)$. Therefore, each S has as many (possibly complex) eigenvalues as it has rows or columns, although some of the eigenvalues may be repeated.

An equivalent definition is that λ is an eigenvalue of S if there is a non-zero vector x with $Sx = \lambda x$, in which case the vector x is called an *eigenvector* of S. From this definition, we see that the matrix S is invertible if and only if zero is not one of its eigenvalues. The *spectral radius* of S, denoted $\rho(S)$, is the maximum of $|\lambda|$, over all eigenvalues λ of S.

Exercise 35.13 *Show that $\rho(S^2) = \rho(S)^2$.*

Exercise 35.14 *We say that S is Hermitian or self-adjoint if $S^\dagger = S$. Show that, if S is Hermitian, then every eigenvalue of S is real. Hint: suppose that $Sx = \lambda x$. Then consider $x^\dagger Sx$.*

If S is an I by I Hermitian matrix with (necessarily real) eigenvalues

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_I,$$

and associated (column) eigenvectors $\{u_i \,|i = 1, ..., I\}$ (which we may assume are mutually orthogonal), then S can be written as

$$S = \lambda_1 u_1 u_1^\dagger + \cdots + \lambda_I u_I u_I^\dagger.$$

This is the *eigenvalue/eigenvector decomposition* of S. The Hermitian matrix S is invertible if and only if all of its eigenvalues are non-zero, in which case we can write the inverse of S as

$$S^{-1} = \lambda_1^{-1} u_1 u_1^\dagger + \cdots + \lambda_I^{-1} u_I u_I^\dagger.$$

A Hermitian matrix S is *positive-definite* if each of its eigenvalues is positive. It follows from the eigenvector decomposition of S that $S = QQ^\dagger$ for the Hermitian, positive-definite matrix

$$Q = \sqrt{\lambda_1}u_1u_1^\dagger + \cdots + \sqrt{\lambda_I}u_Iu_I^\dagger;$$

Q is called the *Hermitian square root* of S.

35.4.1 The Singular-Value Decomposition

Let A be an I by J complex matrix, with $I \leq J$. Let $B = AA^\dagger$ and $C = A^\dagger A$. Let $\lambda_i \geq 0$, for $i = 1, ..., I$, be the eigenvalues of B, and let $\{u^1, ..., u^I\}$ be associated orthonormal eigenvectors of B. Assume that $\lambda_i > 0$ for $i = 1, ..., N \leq I$, and, if $N < I$, $\lambda_i = 0$, for $i = N+1, ..., I$; if $N = I$, then the matrix A is said to have *full rank*. For $i = 1, ..., N$, let $v^i = \lambda_i^{-1/2}A^\dagger u^i$.

Exercise 35.15 *Show that the collection* $\{v^1, ..., v^N\}$ *is orthonormal.*

Let $\{v^{N+1}, ..., v^J\}$ be selected so that $\{v^1, ..., v^J\}$ is orthonormal.

Exercise 35.16 *Show that the sets* $\{u^1, ..., u^N\}$, $\{u^{N+1}, ..., u^I\}$, $\{v^1, ..., v^N\}$, *and* $\{v^{N+1}, ..., v^J\}$ *are orthonormal bases for the subspaces* $CS(A)$, $NS(A^\dagger)$, $CS(A^\dagger)$, *and* $NS(A)$, *respectively.*

Exercise 35.17 *Show that*

$$A = \sum_{i=1}^{N} \sqrt{\lambda_i}u^i(v^i)^\dagger,$$

which is the singular-value decomposition (SVD) of the matrix A.

The SVD of the matrix $A^\dagger$ is then

$$A^\dagger = \sum_{i=1}^{N} \sqrt{\lambda_i}v^i(u^i)^\dagger.$$

Exercise 35.18 *Use the SVD of* A *to obtain the eigenvalue/eigenvector decompositions of* B *and* C*:*

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

and

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

Exercise 35.19 *The pseudo-inverse of the matrix A is the J by I matrix*

$$A^\sharp = \sum_{i=1}^{N} \lambda_i^{-1/2} v^i (u^i)^\dagger.$$

Show that

$$(A^\dagger)^\sharp = (A^\sharp)^\dagger.$$

Show that, if $N = I \leq J$, then

$$A^\sharp = A^\dagger B^{-1},$$

and

$$(A^\dagger)^\sharp = B^{-1} A.$$

Investigate other properties of the pseudo-inverse.

35.5 Matrix Norms

Any matrix can be turned into a vector by vectorization. Therefore, we can define a norm for any matrix by simply vectorizing and taking a norm of the resulting vector. Such norms for matrices may not take full advantage of the matrix properties. An *induced matrix norm* or just a *matrix norm* for matrices is a special type of norm that comes from a vector norm and that respects the matrix properties. If A is a matrix and $||A||$ denotes a matrix norm of A, then we insist that $||Ax|| \leq ||A||||x||$, for all x. All induced matrix norms have this *compatibility property.*

35.5.1 Induced Matrix Norms

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

$$|||Ax||| \leq c||x||,$$

for all x in C^J. It can be written as

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

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

$$|||Ax - Az||| \leq ||A||||x - z||,$$

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

35.5.2 Condition Number of a Square Matrix

Let S be a square, invertible matrix and z the solution to $Sz = h$. We are concerned with the extent to which the solution changes as the right side, h, changes. Denote by δ_h a small perturbation of h, and by δ_z the solution of $S\delta_z = \delta_h$. Then $S(z+\delta_z) = h+\delta_h$. Applying the compatiability condition $||Ax|| \leq ||A||||x||$, we get

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

and

$$||z|| \geq ||h||/||S||.$$

Therefore

$$\frac{||\delta_z||}{||z||} \leq ||S||||S^{-1}||\frac{||\delta_h||}{||h||}. \tag{35.9}$$

The quantity $c = ||S||||S^{-1}||$ is the *condition number* of S, with respect to the given matrix norm. Note that $c \geq 1$: for any non-zero z, we have

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

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

$$c = \lambda_{max}/\lambda_{min},$$

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

If we choose the two vector norms carefully, then we can get an explicit description of $||A||$, but, in general, we cannot.

For example, let $||x|| = ||x||_1$ and $|||Ax||| = ||Ax||_1$ be the 1-norms of the vectors x and Ax, where

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

Exercise 35.20 *Show that the* 1*-norm of* A*, induced by the* 1*-norms of vectors in* C^J *and* C^I*, is*

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

Hints: use basic properties of the absolute value to show that

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

Then let $j = m$ *be the index for which the maximum column sum is reached and select* $x_j = 0$*, for* $j \neq m$*, and* $x_m = 1$*.*

The *infinity norm* of the vector x is

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

Exercise 35.21 *Show that the infinity norm of the matrix A, induced by the infinity norms of vectors in C^J and C^I, is*

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

Exercise 35.22 *Let M be an invertible matrix and $||x||$ any vector norm. Define*

$$||x||_M = ||Mx||.$$

Show that, for any square matrix S, the matrix norm

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

is

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

In [4] this result is used to prove the following lemma:

Lemma 35.1 *Let S be any square matrix and let $\epsilon > 0$ be given. Then there is an invertible matrix M such that*

$$||S||_M \leq \rho(S) + \epsilon.$$

Exercise 35.23 *Show that, for any square matrix S and any induced matrix norm $||S||$, we have $||S|| \geq \rho(S)$. Consequently, for any induced matrix norm $||S||$,*

$$||S|| \geq |\lambda|,$$

for every eigenvalue λ of S.

So we know that

$$\rho(S) \leq ||S||,$$

for every induced matrix norm, but, according to Lemma 35.1, we also have

$$||S||_M \leq \rho(S) + \epsilon.$$

Exercise 35.24 *Show that, if $\rho(S) < 1$, then there is a vector norm on $\mathcal{X}$ for which the induced matrix norm of S is less than one, so that S is a strict contraction with respect to this vector norm.*

35.6 The Euclidean Norm of a Square Matrix

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

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

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

with the maximum over all nonzero vectors x. Since

$$||Ax||_2^2 = x^\dagger A^\dagger Ax,$$

we have

$$||A||_2 = \sqrt{\max\{\frac{x^\dagger A^\dagger Ax}{x^\dagger x}\}}, \tag{35.10}$$

over all nonzero vectors x.

Exercise 35.25 *Show that*

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

that is, the term inside the square-root in Equation (35.10) is the largest eigenvalue of the matrix $A^\dagger A$. Hints: let

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_J \geq 0$$

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

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

while

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

It follows that

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

and

$$||Ax||_2^2 = x^\dagger A^\dagger Ax = \sum_{j=1}^{J}\lambda_j|(u^j)^\dagger x|^2. \tag{35.11}$$

Maximizing $||Ax||_2^2/||x||_2^2$ over $x \neq 0$ is equivalent to maximizing $||Ax||_2^2$, subject to $||x||_2^2 = 1$. The right side of Equation (35.11) is then a convex combination of the λ_j, which will have its maximum when only the coefficient of λ_1 is non-zero.

Exercise 35.26 *Show that, if S is Hermitian, then $||S||_2 = \rho(S)$. Hint: use Exercise (35.13).*

If S is not Hermitian, then the Euclidean norm of S cannot be calculated directly from the eigenvalues of S.

Exercise 35.27 *Let S be the square, non-Hermitian matrix*

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

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

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

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

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

35.6.1 Diagonalizable Matrices

A square matrix S is *diagonalizable* if $\mathcal{X}$ has a basis of eigenvectors of S. In that case, with V be a square matrix whose columns are linearly independent eigenvectors of S and L the diagonal matrix having the eigenvalues of S along its main diagonal, we have $SV = VL$, or $V^{-1}SV = L$.

Exercise 35.28 *Let $T = V^{-1}$ and define $||x||_T = ||Tx||_2$, the Euclidean norm of Tx. Show that the induced matrix norm of S is $||S||_T = \rho(S)$.*

We see from this exercise that, for any diagonalizable matrix S, in particular, for any Hermitian matrix, there is a vector norm such that the induced matrix norm of S is $\rho(S)$. In the Hermitian case we know that, if the eigenvector columns of V are scaled to have length one, then $V^{-1} = V^\dagger$ and $||Tx||_2 = ||V^\dagger x||_2 = ||x||_2$, so that the required vector norm is just the Euclidean norm, and $||S||_T$ is just $||S||_2$, which we know to be $\rho(S)$.

35.6.2 Gerschgorin's Theorem

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

Theorem 35.1 *Let A be J by J. For $j = 1, ..., J$, let C_j be the circle in the complex plane with center A_{jj} and radius $r_j = \sum_{m \neq j} |A_{jm}|$. Then every eigenvalue of A lies within one of the C_j.*

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

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

so that

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

This completes the proof. ∎

35.6.3 Strictly Diagonally Dominant Matrices

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

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

When the matrix S is strictly diagonally dominant, all the eigenvalues of S lie within the union of the spheres with centers S_{ii} and radii S_{ii}. With D the diagonal component of S, the matrix $D^{-1}S$ then has all its eigenvalues within the circle of radius one, centered at $(1, 0)$. Then $\rho(I - D^{-1}S) < 1$. We use this result in our discussion of the Jacobi splitting method.

Chapter 36

The Fourier Transform

In this chapter we review the basic properties of the Fourier transform.

36.1 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

$$F(\gamma) = \int_{-\infty}^{\infty} f(x) e^{i\gamma x} dx. \tag{36.1}$$

36.1.1 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*:

$$f(x) = \frac{1}{2\pi} \int F(\gamma) e^{-i\gamma x} d\gamma. \tag{36.2}$$

The functions $f(x)$ and $F(\gamma)$ are called a *Fourier-transform pair*.

36.1.2 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.

36.2 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

$$\delta(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\omega x}d\omega. \tag{36.3}$$

Obviously, this integral cannot be understood in the usual way. The integral in Equation (36.3) is a symbolic way of saying that

$$\int h(x)(\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\omega x}d\omega)dx = \int h(x)\delta(x)dx = h(0), \tag{36.4}$$

for all $h(x)$ that are continuous at $x = 0$; that is, the integral in Equation (36.3) has the sifting property, so it acts like $\delta(x)$. Interchanging the order of integration in Equation (36.4), we obtain

$$\int h(x)(\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\omega x}d\omega)dx = \frac{1}{2\pi}\int_{-\infty}^{\infty}(\int h(x)e^{-i\omega x}dx)d\omega$$

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} H(-\omega)d\omega = \frac{1}{2\pi}\int_{-\infty}^{\infty} H(\omega)d\omega = h(0).$$

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 [33]).

36.3 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.

36.3.1 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

$$g(x) = \frac{1}{2\pi} \int F(\gamma)H(\gamma)e^{-i\gamma x} dx. \tag{36.5}$$

The function $g(x)$ that results is $g(x) = (f * h)(x)$, the *convolution* of the functions $f(x)$ and $h(x)$, with the latter given by

$$h(x) = \frac{1}{2\pi} \int H(\gamma)e^{-i\gamma x} dx.$$

Note that, if $f(x) = \delta(x)$, then $g(x) = h(x)$; that is, our reconstruction of the object from distorted data is the function $h(x)$ itself. For that reason, the function $h(x)$ is called the *point-spread function* of the imaging system.

Convolution filtering refers to the process of converting any given function, say $f(x)$, into a different function, say $g(x)$, by convolving $f(x)$ with a fixed function $h(x)$. Since this process can be achieved by multiplying $F(\gamma)$ by $H(\gamma)$ and then inverse Fourier transforming, such convolution filters are studied in terms of the properties of the function $H(\gamma)$, known in this context as the *system transfer function*, or the *optical transfer function* (OTF); when γ is a frequency, rather than a spatial frequency, $H(\gamma)$ is called the *frequency-response function* of the filter. The 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 (36.5), replacing $F(\gamma)$ and $H(\gamma)$ with their definitions, as given by Equation (36.1). 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 (36.3) 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

$$g(x) = \int f(t)h(x-t)dt; \tag{36.6}$$

this is the definition of the convolution of the functions f and h.

36.3.2 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.

36.4 Two-Dimensional Fourier Transforms

More generally, we consider a function $f(x, z)$ of two real variables. Its Fourier transformation is

$$F(\alpha, \beta) = \int \int f(x, z) e^{i(x\alpha + z\beta)} dx dz. \tag{36.7}$$

For example, suppose that $f(x, z) = 1$ for $\sqrt{x^2 + z^2} \leq R$, and zero, otherwise. Then we have

$$F(\alpha, \beta) = \int_{-\pi}^{\pi} \int_{0}^{R} e^{-i(\alpha r \cos\theta + \beta r \sin\theta)} r dr d\theta.$$

In polar coordinates, with $\alpha = \rho \cos\phi$ and $\beta = \rho \sin\phi$, we have

$$F(\rho, \phi) = \int_{0}^{R} \int_{-\pi}^{\pi} e^{ir\rho\cos(\theta - \phi)} d\theta r dr.$$

The inner integral is well known;

$$\int_{-\pi}^{\pi} e^{ir\rho\cos(\theta - \phi)} d\theta = 2\pi J_0(r\rho),$$

where J_0 denotes the 0th order Bessel function. Using the identity

$$\int_{0}^{z} t^n J_{n-1}(t) dt = z^n J_n(z),$$

we have

$$F(\rho, \phi) = \frac{2\pi R}{\rho} J_1(\rho R).$$

Notice that, since $f(x, z)$ 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.

36.4.1 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

$$f(x, y) = \frac{1}{4\pi^2} \int \int F(\alpha, \beta) e^{-i(\alpha x + \beta y)} d\alpha d\beta. \tag{36.8}$$

It is important to note that this procedure can be viewed as two one-dimensional Fourier inversions: first, we invert $F(\alpha, \beta)$, as a function of, say, β only, to get the function of α and y

$$g(\alpha, y) = \frac{1}{2\pi} \int F(\alpha, \beta) e^{-i\beta y} d\beta;$$

second, we invert $g(\alpha, y)$, as a function of α, to get

$$f(x, y) = \frac{1}{2\pi} \int g(\alpha, y) e^{-i\alpha x} d\alpha.$$

If we write the functions $f(x, y)$ and $F(\alpha, \beta)$ in polar coordinates, we obtain alternative ways to implement the two-dimensional Fourier inversion. We shall consider these other ways when we discuss the tomography problem of reconstructing a function $f(x, y)$ from line-integral data.

Chapter 37

Bregman-Legendre Functions

In [9] Bauschke and Borwein show convincingly that the Bregman-Legendre functions provide the proper context for the discussion of Bregman projections onto closed convex sets. The summary here follows closely the discussion given in [9].

37.1 Essential smoothness and essential strict convexity

A convex function $f : R^J \to [-\infty, +\infty]$ is *proper* if there is no x with $f(x) = -\infty$ and some x with $f(x) < +\infty$. The *essential domain* of f is $D = \{x | f(x) < +\infty\}$. A proper convex function f is *closed* if it is lower semi-continuous. The *subdifferential* of f at x is the set $\partial f(x) = \{x^* | \langle x^*, z - x\rangle \leq f(z) - f(x), \text{for all } z\}$. The domain of ∂f is the set dom $\partial f = \{x | \partial f(x) \neq \emptyset\}$. The *conjugate function* associated with f is the function $f^*(x^*) = \sup_z(\langle x^*, z\rangle - f(z))$.

Following [111] we say that a closed proper convex function f is *essentially smooth* if int$D$ is not empty, f is differentiable on int$D$ and $x^n \in$ int$D$, with $x^n \to x \in$ bd$D$, implies that $||\nabla f(x^n)|| \to +\infty$. Here int$D$ and bd$D$ denote the interior and boundary of the set D.

A closed proper convex function f is *essentially strictly convex* if f is strictly convex on every convex subset of dom ∂f.

The closed proper convex function f is essentially smooth if and only if the subdifferential $\partial f(x)$ is empty for $x \in$ bd$D$ and is $\{\nabla f(x)\}$ for $x \in$ int$D$ (so f is differentiable on int$D$) if and only if the function f^* is essentially strictly convex.

A closed proper convex function f is said to be a *Legendre function* if it is both essentially smooth and essentialy strictly convex. So f is Legendre if and only if its conjugate function is Legendre, in which case the gradient operator ∇f is a topological isomorphism with ∇f^* as its inverse. The gradient operator ∇f maps int dom f onto int dom f^*. If int dom $f^* = R^J$ then the range of ∇f is R^J and the equation $\nabla f(x) = y$ can be solved for every $y \in R^J$. In order for int dom $f^* = R^J$ it is necessary and sufficient that the Legendre function f be *super-coercive*, that is,

$$\lim_{||x||\to+\infty} \frac{f(x)}{||x||} = +\infty.$$

If the essential domain of f is bounded, then f is super-coercive and its gradient operator is a mapping onto the space R^J.

37.2 Bregman Projections onto Closed Convex Sets

Let f be a closed proper convex function that is differentiable on the nonempty set int$D$. The corresponding *Bregman distance* $D_f(x, z)$ is defined for $x \in R^J$ and $z \in$ int$D$ by

$$D_f(x, z) = f(x) - f(z) - \langle \nabla f(z), x - z \rangle.$$

Note that $D_f(x, z) \geq 0$ always and that $D_f(x, z) = +\infty$ is possible. If f is essentially strictly convex then $D_f(x, z) = 0$ implies that $x = z$.

Let K be a nonempty closed convex set with $K \cap$ int$D \neq \emptyset$. Pick $z \in$ int$D$. The *Bregman projection* of z onto K, with respect to f, is

$$P_K^f(z) = \text{argmin}_{x \in K \cap D} D_f(x, z).$$

If f is essentially strictly convex, then $P_K^f(z)$ exists. If f is strictly convex on D then $P_K^f(z)$ is unique. If f is Legendre, then $P_K^f(z)$ is uniquely defined and is in int$D$; this last condition is sometimes called *zone consistency*.

Example: Let $J = 2$ and $f(x)$ be the function that is equal to one-half the norm squared on D, the nonnegative quadrant, $+\infty$ elsewhere. Let K be the set $K = \{(x_1, x_2) | x_1 + x_2 = 1\}$. The Bregman projection of $(2, 1)$ onto K is $(1, 0)$, which is not in int$D$. The function f is not essentially smooth, although it is essentially strictly convex. Its conjugate is the function f^* that is equal to one-half the norm squared on D and equal to zero elsewhere; it is essentially smooth, but not essentially strictly convex.

If f is Legendre, then $P_K^f(z)$ is the unique member of $K \cap$ int$D$ satisfying the inequality

$$\langle \nabla f(P_K^f(z)) - \nabla f(z), P_K^f(z) - c \rangle \geq 0,$$

for all $c \in K$. From this we obtain the *Bregman Inequality*:

$$D_f(c, z) \geq D_f(c, P_K^f(z)) + D_f(P_K^f(z), z), \tag{37.1}$$

for all $c \in K$.

37.3 Bregman-Legendre Functions

Following Bauschke and Borwein [9], we say that a Legendre function f is a *Bregman-Legendre* function if the following properties hold:

B1: for x in D and any $a > 0$ the set $\{z | D_f(x, z) \leq a\}$ is bounded.
B2: if x is in D but not in int$D$, for each positive integer n, y^n is in int$D$ with $y^n \to y \in \text{bd}D$ and if $\{D_f(x, y^n)\}$ remains bounded, then $D_f(y, y^n) \to 0$, so that $y \in D$.
B3: if x^n and y^n are in int$D$, with $x^n \to x$ and $y^n \to y$, where x and y are in D but not in int$D$, and if $D_f(x^n, y^n) \to 0$ then $x = y$.

Bauschke and Borwein then prove that Bregman's SGP method converges to a member of K provided that one of the following holds: 1) f is Bregman-Legendre; 2) $K \cap \text{int}D \neq \emptyset$ and dom f^* is open; or 3) dom f and dom f^* are both open.

37.4 Useful Results about Bregman-Legendre Functions

The following results are proved in somewhat more generality in [9].
R1: If $y^n \in$ int dom f and $y^n \to y \in$ int dom f, then $D_f(y, y^n) \to 0$.
R2: If x and $y^n \in$ int dom f and $y^n \to y \in$ bd dom f, then $D_f(x, y^n) \to +\infty$.
R3: If $x^n \in D$, $x^n \to x \in D$, $y^n \in$ int D, $y^n \to y \in D$, $\{x, y\} \cap$ int $D \neq \emptyset$ and $D_f(x^n, y^n) \to 0$, then $x = y$ and $y \in$ int D.
R4: If x and y are in D, but are not in int D, $y^n \in$ int D, $y^n \to y$ and $D_f(x, y^n) \to 0$, then $x = y$.
As a consequence of these results we have the following.
R5: If $\{D_f(x, y^n)\} \to 0$, for $y^n \in$ int D and $x \in R^J$, then $\{y^n\} \to x$.

Proof of R5: Since $\{D_f(x, y^n)\}$ is eventually finite, we have $x \in D$. By Property B1 above it follows that the sequence $\{y^n\}$ is bounded; without loss of generality, we assume that $\{y^n\} \to y$, for some $y \in \overline{D}$. If x is in int D, then, by result R2 above, we know that y is also in int D. Applying result R3, with $x^n = x$, for all n, we conclude that $x = y$. If, on the other hand, x is in D, but not in int D, then y is in D, by result R2. There are

two cases to consider: 1) y is in int D; 2) y is not in int D. In case 1) we have $D_f(x, y^n) \to D_f(x, y) = 0$, from which it follows that $x = y$. In case 2) we apply result R4 to conclude that $x = y$. ∎

Chapter 38

The EM Algorithm

The so-called *EM algorithm* [58, 101] is a general framework for deriving iterative methods for maximum-likelihood parameter estimation. There is a problem with the way the EM algorithm is usually described in the literature. That description is fine for the case of discrete random vectors, but needs to be modified to apply to continuous ones. We begin with the usual formulation of the EM algorithm, as it applies to the discrete case.

38.1 The Discrete Case

We denote by Z a random vector, taking values in R^N, by $h : R^N \to R^I$ a function from R^N to R^I, with $N > I$, and $Y = h(Z)$ the corresponding random vector taking values in R^I. The random vector Z has probability function $f(z; x)$, where x is a parameter in the parameter space $\mathcal{X}$. The probability function associated with Y is then

$$g(y; x) = \sum_{z \in h^{-1}(y)} f(z; x) \leq 1. \tag{38.1}$$

The random vector Y is usually called the *incomplete data*, and Z the *complete data*. The EM algorithm is typically used when maximizing $f(z; x)$ is easier than maximizing $g(y; x)$, but we have only y, an instance of Y, and not a value of Z.

The conditional probability function for Z, given $Y = y$ and x, is

$$b(z; y, x) = f(z; x)/g(y; x), \tag{38.2}$$

for $z \in h^{-1}(y)$, and $b(z; y, x) = 0$, otherwise. The *E-step* of the EM algorithm is to calculate the conditional expected value of the random variable

$\log f(Z;x)$, given y and the current estimate x_k of x:

$$Q(x;x_k) = E(\log f(Z;x)|y,x_k) = \sum_{z\in h^{-1}(y)} b(z;y,x_k)\log f(z;x). \quad (38.3)$$

The *M-step* is to select x_{k+1} as a maximizer of $Q(x;x_k)$. Denote by $H(x;x_k)$ the conditional expected value of the random variable $\log b(Z;y,x)$, given y and x_k:

$$H(x;x_k) = \sum_{z\in h^{-1}(y)} b(z;y,x_k)\log b(z;y,x). \quad (38.4)$$

Then, for all $x \in \mathcal{X}$, we have

$$Q(x;x_k) = H(x;x_k) + L(x), \quad (38.5)$$

for $L(x) = \log g(y;x)$.

For positive scalars a and b, let $KL(a,b)$ denote the Kullback-Leibler distance

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

Also let $KL(a,0) = +\infty$ and $KL(0,b) = b$. Extend the KL distance component-wise to vectors with non-negative entries. It follows from the inequality $\log t \le t-1$ that $KL(a,b) \ge 0$ and $KL(a,b) = 0$ if and only if $a = b$. Then we have

$$Q(x;x_k) = -KL(b(\cdot;y,x_k), f(\cdot;x)), \quad (38.6)$$

and

$$H(x_k;x_k) = H(x;x_k) + KL(b(\cdot;y,x_k), b(\cdot;y,x)), \quad (38.7)$$

where

$$KL(b(\cdot;y,x_k), b(\cdot;y,x)) = \sum_z KL(b(z;y,x_k), b(z;y,x)) \ge 0.$$

Therefore,

$$\begin{aligned} L(x_k) &= Q(x_k;x_k) - H(x_k;x_k) \le Q(x_{k+1};x_k) - H(x_k;x_k) \\ &= Q(x_{k+1};x_k) - H(x_{k+1};x_k) - KL(b(x_k), b(x_{k+1})) \\ &= L(x_{k+1}) - KL(b(x_k), b(x_{k+1})). \end{aligned}$$

The sequence $\{L(x_k)\}$ is increasing and non-positive, so convergent. The sequence $\{KL(b(x_k), b(x_{k+1}))\}$ converges to zero.

In the discrete case, the EM algorithm is an *alternating minimization* method. The function $KL(b(\cdot;y,x_k), f(\cdot;x))$ is minimized by the choice

$x = x_{k+1}$, and the function $KL(b(\cdot; y, x), f(\cdot; x_{k+1}))$ is minimized by the choice $x = x_{k+1}$. Therefore, the EM algorithm can be viewed as the result of alternately minimizing $KL(b(\cdot; y, u), f(\cdot; v))$, first with respect to the variable u, and then with respect to the variable v.

Without further assumptions, we can say no more; see [122]. We would like to conclude that the sequence $\{x_k\}$ converges to a maximizer of $L(x)$, but we have no metric on the parameter space $\mathcal{X}$. We need an identity that relates the nonnegative quantity

$$KL(b(\cdot; y, x_k), f(\cdot; x)) - KL(b(\cdot; y, x_k), f(\cdot; x_{k+1}))$$

to the difference, in parameter space, between x and x_{k+1}. For example, for the EMML algorithm in the Poisson mixture case, we have

$$KL(b(\cdot; y, x_k), f(\cdot; x)) - KL(b(\cdot; y, x_k), f(\cdot; x_{k+1})) = KL(x_{k+1}, x).$$

38.2 The continuous case

The usual approach to the EM algorithm in this case is to mimic the discrete case. A problem arises when we try to define $g(y; x)$ as

$$g(y; x) = \int_{z \in h^{-1}(y)} f(z; x) dz;$$

the set $h^{-1}(y)$ typically has measure zero in R^N. We need a different approach.

Suppose that there is a second function $c : R^N \to R^{N-I}$ such that the function $G(z) = G(h(z), c(z)) = (y, w)$ has inverse $H(y, w) = z$. Then, given y, let $W(y) = \{w = c(z) | y = h(z)\}$. Then, with $J(y, w)$ the Jacobian, the pdf of the random vector Y is

$$g(y; x) = \int_{W(y)} f(H(y, w); x) J(y, w) dw,$$

and the pdf for the random vector $W = c(Z)$ is

$$b(H(y, w); y, x) = f(H(y, w); x) J(y, w)/g(y; x),$$

for $w \in W(y)$. Given y, and having found x_k, we minimize

$$KL(b(H(y, w); x_k), f(H(y, w); x)),$$

with respect to x, to get x_{k+1}.

38.2.1 An Example

Suppose that Z_1 and Z_2 are independent and uniformly distributed on the interval $[0, x]$, where $x > 0$ is an unknown parameter. Let $Y = Z_1 + Z_2$. Then

$$g(y; x) = y/x^2,$$

for $0 \leq y \leq x$, and

$$g(y; x) = (2x - y)/x^2,$$

for $x \leq y \leq 2x$. Given y, the maximum likelihood estimate of x is y. The pdf for the random vector $Z = (Z_1, Z_2)$ is

$$f(z_1, z_2; x) = \frac{1}{x^2}\chi_{[0,x]}(z_1)\chi_{[0,x]}(z_2).$$

The conditional pdf of Z, given y and x_k, is

$$b(z_1, z_2; y, x_k) = \frac{1}{y}\chi_{[0,x_k]}(z_1)\chi_{[0,x_k]}(z_2),$$

for $0 \leq y \leq x_k$, and for $x_k \leq y \leq 2x_k$ it is

$$b(z_1, z_2; y, x_k) = \frac{1}{2x_k - y}\chi_{[0,x_k]}(z_1)\chi_{[0,x_k]}(z_2).$$

Suppose that $c(z) = c(z_1, z_2) = z_2$ and $W = c(Z)$. Then $W(y) = [0, y]$ and the conditional pdf of W, given y and x_k is $b(y - w, w; y, x_k)$. If we choose $x_0 \geq y$, then $x_1 = y$, which is the ML estimator. But, if we choose x_0 in the interval $[\frac{y}{2}, y]$, then $x_1 = x_0$ and the EM iteration stagnates. Note that the function $L(x) = \log g(y; x)$ is continuous, but not differentiable. It is concave for x in the interval $[\frac{y}{2}, y]$ and convex for $x \geq y$.

Chapter 39

Using Prior Knowledge in Remote Sensing

The problem is to reconstruct a (possibly complex-valued) function $f : R^D \rightarrow C$ from finitely many measurements g_n, $n = 1, ..., N$, pertaining to f. The function $f(r)$ represents the physical object of interest, such as the spatial distribution of acoustic energy in sonar, the distribution of x-ray-attenuating material in transmission tomography, the distribution of radionuclide in emission tomography, the sources of reflected radio waves in radar, and so on. Often the reconstruction, or estimate, of the function f takes the form of an image in two or three dimensions; for that reason, we also speak of the problem as one of *image reconstruction*. The data are obtained through measurements. Because there are only finitely many measurements, the problem is highly underdetermined and even noise-free data are insufficient to specify a unique solution.

39.1 The Optimization Approach

One way to solve such underdetermined 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 alternatve 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 solutons 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.

39.2 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 reconstructon. The inner product induces a norm and our reconstruction is the 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

$$g_n = \int_S f(r)\overline{h_n(r)}dr, \tag{39.1}$$

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 (39.1) in the context of a Hilbert space, and to write

$$g_n = \langle f, h_n \rangle, \tag{39.2}$$

where the usual Hilbert space inner product is defined by

$$\langle f, h \rangle_2 = \int_S f(r)\overline{h(r)}dr, \tag{39.3}$$

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

$$||f||_2 = \sqrt{\int_S |f(r)|^2 dr} \tag{39.4}$$

is finite.

39.2.1 Minimum-Norm Solutions

Our estimation problem is highly underdetermined; there are infinitely many functions in $L^2(S)$ that are consistent with the data and might be the right answer. Such underdetermined 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 wth 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

$$\mathcal{H}^\dagger a = a_1 h_1(r) + ... + a_N h_N(r). \tag{39.5}$$

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

$$g = \mathcal{H}\mathcal{H}^\dagger a \tag{39.6}$$

for the vector a. Then the function

$$\hat{f}(r) = \mathcal{H}^\dagger a \tag{39.7}$$

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 39.1 *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 (39.4). 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.

39.3 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

$$\langle f, h\rangle_{\mathcal{T}} = \langle \mathcal{T}^{-1} f, \mathcal{T}^{-1} h\rangle_2. \tag{39.8}$$

We can then use this inner product to define the problem to be solved. We now say that

$$g_n = \langle f, t^n\rangle_{\mathcal{T}}, \tag{39.9}$$

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 39.2 *Show that* $\mathcal{T}^*\mathcal{T} = \mathcal{T}\mathcal{T}^\dagger$.

Consequently, we have

$$g_n = \langle f, \mathcal{T}\mathcal{T}^\dagger h^n\rangle_{\mathcal{T}}. \tag{39.10}$$

39.4 Minimum-$\mathcal{T}$-Norm Solutions

The function $\tilde{f}$ consistent with the data and having the smallest $\mathcal{T}$-norm has the algebraic form

$$\hat{f} = \sum_{m=1}^{N} a_m \mathcal{T}\mathcal{T}^\dagger h^m. \tag{39.11}$$

Applying the $\mathcal{T}$-inner product to both sides of Equation (39.11), 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,

$$g_n = \sum_{m=1}^{N} a_m \langle \mathcal{T}^\dagger h^m, \mathcal{T}^\dagger h^n\rangle_2. \tag{39.12}$$

We solve this system for the a_m and insert them into Equation (39.11) to get our reconstruction. The Gram matrix that appears in Equation (39.12) is positive-definite, but is often ill-conditioned; increasing the main diagonal by a percent or so usually is sufficient regularization.

39.5 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.

39.5.1 The $L^2(-\pi, \pi)$ Case

Assume that $f(x) = 0$, for $|x| > \pi$. The minimum-2-norm solution has the form

$$\hat{f}(x) = \sum_{m=1}^{N} a_m e^{i\omega_m x}, \tag{39.13}$$

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

$$\hat{f}(x) = \sum_{n=1}^{N} g_n e^{inx}. \tag{39.14}$$

39.5.2 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.

39.5.3 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

$$\hat{f}(x) = \sum_{m=1}^{N} a_m p(x)e^{imx} = p(x)\sum_{m=1}^{N} a_m e^{imx}, \tag{39.15}$$

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 (39.15) was presented in [**?**], where it was called the PDFT method. The PDFT was based on an earlier non-iterative version of the Gerchberg-Papoulis bandlimited extrapolation procedure [**?**]. The PDFT was then applied to image reconstruction problems in [**?**]. An application of the PDFT was presented in [**?**]. In [**?**] 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 [**?**] and in [**?**] 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 [34].

Chapter 40

Optimization in Remote Sensing

Once again, the basic problem is to reconstruct or estimate a (possibly complex-valued) function $f(r)$ of several real variables, from finitely many measurements pertaining to $f(r)$. As previously, we shall assume that the measurements g_n take the form

$$g_n = \int_S f(r)\overline{h_n(r)}dr, \tag{40.1}$$

for $n = 1, ..., N$. The problem is highly underdetermined; there are infinitely many functions consistent with the data. One approach to solving such problems is to select a cost function $C(f) \geq 0$ and minimize $C(f)$ over all functions $f(r)$ consistent with the measured data. As we saw previously, cost functions that are Hilbert-space norms are reasonable choices. How we might select the cost function is the subject of this chapter.

40.1 The General Form of the Cost Function

We shall consider cost functions of the form

$$C(f) = \int_S F(f(r), p(r))dr, \tag{40.2}$$

where $p(r)$ is a fixed prior estimate of the true $f(r)$ and $F(y, z) \geq 0$ is to be determined. Such cost functions are viewed as measures of distance between the functions $f(r)$ and $p(r)$. Therefore, we also write

$$D(f, p) = \int_S F(f(r), p(r))dr, \tag{40.3}$$

Our goal is to impose reasonable conditions on these distances $D(f,p)$ sufficiently restrictive to eliminate all but a small class of suitable distances.

40.2 The Conditions

In order for $D(f,p)$ to be viewed as a distance measure, we want $D(f,f)=0$ for all appropriate f. Therefore, we require
Axiom 1: $F(y,y)=0$, for all suitable y.

We also want $D(f,p) \geq D(p,p)$ for all appropriate f and p, so we require
Axiom 2: $F_y(y,y)=0$, for all suitable y.

To make $D(f,p)$ strictly convex in f we impose
Axiom 3: $F_{y,y}(y,z)>0$, for all suitable y and z.

Given $p(r)$ and the data, we find our estimate by minimizing $D(f,p)$ over all appropriate $f(r)$ consistent with the data. The Lagrangian is then

$$L(f,\lambda)=D(f,p)+\sum_{n=1}^{N}\lambda_n(g_n-\int_S f(r)\overline{h_n(r)}dr. \tag{40.4}$$

Taking the first partial derivative of $L(f,\lambda)$ with respect to f gives the Euler equation

$$F_y(f(r),p(r))=\sum_{n=1}^{N}\lambda_n h_n(r). \tag{40.5}$$

Given the data, we must find the λ_n for which the resulting $f(r)$ is consist with the data.

As we vary the values of g_n, the values of the λ_n will change also. The functions $t(r)$ satisfying

$$F_y(t(r),p(r))=\sum_{n=1}^{N}\lambda_n h_n(r), \tag{40.6}$$

for some choice of the λ_n will form the family denoted $\mathcal{T}$. The functions consistent with the data we denote by $\mathcal{Q}$. We seek those functions $F(y,z)$ for which Axiom 4 holds:
Axiom 4: In all cases, the member of $\mathcal{T}$ that minimizes $D(f,t)$ is the function $f(r)$ in $\mathcal{Q}$ that minimizes $D(f,p)$.

In [87] it was shown that the functions $F(y,z)$ that satisfy these four axioms must also have the property

$$F_{z,y,y}(y,z)=0,$$

for all suitable y and z. It follows that there is a strictly convex function $H(y)$ such that

$$F(y,z) = H(y) - H(z) - H'(z)(y-z). \tag{40.7}$$

If $\hat{f}(r)$ is the member of $\mathcal{Q}$ that minimizes $D(f,p)$, then

$$D(f,p) = D(f,\hat{f}) + D(\hat{f},p).$$

There are many F that fit this description. If we impose one more axiom, we can reduce the choce significantly.
Axiom 5: Let $\hat{f}$ minimize $D(f,p)$ over f in $\mathcal{Q}$. Then, for any suitable constant c, $\hat{f}$ also minimizes $D(f,cp)$, over f in Q.

Axiom 5': Let $\hat{f}$ minimize $D(f,p)$ over f in $\mathcal{Q}$. Then, for any suitable constant c, $c\hat{f}$ minimizes $D(f,p)$, over f consistent with the data cg_n.

If the function F satisfies either of these two additional axioms, for all appropriate choices of p, then F is a positive multiple of the Kullback-Leibler distance, that is,

$$F(y,z) = c^2[y\log\frac{y}{z} + z - y],$$

for $y > 0$ and $z > 0$.

Bibliography

[1] Agmon, S. (1954) The relaxation method for linear inequalities, *Canadian Journal of Mathematics*, **6**, pp. 382–392.

[2] Anderson, A. and Kak, A. (1984) Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm, *Ultrasonic Imaging*, **6** 81–94.

[3] Aubin, J.-P., (1993) *Optima and Equilibria: An Introduction to Nonlinear Analysis*, Springer-Verlag.

[4] Axelsson, O. (1994) *Iterative Solution Methods*. Cambridge, UK: Cambridge University Press.

[5] Baillon, J., and Haddad, G. (1977) Quelques proprietes des operateurs angle-bornes et n-cycliquement monotones, *Israel J. of Mathematics*, **26** 137-150.

[6] 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.

[7] 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, Elsevier Publ., pp. 11–22.

[8] Bauschke, H., and Borwein, J. (1996) On projection algorithms for solving convex feasibility problems, *SIAM Review*, **38 (3)**, pp. 367–426.

[9] Bauschke, H., and Borwein, J. (1997) "Legendre functions and the method of random Bregman projections." *Journal of Convex Analysis*, **4**, pp. 27–67.

[10] 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.

[11] Bauschke, H., and Lewis, A. (2000) "Dykstra's algorithm with Bregman projections: a convergence proof." *Optimization*, **48**, pp. 409–427.

[12] Bertero, M., and Boccacci, P. (1998) *Introduction to Inverse Problems in Imaging* Bristol, UK: Institute of Physics Publishing.

[13] Bertsekas, D.P. (1997) "A new class of incremental gradient methods for least squares problems." *SIAM J. Optim.*, **7**, pp. 913-926.

[14] Borwein, J. and Lewis, A. (2000) *Convex Analysis and Nonlinear Optimization*. Canadian Mathematical Society Books in Mathematics, New York: Springer-Verlag.

[15] Bracewell, R.C. (1979) "Image reconstruction in radio astronomy." in [78], pp. 81–104.

[16] 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.

[17] 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.

[18] Brodzik, A. and Mooney, J. (1999) "Convex projections algorithm for restoration of limited-angle chromotomographic images." *Journal of the Optical Society of America A* **16 (2)**, pp. 246–257.

[19] 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.

[20] Byrne, C. (1993) "Iterative image reconstruction algorithms based on cross-entropy minimization." *IEEE Transactions on Image Processing* **IP-2**, pp. 96–103.

[21] 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.

[22] 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.

[23] Byrne, C. (1996) "Block-iterative methods for image reconstruction from projections." *IEEE Transactions on Image Processing* **IP-5**, pp. 792–794.

[24] Byrne, C. (1997) "Convergent block-iterative algorithms for image reconstruction from inconsistent data." *IEEE Transactions on Image Processing* **IP-6**, pp. 1296–1304.

[25] 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.

[26] Byrne, C. (1998) "Iterative deconvolution and deblurring with constraints", *Inverse Problems*, **14**, pp. 1455-1467.

[27] Byrne, C. (1999) "Iterative projection onto convex sets using multiple Bregman distances." *Inverse Problems* **15**, pp. 1295–1313.

[28] Byrne, C. (2000) "Block-iterative interior point optimization methods for image reconstruction from limited data." *Inverse Problems* **16**, pp. 1405–1419.

[29] 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.,

[30] Byrne, C. (2001) "Likelihood maximization for list-mode emission tomographic image reconstruction." *IEEE Transactions on Medical Imaging* **20(10)**, pp. 1084–1092.

[31] Byrne, C. (2002) "Iterative oblique projection onto convex sets and the split feasibility problem." *Inverse Problems* **18**, pp. 441–453.

[32] Byrne, C. (2004) "A unified treatment of some iterative algorithms in signal processing and image reconstruction." *Inverse Problems* **20**, pp. 103–120.

[33] Byrne, C. (2005) Choosing parameters in block-iterative or ordered-subset reconstruction algorithms, *IEEE Transactions on Image Processing*, **14 (3)**, pp. 321–327.

[34] Byrne, C. (2005) *Signal Processing: A Mathematical Approach*, AK Peters, Publ., Wellesley, MA.

[35] Byrne, C. (2005) "Feedback in Iterative Algorithms" unpublished lecture notes.

[36] Byrne, C., and Ward, S. (2005) "Estimating the Largest Singular Value of a Sparse Matrix" in preparation.

[37] 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.

[38] Censor, Y. (1981) "Row-action methods for huge and sparse systems and their applications." *SIAM Review*, **23**: 444–464.

[39] 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.

[40] Censor, Y. and Elfving, T. (1994) A multiprojection algorithm using Bregman projections in a product space, *Numerical Algorithms*, **8** 221–239.

[41] 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.

[42] Censor, Y., Bortfeld, T., Martin, B., and Trofimov, A. (2006) "A unified approach for inversion problems in intensity-modulated radiation therapy." , to appear.

[43] Censor, Y., and Reich, S. (1998) "The Dykstra algorithm for Bregman projections." *Communications in Applied Analysis*, **2**, pp. 323–339.

[44] Censor, Y. and Segman, J. (1987) "On block-iterative maximization." *J. of Information and Optimization Sciences* **8**, pp. 275–291.

[45] Censor, Y. and Zenios, S.A. (1997) *Parallel Optimization: Theory, Algorithms and Applications*. New York: Oxford University Press.

[46] 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.

[47] Cheney, W., and Goldstein, A. (1959) "Proximity maps for convex sets." *Proc. Am. Math. Soc.*, **10**, pp. 448–450.

[48] Cimmino, G. (1938) "Calcolo approssimato per soluzioni die sistemi di equazioni lineari." *La Ricerca Scientifica XVI, Series II, Anno IX* **1**, pp. 326–333.

[49] Combettes, P. (1993) The foundations of set theoretic estimation, *Proceedings of the IEEE*, **81 (2)**, pp. 182–208.

[50] Combettes, P. (1996) The convex feasibility problem in image recovery, *Advances in Imaging and Electron Physics*, **95**, pp. 155–270.

[51] 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.

[52] Combettes, P. (2000) "Fejér monotonicity in convex optimization."in *Encyclopedia of Optimization*, C.A. Floudas and P. M. Pardalos, editors, Boston: Kluwer Publ.

[53] Csiszár, I. and Tusnády, G. (1984) "Information geometry and alternating minimization procedures."*Statistics and Decisions* **Supp. 1**, pp. 205–237.

[54] Csiszár, I. (1989) "A geometric interpretation of Darroch and Ratcliff's generalized iterative scaling."*The Annals of Statistics* **17 (3)**, pp. 1409–1413.

[55] 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.

[56] Darroch, J. and Ratcliff, D. (1972) "Generalized iterative scaling for log-linear models."*Annals of Mathematical Statistics* **43**, pp. 1470–1480.

[57] Dax, A. (1990) "The convergence of linear stationary iterative processes for solving singular unstructured systems of linear equations," *SIAM Review*, **32**, pp. 611–635.

[58] 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.

[59] 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.

[60] 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.

[61] 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)**.

[62] 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.

[63] Duda, R., Hart, P., and Stork, D. (2001) *Pattern Classification*, Wiley.

[64] Dugundji, J. (1970) *Topology* Boston: Allyn and Bacon, Inc.

[65] Dykstra, R. (1983) "An algorithm for restricted least squares regression" *J. Amer. Statist. Assoc.*, **78 (384)**, pp. 837–842.

[66] 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.

[67] Elsner, L., Koltracht, L., and Neumann, M. (1992) "Convergence of sequential and asynchronous nonlinear paracontractions." *Numerische Mathematik*, **62**, pp. 305–319.

[68] Farncombe, T. (2000) "Functional dynamic SPECT imaging using a single slow camera rotation" , *Ph.D. thesis, Dept. of Physics, University of British Columbia*.

[69] 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.

[70] Fleming, W. (1965) *Functions of Several Variables*, Addison-Wesley Publ., Reading, MA.

[71] 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.

[72] 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.

[73] Golshtein, E., and Tretyakov, N. (1996) *Modified Lagrangians and Monotone Maps in Optimization*. New York: John Wiley and Sons, Inc.

[74] 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.

[75] Green, P. (1990) "Bayesian reconstructions from emission tomography data using a modified EM algorithm."*IEEE Transactions on Medical Imaging* **9**, pp. 84–93.

[76] 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**: 1–24.

[77] 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.

[78] Herman, G.T. (ed.) (1979) "Image Reconstruction from Projections" , *Topics in Applied Physics, Vol. 32*, Springer-Verlag, Berlin.

[79] Herman, G.T., and Natterer, F. (eds.) "Mathematical Aspects of Computerized Tomography", *Lecture Notes in Medical Informatics, Vol. 8*, Springer-Verlag, Berlin.

[80] Herman, G.T., Censor, Y., Gordon, D., and Lewitt, R. (1985) Comment (on the paper [120]), *Journal of the American Statistical Association* **80**, pp. 22–25.

[81] Herman, G. T. and Meyer, L. (1993) "Algebraic reconstruction techniques can be made computationally efficient."*IEEE Transactions on Medical Imaging* **12**, pp. 600–609.

[82] Herman, G. T. (1999) *private communication.*

[83] Hildreth, C. (1957) A quadratic programming procedure, *Naval Research Logistics Quarterly*, **4**, pp. 79–85. Erratum, ibid., p. 361.

[84] 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.

[85] 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.

[86] 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.

[87] Jones, L., and Byrne, C. (1990) " General entropy criteria for inverse problems, with applications to data compression, pattern classification, and cluster analysis." *IEEE Transactions on Information Theory* **36 (1)**, pp. 23–30.

[88] Kaczmarz, S. (1937) "Angenäherte Auflösung von Systemen linearer Gleichungen." *Bulletin de l'Academie Polonaise des Sciences et Lettres* **A35**, pp. 355–357.

[89] Kak, A., and Slaney, M. (2001) "Principles of Computerized Tomographic Imaging" , SIAM, Philadelphia, PA.

[90] Koltracht, L., and Lancaster, P. (1990) "Constraining strategies for linear iterative processes." *IMA J. Numer. Anal.*, **10**, pp. 555–567.

[91] Kullback, S. and Leibler, R. (1951) "On information and sufficiency." *Annals of Mathematical Statistics* **22**, pp. 79–86.

[92] Landweber, L. (1951) "An iterative formula for Fredholm integral equations of the first kind." *Amer. J. of Math.* **73**, pp. 615–624.

[93] Lange, K. and Carson, R. (1984) "EM reconstruction algorithms for emission and transmission tomography." *Journal of Computer Assisted Tomography* **8**, pp. 306–316.

[94] 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.

[95] 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.

[96] 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.

[97] Lent, A., and Censor, Y. (1980) Extensions of Hildreth's row-action method for quadratic programming, *SIAM Journal on Control and Optiization*, **18**, pp. 444–454.

[98] 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.

[99] Luenberger, D. (1969) *Optimization by Vector Space Methods*. New York: John Wiley and Sons, Inc.

[100] Mann, W. (1953) "Mean value methods in iteration."*Proc. Amer. Math. Soc.* **4**, pp. 506–510.

[101] McLachlan, G.J. and Krishnan, T. (1997) *The EM Algorithm and Extensions*. New York: John Wiley and Sons, Inc.

[102] 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.

[103] Mooney, J., Vickers, V., An, M., and Brodzik, A. (1997) "High-throughput hyperspectral infrared camera."*Journal of the Optical Society of America, A* **14 (11)**, pp. 2951–2961.

[104] Motzkin, T., and Schoenberg, I. (1954) The relaxation method for linear inequalities, *Canadian Journal of Mathematics*, **6**, pp. 393–404.

[105] 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.

[106] Nash, S. and Sofer, A. (1996) *Linear and Nonlinear Programming*. New York: McGraw-Hill.

[107] Natterer, F. (1986) *Mathematics of Computed Tomography*. New York: John Wiley and Sons, Inc.

[108] Natterer, F., and Wübbeling, F. (2001) *Mathematical Methods in Image Reconstruction*. Philadelphia, PA: SIAM Publ.

[109] Peressini, A., Sullivan, F., and Uhl, J. (1988) *The Mathematics of Nonlinear Programming*. Berlin: Springer-Verlag.

[110] Pretorius, P., 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.

[111] Rockafellar, R. (1970) *Convex Analysis*. Princeton, NJ: Princeton University Press.

[112] 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.

[113] Schmidlin, P. (1972) "Iterative separation of sections in tomographic scintigrams." *Nucl. Med.* **15(1)**.

[114] Schroeder, M. (1991) *Fractals, Chaos, Power Laws*, W.H. Freeman, New York.

[115] Shepp, L., and Vardi, Y. (1982) Maximum likelihood reconstruction for emission tomography, *IEEE Transactions on Medical Imaging*, **MI-1**, pp. 113–122.

[116] 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.

[117] Stark, H. and Yang, Y. (1998) *Vector Space Projections: A Numerical Approach to Signal and Image Processing, Neural Nets and Optics*, John Wiley and Sons, New York.

[118] Tanabe, K. (1971) "Projection method for solving a singular system of linear equations and its applications." *Numer. Math.* **17**, pp. 203–214.

[119] Twomey, S. (1996) *Introduction to the Mathematics of Inversion in Remote Sensing and Indirect Measurement*. New York: Dover Publ.

[120] 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.

[121] Wernick, M. and Aarsvold, J., editors (2004) *Emission Tomography: The Fundamentals of PET and SPECT*. San Diego: Elsevier Academic Press.

[122] Wu, C.F. (1983) "On the convergence properties of the EM algorithm" , *Annals of Statistics*, **11**, pp. 95–103.

[123] Yang, Q. (2004) "The relaxed CQ algorithm solving the split feasibility problem." *Inverse Problems*, **20**, pp. 1261–1266.

[124] 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.

[125] Youla, D. (1978) Generalized image restoration by the method of alternating projections, *IEEE Transactions on Circuits and Systems*, **CAS-25 (9)**, pp. 694–702.

Index