Block-Iterative Algorithms

Charles Byrne (Charles_Byrne@uml.edu)
Department of Mathematical Sciences
University of Massachusetts Lowell
Lowell, MA 01854, USA

October 17, 2008

Abstract

The recently presented sequential unconstrained minimization algorithm SUMMA is extended to provide a framework for the derivation of block-iterative, or partial-gradient, optimization methods. This BI-SUMMA includes, and is motivated by, block-iterative versions of the algebraic reconstruction technique (ART) and its multiplicative variant, the MART. The BI-SUMMA approach is designed to provide computationally tractable and quickly convergent algorithms. The rescaled block-iterative expectation maximization maximum likelihood method (RBI-EMML) is closely related to the RBI-MART, but is not a particular case of BI-SUMMA. My papers are available as pdf files at http://faculty.uml.edu/cbyrne/cbyrne.html .

1 Introduction

In recent years, image reconstruction has provided fertile ground for the development of iterative algorithms; this has been particularly true with regard to medical imaging. The *algebraic reconstruction technique* (ART) and its multiplicative version, MART [35], the *expectation maximization maximum likelihood* method (EMML) [48, 43, 52, 42], and the simultaneous and block-iterative MART [47, 28, 22] are just some of the iterative algorithms initially introduced within the medical imaging context. Each of these algorithms can be viewed as providing exact or approximate solutions to systems of linear equations, perhaps with side constraints, such as positivity.

1.1 Medical Image Reconstruction

The reconstruction of images from tomographic data obtained from actual patients poses its own unique challenges, not the least of which is to generate accurate images in a timely fashion. The systems of equations that arise in such areas as transmission and emission tomography are extremely large, the data is noisy, and the mathematical modeling of the scanning process is not as exact as one might wish it to be. The algorithms used must be sufficiently flexible to incorporate the physics of the scanning process, which effectively means that the algorithms must be iterative. These iterative algorithms must either converge rapidly, or, at least, provide useful reconstructions within a few iterations. Not only should few iterations suffice, but each iteration should be relatively inexpensive. Block-iterative algorithms seem to be the methods of choice at the present time.

1.2 The EMML Algorithm for SPECT

In *single photon emission computed tomography* (SPECT) [53], the values y_i are the number of photons detected at the ith detector, for $i = 1, ..., I$. These data are viewed as realizations of independent Poisson random variables, with mean values $(Px)_i = \sum_{j=1}^{J} P_{ij} x_j$, for each i. Here x_j is the unknown expected number of photons emitted, within the scanning time, from the jth pixel in the body, and P_{ij} is the probability that a photon emitted from the jth pixel will be detected at the ith detector. The EMML algorithm has the iterative step

$$x_j^k = x_j^{k-1} s_j^{-1} \sum_{i=1}^{I} P_{ij} \Big(\frac{y_i}{(Px^{k-1})_i} \Big), \tag{1.1}$$

where $s_j = \sum_{i=1}^{I} P_{ij} > 0$. For every positive starting vector x^0, the sequence $\{x^k\}$ converges to a non-negative vector maximizing the likelihood function for the model of independent Poisson counts.

The EMML algorithm is flexible, in that it permits the geometry of the scanner and the patient-specific attenuation to be incorporated in the choice of the P_{ij}, and the Poisson model for the emission conforms with the physics of the situation. However, it is slow to converge, each step of the iteration can be expensive, particularly when I is large, and when the data is noisy, which is the usual case, the image that maximizes likelihood is often not useful (see Appendix B). Stopping the iteration after a few passes, or some other form of regularization, can lead to useful images, but accelerating the algorithm is also important.

1.3 Block-Iterative EMML

The paper of Holte, Schmidlin, *et al.* [38] compares the performance of Schmidlin's method of [47] 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 actual iterative methods employed. Somewhat later, it was noticed that useful images could be obtained quickly if, in the implementation of the EMML algorithm, the summation was performed only over those i in a subset, or block, of the detector indices; then a new block was selected and the process repeated. This *ordered-subset* (OSEM) method [39, 40] quickly became the algorithm of choice, at first, for researchers, and a bit later, for the clinic.

The absence of a solid mathematical foundation for the OSEM led several groups to reexamine other block-iterative methods, particularly BI-MART, the block-iterative version of MART [28, 22]. Unlike the OSEM, the BI-MART always converges to a non-negative solution of the system $y = Px$, whenever there is a non-negative solution, regardless of how the blocks are selected. This suggested that the OSEM is not the correct block-iterative version of the EMML. This problem was resolved with the appearance, in 1996, of RAMLA [9] and the rescaled BI-EMML (RBI-EMML) [11].

Block-iterative methods do not necessarily converge faster than simultaneous ones that use all the equations at each step. The block-iterative methods do provide the opportunity for a rescaling of the equations, which, as we shall see, does lead to significant acceleration of the algorithms.

1.4 Overview

Our main goal in this paper is to provide a framework for the design of block-iterative algorithms. Recently, a *sequential unconstrained minimization algorithm* (SUMMA) [20] was proposed for the derivation of iterative algorithms for constrained optimization. The SUMMA, which is more like a template for algorithms rather than a single algorithm, can also be used to provide computationally tractable iterative methods and to incorporate regularization. In this paper we investigate the expansion of the SUMMA approach to a block-iterative SUMMA (BI-SUMMA) that encompasses block-iterative methods.

We begin with a review of block-iterative versions of ART, MART and the EMML. The convergence proofs of BI-ART and BI-MART will also serve to motivate the BI-SUMMA. We discuss briefly the SUMMA framework, and then proceed to the derivation of the BI-SUMMA.

2 Notation

We let A be an I by J matrix with complex entries, $A^\dagger$ its conjugate transpose, b an arbitrary vector in C^I, $Q = A^\dagger A$, P an I by J matrix with non-negative entries and $s_j = \sum_{i=1}^I P_{ij} > 0$, for $j = 1, ..., J$, and y a vector in R^I with positive entries. For $i = 1, ..., I$, we let a^i denote the ith column of the matrix $A^\dagger$. We denote by $\mathcal{X}$ the subset of R^J consisting of all non-negative vectors x for which Px is a positive vector.

For a positive integer N with $1 \leq N \leq I$, we let $B_1, ..., B_N$ be a partition of the set $\{i = 1, ..., I\}$, with I_n the cardinality of B_n; 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 $L_n = \rho(A_n^\dagger A_n)$ be the spectral radius, or largest eigenvalue, of the matrix $A_n^\dagger A_n$ and we let $L = \rho(A^\dagger A)$.

Similarly, we let P_n be the matrix and y^n the vector obtained from P and y, respectively, by removing all the rows except for those whose index i is in the set B_n. For each n and j, we let

$$s_{nj} = \sum_{i \in B_n} P_{ij},$$

$$m_n = \max\{s_{nj},\ j = 1, ..., J\},$$

and

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

When $N = 1$, $s_{nj} = s_j$, so $\mu = \mu_n = 1$ and

$$m = m_n = \max\{s_j,\ j = 1, ..., J\}.$$

When $N = I$, and $n = i$, $s_{nj} = P_{ij}$, so

$$\mu_i = \mu_n = \max\{P_{ij} s_j^{-1},\ j = 1, ..., J\},$$

and

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

We say that the system $Ax = b$ is consistent if it has solutions x, and $Px = y$ is consistent if it has solutions x whose entries are all non-negative. The norm $||x||$ is the Euclidean norm.

The Kullback-Leibler (KL) or cross-entropy distance [41] between positive numbers α and β is

$$KL(\alpha, \beta) = \alpha \log \frac{\alpha}{\beta} + \beta - \alpha.$$

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

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

With $a_+ = \sum_{j=1}^{J} a_j$, and $b_+ > 0$, we have

$$KL(a, b) = KL(a_+, b_+) + KL(a, \frac{a_+}{b_+} b). \tag{2.1}$$

For each i, let

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

and

$$H_i^+ = \{x \geq 0 | (Px)_i = y_i\}.$$

The orthogonal projection of x onto the hyperplane H_i is

$$R_i(x) = x - \frac{1}{||a^i||^2}((Ax)_i - b_i)a^i.$$

3 The Block-Iterative ART

We begin with BI-ART, the block-iterative version of the algebraic reconstruction technique (ART).

3.1 The BI-ART Iteration

For $k = 1, 2, ...$, $n = k (\mathrm{mod}\, N)$ and the parameters $\gamma_n > 0$ appropriately chosen, the iterative step of the block-iterative ART (BI-ART) is

$$x^k = x^{k-1} - \gamma_n A_n^\dagger (A_n x^{k-1} - b^n). \tag{3.1}$$

3.2 Convergence of BI-ART

For appropriately chosen γ_n, the BI-ART algorithm converges, in the consistent case, for any choice of blocks, and any starting vector x^0.

Theorem 3.1 *Let $0 < \gamma_n \leq L_n^{-1}$. If the system $Ax = b$ is consistent, then the BI-ART sequence $\{x^k\}$ converges to the solution minimizing $||x - x^0||$.*

Proof: Let $A\hat{x} = b$. For each k let

$$G_k(x) = \frac{1}{2}||A_n x - b^n||^2 + \frac{1}{2\gamma_n}||x - x^{k-1}||^2 - \frac{1}{2}||A_n x - A_n x^{k-1}||^2. \tag{3.2}$$

The restriction on γ_n yields the inequality

$$\frac{1}{2\gamma_n}||x - x^{k-1}||^2 - \frac{1}{2}||A_n x - A_n x^{k-1}||^2 \geq 0, \tag{3.3}$$

and so $G_k(x) \geq 0$, for all x. The vector x^k given by Equation (3.1) minimizes $G_k(x)$ and it is easily seen that

$$G_k(x) - G_k(x^k) = \frac{1}{2\gamma_n}||x - x^k||^2. \tag{3.4}$$

Now we can write $G_k(\hat{x})$ two ways. First, from the definition, we have

$$G_k(\hat{x}) = \frac{1}{2\gamma_n}||\hat{x} - x^{k-1}||^2 - \frac{1}{2}||b^n - A_n x^{k-1}||^2. \tag{3.5}$$

Second, from Equation (3.4), we have

$$G_k(\hat{x}) = G_k(x^k) + \frac{1}{2\gamma_n}||\hat{x} - x^k||^2. \tag{3.6}$$

Therefore,

$$||\hat{x} - x^{k-1}||^2 - ||\hat{x} - x^k||^2 = 2\gamma_n G_k(x^k) + \gamma_n||b^n - A_n x^{k-1}||^2, \tag{3.7}$$

from which we draw several conclusions:

- the sequence $\{||\hat{x} - x^k||\}$ is decreasing;
- the sequence $\{G_k(x^k)\}$ converges to zero; and
- the sequence $\{||x^k - x^{k-1}||\}$ 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 $\{||A_n x^{mN+n} - b^n||\}$ converges to zero; and
- the sequence $\{x^{mN+n}\}$ is bounded.

Let $x^{*,1}$ be a cluster point of the sequence $\{x^{mN+1}\}$; then there is subsequence $\{x^{m_rN+1}\}$ converging to $x^{*,1}$. The sequence $\{x^{m_rN+2}\}$ 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 $x^k \to x^*$.

Finally, since the right side of the equation

$$||\hat{x} - x^{k-1}||^2 - ||\hat{x} - x^k||^2 = 2\gamma_n G_k(x^k) + \gamma_n ||b^n - A_n x^{k-1}||^2$$

does not depend on which solution $\hat{x}$ we are using, neither does the left side. Summing over the index k on both sides, we find that

$$||\hat{x} - x^0||^2 - ||\hat{x} - x^*||^2$$

does not depend on which solution $\hat{x}$ we are using. Therefore, minimizing $||\hat{x} - x^0||$ over all solutions $\hat{x}$ is equivalent to minimizing $||\hat{x} - x^*||$ over all solutions $\hat{x}$, for which the answer is clearly $\hat{x} = x^*$. ∎

When the matrix A is normalized so that $||a^i|| = 1$ for each i, then

$$L_n = \rho(A_n^\dagger A_n) = \rho(A_n A_n^\dagger) \leq \text{trace}\,(A_n A_n^\dagger) = I_n.$$

Therefore, the choice of $\gamma_n = 1/I_n$ is acceptable and the resulting BI-ART iterative step becomes

$$x^k = \frac{1}{I_n} \sum_{i \in B_n} R_i(x^{k-1}). \tag{3.8}$$

We turn now to two examples of BI-ART, the ART and Landweber's algorithm.

3.3 The ART

We suppose now that $N = I$ and $B_n = B_i = \{i\}$, for $i = 1, ..., I$. Let $i = k(\text{mod}\, I)$. The iterative step of the ART is

$$x^k = x^{k-1} - \gamma_i((Ax^{k-1})_i - b_i)a^i. \tag{3.9}$$

We know from Theorem 3.1 that, for $0 < \gamma_i \leq \frac{1}{||a^i||^2}$, the ART sequence converges, in the consistent case, to the solution closest to x^0. If we take $\gamma_i = \frac{1}{||a^i||^2}$, then the ART iterative step is

$$x^k = R_i(x^{k-1}). \tag{3.10}$$

3.4 The Landweber Algorithm

We suppose now that $N = 1$, so that $B_1 = \{i = 1, ..., I\}$. The iterative step of Landweber's algorithm is

$$x^k = x^{k-1} - \gamma A^\dagger (Ax^{k-1} - b). \tag{3.11}$$

We know from Theorem 3.1 that, in the consistent case, for $0 < \gamma \leq \frac{1}{L}$, the Landweber sequence converges to the solution closest to x^0.

More can be said about the Landweber algorithm. Using the Krasnoselskii-Mann Theorem (see Appendix A), it can be shown that the Landweber sequence converges to the least-squares solution closest to x^0, for $0 < \gamma < \frac{2}{L}$.

In the inconsistent case, if $N > 1$, the BI-ART will not converge to a least-squares solution, but instead, will exhibit subsequential convergence to a limit cycle consisting of (typically) N distinct vectors.

When the matrix A has been normalized so that $||a^i|| = 1$, for all i, we have $L \leq I$. If we then take the acceptable choice of $\gamma = 1/I$, the Landweber iterative step is that of the Cimmino algorithm [25], and we have

$$x^k = \frac{1}{I} \sum_{i=1}^{I} R_i(x^{k-1}). \tag{3.12}$$

3.5 Why use BI-ART?

For large systems of equations, it may be more efficient to use a block of equations at each step of the iteration, rather than all the equations, or just a single equation. We may also be able to accelerate convergence to a solution using BI-ART, if unfortunate ordering of the blocks is avoided. From the iterative step of BI-ART, we can write

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

$$= 2\gamma_n Re\langle \hat{x} - x^{k-1}, A_n^\dagger(b^n - A_n x^{k-1})\rangle - \gamma_n^2 ||A_n^\dagger(b^n - A_n x^{k-1})||_2^2$$

$$= 2\gamma_n ||b^n - A_n x^{k-1}||_2^2 - \gamma_n^2 ||A_n^\dagger(b^n - A_n x^{k-1})||_2^2.$$

Therefore, we have

$$||\hat{x} - x^{k-1}||_2^2 - ||\hat{x} - x^k||_2^2 \geq (2\gamma_n - \gamma_n^2 L_n)||b^n - A_n x^{k-1}||_2^2. \tag{3.13}$$

From this Inequality (3.13), we see that we make progress toward a solution to the extent that the right side of the inequality,

$$(2\gamma_n - \gamma_n^2 L_n)||b^n - A_n x^{k-1}||_2^2$$

is large. One conclusion we draw from this is that we want to avoid ordering the blocks so that the quantity $||b^n - A_n x^{k-1}||_2^2$ is small. We also want to select γ_n reasonably large, subject to the bound $\gamma_n < 2/L_n$; the maximum of $2\gamma_n - \gamma_n^2 L_n$ is at $\gamma_n = L_n^{-1}$. If we have normalized the matrix A so that the rows of A_n have length one, then the trace of $A_n^\dagger A_n$ is I_n, the number of rows in A_n. Since L_n is not greater than this trace, we have $L_n \leq I_n$, so the choice of $\gamma_n = 1/I_n$ in BI-ART is acceptable, but possibly far from optimal, particularly if A_n is sparse. The choice of $\gamma = 1/I$ in the Landweber algorithm is Cimmino's algorithm for the normalized case.

Inequality (3.13) can be used to give a rough measure of the speed of convergence of BI-ART. The term $||b^n - A_n x^{k-1}||_2^2$ is on the order of I_n, while the term $2\gamma_n - \gamma_n^2 L_n$ has $1/L_n$ for its maximum, so, very roughly, is on the order of $1/I_n$. Consequently, the improvement made in one step of BI-ART is on the order of one. One complete cycle of BI-ART, that is, one complete pass through all the blocks, then corresponds to an improvement on the order of N, the number of blocks. It is a "rule of thumb" that block-iterative methods are capable of improving the speed of convergence by a factor of the number of blocks, if unfortunate ordering of the blocks and selection of the equations within the blocks are avoided, and the parameters are well chosen.

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

3.6 An Upper Bound for the Singular Values of A

When A is not too large, finding $\rho(A^\dagger A)$ poses no significant problem, but, for many of our applications, A is large. Even calculating $A^\dagger A$, not to mention finding its eigenvalues, is expensive in those cases. We would like a good estimate of $\rho(A^\dagger A)$ that can be obtained from A itself. The upper bounds for $\rho(A^\dagger A)$ we present here apply to any matrix A, but will be particularly helpful when A is sparse, that is, most of its entries are zero.

For each i and j, 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 3.2 *([15]) 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 3.1 *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 σ_j is the number of non-zero entries in the jth column of A, and σ is the maximum of the σ_j. ∎

3.7 Using Sparseness

Let each of the rows of the matrix A have length one. Let τ_{nj} be the number of non-zero elements in the jth column of A_n, and let τ_n be the maximum of the τ_{nj}. We know then that $L_n \leq \tau_n$. Therefore, we can choose $\gamma_n < 2/\tau_n$.

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

4 The Block-Iterative MART

We turn now to the block-iterative version of the multiplicative algebraic reconstruction technique (MART). These iterative methods are used to find non-negative solutions to non-negative systems of the form $y = Px$.

4.1 The BI-MART Iteration

For $k = 1, 2, ...$, and $n = k(\mathrm{mod}\, N)$, the iterative step of of the block-iterative MART (BI-MART) is described by

$$\log x_j^k = \log x_j^{k-1} - \gamma_n \delta_j \sum_{i \in B_n} P_{ij} \log \Big(\frac{(Px^{k-1})_i}{y_i} \Big). \tag{4.1}$$

4.2 Convergence of BI-MART

For appropriately chosen γ_n and δ_j, the BI-MART algorithm converges, in the consistent case, for any choice of blocks, and any starting vector x^0.

Theorem 4.1 *Let $0 < s_{nj}\gamma_n\delta_j \leq 1$. If the system $Px = y$ is consistent, then the BI-MART sequence $\{x^k\}$ converges to the non-negative solution in $\mathcal{X}$ minimizing*

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

for any choice of blocks and any $x^0 > 0$.

Proof: Let $P\hat{x} = y$, for some non-negative vector $\hat{x}$. For each k and any x in $\mathcal{X}$, let

$$G_k(x) = KL(P_n x, y^n) + \frac{1}{\gamma_n} \sum_{j=1}^J \delta_j^{-1} KL(x_j, x_j^{k-1}) - KL(P_n x, P_n x^{k-1}). \tag{4.2}$$

Using the Equation (2.1), we see that the restriction on γ_n and δ_j yields the inequality

$$\frac{1}{\gamma_n} \sum_{j=1}^J \delta_j^{-1} KL(x_j, x_j^{k-1}) - KL(P_n x, P_n x^{k-1}) \geq 0 \tag{4.3}$$

and so $G_k(x) \geq 0$, for all non-negative x. The vector x^k given by Equation (4.1) minimizes $G_k(x)$ over all non-negative x and it is easily seen that

$$G_k(x) - G_k(x^k) = \frac{1}{\gamma_n} \sum_{j=1}^J \delta_j^{-1} KL(x_j, x_j^k). \tag{4.4}$$

Now we can write $G_k(\hat{x})$ two ways. First, from the definition, we have

$$G_k(\hat{x}) = \frac{1}{\gamma_n} \sum_{j=1}^J \delta_j^{-1} KL(\hat{x}_j, x_j^{k-1}) - KL(y^n, P_n x^{k-1}). \tag{4.5}$$

Second, from Equation (4.4), we have

$$G_k(\hat{x}) = G_k(x^k) + \frac{1}{\gamma_n} \sum_{j=1}^J \delta_j^{-1} KL(\hat{x}_j, x_j^k). \tag{4.6}$$

Therefore,

$$\sum_{j=1}^{J}\delta_j^{-1}KL(\hat{x}_j,x_j^{k-1})-\sum_{j=1}^{J}\delta_j^{-1}KL(\hat{x}_j,x_j^{k})=\gamma_n\Big[G_k(x^k)+KL(y^n,P_nx^{k-1})\Big], \quad (4.7)$$

from which we draw several conclusions:

- the sequence $\{\sum_{j=1}^J \delta_j^{-1}KL(\hat{x}_j,x_j^k)\}$ is decreasing;
- the sequence $\{G_k(x^k)\}$ converges to zero; and
- the sequence $\{\sum_{j=1}^J \delta_j^{-1}KL(x_j^k,x_j^{k-1})\}$ converges to zero.

In addition, we also learn that, for fixed $n=1,2,...,N$ and $m\to\infty$,

- the sequence $\{KL(y^n,P_nx^{mN+n-1})\}$ converges to zero;
- the sequence $\{KL(P_nx^{mN+n},y^n)\}$ converges to zero; and
- the sequence $\{x^{mN+n}\}$ is bounded.

Let $x^{*,1}$ be a cluster point of the sequence $\{x^{mN+1}\}$; then there is subsequence $\{x^{m_rN+1}\}$ converging to $x^{*,1}$. The sequence $\{x^{m_rN+2}\}$ 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 $Px^*=y$. Replacing the generic solution $\hat{x}$ with the solution x^*, we see that the sequence $\{\sum_{j=1}^J \delta_j^{-1}KL(x_j^*,x_j^k)\}$ is decreasing. But, subsequences of this sequence converge to zero, so the entire sequence converges to zero, and so $x^k\to x^*$.

Finally, since the right side of the equation

$$\sum_{j=1}^{J}\delta_j^{-1}KL(\hat{x}_j,x_j^{k-1})-\sum_{j=1}^{J}\delta_j^{-1}KL(\hat{x}_j,x_j^{k})=\gamma_n\Big[G_k(x^k)+KL(y^n,P_nx^{k-1})\Big]$$

does not depend on which solution $\hat{x}$ we are using, neither does the left side. Summing over the index k on both sides, we find that

$$\sum_{j=1}^{J}\delta_j^{-1}KL(\hat{x}_j,x_j^{0})-\sum_{j=1}^{J}\delta_j^{-1}KL(\hat{x}_j,x_j^{*})$$

does not depend on which solution $\hat{x}$ we are using. Therefore, minimizing the distance $\sum_{j=1}^J \delta_j^{-1}KL(\hat{x}_j,x_j^0)$ over all solutions $\hat{x}$ is equivalent to minimizing $\sum_{j=1}^J \delta_j^{-1}KL(\hat{x}_j,x_j^*)$ over all solutions $\hat{x}$, for which the answer is clearly $\hat{x}=x^*$. ▮

There are two frequently used choices for the parameters γ_n and δ_j. If we choose $\delta_j = 1$, for each j, then we must select γ_n so that

$$0 < \gamma_n \leq m_n^{-1},$$

while, if we select $\delta_j = s_j^{-1}$, then

$$0 < \gamma_n \leq \mu_n^{-1}.$$

The *rescaled* BI-MART (RBI-MART or RBI-SMART) uses the largest permissible value of γ_n in either case.

We turn now to two examples of BI-MART, the MART and the simultaneous MART (SMART)

4.3 The MART

We suppose now that $N = I$ and $B_n = B_i = \{i\}$, for $i = 1, ..., I$. Let $i = k(\mathrm{mod}\, I)$. The iterative step of the MART is

$$x_j^k = x_j^{k-1}\Big(\frac{y_i}{(Px^{k-1})_i}\Big)^{\delta_j \gamma_i P_{ij}}. \tag{4.8}$$

We know from Theorem 4.1 that, for $0 < P_{ij}\delta_j\gamma_i \leq 1$, the MART sequence converges, in the consistent case, to the solution x minimizing $\sum_{j=1}^J \delta_j^{-1} KL(x_j, x_j^0)$. A common choice for the parameters is to select $\delta_j = 1$ and

$$\gamma_i = m_i^{-1}.$$

4.4 The SMART

We suppose now that $N = 1$, so that $B_1 = \{i = 1, ..., I\}$. The iterative step of the SMART is described by

$$\log x_j^k = \log x_j^{k-1} - \gamma\delta_j \sum_{i=1}^I P_{ij} \log\Big(\frac{(Px^{k-1})_i}{y_i}\Big). \tag{4.9}$$

We know from Theorem 4.1 that, in the consistent case, for $0 < s_j\delta_j\gamma \leq 1$, the SMART sequence converges to the non-negative solution x in $\mathcal{X}$ minimizing

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

Common choices for the parameters are $\delta_j = s_j^{-1}$ and $\gamma = 1$. Another choice would be $\delta_j = 1$ and $\gamma = m^{-1}$, where $m = \max\{s_j \,|j = 1, ..., J\}$.

More can be said about the SMART. It can be shown [10] that, in the inconsistent case, the SMART sequence converges to an approximate solution, the unique non-negative minimizer $x \in \mathcal{X}$ of $KL(Px, y)$ that minimizes $\sum_{j=1}^{J} \delta_j^{-1} KL(x_j, x_j^0)$. The *night sky* theorem given there shows that, if P and every matrix obtained from P by removing columns have full rank, then, in the inconsistent case, there is a subset S of the set $\{j = 1, ..., J\}$ with cardinality at most $I - 1$, such that every non-negative minimizer of $KL(Px, y)$ is supported on S (see Appendix B). Consequently, the minimizer is unique. If J is much larger than I, then, in the inconsistent case, the non-negative x minimizing $KL(Px, y)$ will have many zero values. This poses a problem when x is a vectorized image, because these zero values tend to be scattered throughout the image, making it unusable.

In the inconsistent case, if $N > 1$, the BI-MART will not converge to an approximate solution, but instead, always exhibits subsequential convergence to a limit cycle consisting of (typically) N distinct vectors, although no proof of this is known.

4.5 Why use BI-MART?

As with the BI-ART, it may be more efficient to use a block of equations at each step of the iteration, rather than all of them, or just a single one. The SMART can be slow to converge, prompting the search for accelerated versions. The BI-MART does not necessarily converge faster than the SMART algorithm, even with a good choice of the ordering of the blocks. The key to accelerating convergence now is the selection of the parameters δ_j and γ_n.

With the acceptable choice of $\delta_j = s_j^{-1}$ and $\gamma_n = 1$ the BI-MART iteration is described by

$$\log x_j^k = \log x_j^{k-1} - s_j^{-1} \sum_{i \in B_n} P_{ij} \log \Big(\frac{(Px^{k-1})_i}{y_i} \Big). \tag{4.10}$$

so that

$$\log x_j^k = (1 - s_j^{-1} s_{nj}) \log x_j^{k-1} + \Big(s_j^{-1} \sum_{i \in B_n} P_{ij} \log [x_j^{k-1} \frac{y_i}{(Px^{k-1})_i}] \Big). \tag{4.11}$$

From Equation (4.11) we see that the BI-MART involves relaxation, in which $\log x_j^k$ includes some fraction of the current $\log x_j^{k-1}$. This fraction can be unnecessarily large, and the BI-MART algorithm can be accelerated by rescaling.

With the choice of $\delta_j = s_j^{-1}$, the iterative step of the *rescaled* BI-MART (RBI-MART or RBI-SMART) is

$$\log x_j^k = (1 - \mu_n^{-1} s_j^{-1} s_{nj}) \log x_j^{k-1} + \Big(\mu_n^{-1} s_j^{-1} \sum_{i \in B_n} P_{ij} \log [x_j^{k-1} \frac{y_i}{(Px^{k-1})_i}] \Big).$$

$$\tag{4.12}$$

With the choice $\delta_j = 1$, the iterative step of RBI-SMART is

$$\log x_j^k = (1 - m_n^{-1} s_{nj}) \log x_j^{k-1} + \Big(m_n^{-1} \sum_{i \in B_n} P_{ij} \log [x_j^{k-1} \frac{y_i}{(Px^{k-1})_i}] \Big). \tag{4.13}$$

In general, the RBI-SMART uses the parameters γ_n that are as large as possible, subject to the constraints

$$s_{nj} \gamma_n \delta_j \leq 1.$$

Simulation studies have shown that this rescaling can accelerate convergence by roughly a factor of N.

When $N = I$ and each block $B_n = B_i = \{i\}$, the RBI-SMART for the choice $\delta_j = s_j^{-1}$ has the iterative step

$$x_j^k = x_j^{k-1} \exp \Big(\mu_i^{-1} s_j^{-1} P_{ij} \log \frac{y_i}{(Px^{k-1})_i} \Big), \tag{4.14}$$

so that

$$x_j^k = x_j^{k-1} \Big(\frac{y_i}{(Px^{k-1})_i} \Big)^{\mu_i^{-1} s_j^{-1} P_{ij}}. \tag{4.15}$$

For the choice $\delta_J = 1$, the RBI-SMART has the iterative step

$$x_j^k = x_j^{k-1} \exp \Big(m_i^{-1} P_{ij} \log \frac{y_i}{(Px^{k-1})_i} \Big), \tag{4.16}$$

so that

$$x_j^k = x_j^{k-1} \Big(\frac{y_i}{(Px^{k-1})_i} \Big)^{m_i^{-1} P_{ij}}. \tag{4.17}$$

In general, this *rescaled* MART (RMART) algorithm uses the largest values of γ_i consistent with the constraints

$$P_{ij} \delta_j \gamma_i \leq 1.$$

5 The Block-Iterative EMML Algorithm

The *expectation maximization maximum likelihood* (EMML) algorithm we discuss now was first applied to emission tomographic image reconstruction [48]. In that application the entry x_j of the vector x is the unknown mean number of photons emitted from pixel j during the scan time, y_i is the number of photons detected at the ith detector, and P_{ij} is the probability that a photon emitted at j will be detected at i. The quantity s_j is the probability that a photon emitted at j will be detected. It is assumed that the counts y_i are realizations of independent Poisson random variables with means $(Px)_i$. Maximizing the likelihood function with respect to the unknown parameters $x_j \geq 0$ is equivalent to finding a non-negative minimizer of the function $KL(y, Px)$. As with the SMART, the EMML algorithm is usually slow to converge.

5.1 The EMML Algorithm

The *expectation maximization* (EM) method, as it applies to this problem, is called the EMML algorithm, or sometimes the MLEM algorithm. It has the iterative step

$$x_j^k = x_j^{k-1} s_j^{-1} \sum_{i=1}^{I} P_{ij} \Big(\frac{y_i}{(Px^{k-1})_i} \Big). \tag{5.1}$$

It is interesting to compare this iteration with that of SMART:

$$x_j^k = x_j^{k-1} \exp \Big[s_j^{-1} \sum_{i=1}^{I} P_{ij} \log \Big(\frac{y_i}{(Px^{k-1})_i} \Big) \Big]. \tag{5.2}$$

We have the following result concerning the EMML algorithm.

Theorem 5.1 *The EMML sequence $\{x^k\}$ converges to a non-negative minimizer of $KL(y, Px)$, for any choice of $x^0 > 0$.*

It is an open question to which minimizer the EMML sequence converges. In the consistent case, the limit is a non-negative solution of $y = Px$. If there are multiple non-negative solutions of $y = Px$, the limit will depend on $x^0 > 0$, but we do not know how it depends on x^0.

It was noticed that convergence could sometimes be significantly accelerated by summing over only some of the equations at a time [38, 39]. This *ordered-subset* approach (OSEM) has the iterative step

$$x_j^k = x_j^{k-1} s_{nj}^{-1} \sum_{i \in B_n} P_{ij} \Big(\frac{y_i}{(Px^{k-1})_i} \Big). \tag{5.3}$$

However, the OSEM appears to be inadequate, in certain respects.

Convergence of the OSEM, in the consistent case, was proven only when the blocks (or ‘subsets’) exhibit *subset balance*; that is, the quantities s_{nj} are independent of the index n. In addition, the OSEM can fail to converge, in the consistent case, if subset balance is missing. Also, for the case of singleton blocks, the OSEM simply gives a sequence of vectors all parallel to the original x^0. Clearly, if there is a block-iterative variant of EMML, the OSEM is not it.

5.2 KL Projection onto Hyperplanes

As we have seen, the iterative step of the unrelaxed ART is to take as the next x^k the orthogonal projection of the current x^{k-1} onto the hyperplane determined by the current ith equation. The Landweber algorithm, for normalized A and the choice $\gamma = 1/I$, is Cimmino's algorithm, and x^k is the arithmetic mean of the orthogonal projections of x^{k-1} onto the hyperplanes determined by each of the equations. Each step of the BI-ART involves the arithmetic means of some of these projections. The key to formulating the proper block-iterative variants of EMML is to consider generalized projections onto hyperplanes, involving the KL distance, and to mimic the BI-ART situation.

The KL projection of a given $z \geq 0$ onto H_i^+ is the vector x in H_i^+ that minimizes $KL(x, z)$, over all x in H_i^+. We cannot generally compute this projection in closed form. However, suppose we want the vector in H_i^+ that minimizes the weighted KL distance

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

over all x in H_i^+; we denote this weighted projection of z by $Q_i(z)$. Then the Lagrangian is

$$L(x) = \sum_{j=1}^{J} P_{ij} KL(x_j, z_j) + \lambda(\sum_{j=1}^{J} P_{ij} x_j - y_i).$$

Then setting the gradient of $L(x)$ to zero, we have

$$0 = P_{ij} \log\Big(\frac{x_j}{z_j}\Big) + \lambda P_{ij}.$$

Then, for those j such that $P_{ij} \neq 0$, we have $x_j = \alpha z_j$, for some constant $\alpha > 0$. Since $(Px)_i = y_i$, it follows that $\alpha = y_i/(Pz)_i$, and the weighted projection of z onto H_i^+ is

$$x_j = Q_i(z)_j = z_j\Big(\frac{y_i}{(Pz)_i}\Big).$$

Consequently, once we have x^{k-1}, the weighted projection onto H_i^+ is

$$Q_i(x^{k-1})_j = x_j^{k-1}\Big(\frac{y_i}{(Px^{k-1})_i}\Big).$$

This gives us some insight into what is going on with the SMART, BI-MART and the EMML and suggests how we might mimic BI-MART to get BI-EMML.

5.3 Geometric and Arithmetic Averages of Projections

We can describe the SMART iterative step as

$$\log x_j^k = \sum_{i=1}^{I} s_j^{-1} P_{ij} \log \Big(Q_i(x^{k-1})_j \Big);$$

that is, x_j^k is a weighted geometric mean of all the weighted KL projections of x^{k-1}. Similarly, we can write the EMML iterative step as

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

which shows that x_j^k is a weighted arithmetic mean of the same weighted KL projections.

We can describe the MART iteration as

$$\log x_j^k = (1 - \gamma_i \delta_j P_{ij}) \log x_j^{k-1} + \gamma_i \delta_j P_{ij} \log Q_i(x^{k-1})_j,$$

and the BI-MART as

$$\log x_j^k = (1 - \gamma_n \delta_j s_{nj}) \log x_j^{k-1} + \gamma_n \delta_j \sum_{i \in B_n} P_{ij} \log Q_i(x^{k-1})_j.$$

So we see that, in both MART and the more general BI-MART, we have a weighted geometric mean of some of the KL projections, along with the previous x^{k-1}. Now we can see how to extend the EMML to block-iterative versions: we replace the weighted geometric means with weighted arithmetic means.

5.4 The Block-Iterative EMML

The block-iterative EMML (BI-EMML) has the iterative step

$$x_j^k = (1 - \gamma_n \delta_j s_{nj}) x_j^{k-1} + \gamma_n \delta_j \sum_{i \in B_n} P_{ij} Q_i(x^{k-1})_j, \tag{5.4}$$

with $\gamma > 0$ chosen so that

$$s_{nj} \delta_j \gamma_n \leq 1.$$

The *rescaled* BI-EMML (RBI-EMML) uses the largest values of γ_n consistent with these constraints.

The analogue of the MART is the EMART, with the iterative step

$$x_j^k = (1 - \gamma_i \delta_j P_{ij}) x_j^{k-1} + \gamma_i \delta_j P_{ij} Q_i(x^{k-1})_j, \tag{5.5}$$

with $P_{ij} \delta_j \gamma_i \leq 1$. We have the following result concerning the BI-EMML.

Theorem 5.2 *When the system $y = Px$ is consistent, the BI-EMML sequence $\{x^k\}$ converges to a non-negative solution of $y = Px$, for any choice of blocks and any $x^0 > 0$.*

The inequality in the following lemma is the basis for the convergence proof.

Lemma 5.1 *Let $y = Px$ for some nonnegative x. Then for $\{x^k\}$ as in Equation (5.4) we have*

$$\sum_{j=1}^{J} \delta_j^{-1} KL(x_j, x_j^{k-1}) - \sum_{j=1}^{J} \delta_j^{-1} KL(x_j, x_j^k) \geq \tag{5.6}$$

$$\gamma_n \sum_{i \in B_n} KL(y_i, (Px^k)_i). \tag{5.7}$$

Proof: From the iterative step

$$x_j^k = x_j^{k-1}(1 - \delta_j \gamma_n \sigma_{nj}) + x_j^k \delta_j \gamma_n \sum_{i \in B_n} P_{ij} \frac{y_i}{(Px^k)_i} \tag{5.8}$$

we have

$$\log(x_j^k / x_j^{k-1}) = \log\Big((1 - \delta_j \gamma_n \sigma_{nj}) + \delta_j \gamma_n \sum_{i \in B_n} P_{ij} \frac{y_i}{(Px^k)_i}\Big). \tag{5.9}$$

By the concavity of the logarithm we obtain the inequality

$$\log(x_j^k / x_j^{k-1}) \geq \Big((1 - \delta_j \gamma_n \sigma_{nj}) \log 1 + \delta_j \gamma_n \sum_{i \in B_n} P_{ij} \log \frac{y_i}{(Px^k)_i}\Big), \tag{5.10}$$

or

$$\log(x_j^k / x_j^{k-1}) \geq \delta_j \gamma_n \sum_{i \in B_n} P_{ij} \log \frac{y_i}{(Px^k)_i}. \tag{5.11}$$

Therefore

$$\sum_{j=1}^{J} \delta_j^{-1} x_j \log(x_j^{k+1} / x_j^k) \geq \gamma_n \sum_{i \in B_n} (\sum_{j=1}^{J} x_j P_{ij}) \log \frac{y_i}{(Px^k)_i}. \tag{5.12}$$

Also

$$\sum_{j=1}^{J} \delta_j^{-1} (x_j^k - x_j^{k-1}) = \gamma_n \sum_{i \in B_n} ((Px^k)_i - y_i). \tag{5.13}$$

This concludes the proof of the lemma. ∎

From the inequality in (5.7) we can conclude several things:

- the sequence $\{\sum_{j=1}^{J}\delta_j^{-1}KL(x_j, x_j^k)\}$ is decreasing;
- the sequence $\{x^k\}$ is therefore bounded; and
- the sequence $\{\sum_{i\in B_n}KL(y_i,(Px^{mN+n-1})_i)\}$ is converging to zero.

Let x^* be any cluster point of the sequence $\{x^\}$. Then it is not difficult to show that $y = Px^*$. Replacing x with x^* we have that the sequence $\{\sum_{j=1}^{J}\delta_j^{-1}KL(x_j^*, x_j^k)\}$ is decreasing; since a subsequence converges to zero, so does the whole sequence. Therefore x^* is the limit of the sequence $\{x^k\}$. This proves that the algorithm produces a nonnegative solution of $y = Px$. We have been unable, so far, to replace the inequality in (5.7) with an equation in which the right side is independent of the particular solution x chosen. Therefore, in contrast with the BI-MART, we do not know which solution the BI-EMML gives us, how the solution depends on the starting vector x^0, nor how the solution may depend on the choice of blocks.

The behavior of BI-EMML illustrates once again that using block-iterative methods does not, by itself, lead to faster convergence. It seems that the main advantage of the use of these block-iterative methods is the opportunity to select the parameters. As with BI-MART, the key to accelerating the convergence of BI-EMML is the proper choice of the parameters γ_n and δ_j. Recall that we must have

$$\gamma_n\delta_j s_{nj} \leq 1,$$

for all n and j. When we select $\delta_j = s_j^{-1}$, we must then have $\gamma_n \leq \mu_n^{-1}$. When we have $\delta_j = 1$, we need $\gamma_n \leq m_n^{-1}$. Generally speaking, the larger the γ_n the faster the convergence. The *rescaled* BI-EMML (RBI-EMML) uses the largest acceptable value of the γ_n.

5.5 The RAMLA

We must mention a method that closely resembles the EMART, the *row-action maximum likelihood algorithm* (RAMLA), which was discovered independently by Browne and De Pierro [9]. The RAMLA avoids the limit cycle in the inconsistent case by using strong underrelaxation involving a decreasing sequence of relaxation parameters λ_k. The RAMLA is the following:

Algorithm 5.1 (RAMLA) *Let x^0 be an arbitrary positive vector, and $n = k(\mathrm{mod}\, N)$. Let the positive relaxation parameters λ_k converge to zero, with $\sum_{k=0}^{+\infty}\lambda_k = +\infty$. Then,*

$$x_j^k = (1-\lambda_k s_{nj})x_j^{k-1} + \lambda_k x_j^{k-1}\sum_{i\in B_n}P_{ij}\Big(\frac{y_i}{(Px^{k-1})_i}\Big). \tag{5.14}$$

5.6 Generalized Subset Balance

We say that *generalized subset balance* (GSB) holds if, for each n and j, we have

$$s_{nj} = c_n t_j,$$

for some constants c_n and t_j; if $c_n = c$, for all n, then *subset balance* (SB) holds. In [39, 40] convergence of the OSEM to a non-negative solution of $y = Px$ was established, provided that such solutions exist and SB holds.

As we noted previously, when applied to tomographic problems, the OSEM usually provides useful reconstructed images quickly. This is not because the OSEM uses blocks, but because the OSEM is a particular case of the RBI-EMML when GSB holds. To see this, notice that, when GSB holds, we have $s_{nj} = \mu_n s_j$. With the choice of $\delta_j = s_j^{-1}$, and $\gamma_n = \mu_n^{-1}$, we have

$$1 - \gamma_n \delta_j s_{nj} = 0,$$

so that the right side of Equation (5.4) has only a single term and it is the same as the right side of Equation (5.3). Notice also that if we choose $\delta_j = 1$ instead, we do not get OSEM, but a relaxed version of OSEM.

6 Sequential Unconstrained Minimization

The *sequential unconstrained minimization algorithm* (SUMMA) presented in [20] is really a framework for the design of iterative algorithms, rather than a particular algorithm. It can be used to derive iterative algorithms that perform constrained minimization, as well as computationally tractable unconstrained optimization methods. The SUMMA contains, as particular cases, methods for constrained optimization, such as the well known barrier- and penalty-function methods, and proximal minimization techniques, and the Landweber algorithm and the SMART.

In this section we review the SUMMA and extend it to include regularization methods for the Landweber algorithm and the SMART. In the following section, we generalize the SUMMA to obtain block-iterative algorithms, including the BI-ART and BI-MART.

6.1 The SUMMA

The objective is to minimize the function $f(x) : R^J \to R$, possibly subject to the constraint that x lie within the closed convex set C. We shall assume that the problem has solutions and denote an arbitrary solution by $\hat{x}$. For $k = 1, 2, ...$, we minimize the function

$$G_k(x) = f(x) + g_k(x) \tag{6.1}$$

to get the vector x^k. The auxiliary functions $g_k(x)$ are assumed to satisfy the inequalities

$$0 \leq g_{k+1}(x) \leq G_k(x) - G_k(x^k). \tag{6.2}$$

It follows that

$$0 = g_k(x^{k-1}) \leq g_k(x).$$

In [20] this iterative framework was applied to the constrained problem, where it was called the SUMMA, for sequential unconstrained minimization.

In the constrained case, we assume that the functions $g_k(x)$ are defined and finite on the open, convex set D, that C is the closure of D, and that each x^k lies in D. Being able to solve for the x^k at each step is an important issue, and we shall address it later in this paper.

The basic result concerning SUMMA is convergence in function value; specifically, we have the following theorem [20].

Theorem 6.1 *The sequence $\{f(x^k)\}$ converges to $f(\hat{x})$.*

We consider now several examples of SUMMA.

6.2 Examples of SUMMA

The well known barrier- and penalty-function methods for constrained optimization [32] are particular cases of SUMMA, as are proximity-function methods of Teboulle [51] and Censor and Zenios [23], the Landweber algorithm, and the SMART.

6.2.1 Barrier-Function Methods

The objective is to minimize the function $f(x) : R^J \to R$ over x in C, the closure of the open set D. We choose a barrier function $b(x) \geq 0$ that is finite on D and (typically) approaches $+\infty$ at the boundary of D. At each step, we minimize the function

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

to get x^k, which we assume lies within D [32]. Equivalently, we can minimize the function

$$kf(x) + b(x).$$

To put the barrier-function method within the SUMMA framework, we define

$$G_k(x) = f(x) + (k-1)f(x) + b(x) - (k-1)f(x^{k-1}) - b(x^{k-1}); \tag{6.3}$$

and

$$g_k(x) = (k-1)f(x) + b(x) - (k-1)f(x^{k-1}) - b(x^{k-1}). \tag{6.4}$$

Clearly, the vector x^k minimizes $G_k(x)$.

6.2.2 Penalty-Function Methods

Again, the objective is to minimize the function $f(x) : R^J \to R$ over x in C. We select a penalty function $p(x) \geq 0$ having the property that $p(x) = 0$ if and only if $x \in C$. At each step of the algorithm we minimize the function

$$f(x) + kp(x)$$

to get x^k. Equivalently, we can get x^k by minimizing the function

$$p(x) + \frac{1}{k}f(x);$$

this problem has the form of a barrier-function method, so can be included within the SUMMA framework.

6.2.3 Proximal Minimization

One example of the SUMMA is the *proximal minimization algorithm* (PMA), in which, at each step, we minimize the function

$$G_k(x) = f(x) + D_h(x, x^{k-1}) \tag{6.5}$$

to get x^k [23, 14]. The function

$$D_h(x, z) = h(x) - h(z) - \langle \nabla h(z), x - z \rangle$$

is the Bregman distance from x to z; we assume that h is finite, convex and differentiable on the set D and that $f(x)$ is convex. It is easy to see that

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

The equation to be solved for x^k is then

$$0 = \nabla f(x^k) + \nabla h(x^k) - \nabla h(x^{k-1}). \tag{6.6}$$

Notice that the SUMMA approach does not always guarantee that it will be a simple matter to calculate each x^k. Two other examples will illustrate this point.

6.2.4 The Landweber Algorithm as SUMMA

The $G_k(x)$ we use for the Landweber algorithm appeared in the convergence proof of BI-ART. It is

$$G_k(x) = \frac{1}{2}||Ax - b||^2 + \frac{1}{2\gamma}||x - x^{k-1}||^2 - \frac{1}{2}||Ax - Ax^{k-1}||^2. \tag{6.7}$$

Although this choice of $G_k(x)$ does provide an x^k that is easy to calculate, the choice does seem quite ad hoc. Let's consider one more example, before we attempt to make this choice of $G_k(x)$ more plausible.

6.2.5 The SMART as SUMMA

The $G_k(x)$ we need for SMART has also already appeared, in the proof of BI-MART. It is

$$G_k(x) = KL(Px, y) + \frac{1}{\gamma}\sum_{j=1}^{J} \delta_j^{-1} KL(x_j, x_j^{k-1}) - KL(Px, Px^{k-1}). \tag{6.8}$$

Once again, this choice is a fortunate one, in that it makes x^k easy to calculate, but certainly seems ad hoc. To make these two choices plausible, we return to the proximal minimization approach.

6.2.6 A Modified PMA

Suppose that, instead of using the $G_k(x)$ given by Equation (6.5), we use

$$G_k(x) = f(x) + D_h(x, x^{k-1}) - D_f(x, x^{k-1}), \tag{6.9}$$

with the assumption that h is chosen so that

$$D_h(x, z) - D_f(x, z) \geq 0,$$

for all appropriate x and z. The equation to be solved for x^k is now

$$0 = \nabla h(x^k) - \nabla h(x^{k-1}) + \nabla f(x^{k-1}). \tag{6.10}$$

What may potentially make this equation easier to solve than Equation (6.6) is that we choose the function h.

In the Landweber case, the function h is

$$h(x) = \frac{1}{2\gamma}||x||^2,$$

while in the SMART case the function $h(x)$ is

$$h(x) = \frac{1}{\gamma}\sum_{j=1}^{J} x_j(\log x_j) - x_j.$$

In both cases, we were able to solve Equation (6.10) and get x^k in closed form.

We can use the modified PMA approach to impose upper and lower bounds on solutions to $y = Px$.

6.2.7 Incorporating Upper and Lower Bounds

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

$$h(x) = \sum_{j=1}^{J}\Big((x_j - a_j)\log(x_j - a_j) + (b_j - x_j)\log(b_j - x_j)\Big). \tag{6.11}$$

Then we have

$$D_h(x,z) = \sum_{j=1}^{J}\Big(KL(x_j - a_j, z_j - a_j) + KL(b_j - x_j, b_j - z_j)\Big), \tag{6.12}$$

and, as before,

$$D_f(x,z) = KL(Px, Pz). \tag{6.13}$$

Lemma 6.1 *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

$$g'(c) = \frac{a-c}{b-c} - 1 - \log(\frac{a-c}{b-c}) \geq 0. \tag{6.14}$$

We see then that the function $g(c)$ is increasing with c. ∎

As a corollary of Lemma 6.1, we have

Lemma 6.2 *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 6.3 *$D_h(x, z) \geq D_f(x, z)$.*

Proof: We can easily show that

$$D_h(x, z) \geq KL(Px - Pa, Pz - Pa) + KL(Pb - Px, Pb - Pz),$$

along the lines used previously. Then, from Lemma 6.2, we have $KL(Px - Pa, Pz - Pa) \geq KL(Px, Pz) = D_f(x, z)$. ∎

At the kth step of this algorithm we minimize the function

$$f(x) + D_h(x, x^{k-1}) - D_f(x, x^{k-1}) \tag{6.15}$$

to get x^k.

Solving for x_j^k, we obtain

$$x_j^{k+1} = \alpha_j^k a_j + (1 - \alpha_j^k) b_j, \tag{6.16}$$

where

$$(\alpha_j^k)^{-1} = 1 + \Big(\frac{x_j^{k-1} - a_j}{b_j - x_j^{k-1}}\Big) \exp\Big(\sum_{i=1}^{I} P_{ij} \log(y_i/(Px^{k-1})_i)\Big). \tag{6.17}$$

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 [13]. In [46] we used the modified PMA to obtain an iterative image reconstruction algorithm from fan-beam transmission tomographic data. That algorithm included upper and lower bounds, as well as regularization.

Now we consider how the SUMMA framework may be used to regularize the Landweber algorithm and the SMART.

6.3 Regularization

The Landweber algorithm minimizes the function

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

and converges to the least-squares solution closest to x^0. When A is ill-conditioned and b noisy, the norm of any least-squares solution may be unacceptably large. In such cases, we may choose to minimize

$$\frac{1}{2}||Ax - b||^2 + \frac{\epsilon}{2}||x||^2. \tag{6.18}$$

The solution to this problem satisfies the equation

$$(A^\dagger A + \epsilon I)x = A^\dagger b.$$

We would like to have an iterative algorithm that converges to this solution, but does not employ the matrix $(A^\dagger A + \epsilon I)$.

Similarly, for the choice of $\delta_j = 1$, the SMART converges to the non-negative minimizer of the function $f(x) = KL(Px, y)$ for which $KL(x, x^0)$ is minimized. When $y = Px$ has no non-negative solution, these minimizers may have several unwanted zero entries (see Appendix B). We can regularize the problem by minimizing the function

$$KL(Px, y) + \epsilon KL(x, p), \tag{6.19}$$

where p is a positive vector chosen as a prior estimate of the desired solution. As in the Landweber case, we want a tractable iterative algorithm that solves this minimization problem.

6.3.1 A Regularized Landweber Algorithm

We use Equation (6.9), with $f(x)$ as given by Equation (6.18). Once again, we use

$$h(x) = \frac{1}{2\gamma}||x||^2.$$

At each step, we minimize the function

$$G_k(x) = \frac{1}{2}||Ax - b||^2 + \frac{\epsilon}{2}||x||^2 + \frac{1}{2\gamma}||x - x^{k-1}||^2 - \frac{1}{2}||Ax - Ax^{k-1}||^2 - \frac{\epsilon}{2}||x - x^{k-1}||^2. \tag{6.20}$$

The equation to be solved is then

$$0 = A^\dagger(Ax^{k-1} - b) + \epsilon x^k + (\frac{1}{\gamma} - \epsilon)(x^k - x^{k-1}),$$

and we obtain

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

Therefore, we have

$$x^k = (1 - \gamma\epsilon)x^{k-1} + \gamma A^\dagger(b - Ax^{k-1}). \tag{6.21}$$

Notice that our problem is equivalent to minimizing the function

$$F(x) = ||Bx - c||_2^2, \tag{6.22}$$

for

$$B = \begin{bmatrix} A \\ \sqrt{\epsilon} I \end{bmatrix}, \tag{6.23}$$

and

$$c = \begin{bmatrix} b \\ 0 \end{bmatrix}, \tag{6.24}$$

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

$$x^{k+1} = x^k + \gamma B^T(c - Bx^k), \tag{6.25}$$

for $0 < \gamma < 2/\rho(B^TB)$, where $\rho(B^TB)$ is the spectral radius of B^TB.

6.3.2 A Regularized SMART

We use Equation (6.9), with $f(x)$ as given by Equation (6.19) and $\delta_j = 1$. Now we use

$$h(x) = \sum_{j=1}^{J} x_j(\log x_j) - x_j,$$

or, equivalently,

$$h(x) = KL(x, 1),$$

where 1 denotes the vector with all its entries equal to one.

At each step, we minimize the function

$$G_k(x) = KL(Px, y) + \epsilon KL(x, p) + \frac{1}{\gamma}KL(x, x^{k-1}) - KL(Px, Px^{k-1}) - \epsilon KL(x, x^{k-1}). \tag{6.26}$$

The equation to be solved is then

$$0 = \sum_{i=1}^{I} P_{ij} \log\Big(\frac{(Px^{k-1})_i}{y_i}\Big) + \frac{1}{\gamma}\log x_j^k - (\frac{1}{\gamma} - \epsilon)\log x_j^{k-1} - \epsilon \log p_j$$

and we obtain

$$\frac{1}{\gamma}\log x^k = (\frac{1}{\gamma} - \epsilon)\log x^{k-1} + \sum_{i=1}^{I} P_{ij}\log\Big(\frac{y_i}{(Px^{k-1})_i}\Big) + \epsilon \log p_j.$$

Therefore, we have

$$\log x^k = (1 - \gamma\epsilon)\log x^{k-1} + \gamma\sum_{i=1}^{I} P_{ij}\log\Big(\frac{y_i}{(Px^{k-1})_i}\Big) + \gamma\epsilon\log p_j. \tag{6.27}$$

Since the multiplier $\frac{1}{\gamma}$ is now effectively replaced by

$$\frac{1}{\gamma} - \epsilon = \frac{1}{\alpha},$$

we need $s_j\alpha \leq 1$, so that

$$\gamma \leq \frac{1}{s_j + \epsilon},$$

for all j.

With the choice

$$\gamma = \frac{1}{s_j + \epsilon},$$

we have

$$\log x_j^k = \Big(\frac{s_j}{s_j + \epsilon}\Big)\Big[\log x_j^{k-1} + s_j^{-1}\sum_{i=1}^{I} P_{ij} \log\Big(\frac{y_i}{(Px^{k-1})_i}\Big)\Big] + \Big(\frac{\epsilon}{s_j + \epsilon}\Big)\log p_j,$$

and the new x_j^k is a weighted geometric mean of the unregularized SMART iterate and the p_j.

In the next section, we consider a block-iterative version of SUMMA and use it to rederive BI-ART and BI-MART.

7 Block-Iterative SUMMA

We assume now that the function to be minimized has the form

$$f(x) = \sum_{i=1}^{I} f_i(x),$$

where each $f_i(x)$ is non-negative. We also assume that there is $\hat{x}$ in C with $f(\hat{x}) = 0$; therefore, $f_i(\hat{x}) = 0$, for each i. Note that this assumption ensures that each of the functions $f_i(x)$ has a common minimizer. When this is not the case, it is highly likely that the block-iterative SUMMA will exhibit subsequential convergence to a limit cycle, which is what we always see with the MART and can prove for the ART.

For $n = 1, ..., N$ we define

$$f^n(x) = \sum_{i \in B_n} f_i(x).$$

We return to the modified PMA method and develop a block-iterative version.

7.1 BI-SUMMA

For each $k = 1, 2, \ldots$ and $n = k (\bmod N)$, we minimize the function

$$G_k(x) = f^n(x) + \frac{1}{\gamma_n} D_h(x, x^{k-1}) - D_{f^n}(x, x^{k-1}) \tag{7.1}$$

to get x^k. We shall assume that γ_n has been chosen so that

$$D_h(x, z) \geq \gamma_n D_{f^n}(x, z),$$

for all appropriate x and z. Then

$$0 = \frac{1}{\gamma_n} \nabla h(x^k) - \frac{1}{\gamma_n} \nabla h(x^{k-1}) + \nabla f^n(x^{k-1}),$$

or

$$\nabla h(x^k) = \nabla h(x^{k-1}) - \gamma_n \nabla f^n(x^{k-1}). \tag{7.2}$$

From the appearance of the gradient $\nabla f^n(x^{k-1})$, we see that this iterative method is a *partial gradient* or *incremental gradient* approach [7].

Using Equation (7.2) we can show that

$$G_k(x) - G_k(x^k) = \frac{1}{\gamma_n} D_h(x, x^k). \tag{7.3}$$

From the definition, we have

$$G_k(\hat{x}) = \frac{1}{\gamma_n} D_h(\hat{x}, x^{k-1}) - D_{f^n}(\hat{x}, x^{k-1}),$$

and from Equation (7.3) we have

$$G_k(\hat{x}) = G_k(x^k) + \frac{1}{\gamma_n} D_h(\hat{x}, x^k).$$

Therefore,

$$D_h(\hat{x}, x^{k-1}) - D_h(\hat{x}, x^k) = \gamma_n G_k(x^k) + \gamma_n D_{f^n}(\hat{x}, x^{k-1}). \tag{7.4}$$

From this equation we can conclude several things, provided that we make two assumptions about the Bregman distance $D_h(x, z)$.

We assume, first, that for each fixed x in the domain of h, the function $F(z) = D_h(x, z)$ has bounded level sets, and second, that if the sequence $\{D_h(x, x^k)\}$ converges to zero, then $\{x^k\}$ converges to x. Now we can draw our conclusions:

- the sequence $\{D_h(\hat{x}, x^k)\}$ is decreasing;
- the sequence $\{x^k\}$ is bounded;
- the sequences $\{D_{f^n}(\hat{x}, x^{mN+n-1})\}$ converge to zero.

Mimicking earlier proofs, we can show that the sequence $\{x^k\}$ has limit x^*, with $f(x^*) = 0$.

The BI-ART is a particular case of BI-SUMMA, and the $G_k(x)$ is

$$G_k(x) = \frac{1}{2}||A_n x - b^n||^2 + \frac{1}{2\gamma_n}||x - x^{k-1}||^2 - \frac{1}{2}||A_n x - A_n x^{k-1}||^2.$$

The BI-MART is also a particular case of BI-SUMMA, and the $G_k(x)$ is

$$G_k(x) = KL(P_n x, y^n) + \frac{1}{\gamma_n}\sum_{j=1}^{J} \delta_j^{-1} KL(x_j, x_j^{k-1}) - KL(P_n x, P_n x^{k-1}).$$

7.2 Acceleration

As we mentioned previously, block-iterative algorithms do not always converge, but may, in certain cases, exhibit subsequential convergence to a limit cycle consisting of (usually) N distinct vectors. This happens in BI-ART when there is no solution of $Ax = b$, in BI-MART and BI-EMML when there is no non-negative solution of $y = Px$, and in BI-SUMMA when there is no $\hat{x}$ with $f(\hat{x}) = 0$. Except for BI-ART, no proofs of this subsequential convergence have been given for the block-iterative algorithms discussed here. Nevertheless, block-iterative algorithms, including the OSEM, have repeatedly been observed to produce useful approximate solutions much faster than their simultaneous counterparts. We stress that this acceleration is not due merely to the use of blocks. Block-iterative methods provide an opportunity to select parameters and the choice of these parameters greatly affects the rate of convergence.

7.2.1 The BI-ART

We see from Equation (3.13) that, generally speaking, the distance $||\hat{x}-x^k||_2^2$ decreases faster if the parameter γ_n is large, subject to the restriction that $0 < \gamma_n \leq 2/L_n$; the choice of $\gamma_n = L_n^{-1}$ would seem to be a good one. In the case of the ART, we have $L_i = L_n = ||a^i||_2^2$; the ART typically uses the value $\gamma_i = 1/||a^i||^2 = L_i^{-1}$. It has been shown, however, that a certain amount of relaxation can be beneficial; that is, taking a smaller value of γ_i can lead to faster convergence [49]. The ordering of the equations can also have a significant effect on the rate of convergence [37], as Equation (3.13) suggests.

With the choice of $\gamma_n = 1/L_n$, the right side of Equation (3.13) is, very loosely speaking, independent of the value of N; this is the key to understanding the acceleration. One full pass through all the blocks then reduces $||\hat{x} - x^k||^2$ about N times faster than the simultaneous version.

7.2.2 The BI-MART and BI-EMML

Equation (4.7) tells us that, generally speaking, the distance

$$\sum_{j=1}^{J} \delta_j^{-1} KL(\hat{x}_j, x_j^k)$$

will decrease faster if the parameter γ_n is large. As with the BI-ART, the ordering of the equations is also a factor.

With the choices $\delta_j = 1$ and $\gamma_n = m_n^{-1}$, the right side of Equation (4.7) is, roughly speaking, independent of the value of N. As with the BI-ART, one complete pass through all the blocks will reduce $KL(\hat{x}, x^k)$ about N times faster than with SMART.

Equation (5.7) shows that much the same story holds for the BI-EMML algorithm.

7.2.3 The BI-SUMMA

Loosely speaking, Equation (7.4) shows that the distance $D_h(\hat{x}, x^k)$ will decrease more rapidly if the parameter γ_n is large, subject to the restriction that

$$D_h(x, z) \geq \gamma_n D_{f^n}(x, z).$$

One possible choice is to select $D_h(x, z)$ so that

$$D_h(x, z) \geq \gamma \sum_{n=1}^{N} D_{f^n}(x, z),$$

and then take $\gamma_n = \gamma$.

For example, in the BI-MART case, with $\delta_j = 1$ and

$$m = \max\{s_j \,|j = 1, ..., J\},$$

we have

$$KL(x, z) \geq m^{-1} KL(Px, Pz) = m^{-1} \sum_{n=1}^{N} KL(P_n x, P_n z) \geq m^{-1} KL(P_n x, P_n z).$$

Therefore, the choice of $\gamma_n = m^{-1}$ is acceptable. However, convergence requires only that $\gamma_n s_{nj} \leq 1$; the choice of $\gamma_n = m^{-1}$ is roughly N times too small.

8 Projecting onto Convex Sets

As we have seen, projection onto hyperplanes plays an important role in most of the iterative algorithms discussed so far. The BI-ART involves weighted arithmetic means of orthogonal projections onto the hyperplanes H_i, while the BI-MART and BI-EMML employ weighted geometric and arithmetic means of generalized Kullback-Leibler projections onto H_i^+. An obvious extension of these ideas is to consider iterative algorithms based on projection onto closed convex sets.

8.1 The Convex Feasibility Problem

Let C_i, $i = 1, ..., I$, be closed non-empty convex sets in R^J. The *convex feasibility problem* (CFP) is to find a member of C, the intersection of the C_i, if this intersection is non-empty. The *successive orthogonal projections* (SOP) method [36] is the following. Begin with an arbitrary x^0. For $k = 0, 1, ...$, and $i = k(\mathrm{mod}\, I) + 1$, let

$$x^{k+1} = P_i x^k, \tag{8.1}$$

where $P_i x$ denotes the orthogonal projection of x onto the set C_i. Since each of the operators P_i is firmly non-expansive, the product

$$T = P_I P_{I-1} \cdots P_2 P_1 \tag{8.2}$$

is averaged. Since C is not empty, T has fixed points. By the KM Theorem, the sequence $\{x^k\}$ converges to a member of C. It is useful to note that the limit of this sequence will not generally be the point in C closest to x^0; it is if the C_i are hyperplanes, however.

In [8] Bregman extends the SOP method to the case in which the projections are not orthogonal, but are with respect to a Bregman distance; this is the *successive generalized projection* (SGP) algorithm.

8.2 Using Multiple Distances

It is interesting to note that, in the BI-MART and BI-EMML methods, the generalized projections we employ involve weighted KL distances that vary with the hyperplanes. This leads to the conjecture that Bregman's SGP method can be further extended so that, at each step, a different Bregman distance is used. Simple counter-examples exist that show that merely allowing the distances to vary will not suffice. However, it was shown in [14] that such an extension of the SGP is possible, if we employ a generalized relaxation of the projections. This *multi-distance* SGP (MSGP) closely resembles the BI-SUMMA.

For each $k = 1, 2, ...$ and $i = k(\mathrm{mod}\, I)$, we let $M_i(x^{k-1})$ denote the member of C_i minimizing the Bregman distance $D_{f_i}(x, x^{k-1})$. Then we minimize

$$G_k(x) = D_{f_i}(x, M_i(x^{k-1})) + D_h(x, x^{k-1}) - D_{f_i}(x, x^{k-1}) \tag{8.3}$$

to get x^k. We assume that $D_h(x, z)$ is a dominating Bregman distance for the family $\{D_{f_i}(x, z)\}$, that is,

$$D_h(x, z) \geq D_{f_i}(x, z),$$

for all appropriate x and z. With suitable restrictions on the functions h and f_i, the sequence $\{x^k\}$ generated by the MSGP converges to a point in the intersection of the C_i [14].

8.3 The CQ Algorithm

A special case of the CFP is the *split feasibility problem* (SFP), which is to find a member of the closed convex set C in R^J for which Ax is a member of the closed convex set Q in R^I. In [14] the MSGP algorithm was applied to the SFP and an iterative solution method was obtained. That method was not completely satisfactory, in that, like similar iterative solutions given by others, each iterative step involved solving a system of linear equations. Subsequently, a different, and more practical, iterative method for solving the SFP, the CQ algorithm, was discovered [15]. It can be shown that convergence of the CQ algorithm follows from the KM Theorem [16]. Recent work by Combettes and Wajs reveals that the CQ algorithm is a special case of forward-backward splitting [27].

8.4 The Agmon-Motzkin-Schoenberg Algorithm

The Agmon-Motzkin-Schoenberg (AMS) algorithm [1, 45] is an iterative method for solving a system of linear inequalities $Ax \geq b$. Both the ART and the AMS algorithms are examples of the method of projection onto convex sets. The AMS algorithm is a special case of the cyclic subgradient projection (CSP) method, so that convergence of the AMS, in the consistent case, follows from the convergence theorem for the CSP algorithm. In the case of ART, the sets C_i are hyperplanes in R^J; suppose now that we take the C_i to be half-spaces and consider the problem of finding x such that $Ax \geq b$.

For each i let H_i be the half-space $H_i^+ = \{x|(Ax)_i \geq b_i\}$. Then x will be in the intersection of the sets $C_i = H_i^+$ if and only if $Ax \geq b$. Methods for solving this CFP, such as Hildreth's algorithm, are discussed in the book by Censor and Zenios [24]. Of particular interest for us here is the behavior of the AMS algorithm:

Algorithm 8.1 (Agmon-Motzkin-Schoenberg) *Let x^0 be arbitrary. Having found x^k, define*

$$x_j^{k+1} = x_j^k + A_{i(k)j}(b_{i(k)} - (Ax^k)_{i(k)})_+. \tag{8.4}$$

The AMS algorithm converges to a solution of $Ax \geq b$ in the consistent case, that is, if there are solutions to $Ax \geq b$. If there are no solutions, the AMS algorithm converges cyclically, that is, subsequences associated with the same m converge, as has been shown by De Pierro and Iusem [30], and by Bauschke, Borwein and Lewis [6].

8.5 Some Open Questions

Algorithms for solving the CFP fall into two classes: those that employ all the sets C_i at each step of the iteration (the so-called *simultaneous methods*) and those that do not (the *row-action algorithms* or, more generally, *block-iterative methods*). In the consistent case, in which the intersection of the convex sets C_i is nonempty, all reasonable algorithms are expected to converge to a member of that intersection; the limit may or may not be the member of the intersection closest to the starting vector x^0.

In the inconsistent case, in which the intersection of the C_i is empty, simultaneous methods typically converge to a minimizer of a *proximity function* [21], such as

$$f(x) = \textstyle\sum_{i=1}^I ||x - P_{C_i}x||_2^2, \tag{8.5}$$

if a minimizer exists.

Methods that are not simultaneous cannot converge in the inconsistent case, since the limit would then be a member of the (empty) intersection. Such methods often exhibit what is called *cyclic convergence*; that is, subsequences converge to finitely many distinct limits comprising a limit cycle. Once a member of this limit cycle is reached, further application of the algorithm results in passing from one member of the limit cycle to the next. Proving the existence of these limit cycles seems to be a difficult problem. For the particular case of two non-intersecting convex sets, the existence of a limit cycle for the SOP algorithm can be obtained as a consequence of the convergence of the CQ algorithm (see [19], p. 202).

When $Ax = b$ has no solutions, it has been shown that ART converges subsequentially to a limit cycle. Similarly, when the system $y = Px$ has no non-negative solution, BI-MART and BI-EMML have always been observed to exhibit subsequential convergence to a limit cycle, but no proof of the existence of limit cycles for these algorithms has been discovered.

In the proof of convergence of BI-SUMMA, it was necessary to assume that $f(\hat{x}) = 0$, so that each of the functions $f_i(x)$ is minimized simultaneously at $\hat{x}$. For the BI-ART, this means that $Ax = b$ has solutions, and for the BI-MART, that $y = Px$ has non-negative solutions. It seems natural to assume that, in the absence of a simultaneous minimizer of the $f_i(x)$, other instances of BI-SUMMA will also exhibit subsequential convergence to a limit cycle, but there is no proof of this, so far.

8.5.1 Do Limit Cycles Always Exist?

Tanabe [50] showed the existence of a limit cycle for the ART (see also [29]), in which the convex sets are hyperplanes. The SOP method may fail to have a limit cycle for certain choices of the convex sets. For example, if, in R^2, we take C_1 to be the lower half-plane and $C_2 = \{(x, y)|x > 0, y \geq 1/x\}$, then the SOP algorithm fails to produce a limit cycle. However, Gubin, Polyak and Riak [36] prove weak convergence to a limit cycle for the method of SOP in Hilbert space, under the assumption that at least one of the C_i is bounded, hence weakly compact. In [6] Bauschke, Borwein and Lewis present a wide variety of results on the existence of limit cycles. In particular, they prove that if each of the convex sets C_i in Hilbert space is a convex polyhedron, that is, the intersection of finitely many half-spaces, then there is a limit cycle and the subsequential convergence is in norm. This result includes the case in which each C_i is a half-space, so implies the existence of a limit cycle for the AMS algorithm.

8.5.2 What is the Limit Cycle?

Once we know that a limit cycle exists, it is reasonable to ask what its properties are. For the ART case, Eggermont *et al.* [31] have shown that the limit cycle can be made to reduce to a singleton set containing just the least-squares solution, if suitable strong under-relaxation is employed. Browne and De Pierro [9] give a similar result for the BI-EMML. In both cases, the strong under-relaxation compresses the limit cycle into the limit of the simultaneous version of the algorithm, which is the minimizer of a proximity function, the mean-square distance in the case of ART, and the minimizer of $KL(y, Px)$, in the case of BI-EMML. This leads us to ask what the connection is between the vectors of the limit cycle and the limit of the simultaneous version of the algorithm. In particular, we would like to know how the members of the ART limit cycle are related to the least-squares solution, and how to use them to calculate the least-squares solution.

8.5.3 Where is the Limit Cycle?

In [12] it was shown that, if A has full rank and $I = J + 1$, then the vectors of the ART limit cycle are all the same distance from the least-squares solution, that is, the limit cycle lies on a sphere in R^J centered at the least-squares solution (see also [18]). It is a curious fact that the condition $I = J + 1$ appears necessary. There are counter-examples in other cases, so that, if a more general result is available, it will be more complicated than simply having the vectors of the limit cycle lie on a sphere centered at the least-squares solution.

It was also shown there that the least-squares solution could be obtained from the vectors of the limit cycle by means of a *feedback* procedure; related results were also obtained for the BI-MART and BI-EMML (see also [19]). Nevertheless, we still do not have a useful characterization of the vectors of the limit cycle, nor even a proof of its existence, for most of these block-iterative algorithms.

9 Appendix A: The Krasnoselskii-Mann Theorem

For any operator $T : R^J \to R^J$ and $G = I - T$, where I denotes the identity operator, we have

$$||x - y||^2 - ||Tx - Ty||^2 = 2\langle Gx - Gy, x - y\rangle \ - ||Gx - Gy||^2, \tag{9.1}$$

for all x and y in the domain of T. An operator $G : R^J \to R^J$ is ν-inverse strongly monotone if, for each x and y in its domain, we have

$$\langle Gx - Gy, x - y\rangle \ \geq ||Gx - Gy||^2.$$

An operator $N : R^J \to R^J$ is *non-expansive* if, for all x and y in its domain, we have

$$||Nx - Ny|| \leq ||x - y||.$$

An operator $A : R^J \to R^J$ is *averaged* if, for non-expansive operator N and some scalar $\alpha \in (0, 1)$, we have

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

An operator $F : R^J \to R^J$ is *firmly non-expansive* if, for all x and y in its domain, we have

$$\langle Fx - Fy, x - y \rangle \geq ||Fx - Fy||^2.$$

Using the identity in Equation (9.1), one shows that an operator T is non-expansive, averaged, or firmly non-expansive, if and only if its complement G is $\frac{1}{2}$-ism, $\frac{1}{2\alpha}$-ism, for some $0 < \alpha < 1$, or 1-ism, respectively.

The Krasnoselskii-Mann Theorem [44] is the following:

Theorem 9.1 *Let N be a non-expansive operator and $A = (1 - \alpha)I + \alpha N$, for some $\alpha \in (0, 1)$. If the operator N has fixed points, that is, there are vectors x such that $Nx = x$, then, for any starting vector x^0, the sequence $\{A^k x^0\}$ converges to a fixed point of N.*

The class of averaged operators is closed to products [4] and includes orthogonal projections onto closed convex sets, as well as operators of the form $A = I - \gamma\nabla f$, for any convex function f whose gradient is L-Lipschitz continuous, and any γ in the interval $(0, 2/L)$ [3, 34]. For related results, see [26].

10 Appendix B: The Night Sky Theorems

For the real system $Ax = b$, consider the *non-negatively constrained least-squares* problem of minimizing the function $||Ax - b||_2$, subject to the constraints $x_j \geq 0$ for all j. Although there may be multiple solutions $\hat{x}$, we know, at least, that $A\hat{x}$ is the same for all solutions.

According to the Karush-Kuhn-Tucker Theorem, the vector $A\hat{x}$ must satisfy the condition

$$\sum_{i=1}^{I} A_{ij}((A\hat{x})_i - b_i) = 0 \tag{10.1}$$

for all j for which $\hat{x}_j > 0$ for some solution $\hat{x}$. Let S be the set of all indices j for which there exists a solution $\hat{x}$ with $\hat{x}_j > 0$. Then Equation (10.1) must hold for all j in S. Let Q be the matrix obtained from A by deleting those columns whose index j is not in S. Then $Q^T(A\hat{x} - b) = 0$. If Q has full rank and the cardinality of S is greater than or equal to I, then Q^T is one-to-one and $A\hat{x} = b$. We have proven the following result.

Theorem 10.1 *Suppose that A has the full-rank property, that is, A and every matrix Q obtained from A by deleting columns have full rank. Suppose there is no nonnegative* *solution of the system of equations* $Ax = b$*. Then there is a subset* S *of the set* $\{j = 1, 2, ..., J\}$*, with cardinality at most* $I - 1$*, such that, if* $\hat{x}$ *is any minimizer of* $||Ax - b||_2$ *subject to* $x \geq 0$*, then* $\hat{x}_j = 0$ *for* j *not in* S*. Therefore,* $\hat{x}$ *is unique.*

When $\hat{x}$ is a vectorized two-dimensional image and $J > I$, the presence of at most $I-1$ positive pixels makes the resulting image resemble stars in the sky; for that reason this theorem and the related results for the EMML, and SMART algorithms ([10]), as well as for block-iterative versions, are sometimes called *night sky* theorems. The zero-valued pixels typically appear scattered throughout the image. This behavior occurs with all the algorithms discussed so far that impose nonnegativity, whenever the real system $Ax = b$ has no nonnegative solutions.

References

[1] Agmon, S. (1954) “The relaxation method for linear inequalities” , *Canadian Journal of Mathematics*, **6**, pp. 382–392.

[2] 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.

[3] Baillon, J.-B., and Haddad, G. (1977) “Quelques proprietes des operateurs angle-bornes et n-cycliquement monotones” , *Israel J. of Mathematics*, **26** 137-150.

[4] Bauschke, H., and Borwein, J. (1996) “On projection algorithms for solving convex feasibility problems.” *SIAM Review*, **38 (3)**, pp. 367–426.

[5] Bauschke, H., and Borwein, J. (1997) “Legendre functions and the method of random Bregman projections.” *Journal of Convex Analysis*, **4**, pp. 27–67.

[6] Bauschke, H., Borwein, J., and Lewis, A. (1997) “The method of cyclic projections for closed convex sets in Hilbert space.” *Contemporary Mathematics: Recent Developments in Optimization Theory and Nonlinear Analysis*, **204**, American Mathematical Society, pp. 1–38.

[7] Bertsekas, D.P. (1997) “A new class of incremental gradient methods for least squares problems.” *SIAM J. Optim.*, **7**, pp. 913-926.

[8] 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.

[9] 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.

[10] Byrne, C. (1993) “Iterative image reconstruction algorithms based on cross-entropy minimization.” *IEEE Transactions on Image Processing* **IP-2**, pp. 96–103.

[11] Byrne, C. (1996) “Block-iterative methods for image reconstruction from projections.” *IEEE Transactions on Image Processing* **IP-5**, pp. 792–794.

[12] Byrne, C. (1997) “Convergent block-iterative algorithms for image reconstruction from inconsistent data.” *IEEE Transactions on Image Processing* **IP-6**, pp. 1296–1304.

[13] Byrne, C. (1998) “Iterative algorithms for deblurring and deconvolution with constraints,” *Inverse Problems*, **14**, pp. 1455–1467 .

[14] 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.

[15] Byrne, C. (2002) “Iterative oblique projection onto convex sets and the split feasibility problem.” *Inverse Problems* **18**, pp. 441–453.

[16] Byrne, C. (2004) “A unified treatment of some iterative algorithms in signal processing and image reconstruction.” *Inverse Problems* **20**, pp. 103–120.

[17] Byrne, C. (2005) “Choosing parameters in block-iterative or ordered-subset reconstruction algorithms” *IEEE Transactions on Image Processing*, **14 (3)**, pp. 321–327.

[18] Byrne, C. (2005) *Signal Processing: A Mathematical Approach*, AK Peters, Publ., Wellesley, MA.

[19] Byrne, C. (2007) *Applied Iterative Methods*, AK Peters, Publ., Wellesley, MA.

[20] Byrne, C. (2008) "Sequential unconstrained minimization algorithms for constrained optimization." *Inverse Problems*, **24**, 1–27.

[21] 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.

[22] Censor, Y. and Segman, J. (1987) "On block-iterative maximization." *J. of Information and Optimization Sciences* **8**, pp. 275–291.

[23] Censor, Y., and Zenios, S.A. (1992) "Proximal minimization algorithm with D-functions." *Journal of Optimization Theory and Applications*, **73(3)**, pp. 451–464.

[24] Censor, Y. and Zenios, S.A. (1997) *Parallel Optimization: Theory, Algorithms and Applications*. New York: Oxford University Press.

[25] Cimmino, G. (1938) "Calcolo approssimato per soluzioni dei sistemi di equazioni lineari." *La Ricerca Scientifica XVI, Series II, Anno IX* **1**, pp. 326–333.

[26] Combettes, P. (2001) "Quasi-Fejérian Analysis of some optimization algorithms." in *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications*, edited by D. Butnariu, Y. Censor and S. Reich, pp. 115–152, Studies in Computational Mathematics 8. Amsterdam: Elsevier Publ., 2001.

[27] Combettes, P., and Wajs, V. (2005) "Signal recovery by proximal forward-backward splitting." *Multiscale Modeling and Simulation*, **4(4)**, pp. 1168–1200.

[28] Darroch, J. and Ratcliff, D. (1972) "Generalized iterative scaling for log-linear models." *Annals of Mathematical Statistics* **43**, pp. 1470–1480.

[29] Dax, A. (1990) "The convergence of linear stationary iterative processes for solving singular unstructured systems of linear equations," *SIAM Review*, **32**, pp. 611–635.

[30] 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.

[31] 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.

[32] Fiacco, A., and McCormick, G. (1990) *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* Philadelphia, PA: SIAM Classics in Mathematics (reissue).

[33] 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.

[34] Golshtein, E., and Tretyakov, N. (1996) *Modified Lagrangians and Monotone Maps in Optimization.* New York: John Wiley and Sons, Inc.

[35] 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.

[36] 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.

[37] Herman, G. T. and Meyer, L. (1993) "Algebraic reconstruction techniques can be made computationally efficient."*IEEE Transactions on Medical Imaging* **12**, pp. 600–609.

[38] 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.

[39] Hudson, M., Hutton, B., and Larkin, R. (1992) "Accelerated EM reconstruction using ordered subsets." *Journal of Nuclear Medicine*, **33**, p.960.

[40] 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.

[41] Kullback, S. and Leibler, R. (1951) "On information and sufficiency."*Annals of Mathematical Statistics* **22**, pp. 79–86.

[42] 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.

[43] Lange, K. and Carson, R. (1984) “EM reconstruction algorithms for emission and transmission tomography.” *Journal of Computer Assisted Tomography* **8**, pp. 306–316.

[44] Mann, W. (1953) “Mean value methods in iteration.” *Proc. Amer. Math. Soc.* **4**, pp. 506–510.

[45] Motzkin, T., and Schoenberg, I. (1954) “The relaxation method for linear inequalities.” *Canadian Journal of Mathematics*, **6**, pp. 393–404.

[46] Narayanan, M., Byrne, C. and King, M. (2001) “An interior point iterative maximum-likelihood reconstruction algorithm incorporating upper and lower bounds with application to SPECT transmission imaging.” *IEEE Transactions on Medical Imaging* **TMI-20 (4)**, pp. 342–353.

[47] Schmidlin, P. (1972) “Iterative separation of sections in tomographic scintigrams.” *Nucl. Med.* **15(1)**.

[48] Shepp, L., and Vardi, Y. (1982) “Maximum likelihood reconstruction for emission tomography.” *IEEE Transactions on Medical Imaging*, **MI-1**, pp. 113–122.

[49] 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.

[50] Tanabe, K. (1971) “Projection method for solving a singular system of linear equations and its applications.” *Numer. Math.* **17**, pp. 203–214.

[51] Teboulle, M. (1992) “Entropic proximal mappings with applications to nonlinear programming” *Mathematics of Operations Research*, **17(3)**, pp. 670–690.

[52] 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.

[53] Wernick, M. and Aarsvold, J., editors (2004) *Emission Tomography: The Fundamentals of PET and SPECT*. San Diego: Elsevier Academic Press.