Optimization

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

March 4, 2009

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

Contents

Chapter 1

Introduction

In this course we focus on three things:

- **1.** mathematical problems of maximizing or minimizing functions of one or several variables, possibly subject to constraints on those variables;
- **2.** applications giving rise to such optimization problems; and
- **3.** algorithms to solve such problems.

In this chapter we present a brief overview of the topics to be discussed in more detail later.

1.1 Two Types of Applications

Optimization means maximizing or minimizing some function of one or, more often, several variables. There are two distinct types of applications that lead to optimization problems, which, to give them a name, we shall call *problems of optimization* and *problems of inference*. We shall consider both types in this book.

1.1.1 Problems of Optimization

On the one hand, there are problems of optimization, in which optimizing the given function is, more or less, the sole and natural objective. The main goal, maximum profits, shortest commute, is not open to question, although the precise function involved will depend on the simplifications adopted as the real-world problem is turned into mathematics. Examples of such problems are a manufacturer seeking to maximize profits, subject to whatever restrictions the situation imposes, or a commuter trying to

minimize the time it takes to get to work, subject, of course, to speed limits. In converting the real-world problem to a mathematical problem, the manufacturer may or may not ignore non-linearities such as economies of scale, and the commuter may or may not employ probabilistic models of traffic density. The resulting mathematical optimization problem to be solved will depend on such choices, but the original real-world problem is one of optimization, nevertheless.

Operations Research (OR) is a broad field involving a variety of applied optimization problems. Wars and organized violence have always given impetus to technological advances, most significantly during the twentieth century. An important step was taken when scientists employed by the military realized that studying and improving the use of existing technology could be as important as discovering new technology. Conducting research into on-going operations, that is, doing operations research, led to the search for better, indeed, optimal, ways to schedule ships entering port, to design convoys, to paint the under-sides of aircraft, to hunt submarines, and many other seemingly mundane tasks [106]. Problems having to do with the allocation of limited resources arise in a wide variety of applications, all of which fall under the broad umbrella of OR.

1.1.2 Problems of Inference

On the other hand, there are *problems of inference*, in which optimization is a useful tool, but not the primary objective. These are problems in which estimates are to be made from observations. Such problems arise in many remote sensing applications, radio astronomy, or medical imaging, for example, in which, for practical reasons, the data obtained are insufficient or too noisy to specify a unique source, and one turns to optimization methods, such as likelihood maximization or least-squares, to provide usable approximations. In such cases, it is not the optimization of a function that concerns us, but the optimization of technique. We cannot know which reconstructed image is the best, in the sense of most closely describing the true situation, but we do know which techniques of reconstruction are "best" in some specific sense. We choose techniques such as likelihood or entropy maximization, or least-mean-squares minimization, because these methods are "optimal" in some sense, not because any single result obtained using these methods is guaranteed to be the best. Generally, these methods are "best"in some average sense; indeed, this is the basic idea in statistical estimation.

As we shall see, in both types of problems, the optimization usually cannot be performed by algebraic means alone and iterative algorithms are required.

The mathematical tools required do not usually depend on which type of problem we are trying to solve. A manufacturer may use the theory of linear

programming to maximize profits, while an oncologist may use likelihood maximization to image a tumor and linear programming to determine a suitable spatial distribution of radiation intensities for the therapy. The only difference, perhaps, is that the doctor may have some choice in how, or even whether or not, to involve optimization in solving the medical problems, while the manufacturer's problem is an optimization problem from the start, and a linear programming problem once the mathematical model is selected.

1.2 Types of Optimization Problems

The optimization problems we shall discuss differ, one from another, in the nature of the functions being optimized and the constraints that may or may not be imposed. The constraints may, themselves, involve other functions; we may wish to minimize $f(x)$, subject to the constraint $g(x) \leq 0$. The functions may be differentiable, or not, they may be linear, or not. If they are not linear, they may be convex. They may become linear or convex once we change variables. The various problem types have names, such as Linear Programming, Quadratic Programming, Geometric Programming, and Convex Programming; the use of the term 'programming' is an historical accident and has no connection with computer programming.

All of the problems discussed so far involve functions of one or several real variables. In the Calculus of Variations, the function to be optimized is a *functional*, which is a real-valued function of functions. For example, we may wish to find the curve having the shortest length connecting two given points, say $(0, 0)$ and $(1, 1)$, in R^2. The functional to be minimized is

$$J(y) = \int_0^1 \sqrt{1 + (\frac{dy}{dx})^2} dx.$$

We know that the optimal function is a straight line. In general, the optimal function $y = f(x)$ will satisfy a differential equation, known as the Euler-Lagrange Equation.

1.3 Algorithms

The algorithms we shall study include general-purpose optimization methods, as well as techniques tailored to particular types of problems.

1.3.1 Root-Finding

One of the first applications of the derivative that we encounter in Calculus I is optimization, maximizing or minimizing a differentiable real-valued

function $f(x)$ of a single real variable over x in some interval $[a, b]$. Since $f(x)$ is differentiable, it is continuous, so we know that $f(x)$ attains its maximum and minimum values over the interval $[a, b]$. The standard procedure is to differentiate $f(x)$ and compare the values of $f(x)$ at the places where $f'(x) = 0$ with the values $f(a)$ and $f(b)$. These places include the values of x where the optimal values of $f(x)$ occur. However, we may not be able to solve the equation $f'(x) = 0$ algebraically, and may need to employ numerical, approximate techniques. It may, in fact, be simpler to use an iterative technique to minimize $f(x)$ directly.

Perhaps the simplest example of an iterative method is the *bi-section method* for finding a root of a continuous function of a single real variable.

Let $g : R \to R$ be continuous. Suppose that $g(a) < 0$ and $g(b) > 0$. Then, by the Intermediate Value Theorem, we know that there is a point c in (a, b) with $g(c) = 0$. Let $m = \frac{a+b}{2}$ be the mid-point of the interval. If $g(m) = 0$, then we are done. If $g(m) < 0$, replace a with m; otherwise, replace b with m. Now calculate the mid-point of the new interval and continue. At each step, the new interval is half as big as the old one and still contains a root of $g(x)$. The distance from the left end point to the root is not greater than the length of the interval, which provides a good estimate of the accuracy of the approximation.

1.3.2 Iterative Descent Methods

Suppose that we wish to minimize the real-valued function $f : R^J \to R$ of J real variables. If f is Gâteaux-differentiable (see appendix), then the two-sided directional derivative of f, at the point a, in the direction of the unit vector d, is

$$f'(a; d) = \lim_{t \to 0} \frac{1}{t}[f(a + td) - f(a)] = \langle \nabla f(a), d \rangle.$$

According to the Cauchy-Schwarz Inequality, we have

$$|\langle \nabla f(a), d \rangle| \leq ||\nabla f(a)||\, ||d||,$$

with equality if and only if the direction vector d is parallel to the vector $\nabla f(a)$. Therefore, from the point a, the direction of greatest increase of f is $d = \nabla f(a)$, and the direction of greatest decrease is $d = -\nabla f(a)$.

If f is Gâteaux-differentiable, and $f(a) \leq f(x)$, for all x, then $\nabla f(a) = 0$. Therefore, we can, in theory, find the minimum of f by finding the point (or points) $x = a$ where the gradient is zero. For example, suppose we wish to minimize the function

$$f(x, y) = 3x^2 + 4xy + 5y^2 + 6x + 7y + 8.$$

Setting the partial derivatives to zero, we have

$$0 = 6x + 4y + 6,$$

and

$$0 = 4x + 10y + 7.$$

Therefore, minimizing $f(x, y)$ involves solving this system of two linear equations in two unknowns. This is easy, but if f has many variables, not just two, or if f is not a quadratic function, the resulting system will be quite large and may include nonlinear functions, and we may need to employ iterative methods to solve this system. Once we decide that we need to use iterative methods, we may as well consider using them directly on the original optimization problem, rather than to solve the system derived by setting the gradient to zero. We cannot hope to solve all optimization problems simply by setting the gradient to zero and solving the resulting system of equations algebraically.

For $k = 0, 1, ...$, having calculated the current estimate x^k, we select a direction vector d^k such that $f(x^k + \alpha d^k)$ is decreasing, as a function of $\alpha > 0$, and a step-length α_k. Our next estimate is $x^{k+1} = x^k + \alpha_k d^k$. We may choose α_k to minimize $f(x^k + \alpha d^k)$, as a function of α, although this is usually computationally difficult. For (Gâteaux) differentiable f, the gradient, $\nabla f(x)$, is the direction of greatest increase of f, as we move away from the point x. Therefore, it is reasonable, although not required, to select $d^k = -\nabla f(x^k)$ as the new direction vector; then we have a *gradient descent method*.

1.3.3 Solving Systems of Linear Equations

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

1.3.4 Imposing Constraints

The iterative algorithms we shall investigate begin with an initial guess x^0 of the solution, and then generate a sequence $\{x^k\}$, converging, in the best cases, to our solution. Suppose we wish to minimize $f(x)$ over all x in R^J having non-negative entries. An iterative algorithm is said to be an *interior-point method* if each vector x^k has non-negative entries.

1.3.5 Operators

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

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

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

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

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

1.3.6 Search Techniques

In linear programming, it is known that the solution to the problem is one of a finite number of vectors, the vertices, each of which can be calculated. The problem is to avoid having to calculate all of them. Useful algorithms, such as Dantzig's *simplex method*, move from one vertex to another in an efficient way, and, at least most of the time, solve the problem in a fraction of the time that would have been required to check each vertex.

1.3.7 Acceleration

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

Chapter 2

Optimization without Calculus

In our study of optimization, we shall encounter a number of sophisticated techniques, involving first and second partial derivatives, systems of linear equations, nonlinear operators, specialized distance measures, and so on. It is good to begin by looking at what can be accomplished without sophisticated techniques, even without calculus. It is possible to achieve much with powerful, yet simple, inequalities. As someone once remarked, exaggerating slightly, in the right hands, the Cauchy Inequality and integration by parts are all that are really needed.

Students typically encounter optimization problems as applications of differentiation, while the possibility of optimizing without calculus is left unexplored. In this chapter we discuss optimization methods based on the Arithmetic Mean-Geometric Mean Inequality and Cauchy's Inequality.

2.1 The Arithmetic Mean-Geometric Mean Inequality

Let $x_1, ..., x_N$ be positive numbers. According to the famous *Arithmetic Mean-Geometric Mean Inequality*, abbreviated AGM Inequality,

$$G = (x_1 \cdot x_2 \cdots x_N)^{1/N} \leq A = \frac{1}{N}(x_1 + x_2 + ... + x_N), \tag{2.1}$$

with equality if and only if $x_1 = x_2 = ... = x_N$. To prove this, consider the following modification of the product $x_1 \cdots x_N$. Replace the smallest of the x_n, call it x, with A and the largest, call it y, with $x + y - A$. This modification does not change the arithmetic mean of the N numbers, but the product increases, unless $x = y = A$ already, since $xy \leq A(x + y - A)$

(Why?). We repeat this modification, until all the x_n approach A, at which point the product reaches its maximum.

For example, $2 \cdot 3 \cdot 4 \cdot 6 \cdot 20$ becomes $3 \cdot 4 \cdot 6 \cdot 7 \cdot 15$, and then $4 \cdot 6 \cdot 7 \cdot 7 \cdot 11$, $6 \cdot 7 \cdot 7 \cdot 7 \cdot 8$, and finally $7 \cdot 7 \cdot 7 \cdot 7 \cdot 7$.

2.2 An Application of the AGM Inequality: the Number e

We can use the AGM Inequality to show that

$$\lim_{n \to \infty} (1 + \frac{1}{n})^n = e. \tag{2.2}$$

Let $f(n) = (1 + \frac{1}{n})^n$, the product of the $n + 1$ numbers $1, 1 + \frac{1}{n}, ..., 1 + \frac{1}{n}$. Applying the AGM Inequality, we obtain the inequality

$$f(n) \leq \left(\frac{n+2}{n+1}\right)^{n+1} = f(n+1),$$

so we know that the sequence $\{f(n)\}$ is increasing. Now define $g(n) = (1 + \frac{1}{n})^{n+1}$; we show that $g(n) \leq g(n-1)$ and $f(n) \leq g(m)$, for all positive integers m and n. Consider $(1 - \frac{1}{n})^n$, the product of the $n + 1$ numbers $1, 1 - \frac{1}{n}, ..., 1 - \frac{1}{n}$. Applying the AGM Inequality, we find that

$$\left(1 - \frac{1}{n+1}\right)^{n+1} \geq \left(1 - \frac{1}{n}\right)^n,$$

or

$$\left(\frac{n}{n+1}\right)^{n+1} \geq \left(\frac{n-1}{n}\right)^n.$$

Taking reciprocals, we get $g(n) \leq g(n-1)$. Since $f(n) < g(n)$ and $\{f(n)\}$ is increasing, while $\{g(n)\}$ is decreasing, we can conclude that $f(n) \leq g(m)$, for all positive integers m and n. Both sequences therefore have limits. Because the difference

$$g(n) - f(n) = \frac{1}{n}(1 + \frac{1}{n})^n \to 0,$$

as $n \to \infty$, we conclude that the limits are the same. This common limit we can define as the number e.

2.3 Extending the AGM Inequality

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

$$x_1^{a_1} x_2^{a_2} \cdots x_N^{a_N} \leq a_1 x_1 + a_2 x_2 + ... + a_N x_N, \tag{2.3}$$

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

A function $f(x)$ is said to be *convex* over an interval (a, b) if

$$f(a_1t_1 + a_2t_2 + ... + a_Nt_N) \leq a_1f(t_1) + a_2f(t_2) + ... + a_Nf(t_N),$$

for all positive integers N, all a_n as above, and all real numbers t_n in (a, b). If the function $f(x)$ is twice differentiable on (a, b), then $f(x)$ is convex over (a, b) if and only if the second derivative of $f(x)$ is non-negative on (a, b). For example, the function $f(x) = -\log x$ is convex on the positive x-axis. The GAGM Inequality follows immediately.

2.4 Optimization Using the AGM Inequality

We illustrate the use of the AGM Inequality for optimization through several examples.

2.4.1 Example 1

Find the minimum of the function

$$f(x, y) = \frac{12}{x} + \frac{18}{y} + xy,$$

over positive x and y.

We note that the three terms in the sum have a fixed product of 216, so, by the AGM Inequality, the smallest value of $\frac{1}{3}f(x, y)$ is $(216)^{1/3} = 6$ and occurs when the three terms are equal and each equal to 6, so when $x = 2$ and $y = 3$. The smallest value of $f(x, y)$ is therefore 18.

2.4.2 Example 2

Find the maximum value of the product

$$f(x, y) = xy(72 - 3x - 4y),$$

over positive x and y.

The terms x, y and $72 - 3x - 4y$ do not have a constant sum, but the terms $3x$, $4y$ and $72 - 3x - 4y$ do have a constant sum, namely 72, so we rewrite $f(x, y)$ as

$$f(x, y) = \frac{1}{12}(3x)(4y)(72 - 3x - 4y).$$

By the AGM Inequality, the product $(3x)(4y)(72 - 3x - 4y)$ is maximized when the factors $3x$, $4y$ and $72 - 3x - 4y$ are each equal to 24, so when $x = 8$ and $y = 6$. The maximum value of the product is then 1152.

2.4.3 Example 3

Both of the previous two problems can be solved using the standard calculus technique of setting the two first partial derivatives to zero. Here is an example that is not so easily solved in that way: minimize the function

$$f(x,y) = 4x + \frac{x}{y^2} + \frac{4y}{x},$$

over positive values of x and y. Try taking the first partial derivatives and setting them both to zero. Even if we managed to solve this system of coupled nonlinear equations, deciding if we actually have found the minimum is not easy; take a look at the second derivative matrix, the Hessian matrix. We can employ the AGM Inequality by rewriting $f(x,y)$ as

$$f(x,y) = 4\Big(\frac{4x + \frac{x}{y^2} + \frac{2y}{x} + \frac{2y}{x}}{4}\Big).$$

The product of the four terms in the arithmetic mean expression is 16, so the GM is 2. Therefore, $\frac{1}{4}f(x,y) \geq 2$, with equality when all four terms are equal to 2; that is, $4x = 2$, so that $x = \frac{1}{2}$ and $\frac{2y}{x} = 2$, so $y = \frac{1}{2}$ also. The minimum value of $f(x,y)$ is then 8.

2.5 The Hölder and Minkowski Inequalities

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

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

The p-norm of c is defined to be

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

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

2.5.1 Hölder's Inequality

Hölder's Inequality is the following:

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

with equality if and only if

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

for each n.

Hölder's Inequality follows from the GAGM Inequality. To see this, we fix n and apply Inequality (2.3), with

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

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

$$x_2 = \Big(\frac{|d_n|}{\|d\|_q}\Big)^q,$$

and

$$a_2 = \frac{1}{q}.$$

From (2.3) we then have

$$\Big(\frac{|c_n|}{\|c\|_p}\Big)\Big(\frac{|d_n|}{\|d\|_q}\Big) \le \frac{1}{p}\Big(\frac{|c_n|}{\|c\|_p}\Big)^p + \frac{1}{q}\Big(\frac{|d_n|}{\|d\|_q}\Big)^q.$$

Now sum both sides over the index n.

2.5.2 Minkowski's Inequality

Minkowski's Inequality, which is a consequence of Hölder's Inequality, states that

$$\|c + d\|_p \le \|c\|_p + \|d\|_p\,;$$

it is the triangle inequality for the metric induced by the p-norm.

To prove Minkowski's Inequality, we write

$$\sum_{n=1}^{N} |c_n + d_n|^p \le \sum_{n=1}^{N} |c_n|(|c_n + d_n|)^{p-1} + \sum_{n=1}^{N} |d_n|(|c_n + d_n|)^{p-1}.$$

Then we apply Hölder's Inequality to both of the sums.

2.6 Cauchy's Inequality

For the choices $p = q = 2$, Hölder's Inequality becomes the famous Cauchy Inequality, which we rederive in a different way in this section. For simplicity, we assume now that the vectors have real entries and for notational convenience later we use x_n and y_n in place of c_n and d_n.

Let $x = (x_1, ..., x_N)$ and $y = (y_1, ..., y_N)$ be vectors with real entries. The *inner product* of x and y is

$$\langle x, y\rangle = x_1y_1 + x_2y_2 + ... + x_Ny_N. \tag{2.4}$$

The 2-norm of the vector x, which we shall simply call the *norm* of the vector x is

$$\|x\|_2 = \|x\| = \sqrt{\langle x, x\rangle}.$$

Cauchy's Inequality is

$$|\langle x, y\rangle| \leq \|x\|\,\|y\|, \tag{2.5}$$

with equality if and only if there is a real number a such that $x = ay$.

To prove Cauchy's Inequality, we begin with the fact that, for every real number t,

$$0 \leq \|x - ty\|^2 = \|x\|^2 - (2\langle x, y\rangle)t + \|y\|^2t^2.$$

This quadratic in the variable t is never negative, so cannot have two real roots. It follows that the term under the radical sign in the quadratic equation must be non-positive, that is,

$$(2\langle x, y\rangle)^2 - 4\|y\|^2\|x\|^2 \leq 0. \tag{2.6}$$

We have equality in (2.6) if and only if the quadratic has a double real root, say $t = a$. Then we have

$$\|x - ay\|^2 = 0.$$

As an aside, suppose we had allowed the variable t to be complex. Clearly $\|x - ty\|$ cannot be zero for any non-real value of t. Doesn't this contradict the fact that every quadratic has two roots in the complex plane?

The Pólya-Szegö Inequality

We can interpret Cauchy's Inequality as providing an upper bound for the quantity

$$\Big(\sum_{n=1}^{N} x_ny_n\Big)^2.$$

The *Pólya-Szegö Inequality* provides a lower bound for the same quantity. Let $0 < m_1 \leq x_n \leq M_1$ and $0 < m_2 \leq y_n \leq M_2$, for all n. Then

$$\sum_{n=1}^{N} x_n^2 \sum_{n=1}^{N} y_n^2 \leq \frac{M_1M_2 + m_1m_2}{4m_1m_2M_1M_2}\Big(\sum_{n=1}^{N} x_ny_n\Big)^2. \tag{2.7}$$

2.7 Optimizing using Cauchy's Inequality

We present two examples to illustrate the use of Cauchy's Inequality in optimization.

2.7.1 Example 4

Find the largest and smallest values of the function

$$f(x, y, z) = 2x + 3y + 6z, \tag{2.8}$$

among the points (x, y, z) with $x^2 + y^2 + z^2 = 1$.

From Cauchy's Inequality we know that

$$49 = (2^2 + 3^2 + 6^2)(x^2 + y^2 + z^2) \geq (2x + 3y + 6z)^2,$$

so that $f(x, y, z)$ lies in the interval $[-7, 7]$. We have equality in Cauchy's Inequality if and only if the vector $(2, 3, 6)$ is parallel to the vector (x, y, z), that is

$$\frac{x}{2} = \frac{y}{3} = \frac{z}{6}.$$

It follows that $x = t$, $y = \frac{3}{2}t$, and $z = 3t$, with $t^2 = \frac{4}{49}$. The smallest value of $f(x, y, z)$ is -7, when $x = -\frac{2}{7}$, and the largest value is $+7$, when $x = \frac{2}{7}$.

2.7.2 Example 5

The simplest problem in estimation theory is to estimate the value of a constant c, given J data values $z_j = c + v_j$, $j = 1, ..., J$, where the v_j are random variables representing additive noise or measurement error. Assume that the expected values of the v_j are $E(v_j) = 0$, the v_j are uncorrelated, so $E(v_j v_k) = 0$ for j different from k, and the variances of the v_j are $E(v_j^2) = \sigma_j^2 > 0$. A *linear* estimate of c has the form

$$\hat{c} = \sum_{j=1}^{J} b_j z_j. \tag{2.9}$$

The estimate $\hat{c}$ is *unbiased* if $E(\hat{c}) = c$, which forces $\sum_{j=1}^{J} b_j = 1$. The *best* linear unbiased estimator, the BLUE, is the one for which $E((\hat{c} - c)^2)$ is minimized. This means that the b_j must minimize

$$E\Big(\sum_{j=1}^{J}\sum_{k=1}^{J} b_j b_k v_j v_k\Big) = \sum_{j=1}^{J} b_j^2 \sigma_j^2, \tag{2.10}$$

subject to

$$\sum_{j=1}^{J} b_j = 1. \tag{2.11}$$

To solve this minimization problem, we turn to Cauchy's Inequality.

We can write

$$1 = \sum_{j=1}^{J} b_j = \sum_{j=1}^{J} (b_j \sigma_j) \frac{1}{\sigma_j}.$$

Cauchy's Inequality then tells us that

$$1 \leq \sqrt{\sum_{j=1}^{J} b_j^2 \sigma_j^2} \sqrt{\sum_{j=1}^{J} \frac{1}{\sigma_j^2}},$$

with equality if and only if there is a constant, say λ, such that

$$b_j \sigma_j = \lambda \frac{1}{\sigma_j},$$

for each j. So we have

$$b_j = \lambda \frac{1}{\sigma_j^2},$$

for each j. Summing on both sides and using Equation (2.11), we find that

$$\lambda = 1 / \sum_{j=1}^{J} \frac{1}{\sigma_j^2}.$$

The BLUE is therefore

$$\hat{c} = \lambda \sum_{j=1}^{J} \frac{z_j}{\sigma_j^2}. \tag{2.12}$$

When the variances σ_j^2 are all the same, the BLUE is simply the arithmetic mean of the data values z_j.

2.7.3 Example 6

One of the fundamental operations in signal processing is the filtering the data vector $x = \gamma s + n$, to remove the noise component n, while leaving the signal component s relatively unaltered [44]. This can be done either to estimate γ, the amount of the signal vector s present, or to detect if the signal is present at all, that is, to decide if $\gamma = 0$ or not. The noise is

typically known only through its *covariance matrix* Q, which is the positive-definite, symmetric matrix having for its entries $Q_{jk} = E(n_j n_k)$. The filter usually is linear and takes the form of an estimate of γ:

$$\hat{\gamma} = b^T x.$$

We want $|b^T s|^2$ large, and, on average, $|b^T n|^2$ small; that is, we want $E(|b^T n|^2) = b^T E(nn^T)b = b^T Qb$ small. The best choice is the vector b that maximizes the *gain* of the filter, that is, the ratio

$$|b^T s|^2 / b^T Qb.$$

We can solve this problem using the Cauchy Inequality.

Definition 2.1 *Let S be a square matrix. A non-zero vector u is an eigenvector of S if there is a scalar λ such that $Su = \lambda u$. Then the scalar λ is said to be an eigenvalue of S associated with the eigenvector u.*

Definition 2.2 *The transpose, $B = A^T$, of an M by N matrix A is the N by M matrix having the entries $B_{n,m} = A_{m,n}$.*

Definition 2.3 *A square matrix S is symmetric if $S^T = S$.*

A basic theorem in linear algebra is that, for any symmetric N by N matrix S, R^N has an orthogonal basis consisting of eigenvectors of S. If we then define U to be the matrix whose columns are these eigenvectors and L the diagonal matrix with the associated eigenvalues on the diagonal, we can easily see that U is an *orthogonal matrix*, that is, $U^T U = I$. We can then write $S = ULU^T$; this is the *eigenvalue/eigenvector decomposition* of S. The eigenvalues of S are always real numbers.

Definition 2.4 *A J by J matrix Q is non-negative definite if, for every x in R^J, we have $x^T Qx \geq 0$. If $x^T Qx > 0$ whenever x is not the zero vector, then Q is said to be positive definite.*

We leave it to the reader to show that the eigenvalues of a non-negative (positive) definite matrix are always non-negative (positive).

A covariance matrix Q is always non-negative definite, since

$$x^T Qx = E(|\sum_{j=1}^{J} x_j n_j|^2). \tag{2.13}$$

Therefore, its eigenvalues are non-negative; typically, they are actually positive, as we shall assume now. We then let $C = U\sqrt{L}U^T$, the symmetric square root of Q. The Cauchy Inequality then tells us that

$$|b^T s|^2 = |b^T CC^{-1} s|^2 \leq [b^T CC^T b][s^T (C^{-1})^T C^{-1} s],$$

with equality if and only if the vectors $C^T b$ and $C^{-1}s$ are parallel. It follows that

$$b = \alpha(CC^T)^{-1}s = \alpha Q^{-1}s,$$

for any constant α. It is standard practice to select α so that $b^T s = 1$, therefore $\alpha = 1/s^T Q^{-1}s$ and the optimal filter b is

$$b = \frac{1}{s^T Q^{-1} s} Q^{-1}s.$$

2.8 An Inner Product for Square Matrices

The *trace* of a square matrix M, denoted $\mathrm{tr}M$, is the sum of the entries down the main diagonal. Given square matrices A and B with real entries, the trace of the product $B^T A$ defines an inner product, that is

$$\langle A, B\rangle = \mathrm{tr}(B^T A),$$

where the superscript T denotes the transpose of a matrix. This inner product can then be used to define a norm of A, called the *Frobenius norm*, by

$$\|A\|_F = \sqrt{\langle A, A\rangle} = \sqrt{\mathrm{tr}(A^T A)}. \tag{2.14}$$

From the eigenvector/eigenvalue decomposition, we know that, for every symmetric matrix S, there is an orthogonal matrix U such that

$$S = UD(\lambda(S))U^T,$$

where $\lambda(S) = (\lambda_1, ..., \lambda_N)$ is a vector whose entries are eigenvalues of the symmetric matrix S, and $D(\lambda(S))$ is the diagonal matrix whose entries are the entries of $\lambda(S)$. Then we can easily see that

$$\|S\|_F = \|\lambda(S)\|.$$

Denote by $[\lambda(S)]$ the vector of eigenvalues of S, ordered in non-increasing order. We have the following result.

Theorem 2.1 (Fan's Theorem) *Any real symmetric matrices S and R satisfy the inequality*

$$\mathrm{tr}(SR) \leq \langle [\lambda(S)], [\lambda(R)]\rangle,$$

with equality if and only if there is an orthogonal matrix U such that

$$S = UD([\lambda(S)])U^T,$$

and

$$R = UD([\lambda(R)])U^T.$$

From linear algebra, we know that S and R can be simultaneously diagonalized if and only if they commute; this is a stronger condition than simultaneous diagonalization.

If S and R are diagonal matrices already, then Fan's Theorem tells us that

$$\langle \lambda(S), \lambda(R) \rangle \leq \langle [\lambda(S)], [\lambda(R)] \rangle.$$

Since any real vectors x and y are $\lambda(S)$ and $\lambda(R)$, for some symmetric S and R, respectively, we have the following **Hardy-Littlewood-Polya Inequality**:

$$\langle x, y \rangle \leq \langle [x], [y] \rangle.$$

Most of the optimization problems discussed in this chapter fall under the heading of Geometric Programming, which we shall present in a more formal way in a subsequent chapter.

2.9 Exercises

2.1 *Let A be the arithmetic mean of a finite set of positive numbers, with x the smallest of these numbers, and y the largest. Show that*

$$xy \leq A(x + y - A),$$

with equality if and only if $x = y = A$.

2.2 *Minimize the function*

$$f(x) = x^2 + \frac{1}{x^2} + 4x + \frac{4}{x},$$

over positive x. Hint: consider the first two terms and the last two terms separately. Note that the minimum value of $f(x, y)$ is not the one suggested by the AGM Inequality, as applied to the four terms taken together.

2.3 *Find the maximum value of $f(x, y) = x^2 y$, if x and y are restricted to positive real numbers for which $6x + 5y = 45$. Hint: write $6x$ as $3x + 3x$.*

2.4 *Relate Example 4 to eigenvectors and eigenvalues.*

2.5 Young's Inequality *Suppose that p and q are positive numbers greater than one such that $\frac{1}{p} + \frac{1}{q} = 1$. If x and y are positive numbers, then*

$$xy \leq \frac{x^p}{p} + \frac{y^q}{q},$$

with equality if and only if $x^p = y^q$. Hint: use the GAGM Inequality.

2.6 ([126]) *For given constants c and d, find the largest and smallest values of $cx + dy$ taken over all points (x, y) of the ellipse*

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1.$$

2.7 ([126]) *Find the largest and smallest values of $2x + y$ on the circle $x^2 + y^2 = 1$. Where do these values occur? What does this have to do with eigenvectors and eigenvalues?*

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

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

becomes

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

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

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

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

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

Chapter 3

Geometric Programming

Geometric Programming (GP) involves the minimization of functions of a special type, known as posynomials. The first systematic treatment of geometric programming appeared in the book [76], by Duffin, Peterson and Zener, the founders of geometric programming. As we shall see, the Generalized Arithmetic-Geometric Mean Inequality plays an important role in the theoretical treatment of geometric programming.

3.1 An Example of a GP Problem

The following optimization problem was presented originally by Duffin, *et al.* [76] and discussed by Peressini *et al.* in [129]. It illustrates well the type of problem considered in geometric programming. Suppose that 400 cubic yards of gravel must be ferried across a river in an open box of length t_1, width t_2 and height t_3. Each round-trip cost ten cents. The sides and the bottom of the box cost 10 dollars per square yard to build, while the ends of the box cost twenty dollars per square yard. The box will have no salvage value after it has been used. Determine the dimensions of the box that minimize the total cost.

With $t = (t_1, t_2, t_3)$, the cost function is

$$g(t) = \frac{40}{t_1 t_2 t_3} + 20t_1t_3 + 10t_1t_2 + 40t_2t_3, \tag{3.1}$$

which is to be minimized over $t_j > 0$, for $j = 1, 2, 3$. The function $g(t)$ is an example of a posynomial.

3.2 Posynomials and the GP Problem

Functions $g(t)$ of the form

$$g(t) = \sum_{i=1}^{n} c_i \Big(\prod_{j=1}^{m} t_j^{a_{ij}} \Big), \tag{3.2}$$

with $t = (t_1, ..., t_m)$, the $t_j > 0$, $c_i > 0$ and a_{ij} real, are called *posynomials*. The *geometric programming problem*, denoted (GP), is to minimize a given posynomial over positive t. In order for the minimum to be greater than zero, we need some of the a_{ij} to be negative.

We denote by $u_i(t)$ the function

$$u_i(t) = c_i \prod_{j=1}^{m} t_j^{a_{ij}}, \tag{3.3}$$

so that

$$g(t) = \sum_{i=1}^{n} u_i(t). \tag{3.4}$$

For any choice of $\delta_i > 0$, $i = 1, ..., n$, with

$$\sum_{i=1}^{n} \delta_i = 1,$$

we have

$$g(t) = \sum_{i=1}^{n} \delta_i \Big(\frac{u_i(t)}{\delta_i} \Big). \tag{3.5}$$

Applying the Generalized Arithmetic-Geometric Mean (GAGM) Inequality, we have

$$g(t) \geq \prod_{i=1}^{n} \Big(\frac{u_i(t)}{\delta_i} \Big)^{\delta_i}. \tag{3.6}$$

Therefore,

$$g(t) \geq \prod_{i=1}^{n} \Big(\frac{c_i}{\delta_i} \Big)^{\delta_i} \left(\prod_{i=1}^{n} \prod_{j=1}^{m} t_j^{a_{ij}\delta_i} \right), \tag{3.7}$$

or

$$g(t) \geq \prod_{i=1}^{n} \Big(\frac{c_i}{\delta_i} \Big)^{\delta_i} \Big(\prod_{j=1}^{m} t_j^{\sum_{i=1}^{n} a_{ij}\delta_i} \Big), \tag{3.8}$$

Suppose that we can find $\delta_i > 0$ with

$$\sum_{i=1}^{n} a_{ij}\delta_i = 0, \tag{3.9}$$

for each j. Then the inequality in (3.8) becomes

$$g(t) \geq v(\delta), \tag{3.10}$$

for

$$v(\delta) = \prod_{i=1}^{n} \left(\frac{c_i}{\delta_i}\right)^{\delta_i}. \tag{3.11}$$

3.3 The Dual GP Problem

The *dual geometric programming problem*, denoted (DGP), is to maximize the function $v(\delta)$, over all *feasible* $\delta = (\delta_1, ..., \delta_n)$, that is, all positive δ for which

$$\sum_{i=1}^{n} \delta_i = 1, \tag{3.12}$$

and

$$\sum_{i=1}^{n} a_{ij}\delta_i = 0, \tag{3.13}$$

for each $j = 1, ..., m$. Clearly, we have

$$g(t) \geq v(\delta), \tag{3.14}$$

for any positive t and feasible δ. Of course, there may be no feasible δ, in which case (DGP) is said to be *inconsistent*.

As we have seen, the inequality in (3.14) is based on the GAGM Inequality. We have equality in the GAGM Inequality if and only if the terms in the arithmetic mean are all equal. In this case, this says that there is a constant λ such that

$$\frac{u_i(t)}{\delta_i} = \lambda, \tag{3.15}$$

for each $i = 1, ..., n$. Using the fact that the δ_i sum to one, it follows that

$$\lambda = \sum_{i=1}^{n} u_i(t) = g(t), \tag{3.16}$$

and

$$\delta_i = \frac{u_i(t)}{g(t)}, \tag{3.17}$$

for each $i = 1, ..., n$. As the theorem below asserts, if t^* is positive and minimizes $g(t)$, then δ^*, the associated δ from Equation (3.17), is feasible and solves (DGP). Since we have equality in the GAGM Inequality now, we have

$$g(t^*) = v(\delta^*).$$

The main theorem in geometric programming is the following.

Theorem 3.1 *If $t^* > 0$ minimizes $g(t)$, then (DGP) is consistent. In addition, the choice*

$$\delta_i^* = \frac{u_i(t^*)}{g(t^*)} \tag{3.18}$$

is feasible and solves (DGP). Finally,

$$g(t^*) = v(\delta^*); \tag{3.19}$$

that is, there is no duality gap.

Proof: We have

$$\frac{\partial u_i(t^*)}{\partial t_j} = \frac{a_{ij} u_i(t^*)}{t_j^*}, \tag{3.20}$$

so that

$$t_j^* \frac{\partial u_i(t^*)}{\partial t_j} = a_{ij} u_i(t^*), \tag{3.21}$$

for each $j = 1, ..., m$. Since t^* minimizes $g(t)$, we have

$$0 = \frac{\partial g(t^*)}{\partial t_j} = \sum_{i=1}^{n} \frac{\partial u_i(t^*)}{\partial t_j}, \tag{3.22}$$

so that, from Equation (3.21), we have

$$0 = \sum_{i=1}^{n} a_{ij} u_i(t^*), \tag{3.23}$$

for each $j = 1, ..., m$. It follows that δ^* is feasible. Since we have equality in the GAGM Inequality, we know

$$g(t^*) = v(\delta^*). \tag{3.24}$$

Therefore, δ^* solves (DGP). This completes the proof. ∎

3.4 Solving the GP Problem

The theorem suggests how we might go about solving (GP). First, we try to find a feasible δ^* that maximizes $v(\delta)$. This means we have to find a positive solution to the system of $m+1$ linear equations in n unknowns, given by

$$\sum_{i=1}^{n} \delta_i = 1, \tag{3.25}$$

and

$$\sum_{i=1}^{n} a_{ij}\delta_i = 0, \tag{3.26}$$

for $j = 1, ..., m$, such that $v(\delta)$ is maximized. In a later chapter on the MART and SMART algorithms we shall discuss in some detail iterative procedures for finding such δ. If there is no such vector, then (GP) has no minimizer. Once the desired δ^* has been found, we set

$$\delta_i^* = \frac{u_i(t^*)}{v(\delta^*)}, \tag{3.27}$$

for each $i = 1, ..., n$, and then solve for the entries of t^*. This last step can be simplified by taking logs; then we have a system of linear equations to solve for the values $\log t_j^*$.

3.5 Solving the DGP Problem

The iterative multiplicative algebraic reconstruction technique MART can be used to minimize the function $v(\delta)$, subject to linear equality constraints, provided that the matrix involved has nonnegative entries. We cannot apply the MART yet, because the matrix A^T does not satisfy these conditions.

3.5.1 The MART

The Kullback-Leibler, or KL distance [108] between positive numbers a and b is

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

We also define $KL(a, 0) = +\infty$ and $KL(0, b) = b$. Extending to nonnegative vectors $a = (a_1, ..., a_J)^T$ and $b = (b_1, ..., b_J)^T$, we have

$$KL(a, b) = \sum_{j=1}^{J} KL(a_j, b_j) = \sum_{j=1}^{J} \Big(a_j \log \frac{a_j}{b_j} + b_j - a_j \Big).$$

The MART is an iterative algorithm for finding a non-negative solution of the system $Px = y$, for an I by J matrix P with non-negative entries and vector y with positive entries. We also assume that

$$p_j = \sum_{i=1}^{I} P_{ij} > 0,$$

for all $i = 1, ..., I$. When discussing the MART, we say that the system $Px = y$ is *consistent* when it has non-negative solutions. We consider two different versions of the MART.

MART I

The iterative step of the first version of MART, which we shall call MART I, is the following: for $k = 0, 1, ...,$ and $i = k(\mathrm{mod}\, I) + 1$, let

$$x_j^{k+1} = x_j^k \Big(\frac{y_i}{(Px^k)_i} \Big)^{P_{ij}/m_i},$$

for $j = 1, ..., J$, where the parameter m_i is defined to be

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

The MART I algorithm converges, in the consistent case, to the non-negative solution for which the KL distance $KL(x, x^0)$ is minimized.

MART II

The iterative step of the second version of MART, which we shall call MART II, is the following: for $k = 0, 1, ...,$ and $i = k(\mathrm{mod}\, I) + 1$, let

$$x_j^{k+1} = x_j^k \Big(\frac{y_i}{(Px^k)_i} \Big)^{P_{ij}/p_j n_i},$$

for $j = 1, ..., J$, where the parameter n_i is defined to be

$$n_i = \max\{P_{ij} p_j^{-1} | j = 1, ..., J\}.$$

The MART II algorithm converges, in the consistent case, to the non-negative solution for which the KL distance

$$\sum_{j=1}^{J} p_j KL(x_j, x_j^0)$$

is minimized.

3.5.2 Using the MART to Solve the DGP Problem

The entries on the bottom row of A^T are all one, as is the bottom entry of the column vector u, since these entries correspond to the equation $\sum_{i=1}^{I} \delta_i = 1$. By adding suitably large positive multiples of this last equation to the other equations in the system, we obtain an equivalent system, $B^T\delta = s$, for which the new matrix B^T and the new vector s have only positive entries. Now we can apply the MART I algorithm to the system $B^T\delta = s$, letting $P = B^T$, $p_i = \sum_{j=1}^{J+1} B_{ij}$, $\delta = x$, $x^0 = c$ and $y = s$. In the consistent case, the MART I algorithm will find the non-negative solution that minimizes $KL(x, x^0)$, so we select $x^0 = c$. Then the MART I algorithm finds the non-negative δ^* satisfying $B^T\delta^* = s$, or, equivalently, $A^T\delta^* = u$, for which the KL distance

$$KL(\delta, c) = \sum_{i=1}^{I} \Big(\delta_i \log \frac{\delta_i}{c_i} + c_i - \delta_i\Big)$$

is minimized. Since we know that

$$\sum_{i=1}^{I} \delta_i = 1,$$

it follows that minimizing $KL(\delta, c)$ is equivalent to maximizing $v(\delta)$. Using δ^*, we find the optimal t^* solving the GP problem.

For example, the linear system of equations $A^T\delta = u$ corresponding to the posynomial in Equation (3.1) is

$$A^T\delta = u = \begin{bmatrix} -1 & 1 & 1 & 0 \\ -1 & 0 & 1 & 1 \\ -1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Adding two times the last row to the other rows, the system becomes

$$B^T\delta = s = \begin{bmatrix} 1 & 3 & 3 & 2 \\ 1 & 2 & 3 & 3 \\ 1 & 3 & 2 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 1 \end{bmatrix}.$$

The matrix B^T and the vector s are now positive. We are ready to apply the MART.

The MART iteration is as follows. With $j = k(\mathrm{mod}\, J + 1) + 1$, $m_j = \max\{B_{ij} \,|i = 1, 2, ..., I\}$ and $k = 0, 1, ...$, let

$$\delta_i^{k+1} = \delta_i^k \Big(\frac{s_j}{(B^T\delta^k)_j}\Big)^{m_j^{-1}B_{ij}}.$$

The optimal δ^* is $\delta^* = (.4, .2, .2, .2)^T$, the optimal t^* is $t^* = (2, 1, .5)$, and the lowest cost is one hundred dollars.

3.6 Constrained Geometric Programming

Consider now the following variant of the problem of transporting the gravel across the river. Suppose that the bottom and the two sides will be constructed for free from scrap metal, but only four square yards are available. The cost function to be minimized becomes

$$g_0(t) = \frac{40}{t_1 t_2 t_3} + 40 t_2 t_3, \tag{3.28}$$

and the constraint is

$$g_1(t) = \frac{t_1 t_3}{2} + \frac{t_1 t_2}{4} \leq 1. \tag{3.29}$$

With $\delta_1 > 0$, $\delta_2 > 0$, and $\delta_1 + \delta_2 = 1$, we write

$$g_0(t) = \delta_1 \frac{40}{\delta_1 t_1 t_2 t_3} + \delta_2 \frac{40 t_2 t_3}{\delta_2}. \tag{3.30}$$

Since $0 \leq g_1(t) \leq 1$, we have

$$g_0(t) \geq \Big(\delta_1 \frac{40}{\delta_1 t_1 t_2 t_3} + \delta_2 \frac{40 t_2 t_3}{\delta_2}\Big)\Big(g_1(t)\Big)^{\lambda}, \tag{3.31}$$

for any positive λ. The GAGM Inequality then tells us that

$$g_0(t) \geq \left(\Big(\frac{40}{\delta_1 t_1 t_2 t_3}\Big)^{\delta_1}\Big(\frac{40 t_2 t_3}{\delta_2}\Big)^{\delta_2}\right)\Big(g_1(t)\Big)^{\lambda}, \tag{3.32}$$

so that

$$g_0(t) \geq \left(\Big(\frac{40}{\delta_1}\Big)^{\delta_1}\Big(\frac{40}{\delta_2}\Big)^{\delta_2}\right) t_1^{-\delta_1} t_2^{\delta_2 - \delta_1} t_3^{\delta_2 - \delta_1}\Big(g_1(t)\Big)^{\lambda}. \tag{3.33}$$

From the GAGM Inequality, we also know that, for $\delta_3 > 0$, $\delta_4 > 0$ and $\lambda = \delta_3 + \delta_4$,

$$\Big(g_1(t)\Big)^{\lambda} \geq (\lambda)^{\lambda}\left(\Big(\frac{1}{2\delta_3}\Big)^{\delta_3}\Big(\frac{1}{4\delta_4}\Big)^{\delta_4}\right) t_1^{\delta_3+\delta_4} t_2^{\delta_4} t_3^{\delta_3}. \tag{3.34}$$

Combining the inequalities in (3.33) and (3.34), we obtain

$$g_0(t) \geq v(\delta) t_1^{-\delta_1+\delta_3+\delta_4} t_2^{-\delta_1+\delta_2+\delta_4} t_3^{-\delta_1+\delta_2+\delta_3}, \tag{3.35}$$

with

$$v(\delta) = \Big(\frac{40}{\delta_1}\Big)^{\delta_1}\Big(\frac{40}{\delta_2}\Big)^{\delta_2}\Big(\frac{1}{2\delta_3}\Big)^{\delta_3}\Big(\frac{1}{4\delta_4}\Big)^{\delta_4}\Big(\delta_3 + \delta_4\Big)^{\delta_3+\delta_4}, \tag{3.36}$$

and $\delta = (\delta_1, \delta_2, \delta_3, \delta_4)$. If we can find a positive vector δ with

$$\begin{aligned} \delta_1 + \delta_2 &= 1, \\ \delta_3 + \delta_4 &= \lambda, \\ -\delta_1 + \delta_3 + \delta_4 &= 0, \\ -\delta_1 + \delta_2 + \delta_4 &= 0 \\ -\delta_1 + \delta_2 + \delta_3 &= 0, \end{aligned} \tag{3.37}$$

then

$$g_0(t) \geq v(\delta). \tag{3.38}$$

In this particular case, there is a unique positive δ satisfying the equations (3.37), namely

$$\delta_1^* = \frac{2}{3}, \delta_2^* = \frac{1}{3}, \delta_3^* = \frac{1}{3}, \text{and } \delta_4^* = \frac{1}{3}, \tag{3.39}$$

and

$$v(\delta^*) = 60. \tag{3.40}$$

Therefore, $g_0(t)$ is bounded below by 60. If there is t^* such that

$$g_0(t^*) = 60, \tag{3.41}$$

then we must have

$$g_1(t^*) = 1, \tag{3.42}$$

and equality in the GAGM Inequality. Consequently,

$$\frac{3}{2}\frac{40}{t_1^* t_2^* t_3^*} = 3(40 t_2^* t_3^*) = 60, \tag{3.43}$$

and

$$\frac{3}{2} t_1^* t_3^* = \frac{3}{4} t_1^* t_2^* = K. \tag{3.44}$$

Since $g_1(t^*) = 1$, we must have $K = \frac{3}{2}$. We solve these equations by taking logarithms, to obtain the solution

$$t_1^* = 2,\ t_2^* = 1, \text{ and } t_3^* = \frac{1}{2}. \tag{3.45}$$

The change of variables $t_j = e^{x_j}$ converts the constrained (GP) problem into a constrained convex programming problem. The theory of the constrained (GP) problem can then be obtained as a consequence of the theory for the convex problem, which we shall consider in a later chapter.

3.7 Exercises

3.1 *Minimize the function*

$$g(t_1, t_2) = \frac{2}{t_1 t_2} + t_1 t_2 + t_1, \tag{3.46}$$

over $t_1 > 0$, $t_2 > 0$.

3.2 *Minimize the function*

$$g(t_1, t_2) = \frac{1}{t_1 t_2} + t_1 t_2 + t_1 + t_2, \tag{3.47}$$

over $t_1 > 0$, $t_2 > 0$.

3.3 *Minimize the function*

$$g(t_1, t_2, t_3) = \frac{40}{t_1 t_2 t_3} + 20 t_1 t_3 + 10 t_1 t_2 + 40 t_2 t_3, \tag{3.48}$$

over $t_j > 0$, *for* $j = 1, 2, 3$.

Chapter 4

Convex Sets

Convex sets and convex functions play important roles in optimization. In this chapter we survey the basic facts concerning the geometry of convex sets. We begin with the geometry of R^J.

4.1 The Geometry of Real Euclidean Space

We denote by R^J the real Euclidean space consisting of all J-dimensional column vectors $x = (x_1, ..., x_J)^T$ with real entries x_j; here the superscript T denotes the transpose of the 1 by J matrix (or, row vector) $(x_1, ..., x_J)$.

4.1.1 Inner Products

For $x = (x_1, ..., x_J)^T$ and $y = (y_1, ..., y_J)^T$ in R^J, the dot product $x \cdot y$ is defined to be

$$x \cdot y = \sum_{j=1}^{J} x_j y_j. \tag{4.1}$$

Note that we can write

$$x \cdot y = y^T x = x^T y, \tag{4.2}$$

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

$$||x||_2 = \sqrt{x \cdot x} = \sqrt{x^T x}. \tag{4.3}$$

The *Euclidean distance* between two vectors x and y in R^J is $||x - y||_2$.

The space R^J, along with its dot product, is an example of a finite-dimensional Hilbert space.

Definition 4.1 *Let V be a real vector space. The scalar-valued function $\langle u, v\rangle$ is called an inner product on V if the following four properties hold, for all u, w, and v in V, and all real c:*

$$\langle u + w, v\rangle = \langle u, v\rangle + \langle w, v\rangle; \tag{4.4}$$

$$\langle cu, v\rangle = c\langle u, v\rangle; \tag{4.5}$$

$$\langle v, u\rangle = \langle u, v\rangle; \tag{4.6}$$

and

$$\langle u, u\rangle \geq 0, \tag{4.7}$$

with equality in Inequality (4.7) if and only if $u = 0$.

The dot product of vectors is an example of an inner product. The properties of an inner product are precisely the ones needed to prove Cauchy's Inequality, which then holds for any inner product. We shall favor the dot product notation $u \cdot v$ for the inner product of vectors, although we shall occasionally use the matrix multiplication form, $v^T u$ or the inner product notation $\langle u, v\rangle$.

4.1.2 Cauchy's Inequality

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

$$|\langle x, y\rangle| \leq ||x||_2 ||y||_2, \tag{4.8}$$

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

A simple application of Cauchy's inequality gives us

$$||x + y||_2 \leq ||x||_2 + ||y||_2; \tag{4.9}$$

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

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

$$||x + y||_2^2 + ||x - y||_2^2 = 2||x||_2^2 + 2||y||_2^2. \tag{4.10}$$

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

4.2 A Bit of Topology

Having the norm allows us to define the distance between two points x and y in R^J as $||x - y||$. Being able to talk about how close points are to each other enables us to define continuity of functions on R^J and to consider topological notions of closed set, open set, interior of a set and boundary of a set.

Definition 4.2 *A subset B of R^J is closed if, whenever x^k is in B for each non-negative integer k and $||x - x^k|| \to 0$, as $k \to +\infty$, then x is in B.*

For example, $B = [0, 1]$ is closed as a subset of R, but $B = (0, 1)$ is not.

Definition 4.3 *We say that $d \geq 0$ is the distance from the point x to the set B if, for every $\epsilon > 0$, there is b_ϵ in B, with $||x - b_\epsilon||_2 < d + \epsilon$, and no b in B with $||x - b||_2 < d$.*

The distance from the point 0 in R to the set $(0, 1)$ is zero, while its distance to the set $(1, 2)$ is one. It follows easily from the definitions that, if B is closed and $d = 0$, then x is in B.

Definition 4.4 *The closure of a set B is the set of all points x whose distance from B is zero.*

The closure of the interval $B = (0, 1)$ is $[0, 1]$.

Definition 4.5 *A subset U of R^J is open if its complement, the set of all points not in U, is closed.*

Definition 4.6 *Let C be a subset of R^J. A point x in C is said to be an interior point of set C if there is $\epsilon > 0$ such that every point z with $||x - z|| < \epsilon$ is in C. The interior of the set C, written int(C), is the set of all interior points of C. It is also the largest open set contained within C.*

For example, the open interval $(0, 1)$ is the interior of the intervals $(0, 1]$ and $[0, 1]$. A set C is open if and only if $C = \text{int}(C)$.

Definition 4.7 *A point x in R^J is said to be a boundary point of set C if, for every $\epsilon > 0$, there are points y_ϵ in C and z_ϵ not in C, both depending on the choice of ϵ, with $||x - y_\epsilon|| < \epsilon$ and $||x - z_\epsilon|| < \epsilon$. The boundary of C is the set of all boundary points of C. It is also the intersection of the closure of C with the closure of its complement.*

For example, the points $x = 0$ and $x = 1$ are boundary points of the set $(0, 1]$.

Definition 4.8 *For $k = 0, 1, 2, ...$, let x^k be a vector in R^J. The sequence of vectors $\{x^k\}$ is said to converge to the vector z if, given any $\epsilon > 0$, there is positive integer n, usually depending on ϵ, such that, for every $k > n$, we have $||z - x^k|| \leq \epsilon$. Then we say that z is the limit of the sequence.*

For example, the sequence $\{x^k = \frac{1}{k+1}\}$ in R converges to $z = 0$. The sequence $\{(-1)^k\}$ alternates between 1 and -1, so does not converge. However, the subsequence associated with odd k converges to $z = -1$, while the subsequence associated with even k converges to $z = 1$. The values $z = -1$ and $z = 1$ are called *subsequential limit points*, or, sometimes, *cluster points* of the sequence.

Definition 4.9 *A sequence $\{x^k\}$ of vectors in R^J is said to be bounded if there is a constant $b > 0$, such that $||x^k|| \leq b$, for all k.*

A fundamental result in analysis is the following.

Proposition 4.1 *Every convergent sequence of vectors in R^J is bounded. Every bounded sequence of vectors in R^J has at least one convergent subsequence, therefore, has at least one cluster point.*

4.3 Convex Sets in R^J

In preparation for our discussion of linear and nonlinear programming, we consider some of the basic concepts from the geometry of convex sets.

4.3.1 Basic Definitions

We begin with the basic definitions.

Definition 4.10 *A vector z is said to be a convex combination of the vectors x and y if there is α in the interval $[0, 1]$ such that $z = (1 - \alpha)x + \alpha y$.*

Definition 4.11 *A nonempty set C in R^J is said to be convex if, for any distinct points x and y in C, and for any real number α in the interval $(0, 1)$, the point $(1 - \alpha)x + \alpha y$ is also in C; that is, C is closed to convex combinations.*

For example, the unit ball B in R^J, consisting of all x with $||x||_2 \leq 1$, is convex, while the surface of the ball, the set of all x with $||x||_2 = 1$, is not convex.

Definition 4.12 *The convex hull of a set S, denoted conv(S), is the smallest convex set containing S.*

Proposition 4.2 *The convex hull of a set S is the set C of all convex combinations of members of S.*

Definition 4.13 *A subset S of R^J is a subspace if, for every x and y in S and scalars α and β, the linear combination $\alpha x + \beta y$ is again in S.*

A subspace is necessarily a convex set.

Definition 4.14 *The orthogonal complement of a subspace S is the set*

$$S^{\perp} = \{u | u^T s = 0, \text{for every } s \in S\}, \tag{4.11}$$

the set of all vectors u in R^J that are orthogonal to every member of S.

For example, in R^3, the x, y-plane is a subspace and has for its orthogonal complement the z-axis.

Definition 4.15 *A subset M of R^J is a linear manifold if there is a subspace S and a vector b such that*

$$M = S + b = \{x | x = s + b, \text{for some } s \text{ in } S\}.$$

Any linear manifold is convex.

Definition 4.16 *For a fixed column vector a with Euclidean length one and a fixed scalar γ the hyperplane determined by a and γ is the set*

$$H(a, \gamma) = \{z | \langle a, z \rangle = \gamma\}.$$

The hyperplanes $H(a, \gamma)$ are linear manifolds, and the hyperplanes $H(a, 0)$ are subspaces.

Definition 4.17 *Given a subset C of R^J, the affine hull of C, denoted aff(C), is the smallest linear manifold containing C.*

For example, let C be the line segment connecting the two points $(0, 1)$ and $(1, 2)$ in R^2. The affine hull of C is the straight line whose equation is $y = x + 1$.

Definition 4.18 *The dimension of a subset of R^J is the dimension of its affine hull, which is the dimension of the subspace of which it is a translate.*

The set C above has dimension one. A set containing only one point is its own affine hull, since it is a translate of the subspace $\{0\}$.

In R^2, the line segment connecting the points $(0, 1)$ and $(1, 2)$ has no interior; it is a one-dimensional subset of a two-dimensional space and can contain no two-dimensional ball. But, the part of this set without its two end points is a sort of interior, called the *relative interior*.

Definition 4.19 *The relative interior of a subset C of R^J, denoted ri(C), is the interior of C, as defined by considering C as a subset of its affine hull.*

Since a set consisting of a single point is its own affine hull, it is its own relative interior.

Definition 4.20 *A point x in a convex set C is said to be an extreme point of C if the set obtained by removing x from C remains convex.*

Said another way, $x \in C$ is an extreme point of C if x cannot be written as

$$x = (1 - \alpha)y + \alpha z, \tag{4.12}$$

for $y, z \neq x$ and $\alpha \in (0, 1)$. For example, the point $x = 1$ is an extreme point of the convex set $C = [0, 1]$. Every point on the boundary of a sphere in R^J is an extreme point of the sphere. The set of all extreme points of a convex set is denoted Ext(C).

Definition 4.21 *A non-zero vector d is said to be a direction of unboundedness of a convex set C if, for all x in C and all $\gamma \geq 0$, the vector $x + \gamma d$ is in C.*

For example, if C is the non-negative orthant in R^J, then any non-negative vector d is a direction of unboundedness.

Definition 4.22 *A vector a is normal to a convex set C at the point s in C if*

$$\langle a, c - s \rangle \leq 0, \tag{4.13}$$

for all c in C.

Definition 4.23 *Let C be convex and s in C. The normal cone to C at s, denoted $N_C(s)$, is the set of all vectors a that are normal to C at s.*

4.3.2 Orthogonal Projection onto Convex Sets

The following proposition is fundamental in the study of convexity and can be found in most books on the subject; see, for example, the text by Goebel and Reich [91].

Proposition 4.3 *Given any nonempty closed convex set C and an arbitrary vector x in R^J, there is a unique member of C closest to x, denoted $P_C x$, the orthogonal (or metric) projection of x onto C.*

Proof: If x is in C, then $P_C x = x$, so assume that x is not in C. Then $d > 0$, where d is the distance from x to C. For each positive integer n, select c_n in C with $||x - c_n||_2 < d + \frac{1}{n}$. Then the sequence $\{c_n\}$ is bounded; let c^* be any cluster point. It follows easily that $||x - c^*||_2 = d$ and that c^* is in C. If there is any other member c of C with $||x - c||_2 = d$, then, by the Parallelogram Law, we would have $||x - (c^* + c)/2||_2 < d$, which is a contradiction. Therefore, c^* is $P_C x$. ∎

For example, if $C = U$, the unit ball, then $P_C x = x/||x||_2$, for all x such that $||x||_2 > 1$, and $P_C x = x$ otherwise. If C is R^J_+, the nonnegative cone of R^J, consisting of all vectors x with $x_j \geq 0$, for each j, then $P_C x = x_+$, the vector whose entries are $\max(x_j, 0)$. For any closed, convex set C, the distance from x to C is $||x - P_C x||$.

If a nonempty set S is not convex, then the orthogonal projection of a vector x onto S need not be well defined; there may be more than one vector in S closest to x. In fact, it is known that a set S is convex if and only if, for every x not in S, there is a unique point in S closest to x; this is Motzkin's Theorem (see [16], p. 447). Note that there may well be some x for which there is a unique closest point in S, but if S is not convex, then there must be at least one point without a unique closest point in S.

Lemma 4.1 *For $H = H(a, \gamma)$, $z = P_H x$ is the vector*

$$z = P_H x = x + (\gamma - \langle a, x \rangle)a. \tag{4.14}$$

We shall use this fact in our discussion of the ART algorithm.

For an arbitrary nonempty closed convex set C in R^J, the orthogonal projection $T = P_C$ is a nonlinear operator, unless, of course, C is a subspace. We may not be able to describe $P_C x$ explicitly, but we do know a useful property of $P_C x$.

Proposition 4.4 *For a given x, a vector z in C is $P_C x$ if and only if*

$$\langle c - z, z - x \rangle \geq 0, \tag{4.15}$$

for all c in the set C.

Proof: Let c be arbitrary in C and α in $(0, 1)$. Then

$$||x - P_C x||_2^2 \leq ||x - (1 - \alpha)P_C x - \alpha c||_2^2 = ||x - P_C x + \alpha(P_C x - c)||_2^2$$

$$= ||x - P_C x||_2^2 - 2\alpha\langle x - P_C x, c - P_C x \rangle + \alpha^2 ||P_C x - c||_2^2. \tag{4.16}$$

Therefore,

$$-2\alpha\langle x - P_C x, c - P_C x \rangle + \alpha^2 ||P_C x - c||_2^2 \geq 0, \tag{4.17}$$

so that

$$2\langle x - P_C x, c - P_C x\rangle \leq \alpha ||P_C x - c||_2^2. \tag{4.18}$$

Taking the limit, as $\alpha \to 0$, we conclude that

$$\langle c - P_C x, P_C x - x\rangle \geq 0. \tag{4.19}$$

If z is a member of C that also has the property

$$\langle c - z, z - x\rangle \geq 0, \tag{4.20}$$

for all c in C, then we have both

$$\langle z - P_C x, P_C x - x\rangle \geq 0, \tag{4.21}$$

and

$$\langle z - P_C x, x - z\rangle \geq 0. \tag{4.22}$$

Adding on both sides of these two inequalities lead to

$$\langle z - P_C x, P_C x - z\rangle \geq 0. \tag{4.23}$$

But,

$$\langle z - P_C x, P_C x - z\rangle = -||z - P_C x||_2^2, \tag{4.24}$$

so it must be the case that $z = P_C x$. This completes the proof. ∎

4.4 Some Results on Projections

The characterization of the orthogonal projection operator P_C given by Proposition 4.4 has a number of important consequences.

Corollary 4.1 *Let S be any subspace of R^J. Then, for any x in R^J and s in S, we have*

$$\langle P_S x - x, s\rangle = 0. \tag{4.25}$$

Proof: Since S is a subspace, $s + P_S x$ is again in S, for all s, as is cs, for every scalar c. ∎

This corollary enables us to prove the Decomposition Theorem.

Theorem 4.1 *Let S be any subspace of R^J and x any member of R^J. Then there are unique vectors s in S and u in $S^\perp$ such that $x = s + u$. The vector s is $P_S x$ and the vector u is $P_{S^\perp} x$.*

Proof: For the given x we take $s = P_S x$ and $u = x - P_S x$. Corollary 4.1 assures us that u is in $S^\perp$. Now we need to show that this decomposition is unique. To that end, suppose that we can write $x = s_1 + u_1$, with s_1 in S and u_1 in $S^\perp$. Then Proposition 4.4 tells us that, since $s_1 - x$ is orthogonal to every member of S, s_1 must be $P_S x$. ∎

This theorem is often presented in a slightly different manner.

Theorem 4.2 *Let A be a real I by J matrix. Then every vector b in R^I can be written uniquely as $b = Ax + w$, where $A^T w = 0$.*

To derive Theorem 4.2 from Theorem 4.1, we simply let $S = \{Ax | x \in R^J\}$. Then $S^\perp$ is the set of all w such that $A^T w = 0$. It follows that w is the member of the null space of A^T closest to b.

Here are additional consequences of Proposition 4.4.

Corollary 4.2 *Let S be any subspace of R^J, d a fixed vector, and V the linear manifold $V = S + d = \{v = s + d | s \in S\}$, obtained by translating the members of S by the vector d. Then, for every x in R^J and every v in V, we have*

$$\langle P_V x - x, v - P_V x \rangle = 0. \tag{4.26}$$

Proof: Since v and $P_V x$ are in V, they have the form $v = s + d$, and $P_V x = \hat{s} + d$, for some s and $\hat{s}$ in S. Then $v - P_V x = s - \hat{s}$. ∎

Corollary 4.3 *Let H be the hyperplane $H(a, \gamma)$. Then, for every x, and every h in H, we have*

$$\langle P_H x - x, h - P_H x \rangle = 0. \tag{4.27}$$

Corollary 4.4 *Let S be a subspace of R^J. Then $(S^\perp)^\perp = S$.*

Proof: Every x in R^J has the form $x = s + u$, with s in S and u in $S^\perp$. Suppose x is in $(S^\perp)^\perp$. Then $u = 0$. ∎

4.5 Linear and Affine Operators on R^J

If A is a J by J real matrix, then we can define an operator T by setting $Tx = Ax$, for each x in R^J; here Ax denotes the multiplication of the matrix A and the column vector x.

Definition 4.24 *An operator T is said to be a linear operator if*

$$T(\alpha x + \beta y) = \alpha T x + \beta T y, \tag{4.28}$$

for each pair of vectors x and y and each pair of scalars α and β.

Any operator T that comes from matrix multiplication, that is, for which $Tx = Ax$, is linear.

Lemma 4.2 *For $H = H(a, \gamma)$, $H_0 = H(a, 0)$, and any x and y in R^J, we have*

$$P_H(x + y) = P_H x + P_H y - P_H 0, \tag{4.29}$$

so that

$$P_{H_0}(x + y) = P_{H_0} x + P_{H_0} y, \tag{4.30}$$

that is, the operator P_{H_0} is an additive operator. In addition,

$$P_{H_0}(\alpha x) = \alpha P_{H_0} x, \tag{4.31}$$

so that P_{H_0} is a linear operator.

Definition 4.25 *If A is a J by J real matrix and d is a fixed nonzero vector in R^J, the operator defined by $Tx = Ax + d$ is an affine linear operator.*

Lemma 4.3 *For any hyperplane $H = H(a, \gamma)$ and $H_0 = H(a, 0)$,*

$$P_H x = P_{H_0} x + P_H 0, \tag{4.32}$$

so P_H is an affine linear operator.

Lemma 4.4 *For $i = 1, ..., I$ let H_i be the hyperplane $H_i = H(a^i, \gamma_i)$, $H_{i0} = H(a^i, 0)$, and P_i and P_{i0} the orthogonal projections onto H_i and H_{i0}, respectively. Let T be the operator $T = P_I P_{I-1} \cdots P_2 P_1$. Then $Tx = Bx + d$, for some square matrix B and vector d; that is, T is an affine linear operator.*

4.6 The Fundamental Theorems

The Separation Theorem and the Support Theorem provide the foundation for the geometric approach to the calculus of functions of several variables.

A real-valued function $f(x)$ defined for real x has a derivative at $x = x_0$ if and only if there is a unique line through the point $(x_0, f(x_0))$ tangent to the graph of $f(x)$ at that point. If $f(x)$ is not differentiable at x_0, there may be more than one such tangent line, as happens with the function $f(x) = |x|$ at $x_0 = 0$. For functions of several variables the geometric view of differentiation involves tangent hyperplanes.

4.6.1 Basic Definitions

Definition 4.26 *Let S be a subset of R^J and $f : S \to [-\infty, \infty]$ a function defined on S. The subset of R^{J+1} defined by*

$$\text{epi}(f) = \{(x, \gamma) | f(x) \leq \gamma\}$$

is the epi-graph of f. Then we say that f is convex if its epi-graph is a convex set.

Alternative definitions of convex function are presented in the exercises.

Definition 4.27 *The effective domain of a convex function f, denoted* $\text{dom}(f)$*, is the projection onto R^J of its epi-graph; that is,*

$$\text{dom}(f) = \{x |\, (x, \gamma) \in \text{epi}(f)\} = \{x |\, f(x) < +\infty\}.$$

The effective domain of a convex function is a convex set.

Definition 4.28 *A convex function $f(x)$ is proper if there is no x for which $f(x) = -\infty$ and some x for which $f(x) < +\infty$.*

The important role played by hyperplanes tangent to the epigraph of f motivates our study of the relationship between hyperplanes and convex sets.

4.6.2 The Separation Theorem

The Separation Theorem, sometimes called the Geometric Hahn-Banach Theorem, is an easy consequence of the existence of orthogonal projections onto closed convex sets.

Theorem 4.3 (The Separation Theorem) *Let C be a closed nonempty convex set in R^J and x a point not in C. Then there is non-zero vector a in R^J and real number α such that*

$$\langle a, c\rangle \leq \alpha < \langle a, x\rangle,$$

for every c in C.

Proof: Let $z = P_C x$, $a = x - z$, and $\alpha = \langle a, z\rangle$. Then using Proposition 4.4, we have

$$\langle -a, c - z\rangle \geq 0,$$

or, equivalently,

$$\langle a, c\rangle \leq \langle a, z\rangle = \alpha\,,$$

for all c in C. But, we also have

$$\langle a, x\rangle = \langle a, x - z\rangle + \langle a, z\rangle = ||x - z||^2 + \alpha > \alpha.$$

This completes the proof. ∎

4.6.3 The Support Theorem

The Separation Theorem concerns a closed convex set C and a point x outside the set C, and asserts the existence of a hyperplane separating the two. Now we concerned with a point z on the boundary of a convex set C, such as the point $(x, f(x))$ on the boundary of the epigraph of f. The Support Theorem asserts the existence of a hyperplane through such a point, having the convex set entirely contained in one of its half-spaces. If we knew a priori that the point z is $P_C x$ for some x outside C, then we could simply take the vector $a = x - z$ as the normal to the desired hyperplane. The essence of the Support Theorem is to provide such a normal vector without assuming that $z = P_C x$.

For the proofs that follow we shall need the following definitions.

Definition 4.29 *For subsets A and B of R^J, and scalar γ, let the set $A + B$ consist of all vectors v of the form $v = a + b$, and γA consist of all vectors w of the form $w = \gamma a$, for some a in A and b in B. Let x be a fixed member of R^J. Then the set $x + A$ is the set of all vectors y such that $y = x + a$, for some a in A.*

Lemma 4.5 *Let B be the unit ball in R^J, that is, B is the set of all vectors u with $||u|| \leq 1$. Let S be an arbitrary subset of R^J. Then x is in the interior of S if and only if there is some $\epsilon > 0$ such that $x + \epsilon B \subseteq S$, and y is in the closure of S if and only if, for every $\epsilon > 0$, the set $y + \epsilon B$ has nonempty intersection with S.*

We begin with the *Accessibility Lemma*. Note that the relative interior of any non-empty convex set is always non-empty (see [133], Theorem 6.2).

Lemma 4.6 (The Accessibility Lemma) *Let C be a convex set. Let x be in the relative interior of C and y in the closure of C. Then, for all scalars α in the interval $(0, 1]$, the point $(1 - \alpha)x + \alpha y$ is in the relative interior of C.*

Proof: If the dimension of C is less than J, we can transform the problem into a space of smaller dimension. Therefore, without loss of generality, we can assume that the dimension of C is J, its affine hull is all of R^J, and its relative interior is its interior. Let α be fixed, and $B = \{z \,|\, ||z|| \leq 1\}$. We have to show that there is some $\epsilon > 0$ such that the set $(1-\alpha)x + \alpha y + \epsilon B$ is a subset of the set C. We know that y is in the set $C + \epsilon B$ for every $\epsilon > 0$, since y is in the closure of C. Therefore, for all $\epsilon > 0$ we have

$$(1 - \alpha)x + \alpha y + \epsilon B \subseteq (1 - \alpha)x + \alpha(C + \epsilon B) + \epsilon B$$

$$= (1 - \alpha)x + (1 + \alpha)\epsilon B + \alpha C$$

$$= (1 - \alpha)[x + \epsilon(1 + \alpha)(1 - \alpha)^{-1} B] + \alpha C.$$

Since x is in the interior of the set C, we know that

$$[x + \epsilon(1+\alpha)(1-\alpha)^{-1}B] \subseteq C,$$

for ϵ small enough. This completes the proof. ∎

Now we come to the Support Theorem.

Theorem 4.4 (Support Theorem) *Let C be convex, and let z be on the boundary of C. Then there is a non-zero vector a in R^J with $\langle a, z\rangle \geq \langle a, c\rangle$, for all c in C.*

Proof: If the dimension of C is less than J, then every point of C is on the boundary of C. Let the affine hull of C be $M = S + b$. Then the set $C - b$ is contained in the subspace S, which, in turn, can be contained in a hyperplane through the origin, $H(a, 0)$. Then

$$\langle a, c\rangle = \langle a, b\rangle,$$

for all c in C. So we focus on the case in which the dimension of C is J, in which case the interior of C must be non-empty.

Let y be in the interior of C, and, for each $s > 1$, let $z_s = y + s(z - y)$. Note that z_s is not in the closure of C, for any $s > 1$, by the Accessibility Lemma, since z is not in the interior of C. By the Separation Theorem, there are vectors b_s such that

$$\langle b_s, c\rangle < \langle b_s, z_s\rangle,$$

for all c in C. For convenience, we assume that $||b_s|| = 1$, and that $\{s_k\}$ is a sequence with $s_k > 1$ and $\{s_k\} \to 1$, as $k \to \infty$. Let $a_k = b_{s_k}$. Then there is a subsequence of the $\{a_k\}$ converging to some a, with $||a|| = 1$, and

$$\langle a, c\rangle \leq \langle a, z\rangle,$$

for all c in C. This completes the proof. ∎

If we knew that there was a vector x not in C, such that $z = P_C x$, then we could choose $a = x - z$, as in the proof of the Separation Theorem. The point of the Support Theorem is that we cannot assume, a priori, that there is such an x. Once we have the vector a, however, any point $x = z + \lambda a$, for $\lambda \geq 0$, has the property that $z = P_C x$.

4.7 Theorems of the Alternative

The following theorem is a good illustration of a type of theorem known as *Theorems of the Alternative*. These theorems assert that precisely one of two problems will have a solution. The proof illustrates how we should go about proving such theorems.

Theorem 4.5 (Gale I)[88] *Precisely one of the following is true:*

- **(1)** *there is x such that $Ax = b$;*
- **(2)** *there is y such that $A^T y = 0$ and $b^T y = 1$.*

Proof: First, we show that it is not possible for both to be true at the same time. Suppose that $Ax = b$ and $A^T y = 0$. Then $b^T y = x^T A^T y = 0$, so that we cannot have $b^T y = 1$. By Theorem 4.1, the fundamental decomposition theorem from linear algebra, we know that, for any b, there are unique x and w with $A^T w = 0$ such that $b = Ax + w$. Clearly, $b = Ax$ if and only if $w = 0$. Also, $b^T y = w^T y$. Therefore, if alternative **(1)** does not hold, we must have w non-zero, in which case $A^T y = 0$ and $b^T y = 1$, for $y = w/||w||^2$, so alternative **(2)** holds. ∎

In this section we consider several other theorems of this type.

Theorem 4.6 (Farkas' Lemma)[83] *Precisely one of the following is true:*

- **(1)** *there is $x \geq 0$ such that $Ax = b$;*
- **(2)** *there is y such that $A^T y \geq 0$ and $b^T y < 0$.*

Proof: We can restate the lemma as follows: there is a vector y with $A^T y \geq 0$ and $b^T y < 0$ if and only if b is not a member of the convex set $C = \{Ax | x \geq 0\}$. If b is not in C, which is closed and convex, then, by the Separation Theorem, there is a non-zero vector a and real α with

$$a^T b < \alpha \leq a^T Ax = (A^T a)^T x,$$

for all $x \geq 0$. Since $(A^T a)^T x$ is bounded below, as x runs over all non-negative vectors, it follows that $A^T a \geq 0$. Choosing $x = 0$, we have $\alpha \leq 0$. Then let $y = a$. Conversely, if $Ax = b$ does have a non-negative solution x, then $A^T y \geq 0$ implies that $0 \leq y^T Ax = y^T b \geq 0$. ∎

The next theorem can be obtained from Farkas' Lemma.

Theorem 4.7 (Gale II)[88] *Precisely one of the following is true:*

- **(1)** *there is x such that $Ax \leq b$;*
- **(2)** *there is $y \geq 0$ such that $A^T y = 0$ and $b^T y < 0$.*

Proof: First, if both are true, then $0 \leq y^T(b - Ax) = y^T b - 0 = y^T b$, which is a contradiction. Now assume that **(2)** does not hold. Therefore, for every $y \geq 0$ with $A^T y = 0$, we have $b^T y \geq 0$. Let $B = [\, A \quad b \,]$. Then the system $B^T y = [\, 0 \quad -1 \,]^T$ has no non-negative solution. Applying Farkas' Lemma, we find that there is a vector $w = [\, z \quad \gamma \,]^T$ with $Bw \geq 0$ and $[\, 0 \quad -1 \,]\, w < 0$. So, $Az + \gamma b \geq 0$ and $\gamma > 0$. Let $x = -\frac{1}{\gamma} z$ to get $Ax \leq b$, so that **(1)** holds. ∎

Theorem 4.8 (Gordan)[93] *Precisely one of the following is true:*

- **(1)** *there is x such that $Ax < 0$;*
- **(2)** *there is $y \geq 0$, $y \neq 0$, such that $A^T y = 0$.*

Proof: First, if both are true, then $0 < -y^T Ax = 0$, which cannot be true. Now assume that there is no non-zero $y \geq 0$ with $A^T y = 0$. Then, with $e = (1, 1, ..., 1)^T$, $C = [\, A \quad e \,]$, and $d = (0, 0, ..., 0, 1)^T$, there is no non-negative solution of $C^T y = d$. From Farkas' Lemma we then know that there is a vector $z = [\, u \quad \gamma \,]^T$, with $Cz = Au + \gamma e \geq 0$, and $d^T z < 0$. Then $Ax < 0$ for $x = -u$. ∎

Here are several more theorems of the alternative.

Theorem 4.9 (Stiemke I)[142] *Precisely one of the following is true:*

- **(1)** *there is x such that $Ax \leq 0$ and $Ax \neq 0$;*
- **(2)** *there is $y > 0$ such that $A^T y = 0$.*

Theorem 4.10 (Stiemke II)[142] *Let c be a fixed non-zero vector. Precisely one of the following is true:*

- **(1)** *there is x such that $Ax \leq 0$ and $c^T x \geq 0$ and not both $Ax = 0$ and $c^T x = 0$;*
- **(2)** *there is $y > 0$ such that $A^T y = c$.*

Theorem 4.11 (Gale III)[88] *Let c be a fixed non-zero vector. Precisely one of the following is true:*

- **(1)** *there is $x \geq 0$ such that $Ax \geq 0$ and $c^T x < 0$;*
- **(2)** *there is $y \geq 0$ such that $A^T y \leq c$.*

Proof: First, note that we cannot have both true at the same time, since we would then have

$$0 < x^T (c - A^T y) = c^T x - (Ax)^T y \leq c^T x,$$

which is a contradiction. Now suppose that **(2)** does not hold. Then there is no $w \geq 0$ such that

$$[\, A^T \quad I \,]\, w = c.$$

By Farkas' Lemma (Theorem 4.6), it follows that there is x with

$$\begin{bmatrix} A \\ I \end{bmatrix} x \geq 0,$$

and $c^T x < 0$. Therefore, $Ax \geq 0$, $Ix = x \geq 0$, and $c^T x < 0$; therefore, **(1)** holds. ∎

Theorem 4.12 (Von Neumann)[125] *Precisely one of the following is true:*

- **(1)** *there is* $x \geq 0$ *such that* $Ax > 0$*;*
- **(2)** *there is* $y \geq 0$*,* $y \neq 0$*, such that* $A^T y \leq 0$*.*

Proof: If both were true, then we would have

$$0 < (Ax)^T y = x^T (A^T y),$$

so that $A^T y \leq 0$ would be false. Now suppose that **(2)** does not hold. Then there is no $y \geq 0$, $y \neq 0$, with $A^T y \leq 0$. Consequently, there is no $y \geq 0$, $y \neq 0$, such that

$$\begin{bmatrix} A^T \\ -u^T \end{bmatrix} y = \begin{bmatrix} A^T y \\ -u^T y \end{bmatrix} \leq \begin{bmatrix} 0 \\ -1 \end{bmatrix},$$

where $u^T = (1, 1, ..., 1)$. By Theorem 4.11, there is

$$z = \begin{bmatrix} x \\ \alpha \end{bmatrix} \geq 0,$$

such that

$$[\, A \quad -u \,]\, z = [\, A \quad -u \,] \begin{bmatrix} x \\ \alpha \end{bmatrix} \geq 0,$$

and

$$[\, 0 \quad -1 \,]\, z = [\, 0 \quad -1 \,] \begin{bmatrix} x \\ \alpha \end{bmatrix} = -\alpha < 0.$$

Therefore, $\alpha > 0$ and $(Ax)_i - \alpha \geq 0$ for each i, and so $Ax > 0$ and **(1)** holds. ∎

Theorem 4.13 (Tucker)[145] *Precisely one of the following is true:*

- **(1)** *there is* $x \geq 0$ *such that* $Ax \geq 0$*,* $Ax \neq 0$*;*
- **(2)** *there is* $y > 0$ *such that* $A^T y \leq 0$*.*

Theorem 4.14 (Theorem 21.1, [133]) *Let* C *be a convex set, and let* $f_1, ..., f_m$ *be proper convex functions, with* $ri(C) \subseteq dom(f_i)$*, for each* i*. Precisely one of the following is true:*

- **(1)** *there is* $x \in C$ *such that* $f_i(x) < 0$*, for* $i = 1, ..., m$*;*
- **(2)** *there are* $\lambda_i \geq 0$*, not all equal to zero, such that*

$$\lambda_1 f_1(x) + ... + \lambda_m f_m(x) \geq 0,$$

for all x *in* C*.*

Theorem 4.14 is fundamental in proving Helly's Theorem:

Theorem 4.15 (Helly's Theorem) [133] *Let $\{C_i \,|i = 1, ..., I\}$ be a finite collection of (not necessarily closed) convex sets in R^N. If every subcollection of $N+1$ or fewer sets has non-empty intersection, then the entire collection has non-empty intersection.*

For instance, in the two-dimensional plane, if a finite collection of lines is such that every two intersect and every three have a common point of intersection, then they all have a common point of intersection. There is another version of Helly's Theorem that applies to convex inequalities.

Theorem 4.16 *Let there be given a system of the form*

$$f_1(x) < 0, ..., f_k(x) < 0, f_{k+1}(x) \leq 0, ..., f_m(x) \leq 0,$$

where the f_i are convex functions on R^J, and the inequalities may be all strict or all weak. If every subsystem of $J+1$ or fewer inequalities has a solution in a given convex set C, then the entire system has a solution in C.

4.8 Another Proof of Farkas' Lemma

In the previous section, we proved Farkas' Lemma, Theorem 4.6, using the Separation Theorem, the proof of which, in turn, depended here on the existence of the orthogonal projection onto any closed convex set. It is possible to prove Farkas' Lemma directly, along the lines of Gale [88].

Suppose that $Ax = b$ has no non-negative solution. If, indeed, it has no solution whatsoever, then $b = Ax + w$, where $w \neq 0$ and $A^T w = 0$. Then we take $y = -w/||w||^2$. So suppose that $Ax = b$ does have solutions, but not any non-negative ones. The approach is to use induction on the number of columns of the matrix involved in the lemma.

If A has only one column, denoted a^1, then $Ax = b$ can be written as

$$x_1 a^1 = b.$$

Assuming that there are no non-negative solutions, it must follow that $x_1 < 0$. We take $y = -b$. Then

$$b^T y = -b^T b = -||b||^2 < 0,$$

while

$$A^T y = (a^1)^T(-b) = \frac{-1}{x_1} b^T b > 0.$$

Now assume that the lemma holds whenever the involved matrix has no more than $m-1$ columns. We show the same is true for m columns.

If there is no non-negative solution of the system $Ax = b$, then clearly there are no non-negative real numbers $x_1, x_2, ..., x_{m-1}$ such that

$$x_1 a^1 + x_2 a^2 + ... + x_{m-1} a^{m-1} = b,$$

where a^j denotes the jth column of the matrix A. By the induction hypothesis, there must be a vector v with

$$(a^j)^T v \geq 0,$$

for $j = 1, ..., m-1$, and $b^T v < 0$. If it happens that $(a^m)^T v \geq 0$ also, then we are done. If, on the other hand, we have $(a^m)^T v < 0$, then let

$$c^j = (a^j)^T a^m - (a^m)^T a^j,\ j = 1, ..., m-1,$$

and

$$d = (b^T v) a^m - ((a^m)^T v) b.$$

Then there are no non-negative real numbers $z_1, ..., z_{m-1}$ such that

$$z_1 c^1 + z_2 c^2 + ... + z_{m-1} c^{m-1} = d, \tag{4.33}$$

since, otherwise, it would follow from simple calculations that

$$\frac{-1}{(a^m)^T v}\Big([\sum_{j=1}^{m-1} z_j((a^j)^T v)] - b^T v\Big) a^m - \sum_{j=1}^{m-1} z_j((a^m)^T v) a^j = b.$$

Close inspection of this shows all the coefficients to be non-negative, which implies that the system $Ax = b$ has a non-negative solution, contrary to our assumption. It follows, therefore, that there can be no non-negative solution to the system in Equation (4.33).

By the induction hypothesis, it follows that there is a vector u such that

$$(c^j)^T u \geq 0,\ j = 1, ..., m-1,$$

and

$$d^T u < 0.$$

Now let

$$y = ((a^m)^T u) v - ((a^m)^T v) u.$$

We can easily verify that

$$(a^j)^T y = (c^j)^T u \geq 0,\ j = 1, ..., m-1,$$

$$b^T y = d^T u < 0,$$

and

$$(a^m)^T y = 0,$$

so that

$$A^T y \geq 0,$$

and

$$b^T y < 0.$$

This completes the proof.

4.9 Exercises

4.1 *Prove Proposition 4.2.*

4.2 *Show that the subset of R^J consisting of all vectors x with $||x||_2 = 1$ is not convex.*

4.3 *Prove that every subspace of R^J is convex, and every linear manifold is convex.*

4.4 *Prove that every hyperplane $H(a, \gamma)$ is a linear manifold.*

4.5 *Prove Lemmas 4.2, 4.3 and 4.4.*

4.6 *Let C be a convex set and $f : C \subseteq R^J \to (-\infty, \infty]$. Prove that $f(x)$ is a convex function if and only if, for all x and y in C, and for all $0 < \alpha < 1$, we have*

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y).$$

4.7 *Let $f : R^J \to [-\infty, \infty]$. Prove that $f(x)$ is a convex function if and only if, for all $0 < \alpha < 1$, we have*

$$f(\alpha x + (1 - \alpha)y) < \alpha b + (1 - \alpha)c,$$

whenever $f(x) < b$ and $f(y) < c$.

4.8 *Show that the vector a is orthogonal to the hyperplane $H = H(a, \gamma)$; that is, if u and v are in H, then a is orthogonal to $u - v$.*

4.9 *Given a point s in a convex set C, where are the points x for which $s = P_C x$?*

4.10 *Let C be a closed non-empty convex set in R^J, x a vector not in C, and $d > 0$ the distance from x to C. Let*

$$\sigma_C(a) = \sup_{x \in C} \langle a, x \rangle,$$

the support function of C. Show that

$$d = \max_{||a|| \leq 1} \{\langle a, x \rangle - \sigma_C(a)\}.$$

Hints: Consider the unit vector $\frac{1}{d}(x - P_C x)$, and use Cauchy's Inequality and Proposition 4.4.

Remark: If, in the definition of the support function, we take the vectors a to be unit vectors, with $a = (\cos\theta, \sin\theta)$, for $0 \leq \theta < 2\pi$, then we can define the function

$$f(\theta) = \sup_{(x,y)\in C} x\cos\theta + y\sin\theta.$$

In [118] Tom Marzetta considers this function, as well as related functions of θ, such as the radius of curvature function, and establishes relationships between the behavior of these functions and the convex set itself.

4.11 (Rådström Cancellation [15])

- *(a) Show that, for any subset S of R^J, we have $2S \subseteq S + S$, and $2S = S + S$ if S is convex.*
- *(b) Find three finite subsets of R, say A, B, and C, with A not contained in B, but with the property that $A + C \subseteq B + C$.*
- *(c) Show that, if A and B are convex, B is closed, and C is bounded, then $A + C \subseteq B + C$ implies that $A \subseteq B$. Hint: Note that, under these assumptions, $2A + C = A + (A + C) \subseteq 2B + C$.*

Chapter 5

Linear Programming

The term *linear programming* (LP) refers to the problem of optimizing a linear function of several variables over linear equality or inequality constraints. In this chapter we present the problem and establish the basic facts. For a much more detailed discussion, consult [122]. We begin with a review of basic linear algebra.

5.1 Basic Linear Algebra

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

5.1.1 Bases and Dimension

The notions of a basis and of linear independence are fundamental in linear algebra. Let $\mathcal{V}$ be a vector space.

Definition 5.1 *A collection of vectors $\{u^1, ..., u^N\}$ in $\mathcal{V}$ is linearly independent if there is no choice of scalars $\alpha_1, ..., \alpha_N$, not all zero, such that*

$$0 = \alpha_1 u^1 + ... + \alpha_N u^N. \tag{5.1}$$

Definition 5.2 *The span of a collection of vectors $\{u^1, ..., u^N\}$ in $\mathcal{V}$ is the set of all vectors x that can be written as linear combinations of the u^n; that is, for which there are scalars $c_1, ..., c_N$, such that*

$$x = c_1 u^1 + ... + c_N u^N. \tag{5.2}$$

Definition 5.3 *A collection of vectors $\{w^1, ..., w^N\}$ in $\mathcal{V}$ is called a spanning set for a subspace S if the set S is their span.*

Definition 5.4 *A collection of vectors* $\{u^1, ..., u^N\}$ *in* $\mathcal{V}$ *is called a basis for a subspace* S *if the collection is linearly independent and* S *is their span.*

Definition 5.5 *A collection of vectors* $\{u^1, ..., u^N\}$ *in* $\mathcal{V}$ *is called orthonormal if* $\|u^n\|_2 = 1$*, for all* n*, and* $\langle u^m, u^n \rangle = 0$*, for* $m \neq n$*.*

Suppose that S is a subspace of $\mathcal{V}$, that $\{w^1, ..., w^N\}$ is a spanning set for S, and $\{u^1, ..., u^M\}$ is a linearly independent subset of S. Beginning with w_1, we augment the set $\{u^1, ..., u^M\}$ with w_j if w_j is not in the span of the u_m and the w_k previously included. At the end of this process, we have a linearly independent spanning set, and therefore, a basis, for S (Why?). Similarly, beginning with w_1, we remove w_j from the set $\{w^1, ..., w^N\}$ if w_j is a linear combination of the w_k, $k = 1, ..., j-1$. In this way we obtain a linearly independent set that spans S, hence another basis for S. The following lemma will allow us to prove that all bases for a subspace S have the same number of elements.

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

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

We note that the set B_1 is a spanning set for S and has N members. Having obtained the spanning set B_k, with N members and whose first k members are $v_k, ..., v_1$, we form the set $C_{k+1} = B_k \cup \{v_{k+1}\}$, listing the members so that the first $k+1$ of them are $\{v_{k+1}, v_k, ..., v_1\}$. To get the set B_{k+1} we remove the first member of C_{k+1} that is a linear combination of the members to its left; there must be one, since B_k is a spanning set, and so v_{k+1} is a linear combination of the members of B_k. Since the set V is linearly independent, the member removed is from the set W. Continuing in this fashion, we obtain a sequence of spanning sets $B_1, ..., B_N$, each with N members. The set B_N is $B_N = \{v_1, ..., v_N\}$ and v_{N+1} must then be a linear combination of the members of B_N, which contradicts the linear independence of V. ∎

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

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

Lemma 5.2 *For any matrix A, the number of linearly independent rows equals the number of linearly independent columns.*

Proof: See Exercise 5.3.

Definition 5.7 *The rank of A is the number of linearly independent rows or of linearly independent columns of A.*

5.1.2 Systems of Linear Equations

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

$$\begin{array}{rrrrrl} x_1 & +2x_2 & & +2x_4 & +x_5 & = 0 \\ -x_1 & -x_2 & +x_3 & +x_4 & & = 0\,. \\ x_1 & +2x_2 & -3x_3 & -x_4 & -2x_5 & = 0 \end{array} \tag{5.3}$$

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

$$A = \begin{bmatrix} 1 & 2 & 0 & 2 & 1 \\ -1 & -1 & 1 & 1 & 0 \\ 1 & 2 & -3 & -1 & -2 \end{bmatrix}, \tag{5.4}$$

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

$$\begin{array}{rrrrrl} x_1 & & & -2x_4 & +x_5 & = 0 \\ & x_2 & & +2x_4 & & = 0\,. \\ & & x_3 & +x_4 & +x_5 & = 0 \end{array} \tag{5.5}$$

From this simpler system we see that the variables x_4 and x_5 can be freely chosen, with the other three variables then determined by this system of equations. The variables x_4 and x_5 are then independent, the others dependent. The variables x_1, x_2 and x_3 are then called *basic variables*. To obtain a basis of solutions we can let $x_4 = 1$ and $x_5 = 0$, obtaining the solution $x = (2, -2, -1, 1, 0)^T$, and then choose $x_4 = 0$ and $x_5 = 1$ to get the solution $x = (-1, 0, -1, 0, 1)^T$. Every solution to $Ax = 0$ is then a linear combination of these two solutions. Notice that which variables are basic and which are non-basic is somewhat arbitrary, in that we could have chosen as the non-basic variables any two whose columns are independent.

Having decided that x_4 and x_5 are the non-basic variables, we can write the original matrix A as $A = [\,B \quad N\,]$, where B is the square invertible matrix

$$B = \begin{bmatrix} 1 & 2 & 0 \\ -1 & -1 & 1 \\ 1 & 2 & -3 \end{bmatrix}, \tag{5.6}$$

and N is the matrix

$$N = \begin{bmatrix} 2 & 1 \\ 1 & 0 \\ -1 & -2 \end{bmatrix}. \tag{5.7}$$

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

$$Ax = Bx_B + Nx_N = 0, \tag{5.8}$$

so that

$$x_B = -B^{-1}Nx_N. \tag{5.9}$$

5.1.3 Real and Complex Systems of Linear Equations

A system $Ax = b$ of linear equations is called a *complex system*, or a *real system* if the entries of A, x and b are complex, or real, respectively. For any matrix A, we denote by A^T and $A^\dagger$ the transpose and conjugate transpose of A, respectively.

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

$$\tilde{A} = \begin{bmatrix} A_1 & -A_2 \\ A_2 & A_1 \end{bmatrix}, \tag{5.10}$$

by $\tilde{x}$ the real vector

$$\tilde{x} = \begin{bmatrix} x^1 \\ x^2 \end{bmatrix}, \tag{5.11}$$

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

$$\tilde{b} = \begin{bmatrix} b^1 \\ b^2 \end{bmatrix}. \tag{5.12}$$

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

Definition 5.8 *A square matrix A is symmetric if $A^T = A$ and Hermitian if $A^\dagger = A$.*

Definition 5.9 *A non-zero vector x is said to be an eigenvector of the square matrix A if there is a scalar λ such that $Ax = \lambda x$. Then λ is said to be an eigenvalue of A.*

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

$$B = \begin{bmatrix} 1 & 2+i \\ 2-i & 1 \end{bmatrix} \tag{5.13}$$

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

$$\begin{bmatrix} u^1 \\ u^2 \end{bmatrix}, \tag{5.14}$$

and

$$\begin{bmatrix} -u^2 \\ u^1 \end{bmatrix}, \tag{5.15}$$

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

$$\begin{bmatrix} v^1 \\ v^2 \end{bmatrix}, \tag{5.16}$$

and

$$\begin{bmatrix} -v^2 \\ v^1 \end{bmatrix}, \tag{5.17}$$

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

5.2 Primal and Dual Problems

The fundamental problem in linear programming is to minimize the function

$$f(x) = c^T x, \tag{5.18}$$

over the *feasible set* F, that is, the convex set of all $x \geq 0$ with $Ax = b$. Shortly, we shall present an algebraic description of the extreme points of the feasible set F, in terms of *basic feasible solutions*, show that there are at most finitely many extreme points of F and that every member of F can be written as a convex combination of the extreme points, plus a direction

of unboundedness. These results will be used to prove the basic theorems about the primal and dual linear programming problems and to describe the simplex algorithm.

Associated with the basic problem in LP, called the *primary problem*, there is a second problem, the *dual problem*. Both of these problems can be written in two equivalent ways, the canonical form and the standard form.

5.2.1 An Example

Consider the problem of maximizing the function $f(x_1, x_2) = x_1 + 2x_2$, over all $x_1 \geq 0$ and $x_2 \geq 0$, for which the inequalities

$$x_1 + x_2 \leq 40,$$

and

$$2x_1 + x_2 \leq 60$$

are satisfied. The set of points satisfying all four inequalities is the quadrilateral with vertices $(0, 0)$, $(30, 0)$, $(20, 20)$, and $(0, 40)$; draw a picture. Since the level curves of the function f are straight lines, the maximum value must occur at one of these vertices; in fact, it occurs at $(0, 40)$ and the maximum value of f over the constraint set is 80. Rewriting the problem as minimizing the function $-x_1 - 2x_2$, subject to $x_1 \geq 0$, $x_2 \geq 0$,

$$-x_1 - x_2 \geq -40,$$

and

$$-2x_1 - x_2 \geq -60,$$

the problem is now in what is called *primal canonical form.*

5.2.2 Canonical and Standard Forms

Let b and c be fixed vectors and A a fixed matrix. The problem

$$\text{minimize } z = c^T x, \text{ subject to } Ax \geq b, x \geq 0 \text{ (PC)} \tag{5.19}$$

is the so-called *primary problem* of LP, in *canonical form.* The *dual problem* in canonical form is

$$\text{maximize } w = b^T y, \text{ subject to } A^T y \leq c, y \geq 0. \text{ (DC)} \tag{5.20}$$

The primary problem, in *standard form*, is

$$\text{minimize } z = c^T x, \text{ subject to } Ax = b, x \geq 0 \text{ (PS)} \tag{5.21}$$

with the dual problem in standard form given by

$$\text{maximize } w = b^T y, \text{ subject to } A^T y \leq c. \text{ (DS)} \tag{5.22}$$

Notice that the dual problem in standard form does not require that y be nonnegative. Note also that the standard problems make sense only if the system $Ax = b$ is under-determined and has infinitely many solutions. For that reason, we shall assume, for the standard problems, that the I by J matrix A has more columns than rows, so $J > I$, and has full row rank.

If we are given the primary problem in canonical form, we can convert it to standard form by augmenting the variables, that is, by defining

$$u_i = (Ax)_i - b_i, \tag{5.23}$$

for $i = 1, ..., I$, and rewriting $Ax \geq b$ as

$$\tilde{A}\tilde{x} = b, \tag{5.24}$$

for $\tilde{A} = [\, A \quad -I \,]$ and $\tilde{x} = [x^T u^T]^T$.

If we are given the primary problem in standard form, we can convert it to canonical form by writing the equations as inequalities, that is, by replacing $Ax = b$ with the two matrix inequalities $Ax \geq b$, and $(-A)x \geq -b$.

5.2.3 Weak Duality

Consider the problems (PS) and (DS). Say that x is *feasible* if $x \geq 0$ and $Ax = b$. Let F be the set of feasible x. Say that y is *feasible* if $A^T y \leq c$. The *Weak Duality Theorem* is the following:

Theorem 5.1 *Let x and y be feasible vectors. Then*

$$z = c^T x \geq b^T y = w. \tag{5.25}$$

Corollary 5.2 *If z is not bounded below, then there are no feasible y.*

Corollary 5.3 *If x and y are both feasible, and $z = w$, then both x and y are optimal for their respective problems.*

The proof of the theorem and its corollaries are left as exercises.

The nonnegative quantity $c^T x - b^T y$ is called the *duality gap*. The *complementary slackness condition* says that, for optimal x and y, we have

$$x_j(c_j - (A^T y)_j) = 0, \tag{5.26}$$

for each j, which says that the duality gap is zero. Primal-dual algorithms for solving linear programming problems are based on finding sequences $\{x^k\}$ and $\{y^k\}$ that drive the duality gap down to zero [122].

5.2.4 Strong Duality

The *Strong Duality Theorem* makes a stronger statement.

Theorem 5.2 *If one of the problems (PS) or (DS) has an optimal solution, then so does the other and $z = w$ for the optimal vectors.*

Before we consider the proof of the theorem, we need a few preliminary results.

Recall that, for (PS) we assume that the I by J matrix A has more columns than rows, that is, $J > I$, and the rank of A is I. If, for any nonnegative vector x, the columns j for which x_j is positive are linearly independent, then x_j is positive for at most I values of j.

Definition 5.10 *A point x in F is said to be a basic feasible solution if the columns of A corresponding to positive entries of x are linearly independent.*

Therefore, a basic feasible solution can have at most I positive entries.

Now let x be an arbitrary basic feasible solution. Denote by B an invertible matrix obtained from A by deleting $J-I$ columns associated with zero entries of x. Note that, if x has fewer than I positive entries, then some of the columns of A associated with zero values of x_j are retained. The entries of an arbitrary vector y corresponding to the columns not deleted are called the *basic variables*. Then, assuming that the columns of B are the first I columns of A, we write $y^T = (y_B^T, y_N^T)$, and

$$A = [\, B \quad N \,], \tag{5.27}$$

so that $Ay = By_B + Ny_N$, $Ax = Bx_B = b$, and $x_B = B^{-1}b$.

The following theorems are taken from the book by Nash and Sofer [122]. We begin with a characterization of the extreme points of F (recall Definition 4.20).

Theorem 5.3 *A point x is in Ext(F) if and only if x is a basic feasible solution.*

Proof: Suppose that x is a basic feasible solution, and we write $x^T = (x_B^T, 0^T)$, $A = [\, B \quad N \,]$. If x is not an extreme point of F, then there are $y \neq x$ and $z \neq x$ in F, and α in $(0, 1)$, with

$$x = (1 - \alpha)y + \alpha z. \tag{5.28}$$

Then $y^T = (y_B^T, y_N^T)$, $z^T = (z_B^T, z_N^T)$, and $y_N \geq 0$, $z_N \geq 0$. From

$$0 = x_N = (1 - \alpha)y_N + (\alpha)z_N \tag{5.29}$$

it follows that

$$y_N = z_N = 0, \tag{5.30}$$

and $b = By_B = Bz_B = Bx_B$. But, since B is invertible, we have $x_B = y_B = z_B$. This is a contradiction, so x must be in Ext(F).

Conversely, suppose that x is in Ext(F). Since it is in F, we know that $Ax = b$ and $x \geq 0$. By reordering the variables if necessary, we may assume that $x^T = (x_B^T, x_N^T)$, with $x_B > 0$ and $x_N = 0$; we do not know that x_B is a vector of length I, however, so when we write $A = [\, B \quad N \,]$, we do not know that B is square. If B is invertible, then x is a basic feasible solution. If not, we shall construct $y \neq x$ and $z \neq x$ in F, such that

$$x = \frac{1}{2}y + \frac{1}{2}z. \tag{5.31}$$

If $\{B_1, B_2, ..., B_K\}$ are the columns of B and are linearly dependent, then there are constants $p_1, p_2, ..., p_K$, not all zero, with

$$p_1 B_1 + ... + p_K B_K = 0. \tag{5.32}$$

With $p^T = (p_1, ..., p_K)$, we have

$$B(x_B + \alpha p) = B(x_B - \alpha p) = Bx_B = b, \tag{5.33}$$

for all $\alpha \in (0, 1)$. We then select α so small that both $x_B + \alpha p > 0$ and $x_B - \alpha p > 0$. Let

$$y^T = (x_B^T + \alpha p^T, x_N^T) \tag{5.34}$$

and

$$z^T = (x_B^T - \alpha p^T, x_N^T). \tag{5.35}$$

Therefore x is not an extreme point of F, which is a contradiction. This completes the proof. ∎

Lemma 5.3 *There are at most finitely many basic feasible solutions, so there are at most finitely many members of Ext(F).*

Theorem 5.4 *If F is not empty, then Ext(F) is not empty. In that case, let $\{v^1, ..., v^M\}$ be the members of Ext(F). Every x in F can be written as*

$$x = d + \alpha_1 v^1 + ... + \alpha_M v^M, \tag{5.36}$$

for some $\alpha_m \geq 0$, with $\sum_{m=1}^{M} \alpha_m = 1$, and some direction of unboundedness, d.

Proof: We consider only the case in which F is bounded, so there is no direction of unboundedness; the unbounded case is similar. Let x be a feasible point. If x is an extreme point, fine. If not, then x is not a basic

feasible solution. The columns of A that correspond to the positive entries of x are not linearly independent. Then we can find a vector p such that $Ap = 0$ and $p_j = 0$ if $x_j = 0$. If $|\epsilon|$ is small, $x + \epsilon p \geq 0$ and $(x + \epsilon p)_j = 0$ if $x_j = 0$, then $x + \epsilon p$ is in F. We can alter ϵ in such a way that eventually $y = x + \epsilon p$ has one more zero entry than x has, and so does $z = x - \epsilon p$. Both y and z are in F and x is the average of these points. If y and z are not basic, repeat the argument on y and z, each time reducing the number of positive entries. Eventually, we will arrive at the case where the number of non-zero entries is I, and so will have a basic feasible solution. ∎

Proof of the Strong Duality Theorem: Suppose now that x_* is a solution of the problem (PS) and $z_* = c^T x_*$. Without loss of generality, we may assume that x_* is a basic feasible solution, hence an extreme point of F. Then we can write

$$x_*^T = ((B^{-1}b)^T, 0^T), \tag{5.37}$$

$$c^T = (c_B^T, c_N^T), \tag{5.38}$$

and $A = [\, B \quad N \,]$. Every feasible solution has the form

$$x^T = ((B^{-1}b)^T, 0^T) + ((B^{-1}Nv)^T, v^T), \tag{5.39}$$

for some $v \geq 0$. From $c^T x \geq c^T x_*$ we find that

$$(c_N^T - c_B^T B^{-1} N)(v) \geq 0, \tag{5.40}$$

for all $v \geq 0$. It follows that

$$c_N^T - c_B^T B^{-1} N = 0. \tag{5.41}$$

Nw let $y_* = (B^{-1})^T c_B$, or $y_*^T = c_B^T B^{-1}$. We show that y_* is feasible for (DS); that is, we show that

$$A^T y_* \leq c^T. \tag{5.42}$$

Since

$$y_*^T A = (y_*^T B, y_*^T N) = (c_B^T, y_*^T N) = (c_B^T, c_B^T B^{-1} N) \tag{5.43}$$

and

$$c_N^T \geq c_B^T B^{-1} N, \tag{5.44}$$

we have

$$y_*^T A \leq c^T, \tag{5.45}$$

so y_* is feasible for (DS). Finally, we show that

$$c^T x_* = y_*^T b. \tag{5.46}$$

We have

$$y_*^T b = c_B^T B^{-1} b = c^T x_*. \tag{5.47}$$

This completes the proof. ∎

5.2.5 Gale's Strong Duality Theorem

In [88] Gale presents the following theorem:

Theorem 5.5 Gale's Strong Duality Theorem *If both problems (PC) and (DC) have feasible solutions, then both have optimal solutions and the optimal values are equal.*

Proof: We show that there are non-negative vectors x and y such that $Ax \geq b$, $A^T y \leq c$, and $b^T y - c^T x \geq 0$. It will then follow that $z = c^T x = b^T y = w$, so that x and y are both optimal. In matrix notation, we want to find $x \geq 0$ and $y \geq 0$ such that

$$\begin{bmatrix} A & 0 \\ 0 & -A^T \\ -c^T & b^T \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \geq \begin{bmatrix} b \\ -c \\ 0 \end{bmatrix}. \tag{5.48}$$

We assume that there are no $x \geq 0$ and $y \geq 0$ for which the inequalities in (5.48) hold. Then, according to Theorem 4.11, there are non-negative vectors s and t, and non-negative scalar ρ such that

$$\begin{bmatrix} -A^T & 0 & c \\ 0 & A & -b \end{bmatrix} \begin{bmatrix} s \\ t \\ \rho \end{bmatrix} \geq 0, \tag{5.49}$$

and

$$\begin{bmatrix} -b^T & c^T & 0 \end{bmatrix} \begin{bmatrix} s \\ t \\ \rho \end{bmatrix} < 0. \tag{5.50}$$

Note that ρ cannot be zero, for then we would have $A^T s \leq 0$ and $At \geq 0$. Taking feasible vectors x and y, we would find that $s^T Ax \leq 0$, which implies that $b^T s \leq 0$, and $t^T A^T y \geq 0$, which implies that $c^T t \geq 0$. Therefore, we could not also have $c^T t - b^T s < 0$.

Writing out the inequalities, we have

$$\rho c^T t \geq s^T A t \geq s^T(\rho b) = \rho s^T b.$$

Using $\rho > 0$, we find that

$$c^T t \geq b^T s,$$

which is a contradiction. Therefore, there do exist $x \geq 0$ and $y \geq 0$ such that $Ax \geq b$, $A^T y \leq c$, and $b^T y - c^T x \geq 0$. ∎

5.3 Some Examples

We give two well known examples of LP problems.

5.3.1 The Diet Problem

There are nutrients indexed by $i = 1, ..., I$ and our diet must contain at least b_i units of the ith nutrient. There are J foods, indexed by $j = 1, ..., J$, and one unit of the jth food cost c_j dollars and contains A_{ij} units of the ith nutrient. The problem is to minimize the cost, while obtaining at least the minimum amount of each nutrient.

Let $x_j \geq 0$ be the amount of the jth food that we consume. Then we need $Ax \geq b$, where A is the matrix with entries A_{ij}, b is the vector with entries b_i and x is the vector with entries $x_j \geq 0$. With c the vector with entries c_j, the total cost of our food is $z = c^T x$. The problem is then to minimize $z = c^T x$, subject to $Ax \geq b$ and $x \geq 0$. This is the primary LP problem, in canonical form.

5.3.2 The Transport Problem

We must ship products from sources to destinations. There are I sources, indexed by $i = 1, ..., I$, and J destinations, indexed by $j = 1, ..., J$. There are a_i units of product at the ith source, and we must have at least b_j units reaching the jth destination. The customer will pay C_{ij} dollars to get one unit from i to j. Let x_{ij} be the number of units of product to go from the ith source to the jth destination. The producer wishes to maximize income, that is,

$$\text{maximize } \sum_{i,j} C_{ij} x_{ij},$$

subject to

$$x_{ij} \geq 0,$$

$$\sum_{i=1}^{I} x_{ij} \geq b_j,$$

and

$$\sum_{j=1}^{J} x_{ij} \leq a_i.$$

Obviously, we must assume that

$$\sum_{i=1}^{I} a_i \geq \sum_{j=1}^{J} b_j.$$

This problem is not yet in the form of the LP problems considered so far. It also introduces a new feature, namely, it may be necessary to have x_{ij} a non-negative integer, if the products exist only in whole units. This leads to *integer programming*.

5.4 The Simplex Method

In this section we sketch the main ideas of the simplex method. For further details see [122].

Begin with a basic feasible solution of (PS) $\hat{x}$. Assume, as previously, that

$$A = [\, B \quad N \,], \tag{5.51}$$

where B is an I by I invertible matrix obtained by deleting from A some (but perhaps not all) columns associated with zero entries of $\hat{x}$. As before, we assume the variables have been ordered so that the zero entries of $\hat{x}$ have the highest index values. The entries of an arbitrary x corresponding to the first I columns are the basic variables. We write $x^T = (x_B^T, x_N^T)$, and so that $\hat{x}_N = 0$, $A\hat{x} = B\hat{x}_B = b$, and $\hat{x}_B = B^{-1}b$. The current value of z is

$$\hat{z} = c_B^T \hat{x}_B = c_B^T B^{-1} b.$$

We are interested in what happens to z as x_N takes on positive entries.

For any feasible x we have $Ax = b = Bx_B + Nx_n$, so that

$$x_B = B^{-1}b - B^{-1}Nx_N,$$

and

$$z = c^T x = c_B^T x_B + c_N^T x_N = c_B^T (B^{-1}b - B^{-1}Nx_N) + c_N^T x_N.$$

Therefore,

$$z = c_B^T B^{-1} b + (c_N^T - c_B^T B^{-1} N) x_N = \hat{z} + r^T x_N,$$

where

$$r^T = (c_N^T - c_B^T B^{-1} N).$$

The vector r is called the *reduced cost vector*. We define the vector $y^T = c_B^T B^{-1}$ of *simplex multipliers*, and write

$$z - \hat{z} = r^T x_N = (c_N^T - y^T N) x_N.$$

We are interested in how z changes as we move away from $\hat{x}$ and permit x_N to have positive entries.

If x_N is non-zero, then z changes by $r^T x_N$. Therefore, if $r \geq 0$, the current $\hat{z}$ cannot be made smaller by letting x_N have some positive entries; the current $\hat{x}$ is then optimal. Initially, at least, r will have some negative entries, and we use these as a guide in deciding how to select x_N.

Keep in mind that the vectors x_N and r have length $J - I$ and the jth column of N is the $(I + j)$th column of A.

Select an index j such that

$$r_j < 0, \tag{5.52}$$

and r_j is the most negative of the negative entries of r. Then x_{I+j} is called the *entering variable*. Compute $d^j = B^{-1} a^j$, where a^j is the $(I + j)$th column of A, which is the jth column of N. If we allow $(x_N)_j = x_{I+j}$ to be positive, then

$$x_B = B^{-1} b - B^{-1} a^j = B^{-1} b - x_{I+j} d^j.$$

We need to make sure that x_B remains non-negative, so we need

$$(B^{-1} b)_i - x_{I+j} d_i^j \geq 0,$$

for all indices $i = 1, ..., I$. If the ith entry d_i^j is negative, then $(x_B)_i$ increases as x_{I+j} becomes positive; if $d_i^j = 0$, then $(x_B)_i$ remains unchanged. The problem arises when d_i^j is positive.

Find an index s in $\{1, ..., I\}$ for which

$$\frac{(B^{-1} b)_s}{d_s^j} = \min\{\frac{(B^{-1} b)_i}{d_i^j} : d_i^j > 0\}. \tag{5.53}$$

Then x_s is the *leaving variable*, replacing x_{I+j}; that is, the new set of indices corresponding to new basic variables will now include $I + j$, and no longer include s. The new entries of $\hat{x}$ are $\hat{x}_s = 0$ and

$$\hat{x}_{I+j} = \frac{(B^{-1} b)_s}{d_s^j}.$$

We then rearrange the columns of A to redefine B and N, and rearrange the positions of the entries of x, to get the new basic variables vector x_B, the new x_N and the new c. Then we repeat the process.

It is helpful to note that when the columns of A are rearranged and a new B is defined, the new B differs from the old B in only one column. Therefore

$$B_{\text{new}} = B_{\text{old}} - uv^T, \tag{5.54}$$

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

The Sherman-Morrison-Woodbury Identity:

$$(B - uv^T)^{-1} = B^{-1} + \alpha(B^{-1}u)(v^T B^{-1}), \tag{5.55}$$

where

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

We shall illustrate this in the example below.

5.5 An Example of the Simplex Method

Consider once again the problem of maximizing the function $f(x_1, x_2) = x_1 + 2x_2$, over all $x_1 \geq 0$ and $x_2 \geq 0$, for which the inequalities

$$x_1 + x_2 \leq 40,$$

and

$$2x_1 + x_2 \leq 60$$

are satisfied. In (PS) form, the problem is to minimize the function $-x_1 - 2x_2$, subject to $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$, $x_4 \geq 0$,

$$-x_1 - x_2 - x_3 = -40,$$

and

$$-2x_1 - x_2 - x_4 = -60.$$

The matrix A is then

$$A = \begin{bmatrix} -1 & -1 & -1 & 0 \\ -2 & -1 & 0 & -1 \end{bmatrix}, \tag{5.56}$$

the matrix B is

$$B = \begin{bmatrix} -1 & -1 \\ -2 & -1 \end{bmatrix}, \tag{5.57}$$

with inverse

$$B^{-1} = \begin{bmatrix} 1 & -1 \\ -2 & 1 \end{bmatrix}, \tag{5.58}$$

and the matrix N is

$$N = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{5.59}$$

The vector b is $b = (-40, -60)^T$. A general vector x is $x = (x_1, x_2, x_3, x_4)^T$, with $x_B = (x_1, x_2)^T$ and $x_N = (x_3, x_4)^T$, and $c = (-1, -2, 0, 0)^T$, with $c_B = (-1, -2)^T$ and $c_N = (0, 0)^T$. The *feasible set* of points satisfying all four inequalities is the quadrilateral in R^2 with vertices $(0, 0)$, $(30, 0)$, $(20, 20)$, and $(0, 40)$. In R^4, these vertices correspond to the vectors $(0, 0, 40, 60)^T$, $(30, 0, 10, 0)^T$, $(20, 20, 0, 0)^T$, and $(0, 40, 0, 20)^T$. Since we have chosen to start with x_1 and x_2 as our basic variables, we let our starting vector be $\hat{x} = (20, 20, 0, 0)^T$, so that $\hat{x}_B = B^{-1}b = (20, 20)^T$, and $\hat{x}_N = (0, 0)^T$. Then we find that $y^T = c_B^T B^{-1} = (3, -1)^T$, and $y^T N = (-3, 1)^T$. The reduced cost vector is then

$$r^T = c_N^T - y^T N = (0, 0)^T - (-3, 1)^T = (3, -1)^T.$$

Since r^T has a negative entry in its second position, $j = 2$, we learn that the entering variable is going to be $x_{2+j} = x_4$. The fourth column of A is $(0, -1)^T$, so the vector d^2 is

$$d^2 = B^{-1}(0, -1)^T = (1, -1)^T.$$

Therefore, we must select a new positive value for x_4 that satisfies

$$(20, 20) \geq x_4(1, -1).$$

The single positive entry of d^2 is the first one, from which we conclude that the leaving variable will be x_1. We therefore select as the new values of the variables $\hat{x}_1 = 0$, $\hat{x}_2 = 40$, $\hat{x}_3 = 0$, and $\hat{x}_4 = 20$. We then reorder the variables as $x = (x_4, x_2, x_3, x_1)^T$ and rearrange the columns of A accordingly. Having done this, we see that we now have

$$B = B_{\text{new}} = \begin{bmatrix} 0 & -1 \\ -1 & -1 \end{bmatrix}, \tag{5.60}$$

with inverse

$$B^{-1} = \begin{bmatrix} 1 & -1 \\ -1 & 0 \end{bmatrix}, \tag{5.61}$$

and the matrix N is

$$N = \begin{bmatrix} -1 & -1 \\ 0 & -2 \end{bmatrix}. \tag{5.62}$$

Since

$$B_{\text{new}} = B_{\text{old}} - \begin{bmatrix} -1 \\ -1 \end{bmatrix} [\, 1 \quad 0 \,],$$

we can apply the Sherman-Morrison-Woodbury Identity to get B_{new}^{-1}.

The reduced cost vector is now $r^T = (2, 1)^T$. Since it has no negative entries, we have reached the optimal point; the solution is $\hat{x}_1 = 0$, $\hat{x}_2 = 40$, with slack variables $\hat{x}_3 = 0$ and $\hat{x}_4 = 20$.

5.6 Another Example of the Simplex Method

The following example is taken from Fang and Puthenpura [82]. Minimize the function

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = -x_1 - x_2 - x_3,$$

subject to

$$2x_1 + x_4 = 1;$$
$$2x_2 + x_5 = 1;$$
$$2x_3 + x_6 = 1;$$

and $x_i \geq 0$, for $i = 1, ..., 6$. The variables x_4, x_5, and x_6 appear to be slack variables, introduced to obtain equality constraints.

Initially, we define the matrix A to be

$$A = \begin{bmatrix} 2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 & 1 \end{bmatrix}, \tag{5.63}$$

$b = (1, 1, 1)^T$, $c = (-1, -1, -1, 0, 0, 0)^T$ and $x = (x_1, x_2, x_3, x_4, x_5, x_6)^T$.

Suppose we begin with x_4, x_5, and x_6 as the basic variables. We then rearrange the entries of the vector of unknowns so that

$$x = (x_4, x_5, x_6, x_1, x_2, x_3)^T.$$

Now we have to rearrange the columns of A as well; the new A is

$$A = \begin{bmatrix} 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 \end{bmatrix}. \tag{5.64}$$

The vector c must also be redefined; the new one is $c = (0, 0, 0, -1, -1, -1)^T$, so that $c_N = (-1, -1, -1)^T$ and $c_B = (0, 0, 0)^T$.

For this first step of the simplex method we have

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

and

$$N = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

Note that one advantage in choosing the slack variables as the basic variables is that it is easy then to find the corresponding basic feasible solution, which is now

$$\hat{x} = \begin{bmatrix} \hat{x}_4 \\ \hat{x}_5 \\ \hat{x}_6 \\ \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \end{bmatrix} = \begin{bmatrix} \hat{x}_B \\ \hat{x}_N \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The reduced cost vector r is then

$$r = (-1, -1, -1)^T;$$

since it has negative entries, the current basic feasible solution is not optimal.

Suppose that we select a non-basic variable with negative reduced cost, say x_1, which, we must remember, is the fourth entry of the redefined x, so $j = 1$ and $I + j = 4$. Then x_1 is the entering basic variable, and the vector d^1 is then

$$d^1 = B^{-1}a^j = (1, 0, 0)^T.$$

The only positive entry of d^1 is the first one, which means, according to Equation (5.53), that the exiting variable should be x_4. Now the new set of basic variables is $\{x_5, x_6, x_1\}$ and the new set of non-basic variables is $\{x_2, x_3, x_4\}$. The new matrices B and N are

$$B = \begin{bmatrix} 0 & 0 & 2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

and

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

Continuing through two more steps, we find that the optimal solution is $-3/2$, and it occurs at the vector

$$x = (x_1, x_2, x_3, x_4, x_5, x_6)^T = (1/2, 1/2, 1/2, 0, 0, 0)^T.$$

5.7 Some Possible Difficulties

In the first example of the simplex method, we knew all four of the vertices of the feasible region, so we could choose any one of them to get our initial basic feasible solution. We chose to begin with x_1 and x_2 as our basic variables, which meant that the slack variables were zero and our first basic feasible solution was $\hat{x} = (20, 20, 0, 0)^T$. In the second example, we chose the slack variables to be the initial basic variables, which made it easy to find the initial basic feasible solution. Generally, however, finding an initial basic feasible solution may not be easy.

You might think that we can always simply take the slack variables as our initial basic variables, so that the initial B is just the identity matrix, and the initial basic feasible solution is merely the concatenation of the column vectors b and 0, as in the second example. The following example shows why this may not always work.

5.7.1 A Third Example:

Consider the problem of minimizing the function $z = 2x_1 + 3x_2$, subject to

$$3x_1 + 2x_2 = 14,$$

$$2x_1 - 4x_2 - x_3 = 2,$$

$$4x_1 + 3x_2 + x_4 = 19,$$

and $x_i \geq 0$, for $i = 1, ..., 4$. The matrix A is now

$$A = \begin{bmatrix} 3 & 2 & 0 & 0 \\ 2 & -4 & -1 & 0 \\ 4 & 3 & 0 & 1 \end{bmatrix}. \tag{5.65}$$

There are only two slack variables, so we cannot construct our set of basic variables using only slack variables, since the matrix B must be square. We cannot begin with $\hat{x}_1 = \hat{x}_2 = 0$, since this would force $\hat{x}_3 = -2$, which is not permitted. We can choose $\hat{x}_2 = 0$ and solve for the other three, to get $\hat{x}_1 = \frac{14}{3}$, $\hat{x}_3 = \frac{22}{3}$, and $\hat{x}_4 = \frac{1}{3}$. This is relatively easy only because the problem is artificially small. The point here is that, for realistically large LP problems, finding a place to begin the simplex algorithm may not be a simple matter. For more on this matter, see [122].

In both of our first two examples, finding the inverse of the matrix B is easy, since B is only 2 by 2, or 3 by 3. In larger problems, finding B^{-1}, or better, solving $y^T B = c_B^T$ for y^T, is not trivial and can be an expensive part of each iteration. The Sherman-Morrison-Woodbury identity is helpful here.

5.8 Topics for Projects

The simplex method provides several interesting topics for projects.

- **1.** Investigate the issue of finding a suitable starting basic feasible solution. Reference [122] can be helpful in this regard.
- **2.** How can we reduce the cost associated with solving $y^T B = c_B^T$ for y^T at each step of the simplex method?
- **3.** Suppose that, instead of needing the variables to be nonnegative, we need each x_i to lie in the interval $[\alpha_i, \beta_i]$. How can we modify the simplex method to incorporate these constraints?
- **4.** Investigate the role of linear programming and the simplex method in graph theory and networks, with particular attention to the transport problem.
- **5.** There is a sizable literature on the computational complexity of the simplex method. Investigate this issue and summarize your findings.

5.9 Exercises

5.1 *Let $W = \{w^1, ..., w^N\}$ be a spanning set for a subspace S in R^I, and $V = \{v^1, ..., v^M\}$ a linearly independent subset of S. Then, according to Lemma 5.1, $M \leq N$. Let A be the I by M matrix whose columns are the vectors v_m and B the I by N matrix whose columns are the w_n. Since W is a spanning set for S, there is an N by M matrix C such that $A = BC$. Prove Lemma 5.1 by considering the space of solutions of the system $Ax = 0$.*

5.2 *Prove Theorem 5.1 and its corollaries.*

5.3 *Prove Lemma 5.2. Hints: Suppose that A is an I by J matrix, and that the column space of A, that is, the subspace $CS(A)$ of R^I spanned by the columns of A, has dimension K, for some $K \leq J$. Show that there is an I by K matrix U and a K by J matrix M such that $A = UM$. Use $A^T = M^T U^T$ to show that the column space of A^T, the subspace $CS(A^T)$ of R^J spanned by the columns of A^T, has a spanning set with K members. Use the fact that the columns of A^T are the transposes of the rows of A, so that $CS(A^T) = RS(A)^T$, to conclude that the dimensions of $RS(A)$, the row space of A, and $CS(A)$ are the same; this number is the rank of A.*

5.4 *Complete the calculation of the optimal solution for the problem in the second example of the simplex method.*

5.5 *Consider the following problem, taken from [82]. Minimize the function*

$$f(x_1, x_2, x_3, x_4) = -3x_1 - 2x_2,$$

subject to

$$x_1 + x_2 + x_3 = 40,$$

$$2x_1 + x_2 + x_4 = 60,$$

and

$$x_j \geq 0,$$

for $j = 1, ..., 4$. *Use the simplex method to find the optimum solution. Take as a starting vector* $(x^0)^T = (0, 0, 40, 60)^T$.

5.6 *Redo the first example of the simplex method, starting with the vertex* $x_1 = 0$ *and* $x_2 = 0$.

5.7 *Consider the LP problem of maximizing the function* $f(x_1, x_2) = x_1 + 2x_2$, *subject to*

$$-2x_1 + x_2 \leq 2,$$

$$-x_1 + 2x_2 \leq 7,$$

$$x_1 \leq 3,$$

and $x_1 \geq 0$, $x_2 \geq 0$. *Start at* $x_1 = 0$, $x_2 = 0$. *You will find that you have a choice for the entering variable; try it both ways.*

5.8 *Apply the simplex method to the problem of minimizing* $z = -x_1 - 2x_2$, *subject to*

$$-x_1 + x_2 \leq 2,$$

$$-2x_1 + x_2 \leq 1,$$

and $x_1 \geq 0$, $x_2 \geq 0$.

Chapter 6

Matrix Games and Optimization

The theory of two-person games is largely the work of John von Neumann, and was developed somewhat later by von Neumann and Morgenstern [125] as a tool for economic analysis. Two-person zero-sum games provide a nice example of optimization and an opportunity to apply some of the linear algebra and linear programming tools previously discussed. In this chapter we introduce the idea of two-person matrix games and use results from linear programming to prove the Fundamental Theorem of Game Theory.

A two-person game is called a *constant-sum game* if the total payout is the same, each time the game is played. In such cases, we can subtract half the total payout from the payout to each player and record only the difference. Then the total payout appears to be zero, and such games are called *zero-sum games*. We can then suppose that whatever one player wins is paid by the other player. Except for the final section, we shall consider only two-person, zero-sum games.

6.1 Deterministic Solutions

In this two-person game, the first player, call him P1, selects a row of the I by J real matrix A, say i, and the second player selects a column of A, say j. The second player, call her P2, pays the first player A_{ij}. If some $A_{ij} < 0$, then this means that the first player pays the second. Since whatever the first player wins, the second loses, and vice versa, we need only one matrix to summarize the situation.

6.1.1 Optimal Pure Strategies

In our first example, the matrix is

$$A = \begin{bmatrix} 7 & 8 & 4 \\ 4 & 7 & 2 \end{bmatrix}. \tag{6.1}$$

The first player notes that by selecting row $i = 1$, he will get at least 4, regardless of which column the second player plays. The second player notes that, by playing column $j = 3$, she will pay the first player no more than 4, regardless of which row the first player plays. If the first player then begins to play $i = 1$ repeatedly, and the second player notices this consistency, she will still have no motivation to play any column except $j = 3$, because the other pay-outs are both worse than 4. Similarly, so long as the second player is playing $j = 3$ repeatedly, the first player has no motivation to play anything other than $i = 1$, since he will be paid less if he switches. Therefore, both players adopt a *pure strategy* of $i = 1$ and $j = 3$. This game is said to be *deterministic* and the entry $A_{1,3} = 4$ is a *saddle-point* because it is the maximum of its column and the minimum of its row. We then have

$$\max_i \min_j A_{ij} = 4 = \min_j \max_i A_{ij}.$$

Not all such two-person games have saddle-points, however.

6.1.2 Optimal Randomized Strategies

Consider now the two-person game with pay-off matrix

$$A = \begin{bmatrix} 4 & 1 \\ 2 & 3 \end{bmatrix}. \tag{6.2}$$

The first player notes that by selecting row $i = 2$, he will get at least 2, regardless of which column the second player plays. The second player notes that, by playing column $j = 2$, she will pay the first player no more than 3, regardless of which row the first player plays. If both begin by playing in this conservative manner, the first player will play $i = 2$ and the second player will play $j = 2$.

If the first player plays $i = 2$ repeatedly, and the second player notices this consistency, she will be tempted to switch to playing column $j = 1$, thereby losing only 2, instead of 3. If she makes the switch and the first player notices, he will be motivated to switch his play to row $i = 1$, to get a pay-off of 4, instead of 2. The second player will then soon switch to playing $j = 2$ again, hoping that the first player sticks with $i = 1$. But the first player is not stupid, and quickly returns to playing $i = 2$. There is no saddle-point in this game.

For such games, it makes sense for both players to select their play at random, with the first player playing $i = 1$ with probability p and $i = 2$ with probability $1 - p$, and the second player playing column $j = 1$ with probability q and $j = 2$ with probability $1-q$. These are called *randomized strategies*.

When the first player plays $i = 1$, he expects to get $4q+(1-q) = 3q+1$, and when he plays $i = 2$ he expects to get $2q + 3(1 - q) = 3 - q$. Since he plays $i = 1$ with probability p, he expects to get

$$p(3q + 1) + (1 - p)(3 - q) = 4pq - 2p - q + 3 = (4p - 1)q + 3 - 2p.$$

He notices that if he selects $p = \frac{1}{4}$, then he expects to get $\frac{5}{2}$, regardless of what the second player does. If he plays something other than $p = \frac{1}{4}$, his expected winnings will depend on what the second player does. If he selects a value of p less than $\frac{1}{4}$, and $q = 1$ is selected, then he wins $2p + 2$, but this is less than $\frac{5}{2}$. If he selects $p > \frac{1}{4}$ and $q = 0$ is selected, then he wins $3 - 2p$, which again is less than $\frac{5}{2}$. The maximum of these minimum pay-offs occurs when $p = \frac{1}{4}$ and the *max-min* win is $\frac{5}{2}$.

Similarly, the second player, noticing that

$$p(3q + 1) + (1 - p)(3 - q) = (4q - 2)p + 3 - q,$$

sees that she will pay out $\frac{5}{2}$ if she takes $q = \frac{1}{2}$. If she selects a value of q less than $\frac{1}{2}$, and $p = 0$ is selected, then she pays out $3 - q$, which is more than $\frac{5}{2}$. If, on the other hand, she selects a value of q that is greater than $\frac{1}{2}$, and $p = 1$ is selected, then she will pay out $3q+1$, which again is greater than $\frac{5}{2}$. The only way she can be certain to pay out no more than $\frac{5}{2}$ is to select $q = \frac{1}{2}$. The minimum of these maximum pay-outs occurs when she chooses $q = \frac{1}{2}$, and the *min-max* pay-out is $\frac{5}{2}$.

This leads us to the question of whether or not there will always be probability vectors for the players that will lead to the equality of the max-min win and the min-max pay-out.

We make a notational change at this point. From now on the letters p and q will denote probability column vectors, and not individual probabilities, as in this section.

6.1.3 The Min-Max Theorem

Let A be an I by J pay-off matrix. Let

$$P = \{p = (p_1, ..., p_I) \,|\, p_i \geq 0, \sum_{i=1}^{I} p_i = 1\},$$

$$Q = \{q = (q_1, ..., q_J) \,|\, q_j \geq 0, \sum_{j=1}^{J} q_j = 1\},$$

and

$$R = A(Q) = \{Aq \,|\, q \in Q\}.$$

The first player selects a vector p in P and the second selects a vector q in Q. The expected pay-off to the first player is

$$E = \langle p, Aq \rangle \ = p^T Aq.$$

Let

$$m_0 = \max_{r \in R} \min_{p \in P} \langle p, r \rangle,$$

and

$$m^0 = \min_{p \in P} \max_{r \in R} \langle p, r \rangle.$$

Clearly, we have

$$\min_{p \in P} \langle p, r \rangle \leq \langle p, r \rangle \leq \max_{r \in R} \langle p, r \rangle,$$

for all $p \in P$ and $r \in R$. It follows that $m_0 \leq m^0$. The Min-Max Theorem, also known as the Fundamental Theorem of Game Theory, asserts that $m_0 = m^0$.

Theorem 6.1 The Fundamental Theorem of Game Theory *Let A be an arbitrary real I by J matrix. Then there are vectors $\hat{p}$ in P and $\hat{q}$ in Q such that*

$$p^T A\hat{q} \leq \hat{p}^T A\hat{q} \leq \hat{p}^T Aq, \tag{6.3}$$

for all p in P and q in Q.

The quantity $\omega = \hat{p}^T A\hat{q}$ is called the *value of the game*. Notice that if P1 knows that P2 plays according to the mixed-strategy vector $\hat{q}$, P1 could examine the entries $(A\hat{q})_i$, which are his expected pay-offs should he play strategy i, and select the one for which this expected pay-off is largest. It follows from the inequalities in (6.3) that

$$(A\hat{q})_i \leq \omega$$

for all i, and

$$(A\hat{q})_i = \omega$$

for all i for which $\hat{p}_i > 0$. However, if P2 notices what P1 is doing, she can abandon $\hat{q}$ to her advantage.

There are a number of different proofs of the Fundamental Theorem. In an appendix, we present a proof using Fenchel Duality. For the remainder of this chapter we consider various proofs, focusing mainly on linear algebra methods, linear programming, and theorems of the alternative.

6.2 Symmetric Games

A game is said to be *symmetric* if the available strategies are the same for both players, and if the players switch strategies, the outcomes switch also. In other words, the pay-off matrix A is skew-symmetric, that is, A is square and $A_{ji} = -A_{ij}$. For symmetric games, we can use Theorem 4.12 to prove the existence of a randomized solution.

First, we show that there is a probability vector $\hat{p} \geq 0$ such that $\hat{p}^T A \geq 0$. Then we show that

$$p^T A\hat{p} \leq 0 = \hat{p}^T A\hat{p} \leq \hat{p}^T Aq,$$

for all probability vectors p and q. It will then follow that $\hat{p}$ and $\hat{q} = \hat{p}$ are the optimal mixed strategies.

If there is no non-zero $x \geq 0$ such that $x^T A \geq 0$, then there is no non-zero $x \geq 0$ such that $A^T x \geq 0$. Then, by Theorem 4.12, we know that there is $y \geq 0$ with $Ay < 0$; obviously y is not the zero vector, in this case. Since $A^T = -A$, it follows that $y^T A > 0$. Consequently, there is a non-zero $x \geq 0$, such that $x^T A \geq 0$; it is $x = y$. This is a contradiction. So $\hat{p}$ exists.

Since the game is symmetric, we have

$$\hat{p}^T A\hat{p} = (\hat{p}^T A\hat{p})^T = \hat{p}^T A^T \hat{p} = -\hat{p}^T A\hat{p},$$

so that $\hat{p}^T A\hat{p} = 0$.

For any probability vectors p and q we have

$$p^T A\hat{p} = \hat{p}^T A^T p = -\hat{p}^T Ap \leq 0,$$

and

$$0 \leq \hat{p}^T Aq.$$

We conclude that the mixed strategies $\hat{p}$ and $\hat{q} = \hat{p}$ are optimal.

6.2.1 An Example of a Symmetric Game

We present now a simple example of a symmetric game and compute the optimal randomized strategies.

Consider the pay-off matrix

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{6.4}$$

This matrix is skew-symmetric, so the game is symmetric. Let $\hat{p}^T = [1, 0]$; then $\hat{p}^T A = [0, 1] \geq 0$. We show that $\hat{p}$ and $\hat{q} = \hat{p}$ are the optimal randomized strategies. For any probability vectors $p^T = [p_1, p_2]$ and $q^T = [q_1, q_2]$, we have

$$p^T A\hat{p} = -p_2 \leq 0,$$

$$\hat{p}^T A\hat{p} = 0,$$

and

$$\hat{p}^T Aq = q_2 \geq 0.$$

It follows that the pair of strategies $\hat{p} = \hat{q} = [1, 0]^T$ are optimal randomized strategies.

6.2.2 Comments on the Proof of the Min-Max Theorem

In [88], Gale proves the existence of optimal randomized solutions for an arbitrary matrix game by showing that there is associated with such a game a symmetric matrix game and that an optimal randomized solution exists for one if and only if such exists for the other.

6.3 Positive Games

As Gale notes in [88], it is striking that two fundamental mathematical tools in linear economic theory, linear programming and game theory, developed simultaneously, and independently, in the years following the Second World War. More remarkable still was the realization that these two areas are closely related. Gale's proof of the Min-Max Theorem, which relates the game to a linear programming problem and employs his Strong Duality Theorem, provides a good illustration of this close connection.

If the I by J pay-off matrix A has only positive entries, we can use Gale's Strong Duality Theorem 5.5 for linear programming to prove the Min-Max Theorem.

Let b and c be the vectors whose entries are all one. Consider the LP problem of minimizing $z = c^T x$, over all $x \geq 0$ with $A^T x \geq b$; this is the (PC) problem. The (DC) problem is then to maximize $w = b^T y$, over all $y \geq 0$ with $Ay \leq c$. Since A has only positive entries, both (PC) and (DC) are feasible, so, by Gale's Strong Duality Theorem 5.5, we know that there are feasible non-negative vectors $\hat{x}$ and $\hat{y}$ and non-negative μ such that

$$\hat{z} = c^T \hat{x} = \mu = b^T \hat{y} = \hat{w}.$$

Since $\hat{x}$ cannot be zero, μ must be positive.

6.3.1 Exercises

6.1 *Show that the vectors $\hat{p} = \frac{1}{\mu}\hat{x}$ and $\hat{q} = \frac{1}{\mu}\hat{y}$ are probability vectors and are optimal randomized strategies for the matrix game.*

6.2 *Given an arbitrary I by J matrix A, there is $\alpha > 0$ so that the matrix B with entries $B_{ij} = A_{ij} + \alpha$ has only positive entries. Show that any optimal randomized probability vectors for the game with pay-off matrix B are also optimal for the game with pay-off matrix A.*

It follows from these exercises that there exist optimal randomized solutions for any matrix game.

6.3.2 Comments

This proof of the Min-Max Theorem shows that we can associate with a given matrix game a linear programming problem. It follows that we can use the simplex method to find optimal randomized solutions for matrix games. It also suggests that a given linear programming problem can be associated with a matrix game; see Gale [88] for more discussion of this point.

6.4 Learning the Game

In our earlier discussion we saw that the matrix game involving the pay-off matrix

$$A = \begin{bmatrix} 4 & 1 \\ 2 & 3 \end{bmatrix} \tag{6.5}$$

is not deterministic. The best thing the players can do is to select their play at random, with the first player playing $i = 1$ with probability p and $i = 2$ with probability $1 - p$, and the second player playing column $j = 1$ with probability q and $j = 2$ with probability $1 - q$. If the first player, call him P1, selects $p = \frac{1}{4}$, then he expects to get $\frac{5}{2}$, regardless of what the second player, call her P2, does; otherwise his fortunes depend on what P2 does. His optimal mixed-strategy (column) vector is $[1/4, 3/4]^T$. Similarly, the second player notices that the only way she can be certain to pay out no more than $\frac{5}{2}$ is to select $q = \frac{1}{2}$. The minimum of these maximum pay-outs occurs when she chooses $q = \frac{1}{2}$, and the *min-max* pay-out is $\frac{5}{2}$.

Because the pay-off matrix is two-by-two, we are able to determine easily the optimal mixed-strategy vectors for each player. When the pay-off matrix is larger, finding the optimal mixed-strategy vectors is not a simple matter. As we have seen, one approach is to obtain these vectors by solving a related linear-programming problem. In this section we consider other approaches to finding the optimal mixed-strategy vectors.

6.4.1 An Iterative Approach

In [88] Gale presents an iterative approach to learning how best to play a matrix game. The assumptions are that the game is to be played repeatedly

and that the two players adjust their play as they go along, based on the earlier plays of their opponent.

Suppose, for the moment, that P1 knows that P2 is playing the randomized strategy q, where, as earlier, we denote by p and q probability column vectors. The entry $(Aq)_i$ of the column vector Aq is the expected pay-off to P1 if he plays strategy i. It makes sense for P1 then to find the index i for which this expected pay-off is largest and to play that strategy every time. Of course, if P2 notices what P1 is doing, she will abandon q to her advantage.

After the game has been played n times, the players can examine the previous plays and make estimates of what the opponent is doing. Suppose that P1 has played strategy i n_i times, where $n_i \geq 0$ and $n_1 + n_2 + ... + n_I = n$. Denote by p^n the probability column vector whose ith entry is n_i/n. Similarly, calculate q^n. These two probability vectors summarize the tendencies of the two players over the first n plays. It seems reasonable that an attempt to learn the game would involve these probability vectors.

For example, P1 could see which entry of q^n is the largest, assume that P2 is most likely to play that strategy the next time, and play his best strategy against that play of P2. However, if there are several strategies for P2 to choose, it is still unlikely that P2 will choose this strategy the next time. Perhaps P1 could do better by considering his long-run fortunes and examining the vector Aq^n of expected pay-offs. In the exercise below, you are asked to investigate this matter.

6.4.2 Exercise

6.3 *Suppose that both players are attempting to learn how best to play the game by examining the vectors p^n and q^n after n plays. Devise an algorithm for the players to follow that will lead to optimal mixed strategies for both. Simulate repeated play of a particular matrix game to see how your algorithm performs. If the algorithm does its job, but does it slowly, that is, it takes many plays of the game for it to begin to work, investigate how it might be speeded up.*

6.5 Non-Constant-Sum Games

In this final section we consider non-constant-sum games. These are more complicated and the mathematical results more difficult to obtain than in the constant-sum games. Such non-constant-sum games can be used to model situations in which the players may both gain by cooperation, or, when speaking of economic actors, by collusion [74]. We begin with the most famous example of a non-constant-sum game, the Prisoners' Dilemma.

6.5.1 The Prisoners' Dilemma

Imagine that you and your partner are arrested for robbing a bank and both of you are guilty. The two of you are held in separate rooms and given the following options by the district attorney: (1) if you confess, but your partner does not, you go free, while he gets three years in jail; (2) if he confesses, but you do not, he goes free and you get the three years; (3) if both of you confess, you each get two years; (4) if neither of you confesses, each of you gets one year in jail. Let us call you player number one, and your partner player number two. Let strategy one be to remain silent, and strategy two be to confess.

Your pay-off matrix is

$$A = \begin{bmatrix} -1 & -3 \\ 0 & -2 \end{bmatrix}, \tag{6.6}$$

so that, for example, if you remain silent, while your partner confesses, your pay-off is $A_{1,2} = -3$, where the negative sign is used because jail time is undesirable. From your perspective, the game has a deterministic solution; you should confess, assuring yourself of no more than two years in jail. Your partner views the situation the same way and also should confess. However, when the game is viewed, not from one individual's perspective, but from the perspective of the pair of you, we see that by sticking together you each get one year in jail, instead of each of you getting two years; if you cooperate, you both do better.

6.5.2 Two Pay-Off Matrices Needed

In the case of non-constant-sum games, one pay-off matrix is not enough to capture the full picture. Consider the following example of a non-constant-sum game. Let the matrix

$$A = \begin{bmatrix} 5 & 4 \\ 3 & 6 \end{bmatrix} \tag{6.7}$$

be the pay-off matrix for Player One (P_1), and

$$B = \begin{bmatrix} 5 & 6 \\ 7 & 2 \end{bmatrix} \tag{6.8}$$

be the pay-off matrix for Player Two (P_2); that is, $A_{1,2} = 4$ and $B_{2,1} = 7$ means that if P_1 plays the first strategy and P_2 plays the second strategy, then P_1 gains four and P_2 gains seven. Notice that the total pay-off for each play of the game is not constant, so we require two matrices, not one.

Player One, considering only the pay-off matrix A, discovers that the best strategy is a randomized strategy, with the first strategy played three

quarters of the time. Then P_1 has expected gain of $\frac{9}{2}$. Similarly, Player Two, applying the same analysis to his pay-off matrix, B, discovers that he should also play a randomized strategy, playing the first strategy five sixths of the time; he then has an expected gain of $\frac{16}{3}$. However, if P_1 switches and plays the first strategy all the time, while P_2 continues with his randomized strategy, P_1 expects to gain $\frac{29}{6} > \frac{27}{6}$, while the expected gain of P_2 is unchanged. This is very different from what happens in the case of a constant-sum game; there, the sum of the expected gains is constant, and equals zero for a zero-sum game, so P_1 would not be able to increase his expected gain, if P_2 plays his optimal randomized strategy.

6.5.3 An Example: Illegal Drugs in Sports

In a recent article in Scientific American [137], Michael Shermer uses the model of a non-constant-sum game to analyze the problem of doping, or illegal drug use, in sports, and to suggest a solution. He is a former competitive cyclist and his specific example comes from the Tour de France. He is the first player, and his opponent the second player. The choices are to cheat by taking illegal drugs or to stay within the rules. The assumption he makes is that a cyclist who sticks to the rules will become less competitive and will be dropped from his team.

Currently, the likelihood of getting caught is low, and the penalty for cheating is not too high, so, as he shows, the rational choice is for everyone to cheat, as well as for every cheater to lie. He proposes changing the pay-off matrices by increasing the likelihood of being caught, as well as the penalty for cheating, so as to make sticking to the rules the rational choice.

Chapter 7

Convex Functions

In this chapter we investigate further the properties of convex functions, in preparation for our discussion of iterative optimization algorithms.

7.1 Functions of a Single Real Variable

We begin by recalling some of the basic results concerning functions of a single real variable.

7.1.1 Fundamental Theorems

- The Intermediate Value Theorem:

  Theorem 7.1 *Let $f(x)$ be continuous on the interval $[a, b]$. If d is between $f(a)$ and $f(b)$, then there is c between a and b with $f(c) = d$.*

- The Mean Value Theorem (MVT):

  Theorem 7.2 *Let $f(x)$ be continuous on the closed interval $[a, b]$ and differentiable on (a, b). Then, there is c in (a, b) with*

  $$f(b) - f(a) = f'(c)(b - a).$$

- The Extended Mean Value Theorem (EMVT):

  Theorem 7.3 *Let $f(x)$ be twice differentiable on the interval (u, v) and let a and b be in (u, v). Then there is c between a and b with*

  $$f(b) = f(a) + f'(a)(b - a) + \frac{1}{2} f''(c)(b - a)^2.$$

- A MVT for Integrals:

  Theorem 7.4 *Let $g(x)$ be continuous and $h(x)$ integrable with constant sign on the interval $[a, b]$. Then there is c in (a, b) such that*

$$\int_a^b g(x)h(x)dx = g(c)\int_a^b h(x)dx.$$

If $f(x)$ is a function with $f''(x) > 0$ for all x and $f'(a) = 0$, then, from the EMVT, we know that $f(b) > f(a)$, unless $b = a$, so that $x = a$ is a global minimizer of the function $f(x)$. As we shall see, such functions are strictly convex.

7.1.2 Some Proofs

We begin with a proof of the Mean Value Theorem for Integrals. Since $g(x)$ is continuous on the interval $[a, b]$, it takes on its minimum value, say m, and its maximum value, say M, and, by the Intermediate Value Theorem, $g(x)$ also takes on any value in the interval $[m, M]$. Assume, without loss of generality, that $h(x) \geq 0$, for all x in the interval $[a, b]$, so that $\int_a^b h(x)dx \geq 0$. Then we have

$$m\int_a^b h(x)dx \leq \int_a^b g(x)h(x)dx \leq M\int_a^b h(x)dx,$$

which says that the ratio

$$\frac{\int_a^b g(x)h(x)dx}{\int_a^b h(x)dx}$$

lies in the interval $[m, M]$. Consequently, there is a value c in (a, b) for which $g(c)$ has the value of this ratio. This completes the proof.

Now we present two proofs of the EMVT. We begin by using integration by parts, with $u(x) = f'(x)$ and $v(x) = x - b$, to get

$$f(b) - f(a) = \int_a^b f'(x)dx = f'(x)(x-b)|_a^b - \int_a^b f''(x)(x-b)dx,$$

or

$$f(b) - f(a) = -f'(a)(a-b) - \int_a^b f''(x)(x-b)dx.$$

Then, using the MVT for integrals, with $g(x) = f''(x)$ assumed to be continuous, and $h(x) = x - b$, we have

$$f(b) = f(a) + f'(a)(b-a) - f''(c)\int_a^b (x-b)dx,$$

from which the assertion of the theorem follows immediately.

A second proof of the EMVT is as follows. Let a and b be fixed and set

$$F(x) = f(x) + f'(x)(b-x) + A(b-x)^2,$$

for some constant A to be determined. Then $F(b) = f(b)$. Select A so that $F(a) = f(b)$. Then $F(b) = F(a)$, so there is c in (a,b) with $F'(c) = 0$, by the MVT, or, more simply, from Rolle's Theorem. Therefore,

$$0 = F'(c) = f'(c) + f''(c)(b-c) + f'(c)(-1) - 2A(b-c) = (f''(c) - 2A)(b-c).$$

So $A = \frac{1}{2}f''(c)$ and

$$F(x) = f(x) + f'(x)(b-x) + \frac{1}{2}f''(c)(b-x)^2,$$

from which we get

$$F(a) = f(b) = f(a) + f'(a)(b-a) + \frac{1}{2}f''(c)(b-a)^2.$$

This completes the second proof.

7.1.3 Lipschitz Continuity

Let $f : R \to R$ be a differentiable function. From the Mean-Value Theorem we know that

$$f(b) = f(a) + f'(c)(b-a), \tag{7.1}$$

for some c between a and b. If there is a constant L with $|f'(x)| \leq L$ for all x, that is, the derivative is bounded, then we have

$$|f(b) - f(a)| \leq L|b-a|, \tag{7.2}$$

for all a and b; functions that satisfy Equation (7.2) are said to be *L-Lipschitz*.

7.1.4 The Convex Case

We focus now on the special case of convex functions. Earlier, we said that a function $g : S \to [-\infty, \infty]$ is convex if its epi-graph is a convex set, in which case the effective domain of the function g must be a convex set. For a real-valued function g defined on the whole real line we have several conditions on g that are equivalent to being a convex function.

Proposition 7.1 *The following are equivalent:*
1) the epi-graph of $g(x)$ is convex;
2) for all points $a < x < b$

$$g(x) \leq \frac{g(b) - g(a)}{b - a}(x - a) + g(a); \tag{7.3}$$

3) for all points $a < x < b$

$$g(x) \leq \frac{g(b) - g(a)}{b - a}(x - b) + g(b); \tag{7.4}$$

4) for all points a and b in R and for all α in the interval $(0, 1)$

$$g((1 - \alpha)a + \alpha b) \leq (1 - \alpha)g(a) + \alpha g(b). \tag{7.5}$$

The proof of Proposition 7.1 is left as an exercise.

As a result of Proposition 7.1, we can use the following definition of a convex real-valued function.

Definition 7.1 *A function $g : R \to R$ is called convex if, for each pair of distinct real numbers a and b, the line segment connecting the two points $A = (a, g(a))$ and $B = (b, g(b))$ is on or above the graph of $g(x)$; that is, for every α in $(0, 1)$,*

$$g((1 - \alpha)a + \alpha b) \leq (1 - \alpha)g(a) + \alpha g(b).$$

If the inequality is always strict, then $g(x)$ is strictly convex.

The function $g(x) = x^2$ is a simple example of a convex function. If $g(x)$ is convex on an open set in R, then $g(x)$ is continuous there, as well ([129], p. 47).It follows from Proposition 7.1 that, if $g(x)$ is convex, then, for every triple of points $a < x < b$, we have

$$\frac{g(x) - g(a)}{x - a} \leq \frac{g(b) - g(a)}{b - a} \leq \frac{g(b) - g(x)}{b - x}. \tag{7.6}$$

Therefore, for fixed a, the ratio

$$\frac{g(x) - g(a)}{x - a}$$

is an increasing function of x, and, for fixed b, the ratio

$$\frac{g(b) - g(x)}{b - x}$$

is an increasing function of x.

If $g(x)$ is a differentiable function, then convexity can be expressed in terms of properties of the derivative, $g'(x)$; for every triple of points $a < x < b$, we have

$$g'(a) \leq \frac{g(b) - g(a)}{b - a} \leq g'(b). \tag{7.7}$$

If $g(x)$ is differentiable and convex, then $g'(x)$ is an increasing function. In fact, the converse is also true, as we shall see shortly.

Recall that the line tangent to the graph of $g(x)$ at the point $x = a$ has the equation

$$y = g'(a)(x - a) + g(a). \tag{7.8}$$

Theorem 7.5 *For the differentiable function $g(x)$, the following are equivalent:*
1) $g(x)$ is convex;
2) for all a and x we have

$$g(x) \geq g(a) + g'(a)(x - a); \tag{7.9}$$

3) the derivative, $g'(x)$, is an increasing function, or, equivalently,

$$(g'(x) - g'(a))(x - a) \geq 0, \tag{7.10}$$

for all a and x.

Proof: Assume that $g(x)$ is convex. If $x > a$, then

$$g'(a) \leq \frac{g(x) - g(a)}{x - a}, \tag{7.11}$$

while, if $x < a$, then

$$\frac{g(a) - g(x)}{a - x} \leq g'(a). \tag{7.12}$$

In either case, the inequality in (7.9) holds. Now, assume that the inequality in (7.9) holds. Then

$$g(x) \geq g'(a)(x - a) + g(a), \tag{7.13}$$

and

$$g(a) \geq g'(x)(a - x) + g(x). \tag{7.14}$$

Adding the two inequalities, we obtain

$$g(a) + g(x) \geq (g'(x) - g'(a))(a - x) + g(a) + g(x), \tag{7.15}$$

from which we conclude that

$$(g'(x) - g'(a))(x - a) \geq 0. \tag{7.16}$$

So $g'(x)$ is increasing. Finally, we assume the derivative is increasing and show that $g(x)$ is convex. If $g(x)$ is not convex, then there are points $a < b$ such that, for all x in (a, b),

$$\frac{g(x) - g(a)}{x - a} > \frac{g(b) - g(a)}{b - a}. \tag{7.17}$$

By the Mean Value Theorem there is c in (a, b) with

$$g'(c) = \frac{g(b) - g(a)}{b - a}. \tag{7.18}$$

Select x in the interval (a, c). Then there is d in (a, x) with

$$g'(d) = \frac{g(x) - g(a)}{x - a}. \tag{7.19}$$

Then $g'(d) > g'(c)$, which contradicts the assumption that $g'(x)$ is increasing. This concludes the proof. ∎

If $g(x)$ is twice differentiable, we can say more. If we multiply both sides of the inequality in (7.16) by $(x - a)^{-2}$, we find that

$$\frac{g'(x) - g'(a)}{x - a} \geq 0, \tag{7.20}$$

for all x and a. This inequality suggests the following theorem.

Theorem 7.6 *If $g(x)$ is twice differentiable, then $g(x)$ is convex if and only if $g''(x) \geq 0$, for all x.*

Proof: According to the Mean Value Theorem, as applied to the function $g'(x)$, for any points $a < b$ there is c in (a, b) with $g'(b) - g'(a) = g''(c)(b - a)$. If $g''(x) \geq 0$, the right side of this equation is nonnegative, so the left side is also. Now assume that $g(x)$ is convex, which implies that $g'(x)$ is an increasing function. Since $g'(x + h) - g'(x) \geq 0$ for all $h > 0$, it follows that $g''(x) \geq 0$. ∎

The following result, as well as its extension to higher dimensions, will be helpful in our study of iterative optimization.

Theorem 7.7 *Let $h(x)$ be convex and differentiable and its derivative, $h'(x)$, non-expansive, that is,*

$$|h'(b) - h'(a)| \leq |b - a|, \tag{7.21}$$

for all a and b. Then $h'(x)$ is firmly non-expansive, which means that

$$(h'(b) - h'(a))(b - a) \geq (h'(b) - h'(a))^2. \tag{7.22}$$

Proof: Assume that $h'(b) - h'(a) \neq 0$, since the alternative case is trivial. If $h'(x)$ is non-expansive, then the inequality in (7.20) tells us that

$$0 \leq \frac{h'(b) - h'(a)}{b - a} \leq 1,$$

so that

$$\frac{b - a}{h'(b) - h'(a)} \geq 1.$$

Now multiply both sides by $(h'(b) - h'(a))^2$. ∎

In the next section we extend these results to functions of several variables.

7.2 Functions of Several Real Variables

In this section we consider the differentiability of a function of several variables. For more details, see the chapter on differentiability in the appendix.

Let $F : D \subseteq R^J \rightarrow R^N$ be a R^N-valued function of J real variables, defined on domain D with nonempty interior $\text{int}(D)$.

Definition 7.2 *The function $F(x)$ is said to be (Frechet) differentiable at point x^0 in int(D) if there is an N by J matrix $F'(x^0)$ such that*

$$\lim_{h \rightarrow 0} \frac{1}{||h||_2}[F(x^0 + h) - F(x^0) - F'(x^0)h] = 0. \tag{7.23}$$

It can be shown that, if F is differentiable at $x = x^0$, then F is continuous there as well [87].

If $f : R^J \rightarrow R$ is differentiable, then $f'(x^0) = \nabla f(x^0)$, the gradient of f at x^0. The function $f(x)$ is differentiable if each of its first partial derivatives is continuous. If the derivative $f' : R^J \rightarrow R^J$ is, itself, differentiable, then $f'' : R^J \rightarrow R^J$, and $f''(x) = H(x) = \nabla^2 f(x)$, the Hessian matrix whose entries are the second partial derivatives of f. The function $f(x)$ will be twice differentiable if each of the second partial derivatives is continuous. In that case, the mixed second partial derivatives are independent of the order of the variables, the Hessian matrix is symmetric, and the chain rule applies.

Let $f : R^J \to R$ be a differentiable function. From the Mean-Value Theorem ([87], p. 41) we know that, for any two points a and b, there is α in $(0, 1)$ such that

$$f(b) = f(a) + \langle \nabla f((1-\alpha)a + \alpha b), b - a \rangle. \tag{7.24}$$

If there is a constant L with $||\nabla f(x)||_2 \leq L$ for all x, that is, the gradient is bounded in norm, then we have

$$|f(b) - f(a)| \leq L||b - a||_2, \tag{7.25}$$

for all a and b; functions that satisfy Equation (7.25) are said to be *L-Lipschitz*.

We can study multivariate functions $f : R^J \to R$ by using them to construct functions of a single real variable, given by

$$\phi(t) = f(x^0 + t(x - x^0)),$$

where x and x^0 are fixed (column) vectors in R^J. If $f(x)$ is differentiable, then

$$\phi'(t) = \langle \nabla f(x^0 + t(x - x^0)), x - x^0 \rangle.$$

If $f(x)$ is twice continuously differentiable, then

$$\phi''(t) = (x - x^0)^T \nabla^2 f(x^0 + t(x - x^0))(x - x^0).$$

In addition to real-valued functions $f : R^J \to R$, we shall also be interested in functions $F : R^J \to R^J$, such as $F(x) = \nabla f(x)$, whose range is R^J, not R. We say that $F : R^J \to R^J$ is L-Lipschitz if there is $L > 0$ such that

$$||F(b) - F(a)||_2 \leq L||b - a||_2, \tag{7.26}$$

for all a and b.

Suppose $g : R^J \to R$ is differentiable and attains its minimum value. We want to minimize the function $g(x)$. Solving $\nabla g(x) = 0$ to find the optimal $x = x^*$ may not be easy, so we may turn to an iterative algorithm for finding roots of $\nabla g(x)$, or one that minimizes $g(x)$ directly. In the latter case, we may again consider a steepest descent algorithm of the form

$$x^{k+1} = x^k - \gamma \nabla g(x^k), \tag{7.27}$$

for some $\gamma > 0$. We denote by T the operator

$$Tx = x - \gamma \nabla g(x). \tag{7.28}$$

Then, using $\nabla g(x^*) = 0$, we find that

$$||x^* - x^{k+1}||_2 = ||Tx^* - Tx^k||_2. \tag{7.29}$$

We would like to know if there are choices for γ that imply convergence of the iterative sequence. As in the case of functions of a single variable, for functions $g(x)$ that are *convex*, the answer is yes.

7.2.1 The Convex Case

We begin with some definitions.

Definition 7.3 *The function $g(x) : R^J \to R$ is said to be convex if, for each pair of distinct vectors a and b and for every α in the interval $(0, 1)$ we have*

$$g((1-\alpha)a + \alpha b) \leq (1-\alpha)g(a) + \alpha g(b). \tag{7.30}$$

If the inequality is always strict, then $g(x)$ is called strictly convex.

The function $g(x)$ is convex if and only if, for every x and z in R^J and real t, the function $f(t) = g(x + tz)$ is a convex function of t. Therefore, the theorems for the multi-variable case can also be obtained from previous results for the single-variable case.

Definition 7.4 *A convex function $g : R^J \to [-\infty, +\infty]$ is proper if there is no x with $g(x) = -\infty$ and some x with $g(x) < +\infty$.*

Definition 7.5 *The effective domain of g is dom(g)=$D = \{x|g(x) < +\infty\}$.*

Definition 7.6 *A proper convex function g is closed if it is lower semi-continuous, that is, if $g(x) = \liminf g(y)$, as $y \to x$.*

A function g is closed if and only if its epi-graph is a closed set. If g is convex and finite on an open subset of dom(g), then g is continuous there, as well ([133]).

7.2.2 Subdifferentials and Subgradients

Suppose that $g : R^J \to (-\infty, +\infty]$ is convex and $g(x)$ is finite for x in the non-empty closed convex set C. Applying the Support Theorem to the epigraph of g, we obtain the following theorem.

Theorem 7.8 *If x^0 is an interior point of the set C, then there is a non-zero vector d with*

$$g(x) \geq g(x^0) + \langle d, x - x^0 \rangle,$$

for all x.

Proof: The point $(x^0, g(x^0))$ is a boundary point of the epigraph of g. According to the Support Theorem, there is a non-zero vector $a = (b, c)$ in R^{J+1}, with b in R^J and c real, such that

$$\langle b, x \rangle + cr = \langle a, (x, r) \rangle \leq \langle a, (x^0, g(x^0)) \rangle = \langle b, x^0 \rangle + cg(x^0),$$

for all (x, r) in the epigraph of g, that is, all (x, r) with $g(x) \leq r$. The real number c cannot be positive, since $\langle b, x\rangle + cr$ is bounded above, while r can be increased arbitrarily. Also c cannot be zero: if $c = 0$, then b cannot be zero and we would have $\langle b, x\rangle \leq \langle b, x^0\rangle$ for all x in C. But, since x^0 is in the interior of C, there is $t > 0$ such that $x = x^0 + tb$ is in C. So $c < 0$. We then select $d = -\frac{1}{c}b$. ∎

Note that it can happen that $b = 0$; therefore $d = 0$ is possible; see Exercise 7.2.

Definition 7.7 *A vector d is said to be a subgradient of the function $g(x)$ at $x = x^0$ if, for all x, we have*

$$g(x) \geq g(x^0) + \langle d, x - x^0\rangle.$$

The collection of all subgradients of g at $x = x^0$ is called the subdifferential of g at $x = x^0$, denoted $\partial g(x^0)$. The domain of ∂g is the set dom $\partial g = \{x|\partial g(x) \neq \emptyset\}$.

Theorem 7.8 says that the subdifferential of a convex function at an interior point of its domain is non-empty. If the subdifferential consists of a single vector, then g is differentiable at $x = x^0$ and that single vector is its gradient at $x = x^0$.

Note that, by the chain rule, $f'(t) = \nabla g(x + tz) \cdot z$, for the function $f(t) = g(x + tz)$.

Theorem 7.9 *Let $g : R^J \to R$ be differentiable. The following are equivalent:*
1) $g(x)$ is convex;
2) for all a and b we have

$$g(b) \geq g(a) + \langle \nabla g(a), b - a\rangle\,; \tag{7.31}$$

3) for all a and b we have

$$\langle \nabla g(b) - \nabla g(a), b - a\rangle \geq 0. \tag{7.32}$$

As in the case of functions of a single variable, we can say more when the function $g(x)$ is twice differentiable. To guarantee that the second derivative matrix is symmetric, we assume that the second partial derivatives are continuous. Note that, by the chain rule again, $f''(t) = z^T \nabla^2 g(x + tz)z$.

Theorem 7.10 *Let each of the second partial derivatives of $g(x)$ be continuous, so that $g(x)$ is twice continuously differentiable. Then $g(x)$ is convex if and only if the second derivative matrix $\nabla^2 g(x)$ is non-negative definite, for each x.*

Suppose that $g(x) : R^J \to R$ is convex and the function $F(x) = \nabla g(x)$ is L-Lipschitz. We have the following analog of Theorem 7.7.

Theorem 7.11 *Let $h(x)$ be convex and differentiable and its derivative, $\nabla h(x)$, non-expansive, that is,*

$$||\nabla h(b) - \nabla h(a)||_2 \leq ||b - a||_2, \tag{7.33}$$

for all a and b. Then $\nabla h(x)$ is firmly non-expansive, which means that

$$\langle \nabla h(b) - \nabla h(a), b - a\rangle \geq ||\nabla h(b) - \nabla h(a)||_2^2. \tag{7.34}$$

Unlike the proof of Theorem 7.7, the proof of this theorem is not trivial. In [92] Golshtein and Tretyakov prove the following theorem, from which Theorem 7.11 follows immediately.

Theorem 7.12 *Let $g : R^J \to R$ be convex and differentiable. The following are equivalent:*

$$||\nabla g(x) - \nabla g(y)||_2 \leq ||x - y||_2; \tag{7.35}$$

$$g(x) \geq g(y) + \langle \nabla g(y), x - y\rangle + \frac{1}{2}||\nabla g(x) - \nabla g(y)||_2^2; \tag{7.36}$$

and

$$\langle \nabla g(x) - \nabla g(y), x - y\rangle \geq ||\nabla g(x) - \nabla g(y)||_2^2. \tag{7.37}$$

Proof: The only difficult step in the proof is showing that Inequality (7.35) implies Inequality (7.36). To prove this part, let $x(t) = (1 - t)y + tx$, for $0 \leq t \leq 1$. Then

$$g'(x(t)) = \langle \nabla g(x(t)), x - y\rangle, \tag{7.38}$$

so that

$$\int_0^1 \langle \nabla g(x(t)) - \nabla g(y), x - y\rangle dt = g(x) - g(y) - \langle \nabla g(y), x - y\rangle. \tag{7.39}$$

Therefore,

$$g(x) - g(y) - \langle \nabla g(y), x - y\rangle \leq$$

$$\int_0^1 ||\nabla g(x(t)) - \nabla g(y)||_2 ||x(t) - y||_2 dt \tag{7.40}$$

$$\leq \int_0^1 ||x(t) - y||_2^2 dt = \int_0^1 ||t(x-y)||_2^2 dt = \frac{1}{2}||x-y||_2^2, \tag{7.41}$$

according to Inequality (7.35). Therefore,

$$g(x) \leq g(y) + \langle \nabla g(y), x - y \rangle + \frac{1}{2}||x-y||_2^2. \tag{7.42}$$

Now let $x = y - \nabla g(y)$, so that

$$g(y - \nabla g(y)) \leq g(y) + \langle \nabla g(y), \nabla g(y) \rangle + \frac{1}{2}||\nabla g(y)||_2^2. \tag{7.43}$$

Consequently,

$$g(y - \nabla g(y)) \leq g(y) - \frac{1}{2}||\nabla g(y)||_2^2. \tag{7.44}$$

Therefore,

$$\inf g(x) \leq g(y) - \frac{1}{2}||\nabla g(y)||_2^2, \tag{7.45}$$

or

$$g(y) \geq \inf g(x) + \frac{1}{2}||\nabla g(y)||_2^2. \tag{7.46}$$

Now fix y and define the function $h(x)$ by

$$h(x) = g(x) - g(y) - \langle \nabla g(y), x - y \rangle. \tag{7.47}$$

Then $h(x)$ is convex, differentiable, and non-negative,

$$\nabla h(x) = \nabla g(x) - \nabla g(y), \tag{7.48}$$

and $h(y) = 0$, so that $h(x)$ attains its minimum at $x = y$. Applying Inequality (7.46) to the function $h(x)$, with z in the role of x and x in the role of y, we find that

$$\inf h(z) = 0 \leq h(x) - \frac{1}{2}||\nabla h(x)||_2^2. \tag{7.49}$$

From the definition of $h(x)$, it follows that

$$0 \leq g(x) - g(y) - \langle \nabla g(y), x - y \rangle - \frac{1}{2}||\nabla g(x) - \nabla g(y)||_2^2. \tag{7.50}$$

This completes the proof of the implication. ∎

If $g(x)$ is convex and $f(x) = \nabla g(x)$ is L-Lipschitz, then $\frac{1}{L}\nabla g(x)$ is non-expansive, so, by Theorem 7.11, it is firmly non-expansive. It follows that, for $\gamma > 0$, the operator

$$Tx = x - \gamma \nabla g(x) \tag{7.51}$$

is averaged, whenever $0 < \gamma < \frac{2}{L}$. By the KM Theorem 10.2, the iterative sequence $x^{k+1} = Tx^k = x^k - \gamma \nabla g(x^k)$ converges to a minimizer of $g(x)$, whenever minimizers exist.

7.3 Exercises

7.1 *Prove Proposition 7.1.*

7.2 *Show that, if $\hat{x}$ minimizes the function $g(x)$ over all x in R^J, then $x = 0$ is in the sub-differential $\partial g(\hat{x})$.*

7.3 *If $f(x)$ and $g(x)$ are convex functions on R^J, is $f(x) + g(x)$ convex? Is $f(x)g(x)$ convex?*

7.4 *Let $\iota_C(x)$ be the indicator function of the closed convex set C, that is,*

$$\iota_C(x) = \begin{cases} 0, \text{ if } x \in C; \\ +\infty, \text{ if } x \notin C. \end{cases}$$

Show that the subdifferential of the function ι_C at a point c in C is the normal cone to C at the point c, that is, $\partial\iota_C(c) = N_C(c)$, for all c in C.

Chapter 8

Convex Programming

8.1 The Primal Problem

Let f and g_i, $i = 1, ..., I$, be convex functions defined on a non-empty closed convex subset C of R^J. The *primal problem* in *convex programming* (CP) is the following:

$$\text{minimize } f(x), \text{ subject to } g_i(x) \leq 0, \text{ for } i = 1, ..., I. \quad \text{(P)} \tag{8.1}$$

For notational convenience, we define $g(x) = (g_1(x), ..., g_I(x))$. Then (P) becomes

$$\text{minimize } f(x), \text{ subject to } g(x) \leq 0. \quad \text{(P)} \tag{8.2}$$

The *feasible set* for (P) is

$$F = \{x | g(x) \leq 0\}. \tag{8.3}$$

Definition 8.1 *The problem (P) is said to be consistent if F is not empty, and super-consistent if there is x in F with $g_i(x) < 0$ for all $i = 1, ..., I$. Such a point x is then called a Slater point.*

8.1.1 The Perturbed Problem

For each z in R^I let

$$MP(z) = \inf\{f(x) | x \in C, g(x) \leq z\}, \tag{8.4}$$

and $MP = MP(0)$. The convex programming problem (P(z)) is to minimize the function $f(x)$ over x in C with $g(x) \leq z$. The feasible set for (P(z)) is

$$F(z) = \{x | g(x) \leq z\}. \tag{8.5}$$

We shall be interested in properties of the function $MP(z)$, in particular, how the function $MP(z)$ behaves as z moves away from $z = 0$.

For example, let $f(x) = x^2$; the minimum occurs at $x = 0$. Now consider the perturbed problem, minimize $f(x) = x^2$, subject to $x \leq z$. For $z \leq 0$, the minimum of the perturbed problem occurs at $x = z$, and we have $MP(z) = z^2$. For $z > 0$ the minimum of the perturbed problem is the global minimum, which is at $x = 0$, so $MP(z) = 0$. The global minimum of $MP(z)$ also occurs at $z = 0$.

We have the following theorem concerning the function $MP(z)$; see the exercises for related results.

Theorem 8.1 *The function $MP(z)$ is convex and its domain, the set of all z for which $F(z)$ is not empty, is convex. If (P) is super-consistent, then $z = 0$ is an interior point of the domain of $MP(z)$.*

Proof: See [129], Theorem 5.2.6. ∎

From Theorem 7.8 we know that if (P) is super-consistent, then there is a vector d such that

$$MP(z) \geq MP(0) + \langle d, z - 0 \rangle. \tag{8.6}$$

In fact, we can show that, in this case, $d \leq 0$. Suppose that $d_i > 0$ for some i. Since $z = 0$ is in the interior of the domain of $MP(z)$, there is $r > 0$ such that $F(z)$ is not empty for all z with $||z|| < r$. Let $w_j = 0$ for $j \neq i$ and $w_i = r/2$. Then $F(w)$ is not empty and $MP(0) \geq MP(w)$, since $F \subseteq F(w)$. But from Equation (8.6) we have

$$MP(w) \geq MP(0) + \frac{r}{2} d_i > MP(0). \tag{8.7}$$

This is a contradiction, and we conclude that $d \leq 0$.

8.1.2 The Sensitivity Vector

From now on we shall use $\lambda^* = -d$ instead of d. For $z \geq 0$ we have $MP(z) \leq MP(0)$, and

$$\langle \lambda^*, z \rangle \geq MP(0) - MP(z) \geq 0. \tag{8.8}$$

The quantity $\langle \lambda^*, z \rangle$ measures how much $MP(z)$ changes as we increase z away from $z = 0$; for that reason, λ^* is called the *sensitivity vector*, as well as the vector of *Lagrange multipliers*.

The Lagrangian for the problem (P) is the function

$$L(x, \lambda) = f(x) + \sum_{i=1}^{I} \lambda_i g_i(x), \tag{8.9}$$

defined for all x in C and $\lambda \geq 0$.

8.2 From Constrained to Unconstrained

In addition to being a measure of the sensitivity of $MP(z)$ to changes in z, the vector λ^* can be used to convert the original constrained minimization problem (P) into an unconstrained one.

Theorem 8.2 *If the problem (P) has a sensitivity vector $\lambda^* \geq 0$, in particular, when (P) is super-consistent, then*

$$MP(0) = \inf_{x \in C} \Big(f(x) + \langle \lambda^*, g(x) \rangle \Big) = \inf_{x \in C} L(x, \lambda^*). \tag{8.10}$$

Proof: For any x in the set C, the set $F(g(x))$ is non-empty, and

$$MP(g(x)) + \langle \lambda^*, g(x) \rangle \geq MP(0). \tag{8.11}$$

Since

$$f(x) \geq MP(g(x)), \tag{8.12}$$

it follows that

$$f(x) + \langle \lambda^*, g(x) \rangle \geq MP(0). \tag{8.13}$$

Therefore,

$$\inf_{x \in C} \Big(f(x) + \langle \lambda^*, g(x) \rangle \Big) \geq MP(0). \tag{8.14}$$

But

$$\inf_{x \in C} \Big(f(x) + \langle \lambda^*, g(x) \rangle \Big) \leq \inf_{x \in C, g(x) \leq 0} \Big(f(x) + \langle \lambda^*, g(x) \rangle \Big), \tag{8.15}$$

and

$$\inf_{x \in C, g(x) \leq 0} \Big(f(x) + \langle \lambda^*, g(x) \rangle \Big) \leq \inf_{x \in C, g(x) \leq 0} f(x) = MP(0), \tag{8.16}$$

since $\lambda^* \geq 0$ and $g(x) \leq 0$. ∎

Note that the theorem tells us that the two sides of Equation (8.10) are equal, but we cannot conclude from the theorem that if both sides have a minimizer then the minimizers are the same vector.

8.3 Saddle Points

To prepare for our discussion of the Karush-Kuhn-Tucker Theorem and duality, we consider the notion of *saddle points*.

8.3.1 The Primal and Dual Problems

Suppose that X and Y are two non-empty sets and $K : X \times Y \to (-\infty, \infty)$ is a function of two variables. For each x in X, define the function $f(x)$ by the supremum

$$f(x) = \sup_y K(x, y), \tag{8.17}$$

where the supremum is the least upper bound of the real numbers $K(x, y)$, over all y in Y. Then we have

$$K(x, y) \leq f(x), \tag{8.18}$$

for all x. Similarly, for each y in Y, define the function $g(y)$ by

$$g(y) = \inf_x K(x, y); \tag{8.19}$$

here the infimum is the greatest lower bound of the numbers $K(x, y)$, over all x in X. Then we have

$$g(y) \leq K(x, y), \tag{8.20}$$

for all y in Y. Putting together (8.18) and (8.20), we have

$$g(y) \leq K(x, y) \leq f(x), \tag{8.21}$$

for all x and y. Now we consider two problems: the *primal problem* is minimizing $f(x)$ and the *dual problem* is maximizing $g(y)$.

Definition 8.2 *The pair $(\hat{x}, \hat{y})$ is called a saddle point for the function $K(x, y)$ if, for all x and y, we have*

$$K(\hat{x}, y) \leq K(\hat{x}, \hat{y}) \leq K(x, \hat{y}). \tag{8.22}$$

The number $K(\hat{x}, \hat{y})$ is called the saddle value.

For example, the function $K(x, y) = x^2 - y^2$ has $(0, 0)$ for a saddle point, with saddle value zero.

8.3.2 The Main Theorem

We have the following theorem, with the proof left to the reader.

Theorem 8.3 *Let $(\hat{x}, \hat{y})$ be a saddle point for $K(x, y)$. Then $\hat{x}$ solves the primal problem, that is, $\hat{x}$ minimizes $f(x)$, over all x in X, and $\hat{y}$ solves the dual problem, that is, $\hat{y}$ maximizes $g(y)$, over all y in Y. In addition, we have*

$$g(y) \leq K(\hat{x}, \hat{y}) \leq f(x), \tag{8.23}$$

for all x and y, so that the maximum value of $g(y)$ and the minimum value of $f(x)$ are both equal to $K(\hat{x}, \hat{y})$.

8.3.3 A Duality Approach to Optimization

Suppose that our original problem is to minimize a function $f(x)$ over x in some set X. One approach is to find a second set Y and a function $K(x, y)$ of two variables for which Equation (8.17) holds, use Equation (8.19) to construct a second function $g(y)$, defined for y in Y, and then maximize $g(y)$. If a saddle point exists, then, according to the theorem, we have solved the original problem.

8.4 The Karush-Kuhn-Tucker Theorem

The Karush-Kuhn Tucker Theorem gives necessary and sufficient conditions for a vector x^* to be a solution of a super-consistent problem (P).

8.4.1 The KKT Theorem: Saddle-Point Form

This form of the KKT Theorem does not require that the functions involved be differentiable. The *saddle-point* form of the Karush-Kuhn-Tucker (KKT) Theorem is the following.

Theorem 8.4 *Let (P) be super-consistent. Then x^* solves (P) if and only if there is a vector λ^* such that*

- **1)** $\lambda^* \geq 0$*;*
- **2)** $L(x^*, \lambda) \leq L(x^*, \lambda^*) \leq L(x, \lambda^*)$*, for all x and λ;*
- **3)** $\lambda_i^* g_i(x^*) = 0$*, for all $i = 1, ..., I$.*

Proof: Since (P) is super-consistent and x^* solves (P), we know from Theorem 8.2 that there is $\lambda^* \geq 0$ such that

$$f(x^*) = \inf_{x \in C} L(x, \lambda^*). \tag{8.24}$$

We do not yet know that $f(x^*) = L(x^*, \lambda^*)$, however. We do have

$$f(x^*) \leq L(x^*, \lambda^*) = f(x^*) + \langle \lambda^*, g(x^*) \rangle, \tag{8.25}$$

though, and since $\lambda^* \geq 0$ and $g(x^*) \leq 0$, we also have

$$f(x^*) + \langle \lambda^*, g(x^*) \rangle \leq f(x^*). \tag{8.26}$$

Now we can conclude that $f(x^*) = L(x^*, \lambda^*)$ and $\langle \lambda^*, g(x^*) \rangle = 0$. It follows that $\lambda_i^* g_i(x^*) = 0$, for all $i = 1, ..., I$. Since

$$L(x^*, \lambda^*) - L(x^*, \lambda) = \langle \lambda^* - \lambda, g(x^*) \rangle = \langle -\lambda, g(x^*) \rangle \geq 0, \tag{8.27}$$

we also have

$$L(x^*, \lambda) \leq L(x^*, \lambda^*), \tag{8.28}$$

for all $\lambda \geq 0$.

Conversely, suppose that x^* and λ^* satisfy the three conditions of the theorem. First, we show that x^* is feasible for (P), that is, $g(x^*) \leq 0$. Let i be fixed and take λ to have the same entries as λ^*, except that $\lambda_i = \lambda_i^* + 1$. then, $\lambda \geq 0$ and

$$0 \leq L(x^*, \lambda^*) - L(x^*, \lambda) = -g_i(x^*). \tag{8.29}$$

Also,

$$f(x^*) = L(x^*, 0) \leq L(x^*, \lambda^*) = f(x^*) + \langle \lambda^*, g(x^*) \rangle = f(x^*), \tag{8.30}$$

so

$$f(x^*) = L(x^*, \lambda^*) \leq L(x, \lambda^*). \tag{8.31}$$

But we also have

$$L(x^*, \lambda^*) \leq \inf_{x \in C} \Big(f(x) + \langle \lambda^*, g(x) \rangle \Big) \leq \inf_{x \in C, g(x) \leq 0} f(x) = MP(0). \tag{8.32}$$

We conclude that $f(x^*) = MP(0)$, and since x^* is feasible for (P), x^* solves (P). ∎

Condition **3)** is called complementary slackness. If $g_i(x^*) = 0$, we say that the ith constraint is *binding*.

8.4.2 The KKT Theorem- The Gradient Form

Now we assume that the functions $f(x)$ and $g_i(x)$ are differentiable.

Theorem 8.5 *Let (P) be super-consistent. Then x^* solves (P) if and only if there is a vector λ^* such that*

- **1)** $\lambda^* \geq 0$*;*
- **2)** $\lambda_i^* g_i(x^*) = 0$*, for all* $i = 1, ..., I$*;*
- **3)** $\nabla f(x^*) + \sum_{i=1}^{I} \lambda_i^* \nabla g_i(x^*) = 0$.

The proof is similar to the previous one and we omit it. The interested reader should consult [129], p. 185.

8.5 On the Existence of Lagrange Multipliers

As we saw previously, if (P) is super-consistent, then $z = 0$ is in the interior of the domain of the function $MP(z)$, and so the sub-differential of $MP(z)$ is non-empty at $z = 0$. The sub-gradient d was shown to be non-positive and we defined the sensitivity vector, or the vector of Lagrange multipliers, to be $\lambda^* = -d$. Theorem 8.5 tells us that if (P) is super-consistent and x^* solves (P), then the vector $\nabla f(x^*)$ is a non-negative linear combination of the vectors $-\nabla g_i(x^*)$. This sounds like the assertion in Farkas' Lemma.

For any point x, define the set

$$B = \{i | g_i(x) = 0\},$$

and

$$Z(x) = \{z | z^T \nabla g_i(x) \leq 0, \, i \in B(x), \text{ and } z^T \nabla f(x) < 0\}.$$

If $Z(x)$ is empty, then

$$z^T(-\nabla g_i(x)) \geq 0$$

for $i \in B(x)$ implies

$$\nabla f(x) \geq 0,$$

which, by Farkas' Lemma, implies that $\nabla f(x)$ is a non-negative linear combination of the vectors $-\nabla g_i(x)$ for $i \in B(x)$. The objective, then, is to find some condition which, if it holds at the solution x^*, will imply that $Z(x^*)$ is empty; first-order necessary conditions are of this sort. It will then follow that there are non-negative Lagrange multipliers for which

$$\nabla f(x^*) + \sum_{i=1}^{I} \lambda_i^* \nabla g_i(x^*) = 0;$$

for i not in $B(x^*)$ we let $\lambda_i^* = 0$. For more discussion of this issue, see Fiacco and McCormick [85]

8.6 The Problem of Equality Constraints

We consider now what happens when some of the constraints are equalities.

8.6.1 The Problem

Let f and g_i, $i = 1, ..., I$, be differentiable functions defined on R^J. We consider the following problem: minimize $f(x)$, subject to the constraints

$$\begin{cases} g_i(x) = 0, \text{ for } i = 1, ..., m-1; \\ g_i(x) \leq 0, \text{ for } i = m, ..., p. \end{cases} \tag{8.33}$$

If $1 < m - 1 < p$, the constraints are said to be mixed. If $m = 1$, there are only inequality constraints, so, for convex $f(x)$ and $g_i(x)$, the problem is (P), given by (8.1). If $m > 1$, we cannot convert it to a CP problem by rewriting the equality constraints as $g_i(x) \leq 0$ and $-g_i(x) \leq 0$, since then we would lose the convexity property of the constraint functions. Nevertheless, a version of the KKT Theorem holds for such problems.

Definition 8.3 *The feasible set for this problem is the set F of all x satisfying the constraints.*

Definition 8.4 *The problem is said to be consistent if F is not empty.*

Definition 8.5 *Let $\mathcal{I}(x)$ be the set of all indices $1 \leq i \leq p$ for which $g_i(x) = 0$. The point x is regular if the set of gradients $\{\nabla g_i(x) | i \in \mathcal{I}(x)\}$ is linear independent.*

8.6.2 The KKT Theorem for Mixed Constraints

The following version of the KKT Theorem provides a necessary condition for a regular point x^* to be a local constrained minimizer.

Theorem 8.6 *Let x^* be a regular point for the problem in (8.33). If x^* is a local constrained minimizer of $f(x)$, then there is a vector λ^* such that*

- **1)** $\lambda_i^* \geq 0$, *for* $i = m, ..., p$;
- **2)** $\lambda_i^* g_i(x^*) = 0$, *for* $i = m, ..., p$;
- **3)** $\nabla f(x^*) + \sum_{i=1}^{p} \lambda_i^* \nabla g_i(x^*) = 0$.

Note that, if there are some equality constraints, then the vector λ need not be non-negative.

8.6.3 The KKT Theorem for LP

Consider the LP problem (PS): minimize $z = c^T x$, subject to $Ax = b$ and $x \geq 0$. We let

$$z = f(x) = c^T x,$$

$$g_i(x) = b_i - (Ax)_i,$$

for $i = 1, ..., I$, and

$$g_i(x) = -x_j,$$

for $i = I + 1, ..., I + J$ and $j = i - I$. We assume that $I < J$ and that the I by J matrix A has rank I. Then, since $\nabla g_i(x)$ is a^i, the ith column of A^T, the vectors $\{\nabla g_i(x) \, | i = 1, ..., I\}$ are linearly independent and every $x > 0$ is a regular point.

Suppose that a regular point x^* solves (PS). Let λ^* be the vector in R^{I+J} whose existence is guaranteed by Theorem 8.6. Denote by y^* the vector in R^I whose entries are the first I entries of λ^*, and r the non-negative vector in R^J whose entries are the last J entries of λ^*. Then, applying Theorem 8.6, we have $r^T x^* = 0$, $Ax^* = b$, and

$$c - \sum_{i=1}^{I} \lambda_i^* a^i + \sum_{j=1}^{J} r_j(-\delta_j) = 0,$$

or,

$$c - A^T y^* = r \geq 0,$$

where δ_j is the column vector whose jth entry is one and the rest are zero.

The KKT Theorem for this problem is then the following.

Theorem 8.7 *Let A have full rank I. The regular point x^* solves (PS) if and only if there are vectors y^* in R^I and $r \geq 0$ in R^J such that*

- **1)** $Ax^* = b$;
- **2)** $r = c - A^T y^*$;
- **3)** $r^T x^* = 0$.

Then y^* solves (DS).

The first condition in the theorem is *primal feasibility*, the second one is *dual feasibility*, and the third is *complementary slackness*. The first two conditions tell us that x^* is feasible for (PS) and y^* is feasible for (DS). Combining these two conditions with complementary slackness, we can write

$$z^* = c^T x^* = (A^T y^* + r)^T x^* = (A^T y^*)^T x^* + r^T x^* = (y^*)^T b = w^*,$$

so $z^* = w^*$ and there is no duality gap. Invoking Corollary 5.3 to the Weak Duality Theorem, we conclude that x^* and y^* solve their respective problems.

8.7 Two Examples

We illustrate the use of the gradient form of the KKT Theorem with two examples that appeared in the paper of Driscoll and Fox [75].

8.7.1 A Linear Programming Problem

Minimize $f(x_1, x_2) = 3x_1 + 2x_2$, subject to the constraints $2x_1 + x_2 \geq 100$, $x_1 + x_2 \geq 80$, $x_1 \geq 0$ and $x_2 \geq 0$. We define

$$g_1(x_1, x_2) = 100 - 2x_1 - x_2 \leq 0, \tag{8.34}$$

$$g_2(x_1, x_2) = 80 - x_1 - x_2, \tag{8.35}$$

$$g_3(x_1, x_2) = -x_1, \tag{8.36}$$

and

$$g_4(x_1, x_2) = -x_2. \tag{8.37}$$

The Lagrangian is then

$$L(x, \lambda) = 3x_1 + 2x_2 + \lambda_1(100 - 2x_1 - x_2)$$

$$+\lambda_2(80 - x_1 - x_2) - \lambda_3 x_1 - \lambda_4 x_2. \tag{8.38}$$

From the KKT Theorem, we know that if there is a solution x^*, then there is $\lambda^* \geq 0$ with

$$f(x^*) = L(x^*, \lambda^*) \leq L(x, \lambda^*),$$

for all x. For notational simplicity, we write λ in place of λ^*.

Taking the partial derivatives of $L(x, \lambda)$ with respect to the variables x_1 and x_2, we get

$$3 - 2\lambda_1 - \lambda_2 - \lambda_3 = 0, \tag{8.39}$$

and

$$2 - \lambda_1 - \lambda_2 - \lambda_4 = 0. \tag{8.40}$$

The complementary slackness conditions are

$$\lambda_1 = 0 \ , \text{ if } 2x_1 + x_2 \neq 100, \tag{8.41}$$

$$\lambda_2 = 0 \ , \text{ if } x_1 + x_2 \neq 80, \tag{8.42}$$

$$\lambda_3 = 0 \ , \text{ if } x_1 \neq 0, \tag{8.43}$$

and

$$\lambda_4 = 0 \ , \text{ if } x_2 \neq 0. \tag{8.44}$$

A little thought reveals that precisely two of the four constraints must be binding. Examining the six cases, we find that the only case satisfying all the conditions of the KKT Theorem is $\lambda_3 = \lambda_4 = 0$. The minimum occurs at $x_1 = 20$ and $x_2 = 60$ and the minimum value is $f(20, 60) = 180$.

We can use these results to illustrate Theorem 8.2. The sensitivity vector is $\lambda^* = (1, 1, 0, 0)$ and the Lagrangian function at λ^* is

$$L(x, \lambda^*) = 3x_1 + 2x_2 + 1(100 - 2x_1 - x_2). \tag{8.45}$$

In this case, we find that $L(x, \lambda^*) = 180$, for all x.

8.7.2 A Nonlinear Convex Programming Problem

Minimize the function

$$f(x_1, x_2) = (x_1 - 14)^2 + (x_2 - 11)^2,$$

subject to

$$g_1(x_1, x_2) = (x_1 - 11)^2 + (x_2 - 13)^2 - 49 \leq 0,$$

and

$$g_2(x_1, x_2) = x_1 + x_2 - 19 \leq 0.$$

The Lagrangian is then

$$L(x, \lambda) = (x_1 - 14)^2 + (x_2 - 11)^2 +$$

$$\lambda_1\Big((x_1 - 11)^2 + (x_2 - 13)^2 - 49\Big) + \lambda_2\Big(x_1 + x_2 - 19\Big). \tag{8.46}$$

Again, we write λ in place of λ^*. Setting the partial derivatives, with respect to x_1 and x_2, to zero, we get the KKT equations

$$2x_1 - 28 + 2\lambda_1 x_1 - 22\lambda_1 + \lambda_2 = 0, \tag{8.47}$$

and

$$2x_2 - 22 + 2\lambda_1 x_2 - 26\lambda_1 + \lambda_2 = 0. \tag{8.48}$$

The complementary slackness conditions are

$$\lambda_1 = 0 \, , \text{ if } (x_1 - 11)^2 + (x_2 - 13)^2 \neq 49, \tag{8.49}$$

and

$$\lambda_2 = 0\,, \quad \text{if } x_1 + x_2 \neq 19. \tag{8.50}$$

There are four cases to consider. First, if neither constraint is binding, the KKT equations have solution $x_1 = 14$ and $x_2 = 11$, which is not feasible. If only the first constraint is binding, we obtain two solutions, neither feasible. If only the second constraint is binding, we obtain $x_1^* = 11$, $x_2^* = 8$, and $\lambda_2 = 6$. This is the optimal solution. If both constraints are binding, we obtain, with a bit of calculation, two solutions, neither feasible. The minimum value is $f(11, 8) = 18$, and the sensitivity vector is $\lambda^* = (0, 6)$. Using these results, we once again illustrate Theorem 8.2.

The Lagrangian function at λ^* is

$$L(x, \lambda^*) = (x_1 - 14)^2 + (x_2 - 11)^2 + 6(x_1 + x_2 - 19). \tag{8.51}$$

Setting to zero the first partial derivatives of $L(x, \lambda^*)$, we get

$$0 = 2(x_1 - 14) + 6,$$

and

$$0 = 2(x_2 - 11) + 6,$$

so that $x_1^* = 11$ and $x_2^* = 8$. Note that Theorem 8.2 only guarantees that 18 is the infimum of the function $L(x, \lambda^*)$. It does not say that this smallest value must occur at $x = x^*$ or even occurs anywhere; that is, it does not say that $L(x^*, \lambda^*) \leq L(x, \lambda^*)$. This stronger result comes from the KKT Theorem.

8.8 The Dual Problem

The *dual problem* (DP) corresponding to (P) is

$$\text{maximize } h(\lambda) = \inf_{x \in C} L(x, \lambda), \text{ for } \lambda \geq 0. \quad \text{(DP)} \tag{8.52}$$

Let

$$MD = \sup_{\lambda \geq 0} h(\lambda). \tag{8.53}$$

A vector $\lambda \geq 0$ is feasible for (DP) if $h(\lambda) > -\infty$. Then (DP) is consistent if there are feasible λ. Recall that Theorem 8.2 tells us that if a sensitivity vector $\lambda^* \geq 0$ exists, then $h(\lambda^*) = MP$.

8.8.1 When is $MP = MD$?

We have the following theorem.

Theorem 8.8 *Assume that (P) is super-consistent, so that there is a sensitivity vector $\lambda^* \geq 0$, and that MP is finite. Then*

- **1)** $MP = MD$;
- **2)** $MD = h(\lambda^*)$, *so the supremum in Equation (8.53) is attained at* λ^*;
- **3)** *if the infimum in the definition of MP is attained at* x^*, *then* $\langle \lambda^*, g(x^*) \rangle = 0$;
- **4)** *such an* x^* *also minimizes* $L(x, \lambda^*)$ *over* $x \in C$.

Proof: For all $\lambda \geq 0$ we have

$$h(\lambda) = \inf_{x \in C} L(x, \lambda) \leq \inf_{x \in C, g(x) \leq 0} L(x, \lambda) \leq \inf_{x \in C, g(x) \leq 0} f(x) = MP.$$

Therefore, $MD \leq MP$. But we also know that

$$MP = h(\lambda^*) \leq MD,$$

so $MP = MD$, and the supremum in the definition of MD is attained at λ^*. From

$$f(x^*) = MP = \inf_{x \in C} L(x, \lambda^*) \leq \inf_{x \in C, g(x) \leq 0} L(x, \lambda^*)$$

$$\leq L(x^*, \lambda^*) \leq f(x^*),$$

it follows that $\langle \lambda^*, g(x^*) \rangle = 0$. ∎

8.8.2 The Primal-Dual Method

From Theorem 8.8 we see that one approach to solving (P) is to solve (DP) for λ^* and then minimize $L(x, \lambda^*)$ over $x \in C$. This is useful only if solving (DP) is simpler than solving (P) directly. Each evaluation of $h(\lambda)$ involves minimizing $L(x, \lambda)$ over $x \in C$. Once we have found λ^*, we find x^* by minimizing $L(x, \lambda^*)$ over $x \in C$. The advantage is that all the minimizations are over all $x \in C$, not over just the feasible vectors.

8.8.3 An Example

Let $f(x) = \frac{1}{2}||x||_2^2$. The primary problem is to minimize $f(x)$ over all x for which $Ax \geq b$. Then $g_i = b_i - (Ax)_i$, for $i = 1, ..., I$, and the set C is all of R^J. The Lagrangian is then

$$L(x, \lambda) = \frac{1}{2}||x||_2^2 - \lambda^T Ax + \lambda^T b. \tag{8.54}$$

The infimum over x occurs when $x = A^T\lambda$ and so

$$h(\lambda) = \lambda^T b - \frac{1}{2}||A^T\lambda||_2^2. \tag{8.55}$$

For any x satisfying $Ax \geq b$ and any $\lambda \geq 0$ we have $h(\lambda) \leq f(x)$. If x^* is the unique solution of the primal problem and λ^* any solution of the dual problem, we have $f(x^*) = h(\lambda^*)$. The point here is that the constraints in the dual problem are easier to implement in an iterative algorithm, so solving the dual problem is the simpler task.

8.8.4 An Iterative Algorithm for the Dual Problem

In [114] Lent and Censor present the following sequential iterative algorithm for solving the dual problem above. At each step only one entry of the current λ is altered.

Algorithm 8.1 (Lent-Censor) *Let a_i denote the i-th row of the matrix A. Having calculated x^k and $\lambda^k > 0$, let $i = k(\mathrm{mod}\, I) + 1$. Then let*

$$\theta = (b_i - (a_i)^T x^k)/a_i^T a_i, \tag{8.56}$$

$$\delta = \max\{-\lambda_i^k, \omega\theta\}, \tag{8.57}$$

and set

$$\lambda_i^{k+1} = \lambda_i^k + \delta, \tag{8.58}$$

and

$$x^{k+1} = x^k + \delta a_i. \tag{8.59}$$

8.9 Minimum One-Norm Solutions

When the system of linear equations $Ax = b$ is under-determined, it is common practice to seek a solution that also minimizes some objective

function. For example, the *minimum two-norm solution* is the vector x satisfying $Ax = b$ for which the (square of the) two-norm,

$$||x||_2^2 = \sum_{j=1}^{J} x_j^2,$$

is minimized. Alternatively, we may seek the *minimum one-norm solution*, for which the one-norm,

$$||x||_1 = \sum_{j=1}^{J} |x_j|,$$

is minimized.

If the vector x is required to be non-negative, then the one-norm is simply the sum of the entries, and minimizing the one-norm subject to $Ax = b$ becomes a linear programming problem. This is the situation in applications involving image reconstruction.

In *compressed sampling* [73] one seeks a solution of $Ax = b$ having relatively few non-zero entries. The vector x here is not assumed to be non-negative, and the solution is found by minimizing the one-norm, subject to the constraints $Ax = b$. The one-norm is not a linear functional of x, but the problem can still be converted into a linear programming problem.

8.9.1 Reformulation as an LP Problem

The entries of x need not be non-negative, so the problem is not yet a linear programming problem. Let

$$B = [\, A \quad -A \,],$$

and consider the linear programming problem of minimizing the function

$$c^T z = \sum_{j=1}^{2J} z_j,$$

subject to the constraints $z \geq 0$, and $Bz = b$. Let z^* be the solution. We write

$$z^* = \begin{bmatrix} u^* \\ v^* \end{bmatrix}.$$

Then $x^* = u^* - v^*$ minimizes the one-norm, subject to $Ax = b$. To see why this is true, let $\hat{x}$ be the minimum one-norm solution. Write $\hat{u}_j = \hat{x}_j$, if $\hat{x}_j \geq 0$, and $\hat{u}_j = 0$, otherwise. Let $\hat{v}_j = \hat{u}_j - \hat{x}_j$. Then let

$$\hat{z} = \begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix}.$$

The one-norm of $\hat{z}$ is the same as the one-norm of $\hat{x}$, and $B\hat{z} = b$. Therefore, $\hat{x}$ must be a minimum one-norm solution.

8.9.2 Image Reconstruction

In image reconstruction from limited linear-functional data, the vector x is non-negative and arises as a vectorization of a two-dimensional image. The data we have pertaining to x is linear and takes the form $Ax = b$, for some matrix A and vector b. Typically, the problem is under-determined, since the number of entries of x is the number of pixels in the image, which we can make as large as we wish. The problem then is to select, from among all the feasible images, one particular one that has a good chance of being near the correct image. One approach is to take the solution of $Ax = b$ having the minimum Euclidean norm, $||x||_2$. Algorithms such as the projected ART and projected Landweber iterative methods can be used to find such solutions.

Another approach is to find the non-negative solution of $Ax = b$ for which the one-norm,

$$||x||_1 = \sum_{j=1}^{J} |x_j|,$$

is minimized [73]. Since the x_j are to be non-negative, the problem becomes the following: minimize

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

subject to

$$g_i(x) = (Ax)_i - b_i = 0,$$

for $i = 1, ..., I$, and

$$g_i(x) = -x_{i-I} \leq 0,$$

for $i = I + 1, ..., I + J$.

When the system $Ax = b$ is under-determined, the minimum one-norm solution tends to be sparser than the minimum two-norm solution. A simple example will illustrate this point.

Consider the equation $x + 2y = 1$. The minimum two-norm solution is $(0.2, 0.4)$, with two-norm $\frac{\sqrt{5}}{5}$, which is about 0.4472, but one-norm equal to 0.6. The solution $(0, 0.5)$ has two-norm and one-norm equal to 0.5, and the solution $(1.0, 0)$ has two-norm and one-norm equal to 1.0. Therefore, the minimum one-norm solution is $(0, 0.5)$, not $(0.2, 0.4)$.

We can write the one-norm of the vector x as

$$||x||_1 = \sum_{j=1}^{J} \frac{|x_j|^2}{|x_j|}.$$

The PDFT approach to image reconstruction [44] selects the solution of $Ax = b$ that minimizes the weighted two-norm

$$||x||_w^2 = \sum_{j=1}^{J} \frac{|x_j|^2}{p_j} = \sum_{j=1}^{J} |x_j|^2 w_j,$$

where $p_j > 0$ is a prior estimate of the non-negative image x to be reconstructed, and $w_j = p_j^{-1}$. To the extent that p_j accurately models the main features of x, such as which x_j are nearly zero and which are not, the two approaches should give similar reconstructions. The PDFT can be implemented using the ART algorithm (see [138, 139, 140]. For more discussion of one-norm minimization, see the appendix on compressed sensing.

8.10 Exercises

8.1 *Prove Theorem 8.3.*

8.2 *Apply the gradient form of the KKT Theorem to minimize the function* $f(x, y) = (x + 1)^2 + y^2$ *over all* $x \geq 0$ *and* $y \geq 0$.

8.3 ([85]) *Consider the following problem : minimize the function*

$$f(x, y) = |x - 2| + |y - 2|,$$

subject to

$$g(x, y) = y^2 - x \leq 0,$$

and

$$h(x, y) = x^2 + y^2 - 1 = 0.$$

Illustrate this problem graphically, showing lines of constant value of f *and the feasible region of points satisfying the constraints. Where is the solution of the problem? Where is the solution, if the equality constraint is removed? Where is the solution, if both constraints are removed?*

8.4 ([129], Ex. 5.2.9 (a)) *Minimize the function*

$$f(x, y) = \sqrt{x^2 + y^2},$$

subject to

$$x + y \leq 0.$$

Show that the function $MP(z)$ *is not differentiable at* $z = 0$.

8.5 ([129], Ex. 5.2.9 (b)) *Minimize the function*

$$f(x, y) = -2x - y,$$

subject to

$$x + y \leq 1,$$

$$0 \leq x \leq 1,$$

and

$$y \geq 0.$$

Again, show that the function $MP(z)$ is not differentiable at $z = 0$.

8.6 (Duffin; [129], Ex. 5.2.9 (c)) *Minimize the function*

$$f(x, y) = e^{-y},$$

subject to

$$\sqrt{x^2 + y^2} - x \leq 0.$$

Show that the function $MP(z)$ is not continuous at $z = 0$.

8.7 *Apply the theory of convex programming to the primal Quadratic Programming Problem (QP), which is to minimize the function*

$$f(x) = \frac{1}{2} x^T Q x,$$

subject to

$$a^T x \leq c,$$

where a and c are given vectors in R^J.

8.8 *Use Theorem 8.6 to prove that any real N by N symmetric matrix has N mutually orthonormal eigenvectors.*

Chapter 9

Iterative Optimization

We know from beginning calculus that, if we want to optimize a differentiable function $g(x)$ of a single real variable x, we begin by finding the places where the derivative is zero, $g'(x) = 0$. Similarly, if we want to optimize a differentiable function $g(x)$ of a real vector variable x, we begin by finding the places where the gradient is zero, $\nabla g(x) = 0$. Generally, though, this is not the end of the story, for we still have to solve an equation for the optimal x. Unless we are fortunate, solving this equation algebraically may be computationally expensive, or may even be impossible, and we will need to turn to iterative methods. This suggests that we might use iterative methods to minimize $g(x)$ directly, and not solve an equation.

For example, suppose we wish to solve the over-determined system of linear equations $Ax = b$, but we don't know if the system has solutions. In that case, we may wish to minimize the function

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

to get a least-squares solution. We know from linear algebra that if the matrix $A^T A$ is invertible, then the unique minimizer of $g(x)$ is given by

$$x^* = (A^T A)^{-1} A^T b.$$

In many applications, the number of equations and the number of unknowns may be quite large, making it expensive even to calculate the entries of the matrix $A^T A$. In such cases, we can find x^* using an iterative method such as Landweber's Algorithm, which has the iterative step

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

The sequence $\{x^k\}$ converges to x^* for any value of γ in the interval $(0, 2/\lambda_{max})$, where λ_{max} is the largest eigenvalue of the matrix $A^T A$.

In this chapter we shall focus on the optimization of differentiable functions g, leaving to a later chapter the non-differentiable, or non-smooth, case.

9.1 Optimizing Functions of a Single Real Variable

Suppose $g : R \to R$ is differentiable and attains its minimum value. We want to minimize the function $g(x)$. Solving $g'(x) = 0$ to find the optimal $x = x^*$ may not be easy, so we may turn to an iterative algorithm for finding roots of $g'(x)$, or one that minimizes $g(x)$ directly. In the latter case, we may consider an iterative procedure

$$x^{k+1} = x^k - \gamma_k g'(x^k), \tag{9.1}$$

for some sequence $\{\gamma_k\}$ of positive numbers. Such iterative procedures are called *descent algorithms* because, if $g'(x^k) > 0$, then we want to move to the left of x^k, while, if $g'(x^k) < 0$, we want to move to the right.

We shall be particularly interested in algorithms in which $\gamma_k = \gamma$ for all k. We denote by T the operator

$$Tx = x - \gamma g'(x). \tag{9.2}$$

Then, using $g'(x^*) = 0$, we find that

$$|x^* - x^{k+1}| = |Tx^* - Tx^k|. \tag{9.3}$$

9.1.1 Iteration and Operators

The iterative methods we shall consider involve the calculation of a sequence $\{x^k\}$ of vectors in R^J, according to the formula $x^{k+1} = Tx^k$, where T is some function $T : R^J \to R^J$; such functions are called *operators* on R^J. The operator $Tx = x - g'(x)$ above is an operator on R.

Definition 9.1 *An operator T on R^J is continuous at x in the interior of its domain if*

$$\lim_{z \to x} \|Tz - Tx\| = 0.$$

All the operators we shall consider are continuous.

The sequences generated by iterative methods can then be written $\{T^k x^0\}$, where $x = x^0$ is the starting point for the iteration and T^k means apply the operator T k times. If the sequence $\{x^k\}$ converges to a limit vector $\hat{x}$ in the domain of T, then, taking the limit, as $k \to +\infty$, on both sides of

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

and using the continuity of the operator T, we have

$$\hat{x} = T\hat{x},$$

that is, the limit vector $\hat{x}$ is a *fixed point* of T.

Definition 9.2 *A vector x in the domain of the operator T is a fixed point of T if $T\hat{x} = \hat{x}$. The set of all fixed points of T is denoted* $\text{Fix}(T)$.

We have several concerns, when we use iterative methods:

- Does the operator T have any fixed points?
- Does the sequence $\{T^k x_0\}$ converge?
- Does convergence depend on the choice of x^0?
- When the sequence $\{T^k x^0\}$ converges, is the limit a solution to our problem?
- How fast does the sequence $\{T^k x^0\}$ converge?
- How difficult is it to perform a single step, going from x^k to x^{k+1}?
- How does the limit depend on the starting vector x^0?

To answer these questions, we will need to learn about the properties of the particular operator T being used. We begin our study of iterative optimization algorithms with the gradient descent methods, particularly as they apply to convex functions.

9.2 Gradient Operators

Suppose that $g(x)$ is convex and the function $f(x) = g'(x)$ is L-Lipschitz. If $g(x)$ is twice differentiable, this would be the case if

$$0 \leq g''(x) \leq L, \tag{9.4}$$

for all x. If γ is in the interval $(0, \frac{2}{L})$, then the operator $Tx = x - \gamma g'(x)$ is an averaged operator; from the KM Theorem 10.2, we know that the iterative sequence $\{T^k x^0\}$ converges to a minimizer of $g(x)$, whenever a minimizer exists.

If $g(x)$ is convex and $f(x) = g'(x)$ is L-Lipschitz, then $\frac{1}{L}g'(x)$ is non-expansive, so that, by Theorem 7.11 $\frac{1}{L}g'(x)$ is fne and $g'(x)$ is $\frac{1}{L}$-ism. Then, as we shall see later, the operator

$$Tx = x - \gamma g'(x) \tag{9.5}$$

is av whenever $0 < \gamma < \frac{2}{L}$, and so the iterative sequence $x^{k+1} = Tx^k = x^k - \gamma g'(x^k)$ converges to a minimizer of $g(x)$, whenever minimizers exist.

In the next section we extend these results to functions of several variables.

9.3 Optimizing Functions of Several Real Variables

Suppose $g : R^J \to R$ is differentiable and attains its minimum value. We want to minimize the function $g(x)$. Solving $\nabla g(x) = 0$ to find the optimal $x = x^*$ may not be easy, so we may turn to an iterative algorithm for finding roots of $\nabla g(x)$, or one that minimizes $g(x)$ directly. From Cauchy's Inequality, we know that the directional derivative of $g(x)$, at $x = a$, and in the direction of the vector unit vector d, satisfies

$$|g'(a;d)| = |\langle \nabla g(a), d\rangle| \leq \|\nabla g(a)\|_2 \, \|d\|_2,$$

and that $g'(a;d)$ attains its most positive value when the direction d is a positive multiple of $\nabla g(a)$. This suggests *steepest descent* optimization.

Steepest descent iterative optimization makes use of the fact that the direction of greatest increase of $g(x)$ away from $x = x^k$ is in the direction $d = \nabla g(x^k)$. Therefore, we select as the next vector in the iterative sequence

$$x^{k+1} = x^k - \gamma_k \nabla g(x^k), \tag{9.6}$$

for some $\gamma_k > 0$. Ideally, we would choose γ_k so that

$$g(x^k + \gamma_k \nabla g(x^k)) \leq g(x^k + \alpha \nabla g(x^k)),$$

for all α; that is, we would proceed away from x^k, in the direction of $-\nabla g(x^k)$, stopping just as $g(x)$ begins to increase. Then we call this point x^{k+1} and repeat the process. In practice, finding the optimal γ_k is not a simple matter. Instead, one can try a few values of α and accept the best of these few, or one can try to find a constant value γ of the parameter having the property that the iterative step

$$x^{k+1} = x^k - \gamma \nabla g(x^k)$$

leads to a convergent sequence. It is this latter approach that we shall consider here.

We denote by T the operator

$$Tx = x - \gamma \nabla g(x). \tag{9.7}$$

Then, using $\nabla g(x^*) = 0$, we find that

$$\|x^* - x^{k+1}\|_2 = \|Tx^* - Tx^k\|_2. \tag{9.8}$$

We would like to know if there are choices for γ that imply convergence of the iterative sequence. As in the case of functions of a single variable, for functions $g(x)$ that are *convex*, the answer is yes.

If $g(x)$ is convex and $f(x) = \nabla g(x)$ is L-Lipschitz, then $\frac{1}{L}\nabla g(x)$ is non-expansive, Then, as we shall see later, for $\gamma > 0$, the operator

$$Tx = x - \gamma \nabla g(x) \tag{9.9}$$

is averaged, whenever $0 < \gamma < \frac{2}{L}$. It follows that the iterative sequence $x^{k+1} = Tx^k = x^k - \gamma \nabla g(x^k)$ converges to a minimizer of $g(x)$, whenever minimizers exist.

For example, the function $g(x) = \frac{1}{2}\|Ax - b\|_2^2$ is convex and its gradient is

$$f(x) = \nabla g(x) = A^T(Ax - b).$$

A steepest descent algorithm for minimizing $g(x)$ then has the iterative step

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

where the parameter γ_k should be selected so that

$$g(x^{k+1}) < g(x^k).$$

The linear operator that transforms each vector x into A^TAx has the property that

$$\|A^TAx - A^TAy\|_2 \leq \lambda_{max}\|x - y\|_2,$$

where λ_{max} is the largest eigenvalue of the matrix A^TA; this operator is then L-Lipschitz, for $L = \lambda_{max}$. Consequently, the operator that transforms x into $\frac{1}{L}A^TAx$ is non-expansive.

9.4 The Newton-Raphson Approach

The Newton-Raphson approach to minimizing a real-valued function $f : R^J \to R$ involves finding x^* such that $\nabla f(x^*) = 0$.

9.4.1 Functions of a Single Variable

We begin with the problem of finding a root of a function $g : R \to R$. If x^0 is not a root, compute the line tangent to the graph of g at $x = x^0$ and let x^1 be the point at which this line intersects the horizontal axis; that is,

$$x^1 = x^0 - g(x^0)/g'(x^0). \tag{9.10}$$

Continuing in this fashion, we have

$$x^{k+1} = x^k - g(x^k)/g'(x^k). \tag{9.11}$$

This is the *Newton-Raphson algorithm* for finding roots. Convergence, when it occurs, is usually more rapid than gradient descent, but requires that x^0 be sufficiently close to the solution.

Now suppose that $f : R \to R$ is a real-valued function that we wish to minimize by solving $f'(x) = 0$. Letting $g(x) = f'(x)$ and applying the Newton-Raphson algorithm to $g(x)$ gives the iterative step

$$x^{k+1} = x^k - f'(x^k)/f''(x^k). \tag{9.12}$$

This is the Newton-Raphson optimization algorithm. Now we extend these results to functions of several variables.

9.4.2 Functions of Several Variables

The Newton-Raphson algorithm for finding roots of functions $g : R^J \to R^J$ has the iterative step

$$x^{k+1} = x^k - [\mathcal{J}(g)(x^k)]^{-1} g(x^k), \tag{9.13}$$

where $\mathcal{J}(g)(x)$ is the Jacobian matrix of first partial derivatives, $\frac{\partial g_m}{\partial x_j}(x^k)$, for $g(x) = (g_1(x), ..., g_J(x))^T$.

To minimize a function $f : R^J \to R$, we let $g(x) = \nabla f(x)$ and find a root of g. Then the Newton-Raphson iterative step becomes

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k), \tag{9.14}$$

where $\nabla^2 f(x) = \mathcal{J}(g)(x)$ is the Hessian matrix of second partial derivatives of f.

The quadratic approximation to $f(x)$ around the point x^k is

$$f(x) \approx f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k)(x - x^k).$$

The right side of this equation attains its minimum value when

$$0 = \nabla f(x^k) + \nabla^2 f(x^k)(x - x^k),$$

that is, when $x = x^{k+1}$ as given by Equation (9.14).

If $f(x)$ is a quadratic function, that is,

$$f(x) = x^T Q x + x^T b + c,$$

for constant invertible matrix Q and constant vectors b and c, then the Newton-Raphson iteration converges to the answer in one step. Therefore, if $f(x)$ is close to quadratic, the convergence should be reasonably rapid. This leads to the notion of *self-concordant functions*, for which the third derivative of $f(x)$ is small, relative to the second derivative [122].

9.5 Approximate Newton-Raphson Methods

To implement the NR method in this case, at each step of the iteration we need to solve a system of equations involving the Hessian matrix for f. There are many iterative procedures designed to retain much of the advantages of the NR method, but without the use of the Hessian matrix, or, indeed, without the use of the gradient. These methods are discussed in most texts on numerical methods [122]. We sketch briefly some of these approaches.

9.5.1 Avoiding the Hessian Matrix

Quasi-Newton methods, designed to avoid having to calculate the Hessian matrix, are often used instead of the Newton-Raphson algorithm. The iterative step of the quasi-Newton methods is

$$x^{k+1} = x^k - B_k^{-1}\nabla f(x^k), \tag{9.15}$$

where the matrix B_k is an approximation of $\nabla^2 f(x^k)$ that is easier to compute.

In the case of $g : R \to R$, the second derivative of $g(x)$ is approximately

$$g''(x^k) \approx \frac{g'(x^k) - g'(x^{k-1})}{x^k - x^{k-1}}. \tag{9.16}$$

This suggests that, for the case of functions of several variables, the matrix B_k should be selected so that

$$B_k(x^k - x^{k-1}) = \nabla f(x^k) - \nabla f(x^{k-1}). \tag{9.17}$$

In addition to satisfying Equation (9.17), the matrix B_k should also be symmetric and positive-definite. Finally, we should be able to obtain B_{k+1} relatively easily from B_k.

The BFGS Method

The Broyden, Fletcher, Goldfarb, and Shanno (BFGS) method uses the rank-two update formula

$$B_{k+1} = B_k - \frac{(B_k s^k)(B_k s^k)^T}{(s^k)^T B_k s^k} + \frac{y^k (y^k)^T}{(y^k)^T s^k}, \tag{9.18}$$

with

$$s^k = x^{k+1} - x^k, \tag{9.19}$$

and

$$y^k = \nabla f(x^{k+1}) - \nabla f(x^k). \tag{9.20}$$

The Broyden Class

A general class of update methods, known as the Broyden class, uses the update formula

$$B_{k+1} = B_k - \frac{(B_k s^k)(B_k s^k)^T}{(s^k)^T B_k s^k} + \frac{y^k (y^k)^T}{(y^k)^T s^k} + \phi((s^k)^T B_k s^k) u^k (u^k)^T, \quad (9.21)$$

with ϕ a scalar and

$$u^k = \frac{y^k}{(y^k)^T s^k} - \frac{B_k s^k}{(s^k)^T B_k s^k}. \quad (9.22)$$

When $\phi = 0$ we get the BFGS method, while the choice of $\phi = 1$ gives the Davidon, Fletcher, and Powell (DFP) method.

Note that for the updates in the Broyden class, the matrix B_{k+1} has the form

$$B_{k+1} = B_k + x^k (x^k)^T + z^k (z^k)^T,$$

for certain vectors x^k and z^k. Therefore, using the Sherman-Morrison-Woodbury Identity (see Exercise 9.40), the inverse of B_{k+1} can be obtained easily from the inverse of B_k.

9.5.2 Avoiding the Gradient

Quasi-Newton methods use an approximation of the Hessian matrix that is simpler to calculate, but still employ the gradient at each step. For functions $g : R \to R$, the derivative can be approximated by a *finite difference*, that is,

$$g'(x^k) \approx \frac{g(x^k) - g(x^{k-1})}{x^k - x^{k-1}}. \quad (9.23)$$

In the case of functions of several variables, the gradient vector can be approximated by using a finite-difference approximation for each of the first partial derivatives.

9.6 Derivative-Free Methods

In many important applications, calculating values of the function to be optimized is expensive and calculating gradients impractical. In such cases, it is common to use *direct-search methods*. Generally, these are iterative methods that are easy to program, do not employ derivatives or their approximations, require relatively few function evaluations, and are useful even when the measurements are noisy.

9.6.1 Multi-directional Search Algorithms

Methods such as the *multi-directional search* algorithms begin with the values of the function $f(x)$ at $J+1$ points, where x is in R^J, and then use these values to move to a new set of points. These points are chosen to describe a simplex pattern in R^J, that is, they do not all lie on a single hyperplane in R^J. For that reason, these methods are sometimes called *simplex* methods, although they are unrelated to Dantzig's method of the same name. The Nelder-Mead algorithm [123, 109, 119] is one such simplex algorithm.

9.6.2 The Nelder-Mead Algorithm

For simplicity, we follow McKinnon [119] and describe the Nelder-Mead (NM) algorithm only for the case of $J = 2$. The NM algorithm begins with the choice of vertices:

ORDER: obtain b, s, and w, with

$$f(b) \leq f(s) \leq f(w).$$

Then take

$$m = \frac{1}{2}(b+s).$$

Let the *search line* be

$$L(\rho) = m + \rho(m - w),$$

and

$$r = L(1) = 2m - w.$$

- **{if** $f(r) < f(b)$**}** let $e = L(2)$. If $f(e) < f(b)$ *accept* e; otherwise *accept* r.
- **{if** $f(b) \leq f(r)$**}** then
  - **{if** $f(r) < f(s)$**}** *accept* r.
  - **{if** $f(s) \leq f(r)$**}**
    - **{if** $f(r) < f(w)$**}** let $c = L(0.5)$
      - **{if** $f(c) \leq f(r)$**}** *accept* c;
      - **{if** $f(r) < f(c)$**}** go to SHRINK.
    - **{if** $f(w) \leq f(r)$**}** let $c = L(-0.5)$.
      - **{if** $f(c) < f(w)$**}** *accept* c; otherwise go to SHRINK.

Replace w with the *accepted* point and go to ORDER.
SHRINK: Replace s with $\frac{1}{2}(s+b)$ and w with $\frac{1}{2}(w+b)$; go to ORDER.

9.6.3 Comments on the Nelder-Mead Algorithm

Although the Nelder-Mead algorithm is quite popular in many areas of applications, relatively little of a theoretical nature is known. The interested reader is directed to the papers [109, 119], as well as to more recent work by Margaret Wright of NYU.

9.7 Rates of Convergence

In this section we illustrate the concept of *rate of convergence* [22] by considering the fixed-point iteration $x_{k+1} = g(x_k)$, for the twice continuously differentiable function $g : R \to R$. We suppose that $g(z) = z$ and we are interested in the distance $|x_k - z|$.

9.7.1 Basic Definitions

Definition 9.3 *Suppose the sequence $\{x_k\}$ converges to z. If there are positive constants λ and α such that*

$$\lim_{k\to\infty} \frac{|x_{k+1} - z|}{|x_k - z|^\alpha} = \lambda, \tag{9.24}$$

then $\{x_k\}$ is said to converge to z with order α and asymptotic error constant λ. If $\alpha = 1$, the convergence is said to be linear; if $\alpha = 2$, the convergence is said to be quadratic.

9.7.2 Illustrating Quadratic Convergence

According to the Mean Value Theorem,

$$g(x) = g(z) + g'(z)(x - z) + \frac{1}{2}g''(c)(x - z)^2, \tag{9.25}$$

for some c between x and z. Suppose now that $x_k \to z$ and, in addition, $g'(z) = 0$. Then we have

$$x_{k+1} = g(x_k) = z + \frac{1}{2}g''(c_k)(x_k - z)^2, \tag{9.26}$$

for some c_k between x_k and z. Therefore,

$$|x_{k+1} - z| = \frac{1}{2}|g''(c_k)|\,|x_k - z|^2, \tag{9.27}$$

and the convergence is quadratic, with $\lambda = |g''(z)|$.

9.7.3 Motivating the Newton-Raphson Method

Suppose that we are seeking a root z of the function $f : R \to R$. We define

$$g(x) = x - h(x)f(x), \tag{9.28}$$

for some function $h(x)$ to be determined. Then $f(z) = 0$ implies that $g(z) = z$. In order to have quadratic convergence of the iterative sequence $x_{k+1} = g(x_k)$, we want $g'(z) = 0$. From

$$g'(x) = 1 - h'(x)f(x) - h(x)f'(x), \tag{9.29}$$

it follows that we want

$$h(z) = 1/f'(z). \tag{9.30}$$

Therefore, we choose

$$h(x) = 1/f'(x), \tag{9.31}$$

so that

$$g(x) = x - f(x)/f'(x). \tag{9.32}$$

The iteration then takes the form

$$x_{k+1} = g(x_k) = x_k - f(x_k)/f'(x_k), \tag{9.33}$$

which is the Newton-Raphson iteration.

9.8 Feasible-Point Methods

We consider now the problem of minimizing the function $f(x) : R^J \to R$, subject to the equality constraints $Ax = b$, where A is an I by J real matrix, with rank I and $I < J$. The two methods we consider here are *feasible-point methods*, also called *interior-point methods*.

9.8.1 The Reduced Newton-Raphson Method

The first method we consider is a modification of the Newton-Raphson method, in which we begin with a feasible point and each NR step is projected into the null space of the matrix A, to maintain the condition $Ax = b$. The discussion here is taken from [122].

Let $\hat{x}$ be a *feasible point*, that is, $A\hat{x} = b$. Then $x = \hat{x} + p$ is also feasible if p is in the null space of A, that is, $Ap = 0$. Let Z be a $J - I$ by J matrix

whose columns form a basis for the null space of A. We want $p = Zv$ for some v. The best v will be the one for which the function

$$\phi(v) = f(\hat{x} + Zv)$$

is minimized. We can apply to the function $\phi(v)$ the steepest descent method, or Newton-Raphson or any other minimization technique. The steepest descent method, applied to $\phi(v)$, is called the *reduced steepest descent method*; the Newton-Raphson method, applied to $\phi(v)$, is called the *reduced Newton-Raphson method*. The gradient of $\phi(v)$, also called the *reduced gradient*, is

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

and the Hessian matrix of $\phi(v)$, also called the *reduced Hessian matrix*, is

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

where $x = \hat{x} + Zv$, so algorithms to minimize $\phi(v)$ can be written in terms of the gradient and Hessian of f itself.

An Example

Consider the problem of minimizing the function

$$f(x) = \frac{1}{2}x_1^2 - \frac{1}{2}x_3^2 + 4x_1x_2 + 3x_1x_3 - 2x_2x_3,$$

subject to

$$x_1 - x_2 - x_3 = -1.$$

Let $\hat{x} = [1, 1, 1]^T$. Then the matrix A is $A = [1, -1, -1]$ and the vector b is $b = [-1]$. Let the matrix Z be

$$Z = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{9.34}$$

The reduced gradient at $\hat{x}$ is then

$$Z^T \nabla f(\hat{x}) = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 8 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 10 \\ 8 \end{bmatrix}, \tag{9.35}$$

and the reduced Hessian matrix at $\hat{x}$ is

$$Z^T \nabla^2 f(\hat{x}) Z = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 4 & 3 \\ 4 & 0 & -2 \\ 3 & -2 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 9 & 6 \\ 6 & 6 \end{bmatrix}. \tag{9.36}$$

Then the reduced Newton-Raphson equation yields

$$v = \begin{bmatrix} -2/3 \\ -2/3 \end{bmatrix}, \tag{9.37}$$

and the reduced Newton-Raphson direction is

$$p = Zv = \begin{bmatrix} -4/3 \\ -2/3 \\ -2/3 \end{bmatrix}. \tag{9.38}$$

Since the function $\phi(v)$ is quadratic, one reduced Newton-Raphson step suffices to obtain the solution, $x_* = [-1/3, 1/3, 1/3]^T$.

9.8.2 A Primal-Dual Approach

In this approach we begin with the Lagrangian,

$$L(x, \lambda) = f(x) + \lambda^T (b - Ax).$$

Setting to zero the x-gradient of $L(x, \lambda)$, we have to solve the equations

$$\nabla f(x) - A^T \lambda = 0$$

and

$$Ax = b.$$

We define the function $G(x, \lambda)$ taking values in R^2 to be

$$G(x, \lambda) = (\nabla f(x) - A^T \lambda, Ax - b).$$

We then apply the NR method to find a zero of the function G. The Jacobian matrix for G is

$$J_G(x, \lambda) = \begin{bmatrix} \nabla^2 f(x) & -A^T \\ A & 0 \end{bmatrix},$$

so one step of the NR method is

$$(x^{k+1}, \lambda^{k+1})^T = (x^k, \lambda^k)^T - J_G(x^k, \lambda^k)^{-1} G(x^k, \lambda^k). \tag{9.39}$$

Therefore

$$A(x^{k+1} - x^k) = 0.$$

If we begin with a feasible x^0, that is, with $Ax^0 = b$, then each successive step of the Newton-Raphson iteration produces a feasible x^k.

9.9 Simulated Annealing

In this chapter we have focused on the minimization of convex functions. For such functions, a local minimum is necessarily a global one. For non-convex functions, this is not the case. For example, the function $f(x) = x^4 - 8x^3 + 20x^2 - 16.5x + 7$ has a local minimum around $x = 0.6$ and a global minimum around $x = 3.5$. The descent methods we have discussed can get caught at a local minimum that is not global, since we insist on always taking a step that reduces $f(x)$. The *simulated annealing algorithm* [1, 121], also called the *Metropolis algorithm* is sometimes able to avoid being trapped at a local minimum by permitting an occasional step that increases $f(x)$. The name comes from the analogy with the physical problem of lowering the energy of a solid by first raising the temperature, to bring the particles into a disorganized state, and then gradually reducing the temperature, so that a more organized state is achieved.

Suppose we have calculated x^k. We now generate a random direction and a small random step length. If the new vector $x^k + \Delta x$ makes $f(x)$ smaller, we accept the vector as x^{k+1}. If not, then we accept this vector, with probability

$$Prob(\text{accept}) = \exp\Big(\frac{f(x^k) - f(x^k + \Delta x)}{c_k}\Big),$$

where $c_k > 0$, known as the *temperature*, is chosen by the user. As the iteration proceeds, the temperature c_k is gradually reduced, making it easier to accept increases in $f(x)$ early in the process, but harder later. How to select the temperatures is an art, not a science.

9.10 Exercises

9.1 *Apply the Newton-Raphson method to obtain an iterative procedure for finding $\sqrt{a}$, for any positive a. For which x^0 does the method converge? There are two answers, of course; how does the choice of x^0 determine which square root becomes the limit?*

9.2 *Apply the Newton-Raphson method to obtain an iterative procedure for finding $a^{1/3}$, for any real a. For which x^0 does the method converge?*

9.3 *Extend the Newton-Raphson method to complex variables. Redo the previous exercises for the case of complex a. For the complex case, a has two square roots and three cube roots. How does the choice of x^0 affect the limit? Warning: The case of the cube root is not as simple as it may appear.*

9.4 (The Sherman-Morrison-Woodbury Identity) *Let A be an invertible matrix. Show that, if $\omega = 1 + v^T A^{-1} u \neq 0$, then $A + uv^T$ is invertible and*

$$(A + uv^T)^{-1} = A^{-1} - \frac{1}{\omega} A^{-1} u v^T A^{-1}. \tag{9.40}$$

9.5 *Use the reduced Newton-Raphson method to minimize the function $\frac{1}{2} x^T Q x$, subject to $Ax = b$, where*

$$Q = \begin{bmatrix} 0 & -13 & -6 & -3 \\ -13 & 23 & -9 & 3 \\ -6 & -9 & -12 & 1 \\ -3 & 3 & 1 & -1 \end{bmatrix},$$

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

and

$$b = \begin{bmatrix} 3 \\ 2 \end{bmatrix}.$$

Start with

$$x^0 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

9.6 *Use the reduced steepest descent method with an exact line search to solve the problem in the previous exercise.*

Chapter 10

Operators

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

10.1 Operators

For most of the iterative algorithms we shall consider, the iterative step is

$$x^{k+1} = Tx^k, \tag{10.1}$$

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

Our approach here will be to identify several classes of operators for which the iterative sequence is known to converge, to examine the convergence theorems that apply to each class, to describe several applied prob-

lems that can be solved by iterative means, to present iterative algorithms for solving these problems, and to establish that the operator involved in each of these algorithms is a member of one of the designated classes.

10.2 Strict Contractions

The strict contraction operators are perhaps the best known class of operators associated with iterative algorithms.

Definition 10.1 *An operator T on R^J is Lipschitz continuous, with respect to a vector norm $||\cdot||$, or L-Lipschitz, if there is a positive constant L such that*

$$||Tx - Ty|| \leq L||x - y||, \tag{10.2}$$

for all x and y in R^J.

Definition 10.2 *An operator T on R^J is a strict contraction (sc), with respect to a vector norm $||\cdot||$, if there is $r \in (0, 1)$ such that*

$$||Tx - Ty|| \leq r||x - y||, \tag{10.3}$$

for all vectors x and y.

For strict contractions, we have the Banach-Picard Theorem [78]:

Theorem 10.1 *Let T be sc. Then, there is a unique fixed point of T and, for any starting vector x^0, the sequence $\{T^k x^0\}$ converges to the fixed point.*

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

Lemma 10.1 *Let T be an affine operator, that is, T has the form $Tx = Bx + d$, where B is a linear operator, and d is a fixed vector. Then T is a strict contraction if and only if $||B||$, the induced matrix norm of B, is less than one.*

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

When B is not Hermitian, it is not as easy to determine if the affine operator T is sc with respect to a given norm. Instead, we often tailor the norm to the operator T. Suppose that B is a diagonalizable matrix, that is, there is a basis for R^J consisting of eigenvectors of B. Let $\{u^1, ..., u^J\}$

be such a basis, and let $Bu^j = \lambda_j u^j$, for each $j = 1, ..., J$. For each x in R^J, there are unique coefficients a_j so that

$$x = \sum_{j=1}^{J} a_j u^j. \tag{10.4}$$

Then let

$$||x|| = \sum_{j=1}^{J} |a_j|. \tag{10.5}$$

Lemma 10.2 *The expression $||\cdot||$ in Equation (10.5) defines a norm on R^J. If $\rho(B) < 1$, then the affine operator T is sc, with respect to this norm.*

According to Lemma 19.6, for any square matrix B and any $\epsilon > 0$, there is a vector norm for which the induced matrix norm satisfies $||B|| \leq \rho(B) + \epsilon$. Therefore, if B is an arbitrary square matrix with $\rho(B) < 1$, there is a vector norm with respect to which B is sc.

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

10.3 Two Useful Identities

The identities in the next two lemmas relate an arbitrary operator T to its complement, $G = I - T$, where I denotes the identity operator. These identities will allow us to transform properties of T into properties of G that may be easier to work with. A simple calculation is all that is needed to establish the following lemma.

Lemma 10.3 *Let T be an arbitrary operator T on R^J and $G = I - T$. Then*

$$||x - y||_2^2 - ||Tx - Ty||_2^2 = 2(\langle Gx - Gy, x - y\rangle) - ||Gx - Gy||_2^2. \tag{10.6}$$

Lemma 10.4 *Let T be an arbitrary operator T on R^J and $G = I - T$. Then*

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

$$\langle Gx - Gy, x - y\rangle - ||Gx - Gy||_2^2. \tag{10.7}$$

Proof: Use the previous lemma. ∎

10.4 Orthogonal Projection Operators

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

10.4.1 Properties of the Operator P_C

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

$$\langle P_C x - x, c - P_C x \rangle \geq 0, \tag{10.8}$$

for all c in C; see Proposition 4.4.

P_C is Non-expansive

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

$$||Tx - Ty|| \leq ||x - y||. \tag{10.9}$$

Lemma 10.5 *The orthogonal projection operator $T = P_C$ is non-expansive, with respect to the Euclidean norm, that is,*

$$||P_C x - P_C y||_2 \leq ||x - y||_2, \tag{10.10}$$

for all x and y.

Proof: Use Inequality (10.8) to get

$$\langle P_C y - P_C x, P_C x - x \rangle \geq 0, \tag{10.11}$$

and

$$\langle P_C x - P_C y, P_C y - y \rangle \geq 0. \tag{10.12}$$

Add the two inequalities to obtain

$$\langle P_C x - P_C y, x - y \rangle \geq ||P_C x - P_C y||_2^2, \tag{10.13}$$

and use the Cauchy Inequality. ∎

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

P_C is Firmly Non-expansive

Definition 10.3 *An operator T is said to be firmly non-expansive (fne) if*

$$\langle Tx - Ty, x - y\rangle \geq ||Tx - Ty||_2^2, \tag{10.14}$$

for all x and y in R^J.

Lemma 10.6 *An operator T is fne if and only if $G = I - T$ is fne.*

Proof: Use the identity in Equation (10.7). ∎

From Equation (10.13), we see that the operator $T = P_C$ is not simply ne, but fne, as well. A good source for more material on these topics is the book by Goebel and Reich [91].

The Search for Other Properties of P_C

The class of non-expansive operators is too large for our purposes; the operator $Tx = -x$ is non-expansive, but the sequence $\{T^k x^0\}$ does not converge, in general, even though a fixed point, $x = 0$, exists. The class of firmly non-expansive operators is too small for our purposes. Although the convergence of the iterative sequence $\{T^k x^0\}$ to a fixed point does hold for firmly non-expansive T, whenever fixed points exist, the product of two or more fne operators need not be fne; that is, the class of fne operators is not *closed to finite products*. This poses a problem, since, as we shall see, products of orthogonal projection operators arise in several of the algorithms we wish to consider. We need a class of operators smaller than the ne ones, but larger than the fne ones, closed to finite products, and for which the sequence of iterates $\{T^k x^0\}$ will converge, for any x^0, whenever fixed points exist. The class we shall consider is the class of *averaged* operators.

10.5 Averaged Operators

The term 'averaged operator' appears in the work of Baillon, Bruck and Reich [20, 8]. There are several ways to define averaged operators. One way is in terms of the complement operator.

Definition 10.4 *An operator G on R^J is called ν-inverse strongly monotone (ν-ism)[92] (also called co-coercive in [65]) if there is $\nu > 0$ such that*

$$\langle Gx - Gy, x - y\rangle \geq \nu||Gx - Gy||_2^2. \tag{10.15}$$

Lemma 10.7 *An operator T is ne if and only if its complement $G = I - T$ is $\frac{1}{2}$-ism, and T is fne if and only if G is 1-ism, and if and only if G is fne. Also, T is ne if and only if $F = (I + T)/2$ is fne. If G is ν-ism and $\gamma > 0$ then the operator γG is $\frac{\nu}{\gamma}$-ism.*

Definition 10.5 *An operator T is called averaged (av) if $G = I - T$ is ν-ism for some $\nu > \frac{1}{2}$. If G is $\frac{1}{2\alpha}$-ism, for some $\alpha \in (0, 1)$, then we say that T is α-av.*

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

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

Lemma 10.8 *An operator T is av if and only if, for some operator N that is non-expansive in the Euclidean norm, and $\alpha \in (0, 1)$, we have*

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

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

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

is non-expansive.

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

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

Proposition 10.1 *An operator F is firmly non-expansive if and only if $F = \frac{1}{2}(I + N)$, for some non-expansive operator N.*

10.5.1 Gradient Operators

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

Definition 10.6 *An operator T on R^J is monotone if*

$$\langle Tx - Ty, x - y\rangle \geq 0, \tag{10.16}$$

for all x and y.

Firmly non-expansive operators on R^J are monotone operators. Let $g(x) : R^J \to R$ be a differentiable convex function and $f(x) = \nabla g(x)$ its gradient. The operator ∇g is also monotone. If ∇g is non-expansive, then, according to Theorem 7.11, ∇g is fne . If, for some $L > 0$, ∇g is L-Lipschitz, for the 2-norm, that is,

$$||\nabla g(x) - \nabla g(y)||_2 \leq L||x - y||_2, \tag{10.17}$$

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

10.5.2 The Krasnoselskii-Mann Theorem

For any operator T that is averaged, convergence of the sequence $\{T^k x^0\}$ to a fixed point of T, whenever fixed points of T exist, is guaranteed by the Krasnoselskii-Mann (KM) Theorem [116]:

Theorem 10.2 *Let T be averaged. Then the sequence $\{T^k x^0\}$ converges to a fixed point of T, whenever Fix(T) is non-empty.*

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

$$x^{k+1} = Tx^k = (1 - \alpha)x^k + \alpha N x^k. \tag{10.18}$$

The identity in Equation (10.6) is the key to proving Theorem 10.2.

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

$$||z - x^k||_2^2 - ||Tz - x^{k+1}||_2^2 = 2\langle Gz - Gx^k, z - x^k\rangle - ||Gz - Gx^k||_2^2. \tag{10.19}$$

Since, by Lemma 10.8, G is $\frac{1}{2\alpha}$-ism, we have

$$||z - x^k||_2^2 - ||z - x^{k+1}||_2^2 \geq (\frac{1}{\alpha} - 1)||x^k - x^{k+1}||_2^2. \tag{10.20}$$

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

A version of the KM Theorem 10.2, with variable coefficients, appears in Reich's paper [130].

10.6 Affine Linear Operators

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

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

10.6.1 The Hermitian Case

As we shall see later, when B is Hermitian, we can determine if B belongs to these classes by examining its eigenvalues λ:

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

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

10.7 Paracontractive Operators

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

Definition 10.7 *An operator T is called paracontractive (pc), with respect to a given norm, if, for every fixed point y of T, we have*

$$||Tx - y|| < ||x - y||, \tag{10.21}$$

unless $Tx = x$.

Paracontractive operators are studied by Censor and Reich in [61].

Proposition 10.2 *The operators $T = P_C$ are paracontractive, with respect to the Euclidean norm.*

Proof: It follows from Cauchy's Inequality that

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

with equality if and only if

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

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

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

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

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

∎

When we ask if a given operator T is pc, we must specify the norm. We often construct the norm specifically for the operator involved, as we did earlier in our discussion of strict contractions, in Equation (10.5). To illustrate, we consider the case of affine operators.

10.7.1 Linear and Affine Paracontractions

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

Lemma 10.9 *A square matrix B is diagonalizable if all its eigenvalues are distinct.*

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

$$x^m = a_1 x^1 + ... a_{m-1} x^{m-1}, \tag{10.22}$$

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

$$\lambda_m x^m = a_1 \lambda_m x^1 + ... a_{m-1} \lambda_m x^{m-1}. \tag{10.23}$$

From

$$\lambda_m x^m = Ax^m = a_1 \lambda_1 x^1 + ... a_{m-1} \lambda_{m-1} x^{m-1}, \tag{10.24}$$

it follows that

$$a_1(\lambda_m - \lambda_1)x^1 + ... + a_{m-1}(\lambda_m - \lambda_{m-1})x^{m-1} = 0, \tag{10.25}$$

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

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

$$B = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{10.26}$$

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

Proposition 10.3 *Let T be an affine linear operator whose linear part B is diagonalizable, and $|\lambda| < 1$ for all eigenvalues λ of B that are not equal to one. Then the operator T is pc, with respect to the norm given by Equation (10.5).*

Proof: This is Exercise 10.8. ∎

We see from Proposition 10.3 that, for the case of affine operators T whose linear part is not Hermitian, instead of asking if T is av, we can ask if T is pc; since B will almost certainly be diagonalizable, we can answer this question by examining the eigenvalues of B.

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

10.7.2 The Elsner-Koltracht-Neumann Theorem

Our interest in paracontractions is due to the Elsner-Koltracht-Neumann (EKN) Theorem [81]:

Theorem 10.3 *Let T be pc with respect to some vector norm. If T has fixed points, then the sequence $\{T^k x^0\}$ converges to a fixed point of T, for all starting vectors x^0.*

We follow the development in [81].

Theorem 10.4 *Suppose that there is a vector norm on R^J, with respect to which each T_i is a pc operator, for $i = 1, ..., I$, and that $F = \cap_{i=1}^I \text{Fix}(T_i)$ is not empty. For $k = 0, 1, ...$, let $i(k) = k(\text{mod}\, I) + 1$, and $x^{k+1} = T_{i(k)} x^k$. The sequence $\{x^k\}$ converges to a member of F, for every starting vector x^0.*

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

$$||x^{k+1} - y|| = ||T_{i(k)}x^k - y|| \leq ||x^k - y||, \tag{10.27}$$

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

$$||T_i x^* - y|| = ||x^* - y||, \tag{10.28}$$

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

Corollary 10.1 *If T is pc with respect to some vector norm, and T has fixed points, then the iterative sequence $\{T^k x^0\}$ converges to a fixed point of T, for every starting vector x^0.*

Corollary 10.2 *If $T = T_I T_{I-1} \cdots T_2 T_1$, and $F = \cap_{i=1}^{I} \mathrm{Fix}\,(T_i)$ is not empty, then $F = \mathrm{Fix}\,(T)$.*

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

Corollary 10.3 *The product T of two or more pc operators T_i, $i = 1, ..., I$ is again a pc operator, if $F = \cap_{i=1}^{I} \mathrm{Fix}\,(T_i)$ is not empty.*

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

$$||Tx - y|| = ||x - y||. \tag{10.29}$$

Then, since

$$||T_I(T_{I-1} \cdots T_1)x - y|| \leq ||T_{I-1} \cdots T_1 x - y|| \leq ... \leq ||T_1 x - y|| \leq ||x - y||, \tag{10.30}$$

it follows that

$$||T_i x - y|| = ||x - y||, \tag{10.31}$$

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

10.8 Exercises

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

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

$$||x^k - x^{k+n}|| \leq ||x^k - x^{k+1}|| + ... + ||x^{k+n-1} - x^{k+n}||, \tag{10.32}$$

and use

$$||x^{k+m} - x^{k+m+1}|| \leq r^m ||x^k - x^{k+1}||. \tag{10.33}$$

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

10.3 *Suppose that we want to solve the equation*

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

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

10.4 *Prove Lemma 10.2.*

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

10.6 *Prove Lemma 10.7.*

10.7 *Prove Proposition 10.1.*

10.8 *Prove Proposition 10.3.*

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

Chapter 11

The Algebraic Reconstruction Technique

In this chapter and the next several, we study iterative algorithms for solving systems of linear equations, sometimes subject to restrictions on the matrix, the solution, or both. Such problems can arise as optimization problems in their own right, and also as part of procedures for solving other optimization problems.

In our discussion of the geometric programming problem we saw that we need to solve a system of linear equations, subject to positivity constraints, in order to solve the original GP problem. In the simplex algorithm, the Newton-Raphson algorithm and elsewhere, there is a system of linear equations to be solved at each step of the iteration. Obtaining the least-squares approximate solution to an over-determined system of linear equations can be viewed both as an optimization problem and as solving the related system of normal equations. Finding the solution of a system of under-determined linear equations that is closest to a given vector is another optimization problem that can be solved using the methods we shall discuss. Maximizing entropy, subject to linear equality constraints, and maximizing likelihood for estimating the parameters of multivariate Poisson distributions also can be formulated as finding exact or approximate solutions of systems of linear equations.

We begin our detailed discussion of algorithms with a simple problem, solving a general system of linear equations, and a simple method, the algebraic reconstruction technique (ART). We shall permit complex entries for the matrix and vectors involved.

11.1 Background

The ART was introduced by Gordon, Bender and Herman [94] as a method for image reconstruction in transmission tomography. It was noticed somewhat later that the ART is a special case of Kaczmarz's algorithm [104]. For $i = 1, ..., I$, let L_i be the set of pixel indices j for which the j-th pixel intersects the i-th line segment, and let $|L_i|$ be the cardinality of the set L_i. Let $A_{ij} = 1$ for j in L_i, and $A_{ij} = 0$ otherwise. With $i = k(\mathrm{mod}\, I) + 1$, the iterative step of the ART algorithm is

$$x_j^{k+1} = x_j^k + \frac{1}{|L_i|}(b_i - (Ax^k)_i), \tag{11.1}$$

for j in L_i, and

$$x_j^{k+1} = x_j^k, \tag{11.2}$$

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

A somewhat more sophisticated version of ART allows A_{ij} to include the length of the i-th line segment that lies within the j-th pixel; A_{ij} is taken to be the ratio of this length to the length of the diagonal of the j-pixel.

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

11.2 The ART

Let A be a complex matrix with I rows and J columns, and let b be a member of C^I. We want to solve the system $Ax = b$.

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

$$H_i = \{x | (Ax)_i = b_i\}, \tag{11.3}$$

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

$$x^{k+1} = P_{i(k)}x^k. \tag{11.4}$$

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

11.2.1 Calculating the ART

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

$$x_j = z_j + \overline{A_{ij}}(b_i - (Az)_i)/\sum_{m=1}^{J} |A_{im}|^2. \tag{11.5}$$

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

$$\sum_{j=1}^{J} |A_{ij}|^2 = 1, \tag{11.6}$$

for each $i = 1, ..., I$, and that the entries of b have been rescaled accordingly, to preserve the equations $Ax = b$.

The ART is then the following: begin with an arbitrary vector x^0; for each nonnegative integer k, having found x^k, the next iterate x^{k+1} has entries

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

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

11.2.2 Full-cycle ART

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

$$T = P_I P_{I-1} \cdots P_2 P_1. \tag{11.8}$$

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

11.2.3 Relaxed ART

The ART employs orthogonal projections onto the individual hyperplanes. If we permit the next iterate to fall short of the hyperplane, or somewhat beyond it, we get a relaxed version of ART.The relaxed ART algorithm is as follows:

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

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

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

11.2.4 Constrained ART

Let C be a closed, nonempty convex subset of C^J and $P_C x$ the orthogonal projection of x onto C. If there are solutions of $Ax = b$ that lie within C, we can find them using the constrained ART algorithm:

Algorithm 11.2 (Constrained ART) *Let x^0 be arbitrary. For $k = 0, 1, ...$ and $i = k(\mathrm{mod}\, I) + 1$, let*

$$x_j^{k+1} = P_C(x_j^k + \overline{A_{ij}}(b_i - (Ax^k)_i)). \tag{11.10}$$

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

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

$$x_j^{k+1} = (x_j^k + A_{ij}(b_i - (Ax^k)_i))_+, \tag{11.11}$$

where, for any real number a, $a_+ = \max\{a, 0\}$.

The constrained ART converges to a solution of $Ax = b$ within C, whenever such solutions exist.

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

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

Theorem 11.1 *Let A have the full-rank property. Suppose there is no nonnegative solution to the system of equations $Ax = b$. Then there is a subset S of the set $\{j = 1, 2, ..., J\}$, with cardinality at most $I - 1$, such that, if $\hat{x}$ is any minimizer of $||Ax - b||_2$ subject to $x \geq 0$, then $\hat{x}_j = 0$ for j not in S. Therefore, $\hat{x}$ is unique.*

Proof: According to the gradient form of the Karush-Kuhn-Tucker Theorem 8.5, the vector $A\hat{x}$ must satisfy the condition

$$\sum_{i=1}^{I} A_{ij}((A\hat{x})_i - b_i) = 0 \tag{11.12}$$

for all j for which $\hat{x}_j > 0$ for some nonnegative solution $\hat{x}$. Let S be the set of all indices j for which there exists a nonnegative solution $\hat{x}$ with $\hat{x}_j > 0$. Then Equation (11.12) must hold for all j in S. Let Q be the matrix obtained from A by deleting those columns whose index j is not in S. Then $Q^T(A\hat{x} - b) = 0$. If Q has full rank and the cardinality of S is greater than or equal to I, then Q^T is one-to-one and $A\hat{x} = b$. ∎

11.3 Convergence Results for ART

How the ART behaves depends on whether or not the system $Ax = b$ has solutions.

11.3.1 When $Ax = b$ Has Solutions

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

Theorem 11.2 *Let $A\hat{x} = b$ and let x^0 be arbitrary. Let $\{x^k\}$ be generated by Equation (11.7). Then the sequence $\{||\hat{x} - x^k||_2\}$ is decreasing and $\{x^k\}$ converges to the solution of $Ax = b$ closest to x^0.*

11.3.2 When $Ax = b$ Has No Solutions

When there are no exact solutions, the ART does not converge to a single vector, but, for each fixed i, the subsequence $\{x^{nI+i}, n = 0, 1, ...\}$ converges to a vector z^i and the collection $\{z^i | i = 1, ..., I\}$ is called the *limit cycle*. This was shown by Tanabe [143] and also follows from the results of De Pierro and Iusem [70]. For simplicity, we assume that $I > J$, and that the matrix A has full rank, which implies that $Ax = 0$ if and only if $x = 0$. Because the operator $T = P_I P_{i-1} \cdots P_2 P_1$ is av, this subsequential convergence to a limit cycle will follow from the KM Theorem 10.2, once we

have established that T has fixed points. A different proof of subsequential convergence is given in [44].

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

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

11.4 The Geometric Least-Squares Solution

When the system $Ax = b$ has no solutions, it is reasonable to seek an approximate solution, such as the *least squares* solution, $x_{LS} = (A^\dagger A)^{-1}A^\dagger b$, which minimizes $||Ax-b||_2$. It is important to note that the system $Ax = b$ has solutions if and only if the related system $WAx = Wb$ has solutions, where W denotes an invertible matrix; when solutions of $Ax = b$ exist, they are identical to those of $WAx = Wb$. But, when $Ax = b$ does not have solutions, the least-squares solutions of $Ax = b$, which need not be unique, but usually are, and the least-squares solutions of $WAx = Wb$ need not be identical. In the typical case in which $A^\dagger A$ is invertible, the unique least-squares solution of $Ax = b$ is

$$(A^\dagger A)^{-1}A^\dagger b, \tag{11.13}$$

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

$$(A^\dagger W^\dagger WA)^{-1}A^\dagger W^\dagger b, \tag{11.14}$$

and these need not be the same.

A simple example is the following. Consider the system

$$x = 1$$

$$x = 2, \tag{11.15}$$

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

$$2x = 2$$

$$x = 2, \tag{11.16}$$

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

Definition 11.2 *The geometric least-squares solution of $Ax = b$ is the least-squares solution of $WAx = Wb$, for W the diagonal matrix whose entries are the reciprocals of the Euclidean lengths of the rows of A.*

In our example above, the geometric least-squares solution for the first system is found by using $W_{11} = 1 = W_{22}$, so is again $x = 1.5$, while the geometric least-squares solution of the second system is found by using $W_{11} = 0.5$ and $W_{22} = 1$, so that the geometric least-squares solution is $x = 1.5$, not $x = 1.2$.

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

There is a partial answer to the second question. In [35] (see also [44]) it was shown that if the system $Ax = b$ has no exact solution, and if $I = J+1$, then the vectors of the limit cycle lie on a sphere in J-dimensional space having the least-squares solution at its center. This is not true more generally, however.

11.5 Regularized ART

If the entries of b are noisy but the system $Ax = b$ remains consistent (which can easily happen in the under-determined case, with $J > I$), the ART begun at $x^0 = 0$ converges to the solution having minimum Euclidean norm, but this norm can be quite large. The resulting solution is probably useless. Instead of solving $Ax = b$, we *regularize* by minimizing, for example, the function

$$F_\epsilon(x) = ||Ax - b||_2^2 + \epsilon^2 ||x||_2^2. \tag{11.17}$$

The solution to this problem is the vector

$$\hat{x}_\epsilon = (A^\dagger A + \epsilon^2 I)^{-1} A^\dagger b. \tag{11.18}$$

However, we do not want to calculate $A^\dagger A + \epsilon^2 I$ when the matrix A is large. Fortunately, there are ways to find $\hat{x}_\epsilon$, using only the matrix A and the ART algorithm.

We discuss two methods for using ART to obtain regularized solutions of $Ax = b$. The first one is presented in [44], while the second one is due to Eggermont, Herman, and Lent [80].

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

$$\begin{bmatrix} A^\dagger & \epsilon I \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = 0. \tag{11.19}$$

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

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

$$[\, A \quad \epsilon I \,] \begin{bmatrix} x \\ v \end{bmatrix} = b. \tag{11.20}$$

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 the lower component v^∞ satisfies $\epsilon v^\infty = b - A\hat{x}_\epsilon$. We leave to the reader the proofs that these two algorithms perform as we claim.

11.6 Avoiding the Limit Cycle

Generally, the greater the minimum value of $||Ax-b||_2^2$ the more the vectors of the LC are distinct from one another. There are several ways to avoid the LC in ART and to obtain a least-squares solution. One way is the *double ART* (DART) [38]:

11.6.1 Double ART (DART)

We know that any b can be written as $b = A\hat{x} + \hat{w}$, where $A^\dagger \hat{w} = 0$ and $\hat{x}$ is a minimizer of $||Ax - b||_2^2$. The vector $\hat{w}$ is the orthogonal projection of b onto the null space of the matrix transformation $A^\dagger$. Therefore, in Step 1 of DART we apply the ART algorithm to the consistent system of linear equations $A^\dagger w = 0$, beginning with $w^0 = b$. The limit is $w^\infty = \hat{w}$, the member of the null space of $A^\dagger$ closest to b. In Step 2, apply ART to the consistent system of linear equations $Ax = b - w^\infty = A\hat{x}$. The limit is then the minimizer of $||Ax - b||_2$ closest to x^0. Notice that we could also obtain the least-squares solution by applying ART to the system $A^\dagger y = A^\dagger b$, starting with $y^0 = 0$, to obtain the minimum-norm solution, which is $y = A\hat{x}$, and then applying ART to the system $Ax = y$.

11.6.2 Strongly Underrelaxed ART

Another method for avoiding the LC is *strong under-relaxation*, due to Censor, Eggermont and Gordon [54]. Let $t > 0$. Replace the iterative step in ART with

$$x_j^{k+1} = x_j^k + t\overline{A_{ij}}(b_i - (Ax^k)_i). \tag{11.21}$$

In [54] it is shown that, as $t \to 0$, the vectors of the LC approach the geometric least squares solution closest to x^0; a short proof is in [35]. Bertsekas [13] uses strong under-relaxation to obtain convergence of more general incremental methods.

11.7 Exercises

11.1 *Consider the system of two equations in two unknowns*

$$\begin{cases} mx - y = 0; \\ \quad y = 0. \end{cases}$$

Without using a computer or calculator, investigate how the speed of convergence of the ART depends on the value of m.

Chapter 12

Partial Gradient Methods

The partial gradient method reduces the amount of calculation required at each step, compared to a gradient descent algorithm, and, in many cases, accelerates the convergence.

12.1 Decomposing the Objective Function

Since the gradient $\nabla f(x)$ is the direction of greatest increase of the function $f(x)$, it is natural for the iterative step of minimization algorithms to involve the negative of the gradient at the current x^k. The *partial gradient* approach applies to problems having as their goal the minimization of a non-negative function $f(x)$ that has the form

$$f(x) = \sum_{i=1}^{I} f_i(x), \tag{12.1}$$

where each $f_i(x)$ is a non-negative function of the variable x. The gradient of such functions then has the form

$$\nabla f(x) = \sum_{i=1}^{I} \nabla f_i(x). \tag{12.2}$$

For any subset B of the set $\{i = 1, ..., I\}$, the partial gradient associated with the set B is

$$\nabla^B f(x) = \sum_{i \in B} \nabla f_i(x). \tag{12.3}$$

Partial gradient methods, also called *incremental gradient methods* [13], are iterative algorithms in which the gradient of f is replaced by a partial gradient at each step .

12.2 A Partial Gradient Algorithm

Suppose that the set $\{i = 1, ..., I\}$ is partitioned into N disjoint subsets, B_n, $n = 1, ..., N$; the B_n are often called *blocks*. Let $I_n > 0$ be the number of members of B_n. Let

$$f^n(x) = \sum_{i \in B_n} f_i(x), \tag{12.4}$$

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

$$\nabla f^n(x) = \sum_{i \in B_n} \nabla f_i(x). \tag{12.5}$$

Let x^k be the current vector in the iteration, and let $n = n(k) = k(\mod N)+1$. Partial gradient methods will employ only $\nabla^n f(x^k)$ to calculate x^{k+1} from x^k. Such methods are also called *block-iterative* methods. The iterative step of the *partial gradient algorithm* (PGA) is

$$x^{k+1} = x^k - \gamma_n \nabla f^n(x^k). \tag{12.6}$$

We shall see other partial gradient algorithms later, when we consider entropy-based methods and positivity constraints. Now, we consider the convergence of the PGA, in light of the KM Theorem 10.2.

12.3 Convergence of the PGA

Suppose, for the sake of illustration, that each gradient $\nabla f^n(x)$ is $\frac{L}{N}$-Lipschitz, so that $\nabla f(x)$ is L-Lipschitz. Then $\gamma_n \nabla f^n(x)$ is $\frac{N}{\gamma_n L}$-ism. Therefore, for $0 < \gamma_n < \frac{2N}{L}$, the operator

$$T_n = I - \gamma_n \nabla f^n$$

is α_n-ism, where $\alpha_n = \frac{\gamma_n L}{2N}$. The full gradient algorithm we use for comparison has the iterative step

$$x^{k+1} = x^k - \gamma \nabla f(x^k). \tag{12.7}$$

Since $\nabla f(x)$ is L-Lipschitz, we select $\gamma = \frac{1}{L}$. The operator $T = I - \gamma \nabla f$ is then $\frac{1}{2}$-av.

Suppose, in addition, that the minimum value of $f(x)$ is zero, and that this minimum is attained at $x = z$. Then, $x = z$ minimizes each of the $f^n(x)$, as well. From the proof of the KM Theorem 10.2, we have

$$||z - x^k||^2 - ||z - x^{k+1}||^2 \geq (\frac{1}{\alpha_n} - 1)||x^k - x^{k+1}||^2.$$

Let us choose $\gamma_n = \frac{N}{L}$. Then we have

$$||z - x^k||^2 - ||z - x^{k+1}||^2 \geq ||x^k - x^{k+1}||^2.$$

Speaking loosely, we can say that the size of the difference $x^k - x^{k+1}$ is on the order of the difference we would have for the full gradient descent algorithm, with $\gamma = \frac{1}{L}$. Therefore, the squared distance to z, after one complete pass through all the blocks, given by

$$||z - x^0||^2 - ||z - x^N||^2,$$

is roughly N times that after one full gradient step. The calculations are roughly the same for one step of the full gradient method and one pass, through all the blocks, of the PGA, but the improvement made in reaching z can be significantly greater for the PGA. This has been the experience with the partial gradient versions of the EMML and SMART algorithms.

In order to achieve this acceleration, it is important that consecutive functions $f^n(x)$ and $f^{n+1}(x)$ not be too similar. If they are, then their gradients are similar and the consecutive iterative steps can be more or less in the same direction. The desired decrease in the objective function $f(x)$ may not be as large as it would otherwise be.

We assumed that the minimum value of $f(x)$ is $f(z) = 0$, which made z a minimizer of each of the $f^n(x)$ individually. If the minimum value of $f(x)$ is not zero, the point $x = z$ need not be a minimizer of the individual $f^n(x)$ and the operators $T_n = I - \gamma_n \nabla f^n$ need not have a common fixed point. In such cases, we expect to see subsequential convergence to a limit cycle, as with the ART algorithm.

12.4 The Example of the ART

Finding a least-squares solution of the real system $Ax = b$ means minimizing the function

$$f(x) = \frac{1}{2}||Ax - b||_2^2 = \frac{1}{2}\sum_{i=1}^{I}\Big((Ax)_i - b_i\Big)^2, \tag{12.8}$$

having the form of Equation (12.1), with

$$f_i(x) = \frac{1}{2}\Big((Ax)_i - b_i\Big)^2, \tag{12.9}$$

for each $i = 1, ..., I$. We assume that the rows of A have been rescaled to have length one, in which case Cimmino's algorithm for solving the system $Ax = b$ can be shown to have the iterative step

$$x^{k+1} = x^k - \frac{1}{I}\nabla f(x^k). \tag{12.10}$$

Let's consider partial gradient versions of Cimmino's algorithm.

The iterative step of the partial gradient version of Cimmino's algorithm is

$$x^{k+1} = x^k - \frac{1}{I_n}\nabla^n f(x^k). \tag{12.11}$$

Now consider the case in which each B_n contains a single member.

With $N = I$ and $B_n = \{n\}$, for $n = 1, ..., I$, the partial gradient has the entries

$$\Big(\nabla^n f(x)\Big)_j = \Big(\nabla f_n(x)\Big)_j = A_{n,j}\Big((Ax)_n - b_n\Big). \tag{12.12}$$

The partial gradient algorithm becomes

$$x_j^{k+1} = x_j^k + A_{nj}\Big(b_n - (Ax^k)_n\Big). \tag{12.13}$$

This is the iterative step of the ART. When the system $Ax = b$ has solutions, the ART can converge much faster than the Cimmino algorithm. It is important to note, however, that to achieve this accelerated convergence, it is necessary to avoid an ordering of the equations in which equations $n(k)$ and $n(k+1)$ are similar [99]. A random ordering of the equations is usually reasonable. A small amount of relaxation may also improve the speed of convergence [139].

Chapter 13

Block-Iterative ART

13.1 Introduction and Notation

The ART is a sequential algorithm, using only a single equation from the system $Ax = b$ at each step of the iteration. In this chapter we consider iterative procedures for solving $Ax = b$ in which several or all of the equations are used at each step. Such methods are called *block-iterative* and *simultaneous* algorithms, respectively.

We are concerned here with iterative methods for solving, at least approximately, the system of I linear equations in J unknowns symbolized by $Ax = b$. In the applications of interest to us, such as medical imaging, both I and J are quite large, making the use of iterative methods the only feasible approach. It is also typical of such applications that the matrix A is sparse, that is, has relatively few non-zero entries. Therefore, iterative methods that exploit this sparseness to accelerate convergence are of special interest to us.

The *algebraic reconstruction technique* (ART) of Gordon, et al. [94] is a *sequential* method; at each step only one equation is used. The current vector x^{k-1} is projected orthogonally onto the hyperplane corresponding to that single equation, to obtain the next iterate x^k. The iterative step of the ART is

$$x_j^k = x_j^{k-1} + A_{ij}\left(\frac{b_i - (Ax^{k-1})_i}{\sum_{t=1}^{J} |A_{it}|^2}\right), \tag{13.1}$$

where $i = k(\mathrm{mod}\, I)$. The sequence $\{x^k\}$ converges to the solution closest to x^0 in the consistent case, but only converges subsequentially to a limit cycle in the inconsistent case.

Cimmino's method [64] is a *simultaneous* method, in which all the equations are used at each step. The current vector x^{k-1} is projected orthog-

onally onto each of the hyperplanes and these projections are averaged to obtain the next iterate x^k. The iterative step of Cimmino's method is

$$x_j^k = \frac{1}{I}\sum_{i=1}^{I}\left(x_j^{k-1} + A_{ij}\left(\frac{b_i - (Ax^{k-1})_i}{\sum_{t=1}^{J}|A_{it}|^2}\right)\right),$$

which can also be written as

$$x_j^k = x_j^{k-1} + \sum_{i=1}^{I} A_{ij}\left(\frac{b_i - (Ax^{k-1})_i}{I\sum_{t=1}^{J}|A_{it}|^2}\right). \tag{13.2}$$

Landweber's iterative scheme [110] with

$$x^k = x^{k-1} + B^\dagger(d - Bx^{k-1}), \tag{13.3}$$

converges to the least-squares solution of $Bx = d$ closest to x^0, provided that the largest singular value of B does not exceed one. If we let B be the matrix with entries

$$B_{ij} = A_{ij}/\sqrt{I\sum_{t=1}^{J}|A_{it}|^2},$$

and define

$$d_i = b_i/\sqrt{I\sum_{t=1}^{J}|A_{it}|^2},$$

then, since the trace of the matrix $BB^\dagger$ is one, convergence of Cimmino's method follows. However, using the trace in this way to estimate the largest singular value of a matrix usually results in an estimate that is far too large, particularly when A is large and sparse, and therefore in an iterative algorithm with unnecessarily small step sizes.

The appearance of the term

$$I\sum_{t=1}^{J}|A_{it}|^2$$

in the denominator of Cimmino's method suggested to Censor et al. [58] that, when A is sparse, this denominator might be replaced with

$$\sum_{t=1}^{J} s_t|A_{it}|^2,$$

where s_t denotes the number of non-zero entries in the tth column of A. The resulting iterative method is the *component-averaging* (CAV) iteration. Convergence of the CAV method was established by showing that no

singular value of the matrix B exceeds one, where B has the entries

$$B_{ij} = A_{ij} / \sqrt{\sum_{t=1}^{J} s_t |A_{it}|^2}.$$

In [48] we extended this result, to show that no eigenvalue of $A^\dagger A$ exceeds the maximum of the numbers

$$p_i = \sum_{t=1}^{J} s_t |A_{it}|^2.$$

Convergence of CAV then follows, as does convergence of several other methods, including the ART, Landweber's method, the SART [2], the block-iterative CAV (BICAV) [59], the CARP1 method of Gordon and Gordon [95], a block-iterative variant of CARP1 obtained from the DROP method of Censor et al. [56], and the SIRT method [147].

For a positive integer N with $1 \leq N \leq I$, we let $B_1, ..., B_N$ be not necessarily disjoint subsets of the set $\{i = 1, ..., I\}$; the subsets B_n are called *blocks*. We then let A_n be the matrix and b^n the vector obtained from A and b, respectively, by removing all the rows except for those whose index i is in the set B_n. For each n, we let s_{nt} be the number of non-zero entries in the tth column of the matrix A_n, s_n the maximum of the s_{nt}, s the maximum of the s_t, and $L_n = \rho(A_n^\dagger A_n)$ be the spectral radius, or largest eigenvalue, of the matrix $A_n^\dagger A_n$, with $L = \rho(A^\dagger A)$. We denote by A_i the ith row of the matrix A, and by ν_i the length of A_i, so that

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

13.2 Cimmino's Algorithm

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

$$x^k = \frac{1}{I} \sum_{i=1}^{I} P_i x^{k-1}, \tag{13.4}$$

where P_i is the orthogonal projection onto $H(a^i, b_i)$. The iterative step can then be written as

$$x_j^k = x_j^{k-1} + \frac{1}{I}\sum_{i=1}^{I}\left(\frac{A_{ij}(b_i - (Ax^{k-1})_i)}{\nu_i^2}\right). \tag{13.5}$$

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

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

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

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

13.3 The Landweber Algorithms

For simplicity, we assume, in this section, that $\nu_i = 1$ for all i. The Landweber algorithm [110, 12], with the iterative step

$$x^k = x^{k-1} + \gamma A^\dagger(b - Ax^{k-1}), \tag{13.6}$$

converges to the least squares solution closest to the starting vector x^0, provided that $0 < \gamma < 2/\lambda_{max}$, where λ_{max} is the largest eigenvalue of the nonnegative-definite matrix $A^\dagger A$. Loosely speaking, the larger γ is, the faster the convergence. However, precisely because A is large, calculating the matrix $A^\dagger A$, not to mention finding its largest eigenvalue, can be prohibitively expensive. The matrix A is said to be sparse if most of its entries are zero. Useful upper bounds for λ_{max} are then given by Theorems 13.2 and 13.3.

13.3.1 Finding the Optimum γ

The operator

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

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

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

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

Lemma 13.1 *If $\gamma < 0$, then none of the eigenvalues of B is less than one.*

Lemma 13.2 *For*

$$0 \leq \gamma \leq \frac{2}{\lambda_{max} + \lambda_{min}}, \tag{13.7}$$

we have

$$\rho(B) = 1 - \gamma\lambda_{min}; \tag{13.8}$$

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

$$\gamma = \frac{2}{\lambda_{max} + \lambda_{min}}, \tag{13.9}$$

and equals

$$\frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}}. \tag{13.10}$$

Similarly, for

$$\gamma \geq \frac{2}{\lambda_{max} + \lambda_{min}}, \tag{13.11}$$

we have

$$\rho(B) = \gamma\lambda_{max} - 1; \tag{13.12}$$

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

$$\gamma = \frac{2}{\lambda_{max} + \lambda_{min}}, \tag{13.13}$$

and equals

$$\frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}}. \tag{13.14}$$

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

$$\gamma = \frac{2}{\lambda_{max} + \lambda_{min}}, \tag{13.15}$$

in which case we have

$$\|B\|_2 = \frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}} = \frac{c-1}{c+1}, \tag{13.16}$$

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

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

$$P_0 = I - aa^\dagger. \tag{13.17}$$

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

$$B = I - \gamma aa^\dagger \tag{13.18}$$

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

13.3.2 The Projected Landweber Algorithm

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

$$x^{k+1} = (x^k + \gamma A^\dagger(b - Ax^k))_+, \tag{13.19}$$

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

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

$$x^{k+1} = P_C(x^k + \gamma A^\dagger(b - Ax^k)). \tag{13.20}$$

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

Both the Landweber and projected Landweber algorithms are special cases of the CQ algorithm [41], which, in turn, is a special case of the more general iterative fixed point algorithm, the Krasnoselskii/Mann (KM) method, with convergence governed by the KM Theorem 10.2.

13.4 Some Upper Bounds for L

For the iterative algorithms we shall consider here, having a good upper bound for the largest eigenvalue of the matrix $A^\dagger A$ is important. In the applications of interest, principally medical image processing, the matrix A is large; even calculating $A^\dagger A$, not to mention computing eigenvalues, is prohibitively expensive. In addition, the matrix A is typically sparse, but $A^\dagger A$ will not be, in general. In this section we present upper bounds for L that are particularly useful when A is sparse and do not require the calculation of $A^\dagger A$.

13.4.1 Our Basic Eigenvalue Inequality

In [147] van der Sluis and van der Vorst show that certain rescaling of the matrix A results in none of the eigenvalues of $A^\dagger A$ exceeding one. A modification of their proof leads to upper bounds on the eigenvalues of the original $A^\dagger A$ ([48]). For any a in the interval $[0, 2]$ let

$$c_{aj} = c_{aj}(A) = \sum_{i=1}^{I} |A_{ij}|^a,$$

$$r_{ai} = r_{ai}(A) = \sum_{j=1}^{J} |A_{ij}|^{2-a},$$

and c_a and r_a the maxima of the c_{aj} and r_{ai}, respectively. We prove the following theorem.

Theorem 13.1 *For any a in the interval $[0, 2]$, no eigenvalue of the matrix $A^\dagger A$ exceeds the maximum of*

$$\sum_{j=1}^{J} c_{aj}|A_{ij}|^{2-a},$$

over all i, nor the maximum of

$$\sum_{i=1}^{I} r_{ai}|A_{ij}|^a,$$

over all j. Therefore, no eigenvalue of $A^\dagger A$ exceeds $c_a r_a$.

Proof: Let $A^\dagger Av = \lambda v$, and let $w = Av$. Then we have

$$\|A^\dagger w\|^2 = \lambda \|w\|^2.$$

Applying Cauchy's Inequality, we obtain

$$\Big|\sum_{i=1}^{I}\overline{A_{ij}}w_i\Big|^2 \leq \Big(\sum_{i=1}^{I}|A_{ij}|^{a/2}|A_{ij}|^{1-a/2}|w_i|\Big)^2$$

$$\leq \Big(\sum_{i=1}^{I}|A_{ij}|^{a}\Big)\Big(\sum_{i=1}^{I}|A_{ij}|^{2-a}|w_i|^2\Big).$$

Therefore,

$$\|A^\dagger w\|^2 \leq \sum_{j=1}^{J}\Big(c_{aj}(\sum_{i=1}^{I}|A_{ij}|^{2-a}|w_i|^2)\Big) = \sum_{i=1}^{I}\Big(\sum_{j=1}^{J}c_{aj}|A_{ij}|^{2-a}\Big)|w_i|^2$$

$$\leq \max_i\Big(\sum_{j=1}^{J}c_{aj}|A_{ij}|^{2-a}\Big)\|w\|^2.$$

The remaining two assertions follow in similar fashion. ∎

As a corollary, we obtain the following eigenvalue inequality, which is central to our discussion.

Theorem 13.2 *For each $i = 1, 2, ..., I$, let*

$$p_i = \sum_{j=1}^{J}s_j|A_{ij}|^2,$$

and let p be the maximum of the p_i. Then $L \leq p$.

Proof: Take $a = 0$. Then, using the convention that $0^0 = 0$, we have $c_{0j} = s_j$. ∎

Corollary 13.1 *Selecting $a = 1$, we have*

$$L = \|A\|_2^2 \leq \|A\|_1\|A\|_\infty = c_1r_1.$$

Corollary 13.2 *Selecting $a = 2$, we have*

$$L = \|A\|_2^2 \leq \|A\|_F^2,$$

where $\|A\|_F$ denotes the Frobenius norm of A.

Corollary 13.3 *Let G be the matrix with entries*

$$G_{ij} = A_{ij}\sqrt{\alpha_i}\sqrt{\beta_j},$$

where

$$\alpha_i \leq \Big(\sum_{j=1}^{J}s_j\beta_j|A_{ij}|^2\Big)^{-1},$$

for all i. Then $\rho(G^\dagger G) \leq 1$.

Proof: We have

$$\sum_{j=1}^{J} s_j |G_{ij}|^2 = \alpha_i \sum_{j=1}^{J} s_j \beta_j |A_{ij}|^2 \leq 1,$$

for all i. The result follows from Corollary 13.2. ∎

Corollary 13.4 *If* $\sum_{j=1}^{J} s_j |A_{ij}|^2 \leq 1$ *for all* i*, then* $L \leq 1$*.*

Corollary 13.5 *If* $0 < \gamma_i \leq p_i^{-1}$ *for all* i*, then the matrix* B *with entries* $B_{ij} = \sqrt{\gamma_i} A_{ij}$ *has* $\rho(B^\dagger B) \leq 1$*.*

Proof: We have

$$\sum_{j=1}^{J} s_j |B_{ij}|^2 = \gamma_i \sum_{j=1}^{J} s_j |A_{ij}|^2 = \gamma_i p_i \leq 1.$$

Therefore, $\rho(B^\dagger B) \leq 1$, according to the theorem. ∎

Corollary 13.6 *([41]; [146], Th. 4.2) If* $\sum_{j=1}^{J} |A_{ij}|^2 = 1$ *for each* i*, then* $L \leq s$*.*

Proof: For all i we have

$$p_i = \sum_{j=1}^{J} s_j |A_{ij}|^2 \leq s \sum_{j=1}^{J} |A_{ij}|^2 = s.$$

Therefore,

$$L \leq p \leq s.$$

∎

Corollary 13.7 *If, for some* a *in the interval* $[0, 2]$*, we have*

$$\alpha_i \leq r_{ai}^{-1}, \tag{13.21}$$

for each i*, and*

$$\beta_j \leq c_{aj}^{-1}, \tag{13.22}$$

for each j*, then, for the matrix* G *with entries*

$$G_{ij} = A_{ij} \sqrt{\alpha_i} \sqrt{\beta_j},$$

no eigenvalue of $G^\dagger G$ *exceeds one.*

Proof: We calculate $c_{aj}(G)$ and $r_{ai}(G)$ and find that

$$c_{aj}(G) \leq \Big(\max_i \alpha_i^{a/2}\Big)\beta_j^{a/2}\sum_{i=1}^{I}|A_{ij}|^a = \Big(\max_i \alpha_i^{a/2}\Big)\beta_j^{a/2}c_{aj}(A),$$

and

$$r_{ai}(G) \leq \Big(\max_j \beta_j^{1-a/2}\Big)\alpha_i^{1-a/2}r_{ai}(A).$$

Therefore, applying the inequalities (13.21) and (13.22), we have

$$c_{aj}(G)r_{ai}(G) \leq 1,$$

for all i and j. Consequently, $\rho(G^\dagger G) \leq 1$. ∎

13.4.2 Another Upper Bound for L

The next theorem ([41]) provides another upper bound for L that is useful when A is sparse. As previously, for each i and j, we let $e_{ij} = 1$, if A_{ij} is not zero, and $e_{ij} = 0$, if $A_{ij} = 0$. Let $0 < \nu_i = \sqrt{\sum_{j=1}^{J}|A_{ij}|^2}$, $\sigma_j = \sum_{i=1}^{I} e_{ij}\nu_i^2$, and σ be the maximum of the σ_j.

Theorem 13.3 *([41]) No eigenvalue of $A^\dagger A$ exceeds σ.*

Proof: Let $A^\dagger Av = cv$, for some non-zero vector v and scalar c. With $w = Av$, we have

$$w^\dagger AA^\dagger w = cw^\dagger w.$$

Then

$$\Big|\sum_{i=1}^{I}\overline{A_{ij}}w_i\Big|^2 = \Big|\sum_{i=1}^{I}\overline{A_{ij}}e_{ij}\nu_i\frac{w_i}{\nu_i}\Big|^2 \leq \Big(\sum_{i=1}^{I}|A_{ij}|^2\frac{|w_i|^2}{\nu_i^2}\Big)\Big(\sum_{i=1}^{I}\nu_i^2 e_{ij}\Big)$$

$$= \Big(\sum_{i=1}^{I}|A_{ij}|^2\frac{|w_i|^2}{\nu_i^2}\Big)\sigma_j \leq \sigma\Big(\sum_{i=1}^{I}|A_{ij}|^2\frac{|w_i|^2}{\nu_i^2}\Big).$$

Therefore, we have

$$cw^\dagger w = w^\dagger AA^\dagger w = \sum_{j=1}^{J}\Big|\sum_{i=1}^{I}\overline{A_{ij}}w_i\Big|^2$$

$$\leq \sigma\sum_{j=1}^{J}\Big(\sum_{i=1}^{I}|A_{ij}|^2\frac{|w_i|^2}{\nu_i^2}\Big) = \sigma\sum_{i=1}^{I}|w_i|^2 = \sigma w^\dagger w.$$

We conclude that $c \leq \sigma$. ∎

Corollary 13.8 *Let the rows of A have Euclidean length one. Then no eigenvalue of $A^\dagger A$ exceeds the maximum number of non-zero entries in any column of A.*

Proof: We have $\nu_i^2 = \sum_{j=1}^J |A_{ij}|^2 = 1$, for each i, so that $\sigma_j = s_j$ is the number of non-zero entries in the jth column of A, and $\sigma = s$ is the maximum of the σ_j. ∎

When the rows of A have length one, it is easy to see that $L \leq I$, so the choice of $\gamma = \frac{1}{I}$ in the Landweber algorithm, which gives Cimmino's algorithm [64], is acceptable, although perhaps much too small.

The proof of Theorem 13.3 is based on results presented by Arnold Lent in informal discussions with Gabor Herman, Yair Censor, Rob Lewitt and me at MIPG in Philadelphia in the late 1990's.

13.5 The Basic Convergence Theorem

The following theorem is a basic convergence result concerning block-iterative ART algorithms.

Theorem 13.4 *Let $L_n \leq 1$, for $n = 1, 2, ..., N$. If the system $Ax = b$ is consistent, then, for any starting vector x^0, and with $n = n(k) = k(\mathrm{mod}\, N)$ and $\lambda_k \in [\epsilon, 2 - \epsilon]$ for all k, the sequence $\{x^k\}$ with iterative step*

$$x^k = x^{k-1} + \lambda_k A_n^\dagger (b^n - A_n x^{k-1}) \tag{13.23}$$

converges to the solution of $Ax = b$ for which $\|x - x^0\|$ is minimized.

We begin with the following lemma.

Lemma 13.3 *Let T be any (not necessarily linear) operator on R^J, and $S = I - T$, where I denotes the identity operator. Then, for any x and y, we have*

$$\|x - y\|^2 - \|Tx - Ty\|^2 = 2\langle Sx - Sy, x - y\rangle - \|Sx - Sy\|^2. \tag{13.24}$$

The proof is a simple calculation and we omit it here.

Proof of Theorem 13.4: Let $Az = b$. Applying Equation (13.24) to the operator

$$Tx = x + \lambda_k A_n^\dagger (b^n - A_n x),$$

we obtain

$$\|z - x^{k-1}\|^2 - \|z - x^k\|^2 = 2\lambda_k \|b^n - A_n x^{k-1}\|^2 - \lambda_k^2 \|A_n^\dagger b^n - A_n^\dagger A_n x^{k-1}\|^2. \tag{13.25}$$

Since $L_n \leq 1$, it follows that

$$\|A_n^\dagger b^n - A_n^\dagger A_n x^{k-1}\|^2 \leq \|b^n - A_n x^{k-1}\|^2.$$

Therefore,

$$\|z - x^{k-1}\|^2 - \|z - x^k\|^2 \geq (2\lambda_k - \lambda_k^2)\|b^n - A_n x^{k-1}\|^2,$$

from which we draw several conclusions:

- the sequence $\{\|z - x^k\|\}$ is decreasing;
- the sequence $\{\|b^n - A_n x^{k-1}\|\}$ converges to zero.

In addition, for fixed $n = 1, ..., N$ and $m \to \infty$,

- the sequence $\{\|b^n - A_n x^{mN+n-1}\|\}$ converges to zero;
- the sequence $\{x^{mN+n}\}$ is bounded.

Let $x^{*,1}$ be a cluster point of the sequence $\{x^{mN+1}\}$; then there is subsequence $\{x^{m_r N+1}\}$ converging to $x^{*,1}$. The sequence $\{x^{m_r N+2}\}$ is also bounded, and we select a cluster point $x^{*,2}$. Continuing in this fashion, we obtain cluster points $x^{*,n}$, for $n = 1, ..., N$. From the conclusions reached previously, we can show that $x^{*,n} = x^{*,n+1} = x^*$, for $n = 1, 2, ..., N-1$, and $Ax^* = b$. Replacing the generic solution $\hat{x}$ with the solution x^*, we see that the sequence $\{\|x^* - x^k\|\}$ is decreasing. But, subsequences of this sequence converge to zero, so the entire sequence converges to zero, and so $x^k \to x^*$.

Now we show that x^* is the solution of $Ax = b$ that minimizes $\|x - x^0\|$. Since $x^k - x^{k-1}$ is in the range of $A^\dagger$ for all k, so is $x^* - x^0$, from which it follows that x^* is the solution minimizing $\|x - x^0\|$. Another way to get this result is to use Equation (13.25). Since the right side of Equation (13.25) is independent of the choice of solution, so is the left side. Summing both sides over the index k reveals that the difference

$$\|x - x^0\|^2 - \|x - x^*\|^2$$

is independent of the choice of solution. Consequently, minimizing $\|x - x^0\|$ over all solutions x is equivalent to minimizing $\|x - x^*\|$ over all solutions x; the solution to the latter problem is clearly $x = x^*$. ∎

13.6 Simultaneous Iterative Algorithms

In this section we apply the previous theorems to obtain convergence of several simultaneous iterative algorithms for linear systems.

13.6.1 The General Simultaneous Iterative Scheme

In this section we are concerned with simultaneous iterative algorithms having the following iterative step:

$$x_j^k = x_j^{k-1} + \lambda_k \sum_{i=1}^{I} \gamma_{ij} \overline{A_{ij}} (b_i - (Ax^{k-1})_i), \tag{13.26}$$

with $\lambda_k \in [\epsilon, 1]$ and the choices of the parameters γ_{ij} that guarantee convergence. Although we cannot prove convergence for this most general iterative scheme, we are able to prove the following theorems for the separable case of $\gamma_{ij} = \alpha_i \beta_j$.

Theorem 13.5 *If, for some a in the interval $[0, 2]$, we have*

$$\alpha_i \leq r_{ai}^{-1}, \tag{13.27}$$

for each i, and

$$\beta_j \leq c_{aj}^{-1}, \tag{13.28}$$

for each j, then the sequence $\{x^k\}$ given by Equation (13.26) converges to the minimizer of the proximity function

$$\sum_{i=1}^{I} \alpha_i |b_i - (Ax)_i|^2$$

for which

$$\sum_{j=1}^{J} \beta_j^{-1} |x_j - x_j^0|^2$$

is minimized.

Proof: For each i and j, let

$$G_{ij} = \sqrt{\alpha_i} \sqrt{\beta_j} A_{ij},$$

$$z_j = x_j / \sqrt{\beta_j},$$

and

$$d_i = \sqrt{\alpha_i} b_i.$$

Then $Ax = b$ if and only if $Gz = d$. From Corollary 13.7 we have that $\rho(G^\dagger G) \leq 1$. Convergence then follows from Theorem 13.4. ∎

Corollary 13.9 *Let $\gamma_{ij} = \alpha_i\beta_j$, for positive α_i and β_j. If*

$$\alpha_i \leq \Big(\sum_{j=1}^{J} s_j\beta_j|A_{ij}|^2\Big)^{-1}, \tag{13.29}$$

for each i, then the sequence $\{x^k\}$ in (13.26) converges to the minimizer of the proximity function

$$\sum_{i=1}^{I} \alpha_i|b_i - (Ax)_i|^2$$

for which

$$\sum_{j=1}^{J} \beta_j^{-1}|x_j - x_j^0|^2$$

is minimized.

Proof: We know from Corollary 13.3 that $\rho(G^\dagger G) \leq 1$. ∎

13.6.2 Some Convergence Results

We obtain convergence for several known algorithms as corollaries to the previous theorems.

The SIRT Algorithm:

Corollary 13.10 *([147]) For some a in the interval $[0, 2]$ let $\alpha_i = r_{ai}^{-1}$ and $\beta_j = c_{aj}^{-1}$. Then the sequence $\{x^k\}$ in (13.26) converges to the minimizer of the proximity function*

$$\sum_{i=1}^{I} \alpha_i|b_i - (Ax)_i|^2$$

for which

$$\sum_{j=1}^{J} \beta_j^{-1}|x_j - x_j^0|^2$$

is minimized.

For the case of $a = 1$, the iterative step becomes

$$x_j^k = x_j^{k-1} + \sum_{i=1}^{I}\left(\frac{\overline{A_{ij}}(b_i - (Ax^{k-1})_i)}{(\sum_{t=1}^{J}|A_{it}|)(\sum_{m=1}^{I}|A_{mj}|)}\right),$$

which was considered in [97]. The SART algorithm [2] is a special case, in which it is assumed that $A_{ij} \geq 0$, for all i and j.

The CAV Algorithm:

Corollary 13.11 *If $\beta_j = 1$ and α_i satisfies*

$$0 < \alpha_i \leq \Big(\sum_{j=1}^{J} s_j |A_{ij}|^2 \Big)^{-1},$$

for each i, then the algorithm with the iterative step

$$x^k = x^{k-1} + \lambda_k \sum_{i=1}^{I} \alpha_i (b_i - (Ax^{k-1})_i) A_i^\dagger \tag{13.30}$$

converges to the minimizer of

$$\sum_{i=1}^{I} \alpha_i |b_i - (Ax^{k-1})_i|^2$$

for which $\|x - x^0\|$ is minimized.

When

$$\alpha_i = \Big(\sum_{j=1}^{J} s_j |A_{ij}|^2 \Big)^{-1},$$

for each i, this is the relaxed *component-averaging* (CAV) method of Censor et al. [58].

The Landweber Algorithm: When $\beta_j = 1$ and $\alpha_i = \alpha$ for all i and j, we have the relaxed Landweber algorithm. The convergence condition in Equation (13.21) becomes

$$\alpha \leq \Big(\sum_{j=1}^{J} s_j |A_{ij}|^2 \Big)^{-1} = p_i^{-1}$$

for all i, so $\alpha \leq p^{-1}$ suffices for convergence. Actually, the sequence $\{x^k\}$ converges to the minimizer of $\|Ax - b\|$ for which the distance $\|x - x^0\|$ is minimized, for any starting vector x^0, when $0 < \alpha < 1/L$. Easily obtained estimates of L are usually over-estimates, resulting in overly conservative choices of α. For example, if A is first normalized so that $\sum_{j=1}^{J} |A_{ij}|^2 = 1$ for each i, then the trace of $A^\dagger A$ equals I, which tells us that $L \leq I$. But this estimate, which is the one used in Cimmino's method [64], is far too large when A is sparse.

The Simultaneous DROP Algorithm:

Corollary 13.12 *Let* $0 < w_i \leq 1$,

$$\alpha_i = w_i \nu_i^{-2} = w_i \Big(\sum_{j=1}^{J} |A_{ij}|^2 \Big)^{-1}$$

and $\beta_j = s_j^{-1}$, *for each* i *and* j. *Then the simultaneous algorithm with the iterative step*

$$x_j^k = x_j^{k-1} + \lambda_k \sum_{i=1}^{I} \left(\frac{w_i \overline{A_{ij}} (b_i - (Ax^{k-1})_i)}{s_j \nu_i^2} \right), \tag{13.31}$$

converges to the minimizer of the function

$$\sum_{i=1}^{I} \left| \frac{w_i (b_i - (Ax)_i)}{\nu_i} \right|^2$$

for which the function

$$\sum_{j=1}^{J} s_j |x_j - x_j^0|^2$$

is minimized.

For $w_i = 1$, this is the CARP1 algorithm of [95] (see also [72, 58, 59]). The simultaneous DROP algorithm of [56] requires only that the weights w_i be positive, but dividing each w_i by their maximum, $\max_i\{w_i\}$, while multiplying each λ_k by the same maximum, gives weights in the interval $(0, 1]$. For convergence of their algorithm, we need to replace the condition $\lambda_k \leq 2 - \epsilon$ with $\lambda_k \leq \frac{2-\epsilon}{\max_i\{w_i\}}$.

The denominator in CAV is

$$\sum_{t=1}^{J} s_t |A_{it}|^2,$$

while that in CARP1 is

$$s_j \sum_{t=1}^{J} |A_{it}|^2.$$

It was reported in [95] that the two methods differed only slightly in the simulated cases studied.

13.7 Block-iterative Algorithms

The methods discussed in the previous section are *simultaneous*, that is, all the equations are employed at each step of the iteration. We turn now to *block-iterative methods*, which employ only some of the equations at each step. When the parameters are appropriately chosen, block-iterative methods can be significantly faster than simultaneous ones.

13.7.1 The Block-Iterative Landweber Algorithm

For a given set of blocks, the block-iterative Landweber algorithm has the following iterative step: with $n = k(\mathrm{mod}\, N)$,

$$x^k = x^{k-1} + \gamma_n A_n^\dagger (b^n - A_n x^{k-1}). \tag{13.32}$$

The sequence $\{x^k\}$ converges to the solution of $Ax = b$ that minimizes $\|x - x^0\|$, whenever the system $Ax = b$ has solutions, provided that the parameters γ_n satisfy the inequalities $0 < \gamma_n < 1/L_n$. This follows from Theorem 13.4 by replacing the matrices A_n with $\sqrt{\gamma_n} A_n$ and the vectors b^n with $\sqrt{\gamma_n} b^n$.

If the rows of the matrices A_n are normalized to have length one, then we know that $L_n \leq s_n$. Therefore, we can use parameters γ_n that satisfy

$$0 < \gamma_n \leq \Big(s_n \sum_{j=1}^J |A_{ij}|^2 \Big)^{-1}, \tag{13.33}$$

for each $i \in B_n$.

13.7.2 The BICAV Algorithm

We can extend the block-iterative Landweber algorithm as follows: let $n = k(\mathrm{mod}\, N)$ and

$$x^k = x^{k-1} + \lambda_k \sum_{i \in B_n} \gamma_i (b_i - (Ax^{k-1})_i) A_i^\dagger. \tag{13.34}$$

It follows from Theorem 13.2 that, in the consistent case, the sequence $\{x^k\}$ converges to the solution of $Ax = b$ that minimizes $\|x - x^0\|$, provided that, for each n and each $i \in B_n$, we have

$$\gamma_i \leq \Big(\sum_{j=1}^J s_{nj} |A_{ij}|^2 \Big)^{-1}.$$

The BICAV algorithm [59] uses

$$\gamma_i = \Big(\sum_{j=1}^J s_{nj} |A_{ij}|^2 \Big)^{-1}.$$

The iterative step of BICAV is

$$x^k = x^{k-1} + \lambda_k \sum_{i \in B_n} \left(\frac{b_i - (Ax^{k-1})_i}{\sum_{t=1}^J s_{nt} |A_{it}|^2} \right) A_i^\dagger. \tag{13.35}$$

13.7.3 A Block-Iterative CARP1

The obvious way to obtain a block-iterative version of CARP1 would be to replace the denominator term

$$s_j \sum_{t=1}^{J} |A_{it}|^2$$

with

$$s_{nj} \sum_{t=1}^{J} |A_{it}|^2.$$

However, this is problematic, since we cannot redefine the vector of unknowns using $z_j = x_j \sqrt{s_{nj}}$, since this varies with n. In [56], this issue is resolved by taking τ_j to be not less than the maximum of the s_{nj}, and using the denominator

$$\tau_j \sum_{t=1}^{J} |A_{it}|^2 = \tau_j \nu_i^2.$$

A similar device is used in [103] to obtain a convergent block-iterative version of SART. The iterative step of DROP is

$$x_j^k = x_j^{k-1} + \lambda_k \sum_{i \in B_n} \left(\overline{A_{ij}} \frac{(b_i - (Ax^{k-1})_i)}{\tau_j \nu_i^2} \right). \tag{13.36}$$

Convergence of the DROP (*diagonally-relaxed orthogonal projection*) iteration follows from their Theorem 11. We obtain convergence as a corollary of our previous results.

The change of variables is $z_j = x_j \sqrt{\tau_j}$, for each j. Using our eigenvalue bounds, it is easy to show that the matrices C_n with entries

$$(C_n)_{ij} = \left(\frac{A_{ij}}{\sqrt{\tau_j} \nu_i} \right),$$

for all $i \in B_n$ and all j, have $\rho(C_n^\dagger C_n) \leq 1$. The resulting iterative scheme, which is equivalent to Equation (13.36), then converges, whenever $Ax = b$ is consistent, to the solution minimizing the proximity function

$$\sum_{i=1}^{I} \left| \frac{b_i - (Ax)_i}{\nu_i} \right|^2$$

for which the function

$$\sum_{j=1}^{J} \tau_j |x_j - x_j^0|^2$$

is minimized.

13.7.4 Using Sparseness

Suppose, for the sake of illustration, that each column of A has s non-zero elements, for some $s < I$, and we let $r = s/I$. Suppose also that the number of members of B_n is $I_n = I/N$ for each n, and that N is not too large. Then s_n is approximately equal to $rI_n = s/N$. On the other hand, unless A_n has only zero entries, we know that $s_n \geq 1$. Therefore, it is no help to select N for which $s/N < 1$. For a given degree of sparseness s we need not select N greater than s. The more sparse the matrix A, the fewer blocks we need to gain the maximum advantage from the rescaling, and the more we can benefit from parallelization in the calculations at each step of the algorithm in Equation (13.23).

13.8 Iterative Regularization

As we noted in our discussion of the ART, it is often the case that the entries of the vector b in the system $Ax = b$ come from measurements, so are usually noisy. If the entries of b are noisy but the system $Ax = b$ remains consistent (which can easily happen in the under-determined case, with $J > I$), the ART begun at $x^0 = 0$ converges to the solution having minimum norm, but this norm can be quite large. The resulting solution is probably useless. Instead of solving $Ax = b$, we can *regularize* by minimizing, for example, the function $F_\epsilon(x)$ given by

$$F_\epsilon(x) = (1 - \epsilon)\|Ax - b\|_2^2 + \epsilon\|x - p\|_2^2, \tag{13.37}$$

where $\epsilon > 0$ and vector p is a prior estimate of the desired solution.

Lemma 13.4 . *The function F_ϵ always has a unique minimizer $\hat{x}_\epsilon$, given by*

$$\hat{x}_\epsilon = ((1 - \epsilon)A^\dagger A + \epsilon I)^{-1}((1 - \epsilon)A^\dagger b + \epsilon p); \tag{13.38}$$

this is a regularized solution of $Ax = b$. Note that the inverse above always exists.

If $p = 0$, then

$$\hat{x}_\epsilon = (A^\dagger A + \gamma^2 I)^{-1} A^\dagger b, \tag{13.39}$$

for $\gamma^2 = \frac{\epsilon}{1-\epsilon}$. However, we do not want to calculate $A^\dagger A + \gamma^2 I$, in order to solve

$$(A^\dagger A + \gamma^2 I)x = A^\dagger b, \tag{13.40}$$

when the matrix A is large. Fortunately, there are ways to find $\hat{x}_\epsilon$, using only the matrix A. We saw previously how this might be accomplished using the ART; now we show how Landweber's Algorithm can be used to calculate this regularized solution.

13.8.1 Iterative Regularization with Landweber's Algorithm

Our goal is to minimize the function in Equation (13.37), with $p = 0$. Notice that this is equivalent to minimizing the function

$$F(x) = \|Bx - c\|_2^2, \tag{13.41}$$

for

$$B = \begin{bmatrix} A \\ \gamma I \end{bmatrix}, \tag{13.42}$$

and

$$c = \begin{bmatrix} b \\ 0 \end{bmatrix}, \tag{13.43}$$

where 0 denotes a column vector with all entries equal to zero and $\gamma^2 = \frac{\epsilon}{1-\epsilon}$. The Landweber iteration for the problem $Bx = c$ is

$$x^k = x^{k-1} + \alpha B^T(c - Bx^{k-1}), \tag{13.44}$$

for $0 < \alpha < 2/\rho(B^TB)$, where $\rho(B^TB)$ is the spectral radius of B^TB, and we assume that the rows of B have length one. Equation (13.44) can be written as

$$x^{k+1} = (1 - \alpha\gamma^2)x^k + \alpha A^T(b - Ax^k). \tag{13.45}$$

We see from Equation (13.45) that Landweber's Algorithm for solving the regularized least-squares problem amounts to a relaxed version of Landweber's Algorithm applied to the original least-squares problem.

13.9 Exercises

13.1 *Prove Lemma 13.1.*

13.2 (Computer Problem) *Compare the speed of convergence of the ART and Cimmino algorithms.*

13.3 (Computer Problem) *By generating sparse matrices of various sizes, test the accuracy of the estimates of the largest singular-value given above.*

Chapter 14

The Split Feasibility Problem

The *split feasibility problem* (SFP) [55] is to find $c \in C$ with $Ac \in Q$, if such points exist, where A is a real I by J matrix and C and Q are nonempty, closed convex sets in R^J and R^I, respectively. When there is no exact solution to the SFP the CQ algorithm optimizes a certain proximity measure. In this chapter we discuss the CQ algorithm for solving the SFP, as well as recent extensions and applications.

14.1 The CQ Algorithm

In [41] the CQ algorithm for solving the SFP was presented, for the real case. It has the iterative step

$$x^{k+1} = P_C(x^k - \gamma A^T(I - P_Q)Ax^k), \tag{14.1}$$

where I is the identity operator and $\gamma \in (0, 2/\rho(A^T A))$, for $\rho(A^T A)$ the spectral radius of the matrix $A^T A$, which is also its largest eigenvalue. The CQ algorithm can be extended to the complex case, in which the matrix A has complex entries, and the sets C and Q are in C^J and C^I, respectively. The iterative step of the extended CQ algorithm is then

$$x^{k+1} = P_C(x^k - \gamma A^\dagger(I - P_Q)Ax^k). \tag{14.2}$$

The CQ algorithm converges to a solution of the SFP, for any starting vector x^0, whenever the SFP has solutions. When the SFP has no solutions, the CQ algorithm converges to a minimizer of the function

$$f(x) = \frac{1}{2}||P_Q Ax - Ax||_2^2 \tag{14.3}$$

over the set C, provided such constrained minimizers exist. Therefore the CQ algorithm is an iterative constrained optimization method. As shown in [42], convergence of the CQ algorithm is a consequence of the KM Theorem 10.2.

The function $f(x)$ is convex and differentiable on R^J and its derivative is the operator

$$\nabla f(x) = A^T(I - P_Q)Ax; \tag{14.4}$$

see [5].

Lemma 14.1 *The derivative operator ∇f is λ-Lipschitz continuous for $\lambda = \rho(A^TA)$, therefore it is ν-ism for $\nu = \frac{1}{\lambda}$.*

Proof: We have

$$||\nabla f(x) - \nabla f(y)||_2^2 = ||A^T(I - P_Q)Ax - A^T(I - P_Q)Ay||_2^2 \tag{14.5}$$

$$\leq \lambda||(I - P_Q)Ax - (I - P_Q)Ay||_2^2. \tag{14.6}$$

Also

$$||(I - P_Q)Ax - (I - P_Q)Ay||_2^2 = ||Ax - Ay||_2^2 \tag{14.7}$$

$$+||P_QAx - P_QAy||_2^2 - 2\langle P_QAx - P_QAy, Ax - Ay\rangle \tag{14.8}$$

and, since P_Q is fne,

$$\langle P_QAx - P_QAy, Ax - Ay\rangle \geq ||P_QAx - P_QAy||_2^2. \tag{14.9}$$

Therefore,

$$||\nabla f(x) - \nabla f(y)||_2^2 \leq \lambda(||Ax - Ay||_2^2 - ||P_QAx - P_QAy||_2^2) \tag{14.10}$$

$$\leq \lambda||Ax - Ay||_2^2 \leq \lambda^2||x - y||_2^2. \tag{14.11}$$

This completes the proof. ∎

If $\gamma \in (0, 2/\lambda)$ then $B = P_C(I - \gamma A^T(I - P_Q)A)$ is av and, by the KM Theorem 10.2, the orbit sequence $\{B^kx\}$ converges to a fixed point of B, whenever such points exist. If z is a fixed point of B, then $z = P_C(z - \gamma A^T(I - P_Q)Az)$. Therefore, for any c in C we have

$$\langle c - z, z - (z - \gamma A^T(I - P_Q)Az)\rangle \geq 0. \tag{14.12}$$

This tells us that

$$\langle c - z, A^T(I - P_Q)Az\rangle \geq 0, \tag{14.13}$$

which means that z minimizes $f(x)$ relative to the set C.

The CQ algorithm employs the relaxation parameter γ in the interval $(0, 2/L)$, where L is the largest eigenvalue of the matrix A^TA. Choosing the best relaxation parameter in any algorithm is a nontrivial procedure. Generally speaking, we want to select γ near to $1/L$. We saw a simple estimate for L in our discussion of singular values of sparse matrices: if A is normalized so that each row has length one, then the spectral radius of A^TA does not exceed the maximum number of nonzero elements in any column of A. A similar upper bound on $\rho(A^TA)$ was obtained for non-normalized, ϵ-sparse A.

14.2 Particular Cases of the CQ Algorithm

It is easy to find important examples of the SFP: if $C \subseteq R^J$ and $Q = \{b\}$ then solving the SFP amounts to solving the linear system of equations $Ax = b$; if C is a proper subset of R^J, such as the nonnegative cone, then we seek solutions of $Ax = b$ that lie within C, if there are any. Generally, we cannot solve the SFP in closed form and iterative methods are needed.

A number of well known iterative algorithms, such as the Landweber [110] and projected Landweber methods (see [12]), are particular cases of the CQ algorithm.

14.2.1 The Landweber algorithm

With x^0 arbitrary and $k = 0, 1, ...$ let

$$x^{k+1} = x^k + \gamma A^T(b - Ax^k). \tag{14.1}$$

This is the Landweber algorithm.

14.2.2 The Projected Landweber Algorithm

For a general nonempty closed convex C, x^0 arbitrary, and $k = 0, 1, ...$, the projected Landweber method for finding a solution of $Ax = b$ in C has the iterative step

$$x^{k+1} = P_C(x^k + \gamma A^T(b - Ax^k)). \tag{14.2}$$

14.2.3 Convergence of the Landweber Algorithms

From the convergence theorem for the CQ algorithm it follows that the Landweber algorithm converges to a solution of $Ax = b$ and the projected Landweber algorithm converges to a solution of $Ax = b$ in C, whenever

such solutions exist. When there are no solutions of the desired type, the Landweber algorithm converges to a least squares approximate solution of $Ax = b$, while the projected Landweber algorithm will converge to a minimizer, over the set C, of the function $||b - Ax||_2$, whenever such a minimizer exists.

Another example of the CQ algorithm is the *simultaneous algebraic reconstruction technique* (SART) of Anderson and Kak for solving $Ax = b$, for nonnegative matrix A [2]. We discussed SART in the previous chapter.

14.2.4 Application of the CQ Algorithm in Dynamic ET

To illustrate how an image reconstruction problem can be formulated as a SFP, we consider briefly *emission computed tomography* (ET) image reconstruction. The objective in ET is to reconstruct the internal spatial distribution of intensity of a radionuclide from counts of photons detected outside the patient. In static ET the intensity distribution is assumed constant over the scanning time. Our data are photon counts at the detectors, forming the positive vector b and we have a matrix A of detection probabilities; our model is $Ax = b$, for x a nonnegative vector. We could then take $Q = \{b\}$ and $C = R^N_+$, the nonnegative cone in R^N.

In *dynamic* ET [84] the intensity levels at each voxel may vary with time. The observation time is subdivided into, say, T intervals and one static image, call it x^t, is associated with the time interval denoted by t, for $t = 1, ..., T$. The vector x is the concatenation of these T image vectors x^t. The discrete time interval at which each data value is collected is also recorded and the problem is to reconstruct this succession of images.

Because the data associated with a single time interval is insufficient, by itself, to generate a useful image, one often uses prior information concerning the time history at each fixed voxel to devise a model of the behavior of the intensity levels at each voxel, as functions of time. One may, for example, assume that the radionuclide intensities at a fixed voxel are increasing with time, or are concave (or convex) with time. The problem then is to find $x \geq 0$ with $Ax = b$ and $Dx \geq 0$, where D is a matrix chosen to describe this additional prior information. For example, we may wish to require that, for each fixed voxel, the intensity is an increasing function of (discrete) time; then we want

$$x_j^{t+1} - x_j^t \geq 0, \tag{14.3}$$

for each t and each voxel index j. Or, we may wish to require that the intensity at each voxel describes a concave function of time, in which case nonnegative second differences would be imposed:

$$(x_j^{t+1} - x_j^t) - (x_j^{t+2} - x_j^{t+1}) \geq 0. \tag{14.4}$$

In either case, the matrix D can be selected to include the left sides of these inequalities, while the set Q can include the nonnegative cone as one factor.

14.2.5 Related Methods and Applications

One of the obvious drawbacks to the use of the CQ algorithm is that we would need the projections P_C and P_Q to be easily calculated. Several authors have offered remedies for that problem, using approximations of the convex sets by the intersection of hyperplanes and orthogonal projections onto those hyperplanes [150].

In a recent papers [57, 53] Censor *et al.* discuss the application of the CQ algorithm to the problem of intensity-modulated radiation therapy (IMRT) treatment planning. Mathematically speaking, the problem is the *multi-set split feasibility problem* (MSSFP), which is to find x in C, the non-empty intersection of closed, convex sets C_i, for $i = 1, ..., I$, such that Ax is in the non-empty intersection of the closed, convex sets Q_j, for $j = 1, ..., J$. In the CQ algorithm it is assumed that the orthogonal projections onto C and Q are easily calculated, while algorithms for solving the MSSFP assume that the orthogonal projections onto the C_i and Q_j are easily calculated.

The split feasibility problem can be formulated as an optimization problem, namely, to minimize

$$h(x) = \psi_C(x) + \psi_Q(Ax), \tag{14.5}$$

where $\psi_C(x)$ is the indicator function of the set C. The CQ algorithm solves the more general problem of minimizing the function

$$f(x) = \psi_C(x) + ||P_Q Ax - Ax||_2^2. \tag{14.6}$$

The second term in $f(x)$ is differentiable, allowing us to apply the forward-backward splitting method of Combettes and Wajs [66], to be discussed in a subsequent chapter. The CQ algorithm is then a special case of their method.

14.3 Exercises

14.1 *Use the CQ algorithm to prove the following. Let C_1 and C_2 be nonempty, closed convex sets in R^J, with $C_1 \cap C_2 = \emptyset$. Assume that there is a unique $\hat{c}_2$ in C_2 minimizing the function $f(x) = ||c_2 - P_1 c_2||_2$, over all c_2 in C_2. Let $\hat{c}_1 = P_1 \hat{c}_2$. Then $P_2 \hat{c}_1 = \hat{c}_2$. Let z^0 be arbitrary and, for $n = 0, 1, ...$, let*

$$z^{2n+1} = P_1 z^{2n}, \tag{14.7}$$

and

$$z^{2n+2} = P_2 z^{2n+1}. \tag{14.8}$$

Then

$$\{z^{2n+1}\} \to \hat{c}_1, \tag{14.9}$$

and

$$\{z^{2n}\} \to \hat{c}_2. \tag{14.10}$$

Chapter 15

The Multiplicative ART (MART)

The *multiplicative* ART (MART) [94] is an iterative algorithm closely related to the ART. It applies to systems of linear equations $Ax = b$ for which the b_i are positive and the A_{ij} are nonnegative; the solution x we seek will have nonnegative entries. When there are multiple nonnegative solutions, the MART finds the solution that minimizes the cross-entropy to the starting vector; if the entries of the starting vector are all the same, the MART finds the solution that maximizes Shannon entropy.

It is not so easy to see the relation between ART and MART if we look at the most general formulation of MART. For that reason, we begin with a simpler case, in which the relation is most clearly visible.

15.1 A Special Case of MART

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

$$x_j^{k+1} = x_j^k + \frac{1}{|L_i|}(b_i - (Ax^k)_i), \tag{15.1}$$

for j in L_i, and

$$x_j^{k+1} = x_j^k, \tag{15.2}$$

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

equally over each of the pixels that intersects L_i.

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

The MART, in this case, has the iterative step

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i} \Big), \tag{15.3}$$

for those j in L_i, and

$$x_j^{k+1} = x_j^k, \tag{15.4}$$

otherwise. Therefore, we can write the iterative step as

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i} \Big)^{A_{ij}}. \tag{15.5}$$

15.2 MART in the General Case

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

The MART, which can be applied only to nonnegative systems, is a sequential, or row-action, method that uses one equation only at each step of the iteration. We present the general MART algorithm, and then two versions of the MART.

The general MART algorithm is the following [43].

Algorithm 15.1 (The General MART) *Let x^0 be any positive vector, and $i = k(\mathrm{mod}\, I) + 1$. Having found x^k for positive integer k, define x^{k+1} by*

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i} \Big)^{\gamma_j \delta_i A_{ij}}. \tag{15.6}$$

The parameters $\gamma_j > 0$ and $\delta_i > 0$ are to be chosen subject to the inequality

$$\gamma_j \delta_i A_{ij} \leq 1,$$

for all i and j.

The first version of MART that we shall consider, MART I, uses the parameters $\gamma_j = 1$, and

$$\delta_i = 1/\max\{A_{ij} \,|j = 1, ..., J\}.$$

Algorithm 15.2 (MART I) *Let x^0 be any positive vector, and $i = k(\mathrm{mod}\, I)+1$. Having found x^k for positive integer k, define x^{k+1} by*

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i}\Big)^{m_i^{-1}A_{ij}}, \tag{15.7}$$

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

Some treatments of MART leave out the m_i, but require only that the entries of A have been rescaled so that $A_{ij} \leq 1$ for all i and j; this corresponds to the choices $\gamma_j = 1$ and

$$\delta_i = \delta = 1/\max\{A_{ij} \,|i = 1, ..., I,\ j = 1, ..., J\},$$

for each i. Using the m_i is important, however, in accelerating the convergence of MART.

The second version of MART that we shall consider, MART II, uses the parameters $\gamma_j = s_j^{-1}$, and

$$\delta_i = 1/\max\{A_{ij}s_j^{-1} \,|j = 1, ..., J\}.$$

Algorithm 15.3 (MART II) *Let x^0 be any positive vector, and $i = k(\mathrm{mod}\, I) + 1$. Having found x^k for positive integer k, define x^{k+1} by*

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i}\Big)^{n_i^{-1}s_j^{-1}A_{ij}}, \tag{15.8}$$

where $n_i = \delta_i^{-1} = \max\{A_{ij}s_j^{-1} \,|j = 1, 2, ..., J\}$.

Note that the MART II algorithm can be obtained from the MART I algorithm if we first rescale the entries of the matrix A, replacing A_{ij} with $A_{ij}s_j^{-1}$, and redefine the vector of unknowns, replacing each x_j with $x_j s_j$.

The MART can be accelerated by relaxation, as well.

Algorithm 15.4 (Relaxed MART I) *Let x^0 be any positive vector, and $i = k(\mathrm{mod}\, I) + 1$. Having found x^k for positive integer k, define x^{k+1} by*

$$x_j^{k+1} = x_j^k \Big(\frac{b_i}{(Ax^k)_i}\Big)^{\tau_i m_i^{-1}A_{ij}}, \tag{15.9}$$

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

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

15.3 ART and MART as Sequential Projection Methods

We know from our discussion of the ART that the iterative ART step can be viewed as the orthogonal projection of the current vector, x^k, onto H_i, the hyperplane associated with the i-th equation. Can we view MART in a similar way? Yes, but we need to consider a different measure of closeness between nonnegative vectors.

15.3.1 Cross-Entropy or the Kullback-Leibler Distance

For positive numbers u and v, the Kullback-Leibler distance [108] from u to v is

$$KL(u,v) = u\log\frac{u}{v} + v - u. \tag{15.10}$$

We also define $KL(0,0) = 0$, $KL(0,v) = v$ and $KL(u,0) = +\infty$. The KL distance is extended to nonnegative vectors component-wise, so that for nonnegative vectors x and z we have

$$KL(x,z) = \sum_{j=1}^{J} KL(x_j, z_j). \tag{15.11}$$

We turn now to the various uses of the KL distance in the discussion of the MART.

15.3.2 Convergence of MART

In the consistent case, by which we mean that $Ax = b$ has nonnegative solutions, we have the following convergence theorem for MART [43]. We assume that $s_j = \sum_{i=1}^{I} A_{ij}$ is positive, for all j.

Theorem 15.1 *In the consistent case, the general MART algorithm converges to the unique non-negative solution of $Ax = b$ for which the weighted cross-entropy*

$$\sum_{j=1}^{J} \gamma_j^{-1} KL(x_j, x_j^0)$$

is minimized. The MART I algorithm converges to the unique nonnegative solution of $Ax = b$ for which the cross-entropy $KL(x, x^0)$ is minimized. The MART II algorithm converges to the unique non-negative solution of $Ax = b$ for which the weighted cross-entropy

$$\sum_{j=1}^{J} s_j KL(x_j, x_j^0)$$

is minimized.

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

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

15.3.3 Projecting with the KL Distance

Given the vector x^k, we find the vector z in H_i for which the KL distance $f(z) = KL(x^k, z)$ is minimized; this z will be the KL projection of x^k onto H_i. Using a Lagrange multiplier, we find that

$$0 = \frac{\partial f}{\partial z_j}(z) - \lambda_i A_{ij}, \tag{15.12}$$

for some constant λ_i, so that

$$0 = -\frac{x_j^k}{z_j} + 1 - \lambda_i A_{ij}, \tag{15.13}$$

for each j. Multiplying by z_j, we get

$$z_j - x_j^k = z_j A_{ij} \lambda_i. \tag{15.14}$$

For the special case in which the entries of A_{ij} are zero or one, we can solve Equation (15.14) for z_j. We have

$$z_j - x_j^k = z_j \lambda_i, \tag{15.15}$$

for each $j \in L_i$, and $z_j = x_j^k$, otherwise. Multiply both sides by A_{ij} and sum on j to get

$$b_i(1 - \lambda_i) = (Ax^k)_i. \tag{15.16}$$

Therefore,

$$z_j = x_j^k \frac{b_i}{(Ax^k)_i}, \tag{15.17}$$

which is clearly x_j^{k+1}. So, at least in the special case we have been discussing, MART consists of projecting, in the KL sense, onto each of the hyperplanes in succession.

15.3.4 Weighted KL Projections

For the more general case in which the entries A_{ij} are arbitrary nonnegative numbers, we cannot directly solve for z_j in Equation (15.14). There is an alternative, though. Instead of minimizing $KL(x, z)$, subject to $(Az)_i = b_i$, we minimize the weighted KL distance

$$\sum_{j=1}^{J} A_{ij} KL(x_j, z_j), \tag{15.18}$$

subject to the same constraint on z. The optimal z is $Q_i^e x$, which we shall denote here by $Q_i x$, the weighted KL projection of x onto the ith hyperplane. Again using a Lagrange multiplier approach, we find that

$$0 = -A_{ij}(\frac{x_j}{z_j} + 1) - A_{ij}\lambda_i, \tag{15.19}$$

for some constant λ_i. Multiplying by z_j, we have

$$A_{ij} z_j - A_{ij} x_j = A_{ij} z_j \lambda_i. \tag{15.20}$$

Summing over the index j, we get

$$b_i - (Ax)_i = b_i \lambda_i, \tag{15.21}$$

from which it follows that

$$1 - \lambda_i = (Ax)_i / b_i. \tag{15.22}$$

Substituting for λ_i in equation (15.20), we obtain

$$z_j = (Q_i x)_j = x_j \frac{b_i}{(Ax)_i}, \tag{15.23}$$

for all j for which $A_{ij} \neq 0$.

Note that the MART step does not define x^{k+1} to be this weighted KL projection of x^k onto the hyperplane H_i; that is,

$$x_j^{k+1} \neq (Q_i x^k)_j, \tag{15.24}$$

except for those j for which $\frac{A_{ij}}{m_i} = 1$. What is true is that the MART step involves relaxation. Writing

$$x_j^{k+1} = (x_j^k)^{1 - m_i^{-1} A_{ij}} \left(x_j^k \frac{b_i}{(Ax^k)_i} \right)^{m_i^{-1} A_{ij}}, \tag{15.25}$$

we see that x_j^{k+1} is a weighted geometric mean of x_j^k and $(Q_i x^k)_j$.

15.4 Proof of Convergence for MART I

We assume throughout this proof that $\hat{x}$ is a nonnegative solution of $Ax = b$. For $i = 1, 2, ..., I$, let

$$G_i(x, z) = KL(x, z) + m_i^{-1} KL((Ax)_i, b_i) - m_i^{-1} KL((Ax)_i, (Az)_i). \tag{15.26}$$

Lemma 15.1 *For all i, we have $G_i(x, z) \geq 0$ for all x and z.*

Proof: Use Equation (17.35). ∎

Then $G_i(x, z)$, viewed as a function of z, is minimized by $z = x$, as we see from the equation

$$G_i(x, z) = G_i(x, x) + KL(x, z) - m_i^{-1} KL((Ax)_i, (Az)_i). \tag{15.27}$$

Viewed as a function of x, $G_i(x, z)$ is minimized by $x = z'$, where

$$z'_j = z_j \Big(\frac{b_i}{(Az)_i} \Big)^{m_i^{-1} A_{ij}}, \tag{15.28}$$

as we see from the equation

$$G_i(x, z) = G_i(z', z) + KL(x, z'). \tag{15.29}$$

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

Now we calculate $G_i(\hat{x}, x^k)$ in two ways, using, first, the definition, and, second, Equation (15.29). From the definition, we have

$$G_i(\hat{x}, x^k) = KL(\hat{x}, x^k) - m_i^{-1} KL(b_i, (Ax^k)_i). \tag{15.30}$$

From Equation (15.29), we have

$$G_i(\hat{x}, x^k) = G_i(x^{k+1}, x^k) + KL(\hat{x}, x^{k+1}). \tag{15.31}$$

Therefore,

$$KL(\hat{x}, x^k) - KL(\hat{x}, x^{k+1}) = G_i(x^{k+1}, x^k) + m_i^{-1} KL(b_i, (Ax^k)_i). \tag{15.32}$$

From Equation (15.32) we can conclude several things:

- 1) the sequence $\{KL(\hat{x}, x^k)\}$ is decreasing;
- 2) the sequence $\{x^k\}$ is bounded, and therefore has a cluster point, x^*; and
- 3) the sequences $\{G_i(x^{k+1}, x^k)\}$ and $\{m_i^{-1} KL(b_i, (Ax^k)_i)\}$ converge decreasingly to zero, and so $b_i = (Ax^*)_i$ for all i.

Since $b = Ax^*$, we can use x^* in place of the arbitrary solution $\hat{x}$ to conclude that the sequence $\{KL(x^*, x^k)\}$ is decreasing. But, a subsequence converges to zero, so the entire sequence must converge to zero, and therefore $\{x^k\}$ converges to x^*. Finally, since the right side of Equation (15.32) is independent of which solution $\hat{x}$ we have used, so is the left side. Summing over k on the left side, we find that

$$KL(\hat{x}, x^0) - KL(\hat{x}, x^*) \tag{15.33}$$

is independent of which $\hat{x}$ we use. We can conclude then that minimizing $KL(\hat{x}, x^0)$ over all solutions $\hat{x}$ has the same answer as minimizing $KL(\hat{x}, x^*)$ over all such $\hat{x}$; but the solution to the latter problem is obviously $\hat{x} = x^*$. This concludes the proof. ∎

The proof of convergence of MART II is similar, and we omit it. The interested reader may consult [43].

15.5 Comments on the Rate of Convergence of MART

We can see from Equation (15.32),

$$KL(\hat{x}, x^k) - KL(\hat{x}, x^{k+1}) = G_i(x^{k+1}, x^k) + m_i^{-1} KL(b_i, (Ax^k)_i), \tag{15.34}$$

that the decrease in distance to a solution that occurs with each step of MART depends on m_i^{-1} and on $KL(b_i, (Ax^k)_i)$; the latter measures the extent to which the current vector x^k solves the current equation. We see then that it is reasonable to select m_i as we have done, namely, as the smallest positive number c_i for which $A_{ij}/c_i \leq 1$ for all j. We also see that it is helpful if the equations are ordered in such a way that $KL(b_i, (Ax^k)_i)$ is fairly large, for each k. It is not usually necessary to determine an optimal ordering of the equations; the important thing is to avoid ordering the equations so that successive hyperplanes have nearly parallel normal vectors.

15.6 Exercises

15.1 *Prove Lemma 15.1. Hint: Use Lemma 17.5.*

Chapter 16

Rescaled Block-Iterative (RBI) Methods

Image reconstruction problems in tomography are often formulated as statistical likelihood maximization problems in which the pixel values of the desired image play the role of parameters. Iterative algorithms based on cross-entropy minimization, such as the *expectation maximization maximum likelihood* (EMML) method and the *simultaneous multiplicative algebraic reconstruction technique* (SMART) can be used to solve such problems. Because the EMML and SMART are slow to converge for large amounts of data typical in imaging problems acceleration of the algorithms using blocks of data or ordered subsets has become popular. There are a number of different ways to formulate these block-iterative versions of EMML and SMART, involving the choice of certain normalization and regularization parameters. These methods are not faster merely because they are block-iterative; the correct choice of the parameters is crucial [43].

16.1 Overview

The algorithms we discuss here have interesting histories, which we sketch in this section.

16.1.1 The SMART and its variants

Like the ART, the MART has a simultaneous version, called the SMART. Like MART, SMART applies only to nonnegative systems of equations. Unlike MART, SMART is a simultaneous algorithm that uses all equations in each step of the iteration. The SMART was discovered in 1972, independently, by Darroch and Ratcliff, working in statistics, [68] and by Schmidlin

[135] in medical imaging; neither work makes reference to MART. Darroch and Ratcliff do consider block-iterative versions of their algorithm, in which only some of the equations are used at each step, but their convergence proof involves unnecessary restrictions on the system matrix. Censor and Segman [62] seem to be the first to present the SMART and its block-iterative variants explicitly as generalizations of MART.

16.1.2 The EMML and its variants

The *expectation maximization maximum likelihood* (EMML) method turns out to be closely related to the SMART, although it has quite a different history. The EMML algorithm we discuss here is actually a special case of a more general approach to likelihood maximization, usually called the EM algorithm [69]; the book by McLachnan and Krishnan [120] is a good source for the history of this more general algorithm.

It was noticed by Rockmore and Macovski [134] that certain image reconstruction problems posed by medical tomography could be formulated as statistical parameter estimation problems. Following up on this idea, Shepp and Vardi [136] suggested the use of the EM algorithm for solving the reconstruction problem in emission tomography. In [111], Lange and Carson presented an EM-type iterative method for transmission tomographic image reconstruction, and pointed out a gap in the convergence proof given in [136] for the emission case. In [148], Vardi, Shepp and Kaufman repaired the earlier proof, relying on techniques due to Csiszár and Tusnády [67]. In [112] Lange, Bahn and Little improved the transmission and emission algorithms, by including regularization to reduce the effects of noise. The question of uniqueness of the solution in the inconsistent case was resolved in [31].

The MART and SMART were initially designed to apply to consistent systems of equations. Darroch and Ratcliff did not consider what happens in the inconsistent case, in which the system of equations has no non-negative solutions; this issue was resolved in [31], where it was shown that the SMART converges to a non-negative minimizer of the Kullback-Leibler distance $KL(Ax, b)$. The EMML, as a statistical parameter estimation technique, was not originally thought to be connected to any system of linear equations. In [31], it was shown that the EMML leads to a non-negative minimizer of the Kullback-Leibler distance $KL(b, Ax)$, thereby exhibiting a close connection between the SMART and the EMML methods. Consequently, when the non-negative system of linear equations $Ax = b$ has a non-negative solution, the EMML converges to such a solution.

16.1.3 Block-iterative versions of SMART and EMML

As we have seen, Darroch and Ratcliff included what are now called block-iterative versions of SMART in their original paper [68]. Censor and Segman [62] viewed SMART and its block-iterative versions as natural extension of the MART. Consequently, block-iterative variants of SMART have been around for some time. The story with the EMML is quite different.

The paper of Holte, Schmidlin, *et al.* [100] compares the performance of Schmidlin's method of [135] with the EMML algorithm. Almost as an aside, they notice the accelerating effect of what they call *projection interleaving*, that is, the use of blocks. This paper contains no explicit formulas, however, and presents no theory, so one can only make educated guesses as to the precise iterative methods employed. Somewhat later, Hudson, Hutton and Larkin [101, 102] observed that the EMML can be significantly accelerated if, at each step, one employs only some of the data. They referred to this approach as the *ordered subset* EM method (OSEM). They gave a proof of convergence of the OSEM, for the consistent case. The proof relied on a fairly restrictive relationship between the matrix A and the choice of blocks, called *subset balance*. In [34] a revised version of the OSEM, called the *rescaled block-iterative* EMML (RBI-EMML), was shown to converge, in the consistent case, regardless of the choice of blocks.

16.1.4 Basic Assumptions

Methods based on cross-entropy, such as the MART, SMART, EMML and all block-iterative versions of these algorithms apply to nonnegative systems that we denote by $Ax = b$, where b is a vector of positive entries, A is a matrix with entries $A_{ij} \geq 0$ such that for each j the sum $s_j = \sum_{i=1}^{I} A_{ij}$ is positive and we seek a solution x with nonnegative entries. If no nonnegative x satisfies $b = Ax$ we say the system is *inconsistent*.

Simultaneous iterative algorithms employ all of the equations at each step of the iteration; block-iterative methods do not. For the latter methods we assume that the index set $\{i = 1, ..., I\}$ is the (not necessarily disjoint) union of the N sets or *blocks* B_n, $n = 1, ..., N$. We shall require that $s_{nj} = \sum_{i \in B_n} A_{ij} > 0$ for each n and each j. Block-iterative methods like ART and MART for which each block consists of precisely one element are called *row-action* or *sequential* methods. We begin our discussion with the SMART and the EMML method.

16.2 The SMART and the EMML method

Both the SMART and the EMML method provide a solution of $b = Ax$ when such exist and (distinct) approximate solutions in the inconsistent case.

16.2.1 The SMART Algorithm

The SMART algorithm is the following:

Algorithm 16.1 (SMART) *Let x^0 be an arbitrary positive vector. For $k = 0, 1, ...$ let*

$$x_j^{k+1} = x_j^k \exp\Big(s_j^{-1} \sum_{i=1}^{I} A_{ij} \log \frac{b_i}{(Ax^k)_i}\Big). \tag{16.1}$$

The exponential and logarithm in the SMART iterative step are computationally expensive. The main results concerning the SMART are given by the following theorem.

Theorem 16.1 *In the consistent case the SMART converges to the unique nonnegative solution of $b = Ax$ for which the distance $\sum_{j=1}^{J} s_j KL(x_j, x_j^0)$ is minimized. In the inconsistent case it converges to the unique nonnegative minimizer of the distance $KL(Ax, y)$ for which $\sum_{j=1}^{J} s_j KL(x_j, x_j^0)$ is minimized; if A and every matrix derived from A by deleting columns has full rank then there is a unique nonnegative minimizer of $KL(Ax, y)$ and at most $I - 1$ of its entries are nonzero.*

16.2.2 The EMML Algorithm

The EMML method is similar to the SMART, but somewhat less costly to compute.

Algorithm 16.2 (EMML) *Let x^0 be an arbitrary positive vector. For $k = 0, 1, ...$ let*

$$x_j^{k+1} = x_j^k s_j^{-1} \sum_{i=1}^{I} A_{ij} \frac{b_i}{(Ax^k)_i}. \tag{16.2}$$

For the EMML method the main results are the following.

Theorem 16.2 *In the consistent case the EMML algorithm converges to nonnegative solution of $b = Ax$. In the inconsistent case it converges to a nonnegative minimizer of the distance $KL(y, Ax)$; if A and every matrix derived from A by deleting columns has full rank then there is a unique nonnegative minimizer of $KL(y, Ax)$ and at most $I - 1$ of its entries are nonzero.*

In the consistent case there may be multiple nonnegative solutions and the one obtained by the EMML algorithm will depend on the starting vector x^0; how it depends on x^0 is an open question.

These theorems are special cases of more general results on block-iterative methods that we shall consider later in this chapter.

16.2.3 Likelihood Maximization

Both the EMML and SMART are related to likelihood maximization. Minimizing the function $KL(y, Ax)$ is equivalent to maximizing the likelihood when the b_i are taken to be measurements of independent Poisson random variables having means $(Ax)_i$. The entries of x are the parameters to be determined. This situation arises in emission tomography. So the EMML is a likelihood maximizer, as its name suggests.

The connection between SMART and likelihood maximization is a bit more convoluted. Suppose that $s_j = 1$ for each j. The solution of $b = Ax$ for which $KL(x, x^0)$ is minimized necessarily has the form

$$x_j = x_j^0 \exp\Big(\sum_{i=1}^{I} A_{ij}\lambda_i\Big) \tag{16.3}$$

for some vector λ with entries λ_i. This *log linear* form also arises in transmission tomography, where it is natural to assume that $s_j = 1$ for each j and $\lambda_i \leq 0$ for each i. We have the following lemma that helps to connect the SMART algorithm with the transmission tomography problem:

Lemma 16.1 *Minimizing $KL(d, x)$ over x as in Equation (16.3) is equivalent to minimizing $KL(x, x^0)$, subject to $Ax = Ad$.*

The solution to the latter problem can be obtained using the SMART.

With $x_+ = \sum_{j=1}^{J} x_j$ the vector A with entries $p_j = x_j/x_+$ is a probability vector. Let $d = (d_1, ..., d_J)^T$ be a vector whose entries are nonnegative integers, with $K = \sum_{j=1}^{J} d_j$. Suppose that, for each j, p_j is the probability of index j and d_j is the number of times index j was chosen in K trials. The likelihood function of the parameters λ_i is

$$L(\lambda) = \prod_{j=1}^{J} p_j^{d_j} \tag{16.4}$$

so that the log-likelihood function is

$$LL(\lambda) = \sum_{j=1}^{J} d_j \log p_j. \tag{16.5}$$

Since A is a probability vector, maximizing $L(\lambda)$ is equivalent to minimizing $KL(d, p)$ with respect to λ, which, according to the lemma above, can be solved using SMART. In fact, since all of the block-iterative versions of SMART have the same limit whenever they have the same starting vector, any of these methods can be used to solve this maximum likelihood problem. In the case of transmission tomography the λ_i must be nonpositive, so if SMART is to be used, some modification is needed to obtain such a solution.

Those who have used the SMART or the EMML on sizable problems have certainly noticed that they are both slow to converge. An important issue, therefore, is how to accelerate convergence. The partial gradient approach has been helpful in this regard.

16.3 A Partial Gradient Approach

Convergence of the EMML and SMART algorithms, for the consistent case, can be accelerated using the partial gradient approach. The EMML and SMART algorithms both have as their goal the minimization of a function $f(x)$ that has the form

$$f(x) = \sum_{i=1}^{I} f_i(x), \tag{16.6}$$

where each $f_i(x)$ is a non-negative function of the variable x.

16.3.1 The EMML Algorithm

The EMML algorithm minimizes the function $f(x) = KL(b, Ax)$ over non-negative vectors x. The gradient of $f(x)$ at $x = x^k$ has the entries

$$\frac{\partial f}{\partial x_j}(x^k) = \sum_{i=1}^{I} A_{ij}\Big(1 - \frac{b_i}{(Ax^k)_i}\Big) = s_j - \sum_{i=1}^{I} A_{ij}\frac{b_i}{(Ax^k)_i}, \tag{16.7}$$

with

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

We can therefore rewrite the iterative step of the EMML algorithm as

$$x_j^{k+1} = x_j^k - s_j^{-1} x_j^k \nabla f(x^k)_j. \tag{16.8}$$

We see that the iterative step depends not only on the negative of the gradient at x^k, but on the values x_j^k themselves. The closer x_j^k is to zero, the smaller the step, in order to keep the iterates positive.

For $f(x) = KL(b, Ax)$, the functions $f_i(x)$ are

$$f_i(x) = KL(b_i, (Ax)_i).$$

Therefore, a block-iterative version of the EMML iteration, called the BI-EMML algorithm [34], has the iterative step

$$x_j^{k+1} = x_j^k - s_j^{-1} x_j^k \nabla^n f(x^k)_j, \tag{16.9}$$

which can be written as

$$x_j^{k+1} = (1 - s_j^{-1} s_{nj}) x_j^k + x_j^k s_j^{-1} \sum_{i \in B_n} A_{ij} \frac{b_i}{(Ax^k)_i}, \tag{16.10}$$

using

$$s_{nj} = \sum_{i \in B_n} A_{ij}.$$

In the *consistent case*, in which there are non-negative x with $Ax = b$, the BI-EMML algorithm converges to such an x, for any positive starting vector, and any choice of blocks. It is not known to which non-negative solution it converges, however, nor how the limit depends on x^0 and the choice of blocks. Moreover, the BI-EMML algorithm does not necessarily converge faster than the original EMML algorithm.

Note that the iterative step given in Equation (16.10) involves relaxation, in which x^{k+1} includes some fraction of the current x^k. It was pointed out in [34] that this fraction can be unnecessarily large, and that the BI-EMML algorithm can be accelerated by rescaling, that is, by using the iterative step

$$x_j^{k+1} = (1 - m_n^{-1} s_j^{-1} s_{nj}) x_j^k + x_j^k m_n^{-1} s_j^{-1} \sum_{i \in B_n} A_{ij} \frac{b_i}{(Ax^k)_i}, \tag{16.11}$$

with

$$m_n = \max_j \{s_j^{-1} s_{nj}\}.$$

This iterative algorithm is the *rescaled* block-iterative EMML (RBI-EMML). Simulation studies have shown that this rescaling can accelerate convergence by roughly a factor of N, the number of blocks used.

When $N = I$ and the blocks B_n contain only a single member, denoted n, the RBI-EMML has the iterative step

$$x_j^{k+1} = (1 - m_n^{-1} s_j^{-1} A_{nj}) x_j^k + x_j^k m_n^{-1} s_j^{-1} A_{nj} \frac{b_n}{(Ax^k)_n}. \tag{16.12}$$

This is the EM-MART algorithm [34], analogous to the MART, but simpler to implement.

16.3.2 The SMART Algorithm

The SMART algorithm minimizes the function $f(x) = KL(Ax, b)$ over non-negative vectors x. We can therefore describe the iterative step of the SMART algorithm this way:

$$\log x_j^{k+1} = \log x_j^k - s_j^{-1} \nabla f(x^k)_j, \tag{16.13}$$

so that

$$x_j^{k+1} = x_j^k \exp\Big(- s_j^{-1} \nabla f(x^k)_j \Big). \tag{16.14}$$

For $f(x) = KL(Ax, b)$, the functions $f_i(x)$ are

$$f_i(x) = KL((Ax)_i, b_i).$$

Therefore, a block-iterative version of the SMART iteration, called the BI-SMART algorithm [68, 62, 34], has the iterative step

$$x_j^{k+1} = x_j^k \exp\Big(- s_j^{-1} \nabla^n f(x^k)_j \Big), \tag{16.15}$$

which can be written as

$$x_j^{k+1} = x_j^k \exp\Big(s_j^{-1} \sum_{i \in B_n} A_{ij} \log \frac{b_i}{(Ax^k)_i} \Big). \tag{16.16}$$

In the *consistent case*, in which there are non-negative x with $Ax = b$, the BI-SMART algorithm converges to such an x, for any positive starting vector, and any choice of blocks. Furthermore, the solution to which it converges is the one for which the cross-entropy $\sum_{j=1}^J s_j KL(x_j, x_j^0)$ is minimized, for all choice of blocks. As with the BI-EMML, however, the BI-SMART does not necessarily converge faster than the original SMART algorithm.

The iterative step given in Equation (16.16) can be written as

$$\log x_j^{k+1} = \log x_j^k + \Big(s_j^{-1} \sum_{i \in B_n} A_{ij} \log \frac{b_i}{(Ax^k)_i} \Big), \tag{16.17}$$

so that

$$\log x_j^{k+1} = (1 - s_j^{-1} s_{nj}) \log x_j^k + \Big(s_j^{-1} \sum_{i \in B_n} A_{ij} \log \big[x_j^k \frac{b_i}{(Ax^k)_i} \big] \Big). \tag{16.18}$$

From Equation (16.18) we see that the BI-SMART involves relaxation, in which $\log x_j^{k+1}$ includes some fraction of the current $\log x_j^k$. As with the BI-EMML, this fraction can be unnecessarily large, and the BI-SMART algorithm can be accelerated by rescaling, that is, by using the iterative step

$$\log x_j^{k+1} = (1 - m_n^{-1} s_j^{-1} s_{nj}) \log x_j^k + \Big(m_n^{-1} s_j^{-1} \sum_{i \in B_n} A_{ij} \log \big[x_j^k \frac{b_i}{(Ax^k)_i} \big] \Big). \tag{16.19}$$

This iterative algorithm is the *rescaled* block-iterative SMART (RBI-SMART). Simulation studies have shown that, in this case also, this rescaling can accelerate convergence by roughly a factor of N.

When $N = I$ and each block B_n contains only a single member, denoted n, the RBI-SMART has the iterative step

$$x_j^{k+1} = x_j^k \exp\Big(m_n^{-1} s_j^{-1} A_{nj} \log \frac{b_n}{(Ax^k)_n}\Big), \tag{16.20}$$

so that

$$x_j^{k+1} = x_j^k \Big(\frac{b_n}{(Ax^k)_n}\Big)^{m_n^{-1} s_j^{-1} A_{nj}}. \tag{16.21}$$

This is the (rescaled) MART algorithm.

16.4 Exercises

16.1 *Apply the gradient form of the Karush-Kuhn-Tucker Theorem to the two convex programming problems solved by the SMART and EMML algorithms, respectively.*

16.2 *Use the previous exercise to show that, in both cases, if there does not exist a non-negative solution of $Ax = b$, and A and every matrix obtained from A by deleting columns has full rank, then the solution vector x^* has at most $I - 1$ non-zero entries.*

Chapter 17

Sequential Unconstrained Minimization Algorithms

In this chapter we consider an approach to optimization in which the original problem is replaced by a series of simpler problems. This approach can be particularly effective for constrained optimization. Suppose, for example, that we want to minimize $f(x)$, subject to the constraint that x lie within a set C. At the kth step of the iteration we minimize the function $G_k(x) = f(x) + g_k(x)$, with no additional restrictions on x, to get the vector x^k, where the functions $g_k(x)$ are related to the set C in some way. In practice, minimizing $G_k(x)$ may require iteration, but we will not deal with that issue here. In the best case, the sequence $\{x^k\}$ will converge to the solution to the original problem.

17.1 Introduction

In many inverse problems, we have measured data pertaining to the object x, which may be, for example, a vectorized image, as well as prior information about x, such as that its entries are nonnegative. Tomographic imaging is a good example. We want to find an estimate of x that is (more or less) consistent with the data, as well as conforming to the prior constraints. The measured data and prior information are usually not sufficient to determine a unique x and some form of optimization is performed. For example, we may seek the image x for which the entropy is maximized, or a minimum-norm least-squares solution.

There are many well-known methods for minimizing a function $f : R^J \rightarrow R$; we can use the Newton-Raphson algorithm or any of its several approximations, or nonlinear conjugate-gradient algorithms, such as the Fletcher-Reeves, Polak-Ribiere, or Hestenes-Stiefel methods. When

the problem is to minimize the function $f(x)$, subject to constraints on the variable x, the problem becomes much more difficult. For such constrained minimization, we can employ *sequential unconstrained minimization algorithms* [85].

We assume that $f : R^J \to (-\infty, +\infty]$ is a continuous function. Our objective is to minimize $f(x)$ over x in some given closed nonempty set C. At the kth step of a sequential unconstrained minimization algorithm we minimize a function $G_k(x)$ to get the vector x^k. We shall assume throughout that a global minimizer x^k exists for each k. The existence of these minimizers can be established, once additional conditions, such as convexity, are placed on the functions $G_k(x)$; see, for example, Fiacco and McCormick [85], p.95. We shall consider briefly the issue of computing the x^k.

In the best case, the sequence $\{x^k\}$ converges to a constrained minimizer of the original objective function $f(x)$. Obviously, the functions $G_k(x)$ must involve both the function $f(x)$ and the set C. Those methods for which each x^k is *feasible*, that is, each x^k is in C, are called *interior-point* methods, while those for which only the limit of the sequence is in C are called *exterior-point* methods. Barrier-function algorithms are typically interior-point methods, while penalty-function algorithms are exterior-point methods. The purpose of this chapter is to present a fairly broad class of sequential unconstrained minimization algorithms, which we call SUMMA [46]. The SUMMA include both barrier- and penalty-function algorithms, as well as proximity-function methods of Teboulle and Censor and Zenios, and the simultaneous multiplicative algebraic reconstruction technique (SMART).

The sequential unconstrained minimization algorithms (SUMMA) we present here use functions of the form

$$G_k(x) = f(x) + g_k(x), \tag{17.1}$$

with the auxiliary functions $g_k(x)$ chosen so that

$$0 \leq g_{k+1}(x) \leq G_k(x) - G_k(x^k), \tag{17.2}$$

for $k = 1, 2, ...$ We assume throughout that there exists $\hat{x}$ minimizing the function $f(x)$ over x in C. Our main results are that the sequence $\{f(x^k)\}$ is monotonically decreasing to $f(\hat{x})$, and, subject to certain conditions on the function $f(x)$, the sequence $\{x^k\}$ converges to a feasible x^* with $f(x^*) = f(\hat{x})$.

We begin with a brief review of several types of sequential unconstrained minimization methods, including those mentioned previously. Then we state and prove the convergence results for the SUMMA. Finally, we show that each of these methods reviewed previously is a particular case of the SUMMA.

17.2 Barrier-Function Methods (I)

Let $b(x) : R^J \rightarrow (-\infty, +\infty]$ 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 the closed set $C = \overline{D}$, the closure of D. In the barrier-function method, we minimize

$$f(x) + \frac{1}{k} b(x) \tag{17.3}$$

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 $b(x) \rightarrow +\infty$ as x approaches the boundary of D, so not only is x^k prevented from leaving D, it is discouraged from approaching the boundary.

17.2.1 Examples of Barrier Functions

Consider the convex programming (CP) problem of minimizing the convex function $f : R^J \rightarrow R$, subject to $g_i(x) \leq 0$, where each $g_i : R^J \rightarrow R$ 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.

The Logarithmic Barrier Function

A suitable barrier function is the *logarithmic barrier function*

$$b(x) = \epsilon \Big(- \sum_{i=1}^{I} \log(-g_i(x)) \Big), \tag{17.4}$$

for some $\epsilon > 0$. 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.

The Inverse Barrier Function

Another suitable barrier function is the *inverse barrier function*

$$b(x) = \epsilon \sum_{i=1}^{I} \frac{-1}{g_i(x)}, \tag{17.5}$$

defined for those x in D.

In both examples, if ϵ is too large, the minimization pays too much attention to $b(x)$, and not enough to $f(x)$, forcing the $g_i(x)$ to be large negative numbers. For that reason, we take ϵ small. By letting $\epsilon \to 0$, we obtain an iterative method for solving the constrained minimization problem.

An Illustration

We minimize the function $f(u,v) = u^2 + v^2$, subject to the constraint that $u + v \geq 1$. The constraint is then written $g(u,v) = 1 - (u+v) \leq 0$. We use the logarithmic barrier. The vector $x^k = (u_k, v_k)$ minimizing the function

$$G_k(x) = u^2 + v^2 - \frac{1}{k}\log(u + v - 1)$$

has entries

$$u_k = v_k = \frac{1}{4} + \frac{1}{4}\sqrt{1 + \frac{4}{k}}.$$

Notice that $u_k + v_k > 1$, 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.

17.3 Penalty-Function Methods (I)

Instead of minimizing a function $f(x)$ over x in R^J, we sometimes want to minimize a *penalized* version, $f(x) + p(x)$. As with barrier-function methods, the new function $f(x) + p(x)$ may be the function we really want to minimize, and we still need to find a method for doing this. In other cases, it is $f(x)$ that we wish to minimize, and the inclusion of the term $p(x)$ occurs only in the iterative steps of the algorithm. As we shall see, under conditions to be specified later, the penalty-function method can be used to minimize a continuous function $f(x)$ over the nonempty set of minimizers of another continuous function $p(x)$.

17.3.1 Imposing Constraints

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 prove 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.

17.3.2 Examples of Penalty Functions

Consider the CP problem. We wish to minimize the convex function $f(x)$ over all x for which the convex functions $g_i(x) \leq 0$, for $i = 1, ..., I$.

The Absolute-Value Penalty Function

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

$$p(x) = \sum_{i=1}^{I} g_i^+(x). \tag{17.6}$$

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

$$f(x) + kp(x), \tag{17.7}$$

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.

The Courant-Beltrami Penalty Function

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

$$p(x) = \sum_{i=1}^{I} [g_i^+(x)]^2. \tag{17.8}$$

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

$$p(x) = \frac{1}{2} \sum_{i=1}^{I} (g_i(x))^2. \tag{17.9}$$

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 [89]. We survey several instances in which we would want to use a penalized objective function.

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

$$\frac{1}{2}\|Ax - b\|_2^2 + \frac{\epsilon}{2}\|x - z\|_2^2, \tag{17.10}$$

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*.

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*

$$KL(y, Px) = \sum_{i=1}^{I} \Big(y_i \log \frac{y_i}{(Px)_i} + (Px)_i - y_i \Big), \tag{17.11}$$

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 [111, 148, 112, 31].

A similar problem involves minimizing the function $KL(Px, y)$. Once again, noisy results can be avoided by including a penalty term, such as $p(x) = \epsilon KL(x, z)$ [31].

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,

$$f(x) + \sum_{i=1}^{I} \lambda_i g_i(x) = f(x) + p(x); \tag{17.12}$$

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.

Moreau's Proximity-Function Method

The Moreau envelope of the function f is the function

$$m_f(z) = \inf_x \Big\{ f(x) + \frac{1}{2}\|x - z\|_2^2 \Big\}, \tag{17.13}$$

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 [133]). In similar fashion, we can define $m_{f^*} z$ and $\text{prox}_{f^*} z$, where $f^*(z)$ denotes the function conjugate to f.

Proposition 17.1 *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. As we shall see, this leads to the proximal minimization algorithm to be discussed below.

17.3.3 The Roles Penalty Functions Play

From the examples just surveyed, we can distinguish several distinct roles that penalty functions can play.

Impose Constraints

The first role is to penalize violations of constraints, as part of sequential minimization, or even to turn a constrained minimization into an equivalent unconstrained one: the Absolute-Value and Courant-Beltrami penalty functions penalize violations of the constraints $g_i(x) \leq 0$, while Quadratic-Loss penalty function penalizes violations of the constraints $g_i(x) = 0$. The augmented objective functions $f(x) + kp(x)$ now become part of a sequential unconstrained minimization method. It is sometimes possible for $f(x)$ and $f(x) + p(x)$ to have the same minimizers, or for constrained minimizers of $f(x)$ to be the same as unconstrained minimizers of $f(x) + p(x)$, as happens with the Lagrangian in the CP problem.

Regularization

The second role is regularization: in the least-squares problem, the main purpose for adding the norm-squared penalty function in Equation (17.10) is to reduce sensitivity to noise in the entries of the vector b. Also, regularization will usually turn a problem with multiple solutions into one with a unique solution.

Incorporate Prior Information

The third role is to incorporate prior information: when $Ax = b$ is underdetermined, using the penalty function $\epsilon\|x - z\|_2^2$ and letting $\epsilon \to 0$ encourages the solution to be close to the prior estimate z.

Simplify Calculations

A fourth role that penalty functions can play is to simplify calculation: in the case of cross-entropy minimization, adding the penalty functions $KL(z, x)$ and $KL(x, z)$ to the objective functions $KL(y, Px)$ and $KL(Px, y)$, respectively, regularizes the minimization problem. But, as we shall see later, the SMART algorithm minimizes $KL(Px, y)$ by using a sequential approach, in which each minimizer x^k can be calculated in closed form.

Sequential Unconstrained Minimization

More generally, a fifth role for penalty functions is as part of sequential minimization. Here the goal is to replace one computationally difficult minimization with a sequence of simpler ones. Clearly, one reason for the difficulty can be that the original problem is constrained, and the sequential approach uses a series of unconstrained minimizations, penalizing violations of the constraints through the penalty function. However, there are other instances in which the sequential approach serves to simplify the calculations, not to remove constraints, but, perhaps, to replace a nondifferentiable objective function with a differentiable one, or a sequence of differentiable ones, as in Moreau's method.

17.4 Proximity-Function Minimization (I)

Let $f : R^J \rightarrow (-\infty, +\infty]$ 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 $\hat{x}$. The corresponding *Bregman distance* $D_h(x, z)$ is defined for x in D and z in int D by

$$D_h(x, z) = h(x) - h(z) - \langle \nabla h(z), x - z \rangle. \tag{17.14}$$

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 $C = \overline{D}$.

17.4.1 Proximal Minimization Algorithm

At the kth step of the *proximal minimization algorithm* (PMA) [39], we minimize the function

$$G_k(x) = f(x) + D_h(x, x^{k-1}), \tag{17.15}$$

to get x^k. The function

$$g_k(x) = D_h(x, x^{k-1}) \tag{17.16}$$

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

17.4.2 The Method of Auslander and Teboulle

In [6] Auslander and Teboulle consider an iterative method similar to the PMA, in which, at the kth step, one minimizes the function

$$F_k(x) = f(x) + d(x, x^{k-1}) \tag{17.17}$$

to get x^k. Their 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 D, with $H(a, a) = 0$ and

$$\langle \nabla_1 d(b, a), c - b \rangle \leq H(c, a) - H(c, b), \tag{17.18}$$

for all c in D. The notation $\nabla_1 d(x, y)$ denotes the gradient with respect to the vector variable x.

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

$$\langle \nabla_1 d(b, a), c - b \rangle = D_h(c, a) - D_h(c, b) - D_h(b, a) \tag{17.19}$$

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 [6].

17.5 The Simultaneous MART (SMART) (I)

Our next example is the simultaneous multiplicative algebraic reconstruction technique (SMART). For $a > 0$ and $b > 0$, the Kullback-Leibler distance, $KL(a, b)$, is defined as

$$KL(a, b) = a \log \frac{a}{b} + b - a. \tag{17.20}$$

In addition, $KL(0, 0) = 0$, $KL(a, 0 = +\infty$ and $KL(0, b) = b$. The KL distance is then extended to nonnegative vectors coordinate-wise.

17.5.1 The SMART Iteration

The SMART 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 $s_j = \sum_{i=1}^{I} P_{ij} > 0$. For notational convenience, we shall assume that the system $y = Px$ has been normalized so that $s_j = 1$, for each j. 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

$$x_j^k = x_j^{k-1} \exp\Big(\sum_{i=1}^{I} P_{ij} \log(y_i / (Px^{k-1})_i) \Big). \tag{17.21}$$

17.5.2 SMART as Alternating Minimization

In [31] 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

$$r(x)_{ij} = x_j P_{ij} y_i / (Px)_i, \tag{17.22}$$

and

$$q(x)_{ij} = x_j P_{ij}. \tag{17.23}$$

The iterative step of the SMART is to minimize the function $KL(q(x), r(x^{k-1}))$ to get $x = x^k$. Note that $f(x) = KL(q(x), r(x))$.

Now we establish the basic results for the SUMMA.

17.6 Convergence Theorems for SUMMA

At the kth step of the SUMMA we minimize the function $G_k(x)$ to get x^k. In practice, of course, this minimization may need to be performed iteratively; we shall not address this issue here, and shall assume that x^k can be computed. We make the following additional assumptions.

Assumption 1: The functions $g_k(x)$ are finite-valued and continuous on a set D in R^J, with $C = \overline{D}$.

Assumption 2: There is $\hat{x}$ in C with $f(\hat{x}) \leq f(x)$, for all x in C.

Assumption 3: The functions $g_k(x)$ satisfy the inequality in (17.2); that is,

$$0 \leq g_k(x) \leq G_{k-1}(x) - G_{k-1}(x^{k-1}),$$

for $k = 2, 3, \ldots$. Consequently,

$$g_k(x^{k-1}) = 0.$$

Assumption 4: There is a real number α with

$$\alpha \leq f(x),$$

for all x in R^J.

Assumption 5: Each x^k is in D.

Using these assumptions, we can conclude several things about the sequence $\{x^k\}$.

Proposition 17.2 *The sequence $\{f(x^k)\}$ is decreasing, and the sequence $\{g_k(x^k)\}$ converges to zero.*

Proof: We have

$$f(x^{k+1})+g_{k+1}(x^{k+1}) = G_{k+1}(x^{k+1}) \leq G_{k+1}(x^k) = f(x^k)+g_{k+1}(x^k) = f(x^k).$$

Therefore,

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

Since the sequence $\{f(x^k)\}$ is decreasing and bounded below by α, the difference sequence must converge to zero. Therefore, the sequence $\{g_k(x^k)\}$ converges to zero. ∎

Theorem 17.1 *The sequence $\{f(x^k)\}$ converges to $f(\hat{x})$.*

Proof: Suppose that there is $\delta > 0$ with

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

for all k. Since $\hat{x}$ is in C, there is z in D with

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

for all k. From

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

we have

$$g_k(z) - g_{k+1}(z) \geq f(x^k) + g_k(x^k) - f(z) \geq f(x^k) - f(z) \geq \frac{\delta}{2} > 0.$$

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

Theorem 17.2 *Let the restriction of $f(x)$ to x in C have bounded level sets. Then the sequence $\{x^k\}$ is bounded, and $f(x^*) = f(\hat{x})$, for any cluster point x^*. If $\hat{x}$ is unique, $x^* = \hat{x}$ and $\{x^k\} \to \hat{x}$.*

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

Corollary 17.1 *Let $f(x)$ be closed, proper and convex. If $\hat{x}$ is unique, 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 [133] that every level set of $g(x)$ is bounded, so that the sequence $\{x^k\}$ is bounded. ∎

If $\hat{x}$ is not unique, we may still be able to prove convergence of the sequence $\{x^k\}$, for particular cases of SUMMA, as we shall see shortly.

17.7 Barrier-Function Methods (II)

We return now to the barrier-function methods, to show that they are particular cases of the SUMMA. The iterative step of the barrier-function method can be formulated as follows: minimize

$$f(x) + [(k-1)f(x) + b(x)] \tag{17.24}$$

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

$$(k-1)f(x) + b(x) \tag{17.25}$$

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

$$g_k(x) = (k-1)f(x) + b(x) - (k-1)f(x^{k-1}) - b(x^{k-1}) \tag{17.26}$$

is nonnegative, and x^k minimizes the function

$$G_k(x) = f(x) + g_k(x). \tag{17.27}$$

From

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

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 (17.2). This shows that the barrier-function method is a particular case of SUMMA.

The goal is to minimize the objective function $f(x)$, over x in the closed set $C = \overline{D}$, the closure of D. In the barrier-function method, we minimize

$$f(x) + \frac{1}{k}b(x) \tag{17.28}$$

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

From the results for SUMMA, we conclude that $\{f(x^k)\}$ is decreasing to $f(\hat{x})$, and that $\{g_k(x^k)\}$ converges to zero. From the nonnegativity 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 17.3 *Let $f : R^J \to (-\infty, +\infty]$ be a continuous function. Let $b(x) : R^J \to (0, +\infty]$ 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 $\hat{x}$.*

Each step of the barrier method requires the minimization of 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 [132] and Nesterov and Nemirovski [124].

17.8 Penalty-Function Methods (II)

Let M be the non-empty closed set of all x for which the continuous function $p(x)$ attains its minimum value; this value need not be zero. Now we consider the problem of minimizing a continuous function $f(x) : R^J \to (-\infty, +\infty]$ over the closed set M. We assume that the constrained minimum of $f(x)$ is attained at some vector $\hat{x}$ in M. We also assume that the function $p(x)$ has bounded level sets, that is, for all $\gamma \geq 0$, the set $\{x|p(x) \leq \gamma\}$ is bounded.

For $k = 1, 2, ...$, let x^k be a minimizer of the function $f(x) + kp(x)$. As we shall see, we can formulate this penalty-function algorithm as a barrier-function iteration.

17.8.1 Penalty-Function Methods as Barrier-Function Methods

In order to relate penalty-function methods to barrier-function methods, we note that minimizing $f(x)+kp(x)$ 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 $f(x)$.

Now our Assumption 2 simply says that there is a vector $\hat{x}$ at which $p(x)$ attains its minimum; so M is not empty. From our discussion of barrier-function methods, we know that the sequence $\{p(x^k)\}$ is decreasing to a limit $\hat{p} \geq p(\hat{x})$ and the sequence $\{f(x^k)\}$ is increasing. Since $p(x)$ has bounded level sets, the sequence $\{x^k\}$ is bounded; let x^* be an arbitrary cluster point. We then have $p(x^*) = \hat{p}$. It may seem odd that we are trying to minimize $f(x)$ over the set M using a sequence $\{x^k\}$ with $\{f(x^k)\}$ increasing, but remember that these x^k are not in M.

We now show that $f(x^*) = f(\hat{x})$. This does not follow from our previous discussion of barrier-function methods.

Let $s(x) = p(x) - p(\hat{x})$, so that $s(x) \geq 0$ and $s(\hat{x}) = 0$. For each k, let

$$T_k(x) = f(x) + ks(x) = f(x) + kp(x) - kp(\hat{x}).$$

Then x^k minimizes $T_k(x)$.

Lemma 17.1 *The sequence $\{T_k(x^k)\}$ is increasing to some limit $\gamma \leq f(\hat{x})$.*

Proof: Because the penalty function $s(x)$ is nonnegative, we have

$$T_k(x^k) \leq T_k(x^{k+1}) \leq T_k(x^{k+1}) + s(x^{k+1}) = T_{k+1}(x^{k+1}).$$

We also have

$$f(\hat{x}) = f(\hat{x}) + ks(\hat{x}) = T_k(\hat{x}) \geq T_k(x^k),$$

for all k. ∎

Lemma 17.2 *For all cluster points x^* of $\{x^k\}$ we have $s(x^*) = 0$, so that $p(x^*) = p(\hat{x})$ and x^* is in M.*

Proof: For each k we have

$$\alpha + ks(x^k) \leq f(x^k) + ks(x^k) = T_k(x^k) \leq f(\hat{x}),$$

so that

$$0 \leq ks(x^k) \leq f(\hat{x}) - \alpha,$$

for all k. It follows that $\{s(x^k)\}$ converges to zero. By the continuity of $s(x)$, we conclude that $s(x^*) = 0$, so x^* is in M. ∎

Lemma 17.3 *For all cluster points x^* of the sequence $\{x^k\}$ we have $f(x^*) = f(\hat{x})$, so x^* minimizes $f(x)$ over x in M.*

Proof: Let $\{x^{k_n}\} \to x^*$. We have

$$f(x^*) = f(x^*) + s(x^*) = \lim_{n \to +\infty} \Big(f(x^{k_n}) + s(x^{k_n}) \Big)$$

$$\leq \lim_{n\to+\infty} \Big(f(x^{k_n}) + k_n s(x^{k_n}) \Big) \leq f(\hat{x}).$$

Since x^* is in M, it follows that $f(x^*) = f(\hat{x})$. ∎

To assert that the sequence $\{x^k\}$ itself converges, we would need to make additional assumptions. For example, if the minimizer of $f(x)$ over x in M is unique, then the sequence $\{x^k\}$ has $\hat{x}$ for its only cluster point, so must converge to $\hat{x}$.

The following theorem summarizes what we have shown with regard to penalty-function methods.

Theorem 17.4 *Let $f : R^J \to (-\infty, +\infty]$ be a continuous function. Let $p(x) : R^J \to R$ be a continuous function, with bounded level sets, and M the set of all $\tilde{x}$ such that $p(\tilde{x}) \leq p(x)$ for all x in R^J. Let $\hat{x}$ in M minimize $f(\tilde{x})$ over all $\tilde{x}$ in M. For each positive integer k, let x^k minimize the function $f(x) + kp(x)$. Then the sequence $\{f(x^k)\}$ is monotonically increasing to the limit $f(\hat{x})$, and the sequence $\{p(x^k)\}$ is decreasing to $p(\hat{x})$. If $\hat{x}$ is unique, which happens, for example, if $f(x)$ is strictly convex on M, then the sequence $\{x^k\}$ converges to $\hat{x}$.*

17.9 The Proximal Minimization Algorithm (II)

We show now that Assumption 3 holds, so that the PMA is a particular case of the SUMMA. We remind the reader that $f(x)$ is now assumed to be convex and differentiable, so that the Bregman distance $D_f(x, z)$ is defined and nonnegative, for all x in D and z in int$D$.

Lemma 17.4 *For each k we have*

$$G_k(x) = G_k(x^k) + D_f(x, x^k) + D_h(x, x^k). \tag{17.29}$$

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

$$0 = \nabla f(x^k) + \nabla h(x^k) - \nabla h(x^{k-1}). \tag{17.30}$$

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 (17.30), and use the definition of Bregman distances. ∎

It follows from Lemma 17.4 that

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

so Assumption 3 holds.

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

$$G_k(\hat{x}) = f(\hat{x}) + D_h(\hat{x}, x^{k-1}). \tag{17.31}$$

From Equation (17.29) we have

$$G_k(\hat{x}) = G_k(x^k) + D_f(\hat{x}, x^k) + D_h(\hat{x}, x^k), \tag{17.32}$$

so that

$$G_k(\hat{x}) = f(x^k) + D_h(x^k, x^{k-1}) + D_f(\hat{x}, x^k) + D_h(\hat{x}, x^k). \tag{17.33}$$

Therefore,

$$\begin{gathered} D_h(\hat{x}, x^{k-1}) - D_h(\hat{x}, x^k) = \\ f(x^k) - f(\hat{x}) + D_h(x^k, x^{k-1}) + D_f(\hat{x}, x^k). \end{gathered} \tag{17.34}$$

It follows that the sequence $\{D_h(\hat{x}, x^k)\}$ is decreasing and that the sequence $\{D_f(\hat{x}, x^k)\}$ converges to 0. 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.

Let x^* be an arbitrary cluster point, with $\{x^{k_n}\} \to x^*$. If $\hat{x}$ is not in int 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 $f : R^J \to (-\infty, +\infty]$ 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 17.5 *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 17.6 *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 $f(x^*) = f(\hat{x})$.*

17.9.1 The Method of Auslander and Teboulle

The method of Auslander and Teboulle described in a previous section seems not to be a particular case of SUMMA. However, we can adapt the proof of Theorem 17.1 to prove the analogous result for their method. Once again, we assume that $f(\hat{x}) \leq f(x)$, for all x in C.

Theorem 17.7 *For $k = 2, 3, ...$, let x^k minimize the function*

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

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

Proof: First, we show that the sequence $\{f(x^k)\}$ is decreasing. We have

$$f(x^{k-1}) = F_k(x^{k-1}) \geq F_k(x^k) = f(x^k) + d(x^k, x^{k-1}),$$

from which we conclude 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 D 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.

17.10 The Simultaneous MART (II)

It follows from the identities established in [31] that the SMART can also be formulated as a particular case of the SUMMA.

17.10.1 The SMART as a Case of SUMMA

We show now that the SMART is a particular case of the SUMMA. The following lemma is helpful in that regard.

Lemma 17.5 *For any non-negative vectors x and z, with $z_+ = \sum_{j=1}^J z_j > 0$, we have*

$$KL(x, z) = KL(x_+, z_+) + KL(x, \frac{x_+}{z_+} z). \tag{17.35}$$

From the identities established for the SMART in [31], we know that the iterative step of SMART can be expressed as follows: minimize the function

$$G_k(x) = KL(Px, y) + KL(x, x^{k-1}) - KL(Px, Px^{k-1}) \tag{17.36}$$

to get x^k. According to Lemma 17.5, 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 R^J. Each x^k is a positive vector.

It was shown in [31] that

$$G_k(x) = G_k(x^k) + KL(x, x^k), \tag{17.37}$$

from which it follows immediately that Assumption 3 holds for the SMART.

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 [31] 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.

17.10.2 The SMART as a Case of the PMA

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

$$F(x) = \sum_{j=1}^{J} x_j \log x_j. \tag{17.38}$$

Then

$$D_F(x, z) = KL(x, z). \tag{17.39}$$

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

$$D_f(x, z) = KL(Px, Pz). \tag{17.40}$$

Lemma 17.6 $D_F(x, z) \geq D_f(x, z)$.

Proof: We have

$$\begin{aligned} D_F(x, z) &\geq \sum_{j=1}^{J} KL(x_j, z_j) \geq \sum_{j=1}^{J} \sum_{i=1}^{I} KL(P_{ij}x_j, P_{ij}z_j) \\ &\geq \sum_{i=1}^{I} KL((Px)_i, (Pz)_i) = KL(Px, Pz). \end{aligned} \tag{17.41}$$

∎

Then 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

$$f(x) + D_h(x, x^{k-1}). \tag{17.42}$$

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 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 (17.34) 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 17.8 *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.*

17.10.3 The EMML Algorithm

The *expectation maximization maximum likelihood* (EMML) algorithm minimizes the function $f(x) = KL(y, Px)$ over x in $\mathcal{X}$. In [44] the EMML algorithm and the SMART are developed in tandem to reveal how closely related these two methods are. There, the EMML algorithm is derived using alternating minimization, in which the vector x^k is the one for which the function $KL(r(x^{k-1}), q(x))$ is minimized. When we try to put the EMML into the framework of SUMMA, we find that x^k minimizes the function

$$G_k(x) = f(x) + KL(r(x^{k-1}), r(x)), \tag{17.43}$$

over all positive vectors x. However, the functions

$$g_k(x) = KL(r(x^{k-1}), r(x)) \tag{17.44}$$

appear not to satisfy the condition in (17.2). It does not appear to be true that the EMML is a particular case of SUMMA, even though it is true that $\{f(x^k)\}$ does converge monotonically to $f(\hat{x})$ and $\{x^k\}$ does converge to a nonnegative minimizer of $f(x)$. The obvious conjecture is that the EMML is an example of a wider class of sequential unconstrained minimization algorithms for which a nice theory of convergence still holds.

In the next section we present a variant of the SMART, designed to incorporate bounds on the entries of the vector x.

17.11 Minimizing $KL(Px, y)$ with upper and lower bounds on the vector x

Let $a_j < b_j$, for each j. Let $\mathcal{X}_{ab}$ be the set of all vectors x such that $a_j \leq x_j \leq b_j$, for each j. Now, we seek to minimize $f(x) = KL(Px, y)$, over all vectors x in $\mathcal{X} \cap \mathcal{X}_{ab}$. We let

$$F(x) = \sum_{j=1}^{J} \Big((x_j - a_j) \log(x_j - a_j) + (b_j - x_j) \log(b_j - x_j) \Big). \quad (17.45)$$

Then we have

$$D_F(x, z) = \sum_{j=1}^{J} \Big(KL(x_j - a_j, z_j - a_j) + KL(b_j - x_j, b_j - z_j) \Big), \quad (17.46)$$

and, as before,

$$D_f(x, z) = KL(Px, Pz). \quad (17.47)$$

Lemma 17.7 *For any $c > 0$, with $a \geq c$ and $b \geq c$, we have $KL(a - c, b - c) \geq KL(a, b)$.*

Proof: Let $g(c) = KL(a - c, b - c)$ and differentiate with respect to c, to obtain

$$g'(c) = \frac{a - c}{b - c} - 1 - \log(\frac{a - c}{b - c}) \geq 0. \quad (17.48)$$

We see then that the function $g(c)$ is increasing with c. ∎

As a corollary of Lemma 17.7, we have

Lemma 17.8 *Let $a = (a_1, ..., a_J)^T$, and x and z in $\mathcal{X}$ with $(Px)_i \geq (Pa)_i$, $(Pz)_i \geq (Pa)_i$, for each i. Then $KL(Px, Pz) \leq KL(Px - Pa, Pz - Pa)$.*

Lemma 17.9 *$D_F(x, z) \geq D_f(x, z)$.*

Proof: We can easily show that

$$D_F(x, z) \geq KL(Px - Pa, Pz - Pa) + KL(Pb - Px, Pb - Pz),$$

along the lines used previously. Then, from Lemma 17.8, we have

$$KL(Px - Pa, Pz - Pa) \geq KL(Px, Pz) = D_f(x, z).$$

∎

Once again, we let $h(x) = F(x) - f(x)$, which is finite on the closed convex set $\mathcal{X} \cap \mathcal{X}_{ab}$. At the kth step of this algorithm we minimize the function

$$f(x) + D_h(x, x^{k-1}) \tag{17.49}$$

to get x^k.

Solving for x_j^k, we obtain

$$x_j^{k+1} = \alpha_j^k a_j + (1 - \alpha_j^k) b_j, \tag{17.50}$$

where

$$(\alpha_j^k)^{-1} = 1 + \Big(\frac{x_j^{k-1} - a_j}{b_j - x_j^{k-1}}\Big) \exp\Big(\sum_{i=1}^{I} P_{ij} \log(y_i/(Px^{k-1})_i)\Big). \tag{17.51}$$

Since the restriction of $f(x)$ to $\mathcal{X} \cap \mathcal{X}_{ab}$ has bounded level sets, the sequence $\{x^k\}$ is bounded and has cluster points. If $\hat{x}$ is unique, then $\{x^k\} \to \hat{x}$.

This algorithm is closely related to those presented in [37].

17.12 Computation

As we noted previously, we do not address computational issues in any detail in this chapter. Nevertheless, it cannot be ignored that both Equation (17.21) for the SMART and Equations (17.50) and (17.51) for the generalized SMART provide easily calculated iterates, in contrast to other examples of SUMMA. At the same time, showing that these two algorithms are particular cases of SUMMA requires the introduction of functions $G_k(x)$ that appear to be quite ad hoc. The purpose of this section is to motivate these choices of $G_k(x)$ and to indicate how other analogous computationally tractable SUMMA iterative schemes may be derived.

17.12.1 Landweber's Algorithm

Suppose that A is a real I by J matrix and we wish to obtain a least-squares solution $\hat{x}$ of $Ax = b$ by minimizing the function

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

We know that

$$(A^T A)\hat{x} = A^T b, \tag{17.52}$$

so, in a sense, the problem is solved. However, in many applications, the dimensions I and J are quite large, perhaps in the tens of thousands, as in

some image reconstruction problems. Solving Equation (17.52), and even calculating A^TA, can be prohibitively expensive. In such cases, we turn to iterative methods, not necessarily to incorporate constraints on x, but to facilitate calculation. Landweber's algorithm is one such iterative method for calculating a least-squares solution.

The iterative step of Landweber's algorithm is

$$x^k = x^{k-1} - \gamma A^T(Ax^{k-1} - b). \tag{17.53}$$

The sequence $\{x^k\}$ converges to the least-squares solution closest to x^0, for any choice of γ in the interval $(0, 2/\rho(A^TA))$, where $\rho(A^TA)$, the spectral radius of A^TA, is its largest eigenvalue; this is a consequence of the Krasnoselskii-Mann Theorem (see, for example, [42]).

It is easy to verify that the x^k given by Equation (17.53) is the minimizer of the function

$$G_k(x) = \frac{1}{2}\|Ax - b\|^2 + \frac{1}{2\gamma}\|x - x^{k-1}\|^2 - \frac{1}{2}\|Ax - Ax^{k-1}\|^2, \tag{17.54}$$

that, for γ in the interval $(0, 1/\rho(A^TA))$, the iteration in Equation (17.53) is a particular case of SUMMA, and

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

The similarity between the $G_k(x)$ in Equation (17.54) and that in Equation (17.36) is not accidental and both are particular cases of a more general iterative scheme involving proximal minimization.

17.12.2 Extending the PMA

The proximal minimization algorithm (PMA) requires us to minimize the function $G_k(x)$ given by Equation (17.15) to get x^k. How x^k may be calculated was not addressed previously. Suppose, instead of minimizing $G_k(x)$ in Equation (17.15), we minimize

$$G_k(x) = f(x) + D_h(x, x^{k-1}) - D_f(x, x^{k-1}), \tag{17.55}$$

with the understanding that $f(x)$ is convex and

$$D_h(x, z) - D_f(x, z) \geq 0,$$

for all appropriate x and z. The next iterate x^k satisfies the equation

$$0 = \nabla h(x^k) - \nabla h(x^{k-1}) + \nabla f(x^{k-1}), \tag{17.56}$$

so that

$$\nabla h(x^k) = \nabla h(x^{k-1}) - \nabla f(x^{k-1}). \tag{17.57}$$

This iterative scheme is the *interior-point algorithm* (IPA) presented in [39]. If the function $h(x)$ is chosen carefully, then we can solve for x^k easily. The Landweber algorithm, the SMART, and the generalized SMART are all particular cases of this IPA.

Using Lemma 17.4, we can show that

$$G_k(x) - G_k(x^k) = \frac{1}{2\gamma} D_h(x, x^k), \tag{17.58}$$

for all appropriate x, so that the IPA is a particular case of SUMMA. We consider now several other examples.

If we let $h(x) = \frac{1}{2\gamma}\|x\|^2$ in Equation (17.55), the iteration becomes

$$x^k = x^{k-1} - \gamma \nabla f(x^{k-1}). \tag{17.59}$$

If, for example, the operator ∇f is L-Lipschitz continuous, that is,

$$\|\nabla f(x) - \nabla f(z)\| \leq L\|x - z\|,$$

then, for any γ in the interval $(0, 1/2L)$, we have

$$\frac{1}{2\gamma}\|x - z\|^2 \geq L\|x - z\|^2 \geq \langle \nabla f(x) - \nabla f(z), x - z\rangle$$

$$= D_f(x, z) + D_f(z, x) \geq D_f(x, z).$$

Therefore, this iteration is a particular case of SUMMA. It should be noted that, in this case, the Krasnoselskii-Mann Theorem gives convergence for any γ in the interval $(0, 2/L)$.

Finally, we consider what happens if we replace the Euclidean norm with that induced by the local geometry derived from f itself. More specifically, let us take

$$h(x) = \frac{1}{2} x^T \nabla^2 f(x^{k-1}) x,$$

so that

$$D_h(x, x^{k-1}) = \frac{1}{2}(x - x^{k-1})^T \nabla^2 f(x^{k-1})(x - x^{k-1}).$$

Then the IPA iterate x^k becomes

$$x^k = x^{k-1} - \nabla^2 f(x^{k-1})^{-1} \nabla f(x^{k-1}), \tag{17.60}$$

which is the Newton-Raphson iteration. Using the SUMMA framework to study the Newton-Raphson method is work in progress.

Algorithms such as Landweber's and SMART can be slow to converge. It is known that convergence can often be accelerated using incremental gradient (partial gradient, block-iterative, ordered-subset) methods. Using the SUMMA framework to study such incremental gradient methods as the algebraic reconstruction technique (ART), its multiplicative version (MART), and other block-iterative methods is also the subject of on-going work.

17.13 Connections with Karmarkar's Method

As related by Margaret Wright in [149], a revolution in mathematical programming took place around 1984. In that year Narenda Karmarkar discovered the first efficient polynomial-time algorithm for the linear programming problem [105]. Khachian's earlier polynomial-time algorithm for LP was too slow and conventional wisdom prior to 1984 was that the simplex method was "the only game in town" . It was known that, for certain peculiar LP problems, the complexity of the simplex method grew exponentially with the size of the problem, and obtaining a polynomial-time method for LP had been a goal for quite a while. However, for most problems, the popular simplex method was more than adequate. Soon after Karmarkar's result was made known, others discovered that there was a close connection between this method and earlier barrier-function approaches in nonlinear programming [90]. This discovery not only revived barrier-function methods, but established a link between linear and nonlinear programming, two areas that had historically been treated separately.

The primary LP problem in standard form is to minimize $c^T x$, subject to the conditions $Ax = b$ and $x \geq 0$. The barrier-function approach is to use a logarithmic barrier to enforce the condition $x \geq 0$, and then to use the primal-dual approach of Equation (9.39) to maintain the condition $Ax = b$. The function to be minimized, subject to $Ax = b$, is then

$$c^T x - \mu \sum_{j=1}^{J} \log x_j,$$

where $\mu > 0$ is the *barrier parameter*. When this minimization is performed using the primal-dual method described by Equation (9.39), and the NR iteration is begun at a feasible x^0, each subsequent x^k satisfies $Ax^k = b$. The limit of the NR iteration is x_μ. Under reasonable conditions, x_μ will converge to the solution of the LP problem, as $\mu \to 0$. This interior-point approach to solving the LP problem is essentially equivalent to Karmarkar's approach.

17.14 Exercises

17.1 *Prove Lemma 17.5.*

17.2 ([122], Ex. 16.1) *Use the logarithmic barrier method to minimize the function*

$$f(x, y) = x - 2y,$$

subject to the constraints

$$1 + x - y^2 \geq 0,$$

and

$$y \geq 0.$$

17.3 ([122], Ex. 16.5) *Use the quadratic-loss penalty method to minimize the function*

$$f(x, y) = -xy,$$

subject to the equality constraint

$$x + 2y - 4 = 0.$$

Chapter 18

Calculus of Variations

Up to now, we have been concerned with maximizing or minimizing real-valued functions of one or several variables, possibly subject to constraints. In this chapter, we consider another type of optimization problem, maximizing or minimizing *a function of functions*. The functions themselves we shall denote by simply $y = y(x)$, instead of the more common notation $y = f(x)$, and the function of functions will be denoted $J(y)$; in the calculus of variations, such functions of functions are called *functionals*. We then want to optimize $J(y)$ over a class of *admissible* functions $y(x)$. We shall focus on the case in which x is a single real variable, although there are situations in which the functions y are functions of several variables.

When we attempt to minimize a function $g(x_1, ..., x_N)$, we consider what happens to g when we perturb the values x_n to $x_n + \Delta x_n$. In order for $\mathbf{x} = (x_1, ..., x_N)$ to minimize g, it is necessary that

$$g(x_1 + \Delta x_1, ..., x_N + \Delta x_N) \geq g(x_1, ..., x_N),$$

for all perturbations $\Delta x_1, ..., \Delta x_N$. For differentiable g, this means that the gradient of g at $\mathbf{x}$ must be zero. In the calculus of variations, when we attempt to minimize $J(y)$, we need to consider what happens when we perturb the function y to a nearby *admissible* function, denoted $y + \Delta y$. In order for y to minimize $J(y)$, we need

$$J(y + \Delta y) \geq J(y),$$

for all Δy that make $y + \Delta y$ admissible. We end up with something analogous to a first derivative of J, which is then set to zero. The result is a differential equation, called the *Euler-Lagrange Equation*, which must be satisfied by the minimizing y.

18.1 Some Examples

In this section we present some of the more famous examples of problems from the calculus of variations.

18.1.1 The Shortest Distance

Among all the functions $y = y(x)$, defined for x in the interval $[0, 1]$, with $y(0) = 0$ and $y(1) = 1$, the straight-line function $y(x) = x$ has the shortest length. Assuming the functions are differentiable, the formula for the length of such curves is

$$J(y) = \int_0^1 \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx. \tag{18.1}$$

Therefore, we can say that the function $y(x) = x$ minimizes $J(y)$, over all such functions.

In this example, the functional $J(y)$ involves only the first derivative of $y = y(x)$ and has the form

$$J(y) = \int f(x, y(x), y'(x))dx, \tag{18.2}$$

where $f = f(u, v, w)$ is the function of three variables

$$f(u, v, w) = \sqrt{1 + w^2}. \tag{18.3}$$

In general, the functional $J(y)$ can come from almost any function $f(u, v, w)$. In fact, if higher derivatives of $y(x)$ are involved, the function f can be a function of more than three variables. In this chapter we shall confine our discussion to problems involving only the first derivative of $y(x)$.

18.1.2 The Brachistochrone Problem

Consider a frictionless wire connecting the two points $A = (0, 0)$ and $B = (1, 1)$; for convenience, the positive y-axis is downward. A metal ball rolls from point A to point B under the influence of gravity. What shape should the wire take in order to make the travel time of the ball the smallest? This famous problem, known as the *Brachistochrone Problem*, was posed in 1696 by Johann Bernoulli. This event is viewed as marking the beginning of the calculus of variations.

The velocity of the ball along the curve is $v = \frac{ds}{dt}$, where s denotes the arc-length. Therefore,

$$dt = \frac{ds}{v} = \frac{1}{v}\sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx.$$

Because the ball is falling under the influence of gravity only, the velocity it attains after falling from $(0,0)$ to (x,y) is the same as it would have attained had it fallen y units vertically; only the travel times are different. This is because the loss of potential energy is the same either way. The velocity attained after a vertical free fall of y units is $\sqrt{2gy}$. Therefore, we have

$$dt = \frac{\sqrt{1+\left(\frac{dy}{dx}\right)^2}dx}{\sqrt{2gy}}.$$

The travel time from A to B is therefore

$$J(y) = \frac{1}{\sqrt{2g}}\int_0^1 \sqrt{1+\left(\frac{dy}{dx}\right)^2}\frac{1}{\sqrt{y}}dx. \tag{18.4}$$

For this example, the function $f(u,v,w)$ is

$$f(u,v,w) = \frac{\sqrt{1+w^2}}{\sqrt{v}}. \tag{18.5}$$

18.1.3 Minimal Surface Area

Given a function $y = y(x)$ with $y(0) = 1$ and $y(1) = 0$, we imagine revolving this curve around the x-axis, to generate a surface of revolution. The functional $J(y)$ that we wish to minimize now is the surface area. Therefore, we have

$$J(y) = \int_0^1 y\sqrt{1+y'(x)^2}dx. \tag{18.6}$$

Now the function $f(u,v,w)$ is

$$f(u,v,w) = v\sqrt{1+w^2}. \tag{18.7}$$

18.1.4 The Maximum Area

Among all curves of length L connecting the points $(0,0)$ and $(1,0)$, find the one for which the area A of the region bounded by the curve and the x-axis is maximized. The length of the curve is given by

$$L = \int_0^1 \sqrt{1+y'(x)^2}dx, \tag{18.8}$$

and the area, assuming that $y(x) \geq 0$ for all x, is

$$A = \int_0^1 y(x)dx. \tag{18.9}$$

This problem is different from the previous ones, in that we seek to optimize a functional, subject to a second functional being held fixed. Such problems are called *problems with constraints*.

18.1.5 Maximizing Burg Entropy

The *Burg entropy* of a positive-valued function $y(x)$ on $[-\pi, \pi]$ is

$$BE(y) = \int_{-\pi}^{\pi} \log\Big(y(x)\Big)dx. \tag{18.10}$$

An important problem in signal processing is to maximize $BE(y)$, subject to

$$r_n = \int_{-\pi}^{\pi} y(x)e^{-inx}dx, \tag{18.11}$$

for $|n| \leq N$. The r_n are values of the Fourier transform of the function $y(x)$.

18.2 Comments on Notation

The functionals $J(y)$ that we shall consider in this chapter have the form

$$J(y) = \int f(x, y(x), y'(x))dx, \tag{18.12}$$

where $f = f(u, v, w)$ is some function of three real variables. It is common practice, in the calculus of variations literature, to speak of $f = f(x, y, y')$, rather than $f(u, v, w)$. Unfortunately, this leads to potentially confusing notation, such as when $\frac{\partial f}{\partial u}$ is written as $\frac{\partial f}{\partial x}$, which is not the same thing as the total derivative of $f(x, y(x), y'(x))$,

$$\frac{d}{dx}f(x, y(x), y'(x)) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}y'(x) + \frac{\partial f}{\partial y'}y''(x). \tag{18.13}$$

Using the notation of this chapter, Equation (18.13) becomes

$$\frac{d}{dx}f(x, y(x), y'(x)) = \frac{\partial f}{\partial u}(x, y(x), y'(x))+$$

$$\frac{\partial f}{\partial v}(x, y(x), y'(x))y'(x) + \frac{\partial f}{\partial w}(x, y(x), y'(x))y''(x). \tag{18.14}$$

The common notation forces us to view $f(x, y, y')$ both as a function of three unrelated variables, x, y, and y', and as $f(x, y(x), y'(x))$, a function of the single variable x.

For example, suppose that

$$f(u, v, w) = u^2 + v^3 + \sin w,$$

and

$$y(x) = 7x^2.$$

Then

$$f(x, y(x), y'(x)) = x^2 + (7x^2)^3 + \sin(14x), \tag{18.15}$$

$$\frac{\partial f}{\partial x}(x, y(x), y'(x)) = 2x, \tag{18.16}$$

and

$$\frac{d}{dx} f(x, y(x), y'(x)) = \frac{d}{dx}\Big(x^2 + (7x^2)^3 + \sin(14x)\Big)$$

$$= 2x + 3(7x^2)^2(14x) + 14\cos(14x). \tag{18.17}$$

18.3 The Euler-Lagrange Equation

In the problems we shall consider in this chapter, admissible functions are differentiable, with $y(x_1) = y_1$ and $y(x_2) = y_2$; that is, the graphs of the admissible functions pass through the end points (x_1, y_1) and (x_2, y_2). If $y = y(x)$ is one such function and $\eta(x)$ is a differentiable function with $\eta(x_1) = 0$ and $\eta(x_2) = 0$, then $y(x) + \epsilon\eta(x)$ is admissible, for all values of ϵ. For fixed admissible function $y = y(x)$, we define

$$J(\epsilon) = J(y(x) + \epsilon\eta(x)), \tag{18.18}$$

and force $J'(\epsilon) = 0$ at $\epsilon = 0$. The tricky part is calculating $J'(\epsilon)$.

Since $J(y(x) + \epsilon\eta(x))$ has the form

$$J(y(x) + \epsilon\eta(x)) = \int_{x_1}^{x_2} f(x, y(x) + \epsilon\eta(x), y'(x) + \epsilon\eta'(x))dx, \tag{18.19}$$

we obtain $J'(\epsilon)$ by differentiating under the integral sign.

Omitting the arguments, we have

$$J'(\epsilon) = \int_{x_1}^{x_2} \frac{\partial f}{\partial v}\eta + \frac{\partial f}{\partial w}\eta' dx. \tag{18.20}$$

Using integration by parts and $\eta(x_1) = \eta(x_2) = 0$, we have

$$\int_{x_1}^{x_2} \frac{\partial f}{\partial w}\eta' dx = -\int_{x_1}^{x_2} \frac{d}{dx}(\frac{\partial f}{\partial w})\eta dx. \tag{18.21}$$

Therefore, we have

$$J'(\epsilon) = \int_{x_1}^{x_2} \Big(\frac{\partial f}{\partial v} - \frac{d}{dx}(\frac{\partial f}{\partial w})\Big)\eta dx. \tag{18.22}$$

In order for $y = y(x)$ to be the optimal function, this integral must be zero for every appropriate choice of $\eta(x)$, when $\epsilon = 0$. It can be shown without too much trouble that this forces

$$\frac{\partial f}{\partial v} - \frac{d}{dx}(\frac{\partial f}{\partial w}) = 0. \tag{18.23}$$

Equation (18.23) is the *Euler-Lagrange Equation*.

For clarity, let us rewrite that Euler-Lagrange Equation using the arguments of the functions involved. Equation (18.23) is then

$$\frac{\partial f}{\partial v}(x, y(x), y'(x)) - \frac{d}{dx}\Big(\frac{\partial f}{\partial w}(x, y(x), y'(x))\Big) = 0. \tag{18.24}$$

18.4 Special Cases of the Euler-Lagrange Equation

The Euler-Lagrange Equation simplifies in certain special cases.

18.4.1 If f is independent of v

If the function $f(u, v, w)$ is independent of the variable v then the Euler-Lagrange Equation (18.24) becomes

$$\frac{\partial f}{\partial w}(x, y(x), y'(x)) = c, \tag{18.25}$$

for some constant c. If, in addition, the function $f(u, v, w)$ is a function of w alone, then so is $\frac{\partial f}{\partial w}$, from which we conclude from the Euler-Lagrange Equation that $y'(x)$ is constant.

18.4.2 If f is independent of u

Note that we can write

$$\frac{d}{dx} f(x, y(x), y'(x)) = \frac{\partial f}{\partial u}(x, y(x), y'(x))$$

$$+\frac{\partial f}{\partial v}(x, y(x), y'(x))y'(x) + \frac{\partial f}{\partial w}(x, y(x), y'(x))y''(x). \tag{18.26}$$

We also have

$$\frac{d}{dx}\Big(y'(x)\frac{\partial f}{\partial w}(x, y(x), y'(x))\Big) =$$

$$y'(x)\frac{d}{dx}\Big(\frac{\partial f}{\partial w}(x, y(x), y'(x))\Big) + y''(x)\frac{\partial f}{\partial w}(x, y(x), y'(x)).$$

(18.27)

Subtracting Equation (18.27) from Equation (18.26), we get

$$\frac{d}{dx}\Big(f(x,y(x),y'(x)) - y'(x)\frac{\partial f}{\partial w}(x,y(x),y'(x))\Big) =$$

$$\frac{\partial f}{\partial u}(x,y(x),y'(x)) + y'(x)\Big(\frac{\partial f}{\partial v} - \frac{d}{dx}\frac{\partial f}{\partial w}\Big)(x,y(x),y'(x)). \tag{18.28}$$

Now, using the Euler-Lagrange Equation, we see that Equation (18.28) reduces to

$$\frac{d}{dx}\Big(f(x,y(x),y'(x)) - y'(x)\frac{\partial f}{\partial w}(x,y(x),y'(x))\Big) = \frac{\partial f}{\partial u}(x,y(x),y'(x)). \tag{18.29}$$

If it is the case that $\frac{\partial f}{\partial u} = 0$, then equation (18.29) leads to

$$f(x,y(x),y'(x)) - y'(x)\frac{\partial f}{\partial w}(x,y(x),y'(x)) = c, \tag{18.30}$$

for some constant c.

18.5 Using the Euler-Lagrange Equation

We derive and solve the Euler-Lagrange Equation for each of the examples presented previously.

18.5.1 The Shortest Distance

In this case, we have

$$f(u,v,w) = \sqrt{1+w^2}, \tag{18.31}$$

so that

$$\frac{\partial f}{\partial v} = 0,$$

and

$$\frac{\partial f}{\partial u} = 0.$$

We conclude that $y'(x)$ is constant, so $y(x)$ is a straight line.

18.5.2 The Brachistochrone Problem

Equation (18.5) tells us that

$$f(u,v,w) = \frac{\sqrt{1+w^2}}{\sqrt{v}}. \tag{18.32}$$

Then, since

$$\frac{\partial f}{\partial u} = 0,$$

and

$$\frac{\partial f}{\partial w} = \frac{w}{\sqrt{1+w^2}\sqrt{v}},$$

Equation (18.30) tells us that

$$\frac{\sqrt{1+y'(x)^2}}{\sqrt{y(x)}} - y'(x)\frac{y'(x)}{\sqrt{1+y'(x)^2}\sqrt{y(x)}} = c. \tag{18.33}$$

Equivalently, we have

$$\sqrt{y(x)}\sqrt{1+y'(x)^2} = \sqrt{a}. \tag{18.34}$$

Solving for $y'(x)$, we get

$$y'(x) = \sqrt{\frac{a-y(x)}{y(x)}}. \tag{18.35}$$

Separating variables and integrating, using the substitution

$$y = a\sin^2\theta = \frac{a}{2}(1-\cos 2\theta),$$

we obtain

$$x = 2a\int \sin^2\theta d\theta = \frac{a}{2}(2\theta - \sin 2\theta) + k. \tag{18.36}$$

From this, we learn that the minimizing curve is a *cycloid*, that is, the path a point on a circle traces as the circle rolls.

There is an interesting connection, discussed by Simmons in [141] , between the brachistochrone problem and the refraction of light rays. Imagine a ray of light passing from the point $A = (0,a)$, with $a > 0$, to the point $B = (c,b)$, with $c > 0$ and $b < 0$. Suppose that the speed of light is v_1 above the x-axis, and $v_2 < v_1$ below the x-axis. The path consists of two straight lines, meeting at the point $(0,x)$. The total time for the journey is then

$$T(x) = \frac{\sqrt{a^2+x^2}}{v_1} + \frac{\sqrt{b^2+(c-x)^2}}{v_2}.$$

Fermat's Principle of Least Time says that the (apparent) path taken by the light ray will be the one for which x minimizes $T(x)$. From calculus, it follows that

$$\frac{x}{v_1\sqrt{a^2+x^2}} = \frac{c-x}{v_2\sqrt{b^2+(c-x)^2}},$$

and from geometry, we get *Snell's Law*:

$$\frac{\sin\alpha_1}{v_1} = \frac{\sin\alpha_2}{v_2},$$

where α_1 and α_2 denote the angles between the upper and lower parts of the path and the vertical, respectively.

Imagine now a stratified medium consisting of many horizontal layers, each with its own speed of light. The path taken by the light would be such that $\frac{\sin\alpha}{v}$ remains constant as the ray passes from one layer to the next. In the limit of infinitely many infinitely thin layers, the path taken by the light would satisfy the equation $\frac{\sin\alpha}{v} = \text{constant}$, with

$$\sin\alpha = \frac{1}{\sqrt{1+y'(x)^2}}.$$

As we have already seen, the velocity attained by the rolling ball is $v = \sqrt{2gy}$, so the equation to be satisfied by the path $y(x)$ is

$$\sqrt{2gy(x)}\sqrt{1+y'(x)^2} = \text{constant},$$

which is what we obtained from the Euler-Lagrange Equation.

18.5.3 Minimizing the Surface Area

For the problem of minimizing the surface area of a surface of revolution, the function $f(u,v,w)$ is

$$f(u,v,w) = v\sqrt{1+w^2}. \tag{18.37}$$

Once again, $\frac{\partial f}{\partial u} = 0$, so we have

$$\frac{y(x)y'(x)^2}{\sqrt{1+y'(x)^2}} - y(x)\sqrt{1+y'(x)^2} = c. \tag{18.38}$$

It follows that

$$y(x) = b\cosh\frac{x-a}{b}, \tag{18.39}$$

for appropriate a and b.

It is important to note that being a solution of the Euler-Lagrange Equation is a necessary condition for a differentiable function to be a solution to the original optimization problem, but it is not a sufficient condition. The optimal solution may not be a differentiable one, or there may be no optimal solution. In the case of minimum surface area, there may not be any function of the form in Equation (18.39) passing through the two given end points; see Chapter IV of Bliss [14] for details.

18.6 Problems with Constraints

We turn now to the problem of optimizing one functional, subject to a second functional being held constant. The basic technique is similar to ordinary optimization subject to constraints: we use Lagrange multipliers. We begin with a classic example.

18.6.1 The Isoperimetric Problem

A classic problem in the calculus of variations is the *Isoperimetric Problem*: find the curve of a fixed length that encloses the largest area. For concreteness, suppose the curve connects the two points $(0, 0)$ and $(1, 0)$ and is the graph of a function $y(x)$. The problem then is to maximize the area integral

$$\int_0^1 y(x)dx, \tag{18.40}$$

subject to the perimeter being held fixed, that is,

$$\int_0^1 \sqrt{1 + y'(x)^2}dx = P. \tag{18.41}$$

With

$$f(x, y(x), y'(x)) = y(x) + \lambda\sqrt{1 + y'(x)^2},$$

the Euler-Lagrange Equation becomes

$$\frac{d}{dx}\Big(\frac{\lambda y'(x)}{\sqrt{1 + y'(x)^2}}\Big) - 1 = 0, \tag{18.42}$$

or

$$\frac{y'(x)}{\sqrt{1 + y'(x)^2}} = \frac{x - a}{\lambda}. \tag{18.43}$$

Using the substitution $t = \frac{x-a}{\lambda}$ and integrating, we find that

$$(x - a)^2 + (y - b)^2 = \lambda^2, \tag{18.44}$$

which is the equation of a circle. So the optimal function $y(x)$ is a portion of a circle.

What happens if the assigned perimeter P is greater than $\frac{\pi}{2}$, the length of the semicircle connecting $(0,0)$ and $(1,0)$? In this case, the desired curve is not the graph of a function of x, but a parameterized curve of the form $(x(t), y(t))$, for, say, t in the interval $[0,1]$. Now we have one independent variable, t, but two dependent ones, x and y. We need a generalization of the Euler-Lagrange Equation to the multivariate case.

18.6.2 Burg Entropy

According to the Euler-Lagrange Equation for this case, we have

$$\frac{1}{y(x)} + \sum_{n=-N}^{N} \lambda_n e^{-ixn}, \tag{18.45}$$

or

$$y(x) = 1/\sum_{n=-N}^{N} a_n e^{inx}. \tag{18.46}$$

The *spectral factorization* theorem [128] tells us that if the denominator is positive for all x, then it can be written as

$$\sum_{n=-N}^{N} a_n e^{inx} = |\sum_{m=0}^{N} b_m e^{imx}|^2. \tag{18.47}$$

With a bit more work (see [44]), it can be shown that the desired coefficients b_m are the solution to the system of equations

$$\sum_{m=0}^{N} r_{m-k} b_m = 0, \tag{18.48}$$

for $k = 1, 2, ..., N$ and

$$\sum_{m=0}^{N} r_m b_m = 1. \tag{18.49}$$

18.7 The Multivariate Case

Suppose that the integral to be optimized is

$$J(x,y) = \int_a^b f(t, x(t), x'(t), y(t), y'(t))dt, \tag{18.50}$$

where $f(u, v, w, s, r)$ is a real-valued function of five variables. In such cases, the Euler-Lagrange Equation is replaced by the two equations

$$\begin{aligned} \frac{d}{dt}\Big(\frac{\partial f}{\partial w}\Big) - \frac{\partial f}{\partial v} &= 0, \\ \frac{d}{dx}\Big(\frac{\partial f}{\partial r}\Big) - \frac{\partial f}{\partial s} &= 0. \end{aligned} \tag{18.51}$$

We apply this now to the problem of maximum area for a fixed perimeter.

We know from Green's Theorem in two dimensions that the area A enclosed by a curve C is given by the integral

$$A = \frac{1}{2}\oint_C (xdy - ydx) = \frac{1}{2}\int_0^1 (x(t)y'(t) - y(t)x'(t))dt. \tag{18.52}$$

The perimeter P of the curve is

$$P = \int_0^1 \sqrt{x'(t)^2 + y'(t)^2}dt. \tag{18.53}$$

So the problem is to maximize the integral in Equation (18.52), subject to the integral in Equation (18.53) being held constant.

The problem is solved by using a Lagrange multiplier. We write

$$J(x, y) = \int_0^1 \Big(x(t)y'(t) - y(t)x'(t) + \lambda\sqrt{x'(t)^2 + y'(t)^2}\Big)dt. \tag{18.54}$$

The generalized Euler-Lagrange Equations are

$$\frac{d}{dt}\Big(\frac{1}{2}x(t) + \frac{\lambda y'(t)}{\sqrt{x'(t)^2 + y'(t)^2}}\Big) + \frac{1}{2}x'(t) = 0, \tag{18.55}$$

and

$$\frac{d}{dt}\Big(-\frac{1}{2}y(t) + \frac{\lambda x'(t)}{\sqrt{x'(t)^2 + y'(t)^2}}\Big) - \frac{1}{2}y'(t) = 0. \tag{18.56}$$

It follows that

$$y(t) + \frac{\lambda x'(t)}{\sqrt{x'(t)^2 + y'(t)^2}} = c, \tag{18.57}$$

and

$$x(t) + \frac{\lambda y'(t)}{\sqrt{x'(t)^2 + y'(t)^2}} = d. \tag{18.58}$$

Therefore,

$$(x - d)^2 + (y - c)^2 = \lambda^2. \tag{18.59}$$

The optimal curve is then a portion of a circle.

18.8 Finite Constraints

Suppose that we want to minimize the functional

$$J(y) = \int_a^b f(x, y(x), y'(x))dx,$$

subject to the constraint

$$g(x, y(x)) = 0.$$

Such a problem is said to be one of *finite constraints*. In this section we illustrate this type of problem by considering the geodesic problem.

18.8.1 The Geodesic Problem

The space curve $(x(t), y(t), z(t))$, defined for $a \leq t \leq b$, lies on the surface described by $G(x, y, z) = 0$ if $G(x(t), y(t), z(t)) = 0$ for all t in $[a, b]$. The *geodesic problem* is to find the curve of shortest length lying on the surface and connecting points $A = (a_1, a_2, a_3)$ and $B = (b_1, b_2, b_3)$. The functional to be minimized is the arc length

$$J = \int_a^b \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}dt, \tag{18.60}$$

where $\dot{x} = \frac{dx}{dt}$.

We assume that the equation $G(x, y, z) = 0$ can be rewritten as

$$z = g(x, y),$$

that is, we assume that we can solve for the variable z, and that the function g has continuous second partial derivatives. We may not be able to do this for the entire surface, as the equation of a sphere $G(x, y, z) = x^2 + y^2 + z^2 - r^2 = 0$ illustrates, but we can usually solve for z, or one of the other variables, on part of the surface, as, for example, on the upper or lower hemisphere.

We then have

$$\dot{z} = g_x\dot{x} + g_y\dot{y} = g_x(x(t), y(t))\dot{x}(t) + g_y(x(t), y(t))\dot{y}(t), \tag{18.61}$$

where $g_x = \frac{\partial g}{\partial x}$.

Lemma 18.1 *We have*

$$\frac{\partial \dot{z}}{\partial x} = \frac{d}{dt}(g_x).$$

Proof: From Equation (18.61) we have

$$\frac{\partial \dot{z}}{\partial x} = \frac{\partial}{\partial x}(g_x\dot{x} + g_y\dot{y}) = g_{xx}\dot{x} + g_{yx}\dot{y}.$$

We also have

$$\frac{d}{dt}(g_x) = \frac{d}{dt}(g_x(x(t), y(t)) = g_{xx}\dot{x} + g_{xy}\dot{y}.$$

Since $g_{xy} = g_{yx}$, the assertion of the lemma follows. ∎

From the Lemma we have both

$$\frac{\partial \dot{z}}{\partial x} = \frac{d}{dt}(g_x), \tag{18.62}$$

and

$$\frac{\partial \dot{z}}{\partial y} = \frac{d}{dt}(g_y). \tag{18.63}$$

Substituting for z in Equation (18.60), we see that the problem is now to minimize the functional

$$J = \int_a^b \sqrt{\dot{x}^2 + \dot{y}^2 + (g_x\dot{x} + g_y\dot{y})^2}dt, \tag{18.64}$$

which we write as

$$J = \int_a^b F(x, \dot{x}, y, \dot{y})dt. \tag{18.65}$$

The Euler-Lagrange Equations are then

$$\frac{\partial F}{\partial x} - \frac{d}{dt}(\frac{\partial F}{\partial \dot{x}}) = 0, \tag{18.66}$$

and

$$\frac{\partial F}{\partial y} - \frac{d}{dt}(\frac{\partial F}{\partial \dot{y}}) = 0. \tag{18.67}$$

Using

$$\frac{\partial F}{\partial x} = \frac{\partial f}{\partial \dot{z}}\frac{\partial (g_x\dot{x} + g_y\dot{y})}{\partial x}$$
$$= \frac{\partial f}{\partial \dot{z}}\frac{\partial}{\partial x}(\frac{dg}{dt}) = \frac{\partial f}{\partial \dot{z}}\frac{\partial \dot{z}}{\partial x}$$

and

$$\frac{\partial F}{\partial y} = \frac{\partial f}{\partial \dot{z}}\frac{\partial \dot{z}}{\partial y},$$

we can rewrite the Euler-Lagrange Equations as

$$\frac{d}{dt}(\frac{\partial f}{\partial \dot{x}}) + g_x \frac{d}{dt}(\frac{\partial f}{\partial \dot{z}}) = 0, \tag{18.68}$$

and

$$\frac{d}{dt}(\frac{\partial f}{\partial \dot{y}}) + g_y \frac{d}{dt}(\frac{\partial f}{\partial \dot{z}}) = 0. \tag{18.69}$$

Let the function $\lambda(t)$ be defined by

$$\frac{d}{dt}(\frac{\partial f}{\partial \dot{z}}) = \lambda(t) G_z,$$

and note that

$$g_x = -\frac{G_x}{G_z},$$

and

$$g_y = -\frac{G_y}{G_z}.$$

Then the Euler-Lagrange Equations become

$$\frac{d}{dt}(\frac{\partial f}{\partial \dot{x}}) = \lambda(t) G_x, \tag{18.70}$$

and

$$\frac{d}{dt}(\frac{\partial f}{\partial \dot{y}}) = \lambda(t) G_y. \tag{18.71}$$

Eliminating $\lambda(t)$ and extending the result to include z as well, we have

$$\frac{\frac{d}{dt}(\frac{\partial f}{\partial \dot{x}})}{G_x} = \frac{\frac{d}{dt}(\frac{\partial f}{\partial \dot{y}})}{G_y} = \frac{\frac{d}{dt}(\frac{\partial f}{\partial \dot{z}})}{G_z}. \tag{18.72}$$

Notice that we could obtain the same result by calculating the Euler-Lagrange Equation for the functional

$$\int_a^b f(\dot{x}, \dot{y}, \dot{z}) + \lambda(t) G(x(t), y(t), z(t)) dt. \tag{18.73}$$

18.8.2 An Example

Let the surface be a sphere, with equation

$$0 = G(x, y, z) = x^2 + y^2 + z^2 - r^2.$$

Then Equation (18.72) becomes

$$\frac{f\ddot{x} - \dot{x}\dot{f}}{2xf^2} = \frac{f\ddot{y} - \dot{y}\dot{f}}{2yf^2} = \frac{f\ddot{z} - \dot{z}\dot{f}}{2zf^2}.$$

We can rewrite these equations as

$$\frac{\ddot{x}y - x\ddot{y}}{\dot{x}y - x\dot{y}} = \frac{y\ddot{z} - z\ddot{y}}{y\dot{z} - z\dot{y}} = \frac{\dot{f}}{f}.$$

The numerators are the derivatives, with respect to t, of the denominators, which leads to

$$\log|x\dot{y} - y\dot{x}| = \log|y\dot{z} - z\dot{y}| + c_1.$$

Therefore,

$$x\dot{y} - y\dot{x} = c_1(y\dot{z} - z\dot{y}).$$

Rewriting, we obtain

$$\frac{\dot{x} + c_1\dot{z}}{x + c_1 z} = \frac{\dot{y}}{y},$$

or

$$x + c_1 z = c_2 y,$$

which is a plane through the origin. The geodesics on the sphere are great circles, that is, the intersection of the sphere with a plane through the origin.

18.9 Exercises

18.1 *Suppose that the cycloid in the brachistochrone problem connects the starting point* $(0, 0)$ *with the point* $(\pi a, -2a)$*, where* $a > 0$*. Show that the time required for the ball to reach the point* $(\pi a, -2a)$ *is* $\pi\sqrt{\frac{a}{g}}$*.*

18.2 *Show that, for the situation in the previous exercise, the time required for the ball to reach* $(\pi a, -2a)$ *is again* $\pi\sqrt{\frac{a}{g}}$*, if the ball begins rolling at any intermediate point along the cycloid. This is the tautochrone property of the cycloid.*

Chapter 19

Appendix: Metric Spaces and Norms

The inner product on R^J or C^J can be used to define the Euclidean norm $\|x\|_2$ of a vector x, which, in turn, provides a *metric*, or a measure of distance between two vectors, $d(x, y) = \|x - y\|_2$. The notions of metric and norm are actually more general notions, with no necessary connection to the inner product. Throughout this chapter the superscript $\dagger$ denotes the conjugate transpose of a matrix or vector.

19.1 Metric Spaces

We begin with the basic definitions.

Definition 19.1 *Let $\mathcal{S}$ be a non-empty set. We say that the function $d : \mathcal{S} \times \mathcal{S} \to [0, +\infty)$ is a metric if the following hold:*

$$d(s, t) \geq 0, \tag{19.1}$$

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

$$d(s, t) = 0 \tag{19.2}$$

if and only if $s = t$;

$$d(s, t) = d(t, s), \tag{19.3}$$

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

$$d(s, t) \leq d(s, u) + d(u, t). \tag{19.4}$$

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

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

19.2 Analysis in Metric Space

Analysis is concerned with issues of convergence and limits.

Definition 19.2 *A sequence $\{s^k\}$ in the metric space $(\mathcal{S}, d)$ is said to have limit s^* if*

$$\lim_{k \to +\infty} d(s^k, s^*) = 0. \tag{19.5}$$

Any sequence with a limit is said to be convergent.

A sequence can have at most one limit.

Definition 19.3 *The sequence $\{s^k\}$ is said to be a Cauchy sequence if, for any $\epsilon > 0$, there is positive integer m, such that, for any nonnegative integer n,*

$$d(s^m, s^{m+n}) \leq \epsilon. \tag{19.6}$$

Every convergent sequence is a Cauchy sequence.

Definition 19.4 *The metric space $(\mathcal{S}, d)$ is said to be complete if every Cauchy sequence is a convergent sequence.*

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

Definition 19.5 *An infinite sequence $\{s^k\}$ in $\mathcal{S}$ is said to be bounded if there is an element a and a positive constant $b > 0$ such that $d(a, s^k) \leq b$, for all k.*

Definition 19.6 *A subset K of the metric space is said to be closed if, for every convergent sequence $\{s^k\}$ of elements in K, the limit point is again in K. The closure of a set K is the smallest closed set containing K.*

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

Definition 19.7 *For any bounded sequence $\{x^k\}$ in R^J, there is at least one subsequence, often denoted $\{x^{k_n}\}$, that is convergent; the notation implies that the positive integers k_n are ordered, so that $k_1 < k_2 < ....$ The limit of such a subsequence is then said to be a cluster point of the original sequence.*

When we investigate iterative algorithms, we will want to know if the sequence $\{x^k\}$ generated by the algorithm converges. As a first step, we will usually ask if the sequence is bounded? If it is bounded, then it will have at least one cluster point. We then try to discover if that cluster point is really the limit of the sequence. We turn now to metrics that come from norms.

19.3 Norms

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

Definition 19.8 *We say that $\|\cdot\|$ is a norm on C^J if*

$$\|x\| \geq 0, \tag{19.7}$$

for all x,

$$\|x\| = 0 \tag{19.8}$$

if and only if $x = 0$,

$$\|\gamma x\| = |\gamma| \, \|x\|, \tag{19.9}$$

for all x and scalars γ, and

$$\|x + y\| \leq \|x\| + \|y\|, \tag{19.10}$$

for all vectors x and y.

Lemma 19.1 *The function $d(x, y) = \|x - y\|$ defines a metric on C^J.*

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

19.3.1 Some Common Norms on C^J

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

The 1-norm

The 1-norm on C^J is defined by

$$\|x\|_1 = \sum_{j=1}^{J} |x_j|. \tag{19.11}$$

The ∞-norm

The ∞-norm on C^J is defined by

$$\|x\|_\infty = \max\{|x_j| \, | j = 1, ..., J\}. \tag{19.12}$$

The 2-norm

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

$$\|x\|_2 = \sqrt{\langle x, x\rangle} = \sqrt{x^\dagger x}. \tag{19.13}$$

Weighted 2-norms

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

$$\|x\|_Q = \|Ax\|_2 = \sqrt{x^\dagger Q x}, \tag{19.14}$$

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

$$\|x\|_Q = \sqrt{\sum_{j=1}^{J} Q_{jj}|x_j|^2}; \tag{19.15}$$

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

19.4 Eigenvalues and Eigenvectors

Let S be a complex, square matrix. We say that λ is an eigenvalue of S if λ is a root of the complex polynomial $\det(\lambda I - S)$. Therefore, each S has as many (possibly complex) eigenvalues as it has rows or columns, although some of the eigenvalues may be repeated.

An equivalent definition is that λ is an eigenvalue of S if there is a non-zero vector x with $Sx = \lambda x$, in which case the vector x is called an *eigenvector* of S. From this definition, we see that the matrix S is invertible if and only if zero is not one of its eigenvalues. The *spectral radius* of S, denoted $\rho(S)$, is the maximum of $|\lambda|$, over all eigenvalues λ of S.

If S is an I by I Hermitian matrix with (necessarily real) eigenvalues

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_I, \tag{19.16}$$

and associated (column) eigenvectors $\{u_i \,|i = 1, ..., I\}$ (which we may assume are mutually orthogonal), then S can be written as

$$S = \lambda_1 u_1 u_1^\dagger + \cdots + \lambda_I u_I u_I^\dagger. \tag{19.17}$$

This is the *eigenvalue/eigenvector decomposition* of S. The Hermitian matrix S is invertible if and only if all of its eigenvalues are non-zero, in which case we can write the inverse of S as

$$S^{-1} = \lambda_1^{-1} u_1 u_1^\dagger + \cdots + \lambda_I^{-1} u_I u_I^\dagger. \tag{19.18}$$

Definition 19.9 *A Hermitian matrix S is positive-definite if each of its eigenvalues is positive.*

It follows from the eigenvector decomposition of S that $S = QQ^\dagger$ for the Hermitian, positive-definite matrix

$$Q = \sqrt{\lambda_1}u_1u_1^\dagger + \cdots + \sqrt{\lambda_I}u_Iu_I^\dagger; \tag{19.19}$$

Q is called the *Hermitian square root* of S.

19.4.1 The Singular-Value Decomposition

The eigenvector/eigenvalue decomposition applies only to square matrices. The singular-value decomposition is similar, but applies to any matrix.

Definition 19.10 *Let A be an I by J complex matrix. The rank of A is the number of linearly independent rows, which always equals the number of linearly independent columns. The matrix A is said to have full rank if its rank is the smaller of I and J.*

Let $I \leq J$. Let $B = AA^\dagger$ and $C = A^\dagger A$. Let $\lambda_i \geq 0$, for $i = 1, ..., I$, be the eigenvalues of B, and let $\{u^1, ..., u^I\}$ be associated orthonormal eigenvectors of B. Assume that $\lambda_i > 0$ for $i = 1, ..., N \leq I$, and, if $N < I$, $\lambda_i = 0$, for $i = N+1, ..., I$; if $N = I$, then the matrix A has full rank. For $i = 1, ..., N$, let $v^i = \lambda_i^{-1/2}A^\dagger u^i$. It is easily shown that the collection $\{v^1, ..., v^N\}$ is orthonormal. Let $\{v^{N+1}, ..., v^J\}$ be selected so that $\{v^1, ..., v^J\}$ is orthonormal. Then the sets $\{u^1, ..., u^N\}$, $\{u^{N+1}, ..., u^I\}$, $\{v^1, ..., v^N\}$, and $\{v^{N+1}, ..., v^J\}$ are orthonormal bases for the subspaces $CS(A)$, $NS(A^\dagger)$, $CS(A^\dagger)$, and $NS(A)$, respectively, where $CS(A)$ is the subspace spanned by the columns of A, and $NS(A)$ is the set of all vectors orthogonal to the columns of A.

Definition 19.11 *We have*

$$A = \sum_{i=1}^{N} \sqrt{\lambda_i}u^i(v^i)^\dagger, \tag{19.20}$$

which is the singular-value decomposition (SVD) of the matrix A.

Let U and V be the square matrices whose columns are the vectors u^i and v^j, respectively, and L the I by J matrix with entries $L_{ii} = \lambda_i$, and the remaining ones equal to zero. Then the SVD of A can be expressed as follows:

$$A = U\sqrt{L}V^\dagger.$$

The SVD of the matrix $A^\dagger$ is then

$$A^\dagger = \sum_{i=1}^{N} \sqrt{\lambda_i}v^i(u^i)^\dagger. \tag{19.21}$$

Definition 19.12 *The pseudo-inverse of the matrix A is the J by I matrix*

$$A^\sharp = \sum_{i=1}^{N} \lambda_i^{-1/2} v^i (u^i)^\dagger. \tag{19.22}$$

Lemma 19.2 *For any matrix A, we have*

$$(A^\dagger)^\sharp = (A^\sharp)^\dagger. \tag{19.23}$$

For A that has full rank, if $N = I \leq J$, then

$$A^\sharp = A^\dagger B^{-1}, \tag{19.24}$$

and

$$(A^\dagger)^\sharp = B^{-1} A. \tag{19.25}$$

19.5 Matrix Norms

Any matrix can be turned into a vector by vectorization. Therefore, we can define a norm for any matrix by simply vectorizing and taking a norm of the resulting vector. Such norms for matrices may not be compatible with the role of a matrix as representing a linear transformation.

19.5.1 Induced Matrix Norms

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

Definition 19.13 *Let $\|x\|$ be any norm on C^J, not necessarily the Euclidean norm, $\|b\|$ any norm on C^I, and A a rectangular I by J matrix. The induced matrix norm of A, simply denoted $\|A\|$, derived from these two vectors norms, is the smallest positive constant c such that*

$$\|Ax\| \leq c\|x\|, \tag{19.26}$$

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

$$\|A\| = \max_{x \neq 0} \{\|Ax\|/\|x\|\}. \tag{19.27}$$

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

$$\|Ax - Az\| \leq \|A\| \, \|x - z\|, \tag{19.28}$$

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

19.5.2 Condition Number of a Square Matrix

Let S be a square, invertible matrix and z the solution to $Sz = h$. We are concerned with the extent to which the solution changes as the right side, h, changes. Denote by δ_h a small perturbation of h, and by δ_z the solution of $S\delta_z = \delta_h$. Then $S(z+\delta_z) = h+\delta_h$. Applying the compatibility condition $\|Ax\| \leq \|A\|\|x\|$, we get

$$\|\delta_z\| \leq \|S^{-1}\|\|\delta_h\|, \tag{19.29}$$

and

$$\|z\| \geq \|h\|/\|S\|. \tag{19.30}$$

Therefore

$$\frac{\|\delta_z\|}{\|z\|} \leq \|S\|\,\|S^{-1}\|\frac{\|\delta_h\|}{\|h\|}. \tag{19.31}$$

Definition 19.14 *The quantity $c = \|S\|\|S^{-1}\|$ is the condition number of S, with respect to the given matrix norm.*

Note that $c \geq 1$: for any non-zero z, we have

$$\|S^{-1}\| \geq \|S^{-1}z\|/\|z\| = \|S^{-1}z\|/\|SS^{-1}z\| \geq 1/\|S\|. \tag{19.32}$$

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

$$c = \lambda_{max}(S)/\lambda_{min}(S), \tag{19.33}$$

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

19.5.3 Some Examples of Induced Matrix Norms

If we choose the two vector norms carefully, then we can get an explicit description of $\|A\|$, but, in general, we cannot.

For example, let $\|x\| = \|x\|_1$ and $\|Ax\| = \|Ax\|_1$ be the 1-norms of the vectors x and Ax, where

$$\|x\|_1 = \sum_{j=1}^{J} |x_j|. \tag{19.34}$$

Lemma 19.3 *The 1-norm of A, induced by the 1-norms of vectors in C^J and C^I, is*

$$\|A\|_1 = \max\{\sum_{i=1}^{I} |A_{ij}|\,, j = 1, 2, ..., J\}. \tag{19.35}$$

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

$$\|Ax\|_1 \leq \sum_{j=1}^{J}(\sum_{i=1}^{I}|A_{ij}|)|x_j|. \tag{19.36}$$

Then let $j = m$ be the index for which the maximum column sum is reached and select $x_j = 0$, for $j \neq m$, and $x_m = 1$. ∎

The *infinity norm* of the vector x is

$$\|x\|_\infty = \max\{|x_j|, j = 1, 2, ..., J\}. \tag{19.37}$$

Lemma 19.4 *The infinity norm of the matrix A, induced by the infinity norms of vectors in R^J and C^I, is*

$$\|A\|_\infty = \max\{\sum_{j=1}^{J}|A_{ij}|, i = 1, 2, ..., I\}. \tag{19.38}$$

The proof is similar to that of the previous lemma.

Lemma 19.5 *Let M be an invertible matrix and $\|x\|$ any vector norm. Define*

$$\|x\|_M = \|Mx\|. \tag{19.39}$$

Then, for any square matrix S, the matrix norm

$$\|S\|_M = \max_{x\neq 0}\{\|Sx\|_M/\|x\|_M\} \tag{19.40}$$

is

$$\|S\|_M = \|MSM^{-1}\|. \tag{19.41}$$

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

Lemma 19.6 *Let S be any square matrix and let $\epsilon > 0$ be given. Then there is an invertible matrix M such that*

$$\|S\|_M \leq \rho(S) + \epsilon. \tag{19.42}$$

19.5.4 The Euclidean Norm of a Square Matrix

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

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

$$\|A\|_2 = \max\{\|Ax\|_2/\|x\|_2\}, \tag{19.43}$$

with the maximum over all nonzero vectors x. Since

$$\|Ax\|_2^2 = x^\dagger A^\dagger Ax, \tag{19.44}$$

we have

$$\|A\|_2 = \sqrt{\max\{\frac{x^\dagger A^\dagger Ax}{x^\dagger x}\}}, \tag{19.45}$$

over all nonzero vectors x.

Proposition 19.1 *The Euclidean norm of a square matrix is*

$$\|A\|_2 = \sqrt{\rho(A^\dagger A)}; \tag{19.46}$$

that is, the term inside the square-root in Equation (19.45) is the largest eigenvalue of the matrix $A^\dagger A$.

Proof: Let

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_J \geq 0 \tag{19.47}$$

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

$$x = \sum_{j=1}^{J} [(u^j)^\dagger x] u^j, \tag{19.48}$$

while

$$A^\dagger Ax = \sum_{j=1}^{J} [(u^j)^\dagger x] A^\dagger A u^j = \sum_{j=1}^{J} \lambda_j [(u^j)^\dagger x] u^j. \tag{19.49}$$

It follows that

$$\|x\|_2^2 = x^\dagger x = \sum_{j=1}^{J} |(u^j)^\dagger x|^2, \tag{19.50}$$

and

$$\|Ax\|_2^2 = x^\dagger A^\dagger Ax = \sum_{j=1}^{J} \lambda_j |(u^j)^\dagger x|^2. \tag{19.51}$$

Maximizing $\|Ax\|_2^2/\|x\|_2^2$ over $x \neq 0$ is equivalent to maximizing $\|Ax\|_2^2$, subject to $\|x\|_2^2 = 1$. The right side of Equation (19.51) is then a convex combination of the λ_j, which will have its maximum when only the coefficient of λ_1 is non-zero. ∎

According to Corollary 13.1, we have the inequality

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

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

$$S = \begin{bmatrix} i & 2 \\ 0 & i \end{bmatrix}, \tag{19.52}$$

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

$$S^\dagger S = \begin{bmatrix} 1 & -2i \\ 2i & 5 \end{bmatrix} \tag{19.53}$$

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

$$\|S\|_2 = \sqrt{3 + 2\sqrt{2}}. \tag{19.54}$$

19.5.5 Diagonalizable Matrices

Definition 19.15 *A square matrix S is diagonalizable if C^J has a basis of eigenvectors of S.*

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

19.5.6 Gerschgorin's Theorem

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

Theorem 19.1 *Let A be J by J. For $j = 1, ..., J$, let C_j be the circle in the complex plane with center A_{jj} and radius $r_j = \sum_{m \neq j} |A_{jm}|$. Then every eigenvalue of A lies within one of the C_j.*

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

$$(\lambda - A_{jj})u_j = \sum_{m \neq j} A_{jm} u_m, \tag{19.55}$$

so that

$$|\lambda - A_{jj}| \leq \sum_{m \neq j} |A_{jm}||u_m|/|u_j| \leq r_j. \tag{19.56}$$

This completes the proof. ∎

19.5.7 Strictly Diagonally Dominant Matrices

Definition 19.16 *A square I by I matrix S is said to be strictly diagonally dominant if, for each $i = 1, ..., I$,*

$$|S_{ii}| > r_i = \sum_{m \neq i} |S_{im}|. \tag{19.57}$$

When the matrix S is strictly diagonally dominant, all the eigenvalues of S lie within the union of the spheres with centers S_{ii} and radii S_{ii}. With D the diagonal component of S, the matrix $D^{-1}S$ then has all its eigenvalues within the circle of radius one, centered at $(1, 0)$. Then $\rho(I - D^{-1}S) < 1$. This result is used discussing the Jacobi splitting method [45].

19.6 Exercises

19.1 *Show that every convergent sequence is a Cauchy sequence.*

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

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

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

19.5 *Show that your bounded, but not convergent, sequence found in Exercise 19.3 has a cluster point.*

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

19.7 *Show that the 1-norm is a norm.*

19.8 *Show that the ∞-norm is a norm.*

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

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

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

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

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

$$B = \sum_{i=1}^{N} \lambda_i u^i (u^i)^\dagger, \tag{19.58}$$

and

$$C = \sum_{i=1}^{N} \lambda_i v^i (v^i)^\dagger. \tag{19.59}$$

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

$$\|S\| \geq |\lambda|, \tag{19.60}$$

for every eigenvalue λ of S. So we know that

$$\rho(S) \leq \|S\|, \tag{19.61}$$

for every induced matrix norm, but, according to Lemma 19.6, we also have

$$\|S\|_M \leq \rho(S) + \epsilon. \tag{19.62}$$

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

19.16 *Show that, if S is Hermitian, then $\|S\|_2 = \rho(S)$. Hint: use Exercise (19.11).*

Chapter 20

Appendix: Differentiation

The definition of the derivative of a function $g : D \subseteq R \to R$ is a familiar one. In this chapter we examine various ways in which this definition can be extended to functions $f : D \subseteq R^J \to R$ of several variables. Here D is the domain of the function f and we assume that $\text{int}(D)$ is not empty.

20.1 Directional Derivative

We begin with one- and two-sided directional derivatives.

20.1.1 Definitions

The function $g(x) = |x|$ does not have a derivative at $x = 0$, but it has *one-sided directional derivatives* there. The one-sided directional derivative of $g(x)$ at $x = 0$, in the direction of $x = 1$, is

$$g'_+(0;1) = \lim_{t \downarrow 0} \frac{1}{t}[g(0+t) - g(0)] = 1, \tag{20.1}$$

and in the direction of $x = -1$, it is

$$g'_+(0;-1) = \lim_{t \downarrow 0} \frac{1}{t}[g(0-t) - g(0)] = 1. \tag{20.2}$$

However, the two-sided derivative of $g(x) = |x|$ does not exist at $x = 0$.

We can extend the concept of one-sided directional derivatives to functions of several variables.

Definition 20.1 *Let $f : D \subseteq R^J \to R$ be a real-valued function of several variables, let a be in int(D), and let d be a unit vector in R^J. The one-sided directional derivative of $f(x)$, at $x = a$, in the direction of d, is*

$$f'_+(a;d) = \lim_{t \downarrow 0} \frac{1}{t}[f(a+td) - f(a)]. \tag{20.3}$$

Definition 20.2 *The two-sided directional derivative of $f(x)$ at $x = a$, in the direction of d, is*

$$f'(a;d) = \lim_{t \to 0} \frac{1}{t}[f(a+td) - f(a)]. \tag{20.4}$$

If the two-sided directional derivative exists then we have

$$f'(a;d) = f'_+(a;d) = -f'_+(a;-d).$$

Given $x = a$ and d, we define the function $\phi(t) = f(a+td)$, for t such that $a + td$ is in D. The derivative of $\phi(t)$ at $t = 0$ is then

$$\phi'(0) = \lim_{t \to 0} \frac{1}{t}[\phi(t) - \phi(0)] = f'(a;d). \tag{20.5}$$

20.2 Partial Derivatives

For $j = 1, ..., J$, denote by e^j the vector whose entries are all zero, except for a one in the jth position.

Definition 20.3 *If $f'(a;e^j)$ exists, then it is $\frac{\partial f}{\partial x_j}(a)$, the partial derivative of $f(x)$, at $x = a$, with respect to x_j, the jth entry of the variable vector x.*

Definition 20.4 *If the partial derivative, at $x = a$, with respect to x_j, exists for each j, then the gradient of $f(x)$, at $x = a$, is the vector $\nabla f(a)$ whose entries are $\frac{\partial f}{\partial x_j}(a)$.*

20.3 Some Examples

We consider some examples of directional derivatives.

20.3.1 Example 1.

For $(x, y) \neq (0, 0)$, let

$$f(x,y) = \frac{2xy}{x^2 + y^2},$$

and define $f(0, 0) = 1$. Let $d = (\cos\theta, \sin\theta)$. Then it is easy to show that $\phi(t) = \sin 2\theta$, for $t \neq 0$, and $\phi(0) = 1$. If θ is such that $\sin 2\theta = 1$, then $\phi(t)$ is constant, and $\phi'(0) = 0$. But, if $\sin 2\theta \neq 1$, then $\phi(t)$ is discontinuous at $t = 0$, so $\phi(t)$ is not differentiable at $t = 0$. Therefore, $f(x, y)$ has a two-sided directional derivative at $(x, y) = (0, 0)$ only in certain directions.

20.3.2 Example 2.

For $(x, y) \neq (0, 0)$, let

$$f(x, y) = \frac{2xy^2}{x^2 + y^4},$$

and $f(0, 0) = 0$. Again, let $d = (\cos\theta, \sin\theta)$. Then we have

$$\phi'(0) = \frac{2\sin^2\theta}{\cos^2\theta},$$

for $\cos\theta \neq 0$. If $\cos\theta = 0$, then $f(x)$ is the constant zero in that direction, so $\phi'(0) = 0$. Therefore, the function $f(x, y)$ has a two-sided directional derivative at $(x, y) = (0, 0)$, for every vector d. Note that the two partial derivatives are both zero at $(x, y) = (0, 0)$, so $\nabla f(0, 0) = 0$.

20.4 Gâteaux Derivative

Just having a two-sided directional derivative for every d is not sufficient, in most cases; we need something stronger.

Definition 20.5 *If $f(x)$ has a two-sided directional derivative at $x = a$, for every vector d, and, in addition,*

$$f'(a; d) = \langle \nabla f(a), d \rangle,$$

for each d, then $f(x)$ is Gâteaux-differentiable at $x = a$, and $\nabla f(a)$ is the Gâteaux derivative of $f(x)$ at $x = a$, also denoted $f'(a)$.

Example 2 above showed that it is possible for $f(x)$ to have a two-sided directional derivative at $x = a$, for every d, and yet fail to be Gâteaux-differentiable.

From Cauchy's Inequality, we know that

$$|f'(a; d)| = |\langle \nabla f(a), d \rangle| \leq ||\nabla f(a)||_2 \, ||d||_2,$$

and that $f'(a; d)$ attains its most positive value when the direction d is a positive multiple of $\nabla f(a)$. This is the motivation for steepest descent optimization.

For ordinary functions $g : D \subseteq R \to R$, we know that differentiability implies continuity. It is possible for $f(x)$ to be Gâteaux-differentiable at $x = a$ and yet not be continuous at $x = a$; see Ortega and Rheinboldt [127]. This means that the notion of Gâteaux-differentiability is too weak. In order to have a nice theory of multivariate differentiation, the notion of derivative must be strengthened. The stronger notion we seek is Fréchet differentiability.

20.5 Fréchet Derivative

The notion of Fréchet differentiability is the one appropriate for our purposes.

20.5.1 The Definition

Definition 20.6 *We say that $f(x)$ is Fréchet-differentiable at $x = a$ and $\nabla f(a)$ is its Fréchet derivative if*

$$\lim_{||h||\to 0} \frac{1}{||h||}|f(a+h) - f(a) - \langle \nabla f(a), h\rangle| = 0.$$

Notice that the limit in the definition of the Fréchet derivative involves the norm of the incremental vector h, which is where the power of the Fréchet derivative arises. Also, since the norm and the associated inner product can be changed, so can the Fréchet derivative; see Exercise 20.1 for an example. The corresponding limit in the definition of the Gâteaux derivative involves only the scalar t, and therefore requires no norm and makes sense in any vector space.

20.5.2 Properties of the Fréchet Derivative

It can be shown that if $f(x)$ is Fréchet-differentiable at $x = a$, then $f(x)$ is continuous at $x = a$. If $f(x)$ is Gâteaux-differentiable at each point in an open set containing $x = a$, and $\nabla f(x)$ is continuous at $x = a$, then $\nabla f(a)$ is also the Fréchet derivative of $f(x)$ at $x = a$. Since the continuity of $\nabla f(x)$ is equivalent to the continuity of each of the partial derivatives, we learn that $f(x)$ is Fréchet-differentiable at $x = a$ if it is Gâteaux-differentiable in a neighborhood of $x = a$ and the partial derivatives are continuous at $x = a$.

20.6 The Chain Rule

For fixed a and d in R^J, the function $\phi(t) = f(a + td)$, defined for the real variable t, is a composition of the function $f : R^J \to R$ itself and the function $g : R \to R^J$ defined by $g(t) = a + td$; that is, $\phi(t) = f(g(t))$. Writing

$$f(a + td) = f(a_1 + td_1, a_2 + td_2, ..., a_J + td_J),$$

and applying the Chain Rule, we find that

$$f'(a; d) = \phi'(0) = \frac{\partial f}{\partial x_1}(a)d_1 + ... + \frac{\partial f}{\partial x_J}(a)d_J;$$

that is,

$$f'(a;d) = \phi'(0) = \langle \nabla f(a), d \rangle.$$

But we know that $f'(a;d)$ is not always equal to $\langle \nabla f(a), d \rangle$. This means that the Chain Rule is not universally true and must involve conditions on the function f. Clearly, unless the function f is Gâteaux-differentiable, the chain rule cannot hold. For an in-depth treatment of this matter, consult Ortega and Rheinboldt [127].

20.7 Exercises

20.1 *Let Q be a real, positive-definite symmetric matrix. Define the Q-inner product on R^J to be*

$$\langle x, y \rangle_Q = x^T Q y = \langle x, Qy \rangle,$$

and the Q-norm to be

$$||x||_Q = \sqrt{\langle x, x \rangle_Q}.$$

Show that, if $\nabla f(a)$ is the Fréchet derivative of $f(x)$ at $x = a$, for the usual Euclidean norm, then $Q^{-1}\nabla f(a)$ is the Fréchet derivative of $f(x)$ at $x = a$, for the Q-norm. Hint: use the inequality

$$\sqrt{\lambda_J}||h||_2 \leq ||h||_Q \leq \sqrt{\lambda_1}||h||_2,$$

where λ_1 and λ_J denote the greatest and smallest eigenvalues of Q, respectively.

20.2 ([15], Ex. 10, p. 134) *For (x, y) not equal to $(0, 0)$, let*

$$f(x, y) = \frac{x^a y^b}{x^p + y^q},$$

with $f(0, 0) = 0$. In each of the five cases below, determine if the function is continuous, Gâteaux, Fréchet or continuously differentiable at $(0, 0)$.

- *1)* $a = 2$, $b = 3$, $p = 2$, *and* $q = 4$;
- *2)* $a = 1$, $b = 3$, $p = 2$, *and* $q = 4$;
- *3)* $a = 2$, $b = 4$, $p = 4$, *and* $q = 8$;
- *4)* $a = 1$, $b = 2$, $p = 2$, *and* $q = 2$;
- *5)* $a = 1$, $b = 2$, $p = 2$, *and* $q = 4$.

Chapter 21

Appendix: Inner Product Spaces

An *inner product* is a generalization of the dot product between two vectors. An *inner product space* or *pre-Hilbert space* is a vector space on which we have defined an inner product. Such spaces arise in many areas of mathematics and provide a convenient setting for performing optimal approximation.

21.1 Background

We begin by recalling the solution of the vibrating string problem and Sturm-Liouville problems.

21.1.1 The Vibrating String

When we solve the problem of the vibrating string using the technique of separation of variables, the differential equation involving the space variable x, and assuming constant mass density, is

$$y''(x) + \frac{\omega^2}{c^2} y(x) = 0, \tag{21.1}$$

which we can write as an eigenvalue problem

$$y''(x) + \lambda y(x) = 0. \tag{21.2}$$

The solutions to Equation (21.1) are

$$y(x) = \alpha \sin\left(\frac{\omega}{c} x\right).$$

In the vibrating string problem, the string is fixed at both ends, $x = 0$ and $x = L$, so that

$$\phi(0,t) = \phi(L,t) = 0,$$

for all t. Therefore, we must have $y(0) = y(L) = 0$, so that the *eigenfunction solution* that corresponds to the eigenvalue $\lambda_m = \left(\frac{\pi m}{L}\right)^2$ must have the form

$$y(x) = A_m \sin\left(\frac{\omega_m}{c}x\right) = A_m \sin\left(\frac{\pi m}{L}x\right),$$

where $\omega_m = \frac{\pi c m}{L}$, for any positive integer m. Therefore, the boundary conditions limit the choices for the separation constant ω.

We then discover that the eigenfunction solutions corresponding to different λ are *orthogonal*, in the sense that

$$\int_0^L \sin\left(\frac{\pi m}{L}x\right)\sin\left(\frac{\pi n}{L}x\right)dx = 0,$$

for $m \neq n$.

21.1.2 The Sturm-Liouville Problem

The general form for the Sturm-Liouville Problem is

$$\frac{d}{dx}\Big(p(x)y'(x)\Big) + \lambda w(x)y(x) = 0. \tag{21.3}$$

As with the one-dimensional wave equation, boundary conditions, such as $y(a) = y(b) = 0$, where $a = -\infty$ and $b = +\infty$ are allowed, restrict the possible eigenvalues λ to an increasing sequence of positive numbers λ_m. The corresponding eigenfunctions $y_m(x)$ will be $w(x)$-orthogonal, meaning that

$$0 = \int_a^b y_m(x)y_n(x)w(x)dx,$$

for $m \neq n$. For various choices of $w(x)$ and $p(x)$ and various choices of a and b, we obtain several famous sets of "orthogonal" functions.

Well known examples of Sturm-Liouville problems include

- **Legendre:**

$$\frac{d}{dx}\Big((1-x^2)\frac{dy}{dx}\Big) + \lambda y = 0;$$

- **Chebyshev:**

$$\frac{d}{dx}\Big(\sqrt{1-x^2}\frac{dy}{dx}\Big) + \lambda(1-x^2)^{-1/2}y = 0;$$

- **Hermite:**
$$\frac{d}{dx}\left(e^{-x^2}\frac{dy}{dx}\right) + \lambda e^{-x^2} y = 0;$$
and
- **Laguerre:**
$$\frac{d}{dx}\left(xe^{-x}\frac{dy}{dx}\right) + \lambda e^{-x} y = 0.$$

Each of these examples involves an inner product space and an orthogonal basis for that space.

21.2 The Complex Vector Dot Product

An *inner product* is a generalization of the notion of the dot product between two complex vectors.

21.2.1 The Two-Dimensional Case

Let $\mathbf{u} = (a, b)$ and $\mathbf{v} = (c, d)$ be two vectors in two-dimensional space. Let $\mathbf{u}$ make the angle $\alpha > 0$ with the positive x-axis and $\mathbf{v}$ the angle $\beta > 0$. Let $||\mathbf{u}|| = \sqrt{a^2 + b^2}$ denote the length of the vector $\mathbf{u}$. Then $a = ||\mathbf{u}|| \cos\alpha$, $b = ||\mathbf{u}|| \sin\alpha$, $c = ||\mathbf{v}|| \cos\beta$ and $d = ||\mathbf{v}|| \sin\beta$. So $\mathbf{u} \cdot \mathbf{v} = ac + bd = ||\mathbf{u}||||\mathbf{v}||(\cos\alpha\cos\beta + \sin\alpha\sin\beta = ||\mathbf{u}||\,||\mathbf{v}||\cos(\alpha - \beta)$. Therefore, we have

$$\mathbf{u} \cdot \mathbf{v} = ||\mathbf{u}||\,||\mathbf{v}||\cos\theta, \tag{21.1}$$

where $\theta = \alpha - \beta$ is the angle between $\mathbf{u}$ and $\mathbf{v}$. Cauchy's inequality is

$$|\mathbf{u} \cdot \mathbf{v}| \leq ||\mathbf{u}||\,||\mathbf{v}||,$$

with equality if and only if $\mathbf{u}$ and $\mathbf{v}$ are parallel. From Equation (21.1) we know that the dot product $\mathbf{u} \cdot \mathbf{v}$ is zero if and only if the angle between these two vectors is a right angle; we say then that $\mathbf{u}$ and $\mathbf{v}$ are mutually *orthogonal*.

Cauchy's inequality extends to complex vectors $\mathbf{u}$ and $\mathbf{v}$:

$$\mathbf{u} \cdot \mathbf{v} = \sum_{n=1}^{N} u_n \overline{v_n}, \tag{21.2}$$

and Cauchy's Inequality still holds.

Proof of Cauchy's Inequality: To prove Cauchy's inequality for the complex vector dot product, we write $\mathbf{u} \cdot \mathbf{v} = |\mathbf{u} \cdot \mathbf{v}|e^{i\theta}$. Let t be a real variable and consider

$$0 \leq ||e^{-i\theta}\mathbf{u} - t\mathbf{v}||^2 = (e^{-i\theta}\mathbf{u} - t\mathbf{v}) \cdot (e^{-i\theta}\mathbf{u} - t\mathbf{v})$$

$$= ||\mathbf{u}||^2 - t[(e^{-i\theta}\mathbf{u}) \cdot \mathbf{v} + \mathbf{v} \cdot (e^{-i\theta}\mathbf{u})] + t^2||\mathbf{v}||^2$$
$$= ||\mathbf{u}||^2 - t[(e^{-i\theta}\mathbf{u}) \cdot \mathbf{v} + \overline{(e^{-i\theta}\mathbf{u}) \cdot \mathbf{v}}] + t^2||\mathbf{v}||^2$$
$$= ||\mathbf{u}||^2 - 2Re(te^{-i\theta}(\mathbf{u} \cdot \mathbf{v})) + t^2||\mathbf{v}||^2$$
$$= ||\mathbf{u}||^2 - 2Re(t|\mathbf{u} \cdot \mathbf{v}|) + t^2||\mathbf{v}||^2 = ||\mathbf{u}||^2 - 2t|\mathbf{u} \cdot \mathbf{v}| + t^2||\mathbf{v}||^2.$$

This is a nonnegative quadratic polynomial in the variable t, so it cannot have two distinct real roots. Therefore, the discriminant $4|\mathbf{u} \cdot \mathbf{v}|^2 - 4||\mathbf{v}||^2||\mathbf{u}||^2$ must be non-positive; that is, $|\mathbf{u} \cdot \mathbf{v}|^2 \leq ||\mathbf{u}||^2||\mathbf{v}||^2$. This is Cauchy's inequality. ∎

A careful examination of the proof just presented shows that we did not explicitly use the definition of the complex vector dot product, but only some of its properties. This suggested to mathematicians the possibility of abstracting these properties and using them to define a more general concept, an *inner product*, between objects more general than complex vectors, such as infinite sequences, random variables, and matrices. Such an inner product can then be used to define the *norm* of these objects and thereby a distance between such objects. Once we have an inner product defined, we also have available the notions of orthogonality and best approximation.

21.2.2 Orthogonality

Consider the problem of writing the two-dimensional real vector $(3, -2)$ as a linear combination of the vectors $(1, 1)$ and $(1, -1)$; that is, we want to find constants a and b so that $(3, -2) = a(1, 1) + b(1, -1)$. One way to do this, of course, is to compare the components: $3 = a + b$ and $-2 = a - b$; we can then solve this simple system for the a and b. In higher dimensions this way of doing it becomes harder, however. A second way is to make use of the dot product and orthogonality.

The dot product of two vectors (x, y) and (w, z) in R^2 is $(x, y) \cdot (w, z) = xw+yz$. If the dot product is zero then the vectors are said to be *orthogonal*; the two vectors $(1, 1)$ and $(1, -1)$ are orthogonal. We take the dot product of both sides of $(3, -2) = a(1, 1) + b(1, -1)$ with $(1, 1)$ to get

$$1 = (3, -2) \cdot (1, 1) = a(1, 1) \cdot (1, 1) + b(1, -1) \cdot (1, 1) = a(1, 1) \cdot (1, 1) + 0 = 2a,$$

so we see that $a = \frac{1}{2}$. Similarly, taking the dot product of both sides with $(1, -1)$ gives

$$5 = (3, -2) \cdot (1, -1) = a(1, 1) \cdot (1, -1) + b(1, -1) \cdot (1, -1) = 2b,$$

so $b = \frac{5}{2}$. Therefore, $(3, -2) = \frac{1}{2}(1, 1) + \frac{5}{2}(1, -1)$. The beauty of this approach is that it does not get much harder as we go to higher dimensions.

Since the cosine of the angle θ between vectors $\mathbf{u}$ and $\mathbf{v}$ is

$$\cos\theta = \mathbf{u} \cdot \mathbf{v}/||\mathbf{u}||\,||\mathbf{v}||,$$

where $||\mathbf{u}||^2 = \mathbf{u} \cdot \mathbf{u}$, the projection of vector $\mathbf{v}$ on to the line through the origin parallel to $\mathbf{u}$ is

$$\text{Proj}_{\mathbf{u}}(\mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\mathbf{u} \cdot \mathbf{u}}\mathbf{u}.$$

Therefore, the vector $\mathbf{v}$ can be written as

$$\mathbf{v} = \text{Proj}_{\mathbf{u}}(\mathbf{v}) + (\mathbf{v} - \text{Proj}_{\mathbf{u}}(\mathbf{v})),$$

where the first term on the right is parallel to $\mathbf{u}$ and the second one is orthogonal to $\mathbf{u}$.

How do we find vectors that are mutually orthogonal? Suppose we begin with $(1, 1)$. Take a second vector, say $(1, 2)$, that is not parallel to $(1, 1)$ and write it as we did $\mathbf{v}$ earlier, that is, as a sum of two vectors, one parallel to $(1, 1)$ and the second orthogonal to $(1, 1)$. The projection of $(1, 2)$ onto the line parallel to $(1, 1)$ passing through the origin is

$$\frac{(1,1) \cdot (1,2)}{(1,1) \cdot (1,1)}(1,1) = \frac{3}{2}(1,1) = (\frac{3}{2}, \frac{3}{2})$$

so

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

The vectors $(-\frac{1}{2}, \frac{1}{2}) = -\frac{1}{2}(1, -1)$ and, therefore, $(1, -1)$ are then orthogonal to $(1, 1)$. This approach is the basis for the *Gram-Schmidt* method for constructing a set of mutually orthogonal vectors.

21.3 Generalizing the Dot Product: Inner Products

The proof of Cauchy's Inequality rests not on the actual definition of the complex vector dot product, but rather on four of its most basic properties. We use these properties to extend the concept of the complex vector dot product to that of *inner product*. Later in this chapter we shall give several examples of inner products, applied to a variety of mathematical objects, including infinite sequences, functions, random variables, and matrices. For now, let us denote our mathematical objects by $\mathbf{u}$ and $\mathbf{v}$ and the inner product between them as $\langle \mathbf{u}, \mathbf{v} \rangle$. The objects will then be said to be members of an *inner-product space*. We are interested in inner products because they provide a notion of orthogonality, which is fundamental to best approximation and optimal estimation.

21.3.1 Defining an Inner Product and Norm

The four basic properties that will serve to define an inner product are:

- **1:** $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$, with equality if and only if $\mathbf{u} = \mathbf{0}$;
- **2:** $\langle \mathbf{v}, \mathbf{u} \rangle = \overline{\langle \mathbf{u}, \mathbf{v} \rangle}$;
- **3:** $\langle \mathbf{u}, \mathbf{v} + \mathbf{w} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle + \langle \mathbf{u}, \mathbf{w} \rangle$;
- **4:** $\langle c\mathbf{u}, \mathbf{v} \rangle = c\langle \mathbf{u}, \mathbf{v} \rangle$ for any complex number c.

The inner product is the basic ingredient in Hilbert space theory. Using the inner product, we define the *norm* of $\mathbf{u}$ to be

$$||\mathbf{u}|| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$$

and the distance between $\mathbf{u}$ and $\mathbf{v}$ to be $||\mathbf{u} - \mathbf{v}||$.

The Cauchy-Schwarz Inequality: Because these four properties were all we needed to prove the Cauchy inequality for the complex vector dot product, we obtain the same inequality whenever we have an inner product. This more general inequality is the Cauchy-Schwarz Inequality:

$$|\langle \mathbf{u}, \mathbf{v} \rangle| \leq \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle} \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}$$

or

$$|\langle \mathbf{u}, \mathbf{v} \rangle| \leq ||\mathbf{u}|| \, ||\mathbf{v}||,$$

with equality if and only if there is a scalar c such that $\mathbf{v} = c\mathbf{u}$. We say that the vectors $\mathbf{u}$ and $\mathbf{v}$ are *orthogonal* if $\langle \mathbf{u}, \mathbf{v} \rangle = 0$. We turn now to some examples.

21.3.2 Some Examples of Inner Products

Here are several examples of inner products.

- **Inner product of infinite sequences:** Let $\mathbf{u} = \{u_n\}$ and $\mathbf{v} = \{v_n\}$ be infinite sequences of complex numbers. The inner product is then

  $$\langle \mathbf{u}, \mathbf{v} \rangle = \sum u_n \overline{v_n},$$

  and

  $$||\mathbf{u}|| = \sqrt{\sum |u_n|^2}.$$

  The sums are assumed to be finite; the index of summation n is singly or doubly infinite, depending on the context. The Cauchy-Schwarz inequality says that

  $$|\sum u_n \overline{v_n}| \leq \sqrt{\sum |u_n|^2} \sqrt{\sum |v_n|^2}.$$

- **Inner product of functions:** Now suppose that $\mathbf{u} = f(x)$ and $\mathbf{v} = g(x)$. Then,

$$\langle \mathbf{u}, \mathbf{v} \rangle = \int f(x)\overline{g(x)}dx$$

and

$$||\mathbf{u}|| = \sqrt{\int |f(x)|^2 dx}.$$

The integrals are assumed to be finite; the limits of integration depend on the support of the functions involved. The Cauchy-Schwarz inequality now says that

$$|\int f(x)\overline{g(x)}dx| \leq \sqrt{\int |f(x)|^2 dx}\sqrt{\int |g(x)|^2 dx}.$$

- **Inner product of random variables:** Now suppose that $\mathbf{u} = X$ and $\mathbf{v} = Y$ are random variables. Then,

$$\langle \mathbf{u}, \mathbf{v} \rangle = E(X\overline{Y})$$

and

$$||\mathbf{u}|| = \sqrt{E(|X|^2)},$$

which is the standard deviation of X if the mean of X is zero. The expected values are assumed to be finite. The Cauchy-Schwarz inequality now says that

$$|E(X\overline{Y})| \leq \sqrt{E(|X|^2)}\sqrt{E(|Y|^2)}.$$

If $E(X) = 0$ and $E(Y) = 0$, the random variables X and Y are orthogonal if and only if they are *uncorrelated*.

- **Inner product of complex matrices:** Now suppose that $\mathbf{u} = A$ and $\mathbf{v} = B$ are complex matrices. Then,

$$\langle \mathbf{u}, \mathbf{v} \rangle = \text{trace}(B^\dagger A)$$

and

$$||\mathbf{u}|| = \sqrt{\text{trace}(A^\dagger A)},$$

where the trace of a square matrix is the sum of the entries on the main diagonal. As we shall see later, this inner product is simply the complex vector dot product of the vectorized versions of the matrices involved. The Cauchy-Schwarz inequality now says that

$$|\text{trace}(B^\dagger A)| \leq \sqrt{\text{trace}(A^\dagger A)}\sqrt{\text{trace}(B^\dagger B)}.$$

- **Weighted inner product of complex vectors:** Let $\mathbf{u}$ and $\mathbf{v}$ be complex vectors and let Q be a Hermitian positive-definite matrix; that is, $Q^\dagger = Q$ and $\mathbf{u}^\dagger Q\mathbf{u} > 0$ for all nonzero vectors $\mathbf{u}$. The inner product is then

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{v}^\dagger Q\mathbf{u}$$

and

$$||\mathbf{u}|| = \sqrt{\mathbf{u}^\dagger Q\mathbf{u}}.$$

We know from the eigenvector decomposition of Q that $Q = C^\dagger C$ for some matrix C. Therefore, the inner product is simply the complex vector dot product of the vectors $C\mathbf{u}$ and $C\mathbf{v}$. The Cauchy-Schwarz inequality says that

$$|\mathbf{v}^\dagger Q\mathbf{u}| \leq \sqrt{\mathbf{u}^\dagger Q\mathbf{u}}\sqrt{\mathbf{v}^\dagger Q\mathbf{v}}.$$

- **Weighted inner product of functions:** Now suppose that $\mathbf{u} = f(x)$ and $\mathbf{v} = g(x)$ and $w(x) > 0$. Then define

$$\langle \mathbf{u}, \mathbf{v} \rangle = \int f(x)\overline{g(x)}w(x)dx$$

and

$$||\mathbf{u}|| = \sqrt{\int |f(x)|^2 w(x)dx}.$$

The integrals are assumed to be finite; the limits of integration depend on the support of the functions involved. This inner product is simply the inner product of the functions $f(x)\sqrt{w(x)}$ and $g(x)\sqrt{w(x)}$. The Cauchy-Schwarz inequality now says that

$$|\int f(x)\overline{g(x)}w(x)dx| \leq \sqrt{\int |f(x)|^2 w(x)dx}\sqrt{\int |g(x)|^2 w(x)dx}.$$

Once we have an inner product defined, we can speak about orthogonality and best approximation. Important in that regard is the orthogonality principle.

21.4 Best Approximation and the Orthogonality Principle

Imagine that you are standing and looking down at the floor. The point B on the floor that is closest to N, the tip of your nose, is the unique

point on the floor such that the vector from B to any other point A on the floor is perpendicular to the vector from N to B; that is, $\langle BN, BA\rangle = 0$. This is a simple illustration of the *orthogonality principle*. Whenever we have an inner product defined we can speak of orthogonality and apply the orthogonality principle to find best approximations. For notational simplicity, we shall consider only real inner product spaces.

21.4.1 Best Approximation

Let $\mathbf{u}$ and $\mathbf{v}^1, ..., \mathbf{v}^N$ be members of a real inner-product space. For all choices of scalars $a_1, ..., a_N$, we can compute the distance from $\mathbf{u}$ to the member $a_1\mathbf{v}^1 + ...a_N\mathbf{v}^N$. Then, we minimize this distance over all choices of the scalars; let $b_1, ..., b_N$ be this best choice.

The distance squared from $\mathbf{u}$ to $a_1\mathbf{v}^1 + ...a_N\mathbf{v}^N$ is

$$||\mathbf{u} - (a_1\mathbf{v}^1 + ...a_N\mathbf{v}^N)||^2 = \langle \mathbf{u} - (a_1\mathbf{v}^1 + ...a_N\mathbf{v}^N), \mathbf{u} - (a_1\mathbf{v}^1 + ...a_N\mathbf{v}^N)\rangle,$$

$$= ||\mathbf{u}||^2 - 2\langle \mathbf{u}, \sum_{n=1}^{N} a_n\mathbf{v}^n\rangle + \sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m \langle \mathbf{v}^n\mathbf{v}^m\rangle.$$

Setting the partial derivative with respect to a_n equal to zero, we have

$$\langle \mathbf{u}, \mathbf{v}^n\rangle = \sum_{m=1}^{N} a_m \langle \mathbf{v}^m\mathbf{v}^n\rangle.$$

With $\mathbf{a} = (a_1, ..., a_N)^T$,

$$\mathbf{d} = (\langle \mathbf{u}, \mathbf{v}^1\rangle, ..., \langle \mathbf{u}, \mathbf{v}^N\rangle)^T$$

and V the matrix with entries

$$V_{mn} = \langle \mathbf{v}^m, \mathbf{v}^n\rangle,$$

we find that we must solve the system of equations $V\mathbf{a} = \mathbf{d}$. When the vectors $\mathbf{v}^n$ are mutually orthogonal and each has norm equal to one, then $V = I$, the identity matrix, and the desired vector $\mathbf{a}$ is simply $\mathbf{d}$.

21.4.2 The Orthogonality Principle

The *orthogonality principle* provides another way to view the calculation of the best approximation: let the best approximation of $\mathbf{u}$ be the vector

$$\hat{\mathbf{v}} = b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N.$$

Then

$$\langle \mathbf{u} - \hat{\mathbf{v}}, \mathbf{v}^n\rangle = \langle \mathbf{u} - (b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N), \mathbf{v}^n\rangle = 0,$$

for $n = 1, 2, ..., N$. This leads directly to the system of equations

$$\mathbf{d} = V\mathbf{b},$$

which, as we just saw, provides the optimal coefficients.

To see why the orthogonality principle is valid, fix a value of n and consider the problem of minimizing the distance

$$||\mathbf{u} - (b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N + \alpha\mathbf{v}^n)||$$

as a function of α. Writing the norm squared in terms of the inner product, expanding the terms, and differentiating with respect to α, we find that the minimum occurs when

$$\alpha = \langle \mathbf{u} - b_1\mathbf{v}^1 + ...b_N\mathbf{v}^N, \mathbf{v}^n \rangle.$$

But we already know that the minimum occurs when $\alpha = 0$. This completes the proof of the orthogonality principle.

21.5 Gram-Schmidt Orthogonalization

We have seen that the best approximation is easily calculated if the vectors $\mathbf{v}^n$ are mutually orthogonal. But how do we get such a mutually orthogonal set, in general? The Gram-Schmidt Orthogonalization Method is one way to proceed.

Let $\{\mathbf{v}^1, ..., \mathbf{v}^N\}$ be a linearly independent set of vectors in the space R^M, where $N \leq M$. The Gram-Schmidt method uses the $\mathbf{v}^n$ to create an orthogonal basis $\{\mathbf{u}^1, ..., \mathbf{u}^N\}$ for the span of the $\mathbf{v}^n$. Begin by taking $\mathbf{u}^1 = \mathbf{v}^1$. For $j = 2, ..., N$, let

$$\mathbf{u}^j = \mathbf{v}^j - \frac{\mathbf{u}^1 \cdot \mathbf{v}^j}{\mathbf{u}^1 \cdot \mathbf{u}^1}\mathbf{u}^1 - ... - \frac{\mathbf{u}^{j-1} \cdot \mathbf{v}^j}{\mathbf{u}^{j-1} \cdot \mathbf{u}^{j-1}}\mathbf{u}^{j-1}. \tag{21.1}$$

One obvious problem with this approach is that the calculations become increasingly complicated and lengthy as the j increases. In many of the important examples of orthogonal functions that we study in connection with Sturm-Liouville problems, there is a two-term recursive formula that enables us to generate the next orthogonal function from the two previous ones.

Chapter 22

Appendix: Conjugate-Direction Algorithms

Finding the least-squares solution of a possibly inconsistent system of linear equations $Ax = b$ is equivalent to minimizing the quadratic function $f(x) = \frac{1}{2}||Ax - b||_2^2$ and so can be viewed within the framework of optimization. Iterative optimization methods can then be used to provide, or at least suggest, algorithms for obtaining the least-squares solution. The *conjugate gradient method* is one such method.

22.1 Iterative Minimization

Iterative methods for minimizing a real-valued function $f(x)$ over the vector variable x usually take the following form: having obtained x^{k-1}, a new direction vector d^k is selected, an appropriate scalar $\alpha_k > 0$ is determined and the next member of the iterative sequence is given by

$$x^k = x^{k-1} + \alpha_k d^k. \tag{22.1}$$

Ideally, one would choose the α_k to be the value of α for which the function $f(x^{k-1} + \alpha d^k)$ is minimized. It is assumed that the direction d^k is a *descent direction*; that is, for small positive α the function $f(x^{k-1} + \alpha d^k)$ is strictly decreasing. Finding the optimal value of α at each step of the iteration is difficult, if not impossible, in most cases, and approximate methods, using line searches, are commonly used.

Lemma 22.1 *For each k we have*

$$\nabla f(x^k) \cdot d^k = 0. \tag{22.2}$$

Proof: Differentiate the function $f(x^{k-1}+\alpha d^k)$ with respect to the variable α. ∎

Since the gradient $\nabla f(x^k)$ is orthogonal to the previous direction vector d^k and also because $-\nabla f(x)$ is the direction of greatest decrease of $f(x)$, the choice of $d^{k+1} = -\nabla f(x^k)$ as the next direction vector is a reasonable one. With this choice we obtain Cauchy's *steepest descent algorithm* [115]:

Algorithm 22.1 (Steepest Descent) *Let x^0 be arbitrary. Then let*

$$x^{k+1} = x^k - \alpha_{k+1}\nabla f(x^k). \tag{22.3}$$

The steepest descent method need not converge in general and even when it does, it can do so slowly, suggesting that there may be better choices for the direction vectors. For example, the Newton-Raphson method [122] employs the following iteration:

$$x^{k+1} = x^k - \nabla^2 f(x^k)^{-1}\nabla f(x^k), \tag{22.4}$$

where $\nabla^2 f(x)$ is the Hessian matrix for $f(x)$ at x. To investigate further the issues associated with the selection of the direction vectors, we consider the more tractable special case of quadratic optimization.

22.2 Quadratic Optimization

Let A be an arbitrary real I by J matrix. The linear system of equations $Ax = b$ need not have any solutions, and we may wish to find a least-squares solution $x = \hat{x}$ that minimizes

$$f(x) = \frac{1}{2}||b - Ax||_2^2. \tag{22.5}$$

The vector b can be written

$$b = A\hat{x} + \hat{w}, \tag{22.6}$$

where $A^T\hat{w} = 0$ and a least squares solution is an exact solution of the linear system $Qx = c$, with $Q = A^TA$ and $c = A^Tb$. We shall assume that Q is invertible and there is a unique least squares solution; this is the typical case.

We consider now the iterative scheme described by Equation (22.1) for $f(x)$ as in Equation (22.5). For this $f(x)$ the gradient becomes

$$\nabla f(x) = Qx - c. \tag{22.7}$$

The optimal α_k for the iteration can be obtained in closed form.

Lemma 22.2 *The optimal α_k is*

$$\alpha_k = \frac{r^k \cdot d^k}{d^k \cdot Qd^k}, \tag{22.8}$$

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

Lemma 22.3 *Let $||x||_Q^2 = x \cdot Qx$ denote the square of the Q-norm of x. Then*

$$||\hat{x} - x^{k-1}||_Q^2 - ||\hat{x} - x^k||_Q^2 = (r^k \cdot d^k)^2 / d^k \cdot Qd^k \geq 0 \tag{22.9}$$

for any direction vectors d^k.

If the sequence of direction vectors $\{d^k\}$ is completely general, the iterative sequence need not converge. However, if the set of direction vectors is finite and spans R^J and we employ them cyclically, convergence follows.

Theorem 22.1 *Let $\{d^1, ..., d^J\}$ be any finite set whose span is all of R^J. Let α_k be chosen according to Equation (22.8). Then, for $k = 0, 1, ...$, $j = k(\mathrm{mod}\, J) + 1$, and any x^0, the sequence defined by*

$$x^k = x^{k-1} + \alpha_k d^j \tag{22.10}$$

converges to the least squares solution.

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

$$x^{*,j} = x^{*,j-1} + \frac{(c - Qx^{*,j-1}) \cdot d^j}{d^j \cdot Qd^j} d^j. \tag{22.11}$$

Since

$$r^{mJ+j} \cdot d^j \to 0, \tag{22.12}$$

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

$$(c - Qx^{*,j}) \cdot d^j = 0. \tag{22.13}$$

Therefore,

$$x^{*,1} = ... = x^{*,J} = x^* \tag{22.14}$$

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

There is an interesting corollary to this theorem that pertains to a modified version of the ART algorithm. For $k = 0, 1, ...$ and $i = k(\mathrm{mod}\, M) + 1$ and with the rows of A normalized to have length one, the ART iterative step is

$$x^{k+1} = x^k + (b_i - (Ax^k)_i)a^i, \tag{22.15}$$

where a^i is the ith column of A^T. When $Ax = b$ has no solutions, the ART algorithm does not converge to the least-squares solution; rather, it exhibits subsequential convergence to a limit cycle. However, using the previous theorem, we can show that the following modification of the ART, which we shall call the *least squares ART* (LS-ART), converges to the least-squares solution for every x^0:

$$x^{k+1} = x^k + \frac{r^{k+1} \cdot a^i}{a^i \cdot Qa^i} a^i. \tag{22.16}$$

In the quadratic case the steepest descent iteration has the form

$$x^k = x^{k-1} + \frac{r^k \cdot r^k}{r^k \cdot Qr^k} r^k. \tag{22.17}$$

We have the following result.

Theorem 22.2 *The steepest descent method converges to the least-squares solution.*

Proof: As in the proof of the previous theorem, we have

$$||\hat{x} - x^{k-1}||_Q^2 - ||\hat{x} - x^k||_Q^2 = (r^k \cdot d^k)^2 / d^k \cdot Qd^k \geq 0, \tag{22.18}$$

where now the direction vectors are $d^k = r^k$. So, the sequence $\{||\hat{x} - x^k||_Q^2\}$ is decreasing, and therefore the sequence $\{(r^k \cdot r^k)^2 / r^k \cdot Qr^k\}$ must converge to zero. The sequence $\{x^k\}$ is bounded; let x^* be a cluster point. It follows that $c - Qx^* = 0$, so that x^* is the least-squares solution $\hat{x}$. The rest of the proof follows as in the proof of the previous theorem. ∎

22.3 Conjugate Bases for R^J

If the set $\{v^1, ..., v^J\}$ is a basis for R^J, then any vector x in R^J can be expressed as a linear combination of the basis vectors; that is, there are real numbers $a_1, ..., a_J$ for which

$$x = a_1 v^1 + a_2 v^2 + ... + a_J v^J. \tag{22.19}$$

For each x the coefficients a_j are unique. To determine the a_j we write

$$x \cdot v^m = a_1 v^1 \cdot v^m + a_2 v^2 \cdot v^m + ... + a_J v^J \cdot v^m, \tag{22.20}$$

for $m = 1, ..., M$. Having calculated the quantities $x \cdot v^m$ and $v^j \cdot v^m$, we solve the resulting system of linear equations for the a_j.

If the set $\{u^1, ..., u^M\}$ is an orthogonal basis, that is, then $u^j \cdot u^m = 0$, unless $j = m$, then the system of linear equations is now trivial to solve. The solution is $a_j = x \cdot u^j / u^j \cdot u^j$, for each j. Of course, we still need to compute the quantities $x \cdot u^j$.

The least-squares solution of the linear system of equations $Ax = b$ is

$$\hat{x} = (A^T A)^{-1} A^T b = Q^{-1} c. \tag{22.21}$$

To express $\hat{x}$ as a linear combination of the members of an orthogonal basis $\{u^1, ..., u^J\}$ we need the quantities $\hat{x} \cdot u^j$, which usually means that we need to know $\hat{x}$ first. For a special kind of basis, a *Q-conjugate basis*, knowing $\hat{x}$ ahead of time is not necessary; we need only know Q and c. Therefore, we can use such a basis to find $\hat{x}$. This is the essence of the *conjugate gradient method* (CGM), in which we calculate a conjugate basis and, in the process, determine $\hat{x}$.

22.3.1 Conjugate Directions

From Equation (22.2) we have

$$(c - Qx^{k+1}) \cdot d^k = 0, \tag{22.22}$$

which can be expressed as

$$(\hat{x} - x^{k+1}) \cdot Qd^k = (\hat{x} - x^{k+1})^T Qd^k = 0. \tag{22.23}$$

Definition 22.1 *Two vectors x and y are said to be Q-orthogonal (or Q-conjugate, or just conjugate), if $x \cdot Qy = 0$.*

So, the least-squares solution that we seek lies in a direction from x^{k+1} that is Q-orthogonal to d^k. This suggests that we can do better than steepest descent if we take the next direction to be Q-orthogonal to the previous one, rather than just orthogonal. This leads us to *conjugate direction methods.*

Lemma 22.4 *Say that the set $\{p^1, ..., p^n\}$ is a conjugate set for R^J if $p^i \cdot Qp^j = 0$ for $i \neq j$. Any conjugate set that does not contain zero is linearly independent. If $p^n \neq 0$ for $n = 1, ..., J$, then the least-squares vector $\hat{x}$ can be written as*

$$\hat{x} = a_1 p^1 + ... + a_J p^J, \tag{22.24}$$

with $a_j = c \cdot p^j / p^j \cdot Qp^j$ for each j.

Proof: Use the Q-inner product $\langle x, y \rangle_Q = x \cdot Qy$. ∎

Therefore, once we have a conjugate basis, computing the least squares solution is trivial. Generating a conjugate basis can obviously be done using the standard Gram-Schmidt approach.

22.3.2 The Gram-Schmidt Method

Let $\{v^1, ..., v^J\}$ be a linearly independent set of vectors in the space R^M, where $J \leq M$. The Gram-Schmidt method uses the v^j to create an orthogonal basis $\{u^1, ..., u^J\}$ for the span of the v^j. Begin by taking $u^1 = v^1$. For $j = 2, ..., J$, let

$$u^j = v^j - \frac{u^1 \cdot v^j}{u^1 \cdot u^1} u^1 - ... - \frac{u^{j-1} \cdot v^j}{u^{j-1} \cdot u^{j-1}} u^{j-1}. \tag{22.25}$$

To apply this approach to obtain a conjugate basis, we would simply replace the dot products $u^k \cdot v^j$ and $u^k \cdot u^k$ with the Q-inner products, that is,

$$p^j = v^j - \frac{p^1 \cdot Qv^j}{p^1 \cdot Qp^1} p^1 - ... - \frac{p^{j-1} \cdot Qv^j}{p^{j-1} \cdot Qp^{j-1}} p^{j-1}. \tag{22.26}$$

Even though the Q-inner products can always be written as $x \cdot Qy = Ax \cdot Ay$, so that we need not compute the matrix Q, calculating a conjugate basis using Gram-Schmidt is not practical for large J. There is a way out, fortunately.

If we take $p^1 = v^1$ and $v^j = Qp^{j-1}$, we have a much more efficient mechanism for generating a conjugate basis, namely a three-term recursion formula [115]. The set $\{p^1, Qp^1, ..., Qp^{J-1}\}$ need not be a linearly independent set, in general, but, if our goal is to find $\hat{x}$, and not really to calculate a full conjugate basis, this does not matter, as we shall see.

Theorem 22.3 *Let $p^1 \neq 0$ be arbitrary. Let p^2 be given by*

$$p^2 = Qp^1 - \frac{Qp^1 \cdot Qp^1}{p^1 \cdot Qp^1} p^1, \tag{22.27}$$

so that $p^2 \cdot Qp^1 = 0$. *Then, for* $n \geq 2$, *let* p^{n+1} *be given by*

$$p^{n+1} = Qp^n - \frac{Qp^n \cdot Qp^n}{p^n \cdot Qp^n} p^n - \frac{Qp^{n-1} \cdot Qp^n}{p^{n-1} \cdot Qp^{n-1}} p^{n-1}. \tag{22.28}$$

Then, the set $\{p^1, ..., p^J\}$ *is a conjugate set for* R^J. *If* $p^n \neq 0$ *for each* n, *then the set is a conjugate basis for* R^J.

Proof: We consider the induction step of the proof. Assume that $\{p^1, ..., p^n\}$ is a Q-orthogonal set of vectors; we then show that $\{p^1, ..., p^{n+1}\}$ is also, provided that $n \leq J - 1$. It is clear from Equation (22.28) that

$$p^{n+1} \cdot Qp^n = p^{n+1} \cdot Qp^{n-1} = 0. \tag{22.29}$$

For $j \leq n - 2$, we have

$$p^{n+1} \cdot Qp^j = p^j \cdot Qp^{n+1} = p^j \cdot Q^2 p^n - ap^j \cdot Qp^n - bp^j \cdot Qp^{n-1}, \tag{22.30}$$

for constants a and b. The second and third terms on the right side are then zero because of the induction hypothesis. The first term is also zero since

$$p^j \cdot Q^2 p^n = (Qp^j) \cdot Qp^n = 0 \tag{22.31}$$

because Qp^j is in the span of $\{p^1, ..., p^{j+1}\}$, and so is Q-orthogonal to p^n. ∎

The calculations in the three-term recursion formula Equation (22.28) also occur in the Gram-Schmidt approach in Equation (22.26); the point is that Equation (22.28) uses only the first three terms, in every case.

22.4 The Conjugate Gradient Method

The main idea in the *conjugate gradient method* (CGM) is to build the conjugate set as we calculate the least squares solution using the iterative algorithm

$$x^n = x^{n-1} + \alpha_n p^n. \tag{22.32}$$

The α_n is chosen so as to minimize the function of α defined by $f(x^{n-1} + \alpha p^n)$, and so we have

$$\alpha_n = \frac{r^n \cdot p^n}{p^n \cdot Qp^n}, \tag{22.33}$$

where $r^n = c - Qx^{n-1}$. Since the function $f(x) = \frac{1}{2}||Ax - b||_2^2$ has for its gradient $\nabla f(x) = A^T(Ax - b) = Qx - c$, the residual vector $r^n =$

$c - Qx^{n-1}$ is the direction of steepest descent from the point $x = x^{n-1}$. The CGM combines the use of the negative gradient directions from the steepest descent method with the use of a conjugate basis of directions, by using the r^{n+1} to construct the next direction p^{n+1} in such a way as to form a conjugate set $\{p_1, ..., p^J\}$.

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

$$p^{n+1} = r^{n+1} - \frac{r^{n+1} \cdot Qp^n}{p^n \cdot Qp^n} p^n. \tag{22.34}$$

Since the α_n is the optimal choice and

$$r^{n+1} = -\nabla f(x^n), \tag{22.35}$$

we have, according to Equation (22.2),

$$r^{n+1} \cdot p^n = 0. \tag{22.36}$$

Lemma 22.5 *For all n, $r^{n+1} = 0$ whenever $p^{n+1} = 0$, in which case we have $c = Qx^n$, so that x^n is the least-squares solution.*

In theory, the CGM converges to the least squares solution in finitely many steps, since we either reach $p^{n+1} = 0$ or $n + 1 = J$. In practice, the CGM can be employed as a fully iterative method by cycling back through the previously used directions.

An induction proof similar to the one used to prove Theorem 22.3 establishes that the set $\{p^1, ..., p^J\}$ is a conjugate set [115, 122]. In fact, we can say more.

Theorem 22.4 *For $n = 1, 2, ..., J$ and $j = 1, ..., n - 1$ we have*

- a) $r^n \cdot r^j = 0$;
- b) $r^n \cdot p^j = 0$; *and*
- c) $p^n \cdot Qp^j = 0$.

The proof presented here through a series of lemmas is based on that given in [122].

The proof uses induction on the number n. Throughout the following lemmas assume that the statements in the theorem hold for some $n < J$. We prove that they hold also for $n + 1$.

Lemma 22.6 *The vector Qp^j is in the span of the vectors r^j and r^{j+1}.*

Proof: Use the fact that

$$r^{j+1} = r^j - \alpha_j Qp^j. \tag{22.37}$$

∎

Lemma 22.7 *For each n, $r^{n+1} \cdot r^n = 0$.*

Proof: Establish that

$$\alpha_n = \frac{r^n \cdot r^n}{p^n \cdot Qp^n}. \tag{22.38}$$

∎

Lemma 22.8 *For $j = 1, ..., n-1$, $r^{n+1} \cdot r^j = 0$.*

Proof: Use the induction hypothesis. ∎

Lemma 22.9 *For $j = 1, ..., n$, $r^{n+1} \cdot p^j = 0$.*

Proof: First, establish that

$$p^j = r^j - \beta_{j-1} p^{j-1}, \tag{22.39}$$

where

$$\beta_{j-1} = \frac{r^j \cdot Qp^{j-1}}{p^{j-1} \cdot Qp^{j-1}}, \tag{22.40}$$

and

$$r^{n+1} = r^n - \alpha_n Qp^n. \tag{22.41}$$

∎

Lemma 22.10 *For $j = 1, ..., n-1$, $p^{n+1} \cdot Qp^j = 0$.*

Proof: Use

$$Qp^j = \alpha_j^{-1}(r^j - r^{j+1}). \tag{22.42}$$

∎

The final step in the proof is contained in the following lemma.

Lemma 22.11 *For each n, we have $p^{n+1} \cdot Qp^n = 0$.*

Proof: Establish that

$$\beta_n = -\frac{r^{n+1} \cdot r^{n+1}}{r^n \cdot r^n}. \tag{22.43}$$

∎

The convergence rate of the CGM depends on the condition number of the matrix Q, which is the ratio of its largest to its smallest eigenvalues. When the condition number is much greater than one convergence can be accelerated by *preconditioning* the matrix Q; this means replacing Q with $P^{-1/2}QP^{-1/2}$, for some positive-definite approximation P of Q (see [7]).

There are versions of the CGM for the minimization of nonquadratic functions. In the quadratic case the next conjugate direction p^{n+1} is built from the residual r^{n+1} and p^n. Since, in that case, $r^{n+1} = -\nabla f(x^n)$, this suggests that in the nonquadratic case we build p^{n+1} from $-\nabla f(x^n)$ and p^n. This leads to the Fletcher-Reeves method. Other similar algorithms, such as the Polak-Ribiere and the Hestenes-Stiefel methods, perform better on certain problems [122].

Chapter 23

Appendix: Quadratic Programming

The *quadratic-programming* problem (QP) is to minimize a quadratic function, subject to inequality constraints and nonnegativity of the variables. Using the Karush-Kuhn-Tucker Theorem 8.6 for mixed constraints and introducing slack variables, this problem can be reformulated as a linear programming problem and solved by Wolfe's Algorithm [129], a variant of the simplex method. In the case of general constrained optimization, the Newton-Raphson method for finding a stationary point of the Lagrangian can be viewed as solving a sequence of quadratic programming problems. This leads to *sequential quadratic programming* [122].

23.1 The Quadratic-Programming Problem

The primal QP problem is to minimize the quadratic function

$$f(x) = a + c^T x + \frac{1}{2} x^T Q x, \tag{23.1}$$

subject to the constraints

$$Ax \leq b, \tag{23.2}$$

and $x_j \geq 0$, for $j = 1, ..., J$. Here a, b, and c are given, Q is a J by J positive-definite matrix with entries q_{ij}, and A is an I by J matrix with rank I and entries a_{ij}. To allow for some equality constraints, we say that

$$(Ax)_i \leq b_i, \tag{23.3}$$

for $i = 1, ..., K$, and

$$(Ax)_i = b_i, \tag{23.4}$$

for $i = K+1, ..., I$.

We incorporate the nonnegativity constraints $x_j \geq 0$ by requiring

$$-x_j \leq 0, \tag{23.5}$$

for $j = 1, ..., J$. Applying the KKT Theorem 8.6 to this problem, we find that if a regular point x^* is a solution, then there are vectors μ^* and ν^* such that

- **1)** $\mu_i^* \geq 0$, for $i = 1, ..., K$;
- **2)** $\nu_j^* \geq 0$, for $j = 1, ..., J$;
- **3)** $c + Qx^* + A^T\mu^* - v^* \geq 0$;
- **4)** $\mu_i^*((Ax^*)_i - b_i) = 0$, for $i = 1, ..., I$;
- **5)** $x_j^*\nu_j^* = 0$, for $j = 1, ..., J$.

One way to solve this problem is to reformulate it as a linear-programming problem. To that end, we introduce slack variables x_{J+i}, $i = 1, ..., K$, and write the problem as

$$\sum_{j=1}^{J} a_{ij}x_j + x_{J+i} = b_i, \tag{23.6}$$

for $i = 1, ..., K$,

$$\sum_{j=1}^{J} a_{ij}x_j = b_i, \tag{23.7}$$

for $i = K+1, ..., I$,

$$\sum_{j=1}^{J} q_{mj}x_j + \sum_{i=1}^{I} a_{im}\mu_i - \nu_m = -c_m, \tag{23.8}$$

for $m = 1, ..., J$,

$$\mu_i x_{J+i} = 0, \tag{23.9}$$

for $i = 1, ..., K$, and

$$x_j\nu_j = 0, \tag{23.10}$$

for $j = 1, ..., J$. The objective now is to formulate the problem as a primal linear-programming problem in standard form.

The variables x_j, $j = 1, ..., J$, x_{J+i}, $i = 1, ..., K$, and ν_j, $j = 1, ..., J$ must be nonnegative; the variables μ_i are unrestricted, for $i = K + 1, ..., I$, so we write

$$\mu_i = \mu_i^+ - \mu_i^-, \tag{23.11}$$

and require that both μ_i^+ and μ_i^- be nonnegative. Finally, we need a linear function to minimize.

We rewrite Equation (23.6) as

$$\sum_{j=1}^{J} a_{ij} x_j + x_{J+i} + y_i = b_i, \tag{23.12}$$

for $i = 1, ..., K$, Equation (23.7) as

$$\sum_{j=1}^{J} a_{ij} x_j + y_i = b_i, \tag{23.13}$$

for $i = K + 1, ..., I$, and Equation (23.8) as

$$\sum_{j=1}^{J} q_{mj} x_j + \sum_{i=1}^{I} a_{im} \mu_i - \nu_m + y_{I-K+m} = -c_m, \tag{23.14}$$

for $m = 1, ..., J$. Then the problem is to minimize the linear function

$$y_1 + ... + y_{I-K+J}, \tag{23.15}$$

over nonnegative y_i, subject to the equality constraints in the equations (23.12), (23.13), and (23.14). Any solution to the original problem must be a basic feasible solution to this primal linear-programming problem. Wolfe's Algorithm [129] is a modification of the simplex method that guarantees that we never have μ_i and x_{J+i} positive basic variables at the same time, nor x_j and ν_j.

23.2 Sequential Quadratic Programming

Consider once again the CP problem of minimizing the convex function $f(x)$, subject to $g_i(x) \leq 0$, for $i = 1, ..., I$. The Lagrangian is

$$L(x, \lambda) = f(x) + \sum_{i=1}^{I} \lambda_i g_i(x), \tag{23.16}$$

and stationary values of the Lagrangian must satisfy the equation

$$\nabla L(x, \lambda) = 0. \tag{23.17}$$

One step of the Newton-Raphson algorithm has the form

$$\begin{pmatrix} x^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ \lambda^k \end{pmatrix} + \begin{pmatrix} p^k \\ v^k \end{pmatrix}, \tag{23.18}$$

where

$$\begin{bmatrix} \nabla^2_{xx} L(x^k, \lambda^k) & \nabla g(x^k) \\ \nabla g(x^k)^T & 0 \end{bmatrix} \begin{pmatrix} p^k \\ v^k \end{pmatrix} = \begin{pmatrix} -\nabla_x L(x^k, \lambda^k) \\ -g(x^k) \end{pmatrix}. \tag{23.19}$$

The incremental vector $\begin{pmatrix} p^k \\ v^k \end{pmatrix}$ obtained by solving this system is also the solution to the quadratic-programming problem of minimizing the function

$$\frac{1}{2} p^T \nabla^2_{xx} L(x^k, \lambda^k) p + p^T \nabla_x L(x^k, \lambda^k), \tag{23.20}$$

subject to the constraint

$$\nabla g(x^k)^T p + g(x^k) = 0. \tag{23.21}$$

Therefore, the Newton-Raphson algorithm for the original minimization problem can be implemented as a sequence of quadratic programs, each solved by the methods discussed previously. In practice, variants of this approach that employ approximations for the first and second partial derivatives are often used.

Chapter 24

Appendix: Properties of Averaged Operators

We present the fundamental properties of averaged operators, leading to the proof that the class of averaged operators is closed to finite products. Throughout this chapter the term 'non-expansive' will always refer to the Euclidean norm.

24.1 General Properties of Averaged Operators

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

$$\frac{1}{\alpha}(A - (1 - \alpha)I) \tag{24.1}$$

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

Lemma 24.1 *Let $T = (1-\alpha)A+\alpha N$ for some $\alpha \in (0, 1)$. If A is averaged and N is non-expansive then T is averaged.*

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

$$T = (1 - \gamma)I + \gamma[(1 - \alpha)\beta\gamma^{-1}M + \alpha\gamma^{-1}N]. \tag{24.2}$$

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

Corollary 24.1 *If A and B are av and α is in the interval $[0,1]$, then the operator $T = (1-\alpha)A + \alpha B$ formed by taking the convex combination of A and B is av.*

Corollary 24.2 *Let $T = (1-\alpha)F + \alpha N$ for some $\alpha \in (0,1)$. If F is fne and N is ne then T is averaged.*

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

$$T = (1-\omega)I + \omega P_H, \tag{24.3}$$

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

$$P_H x = \frac{1}{2}(x + R_H x), \tag{24.4}$$

for each x.

Lemma 24.2 *The operator $R_H = 2P_H - I$ is an isometry; that is,*

$$||R_H x - R_H y||_2 = ||x - y||_2, \tag{24.5}$$

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

The proof is left as an exercise.

Lemma 24.3 *For $\omega = 1 + \gamma$ in the interval $[1,2)$, we have*

$$(1-\omega)I + \omega P_H = \alpha I + (1-\alpha)R_H, \tag{24.6}$$

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

Once again, the proof is left as an exercise.

24.2 The Main Result

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

If $A = (1-\alpha)I + \alpha N$ is averaged and B is averaged then $T = AB$ has the form $T = (1-\alpha)B + \alpha NB$. Since B is av and NB is ne, it follows from Lemma 10.8 that T is averaged. Summarizing, we have

Proposition 24.1 *If A and B are averaged, then $T = AB$ is averaged.*

It is possible for Fix(AB) to be nonempty while Fix(A)∩Fix(B) is empty; however, if the latter is nonempty, it must coincide with Fix(AB) [10]:

Proposition 24.2 *Let A and B be averaged operators and suppose that Fix(A)∩Fix(B) is nonempty. Then Fix(A)∩Fix(B) =Fix(AB)=Fix(BA).*

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

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

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

$$= ||z - x||_2^2 + (2\nu_B - 1)||BAx - Ax||_2^2 + (2\nu_A - 1)||Ax - x||_2^2. \quad (24.7)$$

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

24.3 Averaged Linear Operators

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

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

$$B = (1 - \alpha)I + \alpha N. \quad (24.8)$$

Therefore

$$N = \frac{1}{\alpha}B + (1 - \frac{1}{\alpha})I \quad (24.9)$$

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

Lemma 24.4 *A linear operator N is ne, in the Euclidean norm, if and only if $||N||_2 = \sqrt{\rho(N^\dagger N)}$, the matrix norm induced by the Euclidean vector norm, does not exceed one.*

The proof is left as an exercise.

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

$$Re(\langle (I-B)x, x\rangle) \geq \nu ||(I-B)x||_2^2, \tag{24.10}$$

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

$$x^\dagger(I-B)x = x^\dagger(I-Q)x, \tag{24.11}$$

for $Q = \frac{1}{2}(B + B^\dagger)$, it follows that $I - Q$ must be non-negative definite. Moreover, if B is av, then B is ne, so that $||B||_2 \leq 1$. Since $||B||_2 = ||B^\dagger||_2$, and $||Q||_2 \leq \frac{1}{2}(||B||_2 + ||B^\dagger||_2)$, it follows that Q must be Euclidean ne. In fact, since N is Euclidean ne if and only if $N^\dagger$ is, B is av if and only if $B^\dagger$ is av. Consequently, if the linear operator B is av, then so is the Hermitian operator Q, and so the eigenvalues of Q must lie in the interval $(-1, 1]$. We also know from Exercise 10.9 that, if B is av, then $|\lambda| < 1$, unless $\lambda = 1$, for every eigenvalue λ of B.

24.3.1 Hermitian Linear Operators

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

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

$$\nu = \frac{1}{\alpha}\lambda + (1 - \frac{1}{\alpha}). \tag{24.12}$$

It follows that $|\nu| \leq 1$ if and only if

$$1 - 2\alpha \leq \lambda \leq 1. \tag{24.13}$$

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

$$-1 < 1 - 2\alpha \leq \lambda \leq 1, \tag{24.14}$$

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

$$-1 < \lambda \leq 1, \tag{24.15}$$

for all eigenvalues λ of B. The choice

$$\alpha_0 = \frac{1 - \lambda_{min}}{2} \tag{24.16}$$

is the smallest α for which

$$N = \frac{1}{\alpha}B + (1 - \frac{1}{\alpha})I \tag{24.17}$$

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

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

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

24.4 Exercises

24.1 *Prove Lemma 24.2.*

24.2 *Prove Lemma 24.3.*

24.3 *Prove Lemma 24.4.*

Chapter 25

Appendix: Fenchel Duality

The duality between convex functions on R^J and their tangent hyperplanes is made explicit through the Legendre-Fenchel transformation. In this appendix we discuss this transformation, state and prove Fenchel's Duality Theorem, and investigate some of its applications.

25.1 The Legendre-Fenchel Transformation

Throughout this section $f : C \subseteq R^J \to R$ is a closed, proper, convex function defined on a non-empty, closed convex set C.

25.1.1 The Fenchel Conjugate

For each fixed vector a in R^J, the affine function $h(x) = \langle a, x\rangle - \gamma$ is beneath the function $f(x)$ if $f(x) - h(x) \geq 0$, for all x; that is,

$$f(x) - \langle a, x\rangle + \gamma \geq 0,$$

or

$$\gamma \geq \langle a, x\rangle - f(x). \tag{25.1}$$

This leads us to the following definition, involving the maximum of the right side of the inequality in (25.1), for each fixed a.

Definition 25.1 *The conjugate function associated with f is the function*

$$f^*(a) = \sup_x(\langle a, x\rangle - f(x)). \tag{25.2}$$

For each fixed a, the value $f^*(a)$ is the smallest value of γ for which the affine function $h(x) = \langle a, x\rangle - \gamma$ is beneath $f(x)$. The passage from f to f^* is the *Legendre-Fenchel Transformation*. Now we repeat this process with $f^*(a)$ in the role of $f(x)$.

25.1.2 The Conjugate of the Conjugate

For each fixed vector x, the affine function $c(a) = \langle a, x\rangle - \gamma$ is beneath the function $f^*(a)$ if $f^*(a) - c(a) \geq 0$, for all a; that is,

$$f^*(a) - \langle a, x\rangle + \gamma \geq 0,$$

or

$$\gamma \geq \langle a, x\rangle - f^*(a). \tag{25.3}$$

This leads us to the following definition, involving the maximum of the right side of the inequality in (25.3), for each fixed x.

Definition 25.2 *The conjugate function associated with f^* is the function*

$$f^{**}(x) = \sup_a(\langle a, x\rangle - f^*(a)). \tag{25.4}$$

For each fixed x, the value $f^{**}(x)$ is the smallest value of γ for which the affine function $c(a) = \langle a, x\rangle - \gamma$ is beneath $f^*(a)$.

Applying the Separation Theorem to the epigraph of the closed, proper, convex function $f(x)$, it can be shown ([133], Theorem 12.1) that $f(x)$ is the point-wise supremum of all the affine functions beneath $f(x)$; that is,

$$f(x) = \sup_{a,\gamma}\{h(x) | f(x) \geq h(x)\}.$$

Therefore,

$$f(x) = \sup_a \Big(\langle a, x\rangle - f^*(a)\Big).$$

This says that

$$f^{**}(x) = f(x). \tag{25.5}$$

25.1.3 Some Examples of Conjugate Functions

- The exponential function $f(x) = \exp(x) = e^x$ has conjugate

$$\exp^*(a) = \begin{cases} a\log a - a, \text{ if } a > 0; \\ 0, \text{ if } a = 0; \\ +\infty, \text{ if } a < 0. \end{cases} \tag{25.6}$$

- The function $f(x) = -\log x$, for $x > 0$, has the conjugate function $f^*(a) = -1 - \log(-a)$, for $a < 0$.
- The function $f(x) = \frac{|x|^p}{p}$ has conjugate $f^*(a) = \frac{|a|^q}{q}$, where $p > 0$, $q > 0$, and $\frac{1}{p} + \frac{1}{q} = 1$.
- Let $\psi_C(x)$ be the *indicator function* of the closed convex set C, that is,
$$\psi_C(x) = \begin{cases} 0, \text{ if } x \in C; \\ +\infty, \text{ if } x \notin C. \end{cases}$$
Then
$$\psi_C^*(a) = \sup_{x \in C} \langle a, x \rangle,$$
which is the *support function* of the set C, usually denoted $\sigma_C(a)$.
- Let $C = \{x| \ ||x||_2 \leq 1\}$, so that
$$\phi(a) = ||a||_2 = \sup_{x \in C} \langle a, x \rangle.$$
Then
$$\phi(a) = \sigma_C(a) = \psi_C^*(a).$$
Therefore,
$$\phi^*(x) = \sigma_C^*(x) = \psi_C^{**}(x) = \psi_C(x) = \begin{cases} 0, \text{ if } x \in C; \\ +\infty, \text{ if } x \notin C. \end{cases}$$

25.1.4 Conjugates and Sub-gradients

We know from the definition of $f^*(a)$ that
$$f^*(a) \geq \langle a, z \rangle - f(z),$$
for all z, and, moreover, $f^*(a)$ is the supremum of these values, taken over all z. If a is a member of the sub-differential $\partial f(x)$, then, for all z, we have
$$f(z) \geq f(x) + \langle a, z - x \rangle,$$
so that
$$\langle a, x \rangle - f(x) \geq \langle a, z \rangle - f(z).$$
It follows that
$$f^*(a) = \langle a, x \rangle - f(x),$$
so that
$$f(x) + f^*(a) = \langle a, x \rangle.$$

If $f(x)$ is a differentiable convex function, then a is in the sub-differential $\partial f(x)$ if and only if $a = \nabla f(x)$. Then we can say

$$f(x) + f^*(\nabla f(x)) = \langle \nabla f(x), x \rangle. \tag{25.7}$$

The conjugate of a differentiable function $f : C \subseteq R^J \rightarrow R$ can then be defined as follows [133]. Let D be the image of the set C under the mapping ∇f. Then, for all $a \in D$ define

$$f^*(a) = \langle a, (\nabla f)^{-1}(a) \rangle - f((\nabla f)^{-1}(a)).$$

25.1.5 The Conjugate of a Concave Function

A function $g : C \subseteq R^J \rightarrow R$ is *concave* if $f(x) = -g(x)$ is convex. One might think that the conjugate of a concave function g is simply the negative of the conjugate of $-g$, but not quite.

The affine function $h(x) = \langle a, x \rangle - \gamma$ is above the concave function $g(x)$ if $h(x) - g(x) \geq 0$, for all x; that is,

$$\langle a, x \rangle - \gamma - g(x) \geq 0,$$

or

$$\gamma \leq \langle a, x \rangle - g(x). \tag{25.8}$$

The conjugate function associated with g is the function

$$g^*(a) = \inf_x (\langle a, x \rangle - g(x)). \tag{25.9}$$

For each fixed a, the value $g^*(a)$ is the largest value of γ for which the affine function $h(x) = \langle a, x \rangle - \gamma$ is above $g(x)$.

It follows, using $f(x) = -g(x)$, that

$$g^*(a) = \inf_x (\langle a, x \rangle + f(x)) = -\sup_x (\langle -a, x \rangle - f(x)) = -f^*(-a).$$

25.2 Fenchel's Duality Theorem

Let $f(x)$ be a proper convex function on $C \subseteq R^J$ and $g(x)$ a proper concave function on $D \subseteq R^J$, where C and D are closed convex sets with non-empty intersection. Fenchel's Duality Theorem deals with the problem of minimizing the difference $f(x) - g(x)$ over $x \in C \cap D$.

We know from our discussion of conjugate functions and differentiability that

$$-f^*(a) \leq f(x) - \langle a, x \rangle,$$

and

$$g^*(a) \leq \langle a, x \rangle - g(x).$$

Therefore,

$$f(x) - g(x) \geq g^*(a) - f^*(a),$$

for all x and a, and so

$$\inf_x \Big(f(x) - g(x)\Big) \geq \sup_a \Big(g^*(a) - f^*(a)\Big).$$

We let C^* be the set of all a such that $f^*(a)$ is finite, with D^* similarly defined.

The Fenchel Duality Theorem, in its general form, as found in [115] and [133], is as follows.

Theorem 25.1 *Assume that $C \cap D$ has points in the relative interior of both C and D, and that either the epigraph of f or that of g has non-empty interior. Suppose that*

$$\mu = \inf_{x \in C \cap D} \Big(f(x) - g(x)\Big)$$

is finite. Then

$$\mu = \inf_{x \in C \cap D} \Big(f(x) - g(x)\Big) = \max_{a \in C^* \cap D^*} \Big(g^*(a) - f^*(a)\Big),$$

where the maximum on the right is achieved at some $a_0 \in C^* \cap D^*$.

If the infimum on the left is achieved at some $x_0 \in C \cap D$, then

$$\max_{x \in C} \Big(\langle x, a_0\rangle - f(x)\Big) = \langle x_0, a_0\rangle - f(x_0),$$

and

$$\min_{x \in D} \Big(\langle x, a_0\rangle - g(x)\Big) = \langle x_0, a_0\rangle - g(x_0).$$

The conditions on the interiors are needed to make use of sub-differentials. For simplicity, we shall limit our discussion to the case of differentiable $f(x)$ and $g(x)$.

25.2.1 Fenchel's Duality Theorem: Differentiable Case

We suppose now that there is $x_0 \in C \cap D$ such that

$$\inf_{x \in C \cap D}(f(x) - g(x)) = f(x_0) - g(x_0),$$

and that

$$\nabla(f - g)(x_0) = 0,$$

or

$$\nabla f(x_0) = \nabla g(x_0). \tag{25.10}$$

Let $\nabla f(x_0) = a_0$. From the equation

$$f(x) + f^*(\nabla f(x)) = \langle \nabla f(x), x \rangle$$

and Equation (25.10),we have

$$f(x_0) - g(x_0) = g^*(a_0) - f^*(a_0),$$

from which it follows that

$$\inf_{x \in C \cap D} (f(x) - g(x)) = \sup_{a \in C^* \cap D^*} (g^*(a) - f^*(a)).$$

This is Fenchel's Duality Theorem.

25.2.2 Optimization over Convex Subsets

Suppose now that $f(x)$ is convex and differentiable on R^J, but we are only interested in its values on the non-empty closed convex set C. Then we redefine $f(x) = +\infty$ for x not in C. The affine function $h(x) = \langle a, x \rangle - \gamma$ is beneath $f(x)$ for all x if and only if it is beneath $f(x)$ for $x \in C$. This motivates our defining the conjugate function now as

$$f^*(a) = \sup_{x \in C} \langle a, x \rangle - f(x).$$

Similarly, let $g(x)$ be concave on D and $g(x) = -\infty$ for x not in D. Then we define

$$g^*(a) = \inf_{x \in D} \langle a, x \rangle - g(x).$$

Let

$$C^* = \{a |\, f^*(a) < +\infty\},$$

and define D^* similarly. We can use Fenchel's Duality Theorem to minimize the difference $f(x) - g(x)$ over the intersection $C \cap D$.

To illustrate the use of Fenchel's Duality Theorem, consider the problem of minimizing the convex function $f(x)$ over the convex set D. Let $C = R^J$ and $g(x) = 0$, for all x. Then

$$f^*(a) = \sup_{x \in C} \langle a, x \rangle - f(x) = \sup_{x} \langle a, x \rangle - f(x),$$

and

$$g^*(a) = \inf_{x \in D} \langle a, x \rangle - g(x) = \inf_{x \in D} \langle a, x \rangle.$$

The supremum is unconstrained and the infimum is with respect to a linear functional. Then, by Fenchel's Duality Theorem, we have

$$\max_a (g^*(a) - f^*(a)) = \inf_{x \in D} f(x).$$

25.3 An Application to Game Theory

In this section we complement our earlier discussion of matrix games by illustrating the application of the Fenchel Duality Theorem to prove the Min-Max Theorem for two-person games.

25.3.1 Pure and Randomized Strategies

In a two-person game, the first player selects a row of the matrix A, say i, and the second player selects a column of A, say j. The second player pays the first player A_{ij}. If some $A_{ij} < 0$, then this means that the first player pays the second. As we discussed previously, there need not be optimal pure strategies for the two players and it may be sensible for them, over the long run, to select their strategies according to some random mechanism. The issues then are which vectors of probabilities will prove optimal and do such optimal probability vectors always exist. The Min-Max Theorem, also known as the Fundamental Theorem of Game Theory, asserts that such optimal probability vectors always exist.

25.3.2 The Min-Max Theorem

In [115], Luenberger uses the Fenchel Duality Theorem to prove the Min-Max Theorem for two-person games. His formulation is in Banach spaces, while we shall limit our discussion to finite-dimensional spaces.

Let A be an I by J pay-off matrix, whose entries represent the payoffs from the second player to the first. Let

$$P = \{p = (p_1, ..., p_I) \,|\, p_i \geq 0, \sum_{i=1}^{I} p_i = 1\},$$

$$Q = \{q = (q_1, ..., q_J) \,|\, q_j \geq 0, \sum_{j=1}^{J} q_j = 1\},$$

and

$$R = A(Q) = \{Aq \,|\, q \in Q\}.$$

The first player selects a vector p in P and the second selects a vector q in Q. The expected pay-off to the first player is

$$E = \langle p, Aq \rangle.$$

Let

$$m_0 = \max_{r \in R} \min_{p \in P} \langle p, r \rangle,$$

and

$$m^0 = \min_{p \in P} \max_{r \in R} \langle p, r \rangle.$$

Clearly, we have

$$\min_{p\in P}\langle p,r\rangle \le \langle p,r\rangle \le \max_{r\in R}\langle p,r\rangle,$$

for all $p \in P$ and $r \in R$. It follows that $m_0 \le m^0$. We show that $m_0 = m^0$.

Define

$$f(x) = \max_{r\in R}\langle x,r\rangle;$$

then f is convex and continuous on R^I. We want $\min_{p\in P} f(p)$.

We apply Fenchel's Duality Theorem, with $f = f$, $g = 0$, $D = P$, and $C = R^I$. Now we have

$$\min_{x\in C\cap D}(f(x) - g(x)) = \min_{p\in P} f(p).$$

We claim that the following are true:

- **1)** $D^* = R^I$;
- **2)** $g^*(a) = \min_{p\in P}\langle p,a\rangle$;
- **3)** $C^* = R$;
- **4)** $f^*(a) = 0$, for all a in R^I.

The first two claims are immediate. To prove the third one, we take a vector $a \in R^I$ that is not in R. Then, by the separation theorem, we can find $x \in R^I$ and $\alpha > 0$ such that

$$\langle x,a\rangle > \alpha + \langle x,r\rangle,$$

for all $r \in R$. Then

$$\langle x,a\rangle - \max_{r\in R}\langle x,r\rangle \ge \alpha > 0.$$

Now take $k > 0$ large and $y = kx$. Since

$$\langle y,r\rangle = k\langle x,r\rangle,$$

we know that

$$\langle y,a\rangle - \max_{r\in R}\langle y,r\rangle = \langle y,a\rangle - f(y) > 0$$

and can be made arbitrarily large by taking $k > 0$ large. It follows that $f^*(a)$ is not finite if a is not in R, so that $C^* = R$.

As for the fourth claim, if $a \in R$, then

$$\langle y,a\rangle - \max_{r\in R}\langle y,r\rangle$$

achieves its maximum value of zero at $y = 0$, so $f^*(a) = 0$.

Finally, we have

$$\min_{p\in P} f(p) = \max_{r\in R} g^*(r) = \max_{r\in R}\min_{p\in P}\langle p, r\rangle.$$

Therefore,

$$\min_{p\in P}\max_{r\in R}\langle p, r\rangle = \max_{r\in R}\min_{p\in P}\langle p, r\rangle.$$

Chapter 26

Appendix: Proximal Minimization

In our discussion of barrier-function methods we considered the PMA algorithm that has, for its iterative step, the minimization of the function

$$f(x) + D_h(x, x^k), \tag{26.1}$$

where $D_h(x, z)$ denotes a Bregman distance from x to z. In this chapter we survey related results concerning proximal minimization.

26.1 Moreau's Proximity Operators

Let $f : R^J \to (-\infty, +\infty]$ be a closed, proper, convex function. When f is differentiable, we can find minimizers of f using techniques such as gradient descent. When f is not necessarily differentiable, the minimization problem is more difficult. One approach is to augment the function f and to convert the problem into one of minimizing a differentiable function. Moreau's approach is one example of this.

26.1.1 The Moreau Envelope

The Moreau envelope of the function f is the function

$$m_f(z) = \inf_x \left\{ f(x) + \frac{1}{2}||x - z||_2^2 \right\}, \tag{26.2}$$

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 [133]). In similar fashion, we can define $m_{f^*} z$ and $\text{prox}_{f^*} z$.

Proposition 26.1 *The infimum of $m_f(z)$, over all z, is the same as the infimum of $f(x)$, over all x.*

Proof: We have

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

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

∎

Later, we shall show that the minimizers of $m_f(z)$ and $f(x)$ are the same, as well.

Both m_f and m_{f^*} are convex and differentiable. The point $x = \text{prox}_f z$ is characterized by the property $z - x \in \partial f(x)$. Consequently, x is a global minimizer of f if and only if $x = \text{prox}_f x$.

For example, consider the indicator function of the convex set C, $f(x) = \psi_C(x)$ that is zero if x is in the closed convex set C and $+\infty$ otherwise. Then $m_f z$ is the minimum of $\frac{1}{2}||x-z||_2^2$ over all x in C, and $\text{prox}_f z = P_C z$, the orthogonal projection of z onto the set C.

If $f : R \to R$ is $f(t) = \omega|t|$, then

$$\text{prox}_f(t) = t - \frac{t}{|t|}\omega, \tag{26.3}$$

for $|t| \leq \omega$, and equals zero, otherwise.

The operators $\text{prox}_f : z \to \text{prox}_f z$ are *proximal operators*. These operators generalize the projections onto convex sets, and, like those operators, are firmly non-expansive [66].

The support function of the convex set C is $\sigma_C(x) = \sup_{u \in C} \langle x, u \rangle$. It is easy to see that $\sigma_C = \psi_C^*$. For $f^*(z) = \sigma_C(z)$, we can find $m_{f^*} z$ using Moreau's Theorem ([133], p.338).

26.1.2 Moreau's Theorem and Applications

Moreau's Theorem generalizes the decomposition of members of R^J with respect to a subspace. For a proof, see the book by Rockafellar [133].

Theorem 26.1 (Moreau's Theorem) *Let f be a closed, proper, convex function. Then*

$$m_f z + m_{f^*} z = \frac{1}{2}||z||^2; \tag{26.4}$$

and

$$\text{prox}_f z + \text{prox}_{f^*} z = z. \tag{26.5}$$

In addition, we have

$$\text{prox}_{f^*} z \in \partial f(\text{prox}_f z),$$

$$\text{prox}_{f^*} z = \nabla m_f(z), \text{ and}$$

$$\text{prox}_f z = \nabla m_{f^*}(z). \tag{26.6}$$

Since $\sigma_C = \psi_C^*$, we have

$$\text{prox}_{\sigma_C} z = z - \text{prox}_{\psi_C} z = z - P_C z. \tag{26.7}$$

The following proposition illustrates the usefulness of these concepts.

Proposition 26.2 *The minimizers of m_f and the minimizers of f are the same.*

Proof: From Moreau's Theorem we know that

$$\nabla m_f(z) = \text{prox}_{f^*} z = z - \text{prox}_f z, \tag{26.8}$$

so $\nabla m_f z = 0$ is equivalent to $z = \text{prox}_f z$. ∎

26.1.3 Iterative Minimization of $m_f z$

Because the minimizers of m_f are also minimizers of f, we can find global minimizers of f using standard iterative methods, such as gradient descent, on m_f. The gradient descent iterative step has the form

$$x^{k+1} = x^k - \gamma_k \nabla m_f(x^k). \tag{26.9}$$

We know from Moreau's Theorem that

$$\nabla m_f z = \text{prox}_{f^*} z = z - \text{prox}_f z, \tag{26.10}$$

so that Equation (26.9) can be written as

$$x^{k+1} = x^k - \gamma_k (x^k - \text{prox}_f x^k)$$

$$= (1 - \gamma_k) x^k + \gamma_k \text{prox}_f x^k. \tag{26.11}$$

Because

$$x^k - \text{prox}_f x^k \in \partial f(\text{prox}_f x^k), \tag{26.12}$$

the iteration in Equation (26.11) has the increment

$$x^{k+1} - x^k \in -\gamma_k \partial f(x^{k+1}), \tag{26.13}$$

in contrast to what we would have with the usual gradient descent method for differentiable f:

$$x^{k+1} - x^k = -\gamma_k \nabla f(x^k). \tag{26.14}$$

It follows from the definition of $\partial f(x^{k+1})$ that $f(x^k) \geq f(x^{k+1})$ for the iteration in Equation (26.11).

26.1.4 Forward-Backward Splitting

In [66] the authors consider the problem of minimizing the function $f = f_1 + f_2$, where f_2 is differentiable and its gradient is λ-Lipschitz continuous. The function f is minimized at the point x if and only if

$$0 \in \partial f(x) = \partial f_1(x) + \nabla f_2(x), \tag{26.15}$$

so we have

$$-\gamma \nabla f_2(x) \in \gamma \partial f_1(x), \tag{26.16}$$

for any $\gamma > 0$. Therefore

$$x - \gamma \nabla f_2(x) - x \in \gamma \partial f_1(x). \tag{26.17}$$

From Equation (26.17) we conclude that

$$x = \text{prox}_{\gamma f_1}(x - \gamma \nabla f_2(x)). \tag{26.18}$$

This suggests an algorithm with the iterative step

$$x^{k+1} = \text{prox}_{\gamma f_1}(x^k - \gamma \nabla f_2(x^k)). \tag{26.19}$$

In order to guarantee convergence, γ is chosen to lie in the interval $(0, 2/\lambda)$. It is also possible to allow γ to vary with the k. This is called the *forward-backward splitting* approach. As noted in [66], the forward-backward splitting approach has, as a particular case, the CQ algorithm of [41, 42].

26.1.5 Generalizing the Moreau Envelope

The Moreau envelope involves the infimum of the function

$$f(x) + \frac{1}{2}||x - z||_2^2. \tag{26.20}$$

Consequently, the Moreau envelope can be generalized in various ways, either by changing the $\frac{1}{2}$ to a variable parameter, or replacing the Euclidean distance by a more general *distance measure*.

For real $\lambda > 0$, the Moreau-Yosida approximation of index λ ([3]) is the function

$$F_\lambda(z) = \inf_x \left\{ f(x) + \frac{1}{2\lambda}||x - z||_2^2 \right\}. \tag{26.21}$$

For fixed λ, the theory is much the same as for the Moreau envelope [3, 4]. For fixed λ, $F_\lambda(z)$ can be viewed as an approximate minimization of $f(x)$, involving regularization based on an additive penalty term. If $z = 0$, then $F_\lambda(0)$ is a norm-constrained minimization of $f(x)$.

26.2 Proximity Operators using Bregman Distances

Several authors have extended Moreau's results by replacing the Euclidean squared distance with a Bregman distance. Let h be a closed proper convex function that is differentiable on the nonempty set $\text{int}D$. The corresponding *Bregman distance* $D_h(x,z)$ is defined for $x \in R^J$ and $z \in \text{int}D$ by

$$D_h(x,z) = h(x) - h(z) - \langle \nabla h(z), x - z \rangle. \tag{26.22}$$

Note that $D_h(x,z) \geq 0$ always and that $D_h(x,z) = +\infty$ is possible. If h is essentially strictly convex then $D_h(x,z) = 0$ implies that $x = z$.

26.2.1 Teboulle's Entropic Proximal Mappings

Teboulle [144] considers the function

$$R(x,z) = f(x) + \epsilon D_h(x,z), \tag{26.23}$$

and shows that, with certain restrictions on f and h, the function $R(\cdot,z)$ attains its minimum value, $R_\epsilon(z)$, at a unique $x = E_h(f,z)$. He then generalizes Moreau's Theorem, proving that the operator $E_h(f,\cdot)$ has properties analogous to the proximity operators $\text{prox}_f(\cdot)$. He then demonstrates that several nonlinear programming problems can be formulated using such functions $R(x,z)$. He is primarily concerned with the behavior of $R_\epsilon(z)$, as z varies, and not as ϵ varies.

Teboulle's method relies on Fenchel's Duality Theorem [133], and therefore requires the conjugate of the function $g(x) = D_h(x,z)$. As he shows,

$$g^*(y) = h^*(y + \nabla h(z)) - h^*(\nabla h(z)). \tag{26.24}$$

His main result requires the joint convexity of the function $D_h(x,z)$.

26.2.2 Proximal Minimization of Censor and Zenios

Censor and Zenios [63] also consider $R(x,z)$. They are less interested in the properties of the operator $E_h(f,\cdot)$ and more interested in the behavior of their PMD iterative algorithm defined by

$$x^{k+1} = \text{argmin}\left(f(x) + D_h(x,x^k)\right). \tag{26.25}$$

In their work, the function h is a Bregman function with zone S. They show that, subject to certain assumptions, if the function f has a minimizer within the closure of S, then the PMD iterates converge to such a minimizer. It is true that their method and results are somewhat more

general, in that they consider also the minimizers of $R(x, z)$ over another closed convex set X; however, this set X is unrelated to the function h.

The PMA algorithm presented in a previous chapter has the same iterative step as the PMD method of Censor and Zenios. However, the assumptions about f and h are different, and our theorem asserts convergence of the iterates to a constrained minimizer of f over $\overline{D}$. In other words, we solve a constrained minimization problem, whereas Censor and Zenios solve the unconstrained minimization problem, under a restrictive assumption on the location of minimizers of f.

26.3 Exercises

26.1 *Since $f^*(a) \geq \langle a, x\rangle - f(x)$ for all a and x, the function of two vector variables given by*

$$W_f(x, a) = f(x) - \langle a, x\rangle + f^*(a)$$

is nonnegative, for all x and a, and so it defines a distance. Show that

$$W_f(x, \nabla f(y)) = D_f(x, y),$$

for all suitable x and y.

Chapter 27

Appendix: Bregman-Legendre Functions

In [11] 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 [11].

27.1 Essential Smoothness and Essential Strict Convexity

Following [133] 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 $||\nabla f(x^n)|| \to +\infty$. Here int$D$ and bd$D$ denote the interior and boundary of the set D. A closed proper convex function f is *essentially strictly convex* if f is strictly convex on every convex subset of dom ∂f.

The closed proper convex function f is essentially smooth if and only if the subdifferential $\partial f(x)$ is empty for $x \in \text{bd}D$ 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 27.1 *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 $f^* = R^J$ then the range of ∇f is R^J and the equation $\nabla f(x) = y$ can be solved for every $y \in R^J$. In order for int dom $f^* = R^J$ it is necessary and sufficient that the Legendre function f be *super-coercive*, that is,

$$\lim_{||x|| \to +\infty} \frac{f(x)}{||x||} = +\infty. \tag{27.1}$$

If the effective domain of f is bounded, then f is super-coercive and its gradient operator is a mapping onto the space R^J.

27.2 Bregman Projections onto Closed Convex Sets

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

$$D_f(x, z) = f(x) - f(z) - \langle \nabla f(z), x - z \rangle. \tag{27.2}$$

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

$$P_K^f(z) = \text{argmin}_{x \in K \cap D} D_f(x, z). \tag{27.3}$$

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

Example: Let $J = 2$ and $f(x)$ be the function that is equal to one-half the norm squared on D, the nonnegative quadrant, $+\infty$ elsewhere. Let K be the set $K = \{(x_1, x_2) | x_1 + x_2 = 1\}$. The Bregman projection of $(2, 1)$ onto K is $(1, 0)$, which is not in int$D$. The function f is not essentially smooth, although it is essentially strictly convex. Its conjugate is the function f^* that is equal to one-half the norm squared on D and equal to zero elsewhere; it is essentially smooth, but not essentially strictly convex.

If f is Legendre, then $P_K^f(z)$ is the unique member of $K \cap$ int$D$ satisfying the inequality

$$\langle \nabla f(P_K^f(z)) - \nabla f(z), P_K^f(z) - c \rangle \geq 0, \tag{27.4}$$

for all $c \in K$. From this we obtain the *Bregman Inequality*:

$$D_f(c, z) \geq D_f(c, P_K^f(z)) + D_f(P_K^f(z), z), \tag{27.5}$$

for all $c \in K$.

27.3 Bregman-Legendre Functions

Following Bauschke and Borwein [11], 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 [23].

27.4 Useful Results about Bregman-Legendre Functions

The following results are proved in somewhat more generality in [11].
R1: If $y^n \in$ int dom f and $y^n \to y \in$ int dom f, then $D_f(y,y^n) \to 0$.
R2: If x and $y^n \in$ int dom f and $y^n \to y \in$ bd dom f, then $D_f(x,y^n) \to +\infty$.
R3: If $x^n \in D$, $x^n \to x \in D$, $y^n \in$ int D, $y^n \to y \in D$, $\{x,y\} \cap$ int $D \neq \emptyset$ and $D_f(x^n,y^n) \to 0$, then $x = y$ and $y \in$ int D.
R4: If x and y are in D, but are not in int D, $y^n \in$ int D, $y^n \to y$ and $D_f(x,y^n) \to 0$, then $x = y$.
As a consequence of these results we have the following.
R5: If $\{D_f(x,y^n)\} \to 0$, for $y^n \in$ int D and $x \in R^J$, then $\{y^n\} \to x$.

Proof of R5: Since $\{D_f(x,y^n)\}$ is eventually finite, we have $x \in D$. By Property B1 above it follows that the sequence $\{y^n\}$ is bounded; without loss of generality, we assume that $\{y^n\} \to y$, for some $y \in \overline{D}$. If x is in int D, then, by result R2 above, we know that y is also in int D. Applying result R3, with $x^n = x$, for all n, we conclude that $x = y$. If, on the other hand, x is in D, but not in int D, then y is in D, by result R2. There are two cases to consider: 1) y is in int D; 2) y is not in int D. In case 1) we have $D_f(x,y^n) \to D_f(x,y) = 0$, from which it follows that $x = y$. In case 2) we apply result R4 to conclude that $x = y$. ∎

Chapter 28

Appendix: Likelihood Maximization

A fundamental problem in statistics is the estimation of underlying population parameters from measured data. For example, political pollsters want to estimate the percentage of voters who favor a particular candidate. They can't ask everyone, so they sample the population and estimate the percentage from the answers they receive from a relative few. Bottlers of soft drinks want to know if their process of sealing the bottles is effective. Obviously, they can't open every bottle to check the process. They open a few bottles, selected randomly according to some testing scheme, and make their assessment of the effectiveness of the overall process after opening a few bottles. As we shall see, optimization plays an important role in the estimation of parameters from data.

28.1 Maximizing the Likelihood Function

Suppose that $\mathbf{Y}$ is a random vector whose probability density function (pdf) $f(\mathbf{y};\mathbf{x})$ is a function of the vector variable $\mathbf{y}$ and is a member of a family of pdf parametrized by the vector variable $\mathbf{x}$. Our data is one instance of $\mathbf{Y}$; that is, one particular value of the variable $\mathbf{y}$, which we also denote by $\mathbf{y}$. We want to estimate the correct value of the variable $\mathbf{x}$, which we shall also denote by $\mathbf{x}$. This notation is standard and the dual use of the symbols $\mathbf{y}$ and $\mathbf{x}$ should not cause confusion. Given the particular $\mathbf{y}$ we can estimate the correct $\mathbf{x}$ by viewing $f(\mathbf{y};\mathbf{x})$ as a function of the second variable, with the first variable held fixed. This function of the parameters only is called the *likelihood function*. A *maximum likelihood* (ML) estimate of the parameter vector $\mathbf{x}$ is any value of the second variable for which the function is maximized. We consider several examples.

28.1.1 Example 1: Estimating a Gaussian Mean

Let $Y_1, ..., Y_I$ be I independent Gaussian (or normal) random variables with known variance $\sigma^2 = 1$ and unknown common mean μ. Let $\mathbf{Y} = (Y_1, ..., Y_I)^T$. The parameter x we wish to estimate is the mean $x = \mu$. Then, the random vector $\mathbf{Y}$ has the pdf

$$f(\mathbf{y}; x) = (2\pi)^{-I/2} \exp(-\frac{1}{2} \sum_{i=1}^{I} (y_i - x)^2).$$

Holding $\mathbf{y}$ fixed and maximizing over x is equivalent to minimizing

$$\sum_{i=1}^{I} (y_i - x)^2$$

as a function of x. The ML estimate is the arithmetic mean of the data,

$$x_{ML} = \frac{1}{I} \sum_{i=1}^{I} y_i.$$

Notice that $E(\mathbf{Y})$, the expected value of $\mathbf{Y}$, is the vector $\mathbf{x}$ all of whose entries are $x = \mu$. The ML estimate is the least squares solution of the overdetermined system of equations $\mathbf{y} = E(\mathbf{Y})$; that is,

$$y_i = x$$

for $i = 1, ..., I$.

The least-squares solution of a system of equations $A\mathbf{x} = \mathbf{b}$ is the vector that minimizes the Euclidean distance between $A\mathbf{x}$ and $\mathbf{b}$; that is, it minimizes the Euclidean norm of their difference, $||A\mathbf{x} - \mathbf{b}||$, where, for any two vectors $\mathbf{a}$ and $\mathbf{b}$ we define

$$||\mathbf{a} - \mathbf{b}||^2 = \sum_{i=1}^{I} (a_i - b_i)^2.$$

As we shall see in the next example, another important measure of distance is the *Kullback-Leibler* (KL) distance between two nonnegative vectors $\mathbf{c}$ and $\mathbf{d}$, given by

$$KL(\mathbf{c}, \mathbf{d}) = \sum_{i=1}^{I} c_i \log(c_i/d_i) + d_i - c_i.$$

28.1.2 Example 2: Estimating a Poisson Mean

Let $Y_1, ..., Y_I$ be I independent Poisson random variables with unknown common mean λ, which is the parameter x we wish to estimate. Let $\mathbf{Y} = (Y_1, ..., Y_I)^T$. Then, the probability function of $\mathbf{Y}$ is

$$f(\mathbf{y}; x) = \prod_{i=1}^{I} \exp(-x) x^{y_i} / (y_i)!.$$

Holding $\mathbf{y}$ fixed and maximizing this likelihood function over positive values of x is equivalent to minimizing the Kullback-Leibler distance between the nonnegative vector $\mathbf{y}$ and the vector $\mathbf{x}$ whose entries are all equal to x, given by

$$KL(\mathbf{y}, \mathbf{x}) = \sum_{i=1}^{I} y_i \log(y_i / x) + x - y_i.$$

The ML estimator is easily seen to be the arithmetic mean of the data,

$$x_{ML} = \frac{1}{I} \sum_{i=1}^{I} y_i.$$

The vector $\mathbf{x}$ is again $E(\mathbf{Y})$, so the ML estimate is once again obtained by finding an approximate solution of the overdetermined system of equations $\mathbf{y} = E(\mathbf{Y})$. In the previous example the approximation was in the least squares sense, whereas here it is in the minimum KL sense; the ML estimate is the arithmetic mean in both cases because the parameter to be estimated is one-dimensional.

28.1.3 Example 3: Estimating a Uniform Mean

Suppose now that $Y_1, ..., Y_I$ are independent random variables uniformly distributed over the interval $[0, 2x]$. The parameter to be determined is their common mean, x. The random vector $\mathbf{Y} = (Y_1, ..., Y_I)^T$ has the pdf

$$f(\mathbf{y}; x) = x^{-I}, \text{for } 2x \geq m,$$

$$f(\mathbf{y}; x) = 0\,, \text{otherwise}\,,$$

where m is the maximum of the y_i. For fixed vector $\mathbf{y}$ the ML estimate of x is $m/2$. The expected value of $\mathbf{Y}$ is $E(\mathbf{Y}) = \mathbf{x}$ whose entries are all equal to x. In this case the ML estimator is not obtained by finding an approximate solution to the overdetermined system $\mathbf{y} = E(\mathbf{Y})$.

Since we can always write

$$\mathbf{y} = E(\mathbf{Y}) + (\mathbf{y} - E(\mathbf{Y})),$$

we can model $\mathbf{y}$ as the sum of $E(\mathbf{Y})$ and mean-zero error or noise. Since $f(\mathbf{y};\mathbf{x})$ depends on $\mathbf{x}$, so does $E(\mathbf{Y})$. Therefore, it makes some sense to consider estimating our parameter vector $\mathbf{x}$ using an approximate solution for the system of equations

$$\mathbf{y} = E(\mathbf{Y}).$$

As the first two examples (as well as many others) illustrate, this is what the ML approach often amounts to, while the third example shows that this is not always the case, however. Still to be determined, though, is the metric with respect to which the approximation is to be performed. As the Gaussian and Poisson examples showed, the ML formalism can provide that metric. In those overly simple cases it did not seem to matter which metric we used, but it does matter.

28.1.4 Example 4: Image Restoration

A standard model for image restoration is the following:

$$\mathbf{y} = A\mathbf{x} + \mathbf{z},$$

where $\mathbf{y}$ is the blurred image, A is an I by J matrix describing the linear imaging system, $\mathbf{x}$ is the desired vectorized restored image, and $\mathbf{z}$ is (possibly correlated) mean-zero additive Gaussian noise. The noise covariance matrix is $Q = E(\mathbf{z}\mathbf{z}^T)$. Then $E(\mathbf{Y}) = A\mathbf{x}$, and the pdf is

$$f(\mathbf{y};\mathbf{x}) = c\exp(-(\mathbf{y}-A\mathbf{x})^T Q^{-1}(\mathbf{y}-A\mathbf{x})),$$

where c is a constant that does not involve $\mathbf{x}$. Holding $\mathbf{y}$ fixed and maximizing $f(\mathbf{y};\mathbf{x})$ with respect to $\mathbf{x}$ is equivalent to minimizing

$$(\mathbf{y}-A\mathbf{x})^T Q^{-1}(\mathbf{y}-A\mathbf{x}).$$

Therefore, the ML solution is obtained by finding a weighted least squares approximate solution of the over-determined linear system $\mathbf{y} = E(\mathbf{Y})$, with the weights coming from the matrix Q^{-1}. When the noise terms are uncorrelated and have the same variance, this reduces to the least squares solution.

28.1.5 Example 5: Poisson Sums

The model of sums of independent Poisson random variables is commonly used in emission tomography and elsewhere. Let P be an I by J matrix with nonnegative entries, and let $\mathbf{x} = (x_1, ..., x_J)^T$ be a vector of nonnegative parameters. Let $Y_1, ..., Y_I$ be independent Poisson random variables

with positive means

$$E(Y_i) = \sum_{j=1}^{J} P_{ij} x_j = (P\mathbf{x})_i.$$

The probability function for the random vector $\mathbf{Y}$ is then

$$f(\mathbf{y}; \mathbf{x}) = c \prod_{i=1}^{I} \exp(-(P\mathbf{x})_i)((P\mathbf{x})_i)^{y_i},$$

where c is a constant not involving $\mathbf{x}$. Maximizing this function of $\mathbf{x}$ for fixed $\mathbf{y}$ is equivalent to minimizing the KL distance $KL(\mathbf{y}, P\mathbf{x})$ over nonnegative $\mathbf{x}$. The expected value of the random vector $\mathbf{Y}$ is $E(\mathbf{Y}) = P\mathbf{x}$ and once again we see that the ML estimate is a nonnegative approximate solution of the system of (linear) equations $\mathbf{y} = E(\mathbf{Y})$, with the approximation in the KL sense. The system $\mathbf{y} = P\mathbf{x}$ may not be over-determined; there may even be exact solutions. But we require in addition that $\mathbf{x} \geq 0$ and there need not be a nonnegative solution to $\mathbf{y} = P\mathbf{x}$. We see from this example that constrained optimization plays a role in solving our problems.

28.1.6 Discrete Mixtures

We say that a discrete random variable Z taking values in the set $\{i = 1, ..., I\}$ is a *mixture* if there are probability vectors f_j and numbers $x_j > 0$, for $j = 1, ..., J$, such that the probability vector for Z is

$$f(i) = \text{Prob}(Z = i) = \sum_{j=1}^{J} x_j f_j(i).$$

We require, of course, that $\sum_{j=1}^{J} x_j = 1$.

The data are N realizations of the random variable Z, denoted z_n, for $n = 1, ..., N$. The column vector $x = (x_1, ..., x_J)^T$ is the parameter vector of mixture probabilities to be estimated. The likelihood function is

$$L(x) = \prod_{n=1}^{N} \Big(x_1 f_1(z_n) + ... + x_J f_J(z_n) \Big),$$

which can be written as

$$L(x) = \prod_{i=1}^{I} \Big(x_1 f_1(i) + ... + x_J f_J(i) \Big)^{n_i},$$

where n_i is the cardinality of the set $\{n | \, i_n = i\}$. Then the log likelihood function is

$$LL(x) = \sum_{i=1}^{I} n_i \log \Big(x_1 f_1(i) + ... + x_J f_J(i) \Big).$$

With y the column vector with entries $y_i = n_i/N$, and P the matrix with entries $P_{ij} = f_j(i)$, we see that

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

so maximizing $LL(x)$ over non-negative vectors x with $\sum_{j=1}^{J}x_j = 1$ is equivalent to minimizing the KL distance $KL(y, Px)$ over the same vectors. The restriction that the entries of x sum to one turns out to be redundant, as we show now.

Applying Theorem 8.5, the gradient form of the Karush-Kuhn-Tucker Theorem, we know that, for any $\hat{x}$ that is a non-negative minimizer of $KL(y, Px)$, we have

$$\sum_{i=1}^{I}P_{ij}\Big(1 - \frac{y_i}{(P\hat{x})_i}\Big) \geq 0,$$

and

$$\sum_{i=1}^{I}P_{ij}\Big(1 - \frac{y_i}{(P\hat{x})_i}\Big) = 0,$$

for all j such that $\hat{x}_j > 0$. Consequently, we can say that

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

for all j. Since, in the mixture problem, we have $s_j = \sum_{i=1}^{I}P_{ij} = 1$ for each j, it follows that

$$\sum_{j=1}^{J}\hat{x}_j = \sum_{i=1}^{I}\Big(\sum_{j=1}^{J}\hat{x}_jP_{ij}\Big)\frac{y_i}{(P\hat{x})_i} = \sum_{i=1}^{I}y_i = 1.$$

So we know now that, for this problem, any non-negative minimizer of $KL(y, Px)$ will be a probability vector that maximizes $LL(x)$. Since the EMML algorithm minimizes $KL(y, Px)$ it can be used to find the maximum-likelihood estimate of the mixture probabilities. It is helpful to remember that there was no mention of Poisson distributions in this example, and that the EMML algorithm can be used to find likelihood maximizers in situations other than that of sums of independent Poisson random variables.

28.2 Alternative Approaches

The ML approach is not always the best approach. As we have seen, the ML estimate is often found by solving, at least approximately, the system of

equations $\mathbf{y} = E(\mathbf{Y})$. Since noise is always present, this system of equations is rarely a correct statement of the situation. It is possible to overfit the mean to the noisy data, in which case the resulting $\mathbf{x}$ can be useless. In such cases Bayesian methods and maximum *a posteriori* estimation, as well as other forms of regularization techniques and penalty function techniques, can help. Other approaches involve stopping iterative algorithms prior to convergence.

In most applications the data is limited and it is helpful to include prior information about the parameter vector $\mathbf{x}$ to be estimated. In the Poisson mixture problem the vector $\mathbf{x}$ must have nonnegative entries. In certain applications, such as transmission tomography, we might have upper bounds on suitable values of the entries of $\mathbf{x}$.

From a mathematical standpoint we are interested in the convergence of iterative algorithms, while in many applications we want usable estimates in a reasonable amount of time, often obtained by running an iterative algorithm for only a few iterations. Algorithms designed to minimize the same cost function can behave quite differently during the early iterations. Iterative algorithms, such as block-iterative or incremental methods, that can provide decent answers quickly will be important.

Chapter 29

Appendix: Reconstruction from Limited Data

The problem is to reconstruct a (possibly complex-valued) function $f : R^D \rightarrow C$ from finitely many measurements g_n, $n = 1, ..., N$, pertaining to f. The function $f(r)$ represents the physical object of interest, such as the spatial distribution of acoustic energy in sonar, the distribution of x-ray-attenuating material in transmission tomography, the distribution of radionuclide in emission tomography, the sources of reflected radio waves in radar, and so on. Often the reconstruction, or estimate, of the function f takes the form of an image in two or three dimensions; for that reason, we also speak of the problem as one of *image reconstruction*. The data are obtained through measurements. Because there are only finitely many measurements, the problem is highly under-determined and even noise-free data are insufficient to specify a unique solution.

29.1 The Optimization Approach

One way to solve such under-determined problems is to replace $f(r)$ with a vector in C^N and to use the data to determine the N entries of this vector. An alternative method is to model $f(r)$ as a member of a family of linear combinations of N preselected basis functions of the multivariable r. Then the data is used to determine the coefficients. This approach offers the user the opportunity to incorporate prior information about $f(r)$ in the choice of the basis functions. Such finite-parameter models for $f(r)$ can be obtained through the use of the minimum-norm estimation procedure,

as we shall see. More generally, we can associate a *cost* with each data-consistent function of r, and then minimize the cost over all the potential solutions to the problem. Using a norm as a cost function is one way to proceed, but there are others. These optimization problems can often be solved only through the use of discretization and iterative algorithms.

29.2 Introduction to Hilbert Space

In many applications the data are related linearly to f. To model the operator that transforms f into the data vector, we need to select an ambient space containing f. Typically, we choose a Hilbert space. The selection of the inner product provides an opportunity to incorporate prior knowledge about f into the reconstruction. The inner product induces a norm and our reconstruction is that function, consistent with the data, for which this norm is minimized. We shall illustrate the method using Fourier-transform data and prior knowledge about the support of f and about its overall shape.

Our problem, then, is to estimate a (possibly complex-valued) function $f(r)$ of D real variables $r = (r_1, ..., r_D)$ from finitely many measurements, g_n, $n = 1, ..., N$. We shall assume, in this chapter, that these measurements take the form

$$g_n = \int_S f(r)\overline{h_n(r)}dr, \tag{29.1}$$

where S denotes the support of the function $f(r)$, which, in most cases, is a bounded set. For the purpose of estimating, or reconstructing, $f(r)$, it is convenient to view Equation (29.1) in the context of a Hilbert space, and to write

$$g_n = \langle f, h_n \rangle, \tag{29.2}$$

where the usual Hilbert space inner product is defined by

$$\langle f, h \rangle_2 = \int_S f(r)\overline{h(r)}dr, \tag{29.3}$$

for functions $f(r)$ and $h(r)$ supported on the set S. Of course, for these integrals to be defined, the functions must satisfy certain additional properties, but a more complete discussion of these issues is outside the scope of this chapter. The Hilbert space so defined, denoted $L^2(S)$, consists (essentially) of all functions $f(r)$ for which the norm

$$||f||_2 = \sqrt{\int_S |f(r)|^2 dr} \tag{29.4}$$

is finite.

29.2.1 Minimum-Norm Solutions

Our estimation problem is highly under-determined; there are infinitely many functions in $L^2(S)$ that are consistent with the data and might be the right answer. Such under-determined problems are often solved by acting conservatively, and selecting as the estimate that function consistent with the data that has the smallest norm. At the same time, however, we often have some prior information about f that we would like to incorporate in the estimate. One way to achieve both of these goals is to select the norm to incorporate prior information about f, and then to take as the estimate of f the function consistent with the data, for which the chosen norm is minimized.

The data vector $g = (g_1, ..., g_N)^T$ is in C^N and the linear operator $\mathcal{H}$ from $L^2(S)$ to C^N takes f to g; so we write $g = \mathcal{H}f$. Associated with the mapping $\mathcal{H}$ is its adjoint operator, $\mathcal{H}^\dagger$, going from C^N to $L^2(S)$ and given, for each vector $a = (a_1, ..., a_N)^T$, by

$$\mathcal{H}^\dagger a(r) = a_1 h_1(r) + ... + a_N h_N(r). \tag{29.5}$$

The operator from C^N to C^N defined by $\mathcal{H}\mathcal{H}^\dagger$ corresponds to an N by N matrix, which we shall also denote by $\mathcal{H}\mathcal{H}^\dagger$. If the functions $h_n(r)$ are linearly independent, then this matrix is positive-definite, therefore invertible.

Given the data vector g, we can solve the system of linear equations

$$g = \mathcal{H}\mathcal{H}^\dagger a \tag{29.6}$$

for the vector a. Then the function

$$\hat{f}(r) = \mathcal{H}^\dagger a(r) \tag{29.7}$$

is consistent with the measured data and is the function in $L^2(S)$ with the smallest norm for which this is true. The function $w(r) = f(r) - \hat{f}(r)$ has the property $\mathcal{H}w = 0$. It is easy to see that

$$||f||_2^2 = ||\hat{f}||_2^2 + ||w||_2^2 \tag{29.8}$$

The estimate $\hat{f}(r)$ is the *minimum-norm solution*, with respect to the norm defined in Equation (29.4). If we change the norm on $L^2(S)$, or, equivalently, the inner product, then the minimum-norm solution will change.

For any continuous linear operator $\mathcal{T}$ on $L^2(S)$, the adjoint operator, denoted $\mathcal{T}^\dagger$, is defined by

$$\langle \mathcal{T}f, h\rangle_2 = \langle f, \mathcal{T}^\dagger h\rangle_2. \tag{29.9}$$

The adjoint operator will change when we change the inner product.

29.3 A Class of Inner Products

Let $\mathcal{T}$ be a continuous, linear, and invertible operator on $L^2(S)$. Define the $\mathcal{T}$ inner product to be

$$\langle f, h\rangle_{\mathcal{T}} = \langle \mathcal{T}^{-1} f, \mathcal{T}^{-1} h\rangle_2. \tag{29.10}$$

We can then use this inner product to define the problem to be solved. We now say that

$$g_n = \langle f, t^n\rangle_{\mathcal{T}}, \tag{29.11}$$

for known functions $t^n(r)$. Using the definition of the $\mathcal{T}$ inner product, we find that

$$g_n = \langle f, h^n\rangle_2 = \langle \mathcal{T} f, \mathcal{T} h^n\rangle_{\mathcal{T}}. \tag{29.12}$$

The adjoint operator for $\mathcal{T}$, with respect to the $\mathcal{T}$-norm, is denoted $\mathcal{T}^*$, and is defined by

$$\langle \mathcal{T} f, h\rangle_{\mathcal{T}} = \langle f, \mathcal{T}^* h\rangle_{\mathcal{T}}. \tag{29.13}$$

Therefore,

$$g_n = \langle f, \mathcal{T}^* \mathcal{T} h^n\rangle_{\mathcal{T}}. \tag{29.14}$$

Lemma 29.1 . *We have* $\mathcal{T}^* \mathcal{T} = \mathcal{T}\mathcal{T}^\dagger$.

Consequently, we have

$$g_n = \langle f, \mathcal{T}\mathcal{T}^\dagger h^n\rangle_{\mathcal{T}}. \tag{29.15}$$

29.4 Minimum-$\mathcal{T}$-Norm Solutions

The function $\tilde{f}$ consistent with the data and having the smallest $\mathcal{T}$-norm has the algebraic form

$$\hat{f} = \sum_{m=1}^{N} a_m \mathcal{T}\mathcal{T}^\dagger h^m. \tag{29.16}$$

Applying the $\mathcal{T}$-inner product to both sides of Equation (29.16), we get

$$g_n = \langle \hat{f}, \mathcal{T}\mathcal{T}^\dagger h^n\rangle_{\mathcal{T}} \tag{29.17}$$

$$= \sum_{m=1}^{N} a_m \langle \mathcal{T}\mathcal{T}^\dagger h^m, \mathcal{T}\mathcal{T}^\dagger h^n\rangle_{\mathcal{T}}. \tag{29.18}$$

Therefore,

$$g_n = \sum_{m=1}^{N} a_m \langle \mathcal{T}^\dagger h^m, \mathcal{T}^\dagger h^n \rangle_2. \tag{29.19}$$

We solve this system for the a_m and insert them into Equation (29.16) to get our reconstruction. The Gram matrix that appears in Equation (29.19) is positive-definite, but is often ill-conditioned; increasing the main diagonal by a percent or so usually is sufficient regularization.

29.5 The Case of Fourier-Transform Data

To illustrate these minimum-$\mathcal{T}$-norm solutions, we consider the case in which the data are values of the Fourier transform of f. Specifically, suppose that

$$g_n = \int_S f(x) e^{-i\omega_n x} dx, \tag{29.20}$$

for arbitrary values ω_n.

29.5.1 The $L^2(-\pi, \pi)$ Case

Assume that $f(x) = 0$, for $|x| > \pi$. The minimum-2-norm solution has the form

$$\hat{f}(x) = \sum_{m=1}^{N} a_m e^{i\omega_m x}, \tag{29.21}$$

with

$$g_n = \sum_{m=1}^{N} a_m \int_{-\pi}^{\pi} e^{i(\omega_m - \omega_n)x} dx. \tag{29.22}$$

For the equi-spaced values $\omega_n = n$ we find that $a_m = g_m$ and the minimum-norm solution is

$$\hat{f}(x) = \sum_{n=1}^{N} g_n e^{inx}. \tag{29.23}$$

29.5.2 The Over-Sampled Case

Suppose that $f(x) = 0$ for $|x| > A$, where $0 < A < \pi$. Then we use $L^2(-A, A)$ as the Hilbert space. For equi-spaced data at $\omega_n = n$, we have

$$g_n = \int_{-\pi}^{\pi} f(x) \chi_A(x) e^{-inx} dx, \tag{29.24}$$

so that the minimum-norm solution has the form

$$\hat{f}(x) = \chi_A(x) \sum_{m=1}^{N} a_m e^{imx}, \tag{29.25}$$

with

$$g_n = 2 \sum_{m=1}^{N} a_m \frac{\sin A(m-n)}{m-n}. \tag{29.26}$$

The minimum-norm solution is support-limited to $[-A, A]$ and consistent with the Fourier-transform data.

29.5.3 Using a Prior Estimate of f

Suppose that $f(x) = 0$ for $|x| > \pi$ again, and that $p(x)$ satisfies

$$0 < \epsilon \leq p(x) \leq E < +\infty, \tag{29.27}$$

for all x in $[-\pi, \pi]$. Define the operator $\mathcal{T}$ by $(\mathcal{T}f)(x) = \sqrt{p(x)}f(x)$. The $\mathcal{T}$-norm is then

$$\langle f, h \rangle_{\mathcal{T}} = \int_{-\pi}^{\pi} f(x)\overline{h(x)}p(x)^{-1}dx. \tag{29.28}$$

It follows that

$$g_n = \int_{-\pi}^{\pi} f(x)p(x)e^{-i\omega_n x}p(x)^{-1}dx, \tag{29.29}$$

so that the minimum $\mathcal{T}$-norm solution is

$$\hat{f}(x) = \sum_{m=1}^{N} a_m p(x) e^{i\omega_m x} = p(x) \sum_{m=1}^{N} a_m e^{i\omega_m x}, \tag{29.30}$$

where

$$g_n = \sum_{m=1}^{N} a_m \int_{-\pi}^{\pi} p(x) e^{i(\omega_m - \omega_n)x} dx. \tag{29.31}$$

If we have prior knowledge about the support of f, or some idea of its shape, we can incorporate that prior knowledge into the reconstruction through the choice of $p(x)$.

The reconstruction in Equation (29.30) was presented in [25], where it was called the PDFT method. The PDFT was based on a non-iterative version of the Gerchberg-Papoulis bandlimited extrapolation procedure,

discussed earlier in [24]. The PDFT was then applied to image reconstruction problems in [26]. An application of the PDFT was presented in [28]. In [27] we extended the PDFT to a nonlinear version, the indirect PDFT (IPDFT), that generalizes Burg's maximum entropy spectrum estimation method. The PDFT was applied to the phase problem in [29] and in [30] both the PDFT and IPDFT were examined in the context of Wiener filter approximation. More recent work on these topics is discussed in the book [44].

When N, the number of data values, is not large, the PDFT can be implemented in a straight-forward manner, by first calculating the matrix P that appears in Equation (29.31), with entries

$$P_{n,m} = \int_{-\pi}^{\pi} p(x) e^{i(\omega_m - \omega_n)x} dx,$$

solving Equation (29.31) for the coefficients a_m, and finally, inserting these coefficients in Equation (29.30). When N is large, calculating the entries of the matrix P can be an expensive step. Since, in such cases, solving the system in Equation (29.31) will probably be done iteratively, it makes sense to consider an iterative alternative to the PDFT that avoids the use of the matrix P. This is the *discrete* PDFT (DPDFT).

The Discrete PDFT (DPDFT)

The PDFT uses the estimate $\hat{f}(x)$ of $f(x)$, consistent with the data, that has the minimum weighted norm

$$\int_{-\pi}^{\pi} |\hat{f}(x)|^2 p(x)^{-1} dx.$$

The discrete PDFT (DPDFT) replaces the functions $f(x)$ and $p(x)$ with finite vectors $f = (f_1, ..., f_J)^T$ and $p = (p_1, ..., p_J)^T$, for some $J > N$; for example, we could have $f_j = f(x_j)$ for some sample points x_j in $(-\pi, \pi)$. The vector p must have positive entries. The integrals that appear in Equation (29.20) are replaced by sums

$$g_n = \sum_{j=1}^{J} f_j E_{n,j}; \tag{29.32}$$

for example, we could use $E_{n,j} = \exp(-i\omega_n x_j)$. Now our estimate is the solution of the system $g = Ef$ for which the weighted norm

$$\sum_{j=1}^{J} |f_j|^2 p_j^{-1}$$

is minimized. To obtain this minimum-weighted-norm solution, we can use the ART algorithm.

The ART will give the minimum-norm solution of $Au = v$ if we begin the iteration at $u^0 = 0$. To obtain the solution with minimum weighted norm

$$\sum_{j=1}^{J} |u_j|^2 p_j^{-1},$$

we replace u_j with $u_j p_j^{-1/2}$, and $A_{n,j}$ with $A_{n,j} p_j^{1/2}$, and then apply the ART.

Chapter 30

Appendix: Compressed Sensing

One area that has attracted much attention lately is *compressed sensing* or *compressed sampling* (CS) [73]. For applications such as medical imaging, CS may provide a means of reducing radiation dosage to the patient without sacrificing image quality. An important aspect of CS is finding sparse solutions of under-determined systems of linear equations, which can often be accomplished by one-norm minimization. Perhaps the best reference to date on CS is [21].

30.1 Compressed Sensing

The objective in CS is exploit sparseness to reconstruct a vector f in R^J from relatively few linear functional measurements [73].

Let $U = \{u^1, u^2, ..., u^J\}$ and $V = \{v^1, v^2, ..., v^J\}$ be two orthonormal bases for R^J, with all members of R^J represented as column vectors. For $i = 1, 2, ..., J$, let

$$\mu_i = \max_{1 \leq j \leq J} \{|\langle u^i, v^j \rangle|\}$$

and

$$\mu(U, V) = \max\{\mu_i \, | i = 1, ..., I\}.$$

We know from Cauchy's Inequality that

$$|\langle u^i, v^j \rangle| \leq 1,$$

and from Parseval's Equation

$$\sum_{j=1}^{J} |\langle u^i, v^j \rangle|^2 = ||u^i||^2 = 1.$$

Therefore, we have

$$\frac{1}{\sqrt{J}} \leq \mu(U,V) \leq 1.$$

The quantity $\mu(U,V)$ is the *coherence* measure of the two bases; the closer $\mu(U,V)$ is to the lower bound of $\frac{1}{\sqrt{J}}$, the more *incoherent* the two bases are.

Let f be a fixed member of R^J; we expand f in the V basis as

$$f = x_1 v^1 + x_2 v^2 + ... + x_J v^J.$$

We say that the coefficient vector $x = (x_1, ..., x_J)$ is s-sparse if s is the number of non-zero x_j.

If s is small, most of the x_j are zero, but since we do not know which ones these are, we would have to compute all the linear functional values

$$x_j = \langle f, v^j \rangle$$

to recover f exactly. In fact, the smaller s is, the harder it would be to learn anything from randomly selected x_j, since most would be zero. The idea in CS is to obtain measurements of f with members of a different orthonormal basis, which we call the U basis. If the members of U are very much like the members of V, then nothing is gained. But, if the members of U are quite unlike the members of V, then each inner product measurement

$$y_i = \langle f, u^i \rangle = f^T u^i$$

should tell us something about f. If the two bases are sufficiently incoherent, then relatively few y_i values should tell us quite a bit about f. Specifically, we have the following result due to Candès and Romberg [51]: suppose the coefficient vector x for representing f in the V basis is s-sparse. Select uniformly randomly $M \leq J$ members of the U basis and compute the measurements $y_i = \langle f, u^i \rangle$. Then, if M is sufficiently large, it is highly probable that $z = x$ also solves the problem of minimizing the one-norm

$$||z||_1 = |z_1| + |z_2| + ... + |z_J|,$$

subject to the conditions

$$y_i = \langle g, a^i \rangle = g^T u^i,$$

for those M randomly selected u^i, where

$$g = z_1 v^1 + z_2 v^2 + ... + z_J v^J.$$

The smaller $\mu(U,V)$ is, the smaller the M is permitted to be without reducing the probability of perfect reconstruction.

30.2 Sparse Solutions

Suppose that A is a real M by N matrix, with $M < N$, and that the linear system $Ax = b$ has infinitely many solutions. For any vector x, we define the *support* of x to be the subset S of $\{1, 2, ..., N\}$ consisting of those n for which the entries $x_n \neq 0$. For any under-determined system $Ax = b$, there will, of course, be at least one solution of minimum support, that is, for which $s = |S|$, the size of the support set S, is minimum. However, finding such a maximally sparse solution requires combinatorial optimization, and is known to be computationally difficult. It is important, therefore, to have a computationally tractable method for finding maximally sparse solutions.

30.2.1 Maximally Sparse Solutions

Consider the problem P_0: among all solutions x of the consistent system $b = Ax$, find one, call it $\hat{x}$, that is maximally sparse, that is, has the minimum number of non-zero entries. Obviously, there will be at least one such solution having minimal support, but finding one, however, is a combinatorial optimization problem and is generally NP-hard.

30.2.2 Minimum One-Norm Solutions

Instead, we can seek a *minimum one-norm* solution, that is, solve the problem P_1: minimize

$$||x||_1 = \sum_{n=1}^{N} |x_n|,$$

subject to $Ax = b$. Problem P_1 can be formulated as a linear programming problem, so is more easily solved. The big questions are: when does P_1 have a unique solution, and when is it $\hat{x}$? The problem P_1 will have a unique solution if and only if A is such that the one-norm satisfies

$$||\hat{x}||_1 < ||\hat{x} + v||_1,$$

for all non-zero v in the null space of A.

30.2.3 Why the One-Norm?

When a system of linear equations $Ax = b$ is under-determined, we can find the *minimum-two-norm solution* that minimizes the square of the two-norm,

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

subject to $Ax = b$. One drawback to this approach is that the two-norm penalizes relatively large values of x_n much more than the smaller ones, so tends to provide non-sparse solutions. Alternatively, we may seek the solution for which the one-norm,

$$||x||_1 = \sum_{n=1}^{N} |x_n|,$$

is minimized. The one-norm still penalizes relatively large entries x_n more than the smaller ones, but much less than the two-norm does. As a result, it often happens that the minimum one-norm solution actually solves P_0 as well.

30.2.4 Comparison with the PDFT

The PDFT approach to solving the under-determined system $Ax = b$ is to select weights $w_n > 0$ and then to find the solution $\tilde{x}$ that minimizes the weighted two-norm given by

$$\sum_{n=1}^{N} |x_n|^2 w_n.$$

Our intention is to select weights w_n so that w_n^{-1} is reasonably close to $|\hat{x}_n|$; consider, therefore, what happens when $w_n^{-1} = |\hat{x}_n|$. We claim that $\tilde{x}$ is also a minimum-one-norm solution.

To see why this is true, note that, for any x, we have

$$\sum_{n=1}^{N} |x_n| = \sum_{n=1}^{N} \frac{|x_n|}{\sqrt{|\hat{x}_n|}} \sqrt{|\hat{x}_n|}$$

$$\leq \sqrt{\sum_{n=1}^{N} \frac{|x_n|^2}{|\hat{x}_n|}} \sqrt{\sum_{n=1}^{N} |\hat{x}_n|}.$$

Therefore,

$$\sum_{n=1}^{N} |\tilde{x}_n| \leq \sqrt{\sum_{n=1}^{N} \frac{|\tilde{x}_n|^2}{|\hat{x}_n|}} \sqrt{\sum_{n=1}^{N} |\hat{x}_n|}$$

$$\leq \sqrt{\sum_{n=1}^{N} \frac{|\hat{x}_n|^2}{|\hat{x}_n|}} \sqrt{\sum_{n=1}^{N} |\hat{x}_n|} = \sum_{n=1}^{N} |\hat{x}_n|.$$

Therefore, $\tilde{x}$ also minimizes the one-norm.

30.2.5 Iterative Reweighting

We want each weight w_n to be a good prior estimate of the reciprocal of $|\hat{x}_n|$. Because we do not yet know $\hat{x}$, we may take a sequential-optimization approach, beginning with weights $w_n^0 > 0$, finding the PDFT solution using these weights, then using this PDFT solution to get a (we hope!) a better choice for the weights, and so on. This sequential approach was successfully implemented in the early 1980's by Michael Fiddy and his students [86].

In [52], the same approach is taken, but with respect to the one-norm. Since the one-norm still penalizes larger values disproportionately, balance can be achieved by minimizing a weighted-one-norm, with weights close to the reciprocals of the $|\hat{x}_n|$. Again, not yet knowing $\hat{x}$, they employ a sequential approach, using the previous minimum-weighted-one-norm solution to obtain the new set of weights for the next minimization. At each step of the sequential procedure, the previous reconstruction is used to estimate the true support of the desired solution.

It is interesting to note that an on-going debate among users of the PDFT has been the nature of the prior weighting. Does w_n approximate $|x_n|$ or $|x_n|^2$? This is close to the issue treated in [52], the use of a weight in the minimum-one-norm approach.

It should be noted again that finding a sparse solution is not usually the goal in the use of the PDFT, but the use of the weights has much the same effect as using the one-norm to find sparse solutions: to the extent that the weights approximate the entries of $\hat{x}$, their use reduces the penalty associated with the larger entries of an estimated solution.

30.3 Why Sparseness?

One obvious reason for wanting sparse solutions of $Ax = b$ is that we have prior knowledge that the desired solution is sparse. Such a problem arises in signal analysis from Fourier-transform data. In other cases, such as in the reconstruction of locally constant signals, it is not the signal itself, but its discrete derivative, that is sparse.

30.3.1 Signal Analysis

Suppose that our signal $f(t)$ is known to consist of a small number of complex exponentials, so that $f(t)$ has the form

$$f(t) = \sum_{j=1}^{J} a_j e^{i\omega_j t},$$

for some small number of frequencies ω_j in the interval $[0, 2\pi)$. For $n = 0, 1, ..., N-1$, let $f_n = f(n)$, and let f be the N-vector with entries f_n;

we assume that J is much smaller than N. The discrete (vector) Fourier transform of f is the vector $\hat{f}$ having the entries

$$\hat{f}_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f_n e^{2\pi i k n/N},$$

for $k = 0, 1, ..., N-1$; we write $\hat{f} = Ef$, where E is the N by N matrix with entries $E_{kn} = \frac{1}{\sqrt{N}} e^{2\pi i k n/N}$. If N is large enough, we may safely assume that each of the ω_j is equal to one of the frequencies $2\pi i k$ and that the vector $\hat{f}$ is J-sparse. The question now is: How many values of $f(n)$ do we need to calculate in order to be sure that we can recapture $f(t)$ exactly? We have the following theorem [50]:

Theorem 30.1 *Let N be prime. Let S be any subset of $\{0, 1, ..., N-1\}$ with $|S| \geq 2J$. Then the vector $\hat{f}$ can be uniquely determined from the measurements f_n for n in S.*

We know that

$$f = E^\dagger \hat{f},$$

where $E^\dagger$ is the conjugate transpose of the matrix E. The point here is that, for any matrix R obtained from the identity matrix I by deleting $N - |S|$ rows, we can recover the vector $\hat{f}$ from the measurements Rf.

If N is not prime, then the assertion of the theorem may not hold, since we can have $n = 0 \bmod N$, without $n = 0$. However, the assertion remains valid for most sets of J frequencies and most subsets S of indices; therefore, with high probability, we can recover the vector $\hat{f}$ from Rf.

Note that the matrix E is *unitary*, that is, $E^\dagger E = I$, and, equivalently, the columns of E form an orthonormal basis for C^N. The data vector is

$$b = Rf = RE^\dagger \hat{f}.$$

In this example, the vector f is not sparse, but can be represented sparsely in a particular orthonormal basis, namely as $f = E^\dagger \hat{f}$, using a sparse vector $\hat{f}$ of coefficients. The *representing basis* then consists of the columns of the matrix $E^\dagger$. The measurements pertaining to the vector f are the values f_n, for n in S. Since f_n can be viewed as the inner product of f with δ^n, the nth column of the identity matrix I, that is,

$$f_n = \langle \delta^n, f \rangle,$$

the columns of I provide the so-called *sampling basis*. With $A = RE^\dagger$ and $x = \hat{f}$, we then have

$$Ax = b,$$

with the vector x sparse. It is important for what follows to note that the matrix A is random, in the sense that we choose which rows of I to use to form R.

30.3.2 Locally Constant Signals

Suppose now that the function $f(t)$ is locally constant, consisting of some number of horizontal lines. We discretize the function $f(t)$ to get the vector $f = (f(0), f(1), ..., f(N))^T$. The discrete derivative vector is $g = (g_1, g_2, ..., g_N)^T$, with

$$g_n = f(n) - f(n-1).$$

Since $f(t)$ is locally constant, the vector g is sparse. The data we will have will not typically be values $f(n)$. The goal will be to recover f from M linear functional values pertaining to f, where M is much smaller than N. We shall assume, from now on, that we have measured, or can estimate, the value $f(0)$.

Our M by 1 data vector d consists of measurements pertaining to the vector f:

$$d_m = \sum_{n=0}^{N} H_{mn} f_n,$$

for $m = 1, ..., M$, where the H_{mn} are known. We can then write

$$d_m = f(0)\Big(\sum_{n=0}^{N} H_{mn}\Big) + \sum_{k=1}^{N}\Big(\sum_{j=k}^{N} H_{mj}\Big) g_k.$$

Since $f(0)$ is known, we can write

$$b_m = d_m - f(0)\Big(\sum_{n=0}^{N} H_{mn}\Big) = \sum_{k=1}^{N} A_{mk} g_k,$$

where

$$A_{mk} = \sum_{j=k}^{N} H_{mj}.$$

The problem is then to find a sparse solution of $Ax = g$. As in the previous example, we often have the freedom to select the linear functions, that is, the values H_{mn}, so the matrix A can be viewed as random.

30.3.3 Tomographic Imaging

The reconstruction of tomographic images is an important aspect of medical diagnosis, and one that combines aspects of both of the previous examples. The data one obtains from the scanning process can often be interpreted as values of the Fourier transform of the desired image; this is precisely the case in magnetic-resonance imaging, and approximately true for x-ray transmission tomography, positron-emission tomography (PET)

and single-photon emission tomography (SPECT). The images one encounters in medical diagnosis are often approximately locally constant, so the associated array of discrete partial derivatives will be sparse. If this sparse derivative array can be recovered from relatively few Fourier-transform values, then the scanning time can be reduced.

We turn now to the more general problem of compressed sampling.

30.4 Compressed Sampling

Our goal is to recover the vector $f = (f_1, ..., f_N)^T$ from M linear functional values of f, where M is much less than N. In general, this is not possible without prior information about the vector f. In compressed sampling, the prior information concerns the sparseness of either f itself, or another vector linearly related to f.

Let U and V be unitary N by N matrices, so that the column vectors of both U and V form orthonormal bases for C^N. We shall refer to the bases associated with U and V as the *sampling basis* and the *representing basis*, respectively. The first objective is to find a unitary matrix V so that $f = Vx$, where x is sparse. Then we want to find a second unitary matrix U such that, when an M by N matrix R is obtained from U by deleting rows, the sparse vector x can be determined from the data $b = RVx = Ax$. Theorems in compressed sensing describe properties of the matrices U and V such that, when R is obtained from U by a random selection of the rows of U, the vector x will be uniquely determined, with high probability, as the unique solution that minimizes the one-norm.

Chapter 31

Appendix: Urn Models

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.

31.1 The Urn Model for Remote Sensing

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

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

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

If all the marbles of one color are in a single urn, the problem is trivial; when I hear a color, I know immediately which urn contained that marble. My list of colors is then a list of urn numbers; I have the complete data

now. My estimate of the number of pieces of paper containing the urn number j is then simply N times the proportion of draws that resulted in urn j being selected.

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

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

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

31.2 Hidden Markov Models

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

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

According to the hidden Markov model, the probability that my as-

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

There are several distinct objectives one can have, when using HMM. We assume throughout this subsection that the data is the list of colors, **c**.

- **Evaluation:** For given probabilities a_{ij} and b_{jk}, what is the probability that the list **c** was generated according to the HMM? Here, the objective is to see if the model is a good description of the data.
- **Decoding:** Given the model, the probabilities and the list **c**, what list $\mathbf{j} = \{j_1, j_2, ..., j_N\}$ of potential visited urns is the most likely? Now, we want to infer the hidden states from the visible ones.
- **Learning:** We are told that there are J urns and I colors, but are not told the probabilities a_{ij} and b_{jk}. We are given several data vectors **c** generated by the HMM; these are the *training sets*. The objective is to learn the probabilities.

Once again, the EM algorithm can play a role in solving these problems [77].

Bibliography

[1] Albright, B. (2007) "An Introduction to simulated annealing. " *The College Mathematics Journal*, **38(1)**, pp. 37–42.

[2] Anderson, A. and Kak, A. (1984) "Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm." *Ultrasonic Imaging*, **6** pp. 81–94.

[3] Attouch, H. (1984) *Variational Convergence for Functions and Operators*, Boston: Pitman Advanced Publishing Program.

[4] Attouch, H., and Wets, R. (1989) "Epigraphical Analysis." *Ann. Inst. Poincare: Anal. Nonlineaire*, **6**.

[5] Aubin, J.-P., (1993) *Optima and Equilibria: An Introduction to Nonlinear Analysis*, Springer-Verlag.

[6] 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.

[7] Axelsson, O. (1994) *Iterative Solution Methods*. Cambridge, UK: Cambridge University Press.

[8] Baillon, J.-B., Bruck, R.E., and Reich, S. (1978) "On the asymptotic behavior of nonexpansive mappings and semigroups in Banach spaces." *Houston Journal of Mathematics*, **4**, pp. 1–9.

[9] Bauschke, H. (1996) "The approximation of fixed points of compositions of nonexpansive mappings in Hilbert space." *Journal of Mathematical Analysis and Applications*, **202**, pp. 150–159.

[10] Bauschke, H., and Borwein, J. (1996) "On projection algorithms for solving convex feasibility problems." *SIAM Review*, **38 (3)**, pp. 367–426.

[11] Bauschke, H., and Borwein, J. (1997) "Legendre functions and the method of random Bregman projections." *Journal of Convex Analysis*, **4**, pp. 27–67.

[12] Bertero, M., and Boccacci, P. (1998) *Introduction to Inverse Problems in Imaging* Bristol, UK: Institute of Physics Publishing.

[13] Bertsekas, D.P. (1997) "A new class of incremental gradient methods for least squares problems." *SIAM J. Optim.*, **7**, pp. 913-926.

[14] Bliss, G.A. (1925) *Calculus of Variations* Carus Mathematical Monographs, American Mathematical Society.

[15] Borwein, J. and Lewis, A. (2000) *Convex Analysis and Nonlinear Optimization*. Canadian Mathematical Society Books in Mathematics, New York: Springer-Verlag.

[16] Boyd, S., and Vandenberghe, L. (2004) *Convex Optimization*. Cambridge, England: Cambridge University Press.

[17] 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.

[18] Bregman, L., Censor, Y., and Reich, S. (1999) "Dykstra's algorithm as the nonlinear extension of Bregman's optimization method." *Journal of Convex Analysis*, **6 (2)**, pp. 319–333.

[19] Browne, J. and A. DePierro, A. (1996) "A row-action alternative to the EM algorithm for maximizing likelihoods in emission tomography." *IEEE Trans. Med. Imag.* **15**, pp. 687–699.

[20] Bruck, R.E., and Reich, S. (1977) "Nonexpansive projections and resolvents of accretive operators in Banach spaces." *Houston Journal of Mathematics*, **3**, pp. 459–470.

[21] Bruckstein, A., Donoho, D., and Elad, M. (2009) "From sparse solutions of systems of equations to sparse modeling of signals and images." *SIAM Review*, **51(1)**, pp. 34–81.

[22] Burden, R.L., and Faires, J.D. (1993) *Numerical Analysis*, Boston: PWS-Kent.

[23] Butnariu, D., Byrne, C., and Censor, Y. (2003) "Redundant axioms in the definition of Bregman functions." *Journal of Convex Analysis*, **10**, pp. 245–254.

[24] 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.

[25] Byrne, C. and Fitzgerald, R. (1982) "Reconstruction from Partial Information, with Applications to Tomography." *SIAM J. Applied Math.* **42(4)**, pp. 933–940.

[26] 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.

[27] 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.

[28] 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.

[29] Byrne, C. and Fiddy, M. (1987) "Estimation of continuous object distributions from Fourier magnitude measurements." *JOSA A* **4**, pp. 412–417.

[30] Byrne, C. and Fiddy, M. (1988) "Images as power spectra; reconstruction as Wiener filter approximation." *Inverse Problems* **4**, pp. 399–409.

[31] Byrne, C. (1993) "Iterative image reconstruction algorithms based on cross-entropy minimization." *IEEE Transactions on Image Processing* **IP-2**, pp. 96–103.

[32] 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.

[33] 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.

[34] Byrne, C. (1996) "Block-iterative methods for image reconstruction from projections." *IEEE Transactions on Image Processing* **IP-5**, pp. 792–794.

[35] Byrne, C. (1997) "Convergent block-iterative algorithms for image reconstruction from inconsistent data." *IEEE Transactions on Image Processing* **IP-6**, pp. 1296–1304.

[36] 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.

[37] Byrne, C. (1998) "Iterative algorithms for deblurring and deconvolution with constraints." *Inverse Problems*, **14**, pp. 1455–1467.

[38] Byrne, C. (2000) "Block-iterative interior point optimization methods for image reconstruction from limited data." *Inverse Problems* **16**, pp. 1405–1419.

[39] Byrne, C. (2001) "Bregman-Legendre Multidistance Projection Algorithms for Convex Feasibility and Optimization." In *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications*, edited by D. Butnariu, Y. Censor and S. Reich, pp. 87-100, Studies in Computational Mathematics 8. Amsterdam: Elsevier Publ., 2001.

[40] 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.

[41] Byrne, C. (2002) "Iterative oblique projection onto convex sets and the split feasibility problem." *Inverse Problems* **18**, pp. 441–453.

[42] Byrne, C. (2004) "A unified treatment of some iterative algorithms in signal processing and image reconstruction." *Inverse Problems* **20**, pp. 103–120.

[43] Byrne, C. (2005) "Choosing parameters in block-iterative or ordered-subset reconstruction algorithms." *IEEE Transactions on Image Processing*, **14 (3)**, pp. 321–327.

[44] Byrne, C. (2005) *Signal Processing: A Mathematical Approach*, AK Peters, Publ., Wellesley, MA.

[45] Byrne, C. (2007) *Applied Iterative Methods*, AK Peters, Publ., Wellesley, MA.

[46] Byrne, C. (2008) "Sequential unconstrained minimization algorithms for constrained optimization." *Inverse Problems*, **24**.

[47] Byrne, C. (2009) "Block-iterative algorithms." *International Transactions in Operations Research*, to appear.

[48] Byrne, C. (2009) "Bounds on the largest singular value of a matrix and the convergence of simultaneous and block-iterative algorithms for sparse linear systems." *International Transactions in Operations Research*, to appear.

[49] Byrne, C., and Ward, S. (2005) "Estimating the largest singular value of a sparse matrix." unpublished notes.

[50] Candès, E., Romberg, J., and Tao, T. (2006) "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information"*IEEE Transactions on Information Theory*, **52(2)**, pp. 489–509.

[51] Candès, E., and Romberg, J. (2007) "Sparsity and incoherence in compressive sampling"*Inverse Problems*, **23(3)**, pp. 969–985.

[52] Candès, E., Wakin, M., and Boyd, S. (2007) "Enhancing sparsity by reweighted l_1 minimization" preprint available at http://www.acm.caltech.edu/ emmanuel/publications.html .

[53] 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.

[54] Censor, Y., Eggermont, P.P.B., and Gordon, D. (1983) "Strong underrelaxation in Kaczmarz's method for inconsistent systems."*Numerische Mathematik* **41**, pp. 83–92.

[55] Censor, Y. and Elfving, T. (1994) "A multi-projection algorithm using Bregman projections in a product space." *Numerical Algorithms*, **8** 221–239.

[56] Censor, Y., Elfving, T., Herman, G.T., and Nikazad, T. (2008) "On diagonally-relaxed orthogonal projection methods." *SIAM Journal on Scientific Computation*, **30(1)**, pp. 473–504.

[57] Censor, Y., Elfving, T., Kopf, N., and Bortfeld, T. "The Multiple-sets Split Feasibility Problem and its Application for Inverse Problems." *Inverse Problems* 21 (2005), 2071-2084.

[58] Censor, Y., Gordon, D., and Gordon, R. (2001) "Component averaging: an efficient iterative parallel algorithm for large and sparse unstructured problems." *Parallel Computing*, **27**, pp. 777–808.

[59] Censor, Y., Gordon, D., and Gordon, R. (2001) "BICAV: A block-iterative, parallel algorithm for sparse systems with pixel-related weighting." *IEEE Transactions on Medical Imaging*, **20**, pp. 1050–1060.

[60] Censor, Y., and Reich, S. (1998) "The Dykstra algorithm for Bregman projections." *Communications in Applied Analysis*, **2**, pp. 323–339.

[61] Censor, Y., and Reich, S. (1996) "Iterations of paracontractions and firmly nonexpansive operators with applications to feasibility and optimization." *Optimization*, **37**, pp. 323–339.

[62] Censor, Y. and Segman, J. (1987) "On block-iterative maximization." *J. of Information and Optimization Sciences* **8**, pp. 275–291.

[63] Censor, Y., and Zenios, S.A. (1992) "Proximal minimization algorithm with D-functions." *Journal of Optimization Theory and Applications*, **73(3)**, pp. 451–464.

[64] Cimmino, G. (1938) "Calcolo approssimato per soluzioni dei sistemi di equazioni lineari." *La Ricerca Scientifica XVI, Series II, Anno IX* **1**, pp. 326–333.

[65] Combettes, P. (2000) "Fejér monotonicity in convex optimization." in *Encyclopedia of Optimization*, C.A. Floudas and P. M. Pardalos, editors, Boston: Kluwer Publ.

[66] Combettes, P., and Wajs, V. (2005) "Signal recovery by proximal forward-backward splitting." *Multiscale Modeling and Simulation*, **4(4)**, pp. 1168–1200.

[67] Csiszár, I. and Tusnády, G. (1984) "Information geometry and alternating minimization procedures." *Statistics and Decisions* **Supp. 1**, pp. 205–237.

[68] Darroch, J. and Ratcliff, D. (1972) "Generalized iterative scaling for log-linear models." *Annals of Mathematical Statistics* **43**, pp. 1470–1480.

[69] 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.

[70] De Pierro, A. and Iusem, A. (1990) "On the asymptotic behavior of some alternate smoothing series expansion iterative methods." *Linear Algebra and its Applications* **130**, pp. 3–24.

[71] Deutsch, F., and Yamada, I. (1998) "Minimizing certain convex functions over the intersection of the fixed point sets of non-expansive mappings." *Numerical Functional Analysis and Optimization*, **19**, pp. 33–56.

[72] Dines, K., and Lyttle, R. (1979) "Computerized geophysical tomography." *Proc. IEEE*, **67**, pp. 1065–1073.

[73] Donoho, D. (2006) "Compressed sampling" *IEEE Transactions on Information Theory*, **52 (4)**. (download preprints at http://www.stat.stanford.edu/ donoho/Reports).

[74] Dorfman, R., Samuelson, P., and Solow, R. (1958) *Linear Programming and Economic Analysis*. New York: McGraw- Hill.

[75] Driscoll, P., and Fox, W. (1996) "Presenting the Kuhn-Tucker conditions using a geometric method." *The College Mathematics Journal*, **38 (1)**, pp. 101–108.

[76] Duffin, R., Peterson, E., and Zener, C. (1967) *Geometric Programming: Theory and Applications*. New York: Wiley.

[77] Duda, R., Hart, P., and Stork, D. (2001) *Pattern Classification*, Wiley.

[78] Dugundji, J. (1970) *Topology* Boston: Allyn and Bacon, Inc.

[79] Dykstra, R. (1983) "An algorithm for restricted least squares regression." *J. Amer. Statist. Assoc.*, **78 (384)**, pp. 837–842.

[80] 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.

[81] Elsner, L., Koltracht, L., and Neumann, M. (1992) "Convergence of sequential and asynchronous nonlinear paracontractions." *Numerische Mathematik*, **62**, pp. 305–319.

[82] Fang, S-C.,and Puthenpura, S. (1993) *Linear Optimization and Extensions: Theory and Algorithms*. New Jersey: Prentice-Hall.

[83] Farkas, J. (1902) "Über die Theorie der einfachen Ungleichungen." *J. Reine Angew. Math.*, **124**, pp. 1–24.

[84] Farncombe, T. (2000) "Functional dynamic SPECT imaging using a single slow camera rotation." *Ph.D. thesis, Dept. of Physics, University of British Columbia*.

[85] Fiacco, A., and McCormick, G. (1990) *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Philadelphia, PA: SIAM Classics in Mathematics (reissue).

[86] Fiddy, M. (2008) *private communication*.

[87] Fleming, W. (1965) *Functions of Several Variables*. Reading, MA: Addison-Wesley.

[88] Gale, D. (1960) *The Theory of Linear Economic Models*. New York: McGraw-Hill.

[89] 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.

[90] Gill, P., Murray, W., Saunders, M., Tomlin, J., and Wright, M. (1986) "On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method." *Mathematical Programming*, **36**, pp. 183–209.

[91] Goebel, K., and Reich, S. (1984) *Uniform Convexity, Hyperbolic Geometry, and Nonexpansive Mappings*, New York: Dekker.

[92] Golshtein, E., and Tretyakov, N. (1996) *Modified Lagrangians and Monotone Maps in Optimization*. New York: John Wiley and Sons, Inc.

[93] Gordan, P. (1873) "Über die Auflösungen linearer Gleichungen mit reelen Coefficienten." *Math. Ann.*, **6**, pp. 23–28.

[94] Gordon, R., Bender, R., and Herman, G.T. (1970) "Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography." *J. Theoret. Biol.* **29**, pp. 471–481.

[95] Gordon, D., and Gordon, R.(2005) "Component-averaged row projections: A robust block-parallel scheme for sparse linear systems." *SIAM Journal on Scientific Computing*, **27**, pp. 1092–1117.

[96] Gubin, L.G., Polyak, B.T. and Raik, E.V. (1967) "The method of projections for finding the common point of convex sets." *USSR Computational Mathematics and Mathematical Physics*, **7**: 1–24.

[97] Hager, B., Clayton, R., Richards, M., Comer, R., and Dziewonsky, A. (1985) "Lower mantle heterogeneity, dynamic typography and the geoid." *Nature*, **313**, pp. 541–545.

[98] Herman, G. T. (1999) *private communication*.

[99] Herman, G. T. and Meyer, L. (1993) "Algebraic reconstruction techniques can be made computationally efficient." *IEEE Transactions on Medical Imaging* **12**, pp. 600–609.

[100] Holte, S., Schmidlin, P., Linden, A., Rosenqvist, G. and Eriksson, L. (1990) "Iterative image reconstruction for positron emission tomography: a study of convergence and quantitation problems." *IEEE Transactions on Nuclear Science* **37**, pp. 629–635.

[101] Hudson, M., Hutton, B., and Larkin, R. (1992) "Accelerated EM reconstruction using ordered subsets." *Journal of Nuclear Medicine*, **33**, p.960.

[102] Hudson, H.M. and Larkin, R.S. (1994) "Accelerated image reconstruction using ordered subsets of projection data." *IEEE Transactions on Medical Imaging* **13**, pp. 601–609.

[103] Jiang, M., and Wang, G. (2003) "Convergence studies on iterative algorithms for image reconstruction." *IEEE Transactions on Medical Imaging*, **22(5)**, pp. 569–579.

[104] Kaczmarz, S. (1937) "Angenäherte Auflösung von Systemen linearer Gleichungen." *Bulletin de l'Academie Polonaise des Sciences et Lettres* **A35**, pp. 355–357.

[105] Karmarkar, N. (1984) "A new polynomial-time algorithm for linear programming." *Combinatorica*, **4**, pp. 373–395.

[106] Körner, T. (1996) *The Pleasures of Counting*. Cambridge, UK: Cambridge University Press.

[107] Kuhn, H., and Tucker, A. (eds.) (1956) *Linear Inequalities and Related Systems*. Annals of Mathematical Studies, No. 38. New Jersey: Princeton University Press.

[108] Kullback, S. and Leibler, R. (1951) "On information and sufficiency." *Annals of Mathematical Statistics* **22**, pp. 79–86.

[109] Lagarias, J., Reeds, J., Wright, M., and Wright, P. (1998) "Convergence properties of the Nelder-Mead simplex method in low dimensions." *SIAM Journal of Optimization*, **9(1)**, pp. 112–147.

[110] Landweber, L. (1951) "An iterative formula for Fredholm integral equations of the first kind." *Amer. J. of Math.* **73**, pp. 615–624.

[111] Lange, K. and Carson, R. (1984) "EM reconstruction algorithms for emission and transmission tomography." *Journal of Computer Assisted Tomography* **8**, pp. 306–316.

[112] 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.

[113] Leahy, R. and Byrne, C. (2000) "Guest editorial: Recent development in iterative image reconstruction for PET and SPECT." *IEEE Trans. Med. Imag.* **19**, pp. 257–260.

[114] Lent, A., and Censor, Y. (1980) "Extensions of Hildreth's row-action method for quadratic programming." *SIAM Journal on Control and Optimization*, **18**, pp. 444–454.

[115] Luenberger, D. (1969) *Optimization by Vector Space Methods*. New York: John Wiley and Sons, Inc.

[116] Mann, W. (1953) "Mean value methods in iteration." *Proc. Amer. Math. Soc.* **4**, pp. 506–510.

[117] Marlow, W. (1978) *Mathematics for Operations Research*. New York: John Wiley and Sons. Reissued 1993 by Dover.

[118] Marzetta, T. (2003) "Reflection coefficient (Schur parameter) representation for convex compact sets in the plane,." *IEEE Transactions on Signal Processing*, **51 (5)**, pp. 1196–1210.

[119] McKinnon, K. (1998) "Convergence of the Nelder-Mead simplex method to a non-stationary point." *SIAM Journal on Optimization*, **9(1)**, pp. 148–158.

[120] McLachlan, G.J. and Krishnan, T. (1997) *The EM Algorithm and Extensions*. New York: John Wiley and Sons, Inc.

[121] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953) "Equation of state calculations by fast computing machines" *J. Chem. Phys.* **21**, pp. 1087–1091.

[122] Nash, S. and Sofer, A. (1996) *Linear and Nonlinear Programming*. New York: McGraw-Hill.

[123] Nelder, J., and Mead, R. (1965) "A simplex method for function minimization" *Computing Journal*, **7**, pp. 308–313.

[124] Nesterov, Y., and Nemirovski, A. (1994) *Interior-Point Polynomial Algorithms in Convex Programming*. Philadelphia, PA: SIAM Studies in Applied Mathematics.

[125] von Neumann, J., and Morgenstern, O. (1944) *Theory of Games and Economic Behavior*. New Jersey: Princeton University Press.

[126] Niven, I. (1981) *Maxima and Minima Without Calculus*. Mathematical Association of America.

[127] J. Ortega and W. Rheinboldt. (2000) *Iterative Solution of Nonlinear Equations in Several Variables*, Classics in Applied Mathematics, 30. Philadelphia, PA: SIAM, 2000

[128] Papoulis, A. (1977) *Signal Analysis*. New York: McGraw-Hill.

[129] Peressini, A., Sullivan, F., and Uhl, J. (1988) *The Mathematics of Nonlinear Programming*. New York: Springer-Verlag.

[130] Reich, S. (1979) "Weak convergence theorems for nonexpansive mappings in Banach spaces." *Journal of Mathematical Analysis and Applications*, **67**, pp. 274–276.

[131] Reich, S. (1980) "Strong convergence theorems for resolvents of accretive operators in Banach spaces." *Journal of Mathematical Analysis and Applications*, pp. 287–292.

[132] Renegar, J. (2001) *A Mathematical View of Interior-Point Methods in Convex Optimization*. Philadelphia, PA: SIAM (MPS-SIAM Series on Optimization).

[133] Rockafellar, R. (1970) *Convex Analysis*. Princeton, NJ: Princeton University Press.

[134] Rockmore, A., and Macovski, A. (1976) "A maximum likelihood approach to emission image reconstruction from projections." *IEEE Transactions on Nuclear Science*, **NS-23**, pp. 1428–1432.

[135] Schmidlin, P. (1972) "Iterative separation of sections in tomographic scintigrams." *Nucl. Med.* **15(1)**.

[136] Shepp, L., and Vardi, Y. (1982) "Maximum likelihood reconstruction for emission tomography." *IEEE Transactions on Medical Imaging*, **MI-1**, pp. 113–122.

[137] Shermer, M. (2008) "The Doping Dilemma" *Scientific American*, April 2008, pp. 82–89.

[138] 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.

[139] 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.

[140] Shieh, M., and Byrne, C. (2006) "Image reconstruction from limited Fourier data." *Journal of the Optical Society of America, A*, **23(11)**, pp. 2732–2736.

[141] Simmons, G. (1972) *Differential Equations, with Applications and Historical Notes*. New York: McGraw-Hill.

[142] Stiemke, E. (1915) "Über positive Lösungen homogener linearer Gleichungen." *Math. Ann*, **76**, pp. 340–342.

[143] Tanabe, K. (1971) "Projection method for solving a singular system of linear equations and its applications." *Numer. Math.* **17**, pp. 203–214.

[144] Teboulle, M. (1992) "Entropic proximal mappings with applications to nonlinear programming." *Mathematics of Operations Research*, **17(3)**, pp. 670–690.

[145] Tucker, A. (1956) "Dual systems of homogeneous linear relations." in [107], pp. 3–18.

[146] van der Sluis, A. (1969) "Condition numbers and equilibration of matrices." *Numer. Math.*, **14**, pp. 14–23.

[147] van der Sluis, A., and van der Vorst, H.A. (1990) "SIRT- and CG-type methods for the iterative solution of sparse linear least-squares problems." *Linear Algebra and its Applications*, **130**, pp. 257–302.

[148] 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.

[149] Wright, M. (2005) "The interior-point revolution in optimization: history, recent developments, and lasting consequences." *Bulletin (New Series) of the American Mathematical Society*, **42(1)**, pp. 39–56.

[150] Yang, Q. (2004) "The relaxed CQ algorithm solving the split feasibility problem." *Inverse Problems*, **20**, pp. 1261–1266.

Index