Lecture Notes on Auxiliary-Function and Projection-Based Optimization Methods in Inverse Problems[1]

Charles L. Byrne[2]

June 16, 2013

[1]The latest version is available at http://faculty.uml.edu/cbyrne/cbyrne.html

[2]Charles_Byrne@uml.edu, Department of Mathematical Sciences, University of Massachusetts Lowell, Lowell, MA 01854

Contents

Preface

A fundamental inverse problem is the reconstruction of a function from finitely many measurements pertaining to that function. This problem is central to radar, sonar, optical imaging, transmission and emission tomography, magnetic resonance imaging, and many other applications. Because the measured data is limited, it cannot serve to determine one single correct answer. In each of these applications some sort of prior information is incorporated in the reconstruction process in order to produce a usable solution. Minimizing a cost function is a standard technique used to single out one solution from the many possibilities. The reconstruction algorithms often employ projection techniques to guarantee that the reconstructed function is consistent with the known constraints. Typical image reconstruction problems involve thousands of data values and iterative algorithms are required to perform the desired optimization.

We begin these notes with a typical remote-sensing problem in which the available data are values of the Fourier transform of the function we wish to reconstruct. The function we wish to reconstruct is the amplitude function associated with a spatially extended object transmitting or reflecting electromagnetic radiation. Problems of this sort arise in a variety of applications, from mapping the sourses of sunspot activity to synthetic-aperture radar and magnetic-resonance imaging. Our example is a somewhat simplified version of what is encountered in the real world, but it serves to illustrate several key aspects of most remote-sensing problems. From this example we see why it is that the data is limited, apart, of course, from the obvious need to limit ourselves to finitely many data values, and come to understand how resolution depends on the relationship between the size of the object being imaged and the frequency of the probing or tranmitted signal.

Because our data is limited and the reconstruction problems are underdetermined, we are led to consider minimum-norm reconstructions. Once we have settled on an appropriate ambient space, usually a Hilbert space, in which to place the function to be reconstructed, it is reasonable to take as the reconstruction the data-consistent member of the space having the smallest norm. If we have additional constraints that we wish to impose, we

can use orthogonal projection onto convex sets to satisfy the constraints. A key step, and one that is too often overlooked, is the choice of the ambient space. As we shall see, soft constraints coming from prior information, such as knowledge of the overall shape of the function being reconstructed, or of some prominent features of that function, can often be incorporated in the reconstruction process through the choice of the ambient space. Although Hilbert space norms are the most convenient, other Banach space norms, or distance measures not derived from norms, such as cross-entropy, can also be helpful.

It is usually the case that the function we wish to reconstruct is a real- or complex-valued function of one or more continuous variables. At some stage of the reconstruction, we must discretize the function or its estimate, if only to plot the estimate at the final step. It can be helpful to introduce the discretization earlier in the process, and most of our discussion here will focus on reconstructing a finite vector in $\mathbb{R}^J$. Once we have decided to base the reconstruction on the minimization of some cost function, we need to find an appropriate algorithm; our focus here will be on iterative minimization algorithms.

In simple terms, the basic problem is to minimize a function $f : X \to (-\infty, \infty]$, over a non-empty subset C of X, where X is an arbitrary set. At the kth step of a sequential optimization algorithm we optimize a function $G_k(x) = f(x) + g_k(x)$ to get x^k. In what we call here an auxiliary-function (AF) algorithm we minimize $G_k(x)$ over $x \in C$, and require that the auxiliary function $g_k(x)$ be chosen so that $g_k(x^{k-1}) = 0$ and $g_k(x) \geq 0$ for all $x \in C$. Auxiliary-function (AF) methods are closely related to sequential unconstrained minimization (SUM) procedures such as the barrier- and penalty-function algorithms. As normally formulated, barrier-function methods and penalty-function methods are not AF methods, but can be reformulated as AF methods. Many projection-based methods are also AF methods.

Our main objective is to select the $g_k(x)$ so that the infinite sequence $\{x^k\}$ generated by our algorithm converges to a solution of the problem; this, of course, requires some topology on the set X. Failing that, we want the sequence $\{f(x^k)\}$ to converge to $d = \inf\{f(x)|x \in C\}$ or, at the very least, for the sequence $\{f(x^k)\}$ to be non-increasing.

An auxiliary-function algorithm is in the SUMMA class if, for all $x \in C$, $G_k(x) - G_k(x^k) \geq g_{k+1}(x) \geq 0$. If $\{x^k\}$ is generated by an algorithm in the SUMMA class, then the sequence $\{f(x^k)\}$ converges to d.

A wide variety of iterative methods, including barrier-function and penalty-function methods, can be formulated to be members of the SUMMA class. Other members of the SUMMA class include projection-based methods, proximal minimization algorithms using Bregman distances, forward-backward splitting methods, the CQ algorithm for the split feasibility problem, the simultaneous MART algorithm, alternating minimization meth-

Chapter 1

Introduction

1.1 Fourier Transform Data

We begin with the fundamental remote-sensing problem of reconstructing a function from finitely many values of its Fourier transform. We choose to begin with this particular problem not only because it is commonly encountered in a variety of applications, but also because it introduces several important issues that we face in other reconstruction problems as well. Because these problems are under-determined, it is reasonable to minimize a cost function to help us select one solution from the many that are possible. This leads us to consideration of algorithms for performing these minimizations.

1.2 Sequential Optimization

Many applications of optimization involve solving large systems of linear or non-linear equations, usually subject to constraints on the variables. When the data is insufficient to specify a single unique solution, one can optimize a real-valued function such as a norm or cross-entropy, subject to consistency with the data and other constraints. Since data is typically noisy, regularized solutions that are not exactly consistent with the measured data are preferred. Many of these methods employ projections onto convex sets, either explicitly or implicitly.

Optimizing a real-valued function $f(x)$ subject to constraints on the independent vector variable x can be a difficult problem to solve; typically, iterative algorithms are required. The idea in sequential optimization is to replace the single difficult optimization problem with a sequence of simpler optimization problems. At the kth step of a sequential optimization algorithm we optimize $f(x) + g_k(x)$ to get x^k. Sequential unconstrained

minimization (SUM) techniques often used to solve such problems [47].

Suppose that the problem is to minimize a function $f : X \to \mathbb{R}$, over $x \in C \subseteq X$, where X is an arbitrary set. At the kth step of a SUM algorithm we minimize the function $f(x) + g_k(x)$ to get x^k. The functions $g_k(x)$ may force x^k to be within C, as with barrier-function methods, or may penalize violations of the constraint, as with penalty-function methods. The $g_k(x)$ may also be selected so that x^k can be expressed in closed form.

Auxiliary-function (AF) methods, which is our topic here, closely resemble SUM methods. In AF methods we minimize $G_k(x) = f(x) + g_k(x)$ over $x \in C$ to get x^k. Certain restrictions are placed on the auxiliary functions $g_k(x)$ to control the behavior of the sequence $\{f(x^k)\}$. In the best of cases, the sequence of minimizers will converge to a solution of the original constrained minimization problem, or, failing that, their function values will converge to the constrained minimum, or, at least, will be nonincreasing. Even when there are no constraints, the problem of minimizing a real-valued function may require iteration; the formalism of AF minimization can be useful in deriving such iterative algorithms, as well as in proving convergence.

1.3 Examples of SUM

Barrier-function algorithms and penalty-function algorithms are two of the best known examples of SUM.

1.3.1 Barrier-Function Methods

Suppose that $C \subseteq \mathbb{R}^J$ and $b : C \to \mathbb{R}$ is a barrier function for C, that is, b has the property that $b(x) \to +\infty$ as x approaches the boundary of C. At the kth step of the iteration we minimize

$$B_k(x) = f(x) + \frac{1}{k} b(x) \tag{1.1}$$

to get x^k. Then each x^k is in C. We want the sequence $\{x^k\}$ to converge to some x^* in the closure of C that solves the original problem. Barrier-function methods are called interior-point methods because each x^k satisfies the constraints.

For example, suppose that we want to minimize the function $f(x) = f(x_1, x_2) = x_1^2 + x_2^2$, subject to the constraint that $x_1 + x_2 \geq 1$. The constraint is then written $g(x_1, x_2) = 1 - (x_1 + x_2) \leq 0$. We use the logarithmic barrier function $b(x) = -\log(x_1 + x_2 - 1)$. For each positive integer k, the vector $x^k = (x_1^k, x_2^k)$ minimizing the function

$$B_k(x) = x_1^2 + x_2^2 - \frac{1}{k} \log(x_1 + x_2 - 1) = f(x) + \frac{1}{k} b(x)$$

has entries

$$x_1^k = x_2^k = \frac{1}{4} + \frac{1}{4}\sqrt{1 + \frac{4}{k}}.$$

Notice that $x_1^k + x_2^k > 1$, so each x^k satisfies the constraint. As $k \to +\infty$, x^k converges to $(\frac{1}{2}, \frac{1}{2})$, which is the solution to the original problem. The use of the logarithmic barrier function forces $x_1 + x_2 - 1$ to be positive, thereby enforcing the constraint on $x = (x_1, x_2)$.

1.3.2 Penalty-Function Methods

Again, our goal is to minimize a function $f : \mathbb{R}^J \to \mathbb{R}$, subject to the constraint that $x \in C$, where C is a non-empty closed subset of $\mathbb{R}^J$. We select a non-negative function $p : \mathbb{R}^J \to \mathbb{R}$ with the property that $p(x) = 0$ if and only if x is in C and then, for each positive integer k, we minimize

$$P_k(x) = f(x) + kp(x), \tag{1.2}$$

to get x^k. We then want the sequence $\{x^k\}$ to converge to some $x^* \in C$ that solves the original problem. In order for this iterative algorithm to be useful, each x^k should be relatively easy to calculate.

If, for example, we should select $p(x) = +\infty$ for x not in C and $p(x) = 0$ for x in C, then minimizing $P_k(x)$ is equivalent to the original problem and we have achieved nothing.

As an example, suppose that we want to minimize the function $f(x) = (x+1)^2$, subject to $x \geq 0$. Let us select $p(x) = x^2$, for $x \leq 0$, and $p(x) = 0$ otherwise. Then $x^k = \frac{-1}{k+1}$, which converges to the right answer, $x^* = 0$, as $k \to \infty$.

1.4 Auxiliary-Function Methods

In this section we define auxiliary-function methods, establish their basic properties, and give several examples to be considered in more detail later.

1.4.1 General AF Methods

Let C be a non-empty subset of an arbitrary set X, and $f : X \to \mathbb{R}$. We want to minimize $f(x)$ over x in C. At the kth step of an auxiliary-function (AF) algorithm we minimize

$$G_k(x) = f(x) + g_k(x) \tag{1.3}$$

over $x \in C$ to obtain x^k. Our main objective is to select the $g_k(x)$ so that the infinite sequence $\{x^k\}$ generated by our algorithm converges to a solution of the problem; this, of course, requires some topology on the

set X. Failing that, we want the sequence $\{f(x^k)\}$ to converge to $d = \inf\{f(x)|x \in C\}$ or, at the very least, for the sequence $\{f(x^k)\}$ to be non-increasing.

1.4.2 AF Requirements

It is part of the definition of AF methods that the auxiliary functions $g_k(x)$ be chosen so that $g_k(x) \geq 0$ for all $x \in C$ and $g_k(x^{k-1}) = 0$. We have the following proposition.

Proposition 1.1 *Let the sequence $\{x^k\}$ be generated by an AF algorithm. Then the sequence $\{f(x^k)\}$ is non-increasing, and, if d is finite, the sequence $\{g_k(x^k)\}$ converges to zero.*

Proof: We have

$$f(x^k) + g_k(x^k) = G_k(x^k) \leq G_k(x^{k-1}) = f(x^{k-1}) + g_k(x^{k-1}) = f(x^{k-1}).$$

Therefore,

$$f(x^{k-1}) - f(x^k) \geq g_k(x^k) \geq 0.$$

Since the sequence $\{f(x^k)\}$ is decreasing and bounded below by d, the difference sequence must converge to zero, if d is finite; therefore, the sequence $\{g_k(x^k)\}$ converges to zero in this case. ∎

The auxiliary functions used in Equation (1.1) do not have these properties but the barrier-function algorithm can be reformulated as an AF method. The iterate x^k obtained by minimizing $B_k(x)$ in Equation (1.1) also minimizes the function

$$G_k(x) = f(x) + [(k-1)f(x) + b(x)] - [(k-1)f(x^{k-1}) + b(x^{k-1})]. \quad (1.4)$$

The auxiliary functions

$$g_k(x) = [(k-1)f(x) + b(x)] - [(k-1)f(x^{k-1}) + b(x^{k-1})] \quad (1.5)$$

now have the desired properties. In addition, we have $G_k(x) - G_k(x^k) = g_{k+1}(x)$ for all $x \in C$, which will become significant shortly.

As originally formulated, the penalty-function methods do not fit into the class of AF methods we consider here. However, a reformulation of the penalty-function approach, with $p(x)$ and $f(x)$ switching roles, permits the penalty-function methods to be studied as barrier-function methods, and therefore as acceptable AF methods.

1.4.3 Majorization Minimization

Majorization minimization (MM), also called optimization transfer, is a technique used in statistics to convert a difficult optimization problem into a sequence of simpler ones [67, 7, 58]. The MM method requires that we majorize the objective function $f(x)$ with $g(x|y)$, such that $g(x|y) \geq f(x)$, for all x, and $g(y|y) = f(y)$. At the kth step of the iterative algorithm we minimize the function $g(x|x^{k-1})$ to get x^k.

The MM algorithms are members of the AF class. At the kth step of an MM iteration we minimize

$$G_k(x) = f(x) + [g(x|x^{k-1}) - f(x)] = f(x) + d(x, x^{k-1}), \tag{1.6}$$

where $d(x, z)$ is some distance function satisfying $d(x, z) \geq 0$ and $d(z, z) = 0$. Since $g_k(x) = d(x, x^{k-1}) \geq 0$ and $g_k(x^{k-1}) = 0$, MM methods are also AF methods; it then follows that the sequence $\{f(x^k)\}$ is non-increasing.

All MM algorithms have the form $x^k = Tx^{k-1}$, where T is the operator defined by

$$Tz = \text{argmin}_x\{f(x) + d(x, z)\}. \tag{1.7}$$

If $d(x, z) = \frac{1}{2}\|x - z\|_2^2$, then T is Moreau's proximity operator $Tz = \text{prox}_f(z)$ [62, 63, 64].

1.4.4 The Method of Auslander and Teboulle

The method of Auslander and Teboulle [1] is a particular example of an MM algorithm. We take C to be a closed, non-empty, convex subset of $\mathbb{R}^J$, with interior U. At the kth step of their method one minimizes a function

$$G_k(x) = f(x) + d(x, x^{k-1}) \tag{1.8}$$

to get x^k. Their distance $d(x, y)$ is defined for x and y in U, and the gradient with respect to the first variable, denoted $\nabla_1 d(x, y)$, is assumed to exist. The distance $d(x, y)$ is not assumed to be a Bregman distance. Instead, they assume that the distance d has an associated *induced proximal distance* $H(a, b) \geq 0$, finite for a and b in U, with $H(a, a) = 0$ and

$$\langle \nabla_1 d(b, a), c - b \rangle \leq H(c, a) - H(c, b), \tag{1.9}$$

for all c in U.

If $d = D_h$, that is, if d is a Bregman distance, then from the equation

$$\langle \nabla_1 d(b, a), c - b \rangle = D_h(c, a) - D_h(c, b) - D_h(b, a) \tag{1.10}$$

we see that D_h has $H = D_h$ for its associated induced proximal distance, so D_h is *self-proximal*, in the terminology of [1].

1.4.5 The EM Algorithm

The *expectation maximization maximum likelihood* (EM) "algorithm" is not a single algorithm, but a framework, or, as the authors of [7] put it, a "prescription" , for constructing algorithms. Nevertheless, we shall refer to it as the EM algorithm.

The EM algorithm is always presented within the context of statistical likelihood maximization, but the essence of this method is not stochastic; the EM methods can be shown to be a subclass of AF methods. We present now the essential aspects of the EM algorithm without relying on statistical concepts.

The problem is to maximize a non-negative function $f : Z \to \mathbb{R}$, where Z is an arbitrary set; in the stochastic context $f(z)$ is a likelihood function of the parameter vector z. We assume that there is $z^* \in Z$ with $f(z^*) \geq f(z)$, for all $z \in Z$.

We also assume that there is a non-negative function $b : \mathbb{R}^J \times Z \to \mathbb{R}$ such that

$$f(z) = \int b(x, z)dx.$$

Having found z^{k-1}, we maximize the function

$$H(z^{k-1}, z) = \int b(x, z^{k-1}) \log b(x, z)dx \tag{1.11}$$

to get z^k. Adopting such an iterative approach presupposes that maximizing $H(z^{k-1}, z)$ is simpler than maximizing $f(z)$ itself. This is the case with the EM algorithms.

The cross-entropy or Kullback-Leibler distance [54] is a useful tool for analyzing the EM algorithm. For positive numbers u and v, the Kullback-Leibler distance from u to v is

$$KL(u, v) = u \log \frac{u}{v} + v - u. \tag{1.12}$$

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 a and b we have

$$KL(a, b) = \sum_{j=1}^{J} KL(a_j, b_j). \tag{1.13}$$

One of the most useful and easily proved facts about the KL distance is contained in the following lemma.

Lemma 1.1 *For non-negative vectors a and b, with $b_+ = \sum_{j=1}^{J} b_j > 0$, we have*

$$KL(a, b) = KL(a_+, b_+) + KL(a, \frac{a_+}{b_+}b). \tag{1.14}$$

This lemma can be extended to obtain the following useful identity; we simplify the notation by setting $b(z) = b(x, z)$.

Lemma 1.2 *For $f(z)$ and $b(x, z)$ as above, and z and w in Z, with $f(w) > 0$, we have*

$$KL(b(z), b(w)) = KL(f(z), f(w)) + KL(b(z), (f(z)/f(w))b(w)). \quad (1.15)$$

Maximizing $H(z^{k-1}, z)$ is equivalent to minimizing

$$G_k(z) = G(z^{k-1}, z) = -f(z) + KL(b(z^{k-1}), b(z)), \quad (1.16)$$

where

$$g_k(z) = KL(b(z^{k-1}), b(z)) = \int KL(b(x, z^{k-1}), b(x, z))dx. \quad (1.17)$$

Since $g_k(z) \geq 0$ for all z and $g_k(z^{k-1}) = 0$, we have an AF method. Without additional restrictions, we cannot conclude that $\{f(z^k)\}$ converges to $f(z^*)$.

We get z^k by minimizing $G_k(z) = G(z^{k-1}, z)$. When we minimize $G(z, z^k)$, we get z^k again. Therefore, we can put the EM algorithm into the alternating minimization (AM) framework of Csiszár and Tusnády [42], to be discussed later.

1.5 The SUMMA Class of AF Methods

As we have seen, whenever the sequence $\{x^k\}$ is generated by an AF algorithm, the sequence $\{f(x^k)\}$ is non-increasing. We want more, however; we want the sequence $\{f(x^k)\}$ to converge to d. This happens for those AF algorithms in the SUMMA class.

1.5.1 The SUMMA Condition

An AF algorithm is said to be in the SUMMA class if the auxiliary functions $g_k(x)$ are chosen so that the SUMMA condition holds; that is,

$$G_k(x) - G_k(x^k) \geq g_{k+1}(x) \geq 0, \quad (1.18)$$

for all $x \in C$. As we just saw, the reformulated barrier-function method is in the SUMMA class. We have the following theorem.

Theorem 1.1 *If the sequence $\{x^k\}$ is generated by an algorithm in the SUMMA class, then the sequence $\{f(x^k)\}$ converges to $d = \inf\{f(x)|x \in C\}$.*

Proof: Suppose that there is $d^* > d$ with $f(x^k) \geq d^*$, for all k. Then there is z in C with

$$f(x^k) \geq d^* > f(z) \geq d,$$

for all k. From the inequality (1.18) we have

$$g_{k+1}(z) \leq G_k(z) - G_k(x^k),$$

and so, for all k,

$$g_k(z) - g_{k+1}(z) \geq f(x^k) + g_k(x^k) - f(z) \geq f(x^k) - f(z) \geq d^* - f(z) > 0.$$

This tells us that the nonnegative sequence $\{g_k(z)\}$ is decreasing, but that successive differences remain bounded away from zero, which cannot happen. ∎

1.5.2 Auslander and Teboulle Revisited

The method of Auslander and Teboulle described previously seems not to be a particular case of SUMMA. However, we can adapt the proof of Theorem 1.1 to prove the analogous result for their method. We assume that $f(\hat{x}) \leq f(x)$, for all x in C.

Theorem 1.2 *For $k = 2, 3, ...$, let x^k minimize the function*

$$G_k(x) = f(x) + d(x, x^{k-1}).$$

If the distance d has an induced proximal distance H, then $\{f(x^k)\} \to f(\hat{x})$.

Proof: We know that the sequence $\{f(x^k)\}$ is decreasing and the sequence $\{d(x^k, x^{k-1})\}$ converges to zero. Now suppose that

$$f(x^k) \geq f(\hat{x}) + \delta,$$

for some $\delta > 0$ and all k. Since $\hat{x}$ is in C, there is z in U with

$$f(x^k) \geq f(z) + \frac{\delta}{2},$$

for all k. Since x^k minimizes $F_k(x)$, it follows that

$$0 = \nabla f(x^k) + \nabla_1 d(x^k, x^{k-1}).$$

Using the convexity of the function $f(x)$ and the fact that H is an induced proximal distance, we have

$$0 < \frac{\delta}{2} \leq f(x^k) - f(z) \leq \langle -\nabla f(x^k), z - x^k \rangle =$$

$$\langle \nabla_1 d(x^k, x^{k-1}), z - x^k \rangle \leq H(z, x^{k-1}) - H(z, x^k).$$

Therefore, the nonnegative sequence $\{H(z, x^k)\}$ is decreasing, but its successive differences remain bounded below by $\frac{\delta}{2}$, which is a contradiction. ∎

It is interesting to note that the Auslander-Teboulle approach places a restriction on the function $d(x, y)$, the existence of the induced proximal distance H, that is unrelated to the objective function $f(x)$, but this condition is helpful only for convex $f(x)$. In contrast, the SUMMA approach requires that

$$0 \leq g_{k+1}(x) \leq G_k(x) - G_k(x^k),$$

which involves the $f(x)$ being minimized, but does not require that this $f(x)$ be convex.

1.5.3 Proximal Minimization

Let $f : \mathbb{R}^J \to (-\infty, +\infty]$ be a convex differentiable function. Let h be another convex function, with effective domain D, that is differentiable on the nonempty open convex set int D. Assume that $f(x)$ is finite on $C = \overline{D}$ and attains its minimum value on C at $\hat{x}$. The corresponding *Bregman distance* $D_h(x, z)$ is defined for x in D and z in int D by

$$D_h(x, z) = h(x) - h(z) - \langle \nabla h(z), x - z \rangle. \tag{1.19}$$

Note that $D_h(x, z) \geq 0$ always. If h is essentially strictly convex, then $D_h(x, z) = 0$ implies that $x = z$. Our objective is to minimize $f(x)$ over x in $C = \overline{D}$.

At the kth step of a *proximal minimization algorithm* (PMA) [40, 26], we minimize the function

$$G_k(x) = f(x) + D_h(x, x^{k-1}), \tag{1.20}$$

to get x^k. The function

$$g_k(x) = D_h(x, x^{k-1}) \tag{1.21}$$

is nonnegative and $g_k(x^{k-1}) = 0$. We assume that each x^k lies in int D. As we shall see,

$$G_k(x) - G_k(x^k) \geq D_h(x, x^k) = g_{k+1}(x) \geq 0, \tag{1.22}$$

so the PMA is in the SUMMA class.

The PMA can present some computational obstacles. When we minimize $G_k(x)$ to get x^k we find that we must solve the equation

$$\nabla h(x^{k-1}) - \nabla h(x^k) \in \partial f(x^k), \tag{1.23}$$

where the set $\partial f(x)$ is the sub-differential of f at x, given by

$$\partial f(x) := \{u | \langle u, y - x\rangle \leq f(y) - f(x), \text{for all } y\}. \tag{1.24}$$

When $f(x)$ is differentiable, we must solve

$$\nabla f(x^k) + \nabla h(x^k) = \nabla h(x^{k-1}). \tag{1.25}$$

A modification of the PMA, called the IPA for *interior-point algorithm* [26, 30], is designed to overcome these computational obstacles. We discuss the IPA in the next subsection. Another modification of the PMA that is similar to the IPA is the *forward-backward splitting* (FBS) method to be discussed in a later section.

1.5.4 The IPA

In this subsection we describe a modification of the PMA, an interior-point algorithm called the IPA, that helps us overcome the computational obstacles encountered in the PMA. To simplify the discussion, we assume in this subsection that $f(x)$ is differentiable.

At the kth step of the PMA we minimize

$$G_k(x) = f(x) + D_h(x, x^{k-1}), \tag{1.26}$$

where $h(x)$ is as in the previous subsection. Writing

$$a(x) = h(x) + f(x), \tag{1.27}$$

we must solve the equation

$$\nabla a(x^k) = \nabla a(x^{k-1}) - \nabla f(x^{k-1}). \tag{1.28}$$

In the IPA we select $a(x)$ so that Equation (1.28) is easily solved and so that $h(x) = a(x) - f(x)$ is convex and differentiable. Later in this paper we shall present several examples of the IPA. The projected gradient descent algorithm, discussed in the next subsection, is one such example.

1.5.5 Projected Gradient Descent

The problem now is to minimize $f : \mathbb{R}^J \to \mathbb{R}$, over the closed, non-empty convex set C, where f is convex and differentiable on $\mathbb{R}^J$. We assume now that the gradient operator ∇f is L-Lipschitz continuous; that is, for all x and y, we have

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2. \tag{1.29}$$

To employ the IPA approach, we let $0 < \gamma < \frac{1}{L}$ and select the function

$$a(x) = \frac{1}{2\gamma}\|x\|_2^2; \tag{1.30}$$

the upper bound on γ guarantees that the function $h(x) = a(x) - f(x)$ is convex.

At the kth step we minimize

$$G_k(x) = f(x) + D_h(x, x^{k-1})$$

$$= f(x) + \frac{1}{2\gamma}\|x - x^{k-1}\|_2^2 - D_f(x, x^{k-1}), \tag{1.31}$$

over $x \in C$. The solution x^k is in C and satisfies the inequality

$$\langle x^k - (x^{k-1} - \gamma\nabla f(x^{k-1}), c - x^k\rangle \geq 0, \tag{1.32}$$

for all $c \in C$. It follows then that

$$x^k = P_C(x^{k-1} - \gamma\nabla f(x^{k-1})); \tag{1.33}$$

here P_C denotes the orthogonal projection onto C. This is the projected gradient descent algorithm. For convergence we must require that f have certain additional properties to be discussed later. Note that the auxiliary function

$$g_k(x) = \frac{1}{2\gamma}\|x - x^{k-1}\|_2^2 - D_f(x, x^{k-1}) \tag{1.34}$$

is unrelated to the set C, so is not used here to incorporate the constraint; it is used to provide a closed-form iterative scheme.

When $C = \mathbb{R}^J$ we have no constraint and the problem is simply to minimize f. Then the iterative algorithm becomes

$$x^k = x^{k-1} - \gamma\nabla f(x^{k-1}); \tag{1.35}$$

this is the gradient descent algorithm.

1.5.6 Relaxed Gradient Descent

In the gradient descent method we move away from the current x^{k-1} by the vector $\gamma\nabla f(x^{k-1})$. In relaxed gradient descent, the magnitude of the movement is reduced by α, where $\alpha \in (0, 1)$. Such relaxation methods are sometimes used to accelerate convergence. The relaxed gradient descent method can also be formulated as an AF method.

At the kth step we minimize

$$G_k(x) = f(x) + \frac{1}{2\gamma\alpha}\|x - x^{k-1}\|_2^2 - D_f(x, x^{k-1}), \tag{1.36}$$

obtaining

$$x^k = x^{k-1} - \alpha\gamma\nabla f(x^{k-1}). \tag{1.37}$$

1.5.7 Regularized Gradient Descent

In many applications the function to be minimized involves measured data, which is typically noisy, as well as some less than perfect model of how the measured data was obtained. In such cases, we may not want to minimize $f(x)$ exactly. In regularization methods we add to $f(x)$ another function that is designed to reduce sensitivity to noise and model error.

For example, suppose that we want to minimize

$$\alpha f(x) + \frac{1-\alpha}{2}\|x - p\|_2^2, \tag{1.38}$$

where p is chosen a priori.

At the kth step we minimize

$$G_k(x) = f(x) + \frac{1}{2\gamma\alpha}\|x - x^{k-1}\|_2^2 - \frac{1}{\alpha}(x, x^{k-1})] + \frac{1-\alpha}{2\gamma\alpha}\|x - p\|_2^2, \tag{1.39}$$

obtaining

$$x^k = \alpha(x^{k-1} - \gamma\nabla f(x^{k-1})) + (1-\alpha)p. \tag{1.40}$$

If we select $p = 0$ the iterative step becomes

$$x^k = \alpha(x^{k-1} - \gamma\nabla f(x^{k-1})). \tag{1.41}$$

1.6 A Convergence Theorem

So far, we haven't discussed the restrictions necessary to prove convergence of these iterative algorithms. The AF framework can be helpful in this regard, as we illustrate now.

The following theorem concerns convergence of the projected gradient descent algorithm with iterative step given by Equation (1.33).

Theorem 1.3 *Let $f : \mathbb{R}^J \to \mathbb{R}$ be differentiable, with L-Lipschitz continuous gradient. For γ in the interval $(0, \frac{1}{L})$ the sequence $\{x^k\}$ given by Equation (1.33) converges to a minimizer of f, over $x \in C$, whenever minimizers exist.*

Proof: The auxiliary function

$$g_k(x) = \frac{1}{2\gamma}\|x - x^{k-1}\|_2^2 - D_f(x, x^{k-1}) \tag{1.42}$$

can be rewritten as

$$g_k(x) = D_h(x, x^{k-1}), \tag{1.43}$$

where

$$h(x) = \frac{1}{2\gamma}\|x\|_2^2 - f(x). \tag{1.44}$$

Therefore, $g_k(x) \geq 0$ whenever $h(x)$ is a convex function.

We know that $h(x)$ is convex if and only if

$$\langle \nabla h(x) - \nabla h(y), x - y\rangle \geq 0, \tag{1.45}$$

for all x and y. This is equivalent to

$$\frac{1}{\gamma}\|x - y\|_2^2 - \langle \nabla f(x) - \nabla f(y), x - y\rangle \geq 0. \tag{1.46}$$

Since ∇f is L-Lipschitz, the inequality (1.46) holds whenever $0 < \gamma < \frac{1}{L}$.

A relatively simple calculation shows that

$$G_k(x) - G_k(x^k) =$$

$$\frac{1}{2\gamma}\|x - x^k\|_2^2 + \frac{1}{\gamma}\langle x^k - (x^{k-1} - \gamma\nabla f(x^{k-1})), x - x^k\rangle. \tag{1.47}$$

From Equation (1.33) it follows that

$$G_k(x) - G_k(x^k) \geq \frac{1}{2\gamma}\|x - x^k\|_2^2, \tag{1.48}$$

for all $x \in C$, so that, for all $x \in C$, we have

$$G_k(x) - G_k(x^k) \geq \frac{1}{2\gamma}\|x - x^k\|_2^2 - D_f(x, x^k) = g_{k+1}(x). \tag{1.49}$$

Now let $\hat{x}$ minimize $f(x)$ over all $x \in C$. Then

$$G_k(\hat{x}) - G_k(x^k) = f(\hat{x}) + g_k(\hat{x}) - f(x^k) - g_k(x^k)$$

$$\leq f(\hat{x}) + G_{k-1}(\hat{x}) - G_{k-1}(x^{k-1}) - f(x^k) - g_k(x^k),$$

so that

$$\Big(G_{k-1}(\hat{x}) - G_{k-1}(x^{k-1})\Big) - \Big(G_k(\hat{x}) - G_k(x^k)\Big) \geq f(x^k) - f(\hat{x}) + g_k(x^k) \geq 0.$$

Therefore, the sequence $\{G_k(\hat{x}) - G_k(x^k)\}$ is decreasing and the sequences $\{g_k(x^k)\}$ and $\{f(x^k) - f(\hat{x})\}$ converge to zero.

From

$$G_k(\hat{x}) - G_k(x^k) \geq \frac{1}{2\gamma}\|\hat{x} - x^k\|_2^2,$$

it follows that the sequence $\{x^k\}$ is bounded. Let $\{x^{k_n}\}$ converge to x^* with $\{x^{k_n+1}\}$ converging to x^{**}; we then have $f(x^*) = f(x^{**}) = f(\hat{x})$.

Replacing the generic $\hat{x}$ with x^{**}, we find that $\{G_{k_n+1}(x^{**}) - G_{k_n+1}(x^{k_n+1})\}$ is decreasing. By Equation (1.47), this subsequence converges to zero; therefore, the entire sequence $\{G_k(x^{**}) - G_k(x^k)\}$ converges to zero. From the inequality in (1.48), we conclude that the sequence $\{\|x^{**} - x^k\|_2^2\}$ converges to zero, and so $\{x^k\}$ converges to x^{**}. This completes the proof of the theorem. ∎

Chapter 2

Fourier Transform Data

We begin with an example from remote sensing that will illustrate several of the issues we shall consider in more detail later.

2.1 Fourier Series Expansions

Let $f(x) : [-L, L] \to \mathbb{C}$ have Fourier series representation

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{in\pi x/L}, \tag{2.1}$$

where the Fourier coefficient c_n is given by

$$c_n = \frac{1}{2L} \int_{-L}^{L} f(x) e^{-in\pi x/L} dx. \tag{2.2}$$

We shall see how Fourier coefficients can arise as data obtained through measurements. However, we shall be able to measure only a finite number of the Fourier coefficients. One issue that will concern us is the effect on the approximation of $f(x)$ if we use some, but not all, of its Fourier coefficients.

2.2 The Discrete Fourier Transform

Suppose that we have c_n for $|n| \leq N$. It is not unreasonable to try to estimate the function $f(x)$ using the *discrete Fourier transform* (DFT) estimate, which is

$$f_{DFT}(x) = \sum_{n=-N}^{N} c_n e^{in\pi x/L}. \tag{2.3}$$

In Figure 2.1 below, the function $f(x)$ is the solid-line figure in both graphs. In the bottom graph, we see the true $f(x)$ and a DFT estimate. The top graph is the result of *band-limited extrapolation*, a technique for predicting missing Fourier coefficients that we shall discuss later.

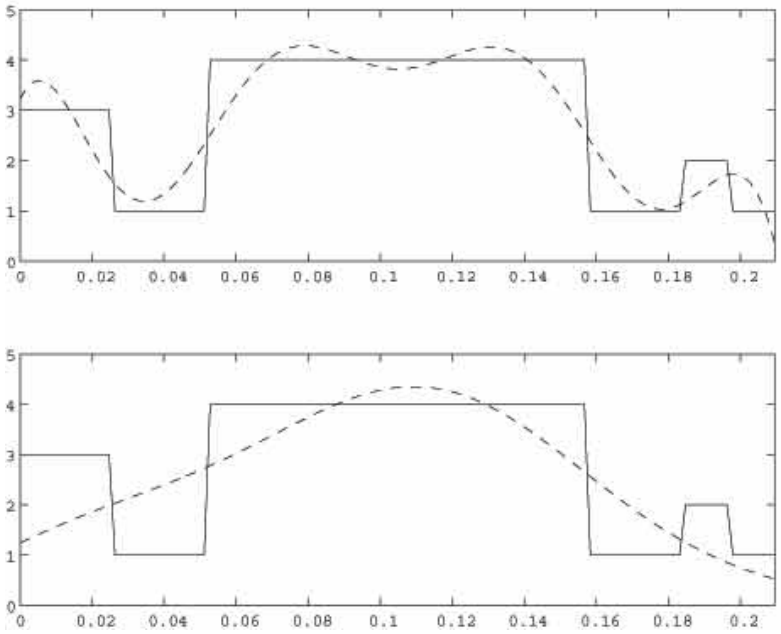

Figure 2.1: The non-iterative band-limited extrapolation method (MDFT) (top) and the DFT (bottom) for 30 times over-sampled data.

2.3 The Unknown Amplitude Problem

In this example, we imagine that each point x in the interval $[-L, L]$ is sending a signal at the frequency ω, each with its own amplitude $f(x)$; that is, the signal sent by the point x is

$$f(x)e^{i\omega t}; \tag{2.4}$$

here the amplitude contains both magnitude and phase, so is complex. We imagine that the amplitude function $f(x)$ is unknown and we want to determine it. It could be the case that the signals originate at the points x, as with light or radio waves from the sun, or are simply reflected from the points x, as is sunlight from the moon or radio waves in radar.

Now let us consider what is received by a point P on the circumference of a circle centered at the origin and having large radius D. The point P corresponds to the angle θ as shown in Figure 2.2. It takes a finite time for the signal sent from x at time t to reach P, so there is a delay.

We assume that c is the speed at which the signal propagates. Because D is large relative to L, we make the *far-field assumption*, which allows us to approximate the distance from x to P by $D - x\cos(\theta)$. Therefore, what P receives at time t from x is approximately what was sent from x at time $t - \frac{1}{c}(D - x\cos(\theta))$.

At time t, the point P receives from x the signal

$$f(x)e^{i\omega(t-\frac{1}{c}(D-x\cos(\theta)))}, \tag{2.5}$$

or

$$e^{i\omega(t-\frac{1}{c}D)}f(x)e^{i\omega x\cos(\theta)/c}. \tag{2.6}$$

Therefore, from our measurement at P, we obtain

$$e^{i\omega(t-\frac{1}{c}D)}\int_{-L}^{L} f(x)e^{i\omega x\cos(\theta)/c}dx. \tag{2.7}$$

Consequently, from measurements in the farfield we obtain the values

$$\int_{-L}^{L} f(x)e^{i\omega x\cos(\theta)/c}dx, \tag{2.8}$$

where θ can be chosen as any angle between 0 and 2π. When we select θ so that

$$\frac{\omega\cos(\theta)}{c} = \frac{n\pi}{L}, \tag{2.9}$$

we have c_{-n}.

2.4 Limited Data

Note that we will be able to solve Equation (2.9) for θ only if we have

$$|n| \leq \frac{L\omega}{\pi c}. \tag{2.10}$$

This tells us that we can measure only finitely many of the Fourier coefficients of $f(x)$. It is common in signal processing to speak of the *wavelength* of a sinusoidal signal; the wavelength associated with a given ω and c is

$$\lambda = \frac{2\pi c}{\omega}. \tag{2.11}$$

Therefore, we can measure c_n for $|n|$ not greater than $\frac{2L}{\lambda}$, which is the length of the interval $[-L, L]$, measured in units of wavelength λ. We get more Fourier coefficients when the product $L\omega$ is larger; this means that when L is small, we want ω to be large, so that λ is small and we can measure more Fourier coefficients. As we saw previously, using these finitely many Fourier coefficients to calculate the DFT reconstruction of $f(x)$ can lead to a poor estimate of $f(x)$, particularly when we don't have many Fourier coefficients.

2.5 Can We Get More Data?

As we just saw, we can make measurements at any point P in the far-field; perhaps we do not need to limit ourselves to just those angles that lead to the limited number of Fourier coefficients c_n.

We define the Fourier transform of the function $f(x)$ to be the function

$$F(\gamma) = \int_{-L}^{L} f(x) e^{i\gamma x} dx. \tag{2.12}$$

Therefore, when we measure the signals received at the point P in the far-field, we obtain the value $F(\gamma)$ for $\gamma = \omega \cos(\theta)/c$. Therefore, in principle, we have available to us all the values of $F(\gamma)$ for γ in the interval $[-\omega/c, \omega/c]$. These are not all of the non-zero values of $F(\gamma)$, of course, since $F(\gamma)$ is band-limited, but not support-limited.

2.6 Over-Sampling

It is sometimes argued that once we have obtained all the values of c_n that are available to us, there is no more information about $f(x)$ that we can obtain through further measurements in the far-field; this is wrong. It may come as somewhat of a surprise, but from the theory of complex analytic functions we can prove that there is enough data available to us here to reconstruct $f(x)$ perfectly, at least in principle. The drawback, in practice, is that the measurements would have to be free of noise and impossibly accurate. All is not lost, however.

Suppose, for the sake of illustration, that we measure the far-field signals at points P corresponding to angles θ that satisfy

$$\frac{\omega \cos(\theta)}{c} = \frac{n\pi}{2L}, \tag{2.13}$$

instead of

$$\frac{\omega \cos(\theta)}{c} = \frac{n\pi}{L}.$$

Now we have twice as many data points and from our new measurements we can obtain

$$a_m = \frac{1}{4L}\int_{-L}^{L} f(x)e^{-ix\frac{m\pi}{2L}}\,dx = \frac{1}{4L}\int_{-2L}^{2L} f(x)e^{-ix\frac{m\pi}{2L}}\,dx, \tag{2.14}$$

for $|m| \leq M$, which are Fourier coeffcients of $f(x)$ when viewed as a function defined on the interval $[-2L, 2L]$, but still zero outside $[-L, L]$. We say now that our data is *twice over-sampled*. Note that we call it *over-sampled* because the rate at which we are sampling is higher, even though the distance between samples is lower.

For clarity, let us denote the function defined on the interval $[-2L, 2L]$ that equals $f(x)$ for x in $[-L, L]$ and is equal to zero elsewhere as $g(x)$. We have twice the number of Fourier coefficients that we had previously, but for the function $g(x)$. A DFT reconstruction using this larger set of Fourier coefficients will reconstruct $g(x)$ on the interval $[-2L, 2L]$; this DFT estimate is

$$g_{DFT}(x) = \sum_{m=-M}^{M} a_m e^{im\pi x/2L}, \tag{2.15}$$

for $|x| \leq 2L$. This will give us a reconstruction of $f(x)$ itself over the interval $[-L, L]$, but will also give us a reconstruction of the rest of $g(x)$, which we already know to be zero. So we are wasting the additional data by reconstructing $g(x)$ instead of $f(x)$. We need to use our prior knowledge that $g(x) = 0$ for $L < |x| \leq 2L$.

We want to use the prior knowledge that $f(x) = 0$ for $L < |x| \leq 2L$ to improve our reconstruction. Suppose that we take as our reconstruction the *modified DFT* (MDFT) [13]:

$$f_{MDFT}(x) = \sum_{j=-M}^{M} b_j e^{ij\pi x/2L}, \tag{2.16}$$

for $|x| \leq L$, and zero elsewhere, with the b_j chosen so that $f_{MDFT}(x)$ is consistent with the measured data. Calculating this estimator involves solving a system of linear equations for the b_j.

We must have

$$a_m = \frac{1}{4L}\int_{-L}^{L} f_{MDFT}(x)e^{-ix\frac{m\pi}{2L}}\,dx, \tag{2.17}$$

for $|m| \leq M$. Since

$$\int_{-L}^{L} e^{ix\frac{j\pi}{2L}}e^{-ix\frac{m\pi}{2L}}\,dx = 2L\frac{\sin(j-m)\pi/2}{(j-m)\pi/2}, \tag{2.18}$$

the system to be solved for the b_j is

$$a_m = \frac{1}{2} \sum_{j=-M}^{M} b_j \frac{\sin(j-m)\pi/2}{(j-m)\pi/2}, \tag{2.19}$$

for $|m| \leq M$. The top graph in Figure (2.1) illustrates the improvement over the DFT that can be had using the MDFT. In that figure, we took data that was thirty times over-sampled, not just twice over-sampled, as in our previous discussion. Consequently, we had thirty times the number of Fourier coefficients we would have had otherwise, but for an interval thirty times longer. To get the top graph, we used the MDFT, with the prior knowledge that $f(x)$ was non-zero only within the central thirtieth of the long interval. The bottom graph shows the DFT reconstruction using the larger data set, but only for the central thirtieth of the full period, which is where the original $f(x)$ is non-zero.

2.7 Band-Limited Extrapolation

Once we have solved the system of linear equations in (2.19) for the b_j, we can use Equation (2.19) with any integer values of m to obtain the Fourier coeffcients of $f_{MDFT}(x)$ that were not measured. In this way, we extrapolate the measured data to an infinite sequence. These extrapolated values are the true Fourier coeffcients of $f_{MDFT}(x)$, but estimated values of the unmeasured Fourier coeffcients of $f(x)$ itself.

2.8 A Projection-Based View

When we view the function $f(x)$ as a member of the Hilbert space $L^2(-L, L)$, we find that the DFT estimate of $f(x)$ is the orthogonal projection of the zero function onto the closed convex subset of all members of $L^2(-L, L)$ that are consistent with the data; that is, the DFT estimate is the member of $L^2(-L, L)$ that has minimum norm among all those members consistent with the data. The MDFT estimate is the member of $L^2(-2L, 2L)$ of minimum norm among all members that are both consistent with the data and supported on the interval $[-L, L]$. The MDFT estimate is also the member of $L^2(-L, L)$ of minimum norm consistent with the over-sampled data. The MDFT is not the DFT in this case, since the functions $e^{ij\pi x/2L}$ are not orthogonal with respect to the usual inner product on $L^2(-L, L)$.

2.9 Other Forms of Prior Knowledge

As we just showed, knowing that we have over-sampled in our measurements can help us improve the resolution in our estimate of $f(x)$. We may

have other forms of prior knowledge about $f(x)$ that we can use. If we know something about large-scale features of $f(x)$, but not about finer details, we can use the PDFT estimate, which is a generalization of the MDFT [14, 15].

We can write the MDFT estimate above as

$$f_{MDFT}(x) = \chi_{[-L,L]}(x) \sum_{j=-M}^{M} b_j e^{ij\pi x/2L}; \tag{2.20}$$

here $\chi_{[-L,L]}(x)$ is one for $|x| \leq L$, and zero, otherwise. Written this way, we see that the second factor has the algebraic form of the DFT estimate, while the first factor incorporates our prior knowledge that $f(x)$ is zero for $|x| > L$.

Suppose that we have some prior knowledge of the function $|f(x)|$ beyond simply support information. Let us select $p(x) > 0$ as a prior estimate of $|f(x)|$ and let our PDFT estimate of $f(x)$ have the form

$$f_{PDFT}(x) = p(x) \sum_{j=-M}^{M} d_j e^{ij\pi x/2L}, \tag{2.21}$$

with the coefficients d_j computed by forcing $f_{PDFT}(x)$ to be consistent with the measured data. Again, this involves solving a system of linear equations, although there are other ways to handle this. The PDFT approach extends to higher dimensions, as we illustrate in the following example.

The original image on the upper right of Figure 2.3 is a discrete rectangular array of intensity values simulating a slice of a head. The data was obtained by taking the two-dimensional discrete Fourier transform of the original image, and then discarding, that is, setting to zero, all these spatial frequency values, except for those in a smaller rectangular region around the origin. The problem then is under-determined. A minimum-norm solution would seem to be a reasonable reconstruction method.

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

would already know about the skull; it would be the interior that we want to know about.

Using our knowledge of the presence of a skull, which we might have obtained from the minimum-norm reconstruction itself, we construct the prior estimate shown in the upper left. Now we use the same data as before, and calculate a minimum-weighted-norm reconstruction, using as the weight vector the reciprocals of the values of the prior image. This minimum-weighted-norm reconstruction is shown on the lower left; it is clearly almost the same as the original image. The calculation of the minimum-weighted norm solution can be done iteratively using the ART algorithm [73].

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

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

2.10 A Broader View

In the cases just discussed we have linear data pertaining to an unknown function that we wish to approximate. We view the function as a member of a Hilbert space and the data as linear functional values. To be more specific, we take f to be a member of a Hilbert space H having inner product $\langle f, g\rangle$, and model the measured data β_n as $\beta_n = \langle f, g_n\rangle$, where the g_n are known members of H and $|n| \leq N$. The minimum-norm approximation of f is $\hat{f}$ given by

$$\hat{f} = \sum_{m=-N}^{N} \alpha_m g_m, \tag{2.22}$$

where the α_m are found by solving the system of linear equations

$$\beta_n = \sum_{m=-N}^{N} \langle g_m, g_n\rangle \, \alpha_m, \tag{2.23}$$

for $|n| \leq N$.

When we have the limited data

$$c_n = \frac{1}{2L} \int_{-L}^{L} f(x) e^{-in\pi x/L} dx, \tag{2.24}$$

for $|n| \leq N$, it is natural to imagine $f(x)$ to be in the Hilbert space $L^2(-L, L)$ and for the inner product to be the usual one. In that case, the minimum-norm approximation of $f(x)$ is the DFT, which is a linear combination of the functions

$$g_n(x) = e^{in\pi x/L}. \tag{2.25}$$

But we can rewrite Equation (2.24) as

$$c_n = \frac{1}{2L}\int_{-L}^{L} f(x)p(x)e^{-in\pi x/L}\, p(x)^{-1}dx. \tag{2.26}$$

Now we may consider $f(x)$ to be a member of the Hilbert space having the inner product

$$\langle f, g\rangle = \int_{-L}^{L} f(x)\overline{g(x)}p(x)^{-1}dx. \tag{2.27}$$

Now the $g_n(x)$ are the functions

$$g_n(x) = p(x)e^{in\pi x/L}, \tag{2.28}$$

and the minimum-norm solution in this weighted L^2 space is the PDFT, which is a linear combination of these new $g_n(x)$.

2.11 Lessons Learned

For $f : [-L, L] \to \mathbb{C}$, the Fourier coeffcients of $f(x)$ are values of the Fourier transform $F(\gamma)$, at the values $\gamma = \frac{n\pi}{L}$. According to the Shannon Sampling Theorem, the band-limited function $F(\gamma)$ can be completely recovered from the values $F(\frac{n\pi}{L})$, which is just another way of saying that $F(\gamma)$ can be determined from knowledge of $f(x)$, which we would have if we knew all its Fourier coeffcients. The spacing $\Delta = \frac{\pi}{L}$ between the samples of $F(\gamma)$ is called the *Nyquist spacing*.

In the unknown-amplitude problem considered previously, we were able to measure $F(\frac{n\pi}{L})$ for only those n with $|n| \leq N$. It would be wrong, however, to conclude that Shannon's Theorem tells us that there is no further information to be had by measuring at other points P in the farfield; Shannon's Theorem tells us that the infinitely many samples at the Nyquist spacing are sufficient, but not finitely many. By over-sampling we can obtain further data pertaining to $f(x)$ which can help us improve the estimate, providing that we process it properly.

In our discussion of the MDFT, we over-sampled by a factor of two and in Figure 2.1 we over-sampled by a factor of thirty. There is a practical limit to how much over-sampling can be used, however. The matrix

that occurs in the system of equations (2.19) will become increasingly ill-conditioned as the degree of over-sampling increases, until slight errors in the measurements will cause the MDFT estimate to be unusable. We can guard against this ill-conditioning, up to a point, by adding to this matrix a small positive multiple of the identity matrix.

The ill-conditioning can be turned to our advantage sometimes, though. If we calculate the MDFT estimate for $f(x)$, assuming that $f(x)$ is supported on the interval $[-L, L]$, but $f(x)$ is not exactly zero outside this interval, then the MDFT estimate will reflect this error, and have an unrealistically high norm. If we do not know the exact support of $f(x)$, we can perform several MDFT estimates, each with different guesses as to the true support. When these guesses are too small, we will get a high norm; when we have selected an interval that does support $f(x)$, the norm will be much more reasonable. This idea was used to develop a method for solving the *phase problem* [18].

In the phase problem our data are $|c_n|$, not c_n. In [18] we used the MDFT approximation based on estimates of the true phases combined with the magnitude data. When the phases were incorrect the norm of the MDFT approximation was too high. By monitoring the MDFT norm, we were able to modify the phases iteratvely, to approach the correct values and obtain a decent reconstruction of the image.

When we have limited linear functional data pertaining to $f(x)$ it is reasonable to consider a minimum-norm approximation, subject to data consistency. At the same time, it is important to keep in mind that there will be multiple ways to represent the linear functional data, each corresponding to the particular ambient Hilbert space that we choose to use. Projection onto closed convex sets is a useful tool for incorporating hard constraints, such as data consistency, while the proper choice of the ambient Hilbert space can help us include soft constraints, such as prior knowledge of the general shape of the function being estimated.

Hilbert space norms are the easiest to deal with, but there is good reason to consider minimum-norm solutions for norms that are not Hilbert space norms, such as the L^1 or l^1 norms. As we shall see in subsequent chapters, some constraints, such as non-negativity, can be imposed through the use of distance measures like the Bregman distances that do not come from Hilbert space or Banach space norms.

2.12 Continuous or Discrete?

Throughout this chapter we have considered only the reconstruction of a function of a continuous real variable, although we could have extended the discussion to the multi-variate case. Obviously, at the end of the estimation process we must have a discrete version of our estimator in order to plot a

graph or make an image. It is also possible to perform the discretization step earlier in the process. Transforming the problem into the estimation of a finite vector can simplify the calculations in the PDFT, for example.

Suppose that our data is

$$c_n = \frac{1}{2L}\int_{-L}^{L} f(x)e^{-in\pi x/L}dx, \tag{2.29}$$

for $|n| \leq N$. The system of equations that we must solve to get the coeffcients d_j in Equation (2.21) is

$$2\pi c_n = \sum_{j=-N}^{N} d_j P((j-n)\pi/L), \tag{2.30}$$

where $P(\gamma)$ is the Fourier transform of $p(x)$ given by

$$P(\gamma) = \int_{-L}^{L} p(x)e^{ix\gamma}dx. \tag{2.31}$$

Calculating the needed values of $P(\gamma)$ can be difficult, especially in two dimensional image-processing problems. As was demonstrated in [72, 73], discretizing the problem at the beginning makes it possible to avoid this computation. The PDFT is a minimum-weighted-norm solution and its discrete version can be found using an iterative ART or Landweber algorithm and a discrete approximation of $p(x)$.

2.13 A Projection-Based Characterization of Bregman Distances

Suppose that X is a member of a Hilbert space $\mathcal{H}$ that we wish to estimate from data values

$$d_m = \langle X, G_m\rangle, \tag{2.32}$$

for $m = 1, ..., N$. The member Y of $\mathcal{H}$ of smallest norm that is also consistent with the data is

$$Y = \sum_{n=1}^{N} a_n G_n, \tag{2.33}$$

where the a_n are determined by making Y consistent with the data, which means solving the system of linear equations

$$d_m = \sum_{n=1}^{N} a_n \langle G_n, G_m\rangle, \tag{2.34}$$

for $m = 1, ..., N$. It follows from the orthogonality principle in Hilbert space that, among all members of the subspace $\mathcal{G}$ of $\mathcal{H}$ spanned by the set $\{G_1, ..., G_N\}$, the one closest to X is Y. The orthogonal projection of 0 onto the affine set of data-consistent members of $\mathcal{H}$ is also the best approximation of X in $\mathcal{G}$.

As we have seen, a useful method for reconstructing an unknown function from limited linear-functional data is to project a prior estimate of the function onto the subset of functions consistent with the data. The distance measure used is often a Hilbert-space distance, but other distances, such as the l^1 distance or the Kullback-Leibler (cross-entropy) distance are also helpful choices. In this section we consider the problem of reconstructing a positive function $R(x)$, given the data

$$r_n = \int R(x) g_n(x) dx, \tag{2.35}$$

for $n = 1, ..., N$. The distance we use has the form

$$D(Q, P) = \int f(Q(x), P(x)) dx, \tag{2.36}$$

where $Q(x)$ and $P(x)$ are positive functions and $f : \mathbb{R}^2 \to \mathbb{R}_+$ satisfies certain conditions to be presented shortly. The estimation procedure is the following: given a prior estimate $P(x) > 0$ of $R(x)$, we obtain our estimate $Q(x) = S(x)$ by minimizing $D(Q, P)$ over all $Q(x) \in \mathcal{Q}$, where $\mathcal{Q}$ denotes the collection of all positive functions $Q(x)$ consistent with the data.

Our goal in reconstruction is to use the data to find a function that is close to $R(x)$ in some reasonable sense. In the cases considered previously in this chapter, we have always forced our reconstruction to be consistent with the data. It is not obvious that minimizing $D(Q, P)$ over all $Q \in \mathcal{Q}$ necessarily provides an estimate $S(x)$ that is close to $R(x)$. Perhaps simply asking that the reconstruction be consistent with the data is not making the best use of the data. But what else can we do? It would be comforting to know that forcing data consistency is a good idea.

From the Euler-Lagrange differential equation we know that

$$f_y(S(x), P(x)) = \sum_{n=1}^{N} a_n g_n(x), \tag{2.37}$$

for some coefficients a_n chosen to make $S(x)$ consistent with the data. Let $\mathcal{T}$ be the collection of all positive functions $T(x)$ with the property that

$$f_y(T(x), P(x)) = \sum_{n=1}^{N} t_n g_n(x), \tag{2.38}$$

for some choice of coefficients t_n. The question we ask in this section is the following: for which distances D do we always have

$$\text{argmin}_{t\in\mathcal{T}} D(R,T) = \text{argmin}_{Q\in\mathcal{Q}} D(Q,P)? \tag{2.39}$$

To be clear, we are asking that, for fixed $P(x)$ and fixed $g_n(x), n = 1, ..., N$, Equation (2.39) hold, for any choice of $R(x)$ and the r_n. In other words, is the data-consistent solution of Equation (2.38) always the solution of Equation (2.38) closest to R? As we shall see, the property in Equation (2.39), called *directed orthogonality* in [52], characterizes distances D that have the algebraic form of Bregman distances.

In order for the distance given by Equation (2.36) to have the property $D(P,P) = 0$ we assume that $f(y,y) = 0$, for all $y > 0$. In order to have $D(Q,P) \geq D(P,P)$ we assume that $f_y(y,y) = 0$, for all $y > 0$. We also want $D(Q,P)$ to be strictly convex in the variable Q, so we assume that $f_{yy}(y,z) > 0$, for all $y > 0$ and $z > 0$.

In [52] it was shown that, under suitable technical restrictions, if D has the directed orthogonality property, then $f_{zyy}(y,z) = 0$, for all $y > 0$ and $z > 0$. From this, it follows that

$$f(y,z) = J(y) - J(z) - J'(z)(y-z), \tag{2.40}$$

where $J^{''}(z) > 0$. Therefore, D has the algebraic form of a Bregman distance. It was also established in [52] that D has the directed orthogonality property if and only if the triangle equality always holds; that is,

$$D(R,P) = D(R,S) + D(S,P). \tag{2.41}$$

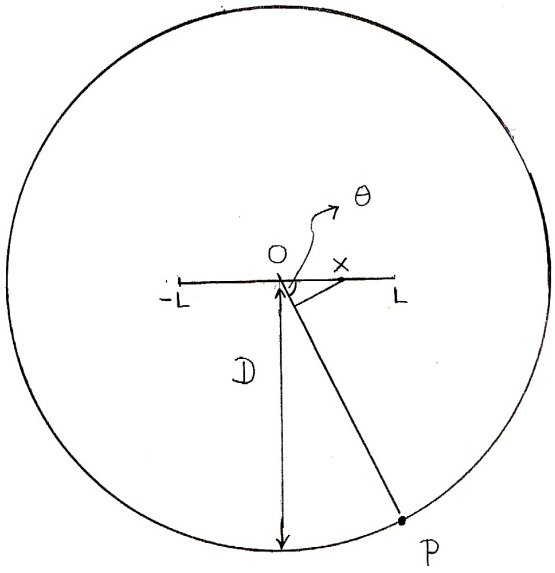

Figure 2.2: Farfield Measurements. The distance from x to P is approximately $D - x\cos\theta$.

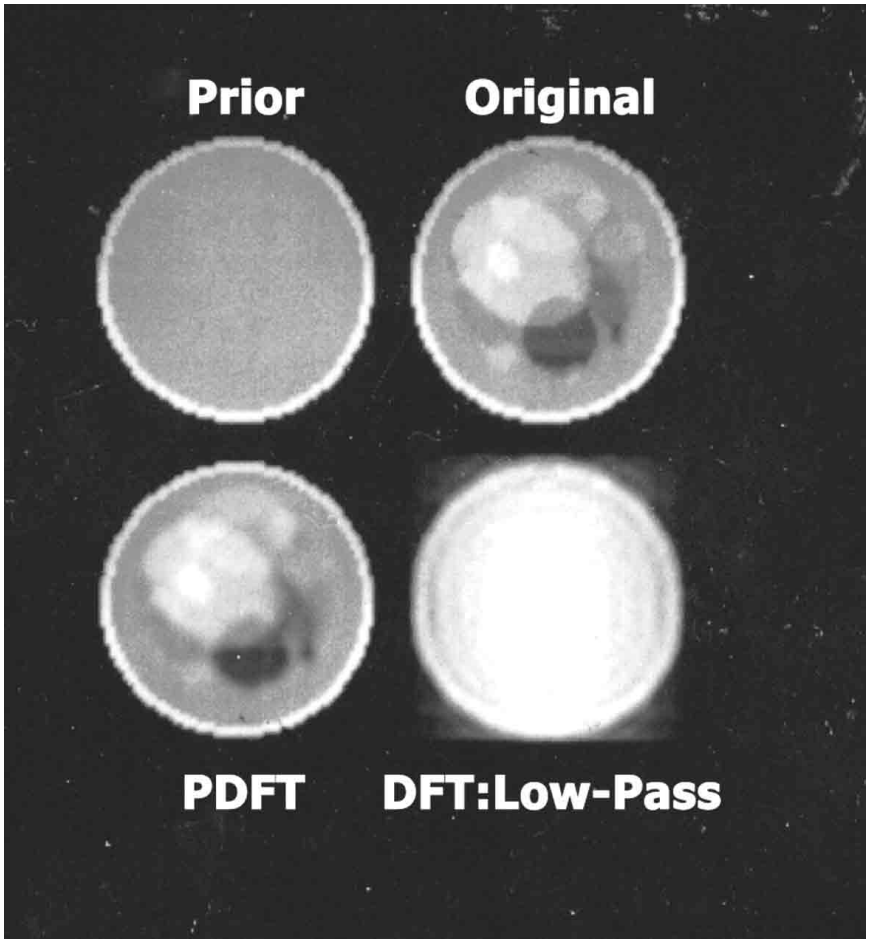

Figure 2.3: Extracting information in image reconstruction. The original is in the upper right, the DFT is the minimum-two-norm reconstruction, the PDFT is a minimum-weighted-two-norm reconstruction, and the top left is the prior estimate.

Chapter 3

Barrier-function Methods

3.1 Barrier Functions

Let $b(x) : \mathbb{R}^J \to (-\infty, +\infty]$ be continuous, with effective domain the set

$$D = \{x | \, b(x) < +\infty\}.$$

The goal is to minimize the objective function $f(x)$, over x in C, the closure of D. We assume that there is $\hat{x} \in C$ with $f(\hat{x}) \leq f(x)$, for all x in C.

In the barrier-function method, we minimize

$$B_k(x) = f(x) + \frac{1}{k} b(x) \tag{3.1}$$

over x in D to get x^k. Each x^k lies within D, so the method is an interior-point algorithm. If the sequence $\{x^k\}$ converges, the limit vector x^* will be in C and $f(x^*) = f(\hat{x})$.

Barrier functions typically have the property that $b(x) \to +\infty$ as x approaches the boundary of D, so not only is x^k prevented from leaving D, it is discouraged from approaching the boundary.

3.2 Examples of Barrier Functions

Consider the convex programming (CP) problem of minimizing the convex function $f : \mathbb{R}^J \to \mathbb{R}$, subject to $g_i(x) \leq 0$, where each $g_i : \mathbb{R}^J \to \mathbb{R}$ is convex, for $i = 1, ..., I$. Let $D = \{x | g_i(x) < 0, i = 1, ..., I\}$; then D is open. We consider two barrier functions appropriate for this problem.

3.2.1 The Logarithmic Barrier Function

A suitable barrier function is the *logarithmic barrier function*

$$b(x) = \Big(-\sum_{i=1}^{I} \log(-g_i(x)) \Big). \tag{3.2}$$

The function $-\log(-g_i(x))$ is defined only for those x in D, and is positive for $g_i(x) > -1$. If $g_i(x)$ is near zero, then so is $-g_i(x)$ and $b(x)$ will be large.

3.2.2 The Inverse Barrier Function

Another suitable barrier function is the *inverse barrier function*

$$b(x) = \sum_{i=1}^{I} \frac{-1}{g_i(x)}, \tag{3.3}$$

defined for those x in D.

In both examples, when k is small, the minimization pays more attention to $b(x)$, and less to $f(x)$, forcing the $g_i(x)$ to be large negative numbers. But, as k grows larger, more attention is paid to minimizing $f(x)$ and the $g_i(x)$ are allowed to be smaller negative numbers. By letting $k \to \infty$, we obtain an iterative method for solving the constrained minimization problem.

Barrier-function methods are particular cases of the SUMMA. The iterative step of the barrier-function method can be formulated as follows: minimize

$$f(x) + [(k-1)f(x) + b(x)] \tag{3.4}$$

to get x^k. Since, for $k = 2, 3, ...$, the function

$$(k-1)f(x) + b(x) \tag{3.5}$$

is minimized by x^{k-1}, the function

$$g_k(x) = (k-1)f(x) + b(x) - (k-1)f(x^{k-1}) - b(x^{k-1}) \tag{3.6}$$

is nonnegative, and x^k minimizes the function

$$G_k(x) = f(x) + g_k(x). \tag{3.7}$$

From

$$G_k(x) = f(x) + (k-1)f(x) + b(x) - (k-1)f(x^{k-1}) - b(x^{k-1}),$$

it follows that

$$G_k(x) - G_k(x^k) = kf(x) + b(x) - kf(x^k) - b(x^k) = g_{k+1}(x),$$

so that $g_{k+1}(x)$ satisfies the condition in (1.18). This shows that the barrier-function method is a particular case of SUMMA.

From the properties of SUMMA algorithms, we conclude that $\{f(x^k)\}$ is decreasing to $f(\hat{x})$, and that $\{g_k(x^k)\}$ converges to zero. From the non-negativity of $g_k(x^k)$ we have that

$$(k-1)(f(x^k) - f(x^{k-1})) \geq b(x^{k-1}) - b(x^k).$$

Since the sequence $\{f(x^k)\}$ is decreasing, the sequence $\{b(x^k)\}$ must be increasing, but might not be bounded above.

If $\hat{x}$ is unique, and $f(x)$ has bounded level sets, then it follows, from our discussion of SUMMA, that $\{x^k\} \to \hat{x}$. Suppose now that $\hat{x}$ is not known to be unique, but can be chosen in D, so that $G_k(\hat{x})$ is finite for each k. From

$$f(\hat{x}) + \frac{1}{k}b(\hat{x}) \geq f(x^k) + \frac{1}{k}b(x^k)$$

we have

$$\frac{1}{k}\Big(b(\hat{x}) - b(x^k)\Big) \geq f(x^k) - f(\hat{x}) \geq 0,$$

so that

$$b(\hat{x}) - b(x^k) \geq 0,$$

for all k. If either f or b has bounded level sets, then the sequence $\{x^k\}$ is bounded and has a cluster point, x^* in C. It follows that $b(x^*) \leq b(\hat{x}) < +\infty$, so that x^* is in D. If we assume that $f(x)$ is convex and $b(x)$ is strictly convex on D, then we can show that x^* is unique in D, so that $x^* = \hat{x}$ and $\{x^k\} \to \hat{x}$.

To see this, assume, to the contrary, that there are two distinct cluster points x^* and x^{**} in D, with

$$\{x^{k_n}\} \to x^*,$$

and

$$\{x^{j_n}\} \to x^{**}.$$

Without loss of generality, we assume that

$$0 < k_n < j_n < k_{n+1},$$

for all n, so that

$$b(x^{k_n}) \leq b(x^{j_n}) \leq b(x^{k_{n+1}}).$$

Therefore,

$$b(x^*) = b(x^{**}) \leq b(\hat{x}).$$

From the strict convexity of $b(x)$ on the set D, and the convexity of $f(x)$, we conclude that, for $0 < \lambda < 1$ and $y = (1-\lambda)x^* + \lambda x^{**}$, we have $b(y) < b(x^*)$ and $f(y) \leq f(x^*)$. But, we must then have $f(y) = f(x^*)$. There must then be some k_n such that

$$G_{k_n}(y) = f(y) + \frac{1}{k_n}b(y) < f(x_{k_n}) + \frac{1}{k_n}b(x_{k_n}) = G_{k_n}(x^{k_n}).$$

But, this is a contradiction. ∎

The following theorem summarizes what we have shown with regard to the barrier-function method.

Theorem 3.1 *Let $f : \mathbb{R}^J \to (-\infty, +\infty]$ be a continuous function. Let $b(x) : \mathbb{R}^J \to (0, +\infty]$ be a continuous function, with effective domain the nonempty set D. Let $\hat{x}$ minimize $f(x)$ over all x in $C = \overline{D}$. For each positive integer k, let x^k minimize the function $f(x) + \frac{1}{k}b(x)$. Then the sequence $\{f(x^k)\}$ is monotonically decreasing to the limit $f(\hat{x})$, and the sequence $\{b(x^k)\}$ is increasing. If $\hat{x}$ is unique, and $f(x)$ has bounded level sets, then the sequence $\{x^k\}$ converges to $\hat{x}$. In particular, if $\hat{x}$ can be chosen in D, if either $f(x)$ or $b(x)$ has bounded level sets, if $f(x)$ is convex and if $b(x)$ is strictly convex on D, then $\hat{x}$ is unique in D and $\{x^k\}$ converges to $\hat{x}$.*

At the kth step of the barrier method we must minimize the function $f(x) + \frac{1}{k}b(x)$. In practice, this must also be performed iteratively, with, say, the Newton-Raphson algorithm. It is important, therefore, that barrier functions be selected so that relatively few Newton-Raphson steps are needed to produce acceptable solutions to the main problem. For more on these issues see Renegar [68] and Nesterov and Nemirovski [66].

Chapter 4

Penalty-function Methods

4.1 Interior- and Exterior-Point Methods

When we add a barrier function to $f(x)$ we restrict the domain. When the barrier function is used in a sequential unconstrained minimization algorithm, the vector x^k that minimizes the function $f(x)+\frac{1}{k}b(x)$ lies in the effective domain D of $b(x)$, and we proved that, under certain conditions, the sequence $\{x^k\}$ converges to a minimizer of the function $f(x)$ over the closure of D. The constraint of lying within the set $\overline{D}$ is satisfied at every step of the algorithm; for that reason such algorithms are called interior-point methods. Constraints may also be imposed using a penalty function. In this case, violations of the constraints are discouraged, but not forbidden. When a penalty function is used in a sequential unconstrained minimization algorithm, the x^k need not satisfy the constraints; only the limit vector need be feasible.

4.2 Examples of Penalty Functions

Consider the convex programming problem. We wish to minimize the convex function $f(x)$ over all x for which the convex functions $g_i(x) \leq 0$, for $i = 1, ..., I$. At the kth step of a penalty-function algorithm we minimize the function

$$P_k(x) = f(x) + kp(x), \tag{4.1}$$

to get x^k.

4.2.1 The Absolute-Value Penalty Function

We let $g_i^+(x) = \max\{g_i(x), 0\}$, and

$$p(x) = \sum_{i=1}^{I} g_i^+(x). \tag{4.2}$$

This is the *Absolute-Value* penalty function; it penalizes violations of the constraints $g_i(x) \leq 0$, but does not forbid such violations. Then, for $k = 1, 2, ...,$ we minimize $P_k(x)$ to get x^k. As $k \to +\infty$, the penalty function becomes more heavily weighted, so that, in the limit, the constraints $g_i(x) \leq 0$ should hold. Because only the limit vector satisfies the constraints, and the x^k are allowed to violate them, such a method is called an *exterior-point* method.

4.2.2 The Courant-Beltrami Penalty Function

The *Courant-Beltrami* penalty-function method is similar, but uses

$$p(x) = \sum_{i=1}^{I} [g_i^+(x)]^2. \tag{4.3}$$

4.2.3 The Quadratic-Loss Penalty Function

Penalty methods can also be used with equality constraints. Consider the problem of minimizing the convex function $f(x)$, subject to the constraints $g_i(x) = 0$, $i = 1, ..., I$. The *quadratic-loss* penalty function is

$$p(x) = \frac{1}{2} \sum_{i=1}^{I} (g_i(x))^2. \tag{4.4}$$

The inclusion of a penalty term can serve purposes other than to impose constraints on the location of the limit vector. In image processing, it is often desirable to obtain a reconstructed image that is locally smooth, but with well defined edges. Penalty functions that favor such images can then be used in the iterative reconstruction [48]. We survey several instances in which we would want to use a penalized objective function.

4.2.4 Regularized Least-Squares

Suppose we want to solve the system of equations $Ax = b$. The problem may have no exact solution, precisely one solution, or there may be infinitely many solutions. If we minimize the function

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

we get a *least-squares* solution, generally, and an exact solution, whenever exact solutions exist. When the matrix A is ill-conditioned, small changes in the vector b can lead to large changes in the solution. When the vector b comes from measured data, the entries of b may include measurement errors, so that an exact solution of $Ax = b$ may be undesirable, even when such exact solutions exist; exact solutions may correspond to x with unacceptably large norm, for example. In such cases, we may, instead, wish to minimize a function such as

$$\frac{1}{2}\|Ax - b\|_2^2 + \frac{\epsilon}{2}\|x - z\|_2^2, \tag{4.5}$$

for some vector z. If $z = 0$, the minimizing vector x_ϵ is then a *norm-constrained* least-squares solution. We then say that the least-squares problem has been *regularized*. In the limit, as $\epsilon \to 0$, these regularized solutions x_ϵ converge to the least-squares solution closest to z.

Suppose the system $Ax = b$ has infinitely many exact solutions. Our problem is to select one. Let us select z that incorporates features of the desired solution, to the extent that we know them *a priori*. Then, as $\epsilon \to 0$, the vectors x_ϵ converge to the exact solution closest to z. For example, taking $z = 0$ leads to the *minimum-norm solution*.

4.2.5 Minimizing Cross-Entropy

In image processing, it is common to encounter systems $Px = y$ in which all the terms are non-negative. In such cases, it may be desirable to solve the system $Px = y$, approximately, perhaps, by minimizing the *cross-entropy* or *Kullback-Leibler distance*

$$KL(y, Px) = \sum_{i=1}^{I} \Big(y_i \log \frac{y_i}{(Px)_i} + (Px)_i - y_i \Big), \tag{4.6}$$

over vectors $x \geq 0$. When the vector y is noisy, the resulting solution, viewed as an image, can be unacceptable. It is wise, therefore, to add a penalty term, such as $p(x) = \epsilon KL(z, x)$, where $z > 0$ is a prior estimate of the desired x [56, 74, 57, 20].

A similar problem involves minimizing the function $KL(Px, y)$. Once again, noisy results can be avoided by including a penalty term, such as $p(x) = \epsilon KL(x, z)$ [20].

4.2.6 The Lagrangian in Convex Programming

When there is a sensitivity vector λ for the CP problem, minimizing $f(x)$ is equivalent to minimizing the Lagrangian,

$$f(x) + \sum_{i=1}^{I} \lambda_i g_i(x) = f(x) + p(x); \tag{4.7}$$

in this case, the addition of the second term, $p(x)$, serves to incorporate the constraints $g_i(x) \leq 0$ in the function to be minimized, turning a constrained minimization problem into an unconstrained one. The problem of minimizing the Lagrangian still remains, though. We may have to solve that problem using an iterative algorithm.

4.2.7 Infimal Convolution

The *infimal convolution* of the functions f and g is defined as

$$(f \oplus g)(z) = \inf_x \Big\{ f(x) + g(z-x) \Big\}.$$

The *infimal deconvolution* of f and g is defined as

$$(f \ominus g)(z) = \sup_x \Big\{ f(z-x) - g(x) \Big\}.$$

4.2.8 Moreau's Proximity-Function Method

The Moreau envelope of the function f is the function

$$m_f(z) = \inf_x \Big\{ f(x) + \frac{1}{2}\|x-z\|_2^2 \Big\}, \tag{4.8}$$

which is also the *infimal convolution* of the functions $f(x)$ and $\frac{1}{2}\|x\|_2^2$. It can be shown that the infimum is uniquely attained at the point denoted $x = \text{prox}_f z$ (see [69]). In similar fashion, we can define $m_{f^*} z$ and $\text{prox}_{f^*} z$, where $f^*(z)$ denotes the function conjugate to f.

Let z be fixed and $\hat{x}$ minimize the function

$$f(x) + \frac{1}{2\gamma}\|x-z\|_2^2. \tag{4.9}$$

Then we have

$$0 \in \partial f(\hat{x}) + \frac{1}{\gamma}(\hat{x} - z),$$

or

$$z - \hat{x} \in \partial(\gamma f)(\hat{x}).$$

If $z - x \in \partial f(x)$ and $z - y \in \partial f(y)$, then $x = y$: we have

$$f(y) - f(x) \geq \langle z - x, y - x \rangle,$$

and

$$f(x) - f(y) \geq \langle z - y, x - y \rangle = -\langle z - y, y - x \rangle.$$

Adding, we get

$$0 \geq \langle y - x, y - x \rangle = \|x - y\|_2^2.$$

We can then say that $x = \text{prox}_f(z)$ is characterized by the inequality

$$z - x \in \partial f(x). \tag{4.10}$$

Consequently, we can write

$$\hat{x} = \text{prox}_{\gamma f}(z).$$

Proposition 4.1 *The infimum of $m_f(z)$, over all z, is the same as the infimum of $f(x)$, over all x.*

Proof: We have

$$\inf_z m_f(z) = \inf_z \inf_x \{f(x) + \frac{1}{2}\|x - z\|_2^2\}$$

$$= \inf_x \inf_z \{f(x) + \frac{1}{2}\|x - z\|_2^2\} = \inf_x \{f(x) + \frac{1}{2} \inf_z \|x - z\|_2^2\} = \inf_x f(x).$$

∎

The minimizers of $m_f(z)$ and $f(x)$ are the same, as well. Therefore, one way to use Moreau's method is to replace the original problem of minimizing the possibly non-smooth function $f(x)$ with the problem of minimizing the smooth function $m_f(z)$. Another way is to convert Moreau's method into a sequential minimization algorithm, replacing z with x^{k-1} and minimizing with respect to x to get x^k. This leads to the proximal minimization algorithm.

4.3 Basic Facts

Again, our objective is to find a sequence $\{x^k\}$ such that $\{f(x^k)\} \to d$. We select a penalty function $p(x)$ with $p(x) \geq 0$ and $p(x) = 0$ if and only if x is in C. For $k = 1, 2, ...$, let x^k be a minimizer of the function $P_k(x)$. As we shall see, we can formulate this penalty-function algorithm as a barrier-function iteration.

In order to relate penalty-function methods to barrier-function methods, we note that minimizing $P_k(x) = f(x) + kp(x)$ is equivalent to minimizing $p(x) + \frac{1}{k}f(x)$. This is the form of the barrier-function iteration, with $p(x)$ now in the role previously played by $f(x)$, and $f(x)$ now in the role previously played by $b(x)$. We are not concerned here with the effective domain of $f(x)$. Therefore, we can now mimic most, but not all, of what we did for barrier-function methods. We assume that there is a real α such that $\alpha \leq f(x)$, for all x in $\mathbb{R}^J$.

Lemma 4.1 *The sequence $\{P_k(x^k)\}$ is increasing, bounded above by d and converges to some $\gamma \leq d$.*

Proof: We have

$$P_k(x^k) \leq P_k(x^{k+1}) \leq P_k(x^{k+1}) + p(x^{k+1}) = P_{k+1}(x^{k+1}).$$

Also, for any $z \in C$, and for each k, we have

$$f(z) = f(z) + kp(z) = P_k(z) \geq P_k(x^k);$$

therefore $d \geq \gamma$. ∎

Lemma 4.2 *The sequence $\{p(x^k)\}$ is decreasing to zero, the sequence $\{f(x^k)\}$ is increasing and converging to some $\beta \leq d$.*

Proof: Since x^k minimizes $P_k(x)$ and x^{k+1} minimizes $P_{k+1}(x)$, we have

$$f(x^k) + kp(x^k) \leq f(x^{k+1}) + kp(x^{k+1}),$$

and

$$f(x^{k+1}) + (k+1)p(x^{k+1}) \leq f(x^k) + (k+1)p(x^k).$$

Consequently, we have

$$(k+1)[p(x^k) - p(x^{k+1})] \geq f(x^{k+1}) - f(x^k) \geq k[p(x^k) - p(x^{k+1})].$$

Therefore,

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

and

$$f(x^{k+1}) - f(x^k) \geq 0.$$

From

$$f(x^k) \leq f(x^k) + kp(x^k) = P_k(x^k) \leq \gamma \leq d,$$

it follows that the sequence $\{f(x^k)\}$ is increasing and converges to some $\beta \leq \gamma$. Since

$$\alpha + kp(x^k) \leq f(x^k) + kp(x^k) = P_k(x^k) \leq \gamma$$

for all k, we have $0 \leq kp(x^k) \leq \gamma - \alpha$. Therefore, the sequence $\{p(x^k)\}$ converges to zero. ∎

We want $\beta = d$. To obtain this result, it appears that we need to make more assumptions: we assume, therefore, that X is a complete metric space, C is closed in X, the functions f and p are continuous and f has compact level sets. From these assumptions, we are able to assert that the sequence $\{x^k\}$ is bounded, so that there is a convergent subsequence; let $\{x^{k_n}\} \to x^*$. It follows that $p(x^*) = 0$, so that x^* is in C. Then

$$f(x^*) = f(x^*) + p(x^*) = \lim_{n \to +\infty} (f(x^{k_n}) + p(x^{k_n})) \leq \lim_{n \to +\infty} P_{k_n}(x^{k_n}) = \gamma \leq d.$$

But $x^* \in C$, so $f(x^*) \geq d$. Therefore, $f(x^*) = d$.

It may seem odd that we are trying to minimize $f(x)$ over the set C using a sequence $\{x^k\}$ with $\{f(x^k)\}$ increasing, but remember that these x^k are not in C.

Definition 4.1 *Let X be a complete metric space. A real-valued function $p(x)$ on X has* compact level sets *if, for all real γ, the level set $\{x|p(x) \leq \gamma\}$ is compact.*

Theorem 4.1 *Let X be a complete metric space, $f(x)$ be a continuous function, and the restriction of $f(x)$ to x in C have compact level sets. Then the sequence $\{x^k\}$ is bounded and has convergent subsequences. Furthermore, $f(x^*) = d$, for any subsequential limit point $x^* \in X$. If $\hat{x}$ is the unique minimizer of $f(x)$ for $x \in C$, then $x^* = \hat{x}$ and $\{x^k\} \to \hat{x}$.*

Proof: From the previous theorem we have $f(x^*) = d$, for all subsequential limit points x^*. But, by uniqueness, $x^* = \hat{x}$, and so $\{x^k\} \to \hat{x}$. ∎

Corollary 4.1 *Let $C \subseteq \mathbb{R}^J$ be closed and convex. Let $f(x) : \mathbb{R}^J \to \mathbb{R}$ be closed, proper and convex. If $\hat{x}$ is the unique minimizer of $f(x)$ over $x \in C$, the sequence $\{x^k\}$ converges to $\hat{x}$.*

Proof: Let $\iota_C(x)$ be the indicator function of the set C, that is, $\iota_C(x) = 0$, for all x in C, and $\iota_C(x) = +\infty$, otherwise. Then the function $g(x) = f(x) + \iota_C(x)$ is closed, proper and convex. If $\hat{x}$ is unique, then we have

$$\{x|f(x) + \iota_C(x) \leq f(\hat{x})\} = \{\hat{x}\}.$$

Therefore, one of the level sets of $g(x)$ is bounded and nonempty. It follows from Corollary 8.7.1 of [69] that every level set of $g(x)$ is bounded, so that the sequence $\{x^k\}$ is bounded. ∎

If $\hat{x}$ is not unique, we can still prove convergence of the sequence $\{x^k\}$, for particular cases of SUMMA.

Chapter 5

Proximal Minimization

In this chapter we consider the use of Bregman distances in constrained optimization through the *proximal minimization* method. The *proximal minimization algorithm* (PMA) is in the SUMMA class and this fact is used to establish important properties of the PMA. A detailed discussion of the PMA and its history is found in the book by Censor and Zenios [40].

5.1 The Basic Problem

We want to minimize a convex function $f : \mathbb{R}^J \to \mathbb{R}$ over a closed, non-empty convex subset $C \subseteq \mathbb{R}^J$. If the problem is ill-conditioned in some way, perhaps because the function $f(x)$ is not strictly convex, then regularization is needed.

For example, the least-squares approximate solution of $Ax = b$ is obtained by minimizing the function $f(x) = \frac{1}{2}\|Ax - b\|_2^2$ over all x. When the matrix A is ill-conditioned the least-squares solution may have a large two-norm. To regularize the least-squares problem we can impose a norm constraint and minimize

$$\frac{1}{2}\|Ax - b\|_2^2 + \frac{\epsilon}{2}\|x\|_2^2, \tag{5.1}$$

where $\epsilon > 0$ is small.

Returning to our original problem, we can impose strict convexity and regularize by minimizing the function

$$f(x) + \frac{1}{2k}\|x - a\|_2^2 \tag{5.2}$$

to get x^k, for some selected vector a and $k = 1, 2, ...$ One difficulty with this approach is that, for small k, there may be too much emphasis on

the second term in Equation (5.2), while the problem becomes increasingly ill-conditioned as k increases. As pointed out in [40], one way out of this difficulty is to obtain x^k by minimizing

$$f(x) + \frac{\gamma}{2}\|x - x^{k-1}\|_2^2. \tag{5.3}$$

This suggests a more general technique for constrained optimization, called *proximal minimization* with D-functions in [40].

5.2 Proximal Minimization

Let $f : \mathbb{R}^J \to (-\infty, +\infty]$ be a closed, proper, and convex function. Let h be a closed proper convex function, with effective domain D, that is differentiable on the nonempty open convex set int D. Assume that $f(x)$ is finite on $C = \overline{D}$ and attains its minimum value on C at $\hat{x}$. The corresponding *Bregman distance* $D_h(x, z)$ is defined for x in D and z in int D by

$$D_h(x, z) = h(x) - h(z) - \langle \nabla h(z), x - z \rangle. \tag{5.4}$$

Note that $D_h(x, z) \geq 0$ always. If h is essentially strictly convex, then $D_h(x, z) = 0$ implies that $x = z$. Our objective is to minimize $f(x)$ over x in $C = \overline{D}$.

At the kth step of the *proximal minimization algorithm* (PMA) with D-functions [40, 26], we minimize the function

$$G_k(x) = f(x) + D_h(x, x^{k-1}), \tag{5.5}$$

to get x^k. The function

$$g_k(x) = D_h(x, x^{k-1}) \tag{5.6}$$

is nonnegative and $g_k(x^{k-1}) = 0$. We assume that each x^k lies in the interior of D. The *quadratic* PMA described in Equation (5.3) uses the function $h(x) = \frac{\gamma}{2}\|x\|_2^2$.

5.3 The PMA is in SUMMA

We show now that the PMA is a particular case of the SUMMA. We remind the reader that $f(x)$ is now assumed to be convex.

Lemma 5.1 *For each k we have*

$$G_k(x) - G_k(x^k) \geq D_h(x, x^k) = g_{k+1}(x). \tag{5.7}$$

Proof: Since x^k minimizes $G_k(x)$ within the set D, we have

$$0 \in \partial f(x^k) + \nabla h(x^k) - \nabla h(x^{k-1}), \tag{5.8}$$

so that

$$\nabla h(x^{k-1}) = u^k + \nabla h(x^k), \tag{5.9}$$

for some u^k in $\partial f(x^k)$. Then

$$G_k(x) - G_k(x^k) = f(x) - f(x^k) + h(x) - h(x^k) - \langle \nabla h(x^{k-1}), x - x^k \rangle.$$

Now substitute, using Equation (5.9), to get

$$G_k(x) - G_k(x^k) = f(x) - f(x^k) - \langle u^k, x - x^k \rangle + D_h(x, x^k). \tag{5.10}$$

Therefore,

$$G_k(x) - G_k(x^k) \geq D_h(x, x^k),$$

since u^k is in $\partial f(x^k)$. ∎

5.4 Convergence of the PMA

From the discussion of the SUMMA we know that $\{f(x^k)\}$ is monotonically decreasing to $f(\hat{x})$. As we noted previously, if the sequence $\{x^k\}$ is bounded, and $\hat{x}$ is unique, we can conclude that $\{x^k\} \to \hat{x}$.

Suppose that $\hat{x}$ is not known to be unique, but can be chosen in D; this will be the case, of course, whenever D is closed. Then $G_k(\hat{x})$ is finite for each k. From the definition of $G_k(x)$ we have

$$G_k(\hat{x}) = f(\hat{x}) + D_h(\hat{x}, x^{k-1}). \tag{5.11}$$

From Equation (5.10) we have

$$G_k(\hat{x}) = G_k(x^k) + f(\hat{x}) - f(x^k) - \langle u^k, \hat{x} - x^k \rangle + D_h(\hat{x}, x^k), \tag{5.12}$$

so that

$$G_k(\hat{x}) = f(x^k) + D_h(x^k, x^{k-1}) + f(\hat{x}) - f(x^k) - \langle u^k, \hat{x} - x^k \rangle + D_h(\hat{x}, x^k) \tag{5.13}$$

Therefore,

$$D_h(\hat{x}, x^{k-1}) - D_h(\hat{x}, x^k) =$$

$$f(x^k) - f(\hat{x}) + D_h(x^k, x^{k-1}) + f(\hat{x}) - f(x^k) - \langle u^k, \hat{x} - x^k \rangle. \tag{5.14}$$

It follows that the sequence $\{D_h(\hat{x}, x^k)\}$ is decreasing and that $\{f(x^k)\}$ converges to $f(\hat{x})$. If either the function $f(x)$ or the function $D_h(\hat{x}, \cdot)$ has

bounded level sets, then the sequence $\{x^k\}$ is bounded, has cluster points x^* in C, and $f(x^*) = f(\hat{x})$, for every x^*. We now show that $\hat{x}$ in D implies that x^* is also in D, whenever h is a Bregman -Legendre function (see Chapter **??**).

Let x^* be an arbitrary cluster point, with $\{x^{k_n}\} \to x^*$. If $\hat{x}$ is not in the interior of D, then, by Property B2 of Bregman-Legendre functions, we know that

$$D_h(x^*, x^{k_n}) \to 0,$$

so x^* is in D. Then the sequence $\{D_h(x^*, x^k)\}$ is decreasing. Since a subsequence converges to zero, we have $\{D_h(x^*, x^k)\} \to 0$. From Property R5, we conclude that $\{x^k\} \to x^*$.

If $\hat{x}$ is in int D, but x^* is not, then $\{D_h(\hat{x}, x^k)\} \to +\infty$, by Property R2. But, this is a contradiction; therefore x^* is in D. Once again, we conclude that $\{x^k\} \to x^*$.

Now we summarize our results for the PMA. Let $f : \mathbb{R}^J \to (-\infty, +\infty]$ be closed, proper, convex and differentiable. Let h be a closed proper convex function, with effective domain D, that is differentiable on the nonempty open convex set int D. Assume that $f(x)$ is finite on $C = \overline{D}$ and attains its minimum value on C at $\hat{x}$. For each positive integer k, let x^k minimize the function $f(x) + D_h(x, x^{k-1})$. Assume that each x^k is in the interior of D.

Theorem 5.1 *If the restriction of $f(x)$ to x in C has bounded level sets and $\hat{x}$ is unique, and then the sequence $\{x^k\}$ converges to $\hat{x}$.*

Theorem 5.2 *If $h(x)$ is a Bregman-Legendre function and $\hat{x}$ can be chosen in D, then $\{x^k\} \to x^*$, x^* in D, with $f(x^*) = f(\hat{x})$.*

5.5 The Newton-Raphson Algorithm

The Newton-Raphson algorithm for minimizing a function $f : \mathbb{R}^J \to \mathbb{R}$ has the iterative step

$$x^k = x^{k-1} - \nabla^2 f(x^{k-1})^{-1} \nabla f(x^{k-1}). \tag{5.15}$$

Suppose now that f is twice differentiable and convex. It is interesting to note that, having calculated x^{k-1}, we can obtain x^k by minimizing

$$G_k(x) = f(x) + (x - x^{k-1})^T \nabla^2 f(x^{k-1})(x - x^{k-1}) - D_f(x, x^{k-1}). \tag{5.16}$$

5.6 Another Job for the PMA

As we have seen, the original goal of the PMA is to minimize a convex function $f(x)$ over the closure of the domain of $h(x)$. Since the PMA is a

SUMMA algorithm, we know that, whenever the sequence converges, the limit x^* satisfies $f(x^*) = d$, where d is the finite infimum of $f(x)$ over x in the interior of the domain of h. This suggests another job for the PMA.

Consider the problem of minimizing a differentiable convex function $h : \mathbb{R}^J \to \mathbb{R}$ over all x for which $Ax = b$, where A is an M by N matrix with rank M and b is arbitrary. With

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2, \tag{5.17}$$

and x^0 arbitrary we minimize

$$f(x) + D_h(x, x^{k-1}) \tag{5.18}$$

to get x^k. Whenever the sequence $\{x^k\}$ converges to some x^* we have $Ax^* = b$. If $\nabla h(x^0)$ is in the range of A^T, then so is $\nabla h(x^*)$ and x^* minimizes $h(x)$ over all x with $Ax = b$.

5.7 The Goldstein-Osher Algorithm

In [50] Goldstein and Osher present a modified version of the PMA for the problem of minimizing $h(x)$ over all x with $Ax = b$. Instead of minimizing the function in Equation (5.18), they obtain the next iterate x^k by minimizing

$$\frac{1}{2}\|Ax - b^{k-1}\|_2^2 + h(x), \tag{5.19}$$

where $b^)$ is arbitrary and for $k = 2, 3, ...$ they define

$$b^{k-1} = b + b^{k-2} - Ax^{k-1}. \tag{5.20}$$

Assuming that AA^T is invertible, we have the following theorem, which is not in [50].

Theorem 5.3 *If the sequence $\{x^k\}$ converges to some x^*, and $Ax^* = b$, then the sequence $\{b^k\}$ converges to some b^* and x^* minimizes the function $h(x)$ over all x such that $Ax = b$.*

Proof: From

$$0 = A^T(Ax^k - b^{k-1}) + \nabla h(x^k)$$

we have

$$b^{k-1} = b + (AA^T)^{-1}A\nabla h(x^k),$$

and the right side converges to $b + (AA^T)^{-1}A\nabla h(x^*)$. Let z be such that $Az = b$. For each k we have

$$\frac{1}{2}\|Ax^k - b^{k-1}\|_2^2 + h(x^k) \leq \frac{1}{2}\|Az - b^{k-1}\|_2^2 + h(z).$$

Taking the limit as $k \to \infty$, we get

$$\frac{1}{2}\|b - b^*\|_2^2 + h(x^*) \leq \frac{1}{2}\|b - b^*\|_2^2 + h(z),$$

from which the assertion of the theorem follows immediately. ∎

In [50], the authors, hoping to rest their algorithm on the theoretical foundation of the PMA, claim that their modification of the PMA is equivalent to the PMA itself in this case; this is false, in general. For one thing, the Bregman distance D_h does not determine a unique h; for y arbitrary and $g(x) = D_h(x, y)$ we have $D_g = D_h$. For another,we have the theorem above for the Goldstein-Osher algorithm, while the x^* given by the PMA need not minimize h over all x with $Ax = b$. In order for the x^* given by the PMA to minimize $h(x)$ over $Ax = b$, we need $\nabla h(x^0)$ in the range of A^T. The only way in which the PMA can distinguish between h and g is through the selection of the initial x^0. In fact, the authors of [50] say nothing about the choice of x^0 or of b^0. What is true is this: if $b^0 = b + (AA^T)^{-1}A\nabla h(x^0)$ then the sequence $\{x^k\}$ is the same for both algorithms, so that convergence results for the PMA can be assumed for the Goldstein-Osher algorithm.

5.8 A Question about the PMA

We obtain x^k by minimizing

$$f(x) + D_h(x, x^{k-1}).$$

Let $D = \{x | h(x) < +\infty\}$. Suppose that the sequence $\{x^k\}$ converges to x^* in the closure of D. Since the algorithm is in the SUMMA class, we know that

$$f(x^*) = \min\{f(x) | x \in \overline{D}\}.$$

Let M be the set of all $z \in \overline{D}$ with $f(z) = f(x^*)$. The question is this: does x^* also minimize $h(z)$ over all $z \in M$? The answer is probably no, in general, since the Bregman distance D_h does not specify h uniquely. Suppose that $D_h = D_g$. The only way that the iterative sequence can distinguish between h and g is through the choice of x^0, so x^0 must play some role in the answer.

Suppose that $g(x) = D_g(x, x^0)$. Then $D_h = D_g$ if and only if h has the form

$$h(x) = D_g(x, p) + h(p),$$

for some p. For the case of

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

we know that x^* minimizes $h(z)$ over all z with $Az = b$ if and only if $\nabla h(x^*)$ is in the range of A^T, or equivalently, $\nabla h(x^0)$ is in the range of A^T. In this case, $\nabla g(x^0) = 0$, so clearly $\nabla g(x^0)$ is in the range of A^T. We also have

$$\nabla h(x^0) = -\nabla g(p),$$

so x^* will minimize $h(z)$ over $z \in M$ if and only if $\nabla g(p)$ is on the range of A^T.

A conjecture is: for the general case, x^* will minimize $h(z)$ over $z \in M$ provided that $h(x) = D_h(x, x^0)$. There are a number of examples, involving both the Euclidean and KL distances, for which the conjecture is true.

Chapter 6

An Interior-Point Algorithm- The IPA

6.1 The IPA

The IPA is a modification of the PMA designed to overcome some of the computational obstacles encountered in the PMA [26, 30]. At the kth step of the IPA we minimize

$$G_k(x) = f(x) + D_h(x, x^{k-1}) \tag{6.1}$$

over either $x \in \mathbb{R}^J$ or $x \in C$, where $h(x)$ is as in the previous section. We have selected $a(x)$ so that $h(x) = a(x) - f(x)$ is convex and differentiable, and the equation

$$\nabla a(x^k) = \nabla a(x^{k-1}) - \nabla f(x^{k-1}) \tag{6.2}$$

is easily solved. As we saw previously, the projected gradient descent algorithm is an example of the IPA. In this section we consider several other examples and some potential generalizations.

6.2 The Landweber and Projected Landweber Algorithms

The Landweber (LW) and projected Landweber (PLW) algorithms are IPA methods. The objective now is to minimize the function

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2, \tag{6.3}$$

over $x \in \mathbb{R}^J$ or $x \in C$, where A is a real I by J matrix. The gradient of $f(x)$ is

$$\nabla f(x) = A^T(Ax - b) \tag{6.4}$$

and is L-Lipschitz continuous for $L = \rho(A^T A)$, the largest eiqenvalue of $A^T A$. The Bregman distance associated with $f(x)$ is

$$D_f(x, z) = \|Ax - Az\|_2^2. \tag{6.5}$$

We let

$$a(x) = \frac{1}{2\gamma}\|x\|_2^2, \tag{6.6}$$

where $0 < \gamma < \frac{1}{L}$, so that the function $h(x) = a(x) - f(x)$ is convex.

At the kth step of the PLW we minimize

$$G_k(x) = f(x) + D_h(x, x^{k-1}) \tag{6.7}$$

over $x \in C$ to get

$$x^k = P_C(x^{k-1} - \gamma A^T(Ax^{k-1} - b)); \tag{6.8}$$

in the case of $C = \mathbb{R}^J$ we get the Landweber algorithm.

6.3 The Simultaneous MART

The simultaneous MART (SMART) minimizes the Kullback-Leibler distance $f(x) = KL(Px, y)$, where y is a positive vector, P is an I by J matrix with non-negative entries P_{ij} for which $s_j = \sum_{i=1}^I P_{ij} = 1$, for all j, and we seek a non-negative solution of the system $y = Px$.

The Bregman distance associated with the function $f(x) = KL(Px, y)$ is

$$D_f(x, z) = KL(Px, Pz). \tag{6.9}$$

We select $a(x)$ to be

$$a(x) = \sum_{j=1}^J x_j \log(x_j) - x_j. \tag{6.10}$$

It follows from the inequality in (1.14) that $h(x)$ is convex and

$$D_h(x, z) = KL(x, z) - KL(Px, Pz) \geq 0. \tag{6.11}$$

At the kth step of the SMART we minimize

$$G_k(x) = f(x) + D_h(x, x^{k-1}) =$$

$$KL(Px, y) + KL(x, x^{k-1}) - KL(Px, Px^{k-1}) \tag{6.12}$$

to get

$$x_j^k = x_j^{k-1} \exp\Big(\sum_{i=1}^{I} P_{ij} \log \frac{y_i}{(Px^{k-1})_i}\Big). \tag{6.13}$$

Chapter 7

The Forward-Backward Splitting Algorithm

7.1 The FBS Algorithm

The *forward-backward splitting* methods (FBS) [41, 33] form a broad class of SUMMA algorithms closely related the IPA. Note that minimizing $G_k(x)$ in Equation (6.1) over $x \in C$ is equivalent to minimizing

$$G_k(x) = \iota_C(x) + f(x) + D_h(x, x^{k-1}) \tag{7.1}$$

over all $x \in \mathbb{R}^J$, where $\iota_C(x) = 0$ for $x \in C$ and $\iota_C(x) = +\infty$ otherwise. This suggests a more general iterative algorithm, the FBS.

7.2 FBS as SUMMA

Suppose that we want to minimize the function $f_1(x) + f_2(x)$, where both functions are convex and $f_2(x)$ is differentiable with an L-Lipschitz continuous gradient. At the kth step of the FBS algorithm we obtain x^k by minimizing

$$G_k(x) = f_1(x) + f_2(x) + \frac{1}{2\gamma}\|x - x^{k-1}\|_2^2 - D_{f_2}(x, x^{k-1}), \tag{7.2}$$

over all $x \in \mathbb{R}^J$, where $0 < \gamma < \frac{1}{2\gamma}$. As we shall see,

$$G_k(x) - G_k(x^k) \geq \frac{1}{2\gamma}\|x - x^k\|_2^2 \geq g_{k+1}(x), \tag{7.3}$$

which shows that the FBS algorithm is in the SUMMA class.

7.3 Moreau's Proximity Operators

Let $f : \mathbb{R}^J \to \mathbb{R}$ be convex. For each $z \in \mathbb{R}^J$ the function

$$m_f(z) := \min_x \{f(x) + \frac{1}{2}\|x - z\|_2^2\} \tag{7.4}$$

is minimized by a unique x [69]. The operator that associates with each z the minimizing x is Moreau's proximity operator, and we write $x = \text{prox}_f(z)$. The operator prox_f extends the notion of orthogonal projection onto a closed convex set [62, 63, 64]. We have $x = \text{prox}_f(z)$ if and only if $z - x \in \partial f(x)$. Proximity operators are also firmly non-expansive [41]; indeed, the proximity operator prox_f is the resolvent of the maximal monotone operator $B(x) = \partial f(x)$ and all such resolvent operators are firmly non-expansive [10].

7.4 Convergence of the FBS Algorithm

Our objective here is to provide an elementary proof of convergence for the forward-backward splitting (FBS) algorithm; a detailed discussion of this algorithm and its history is given by Combettes and Wajs in [41].

Let $f : \mathbb{R}^J \to \mathbb{R}$ be convex, with $f = f_1 + f_2$, both convex, f_2 differentiable, and ∇f_2 L-Lipschitz continuous. The iterative step of the FBS algorithm is

$$x^k = \text{prox}_{\gamma f_1}\Big(x^{k-1} - \gamma \nabla f_2(x^{k-1})\Big). \tag{7.5}$$

As we shall show, convergence of the sequence $\{x^k\}$ to a solution can be established, if γ is chosen to lie within the interval $(0, 1/L]$.

Let $f : \mathbb{R}^J \to \mathbb{R}$ be convex, with $f = f_1 + f_2$, both convex, f_2 differentiable, and ∇f_2 L-Lipschitz continuous. Let $\{x^k\}$ be defined by Equation (7.5) and let $0 < \gamma \leq 1/L$.

For each $k = 1, 2, ...$ let

$$G_k(x) = f(x) + \frac{1}{2\gamma}\|x - x^{k-1}\|_2^2 - D_{f_2}(x, x^{k-1}), \tag{7.6}$$

where

$$D_{f_2}(x, x^{k-1}) = f_2(x) - f_2(x^{k-1}) - \langle \nabla f_2(x^{k-1}), x - x^{k-1} \rangle. \tag{7.7}$$

Since $f_2(x)$ is convex, $D_{f_2}(x, y) \geq 0$ for all x and y and is the Bregman distance formed from the function f_2 [9].

The auxiliary function

$$g_k(x) = \frac{1}{2\gamma}\|x - x^{k-1}\|_2^2 - D_{f_2}(x, x^{k-1}) \tag{7.8}$$

can be rewritten as

$$g_k(x) = D_h(x, x^{k-1}), \tag{7.9}$$

where

$$h(x) = \frac{1}{2\gamma}\|x\|_2^2 - f_2(x). \tag{7.10}$$

Therefore, $g_k(x) \geq 0$ whenever $h(x)$ is a convex function.

We know that $h(x)$ is convex if and only if

$$\langle \nabla h(x) - \nabla h(y), x - y \rangle \geq 0, \tag{7.11}$$

for all x and y. This is equivalent to

$$\frac{1}{\gamma}\|x - y\|_2^2 - \langle \nabla f_2(x) - \nabla f_2(y), x - y \rangle \geq 0. \tag{7.12}$$

Since ∇f_2 is L-Lipschitz, the inequality (7.12) holds for $0 < \gamma \leq 1/L$.

Lemma 7.1 *The x^k that minimizes $G_k(x)$ over x is given by Equation (7.5).*

Proof: We know that x^k minimizes $G_k(x)$ if and only if

$$0 \in \nabla f_2(x^k) + \frac{1}{\gamma}(x^k - x^{k-1}) - \nabla f_2(x^k) + \nabla f_2(x^{k-1}) + \partial f_1(x^k),$$

or, equivalently,

$$\left(x^{k-1} - \gamma \nabla f_2(x^{k-1})\right) - x^k \in \partial(\gamma f_1)(x^k).$$

Consequently,

$$x^k = \text{prox}_{\gamma f_1}(x^{k-1} - \gamma \nabla f_2(x^{k-1})).$$

∎

Theorem 7.1 *The sequence $\{x^k\}$ converges to a minimizer of the function $f(x)$, whenever minimizers exist.*

Proof: A relatively simple calculation shows that

$$G_k(x) - G_k(x^k) = \frac{1}{2\gamma}\|x - x^k\|_2^2 +$$

$$f_1(x) - f_1(x^k) - \frac{1}{\gamma}\langle (x^{k-1} - \gamma \nabla f_2(x^{k-1})) - x^k, x - x^k \rangle. \tag{7.13}$$

Since

$$(x^{k-1} - \gamma \nabla f_2(x^{k-1})) - x^k \in \partial(\gamma f_1)(x^k),$$

it follows that

$$f_1(x) - f_1(x^k) - \frac{1}{\gamma}\langle (x^{k-1} - \gamma \nabla f_2(x^{k-1})) - x^k, x - x^k \rangle \geq 0.$$

Therefore,

$$G_k(x) - G_k(x^k) \geq \frac{1}{2\gamma}\|x - x^k\|_2^2 \geq g_{k+1}(x). \tag{7.14}$$

Consequently, the inequality in (1.18) holds and the iteration fits into the SUMMA class.

Now let $\hat{x}$ minimize $f(x)$ over all x. Then

$$G_k(\hat{x}) - G_k(x^k) = f(\hat{x}) + g_k(\hat{x}) - f(x^k) - g_k(x^k)$$

$$\leq f(\hat{x}) + G_{k-1}(\hat{x}) - G_{k-1}(x^{k-1}) - f(x^k) - g_k(x^k),$$

so that

$$\Big(G_{k-1}(\hat{x}) - G_{k-1}(x^{k-1})\Big) - \Big(G_k(\hat{x}) - G_k(x^k)\Big) \geq f(x^k) - f(\hat{x}) + g_k(x^k) \geq 0.$$

Therefore, the sequence $\{G_k(\hat{x}) - G_k(x^k)\}$ is decreasing and the sequences $\{g_k(x^k)\}$ and $\{f(x^k) - f(\hat{x})\}$ converge to zero.

From

$$G_k(\hat{x}) - G_k(x^k) \geq \frac{1}{2\gamma}\|\hat{x} - x^k\|_2^2,$$

it follows that the sequence $\{x^k\}$ is bounded. Therefore, we may select a subsequence $\{x^{k_n}\}$ converging to some x^{**}, with $\{x^{k_n-1}\}$ converging to some x^*, and therefore $f(x^*) = f(x^{**}) = f(\hat{x})$.

Replacing the generic $\hat{x}$ with x^{**}, we find that $\{G_k(x^{**}) - G_k(x^k)\}$ is decreasing to zero. From the inequality in (7.14), we conclude that the sequence $\{\|x^* - x^k\|_2^2\}$ converges to zero, and so $\{x^k\}$ converges to x^*. This completes the proof of the theorem. ∎

7.5 Some Examples

We present some examples to illustrate the application of the convergence theorem.

7.5.1 Projected Gradient Descent

Let C be a non-empty, closed convex subset of $\mathbb{R}^J$ and $f_1(x) = \iota_C(x)$, the function that is $+\infty$ for x not in C and zero for x in C. Then $\iota_C(x)$ is convex, but not differentiable. We have $\text{prox}_{\gamma f_1} = P_C$, the orthogonal projection onto C. The iteration in Equation (7.5) becomes

$$x^k = P_C\Big(x^{k-1} - \gamma \nabla f_2(x^{k-1})\Big). \tag{7.15}$$

The sequence $\{x^k\}$ converges to a minimizer of f_2 over $x \in C$, whenever such minimizers exist, for $0 < \gamma \leq 1/L$.

7.5.2 The CQ Algorithm

Let A be a real I by J matrix, $C \subseteq \mathbb{R}^J$, and $Q \subseteq \mathbb{R}^I$, both closed convex sets. The split feasibility problem (SFP) is to find x in C such that Ax is in Q. The function

$$f_2(x) = \frac{1}{2}\|P_Q Ax - Ax\|_2^2 \tag{7.16}$$

is convex, differentiable and ∇f_2 is L-Lipschitz for $L = \rho(A^T A)$, the spectral radius of $A^T A$. The gradient of f_2 is

$$\nabla f_2(x) = A^T(I - P_Q)Ax. \tag{7.17}$$

We want to minimize the function $f_2(x)$ over x in C, or, equivalently, to minimize the function $f(x) = \iota_C(x) + f_2(x)$. The projected gradient descent algorithm has the iterative step

$$x^k = P_C\Big(x^{k-1} - \gamma A^T(I - P_Q)Ax^{k-1}\Big); \tag{7.18}$$

this iterative method was called the CQ-algorithm in [27, 28]. The sequence $\{x^k\}$ converges to a solution whenever f_2 has a minimum on the set C, for $0 < \gamma \leq 1/L$.

In [38, 37] the CQ algorithm was extended to a multiple-sets algorithm and applied to the design of protocols for intensity-modulated radiation therapy.

7.5.3 The Projected Landweber Algorithm

The problem is to minimize the function

$$f_2(x) = \frac{1}{2}\|Ax - b\|_2^2,$$

over $x \in C$. This is a special case of the SFP and we can use the CQ-algorithm, with $Q = \{b\}$. The resulting iteration is the projected Landweber algorithm [8]; when $C = \mathbb{R}^J$ it becomes the Landweber algorithm [55].

7.5.4 Minimizing f_2 over a Linear Manifold

Suppose that we want to minimize f_2 over x in the linear manifold $M = S + p$, where S is a subspace of $\mathbb{R}^J$ of dimension $I < J$ and p is a fixed vector. Let A be an I by J matrix such that the I columns of A^T form a basis for S. For each $z \in \mathbb{R}^I$ let

$$d(z) = f_2(A^T z + p),$$

so that d is convex, differentiable, and its gradient,

$$\nabla d(z) = A \nabla f_2(A^T z + p),$$

is K-Lipschitz continuous, for $K = \rho(A^T A)L$. The sequence $\{z^k\}$ defined by

$$z^k = z^{k-1} - \gamma \nabla d(z^{k-1}) \tag{7.19}$$

converges to a minimizer of d over all z in $\mathbb{R}^I$, whenever minimizers exist, for $0 < \gamma \leq 1/K$.

From Equation (7.19) we get

$$x^k = x^{k-1} - \gamma A^T A \nabla f_2(x^{k-1}), \tag{7.20}$$

with $x^k = A^T z^k + p$. The sequence $\{x^k\}$ converges to a minimizer of f_2 over all x in M.

Suppose now that we begin with an algorithm having the iterative step

$$x^k = x^{k-1} - \gamma A^T A \nabla f_2(x^{k-1}), \tag{7.21}$$

where A is any real I by J matrix having rank I. Let x^0 be in the range of A^T, so that $x^0 = A^T z^0$, for some $z^0 \in \mathbb{R}^I$. Then each $x^k = A^T z^k$ is again in the range of A^T, and we have

$$A^T z^k = A^T z^{k-1} - \gamma A^T A \nabla f_2(A^T z^{k-1}). \tag{7.22}$$

With $d(z) = f_2(A^T z)$, we can write Equation (7.22) as

$$A^T \Big(z^k - (z^{k-1} - \gamma \nabla d(z^{k-1})) \Big) = 0. \tag{7.23}$$

Since A has rank I, A^T is one-to-one, so that

$$z^k - z^{k-1} - \gamma \nabla d(z^{k-1}) = 0. \tag{7.24}$$

The sequence $\{z^k\}$ converges to a minimizer of d, over all $z \in \mathbb{R}^I$, whenever such minimizers exist, for $0 < \gamma \leq 1/K$. Therefore, the sequence $\{x^k\}$ converges to a minimizer of f_2 over all x in the range of A^T.

7.6 Feasible-Point Algorithms

Suppose that we want to minimize a convex differentiable function $f(x)$ over x such that $Ax = b$, where A is an I by J full-rank matrix, with $I < J$. If $Ax^k = b$ for each of the vectors $\{x^k\}$ generated by the iterative algorithm, we say that the algorithm is a feasible-point method.

7.6.1 The Projected Gradient Algorithm

Let C be the feasible set of all x in $\mathbb{R}^J$ such that $Ax = b$. For every z in $\mathbb{R}^J$, we have

$$P_C z = P_{NS(A)} z + A^T (AA^T)^{-1} b, \tag{7.25}$$

where $NS(A)$ is the null space of A. Using

$$P_{NS(A)} z = z - A^T (AA^T)^{-1} Az, \tag{7.26}$$

we have

$$P_C z = z + A^T (AA^T)^{-1} (b - Az). \tag{7.27}$$

Using Equation (7.5), we get the iteration step for the projected gradient algorithm:

$$x^k = x^{k-1} - \gamma P_{NS(A)} \nabla f(x^{k-1}), \tag{7.28}$$

which converges to a solution for $0 < \gamma \leq 1/L$, whenever solutions exist.

Next we present a somewhat simpler approach.

7.6.2 The Reduced Gradient Algorithm

Let x^0 be a feasible point, that is, $Ax^0 = b$. Then $x = x^0 + p$ is also feasible if p is in the null space of A, that is, $Ap = 0$. Let Z be a J by $J - I$ matrix whose columns form a basis for the null space of A. We want $p = Zv$ for some v. The best v will be the one for which the function

$$\phi(v) = f(x^0 + Zv)$$

is minimized. We can apply to the function $\phi(v)$ the steepest descent method, or the Newton-Raphson method, or any other minimization technique.

The steepest descent method, applied to $\phi(v)$, is called the reduced steepest descent algorithm [65]. The gradient of $\phi(v)$, also called the reduced gradient, is

$$\nabla \phi(v) = Z^T \nabla f(x),$$

where $x = x^0 + Zv$; the gradient operator $\nabla\phi$ is then K-Lipschitz, for $K = \rho(A^TA)L$.

Let x^0 be feasible. The iteration in Equation (7.5) now becomes

$$v^k = v^{k-1} - \gamma\nabla\phi(v^{k-1}), \tag{7.29}$$

so that the iteration for $x^k = x^0 + Zv^k$ is

$$x^k = x^{k-1} - \gamma ZZ^T\nabla f(x^{k-1}). \tag{7.30}$$

The vectors x^k are feasible and the sequence $\{x^k\}$ converges to a solution, whenever solutions exist, for any $0 < \gamma < \frac{1}{K}$.

7.6.3 The Reduced Newton-Raphson Method

The same idea can be applied to the Newton-Raphson method. The Newton-Raphson method, applied to $\phi(v)$, is called the reduced Newton-Raphson method [65]. The Hessian matrix of $\phi(v)$, also called the reduced Hessian matrix, is

$$\nabla^2\phi(v) = Z^T\nabla^2 f(c)Z,$$

so that the reduced Newton-Raphson iteration becomes

$$x^k = x^{k-1} - Z\Big(Z^T\nabla^2 f(x^{k-1})Z\Big)^{-1}Z^T\nabla f(x^{k-1}). \tag{7.31}$$

Let c^0 be feasible. Then each x^k is feasible. The sequence $\{x^k\}$ is not guaranteed to converge.

Chapter 8

The SMART and EMML Algorithms

In this chapter we discuss the simultaneous multiplicative algebraic reconstruction technique (SMART) and the expectation maximization maximum likelihood (EMML) algorithms.

8.1 The SMART Iteration

The SMART [44, 70, 39, 20, 21, 22] minimizes the function $f(x) = KL(Px, y)$, over nonnegative vectors x. Here y is a vector with positive entries, and P is a matrix with nonnegative entries, such that $s_j = \sum_{i=1}^{I} P_{ij} > 0$. Denote by $\mathcal{X}$ the set of all nonnegative x for which the vector Px has only positive entries.

Having found the vector x^{k-1}, the next vector in the SMART sequence is x^k, with entries given by

$$x_j^k = x_j^{k-1} \exp s_j^{-1} \Big(\sum_{i=1}^{I} P_{ij} \log(y_i/(Px^{k-1})_i) \Big). \tag{8.1}$$

8.2 The EMML Iteration

The EMML algorithm [43, 71, 56, 74, 57, 20, 21, 22] minimizes the function $f(x) = KL(y, Px)$, over nonnegative vectors x. Having found the vector x^{k-1}, the next vector in the EMML sequence is x^k, with entries given by

$$x_j^k = x_j^{k-1} s_j^{-1} \Big(\sum_{i=1}^{I} P_{ij} (y_i/(Px^{k-1})_i) \Big). \tag{8.2}$$

8.3 The EMML and the SMART as AM Methods

In [20] the SMART was derived using the following alternating minimization approach.

For each $x \in \mathcal{X}$, let $r(x)$ and $q(x)$ be the I by J arrays with entries

$$r(x)_{ij} = x_j P_{ij} y_i / (Px)_i, \tag{8.3}$$

and

$$q(x)_{ij} = x_j P_{ij}. \tag{8.4}$$

In the iterative step of the SMART we get x^k by minimizing the function

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

over $x \geq 0$. Note that $KL(Px, y) = KL(q(x), r(x))$.

Similarly, the iterative step of the EMML is to minimize the function $KL(r(x^{k-1}), q(x))$ to get $x = x^k$. Note that $KL(y, Px) = KL(r(x), q(x))$. It follows from the identities established in [20] that the SMART can also be formulated as a particular case of the SUMMA.

8.4 The SMART as a Case of SUMMA

We show now that the SMART is a particular case of the SUMMA. Lemma 1.14 is helpful here. For notational convenience, we assume, for the remainder of this section, that $s_j = 1$ for all j. From the identities established for the SMART in [20], we know that the iterative step of SMART can be expressed as follows: minimize the function

$$G_k(x) = KL(Px, y) + KL(x, x^{k-1}) - KL(Px, Px^{k-1}) \tag{8.5}$$

to get x^k. According to Lemma 1.14, the quantity

$$g_k(x) = KL(x, x^{k-1}) - KL(Px, Px^{k-1})$$

is nonnegative, since $s_j = 1$. The $g_k(x)$ are defined for all nonnegative x; that is, the set D is the closed nonnegative orthant in $\mathbb{R}^J$. Each x^k is a positive vector.

It was shown in [20] that

$$G_k(x) = G_k(x^k) + KL(x, x^k), \tag{8.6}$$

from which it follows immediately that the SMART is in the SUMMA class.

Because the SMART is a particular case of the SUMMA, we know that the sequence $\{f(x^k)\}$ is monotonically decreasing to $f(\hat{x})$. It was shown in [20] that if $y = Px$ has no nonnegative solution and the matrix P and every submatrix obtained from P by removing columns has full rank, then $\hat{x}$ is unique; in that case, the sequence $\{x^k\}$ converges to $\hat{x}$. As we shall see, the SMART sequence always converges to a nonnegative minimizer of $f(x)$. To establish this, we reformulate the SMART as a particular case of the PMA.

8.5 The SMART as a Case of the PMA

We take $F(x)$ to be the function

$$F(x) = \sum_{j=1}^{J} x_j \log x_j. \tag{8.7}$$

Then

$$D_F(x, z) = KL(x, z). \tag{8.8}$$

For nonnegative x and z in $\mathcal{X}$, we have

$$D_f(x, z) = KL(Px, Pz). \tag{8.9}$$

Lemma 8.1 $D_F(x, z) \geq D_f(x, z)$.

Proof: We have

$$\begin{aligned} D_F(x, z) &\geq \sum_{j=1}^{J} KL(x_j, z_j) \geq \sum_{j=1}^{J} \sum_{i=1}^{I} KL(P_{ij}x_j, P_{ij}z_j) \\ &\geq \sum_{i=1}^{I} KL((Px)_i, (Pz)_i) = KL(Px, Pz). \end{aligned} \tag{8.10}$$

∎

We let $h(x) = F(x) - f(x)$; then $D_h(x, z) \geq 0$ for nonnegative x and z in $\mathcal{X}$. The iterative step of the SMART is to minimize the function

$$f(x) + D_h(x, x^{k-1}). \tag{8.11}$$

So the SMART is a particular case of the PMA.

The function $h(x) = F(x) - f(x)$ is finite on D the nonnegative orthant of $\mathbb{R}^J$, and differentiable on the interior, so $C = D$ is closed in this example. Consequently, $\hat{x}$ is necessarily in D. From our earlier discussion of the

PMA, we can conclude that the sequence $\{D_h(\hat{x}, x^k)\}$ is decreasing and the sequence $\{D_f(\hat{x}, x^k)\} \to 0$. Since the function $KL(\hat{x}, \cdot)$ has bounded level sets, the sequence $\{x^k\}$ is bounded, and $f(x^*) = f(\hat{x})$, for every cluster point. Therefore, the sequence $\{D_h(x^*, x^k)\}$ is decreasing. Since a subsequence converges to zero, the entire sequence converges to zero. The convergence of $\{x^k\}$ to x^* follows from basic properties of the KL distance.

From the fact that $\{D_f(\hat{x}, x^k)\} \to 0$, we conclude that $P\hat{x} = Px^*$. Equation (5.14) now tells us that the difference $D_h(\hat{x}, x^{k-1}) - D_h(\hat{x}, x^k)$ depends on only on $P\hat{x}$, and not directly on $\hat{x}$. Therefore, the difference $D_h(\hat{x}, x^0) - D_h(\hat{x}, x^*)$ also depends only on $P\hat{x}$ and not directly on $\hat{x}$. Minimizing $D_h(\hat{x}, x^0)$ over nonnegative minimizers $\hat{x}$ of $f(x)$ is therefore equivalent to minimizing $D_h(\hat{x}, x^*)$ over the same vectors. But the solution to the latter problem is obviously $\hat{x} = x^*$. Thus we have shown that the limit of the SMART is the nonnegative minimizer of $KL(Px, y)$ for which the distance $KL(x, x^0)$ is minimized.

The following theorem summarizes the situation with regard to the SMART.

Theorem 8.1 *In the consistent case the SMART converges to the unique nonnegative solution of $y = Px$ 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 $KL(Px, y)$ for which $\sum_{j=1}^{J} s_j KL(x_j, x_j^0)$ is minimized; if P and every matrix derived from P by deleting columns has full rank then there is a unique nonnegative minimizer of $KL(Px, y)$ and at most $I - 1$ of its entries are nonzero.*

8.6 SMART and EMML as Projection Methods

For each $i = 1, 2, ..., I$, let H_i be the hyperplane

$$H_i = \{z | (Pz)_i = y_i\}. \tag{8.12}$$

The KL projection of a given positive x onto H_i is the z in H_i that minimizes the KL distance $KL(z, x)$. Generally, the KL projection onto H_i cannot be expressed in closed form. However, the z in H_i that minimizes the weighted KL distance

$$\sum_{j=1}^{J} P_{ij} KL(z_j, x_j) \tag{8.13}$$

is $T_i(x)$ given by

$$T_i(x)_j = x_j y_i / (Px)_i. \tag{8.14}$$

Both the SMART and the EMML can be described in terms of the T_i.

The iterative step of the SMART algorithm can be expressed as

$$x_j^k = \prod_{i=1}^{I} (T_i(x^{k-1})_j)^{P_{ij}}. \tag{8.15}$$

We see that x_j^k is a weighted geometric mean of the terms $T_i(x^{k-1})_j$.

The iterative step of the EMML algorithm can be expressed as

$$x_j^k = \sum_{i=1}^{I} P_{ij} T_i(x^{k-1})_j. \tag{8.16}$$

We see that x_j^k is a weighted arithmetic mean of the terms $T_i(x^{k-1})_j$, using the same weights as in the case of SMART.

8.7 The MART and EMART Algorithms

The MART algorithm has the iterative step

$$x_j^k = x_j^{k-1} (y_i / (Px^{k-1})_i)^{P_{ij} m_i^{-1}}, \tag{8.17}$$

where $i = (k-1) (\mathrm{mod}\, I) + 1$ and

$$m_i = \max\{P_{ij} | j = 1, 2, ..., J\}. \tag{8.18}$$

When there are non-negative solutions of the system $y = Px$, the sequence $\{x^k\}$ converges to the solution x that minimizes $KL(x, x^0)$ [23, 24, 25]. We can express the MART in terms of the weighted KL projections $T_i(x^{k-1})$;

$$x_j^k = (x_j^{k-1})^{1 - P_{ij} m_i^{-1}} (T_i(x^{k-1})_j)^{P_{ij} m_i^{-1}}. \tag{8.19}$$

We see then that the iterative step of the MART is a relaxed weighted KL projection onto H_i, and a weighted geometric mean of the current x_j^k and $T_i(x^{k-1})_j$. The expression for the MART in Equation (8.19) suggests a somewhat simpler iterative algorithm involving a weighted arithmetic mean of the current x_j^{k-1} and $T_i(x^{k-1})_j$; this is the EMART algorithm.

The iterative step of the EMART algorithm is

$$x_j^k = (1 - P_{ij} m_i^{-1}) x_j^{k-1} + P_{ij} m_i^{-1} T_i(x^{k-1})_j. \tag{8.20}$$

Whenever the system $y = Px$ has non-negative solutions, the EMART sequence $\{x^k\}$ converges to a non-negative solution, but nothing further is known about this solution. One advantage that the EMART has over the MART is the substitution of multiplication for exponentiation.

Block-iterative versions of SMART and EMML have also been investigated; see [23, 24, 25] and the references therein.

8.8 Possible Extensions of MART and EMART

As we have seen, the iterative steps of the MART and the EMART are relaxed weighted KL projections onto the hyperplane H_i, resulting in vectors that are not within H_i. This suggests variants of MART and EMART in which, at the end of each iterative step, a further weighted KL projection onto H_i is performed. In other words, for MART and EMART the new vector would be $T_i(x^k)$, instead of x^k as given by Equations (8.17) and (8.20), respectively. Research into the properties of these new algorithms is ongoing.

Chapter 9

Regularization Methods

9.1 The Issue of Sensitivity to Noise

Many of the algorithms we have discussed here are used to solve for exact or approximate solutions of large systems of linear equations, with or without additional constraints. It is often the case, particularly in remote-sensing applications, that the vector b in the system $Ax = b$ is obtained through measurements and its entries are noisy. It is also frequently the case that the matrix A describing the relationship between the measurements and the desired x is a simplification of the actual physical situation. The result is that an exact solution of $Ax = b$ may not be desirable, even when exact solutions exist. When additional constraints, such as the positivity of x, are imposed, otherwise consistent systems $Ax = b$ may become inconsistent, requiring an approximate solution. Regularization is a general method for reducing the sensitivity of the answer to noise and model error in the system.

9.2 Non-Negatively Constrained Least-Squares

If there is no solution to a system of linear equations $Ax = b$, then we may seek a *least-squares* "solution" , which is a minimizer of the function

$$f(x) = \frac{1}{2}\sum_{i=1}^{I}\Big((\sum_{m=1}^{J} A_{im}x_m) - b_i\Big)^2 = \frac{1}{2}||Ax - b||^2. \tag{9.1}$$

The partial derivative of $f(x)$ with respect to the variable x_j is

$$\frac{\partial f}{\partial x_j}(x) = \sum_{i=1}^{I} A_{ij}\Big((\sum_{m=1}^{J} A_{im}x_m) - b_i\Big). \tag{9.2}$$

Setting the gradient equal to zero, we find that to get a least-squares solution we must solve the system of equations

$$A^T(Ax - b) = 0. \tag{9.3}$$

Now we consider what happens when the additional constraints $x_j \geq 0$ are imposed.

This problem becomes a convex programming problem. Let $\hat{x}$ be a non-negatively constrained least-squares solution. According to the Karush-Kuhn-Tucker Theorem, for those values of j for which $\hat{x}_j$ is not zero the corresponding Lagrange multiplier is $\lambda_j^* = 0$ and $\frac{\partial f}{\partial x_j}(\hat{x}) = 0$. Therefore, if $\hat{x}_j \neq 0$,

$$0 = \sum_{i=1}^{I} A_{ij}\Big((\sum_{m=1}^{J} A_{im}\hat{x}_m) - b_i\Big). \tag{9.4}$$

Let Q be the I by K matrix obtained from A by deleting rows j for which $\hat{x}_j = 0$. Then we can write

$$Q^T(A\hat{x} - b) = 0. \tag{9.5}$$

If Q has $K \geq I$ columns and has full rank, then Q^T is a one-to-one linear transformation, which implies that $A\hat{x} = b$. Therefore, when there is no non-negative solution of $Ax = b$, and Q has full rank, which is the typical case, the Q must have fewer than I columns, which means that $\hat{x}$ has fewer than I non-zero entries.

This result has some practical implications in medical image reconstruction. In the hope of improving the resolution of the reconstructed image, we may be tempted to take J, the number of pixels, larger than I, the number of equations arising from photon counts or line integrals. Since the vector b consists of measured data, it is noisy and there may well not be a non-negative solution of $Ax = b$. As a result, the image obtained by non-negatively constrained least-squares will have at most $I - 1$ non-zero entries; many of the pixels will be zero and they will be scattered throughout the image, making it unusable for diagnosis. The reconstructed images resemble stars in a night sky, and, as a result, the theorem is sometimes described as the "night sky" theorem.

This "night sky" phenomenon is not restricted to least squares. The same thing happens with methods based on the Kullback-Leibler distance, such as MART, EMML and SMART.

9.3 The EMML Algorithm

As we saw previously, the sequence $\{x^k\}$ generated by the EMML iterative step in Equation (8.2) converges to a non-negative minimizer $\hat{x}$ of $f(x) =$

$KL(y, Px)$, and we have

$$\hat{x}_j = \hat{x}_j s_j^{-1} \sum_{i=1}^{I} P_{ij} \frac{y_i}{(P\hat{x})_i}, \tag{9.6}$$

for all j. We consider what happens when there is no non-negative solution of the system $y = Px$.

For those values of j for which $\hat{x}_j > 0$, we have

$$s_j = \sum_{i=1}^{I} P_{ij} = \sum_{i=1}^{I} P_{ij} \frac{y_i}{(P\hat{x})_i}. \tag{9.7}$$

Now let Q be the I by K matrix obtained from P by deleting rows j for which $\hat{x}_j = 0$. If Q has full rank and $K \geq I$, then Q^T is one-to-one, so that $1 = \frac{y_i}{(P\hat{x})_i}$ for all i, or $y = P\hat{x}$. But we are assuming that there is no non-negative solution of $y = Px$. Consequently, we must have $K < I$ and $I - K$ of the entries of $\hat{x}$ are zero.

9.4 Norm-Constrained Least-Squares

One way to regularize the least-squares problem is to minimize not $\|b - Ax\|_2$, but, say,

$$f(x) = \|b - Ax\|_2^2 + \epsilon^2 \|x\|_2^2, \tag{9.8}$$

for some small $\epsilon > 0$. Now we are still trying to make $\|b - Ax\|_2$ small, but managing to keep $\|x\|_2$ from becoming too large in the process. This leads to a *norm-constrained least-squares* solution.

The minimizer of $f(x)$ is the unique solution $\hat{x}_\epsilon$ of the system

$$(A^T A + \epsilon^2 I)x = A^T b. \tag{9.9}$$

When I and J are large, we need ways to solve this system without having to deal with the matrix $A^T A + \epsilon^2 I$. The Landweber method allows us to avoid $A^T A$ in calculating the least-squares solution. Is there a similar method to use now? Yes, there is.

9.5 Regularizing Landweber's Algorithm

Our goal is to minimize the function $f(x)$ in Equation (9.8). Notice that this is equivalent to minimizing the function

$$F(x) = ||Bx - c||_2^2, \tag{9.10}$$

for

$$B = \begin{bmatrix} A \\ \epsilon I \end{bmatrix}, \tag{9.11}$$

and

$$c = \begin{bmatrix} b \\ 0 \end{bmatrix}, \tag{9.12}$$

where 0 denotes a column vector with all entries equal to zero. The Landweber iteration for the problem $Bx = c$ is

$$x^{k+1} = x^k + \alpha B^T(c - Bx^k), \tag{9.13}$$

for $0 < \alpha < 2/\rho(B^T B)$, where $\rho(B^T B)$ is the largest eigenvalue, or the spectral radius, of $B^T B$. Equation (9.13) can be written as

$$x^{k+1} = (1 - \alpha\epsilon^2)x^k + \alpha A^T(b - Ax^k). \tag{9.14}$$

9.6 Regularizing the ART

We would like to get the regularized solution $\hat{x}_\epsilon$ by taking advantage of the faster convergence of the ART. 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 [29], while the second one is due to Eggermont, Herman, and Lent [45].

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

$$\begin{bmatrix} A^T & \epsilon I \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = 0. \tag{9.15}$$

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

The method of Eggermont *et al.* is similar. In their method we use ART to solve the system of equations given in matrix form by

$$\begin{bmatrix} A & \epsilon I \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} = b. \tag{9.16}$$

We begin at $x^0 = 0$ and $v^0 = 0$. Then, the limit vector has for its upper component $x^\infty = \hat{x}_\epsilon$, and $\epsilon v^\infty = b - A\hat{x}_\epsilon$.

9.7 Regularizing SMART and EMML

We can regularize the SMART by minimizing

$$f(x) = (1-\alpha)KL(Px, y) + \alpha KL(x, p), \tag{9.17}$$

where p is a positive prior estimate of the desired answer, and α is in the interval $(0, 1)$. The iterative step of the regularized SMART is

$$x_j^k = \Big(x_j^{k-1} \exp\Big(s_j^{-1} \sum_{i=1}^{I} P_{ij} \log(y_i/(Px^{k-1})_i)\Big)\Big)^{1-\alpha} p_j^{\alpha}. \tag{9.18}$$

Similarly, we regularize the EMML algorithm by minimizing

$$f(x) = (1-\alpha)KL(y, Px) + \alpha KL(p, x). \tag{9.19}$$

The iterative step of the regularized EMML is

$$x_j^k = (1-\alpha)x_j^{k-1} s_j^{-1}\Big(\sum_{i=1}^{I} P_{ij}(y_i/(Px^{k-1})_i)\Big) + \alpha p_j. \tag{9.20}$$

Chapter 10

Alternating Minimization

As we have seen, the SMART is best derived as an alternating minimization (AM) algorithm. The main reference for alternating minimization is the paper [42] of Csiszár and Tusnády. As the authors of [74] remark, the geometric argument in [42] is "deep, though hard to follow". As we shall see, all AM methods for which the five-point property of [42] holds fall into the SUMMA class (see [32]).

10.1 Alternating Minimization

The alternating minimization approach provides a useful framework for the derivation of iterative optimization algorithms. As we shall see, convergence of an AM algorithm can be established, provided that the *five-point property* of [42] holds. In this section we discuss this five-point property and use it to obtain a somewhat simpler proof of convergence of AM algorithms. We then show that all AM algorithms with the five-point property are in the SUMMA class.

10.1.1 The AM Framework

Suppose that P and Q are arbitrary non-empty sets and the function $\Theta(p,q)$ satisfies $-\infty < \Theta(p,q) \leq +\infty$, for each $p \in P$ and $q \in Q$. We assume that, for each $p \in P$, there is $q \in Q$ with $\Theta(p,q) < +\infty$. Therefore, $b = \inf_{p \in P,\, q \in Q} \Theta(p,q) < +\infty$. We assume also that $b > -\infty$; in many applications, the function $\Theta(p,q)$ is non-negative, so this additional assumption is unnecessary. We do not always assume there are $\hat{p} \in P$ and $\hat{q} \in Q$ such that $\Theta(\hat{p},\hat{q}) = b$; when we do assume that such a $\hat{p}$ and $\hat{q}$ exist, we will not assume that $\hat{p}$ and $\hat{q}$ are unique with that property. The objective is to generate a sequence $\{(p^n, q^n)\}$ such that $\Theta(p^n, q^n) \to b$.

10.1.2 The AM Iteration

The general AM method proceeds in two steps: we begin with some q^0, and, having found q^n, we

- **1.** minimize $\Theta(p, q^n)$ over $p \in P$ to get $p = p^{n+1}$, and then
- **2.** minimize $\Theta(p^{n+1}, q)$ over $q \in Q$ to get $q = q^{n+1}$.

In certain applications we consider the special case of alternating cross-entropy minimization. In that case, the vectors p and q are non-negative, and the function $\Theta(p, q)$ will have the value $+\infty$ whenever there is an index j such that $p_j > 0$, but $q_j = 0$. It is important for those particular applications that we select q^0 with all positive entries. We therefore assume, for the general case, that we have selected q^0 so that $\Theta(p, q^0)$ is finite for all p.

The sequence $\{\Theta(p^n, q^n)\}$ is decreasing and bounded below by b, since we have

$$\Theta(p^n, q^n) \geq \Theta(p^{n+1}, q^n) \geq \Theta(p^{n+1}, q^{n+1}). \tag{10.1}$$

Therefore, the sequence $\{\Theta(p^n, q^n)\}$ converges to some $B \geq b$. Without additional assumptions, we can say little more.

We know two things:

$$\Theta(p^{n+1}, q^n) - \Theta(p^{n+1}, q^{n+1}) \geq 0, \tag{10.2}$$

and

$$\Theta(p^n, q^n) - \Theta(p^{n+1}, q^n) \geq 0. \tag{10.3}$$

Equation 10.3 can be strengthened to

$$\Theta(p, q^n) - \Theta(p^{n+1}, q^n) \geq 0. \tag{10.4}$$

We need to make these inequalities more precise.

10.1.3 The Five-Point Property for AM

The five-point property is the following: for all $p \in P$ and $q \in Q$ and $n = 1, 2, ...$

The Five-Point Property

$$\Theta(p, q) + \Theta(p, q^{n-1}) \geq \Theta(p, q^n) + \Theta(p^n, q^{n-1}). \tag{10.5}$$

10.1.4 The Main Theorem for AM

We want to find sufficient conditions for the sequence $\{\Theta(p^n, q^n)\}$ to converge to b, that is, for $B = b$. The following is the main result of [42].

Theorem 10.1 *If the five-point property holds then $B = b$.*

Proof: Suppose that $B > b$. Then there are p' and q' such that $B > \Theta(p', q') \geq b$. From the five-point property we have

$$\Theta(p', q^{n-1}) - \Theta(p^n, q^{n-1}) \geq \Theta(p', q^n) - \Theta(p', q'), \tag{10.6}$$

so that

$$\Theta(p', q^{n-1}) - \Theta(p', q^n) \geq \Theta(p^n, q^{n-1}) - \Theta(p', q') \geq 0. \tag{10.7}$$

All the terms being subtracted can be shown to be finite. It follows that the sequence $\{\Theta(p', q^{n-1})\}$ is decreasing, bounded below, and therefore convergent. The right side of Equation (10.7) must therefore converge to zero, which is a contradiction. We conclude that $B = b$ whenever the five-point property holds in AM. ∎

10.1.5 The Three- and Four-Point Properties

In [42] the five-point property is related to two other properties, the three- and four-point properties. This is a bit peculiar for two reasons: first, as we have just seen, the five-point property is sufficient to prove the main theorem; and second, these other properties involve a second function, $\Delta : P \times P \to [0, +\infty]$, with $\Delta(p, p) = 0$ for all $p \in P$. The three- and four-point properties jointly imply the five-point property, but to get the converse, we need to use the five-point property to define this second function; it can be done, however.

The three-point property is the following:

The Three-Point Property

$$\Theta(p, q^n) - \Theta(p^{n+1}, q^n) \geq \Delta(p, p^{n+1}), \tag{10.8}$$

for all p. The four-point property is the following:

The Four-Point Property

$$\Delta(p, p^{n+1}) + \Theta(p, q) \geq \Theta(p, q^{n+1}), \tag{10.9}$$

for all p and q.

It is clear that the three- and four-point properties together imply the five-point property. We show now that the three-point property and the

four-point property are implied by the five-point property. For that purpose we need to define a suitable $\Delta(p, \tilde{p})$. For any p and $\tilde{p}$ in P define

$$\Delta(p, \tilde{p}) = \Theta(p, q(\tilde{p})) - \Theta(p, q(p)), \tag{10.10}$$

where $q(p)$ denotes a member of Q satisfying $\Theta(p, q(p)) \leq \Theta(p, q)$, for all q in Q. Clearly, $\Delta(p, \tilde{p}) \geq 0$ and $\Delta(p, p) = 0$. The four-point property holds automatically from this definition, while the three-point property follows from the five-point property. Therefore, it is sufficient to discuss only the five-point property when speaking of the AM method.

10.2 Alternating Bregman Distance Minimization

The general problem of minimizing $\Theta(p, q)$ is simply a minimization of a real-valued function of two variables, $p \in P$ and $q \in Q$. In many cases the function $\Theta(p, q)$ is a distance between p and q, either $\|p - q\|_2^2$ or $KL(p, q)$. In the case of $\Theta(p, q) = \|p - q\|_2^2$, each step of the alternating minimization algorithm involves an orthogonal projection onto a closed convex set; both projections are with respect to the same Euclidean distance function. In the case of cross-entropy minimization, we first project q^n onto the set P by minimizing the distance $KL(p, q^n)$ over all $p \in P$, and then project p^{n+1} onto the set Q by minimizing the distance function $KL(p^{n+1}, q)$. This suggests the possibility of using alternating minimization with respect to more general distance functions. We shall focus on Bregman distances.

10.2.1 Bregman Distances

Let $f : \mathbb{R}^J \to \mathbb{R}$ be a Bregman function [9, 40, 12], and so $f(x)$ is convex on its domain and differentiable in the interior of its domain. Then, for x in the domain and z in the interior, we define the Bregman distance $D_f(x, z)$ by

$$D_f(x, z) = f(x) - f(z) - \langle \nabla f(z), x - z \rangle. \tag{10.11}$$

For example, the KL distance is a Bregman distance with associated Bregman function

$$f(x) = \sum_{j=1}^{J} x_j \log x_j \, - x_j. \tag{10.12}$$

Suppose now that $f(x)$ is a Bregman function and P and Q are closed convex subsets of the interior of the domain of $f(x)$. Let p^{n+1} minimize $D_f(p, q^n)$ over all $p \in P$. It follows then that

$$\langle \nabla f(p^{n+1}) - \nabla f(q^n), p - p^{n+1} \rangle \geq 0, \tag{10.13}$$

for all $p \in P$. Since

$$D_f(p, q^n) - D_f(p^{n+1}, q^n) =$$

$$D_f(p, p^{n+1}) + \langle \nabla f(p^{n+1}) - \nabla f(q^n), p - p^{n+1} \rangle, \tag{10.14}$$

it follows that the three-point property holds, with

$$\Theta(p, q) = D_f(p, q), \tag{10.15}$$

and

$$\Delta(p, \hat{p}) = D_f(p, \tilde{p}). \tag{10.16}$$

To get the four-point property we need to restrict D_f somewhat; we assume from now on that $D_f(p, q)$ is jointly convex, that is, it is convex in the combined vector variable (p, q) (see [4]). Now we can invoke a lemma due to Eggermont and LaRiccia [46].

10.2.2 The Eggermont-LaRiccia Lemma

Lemma 10.1 *Suppose that the Bregman distance $D_f(p, q)$ is jointly convex. Then it has the four-point property.*

Proof: By joint convexity we have

$$D_f(p, q) - D_f(p^n, q^n) \geq$$

$$\langle \nabla_1 D_f(p^n, q^n), p - p^n \rangle + \langle \nabla_2 D_f(p^n, q^n), q - q^n \rangle,$$

where ∇_1 denotes the gradient with respect to the first vector variable. Since q^n minimizes $D_f(p^n, q)$ over all $q \in Q$, we have

$$\langle \nabla_2 D_f(p^n, q^n), q - q^n \rangle \geq 0,$$

for all q. Also,

$$\langle \nabla_1(p^n, q^n), p - p^n \rangle = \langle \nabla f(p^n) - \nabla f(q^n), p - p^n \rangle.$$

It follows that

$$D_f(p, q^n) - D_f(p, p^n) = D_f(p^n, q^n) + \langle \nabla_1(p^n, q^n), p - p^n \rangle$$

$$\leq D_f(p, q) - \langle \nabla_2 D_f(p^n, q^n), q - q^n \rangle \leq D_f(p, q).$$

Therefore, we have

$$D_f(p, p^n) + D_f(p, q) \geq D_f(p, q^n).$$

This is the four-point property. ∎

We now know that the alternating minimization method works for any Bregman distance that is jointly convex. This includes the Euclidean and the KL distances [4].

10.3 Minimizing a Proximity Function

We present now an example of alternating Bregman distance minimization taken from [34]. The problem is the *convex feasibility problem* (CFP), to find a member of the intersection $C \subseteq \mathbb{R}^J$ of finitely many closed convex sets C_i, $i = 1, ..., I$, or, failing that, to minimize the proximity function

$$F(x) = \sum_{i=1}^{I} D_i(\overleftarrow{P}_i x, x), \tag{10.17}$$

where f_i are Bregman functions for which D_i, the associated Bregman distance, is jointly convex, and $\overleftarrow{P}_i x$ are the *left* Bregman projection of x onto the set C_i, that is, $\overleftarrow{P}_i x \in C_i$ and $D_i(\overleftarrow{P}_i x, x) \leq D_i(z, x)$, for all $z \in C_i$. Because each D_i is jointly convex, the function $F(x)$ is convex.

The problem can be formulated as an alternating minimization, where $P \subseteq \mathbb{R}^{IJ}$ is the product set $P = C_1 \times C_2 \times ... \times C_I$. A typical member of P has the form $p = (c^1, c^2, ..., c^I)$, where $c^i \in C_i$, and $Q \subseteq \mathbb{R}^{IJ}$ is the *diagonal* subset, meaning that the elements of Q are the I-fold product of a single x; that is $Q = \{d(x) = (x, x, ..., x) \in \mathbb{R}^{IJ}\}$. We then take

$$\Theta(p, q) = \sum_{i=1}^{I} D_i(c^i, x), \tag{10.18}$$

and $\Delta(p, \tilde{p}) = \Theta(p, \tilde{p})$.

In [36] a similar iterative algorithm was developed for solving the CFP, using the same sets P and Q, but using alternating projection, rather than alternating minimization. Now it is not necessary that the Bregman distances be jointly convex. Each iteration of their algorithm involves two steps:

- 1. minimize $\sum_{i=1}^{I} D_i(c^i, x^n)$ over $c^i \in C_i$, obtaining $c^i = \overleftarrow{P}_i x^n$, and then
- 2. minimize $\sum_{i=1}^{I} D_i(x, \overleftarrow{P}_i x^n)$.

Because this method is an alternating projection approach, it converges only when the CFP has a solution, whereas the previous alternating minimization method minimizes $F(x)$, even when the CFP has no solution.

10.4 Right and Left Projections

Because Bregman distances D_f are not generally symmetric, we can speak of *right* and *left* Bregman projections onto a closed convex set. For any allowable vector x, the *left* Bregman projection of x onto C, if it exists, is

the vector $\overleftarrow{P}_C x \in C$ satisfying the inequality $D_f(\overleftarrow{P}_C x, x) \leq D_f(c, x)$, for all $c \in C$. Similarly, the *right* Bregman projection is the vector $\overrightarrow{P}_C x \in C$ satisfying the inequality $D_f(x, \overrightarrow{P}_C x) \leq D_f(x, c)$, for any $c \in C$.

The alternating minimization approach described above to minimize the proximity function

$$F(x) = \sum_{i=1}^{I} D_i(\overleftarrow{P}_i x, x) \tag{10.19}$$

can be viewed as an alternating projection method, but employing both right and left Bregman projections.

Consider the problem of finding a member of the intersection of two closed convex sets C and D. We could proceed as follows: having found x^n, minimize $D_f(x^n, d)$ over all $d \in D$, obtaining $d = \overrightarrow{P}_D x^n$, and then minimize $D_f(c, \overrightarrow{P}_D x^n)$ over all $c \in C$, obtaining $c = x^{n+1} = \overleftarrow{P}_C \overrightarrow{P}_D x^n$. The objective of this algorithm is to minimize $D_f(c, d)$ over all $c \in C$ and $d \in D$; such a minimum may not exist, of course.

In [5] the authors note that the alternating minimization algorithm of [34] involves right and left Bregman projections, which suggests to them iterative methods involving a wider class of operators that they call "Bregman retractions".

10.5 More Proximity Function Minimization

Proximity function minimization and right and left Bregman projections play a role in a variety of iterative algorithms. We survey several of them in this section.

10.5.1 Cimmino's Algorithm

Our objective here is to find an exact or approximate solution of the system of I linear equations in J unknowns, written $Ax = b$. For each i let

$$C_i = \{z | (Az)_i = b_i\}, \tag{10.20}$$

and $P_i x$ be the orthogonal projection of x onto C_i. Then

$$(P_i x)_j = x_j + \alpha_i A_{ij}(b_i - (Ax)_i), \tag{10.21}$$

where

$$(\alpha_i)^{-1} = \sum_{j=1}^{J} A_{ij}^2. \tag{10.22}$$

Let

$$F(x) = \sum_{i=1}^{I} \|P_i x - x\|_2^2. \tag{10.23}$$

Using alternating minimization on this proximity function gives Cimmino's algorithm, with the iterative step

$$x_j^k = x_j^{k-1} + \frac{1}{I} \sum_{i=1}^{I} \alpha_i A_{ij} (b_i - (Ax^{k-1})_i). \tag{10.24}$$

10.5.2 Simultaneous Projection for Convex Feasibility

Now we let C_i be any closed convex subsets of $\mathbb{R}^J$ and define $F(x)$ as in the previous section. Again, we apply alternating minimization. The iterative step of the resulting algorithm is

$$x^k = \frac{1}{I} \sum_{i=1}^{I} P_i x^{k-1}. \tag{10.25}$$

The objective here is to minimize $F(x)$, if there is a minimum.

10.5.3 The Bauschke-Combettes-Noll Problem

In [6] Bauschke, Combettes and Noll consider the following problem: minimize the function

$$\Theta(p, q) = \Lambda(p, q) = \phi(p) + \psi(q) + D_f(p, q), \tag{10.26}$$

where ϕ and ψ are convex on $\mathbb{R}^J$, $D = D_f$ is a Bregman distance, and $P = Q$ is the interior of the domain of f. They assume that

$$b = \inf_{(p,q)} \Lambda(p, q) > -\infty, \tag{10.27}$$

and seek a sequence $\{(p^n, q^n)\}$ such that $\{\Lambda(p^n, q^n)\}$ converges to b. The sequence is obtained by the AM method, as in our previous discussion. They prove that, if the Bregman distance is jointly convex, then $\{\Lambda(p^n, q^n)\} \downarrow b$. In this subsection we obtain this result by showing that $\Lambda(p, q)$ has the five-point property whenever $D = D_f$ is jointly convex. Our proof is loosely based on the proof of the Eggermont-LaRiccia lemma.

The five-point property for $\Lambda(p, q)$ is

$$\Lambda(p, q^{n-1}) - \Lambda(p^n, q^{n-1}) \geq \Lambda(p, q^n) - \Lambda(p, q). \tag{10.28}$$

A simple calculation shows that the inequality in (10.28) is equivalent to

$$\Lambda(p,q) - \Lambda(p^n,q^n) \geq$$

$$D(p,q^n) + D(p^n,q^{n-1}) - D(p,q^{n-1}) - D(p^n,q^n). \tag{10.29}$$

By the joint convexity of $D(p,q)$ and the convexity of ϕ and ψ we have

$$\Lambda(p,q) - \Lambda(p^n,q^n) \geq$$

$$\langle \nabla_p \Lambda(p^n,q^n), p-p^n\rangle + \langle \nabla_q \Lambda(p^n,q^n), q-q^n\rangle, \tag{10.30}$$

where $\nabla_p \Lambda(p^n,q^n)$ denotes the gradient of $\Lambda(p,q)$, with respect to p, evaluated at (p^n,q^n).

Since q^n minimizes $\Lambda(p^n,q)$, it follows that

$$\langle \nabla_q \Lambda(p^n,q^n), q-q^n\rangle = 0, \tag{10.31}$$

for all q. Therefore,

$$\Lambda(p,q) - \Lambda(p^n,q^n) \geq \langle \nabla_p \Lambda(p^n,q^n), p-p^n\rangle . \tag{10.32}$$

We have

$$\langle \nabla_p \Lambda(p^n,q^n), p-p^n\rangle =$$

$$\langle \nabla f(p^n) - \nabla f(q^n), p-p^n\rangle + \langle \nabla \phi(p^n), p-p^n\rangle. \tag{10.33}$$

Since p^n minimizes $\Lambda(p,q^{n-1})$, we have

$$\nabla_p \Lambda(p^n,q^{n-1}) = 0, \tag{10.34}$$

or

$$\nabla \phi(p^n) = \nabla f(q^{n-1}) - \nabla f(p^n), \tag{10.35}$$

so that

$$\langle \nabla_p \Lambda(p^n,q^n), p-p^n\rangle = \langle \nabla f(q^{n-1}) - \nabla f(q^n), p-p^n\rangle \tag{10.36}$$

$$= D(p,q^n) + D(p^n,q^{n-1}) - D(p,q^{n-1}) - D(p^n,q^n). \tag{10.37}$$

Using (10.32) we obtain the inequality in (10.29). This shows that $\Lambda(p,q)$ has the five-point property whenever the Bregman distance $D = D_f$ is jointly convex. From our previous discussion of AM, we conclude that the sequence $\{\Lambda(p^n,q^n)\}$ converges to b; this is Corollary 4.3 of [6].

As we saw previously, the expectation maximization maximum likelihood (EM) method involves alternating minimization of a function of the form $\Lambda(p,q)$.

If $\psi = 0$, then $\{\Lambda(p^n,q^n)\}$ converges to b, even without the assumption that the distance D_f is jointly convex. In such cases, $\Lambda(p,q)$ has the form of the objective function in proximal minimization and therefore the problem falls into the SUMMA class (see Lemma 5.1).

10.6 AM as SUMMA

We show now that the SUMMA class of AF methods includes all the AM methods for which the five-point property holds.

For each p in the set P, define $q(p)$ in Q as a member of Q for which $\Theta(p, q(p)) \leq \Theta(p, q)$, for all $q \in Q$. Let $f(p) = \Theta(p, q(p))$.

At the nth step of AM we minimize

$$G_n(p) = \Theta(p, q^{n-1}) = \Theta(p, q(p)) + \Big(\Theta(p, q^{n-1}) - \Theta(p, q(p))\Big) \quad (10.38)$$

to get p^n. With

$$g_n(p) = \Big(\Theta(p, q^{n-1}) - \Theta(p, q(p))\Big) \geq 0, \quad (10.39)$$

we can write

$$G_n(p) = f(p) + g_n(p). \quad (10.40)$$

According to the five-point property, we have

$$G_n(p) - G_n(p^n) \geq \Theta(p, q^n) - \Theta(p, q(p)) = g_{n+1}(p). \quad (10.41)$$

It follows that AM is a member of the SUMMA class.

Chapter 11

Appendix One: Theorem 1.3 Revisited

11.1 Improving Theorem 1.3

The proof of Theorem 1.3 made use of the restriction that γ be in the interval $(0, \frac{1}{L})$. For convergence, we need only that γ be in the interval $(0, \frac{2}{L})$, as the following theorem asserts.

Theorem 11.1 *Let $f : \mathbb{R}^J \to \mathbb{R}$ be differentiable, with L-Lipschitz continuous gradient. For γ in the interval $(0, \frac{2}{L})$ the sequence $\{x^k\}$ given by Equation (1.35) converges to a minimizer of f, whenever minimizers exist.*

11.2 Properties of the Gradient

Theorem 11.2 *Let $g : \mathbb{R}^J \to \mathbb{R}$ be differentiable. The following are equivalent:*

- **1)** *$g(x)$ is convex;*
- **2)** *for all a and b we have*

$$g(b) \geq g(a) + \langle \nabla g(a), b - a \rangle \,; \tag{11.1}$$

- **3)** *for all a and b we have*

$$\langle \nabla g(b) - \nabla g(a), b - a \rangle \geq 0. \tag{11.2}$$

Because the operator ∇f is L-Lipschitz continuous, the gradient of the function $g(x) = \frac{1}{L} f(x)$ is non-expansive, that is,

$$\|\nabla g(x) - \nabla g(y)\| \leq \|x - y\|, \tag{11.3}$$

for all x and y.

11.3 Non-expansive gradients

In [51] Golshtein and Tretyakov prove the following theorem.

Theorem 11.3 *Let $g : \mathbb{R}^J \to \mathbb{R}$ be convex and differentiable. The following are equivalent:*

- **1)**

$$||\nabla g(x) - \nabla g(y)||_2 \leq ||x - y||_2; \tag{11.4}$$

- **2)**

$$g(x) \geq g(y) + \langle \nabla g(y), x - y \rangle + \frac{1}{2}||\nabla g(x) - \nabla g(y)||_2^2; \tag{11.5}$$

and

- **3)**

$$\langle \nabla g(x) - \nabla g(y), x - y \rangle \geq ||\nabla g(x) - \nabla g(y)||_2^2. \tag{11.6}$$

Proof: The only non-trivial step in the proof is showing that Inequality (11.4) implies Inequality (11.5). From Theorem 11.2 we see that Inequality (11.4) implies that the function $h(x) = \frac{1}{2}\|x\|^2 - g(x)$ is convex, and that

$$\frac{1}{2}\|x - y\|^2 \geq g(x) - g(y) - \langle \nabla g(y), x - y \rangle ,$$

for all x and y. Now fix y and define

$$d(z) = D_g(z, y) = g(z) - g(y) - \langle \nabla g(y), z - y \rangle,$$

for all z. Since the function $g(z)$ is convex, so is $d(z)$. Since

$$\nabla d(z) = \nabla g(z) - \nabla g(y),$$

it follows from Inequality (11.4) that

$$\|\nabla d(z) - \nabla d(x)\| \leq \|z - x\|,$$

for all x and z. Then, from our previous calculations, we may conclude that

$$\frac{1}{2}\|z - x\|^2 \geq d(z) - d(x) - \langle \nabla d(x), z - x \rangle ,$$

for all z and x.

Now let x be arbitrary and

$$z = x - \nabla g(x) + \nabla g(y).$$

Then

$$0 \leq d(z) \leq d(x) - \frac{1}{2}\|\nabla g(x) - \nabla g(y)\|^2.$$

This completes the proof. ∎

This proof is not the same as the one given in [51]. Now we can prove Theorem 11.1.

11.4 Proof of Theorem 11.1

Let $f(z) \leq f(x)$, for all x; then $\nabla f(z) = 0$. Then

$$\|z - x^k\|^2 = \|z - x^{k-1} + \gamma \nabla f(x^{k-1})\|^2 =$$

$$\|z - x^{k-1}\|^2 - 2\gamma \langle \nabla f(z) - \nabla f(x^{k-1}), z - x^{k-1} \rangle + \gamma^2 \|\nabla f(z) - \nabla f(x^{k-1})\|^2.$$

Therefore,

$$\|z - x^{k-1}\|^2 - \|z - x^k\|^2 =$$

$$2\gamma L \langle \nabla g(z) - \nabla g(x^{k-1}), z - x^{k-1} \rangle - \gamma^2 L^2 \|\nabla g(z) - \nabla g(x^{k-1})\|^2 \geq$$

$$(2\gamma L - \gamma^2 L^2) \|\nabla g(z) - \nabla g(x^{k-1})\|^2.$$

Since $0 < \gamma < \frac{2}{L}$, the sequence $\{\|z - x^k\|\}$ is decreasing and the sequence $\{\|\nabla f(z) - \nabla f(x^k)\|\}$ converges to zero. There is then a subsequence of $\{x^k\}$ converging to some x^* with $\nabla f(x^*) = 0$, so that x^* is a minimizer of f. Replacing the generic z with x^*, we find that the sequence $\{x^k\}$ converges to x^*. This completes the proof. ∎

We can interpret Theorem 11.3 as saying that, if g is convex and differentiable, and its gradient is non-expansive in the 2-norm, then the gradient of g is a firmly non-expansive operator [28].

If $f : \mathbb{R}^J \to \mathbb{R}$ is convex and differentiable, and its gradient is L-Lipschitz continuous, that is,

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2,$$

then the gradient of $g(x) = \frac{1}{L} f(x)$ is a firmly non-expansive operator. It then follows that the operator $I - \gamma \nabla f$ is an averaged operator, for any γ in the interval $(0, \frac{2}{L})$ [28].

Chapter 12

Appendix Two: Bregman-Legendre Functions

In [3] 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 [3].

12.1 Essential Smoothness and Essential Strict Convexity

Following [69] 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 $x^n \in \text{int}D$, with $x^n \to x \in \text{bd}D$, implies that $||\nabla f(x^n)||_2 \to +\infty$. Here

$$D = \{x | f(x) < +\infty\},$$

and 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 $x \in \text{bd}D$ and is $\{\nabla f(x)\}$ for $x \in \text{int}D$ (so f is differentiable on int$D$) if and only if the function f^* is essentially strictly convex.

Definition 12.1 *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 $f^* = \mathbb{R}^J$ then the range of ∇f is $\mathbb{R}^J$ and the equation $\nabla f(x) = y$ can be solved for every $y \in \mathbb{R}^J$. In order for int dom $f^* = \mathbb{R}^J$ it is necessary and sufficient that the Legendre function f be *super-coercive*, that is,

$$\lim_{||x||_2 \to +\infty} \frac{f(x)}{||x||_2} = +\infty. \tag{12.1}$$

If the effective domain of f is bounded, then f is super-coercive and its gradient operator is a mapping onto the space $\mathbb{R}^J$.

12.2 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 $x \in \mathbb{R}^J$ and $z \in$ int$D$ by

$$D_f(x, z) = f(x) - f(z) - \langle \nabla f(z), x - z \rangle. \tag{12.2}$$

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

$$\overleftarrow{P}_K^f(z) = \text{argmin}_{x \in K \cap D} D_f(x, z). \tag{12.3}$$

If f is essentially strictly convex, then $\overleftarrow{P}_K^f(z)$ exists. If f is strictly convex on D then $\overleftarrow{P}_K^f(z)$ is unique. If f is Legendre, then $\overleftarrow{P}_K^f(z)$ 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 $\overleftarrow{P}^f_K(z)$ is the unique member of $K \cap \text{int}D$ satisfying the inequality

$$\langle \nabla f(\overleftarrow{P}^f_K(z)) - \nabla f(z), \overleftarrow{P}^f_K(z) - c \rangle \geq 0, \tag{12.4}$$

for all $c \in K$. From this we obtain the *Bregman Inequality*:

$$D_f(c, z) \geq D_f(c, \overleftarrow{P}^f_K(z)) + D_f(\overleftarrow{P}^f_K(z), z), \tag{12.5}$$

for all $c \in K$.

12.3 Bregman-Legendre Functions

Following Bauschke and Borwein [3], 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.

The Bregman functions form a class closely related to the Bregman-Legendre functions. For details see [12].

12.4 Useful Results about Bregman-Legendre Functions

The following results are proved in somewhat more generality in [3].
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 $y^n \in \text{int } D$ and $x \in \mathbb{R}^J$, 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 13

Appendix Three: Urn Models in Remote Sensing

13.1 Chapter Summary

Many inverse problems are problems of *remote sensing*, which we might also call *indirect measurement*. In such problems we do not have direct access to what we are really interested in, and must be content to measure something else that is related to, but not the same as, what interests us. For example, we want to know what is in the suitcases of airline passengers, but, for practical reasons, we cannot open every suitcase. Instead, we x-ray the suitcases. A recent paper describes progress in detecting nuclear material in cargo containers by measuring the scattering, by the shielding, of cosmic rays; you can't get much more *remote* than that. It is a good idea to consider a model that, although quite simple, manages to capture many of the important features of remote sensing applications. To convince the reader that this is indeed a useful model, we relate it to the problem of image reconstruction in *single-photon computed emission tomography* (SPECT).

13.2 The Urn Model

There seems to be a tradition in physics of using simple models or examples involving urns and marbles to illustrate important principles. In keeping with that tradition, we have here two examples, to illustrate various aspects of remote sensing.

Suppose that we have J urns numbered $j = 1, ..., J$, each containing marbles of various colors. Suppose that there are I colors, numbered $i = 1, ..., I$. Suppose also that there is a box containing a large number of small

pieces of paper, and on each piece is written the number of one of the J urns. Assume that I know the precise contents of each urn. My objective is to determine the precise contents of the box, that is, to estimate, for each $j = 1, ..., J$, the probability of selecting the jth urn, which is the relative number of pieces of paper containing the number j.

Out of my view, my assistant removes one piece of paper from the box, takes one marble from the indicated urn, announces to me the color of the marble, and then replaces both the piece of paper and the marble. This action is repeated N times, at the end of which I have a long list of colors, $\mathbf{i} = \{i_1, i_2, ..., i_N\}$, where i_n denotes the color of the nth marble drawn. This list $\mathbf{i}$ is my data, from which I must determine the contents of the box.

This is a form of remote sensing; what we have access to is related to, but not equal to, what we are interested in. What I wish I had is the list of urns used, $\mathbf{j} = \{j_1, j_2, ..., j_N\}$; instead I have $\mathbf{i}$, the list of colors. Sometimes data such as the list of colors is called "incomplete data" , in contrast to the "complete data" , which would be the list $\mathbf{j}$ of the actual urn numbers drawn from the box.

Using our urn model, we can begin to get a feel for the *resolution problem*. If all the marbles of one color are in a single urn, the problem is trivial; when I hear a color, I know immediately which urn contained that marble. My list of colors is then a list of urn numbers; I have the complete data now. My estimate of the number of pieces of paper containing the urn number j is then simply the proportion of draws that resulted in urn j being selected.

At the other extreme, suppose two urns have identical contents. Then I cannot distinguish one urn from the other and I am unable to estimate more than the total number of pieces of paper containing either of the two urn numbers. If the two urns have nearly the same contents, we can distinguish them only by using a very large N. This is the resolution problem.

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

13.3 Some Mathematical Notation

To introduce some mathematical notation, let us denote by x_j the proportion of the pieces of paper that have the number j written on them. Let P_{ij} be the proportion of the marbles in urn j that have the color i. Let y_i be the proportion of times the color i occurs in the list of colors. The expected proportion of times i occurs in the list is $E(y_i) = \sum_{j=1}^{J} P_{ij}x_j = (Px)_i$, where P is the I by J matrix with entries P_{ij} and x is the J by 1 column

vector with entries x_j. A reasonable way to estimate x is to replace $E(y_i)$ with the actual y_i and solve the system of linear equations $y_i = \sum_{j=1}^{J} P_{ij}x_j$, $i = 1, ..., I$. Of course, we require that the x_j be nonnegative and sum to one, so special algorithms may be needed to find such solutions. In a number of applications that fit this model, such as medical tomography, the values x_j are taken to be parameters, the data y_i are statistics, and the x_j are estimated by adopting a probabilistic model and maximizing the likelihood function. Iterative algorithms, such as the expectation maximization (EMML) algorithm are often used for such problems.

13.4 An Application to SPECT Imaging

In *single-photon computed emission tomography* (SPECT) the patient is injected with a chemical to which a radioactive tracer has been attached. Once the chemical reaches its destination within the body the photons emitted by the radioactive tracer are detected by gamma cameras outside the body. The objective is to use the information from the detected photons to infer the relative concentrations of the radioactivity within the patient.

We discretize the problem and assume that the body of the patient consists of J small volume elements, called *voxels*, analogous to *pixels* in digitized images. We let $x_j \geq 0$ be the unknown amount of the radioactivity that is present in the jth voxel, for $j = 1, ..., J$. There are I detectors, denoted $\{i = 1, 2, ..., I\}$. For each i and j we let P_{ij} be the known probability that a photon that is emitted from voxel j is detected at detector i. We denote by i_n the detector at which the nth emitted photon is detected. This photon was emitted at some voxel, denoted j_n; we wish that we had some way of learning what each j_n is, but we must be content with knowing only the i_n. After N photons have been emitted, we have as our data the list $\mathbf{i} = \{i_1, i_2, ..., i_N\}$; this is our *incomplete data*. We wish we had the *complete data*, that is, the list $\mathbf{j} = \{j_1, j_2, ..., j_N\}$, but we do not. Our goal is to estimate the frequency with which each voxel emitted a photon, which we assume, reasonably, to be proportional to the unknown amounts x_j, for $j = 1, ..., J$.

This problem is completely analogous to the urn problem previously discussed. Any mathematical method that solves one of these problems will solve the other one. In the urn problem, the colors were announced; here the detector numbers are announced. There, I wanted to know the urn numbers; here I want to know the voxel numbers. There, I wanted to estimate the frequency with which the jth urn was used; here, I want to estimate the frequency with which the jth voxel is the site of an emission. In the urn model, two urns with nearly the same contents are hard to distinguish unless N is very large; here, two neighboring voxels will be very hard to distinguish (i.e., to resolve) unless N is very large. But in the

SPECT case, a large N means a high dosage, which will be prohibited by safety considerations. Therefore, we have a built-in resolution problem in the SPECT case.

Both problems are examples of probabilistic mixtures, in which the mixing probabilities are the x_j that we seek. The *maximum likelihood* (ML) method of statistical parameter estimation can be used to solve such problems.

Bibliography

1. Auslander, A., and Teboulle, M. (2006) "Interior gradient and proximal methods for convex and conic optimization" *SIAM Journal on Optimization*, **16(3)**, pp. 697–725.

2. Bauschke, H., and Borwein, J. (1996) "On projection algorithms for solving convex feasibility problems." *SIAM Review*, **38 (3)**, pp. 367–426.

3. Bauschke, H., and Borwein, J. (1997) "Legendre functions and the method of random Bregman projections." *Journal of Convex Analysis*, **4**, pp. 27–67.

4. Bauschke, H., and Borwein, J. (2001) "Joint and separate convexity of the Bregman distance." in [11], pp. 23–36.

5. Bauschke, H., and Combettes, P. (2003) "Iterating Bregman retractions." *SIAM Journal on Optimization*, **13**, pp. 1159–1173.

6. Bauschke, H., Combettes, P., and Noll, D. (2006) "Joint minimization with alternating Bregman proximity operators." *Pacific Journal of Optimization*, **2**, pp. 401–424.

7. Becker, M., Yang, I., and Lange, K. (1997) "EM algorithms without missing data." *Stat. Methods Med. Res.*, **6**, pp. 38–54.

8. Bertero, M., and Boccacci, P. (1998) *Introduction to Inverse Problems in Imaging* Bristol, UK: Institute of Physics Publishing.

9. 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.

10. Bruck, R., and Reich, S. (1977) "Nonexpansive projections and resolvents of accretive operators in Banach spaces." *Houston J. of Mathematics* **3**, pp. 459–470.

11. Butnariu, D., Censor, Y., and Reich, S. (eds.) (2001) *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications*, Studies in Computational Mathematics 8. Amsterdam: Elsevier Publ.

12. Butnariu, D., Byrne, C., and Censor, Y. (2003) "Redundant axioms in the definition of Bregman functions." *Journal of Convex Analysis*, **10**, pp. 245–254.

13. Byrne, C. and Fitzgerald, R. (1979) "A unifying model for spectrum estimation." In *Proceedings of the RADC Workshop on Spectrum Estimation*, Griffiss AFB, Rome, NY, October.

14. Byrne, C. and Fitzgerald, R. (1982) "Reconstruction from partial information, with applications to tomography." *SIAM J. Applied Math.* **42(4)**, pp. 933–940.

15. Byrne, C., Fitzgerald, R., Fiddy, M., Hall, T., and Darling, A. (1983) "Image restoration and resolution enhancement." *J. Opt. Soc. Amer.* **73**, pp. 1481–1487.

16. Byrne, C. and Fitzgerald, R. (1984) "Spectral estimators that extend the maximum entropy and maximum likelihood methods." *SIAM J. Applied Math.* **44(2)**, pp. 425–442.

17. Byrne, C., Levine, B.M., and Dainty, J.C. (1984) "Stable estimation of the probability density function of intensity from photon frequency counts." *JOSA Communications* **1(11)**, pp. 1132–1135.

18. Byrne, C. and Fiddy, M. (1987) "Estimation of continuous object distributions from Fourier magnitude measurements." *JOSA A* **4**, pp. 412–417.

19. Byrne, C. and Fiddy, M. (1988) "Images as power spectra; reconstruction as Wiener filter approximation." *Inverse Problems* **4**, pp. 399–409.

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. (2001) "Bregman-Legendre multi-distance projection algorithms for convex feasibility and optimization." in [11], pp. 87–100.

27. Byrne, C. (2002) "Iterative oblique projection onto convex sets and the split feasibility problem." *Inverse Problems* **18**, pp. 441–453.

28. Byrne, C. (2004) "A unified treatment of some iterative algorithms in signal processing and image reconstruction." *Inverse Problems* **20**, pp. 103–120.

29. Byrne, C. (2005) *Signal Processing: A Mathematical Approach*, AK Peters, Publ., Wellesley, MA.

30. Byrne, C. (2008) "Sequential unconstrained minimization algorithms for constrained optimization." *Inverse Problems*, **24(1)**, article no. 015013.

31. Byrne, C. (2011) *A First Course in Optimization*, available as a pdf file at my web site.

32. Byrne, C. (2013) "Alternating minimization as sequential unconstrained minimization: a survey." *Journal of Optimization Theory and Applications*, electronic **154(3)**, DOI 10.1007/s1090134-2, (2012), and hardcopy **156(3)**, February, 2013, pp. 554–566.

33. Byrne, C. (2013) "An elementary proof of convergence of the forward-backward splitting algorithm." to appear in the *Journal of Nonlinear and Convex Analysis*.

34. 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.

35. Byrne, C., and Eggermont, P. (2011) "EM Algorithms." in *Handbook of Mathematical Methods in Imaging*, Otmar Scherzer, ed., Springer-Science.

36. Censor, Y. and Elfving, T. (1994) "A multi-projection algorithm using Bregman projections in a product space." *Numerical Algorithms*, **8** 221–239.

37. Censor, Y., Bortfeld, T., Martin, B., and Trofimov, A. "A unified approach for inversion problems in intensity-modulated radiation therapy." *Physics in Medicine and Biology* 51 (2006), 2353-2365.

38. Censor, Y., Elfving, T., Kopf, N., and Bortfeld, T. (2005) "The multiple-sets split feasibility problem and its application for inverse problems." *Inverse Problems*, **21** , pp. 2071-2084.

39. Censor, Y. and Segman, J. (1987) "On block-iterative maximization." *J. of Information and Optimization Sciences* **8**, pp. 275–291.

40. Censor, Y. and Zenios, S.A. (1997) *Parallel Optimization: Theory, Algorithms and Applications*. New York: Oxford University Press.

41. Combettes, P., and Wajs, V. (2005) "Signal recovery by proximal forward-backward splitting." *Multiscale Modeling and Simulation*, **4(4)**, pp. 1168–1200.

42. Csiszár, I. and Tusnády, G. (1984) "Information geometry and alternating minimization procedures." *Statistics and Decisions* **Supp. 1**, pp. 205–237.

43. 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.

44. Darroch, J. and Ratcliff, D. (1972) "Generalized iterative scaling for log-linear models." *Annals of Mathematical Statistics* **43**, pp. 1470–1480.

45. 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.

46. Eggermont, P., and LaRiccia, V. (2001) *Maximum Penalized Likelihood Estimation*. New York: Springer.

47. Fiacco, A., and McCormick, G. (1990) *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Philadelphia, PA: SIAM Classics in Mathematics (reissue).

48. 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.

49. Goebel, K., and Reich, S. (1984) *Uniform Convexity, Hyperbolic Geometry, and Nonexpansive Mappings*, New York: Dekker.

50. Goldstein, S., and Osher, S. (2008) "The split Bregman algorithm for L^1 regularized problems. " UCLA CAM Report 08-29, UCLA, Los Angeles.

51. Golshtein, E., and Tretyakov, N. (1996) *Modified Lagrangians and Monotone Maps in Optimization*. New York: John Wiley and Sons, Inc.

52. Jones, L., and Byrne, C. (1990) "General entropy criteria for inverse problems, with applications to data compression, pattern classification, and cluster analysis." *IEEE Trans. Information Th.*, **36(1)**, pp. 23–30.

53. Krasnosel'skiĭ, M. (1955) "Two remarks on the method of successive approximations" (in Russian). *Uspekhi Mathematicheskikh Nauk*, **10**, pp. 123–127.

54. Kullback, S. and Leibler, R. (1951) "On information and sufficiency." *Annals of Mathematical Statistics* **22**, pp. 79–86.

55. Landweber, L. (1951) "An iterative formula for Fredholm integral equations of the first kind." *Amer. J. of Math.* **73**, pp. 615–624.

56. Lange, K. and Carson, R. (1984) "EM reconstruction algorithms for emission and transmission tomography." *Journal of Computer Assisted Tomography* **8**, pp. 306–316.

57. 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.

58. Lange, K., Hunter, D., and Yang, I. (2000) "Optimization transfer using surrogate objective functions (with discussion)." *J. Comput. Graph. Statist.*, **9**, pp. 1–20.

59. Mann, W. (1953) "Mean value methods in iteration." *Proceedings of the American Mathematical Society*, **4**, pp. 506–510.

60. Masad, E., and Reich, S. (2007) "A note on the multiple-set split convex feasibility problem in Hilbert space." *J. Nonlinear Convex Analysis*, **8**, pp. 367–371.

61. McLachlan, G.J. and Krishnan, T. (1997) *The EM Algorithm and Extensions*. New York: John Wiley and Sons, Inc.

62. Moreau, J.-J. (1962) "Fonctions convexes duales et points proximaux dans un espace hilbertien." *C.R. Acad. Sci. Paris Sér. A Math.*, **255**, pp. 2897–2899.

63. Moreau, J.-J. (1963) "Propriétés des applications 'prox'." *C.R. Acad. Sci. Paris Sér. A Math.*, **256**, pp. 1069–1071.

64. Moreau, J.-J. (1965) "Proximité et dualité dans un espace hilbertien." *Bull. Soc. Math. France*, **93**, pp. 273–299.

65. Nash, S. and Sofer, A. (1996) *Linear and Nonlinear Programming*. New York: McGraw-Hill.

66. Nesterov, Y., and Nemirovski, A. (1994) *Interior-Point Polynomial Algorithms in Convex Programming*. Philadelphia, PA: SIAM Studies in Applied Mathematics.

67. Ortega, J., and Rheinboldt, W. (2000) *Iterative Solution of Nonlinear Equations in Several Variables*, Classics in Applied Mathematics, 30. Philadelphia, PA: SIAM, 2000

68. Renegar, J. (2001) *A Mathematical View of Interior-Point Methods in Convex Optimization*. Philadelphia, PA: SIAM (MPS-SIAM Series on Optimization).

69. Rockafellar, R. (1970) *Convex Analysis*. Princeton, NJ: Princeton University Press.

70. Schmidlin, P. (1972) "Iterative separation of sections in tomographic scintigrams." *Nucl. Med.* **15(1)**.

71. Shepp, L., and Vardi, Y. (1982) "Maximum likelihood reconstruction for emission tomography." *IEEE Transactions on Medical Imaging*, **MI-1**, pp. 113–122.

72. Shieh, M., Byrne, C., and Fiddy, M. (2006) "Image reconstruction: a unifying model for resolution enhancement and data extrapolation: Tutorial." *Journal of the Optical Society of America, A*, **23(2)**, pp. 258–266.

73. Shieh, M., Byrne, C., Testorf, M., and Fiddy, M. (2006) "Iterative image reconstruction using prior knowledge." *Journal of the Optical Society of America, A*, **23(6)**, pp. 1292–1300.

74. 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.