Inverse Problems **14** (1998) 1455–1467. Printed in the UK PII: S0266-5611(98)95297-2

Iterative algorithms for deblurring and deconvolution with constraints

Charles Byrne†

Department of Mathematical Sciences, University of Massachusetts at Lowell, Lowell, MA 01854, USA

Received 23 June 1998, in final form 10 September 1998

Abstract. The deblurring problem is that of recovering the function $c = c(t)$ from (noisy) values $\int h(s,t)c(t)\,\mathrm{d}t = d(s)$. The discrete finite version of the problem is to solve the system of linear equations $Hc = d$ for c, where H is a matrix and d and c are vectors. When the kernel $h(s,t)$ is a function of the difference $(s-t)$, the deblurring problem becomes a deconvolution problem. The use of iterative algorithms to effect deblurring subject to non-negativity constraints on c has been presented by Snyder *et al* for the case of non-negative kernel function h. In this paper we extend these algorithms to include upper and lower bounds on the entries of the desired solution. We show that any linear deblurring problem involving a real kernel h can be transformed into a linear deblurring problem involving a non-negative kernel. Therefore our algorithms apply to general deblurring and deconvolution problems. These algorithms converge to a solution of the system of equations $y = Px$, with $P = [P_{ij}]$, $P_{ij} \geqslant 0$ for $i = 1, \ldots, I$, $j = 1, \ldots, J$, satisfying the vector inequalities $a \leqslant x \leqslant b$, whenever such a solution exists. When there is no solution satisfying the constraints the simultaneous versions converge to an approximate solution that minimizes a cost function related to the Kullback–Leibler cross-entropy and the Fermi–Dirac generalized entropy.

1. Introduction

The deblurring problem, in its continuous formulation, is that of recovering the function $c = c(t)$ from (noisy) values $\int h(s,t)c(t)\,\mathrm{d}t = d(s)$. The discrete finite version of the problem is to solve the system of linear equations $Hc = d$ for c, where H is an I by J matrix and d and c are vectors. When the kernel $h(s,t)$ is a function of the difference $(s-t)$, the deblurring problem becomes a deconvolution problem. In that case the entries of H depend only on $i - j$, so the matrix H is Toeplitz, that is, constant along each diagonal. Typically, the matrix H is ill-conditioned. We are usually free to choose the dimension J of the vector c, but do not have the same freedom in the choice of I, so the problem becomes under-determined. In any case, the data are typically inadequate and it is helpful to include some prior information about the desired solution c.

The use of iterative algorithms to effect deblurring subject to non-negativity constraints on c has been presented by Snyder *et al* [1], for the case of non-negative kernel function h. In [2] Vardi and Lee show that a wide variety of linear inverse problems involving the inversion of integral transforms of the form $\int h(s,t)c(t)\,\mathrm{d}t = d(s)$ can be reformulated as statistical estimation from incomplete data, suggesting the use of the expectation maximization maximum likelihood method (EMML) [3]; they also restrict their attention to non-negative

† E-mail address: byrnec@cs.uml.edu

kernels. In his comments on [2] following that paper, Lindsay notes the similarity between work on optimization and efforts within the literature on statistical mixture problems, where analogous integral equations occur and iterative algorithms for their solution are sought (see [4–8]). In mixture problems the kernel function $h(s, t)$ is a family of probability density functions of the variable s, with t playing the role of a random parameter, so it is natural that $h(s, t)$ be non-negative. In this paper we extend these algorithms to include upper and lower bounds on the entries of the desired solution. In [9] Vardi and Zhang maximize the likelihood function, subject to such upper and lower bound constraints on the mixing probability, using a variant of the iterative expectation maximization (EM) algorithm. Their algorithm is similar to one of the algorithms presented here.

In deblurring and deconvolution problems the kernel function may well have negative values. We may wish to view the integral $\int h(s, t)c(t)\,\mathrm{d}t = d(s)$ as defining a general integral transform, such as a Fourier sine or cosine transform or a band-limiting convolution with a sinc function, in which cases non-negativity of the kernel does not hold. Although our algorithms require that the matrix (or the discretized kernel) of the system have non-negative entries, as in the previous work, we show that any linear deblurring problem involving a real kernel h can be transformed into a linear deblurring problem involving a non-negative kernel; therefore our algorithms apply to general deblurring and deconvolution problems.

We extend the EMML [3, 10–16], the multiplicative algebraic reconstruction technique (MART) [20, 22], the simultaneous MART (SMART) [14–16] and their rescaled block-iterative versions [17–19] (see also [21, 23]) to obtain algorithms that converge to a solution of the system of equations $y = Px$, with $P = [P_{ij}]$, $P_{ij} \geqslant 0$ for $i = 1, \ldots, I$, $j = 1, \ldots, J$, satisfying the vector inequalities $a \leqslant x \leqslant b$, whenever such a solution exists. When there is no solution satisfying the constraints the simultaneous algorithms, which employ only a single block consisting of all the data, converge to an approximate solution that minimizes a cost function related to the Kullback–Leibler cross-entropy and the Fermi–Dirac generalized entropy.

The cross-entropy or Kullback–Leibler (KL) distance [24] will play a major role in what follows. For scalars $a > 0$ and $b > 0$ define

$$KL(a, b) = a \log \frac{a}{b} + b - a \tag{1.1}$$

$KL(a, 0) = +\infty$, $KL(0, b) = b$ and $KL(0, 0) = 0$. Since the function $f(t) = t - 1 - \log t$ is non-negative and equals zero if and only if $t = 1$, it follows that $KL(a, b) = a[\frac{b}{a} - 1 - \log \frac{b}{a}] = 0$ if and only if $a = b$. We extend the KL distance to vectors componentwise; that is, for vectors $x = (x_1, \ldots, x_J)^T$ and $z = (z_1, \ldots, z_J)^T$ with non-negative entries, we define $KL(x, z)$ as

$$KL(x, z) = \sum_j KL(x_j, z_j) = \sum_{j=1}^{J} x_j \log \frac{x_j}{z_j} + z_j - x_j \tag{1.2}$$

with $\sum_j = \sum_{j=1}^{J}$. It follows that $KL(x, z) = 0$ if and only if the vectors x and z are the same vector.

The EMML and SMART algorithms and their rescaled block-iterative versions (RBI-EMML and RBI-SMART) produce a non-negative solution of the system of linear equations $y = Px$, whenever such a solution exists, where $y > 0$, $P = [P_{ij}]$, $P_{ij} \geqslant 0$ for $i = 1, \ldots, I$, $j = 1, \ldots, J$ and a non-negative vector x is sought. With x^0 the starting vector, the RBI-SMART algorithms converge to the solution x for which the Kullback–Leibler distance $KL(x, x^0)$ is minimized.

We obtain block-iterative algorithms by partitioning the set $\{i = 1, \ldots, I\}$ into disjoint subsets S_n, $n = 1, \ldots, N$, for $N > 1$. By suitably rescaling the equations, we can accelerate the convergence of these algorithms [19]. Specifically, for each n let m_n be the maximum over $j = 1, \ldots, J$ of the sums $\sum_{i \in S_n} P_{ij}$. For each i we then divide both sides of the ith equation in $y = Px$ by m_n, where n is such that i is in S_n. We shall assume that this has been done, without changing our original notation $y = Px$. Therefore we now have that, for each j, the sum $\sum_{i \in S_n} P_{ij}$ is at most one and equals one for some values of j. The RBI-SMART algorithm is as follows. Starting with x^0, for $k = 0, 1, \ldots$ let the block index be $n = k(\text{mod } N) + 1$ and set

$$x_j^{k+1} = x_j^k \exp\left(\sum_{i \in S_n} P_{ij} \log\left(\frac{y_i}{Px_i^k}\right)\right) \qquad j = 1, \ldots, J, \tag{1.3}$$

where $Px_i^k = (Px^k)_i$ is the ith entry of the vector Px^k. When there is a non-negative solution of $y = Px$ this algorithm converges to that non-negative solution for which the distance $KL(x, x^0)$ is minimized. When each of the blocks S_n contains only one index we get the rescaled MART (RMART) algorithm. When $N = 1$ we get the SMART algorithm.

The rescaled block-iterative EMML algorithm (RBI-EMML) is as follows. Starting with x^0, for $k = 0, 1, \ldots$ let the block index be $n = k(\text{mod } N) + 1$ and set

$$x_j^{k+1} = x_j^k\left(1 - \sum_{i \in S_n} P_{ij}\right) + x_j^k\left(\sum_{i \in S_n} P_{ij}\left(\frac{y_i}{Px_i^k}\right)\right) \qquad j = 1, \ldots, J. \tag{1.4}$$

When there is a non-negative solution of $y = Px$ this algorithm converges to such a solution. When each of the blocks S_n contains only one index we get the rescaled EMART (REMART) algorithm. When $N = 1$ we get the EMML algorithm, because the columns of P have been normalized to have sum equal to one; without this normalization we get a closely related algorithm. The RBI-EMML algorithm differs from the row action maximum likelihood approach (RAMLA) of Browne and De Pierro [25] only in the way the rescaling is used: we rescale to accelerate convergence, they rescale differently to minimize the effects of noise in the data.

Suppose that, instead of the constraints $x_j \geqslant 0$, we wish to impose the constraints that x be contained within a box in R^J, that is, $a_j \leqslant x_j \leqslant b_j$, for $j = 1, \ldots, J$, where $a = \{a_j\}$ and $b = \{b_j\}$ are prior lower and upper vector bounds on $x = \{x_j\}$. The algorithms we present below extend the MART and EMML algorithms to incorporate these inequality constraints, hence the names ABMART and ABEMML. The ABEMML and ABMART algorithms converge to a solution of $y = Px$ with $a \leqslant x \leqslant b$ and, in addition, the ABMART algorithm minimizes the quantity

$$\psi^0(x) = KL(x - a, x^0 - a) + KL(b - x, b - x^0) \tag{1.5}$$

over these same x, provided $a < x^0 < b$ and there is a solution of $y = Px$ with $a \leqslant x \leqslant b$. The negative of the quantity in (1.5) is a generalization of the Fermi–Dirac generalized entropy, which is obtained by taking $a_j = 0$ and $b_j = 1$ for all $j = 1, \ldots, J$.

The focus of this paper is on presenting the algorithms and establishing their theoretical properties; proofs of convergence are given in detail. Application of these algorithms to practical image processing problems is in progress [26], but we do not discuss applications in this paper. Noise is usually an important consideration in deblurring and deconvolution. When the algorithms presented here are applied to actual data, we expect some form of regularization to be used to stabilize them. This has been discussed at length in the literature on the EM algorithm and we would only wish to say that the same concerns arise here and that the same remedies should be used. These include the use of penalty functions, smoothness constraints, stopping rules and the like (see the discussion in [14] and references given therein).

2. From general kernels to non-negative ones

The algorithms we present below require that the kernel matrix have all non-negative entries. Because we want to be able to apply our algorithms to more general situations, in which the blurring kernel function h is permitted to take on negative values, we must consider how we might transform such more general problems into those for which the algorithms apply. We note that, when the transformation is applied, certain special structure, such as the Toeplitz property of the matrix, will almost certainly be destroyed.

In some special cases there are tricks that we can use to achieve non-negativity of the kernel. Suppose, for example that we have limited Fourier transform data,

$$f(n) = \int_{-\infty}^{+\infty} F(\omega)\exp(-\mathrm{i}n\omega)\,\mathrm{d}\omega/2\pi \qquad |n| \leqslant N \tag{2.1}$$

from which we want to reconstruct the original function $F(\omega)$. First, we convert to sine and cosine Fourier transformation. Then, adding the function that is one everywhere to the sine and cosine functions we obtain a non-negative kernel function; since we have the value of $f(0)$ we add $f(0)$ to each of the sine and cosine transform values before inverting. Similarly, if we have values of a sinc convolution transformation, including the DC term, describing a band-limiting operation, we can use the fact that sinc is bounded below by a negative constant to subtract this constant from sinc, obtaining a non-negative kernel. For more general situations, we proceed as described below.

Suppose that $Hc = d$ is an arbitrary (real) system of linear equations, with the matrix $H = [H_{ij}]$. Rescaling the equations if necessary, we may assume that for each j the column sum $\sum_i H_{ij}$ is nonzero; note that if a particular rescaling of one equation to make the first column sum nonzero causes another column sum to become zero, we simply choose a different rescaling. Since there are finitely many columns to worry about, we can always succeed in making all the column sums nonzero. Now redefine H and c as follows. Replace H_{kj} with $G_{kj} = \frac{H_{kj}}{\sum_i H_{ij}}$ and c_j with $g_j = c_j \sum_i H_{ij}$. The product Hc is equal to Gg and the new matrix G has column sums equal to one. The system $Gg = d$ still holds, but now we know that $\sum_i d_i = d_+ = \sum_j g_j = g_+$. Let U be the matrix whose entries are all one and let $t \geqslant 0$ be large enough so that $B = G + tU$ has all non-negative entries. Then $Bg = Gg + (tg_+)u$, where u is the vector whose entries are all one. So the new system of equations to solve is $Bg = d + (td_+)u = y$.

In the algorithms that follow we make the further assumption that the column sums of the matrix are all one. To achieve this, we make one additional renormalization. Replace B_{kj} with $P_{kj} = \frac{B_{kj}}{\sum_i B_{ij}}$ and g_j with $x_j = g_j \sum_i B_{ij}$; the product Bg is equal to Px and the new matrix P is non-negative and has column sums equal to one. For each i we then divide both sides of the ith equation in $y = Px$ by m_n, where n is such that i is in S_n; we shall assume that this has been done, without changing our notation $y = Px$. From prior upper and lower bounds on the entries of the original vector c we construct upper and lower bounds on the entries of the vector x, that is, we seek x so that $y = Px$ and, for each j, $a_j \leqslant x_j \leqslant b_j$.

If there are such vectors x satisfying these equations and constraints, then each of our algorithms converges to such a vector and, by transforming back to the original c, we obtain a deblurred solution consistent with the original data and constraints. If there are no vectors c satisfying the original constraints and data equations, then there will be no vector x satisfying $y = Px$ with $a_j \leqslant x_j \leqslant b_j$ for each j. In this case our simultaneous algorithms converge to vectors satisfying the constraint inequalities and providing approximate solutions to the linear equations.

3. The ABMART algorithm

The ABMART algorithm is as follows. We assume that $Pa_i < y_i < Pb_i$ for all i and that $a_j < x_j^0 < b_j$ for all j. Then, for $j = 1, \ldots, J, k = 0, 1, \ldots$ and $n = k(\text{mod } N) + 1$, we have

$$x_j^{k+1} = \alpha_j^k b_j + (1 - \alpha_j^k)a_j \tag{3.1}$$

with

$$\alpha_j^k = \frac{[c_j^k \prod^n (d_i^k)^{P_{ij}}]}{[1 + c_j^k \prod^n (d_i^k)^{P_{ij}}]} \tag{3.2}$$

$$c_j^k = \frac{(x_j^k - a_j)}{(b_j - x_j^k)} \tag{3.3}$$

and

$$d_i^k = \frac{(y_i - Pa_i)(Pb_i - Px_i^k)}{(Pb_i - y_i)(Px_i^k - Pa_i)} \tag{3.4}$$

where $\prod^n$ denotes the product over those indices i that are in the block S_n. All terms in (3.1)–(3.4) are positive. We see from (3.1) that each term of the iterative sequence $\{x_j^k\}$ is a convex combination of the a_j and b_j. The iteration proceeds until convergence to a convex combination for which $y = Px$, if such exists. If there is no such solution of $y = Px$ then, for the simultaneous case, in which $N = 1$, the algorithm will converge to an approximate solution satisfying the constraints, as we shall see below. Specifically, the limit is the unique vector satisfying $a \leqslant x \leqslant b$ for which the function $KL(Px - Pa, y - Pa) + KL(Pb - Px, Pb - y)$ is minimized. In general, however, when we use more than one block and there is no solution satisfying the constraints then the algorithm cannot converge to a single vector. We suspect that we get the same sort of limit cycle behaviour found in other block-iterative methods [11], but we have no proof of this, as yet. If $a_j = 0$ and $b_j = +\infty$ for all j then we get the RBI-SMART algorithm.

We shall prove the following theorem.

Theorem 3.1. *If there is a solution x of the system of linear equations $y = Px$ that satisfies the constraints $a_j \leqslant x_j \leqslant b_j$, $j = 1, \ldots, J$, then, for any N and any partition $\{S_1, \ldots, S_N\}$ of the set $\{i = 1, \ldots, I\}$, the ABMART algorithm (3.1)–(3.4) converges to that solution of $y = Px$ satisfying the constraints for which $\{KL(x - a, x^0 - a) + KL(b - x, b - x^0)\}$ is minimized. If there are no such solutions, then, for $N = 1$, the iterative algorithm converges to the minimizer of $KL(Px - Pa, y - Pa) + KL(Pb - Px, Pb - y)$ for which $KL(x - a, x^0 - a) + KL(b - x, b - x^0)$ is minimized.*

4. Proof of convergence of ABMART

To prove convergence of the algorithm (3.1)–(3.4) we introduce $H_n(x, z)$, $n = 1, \ldots N$, each a measure of the distance between any two vectors x and z satisfying our constraints, given by

$$\begin{aligned} H_n(x, z) = {} & KL(x - a, z - a) - \sum_{i \in S_n} KL(Px_i - Pa_i, Pz_i - Pa_i) \\ & + \sum_{i \in S_n} KL(Px_i - Pa_i, y_i - Pa_i) + KL(b - x, b - z) \\ & - \sum_{i \in S_n} KL(Pb_i - Px_i, Pb_i - Pz_i) + \sum_{i \in S_n} KL(Pb_i - Px_i, Pb_i - y_i). \end{aligned} \tag{4.1}$$

This distance measure is non-negative (corollary 4.1 below) and $H_n(x, x^k)$ achieves its minimum at $x = x^{k+1}$ (proposition 4.2 below). Note that $H_n(x, z) \geqslant H_n(x, x) = \sum_{i \in S_n} KL(Px_i - Pa_i, y_i - Pa_i) + \sum_{i \in S_n} KL(Pb_i - Px_i, Pb_i - y_i)$, for all suitable x and z.

The proof of the next proposition is a straightforward calculation and we omit the details.

Proposition 4.1. *For any non-negative vectors x and z with $x_+ = \sum_j x_j$ we have*

$$KL(x, z) = KL\left(x, \left(\frac{x_+}{z_+}\right) z\right) + KL(x_+, z_+)$$

so that $KL(x, z) \geqslant KL(x_+, z_+)$.

Corollary 4.1. *Since $\sum_{i \in S_n} P_{ij} \leqslant 1$ we have $KL(x, z) \geqslant \sum_{i \in S_n} KL(Px_i, Pz_i)$, for all vectors $x \geqslant 0$ and $z \geqslant 0$.*

Proposition 4.2. *For any vector x with $a \leqslant x \leqslant b$ we have*

$$H_n(x, x^k) = H_n(x^{k+1}, x^k) + KL(x - a, x^{k+1} - a) + KL(b - x, b - x^{k+1}). \tag{4.2}$$

Proof. The assertion follows from a lengthy, but elementary, calculation based on the definition of x^{k+1} and of H_n. □

We prove convergence in general, for the consistent case in which there exist solutions of $y = Px$ satisfying the constraints $a \leqslant x \leqslant b$. Then we prove convergence to an approximate solution for the inconsistent case, provided that $N = 1$.

For any x satisfying the constraints and $k = 0, 1, \ldots$ we define

$$\psi^k(x) = KL(x - a, x^k - a) + KL(b - x, b - x^k). \tag{4.3}$$

Then

$$\begin{aligned}
\psi^k(x) = H_n(x, x^k) + \Bigg[& \sum_{i \in S_n} KL(Px_i - Pa_i, Px_i^k - Pa_i) \\
& + \sum_{i \in S_n} KL(Pb_i - Px_i, Pb_i - Px_i^k) - H_n(x, x) \Bigg] \\
= H_n(x^{k+1}, x^k) + \psi^{k+1}(x) + \Bigg[& \sum_{i \in S_n} KL(Px_i - Pa_i, Px_i^k - Pa_i) \\
& + \sum_{i \in S_n} KL(Pb_i - Px_i, Pb_i - Px_i^k) - H_n(x, x) \Bigg] \\
\geqslant \psi^{k+1}(x) + & \sum_{i \in S_n} KL(Px_i - Pa_i, Px_i^k - Pa_i) \\
& + \sum_{i \in S_n} KL(Pb_i - Px_i, Pb_i - Px_i^k) + H_n(x^{k+1}, x^{k+1}) - H_n(x, x).
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\psi^k(x) - \psi^{k+1}(x) \geqslant & \sum_{i \in S_n} KL(Px_i - Pa_i, Px_i^k - Pa_i) + \sum_{i \in S_n} KL(Pb_i - Px_i, Pb_i - Px_i^k) \\
& + H_n(x^{k+1}, x^{k+1}) - H_n(x, x).
\end{aligned} \tag{4.4}$$

The consistent case. Let x be a solution of $y = Px$ such that $a \leqslant x \leqslant b$. Using this x in (4.4) we find that, since $H_n(x, x) = 0$, the sequence $\{\psi^k(x)\}$ is decreasing and so the sequence of its successive differences converges to zero. Consequently, the non-negative sequence $\{\sum_{i\in S_n} KL(y_i - Pa_i, Px_i^k - Pa_i)\}$ converges to zero. Since the sequence $\{x^k\}$ is bounded, we can select a cluster point, say x^*. It follows from the fact that the sequence $\{\sum_{i\in S_n} KL(y_i - Pa_i, Px_i^k - Pa_i)\}$ converges to zero that $y = Px^*$ and x^* satisfies the constraints. Therefore, we may use x^* in place of x. Then, we have that the sequence $\{\psi^k(x^*)\}$ is decreasing and so is converging to zero; therefore $\{x^k\}$ converges to x^*. Since, for each k, the difference $\psi^k(x) - \psi^{k+1}(x)$ does not depend on which solution x we are using, it follows that $\psi^0(x) - [KL(x-a, x^*-a) + KL(b-x, b-x^*)]$ is also independent of which solution we use. Therefore we have that x^* is the constrained solution for which $\{KL(x-a, x^0-a) + KL(b-x, b-x^0)\}$ is minimized.

The inconsistent case. The proof of convergence is similar for the inconsistent case, with $N = 1$. Now we let x minimize $H_1(x, x) = KL(Px - Pa, y - Pa) + KL(Pb - Px, Pb - y)$ subject to the constraints $a \leqslant x \leqslant b$. Again using (4.4), now with $N = 1$ and so $n = 1$ we find that, since $H_1(x^{k+1}, x^{k+1}) \geqslant H_1(x, x)$, the sequence $\{\psi^k(x) - \psi^{k+1}(x)\}$ is decreasing. The remainder of the proof is similar to the consistent case. This concludes the proof of theorem 3.1.

5. The ABEMML algorithm

The ABEMML algorithm is as follows. We assume that $Pa_i < y_i < Pb_i$ for all i and that $a_j < x_j^0 < b_j$ for all j. Then, for each $j = 1, \ldots, J, k = 0, 1, \ldots$ and $n = k(\text{mod } N) + 1$, we have

$$x_j^{k+1} = \alpha_j^k b_j + (1 - \alpha_j^k) a_j \tag{5.1}$$

with

$$\gamma_j^k = (x_j^k - a_j) e_j^k \tag{5.2}$$

$$\beta_j^k = (b_j - x_j^k) f_j^k \tag{5.3}$$

$$e_j^k = \left(1 - \sum_{i\in S_n} P_{ij}\right) + \sum_{i\in S_n} P_{ij} \left(\frac{y_i - Pa_i}{Px_i^k - Pa_i}\right) \tag{5.4}$$

$$f_j^k = \left(1 - \sum_{i\in S_n} P_{ij}\right) + \sum_{i\in S_n} P_{ij} \left(\frac{Pb_i - y_i}{Pb_i - Px_i^k}\right) \tag{5.5}$$

$$d_j^k = \gamma_j^k + \beta_j^k \tag{5.6}$$

and

$$\alpha_j^k = \gamma_j^k / d_j^k. \tag{5.7}$$

We see from (5.1) that each term of the iterative sequence $\{x_j^k\}$ is a convex combination of the a_j and b_j. The iteration proceeds until convergence to a convex combination for which $y = Px$, if such exists. If there is no such solution of $y = Px$ then, for the simultaneous case, in which $N = 1$, the algorithm will converge to an approximate solution satisfying the constraints, as we shall see below. Specifically, the limit is the unique vector satisfying $a \leqslant x \leqslant b$ for which the function $KL(y - Pa, Px - Pa) + KL(Pb - y, Pb - Px)$ is minimized. In general, however, when we use more than one block and there is no solution satisfying the constraints the algorithm cannot converge to a single vector. As with ABMART we probably get the same sort of limit cycle behaviour found in other block-iterative methods. If $a_j = 0$ and $b_j = +\infty$ for all j then we get the RBI-EMML algorithm.

We shall prove the following theorem.

Theorem 5.1. *If there is a solution x of the system of linear equations $y = Px$ that satisfies the constraints $a_j \leqslant x_j \leqslant b_j$, $j = 1, \ldots, J$, then, for any N and any partition $\{S_1, \ldots, S_N\}$ of the set $\{i = 1, \ldots, I\}$, the ABEMML algorithm (5.1)–(5.6) converges to a solution of $y = Px$ satisfying the constraints. If there are no such solutions, then, for $N=1$, the iterative algorithm converges to a minimizer of $KL(Px - Pa, y - Pa) + KL(Pb - Px, Pb - y)$.*

6. Proof of convergence of ABEMML

We first prove convergence of ABEMML for the consistent case, in which there exist solutions of $y = Px$ satisfying the bounds $a \leqslant x \leqslant b$. Then we prove convergence for the inconsistent case for $N = 1$: in this case the limit minimizes the cost function $KL(y - Pa, Px - Pa) + KL(Pb - y, Pb - Px)$ over x with $a \leqslant x \leqslant b$.

The consistent case. Assume throughout this section that $y = Px$ and $a \leqslant x \leqslant b$. We shall prove the following inequality:

$$\psi^k(x) - \psi^{k+1}(x) \geqslant \left[\sum_{i\in S_n} KL(y_i - Pa_i, Px_i^k - Pa_i) + \sum_{i\in S_n} KL(Pb_i - y_i, Pb_i - Px_i^k)\right]. \tag{6.1}$$

From (6.1) it will follow that the sequence $\{\psi^k(x)\}$ is decreasing and that the sequence $\{\sum_{i\in S_n} KL(y_i - Pa_i, Px_i^k - Pa_i) + \sum_{i\in S_n} KL(Pb_i - y_i, Pb_i - Px_i^k)\}$ converges to zero. From (5.1)–(5.7) we find that

$$d_j^k(x_j^{k+1} - a_j) = (b_j - a_j)\gamma_j^k \tag{6.2}$$

and

$$d_j^k(b_j - x_j^{k+1}) = (b_j - a_j)\beta_j^k. \tag{6.3}$$

For the left side of (6.1) we have

$$\psi^k(x) - \psi^{k+1}(x) = \sum_{j=1}^J (x_j - a_j)\log\left[\left(1 - \sum_{i\in S_n} P_{ij}\right) + \sum_{i\in S_n} P_{ij}\frac{y_i - Pa_i}{Px_i^k - Pa_i}\right]$$
$$+ \sum_{j=1}^J (b_j - x_j)\log\left[\left(1 - \sum_{i\in S_n} P_{ij}\right) + \sum_{i\in S_n} P_{ij}\frac{Pb_i - y_i}{Pb_i - Px_i^k}\right]$$
$$+ KL(b - a, d^k) - \sum_{j=1}^J d_j^k + \sum_{j=1}^J (b_j - a_j).$$

Using the concavity of the log function and the fact that $\sum_{j=1}^J d_j^k = \sum_{j=1}^J (b_j - a_j)$ we have that the left side of (6.1) is greater than or equal to

$$\sum_{j=1}^J (x_j - a_j)\left[\left(1 - \sum_{i\in S_n} P_{ij}\right)\log(1) + \sum_{i\in S_n} P_{ij}\log\left[\frac{y_i - Pa_i}{Px_i^k - Pa_i}\right]\right]$$
$$+ \sum_{j=1}^J (b_j - x_j)\left[\left(1 - \sum_{i\in S_n} P_{ij}\right)\log(1) + \sum_{i\in S_n} P_{ij}\log\left[\frac{Pb_i - y_i}{Pb_i - Px_i^k}\right]\right]$$

$$+KL(b-a, d^k) \geqslant \sum_{j=1}^{J}(x_j - a_j)\left[\sum_{i\in S_n} P_{ij}\log\left[\frac{y_i - Pa_i}{Px_i^k - Pa_i}\right]\right]$$

$$+\sum_{j=1}^{J}(b_j - x_j)\left[\sum_{i\in S_n} P_{ij}\log\left[\frac{Pb_i - y_i}{Pb_i - Px_i^k}\right]\right]$$

$$= \sum_{i\in S_n}[KL(y_i - Pa_i, Px_i^k - Pa_i) + KL(Pb_i - y_i, Pb_i - Px_i^k)].$$

This concludes the proof of (6.1). The proof of convergence follows from (6.1) in much the same way as that given for the ABMART. One significant difference is that we no longer have that the left side of (6.1) is independent of the choice of solution x. Consequently, we cannot characterize the solution we obtain using the ABEMML algorithm.

The inconsistent case. Now we consider the behaviour of the iterative scheme (5.1) for the case of $N = 1$, when there may not be a solution of $y = Px$ satisfying the constraints. In place of (5.1)–(5.7) we now write

$$x_j^{k+1} = \frac{\gamma(x^k)_j b_j + \beta(x^k)_j a_j}{d(x^k)_j} \tag{6.4}$$

where, for any suitable x we define the vectors $\gamma(x)$, $\beta(x)$, $e(x)$, $f(x)$ and $d(x)$ as follows:

$$\gamma(x)_j = (x_j - a_j)e(x)_j \tag{6.5}$$

$$\beta(x)_j = (b_j - x_j)f(x)_j \tag{6.6}$$

$$e(x)_j = \sum_{i=1}^{I} P_{ij}\left(\frac{y_i - Pa_i}{Px_i - Pa_i}\right) \tag{6.7}$$

$$f(x)_j = \sum_{i=1}^{I} P_{ij}\left(\frac{Pb_i - y_i}{Pb_i - Px_i}\right) \tag{6.8}$$

and

$$d(x)_j = \gamma(x)_j + \beta(x)_j. \tag{6.9}$$

We shall assume throughout this section that x minimizes $KL(y-Pa, Px-Pa)+KL(Pb-y, Pb-Px)$ over all x satisfying the constraints.

As in the proof of convergence of the RBI-EMML algorithm [18, 19], it is helpful to consider doubly indexed arrays. For each $z = \{z_j\}$ with $a_j < z_j < b_j$, $j = 1, \ldots, J$ define, for $i = 1, \ldots, I$, $j = 1, \ldots, J$,

$$a(z)_{ij} = (z_j - a_j)P_{ij}\frac{y_i - Pa_i}{Pz_i - Pa_i} \tag{6.10}$$

$$b(z)_{ij} = (b_j - z_j)P_{ij}\frac{Pb_i - y_i}{Pb_i - Pz_i} \tag{6.11}$$

and

$$q(z)_{ij} = P_{ij}z_j. \tag{6.12}$$

Let $a(z)$, $b(z)$ and $q(z)$ be the arrays with entries $a(z)_{ij}$, $b(z)_{ij}$ and $q(z)_{ij}$, respectively. Then, for each $j = 1, \ldots, J$, we have

$$\sum_i a(z)_{ij} = \gamma(z)_j \tag{6.13}$$

$$\sum_i b(z)_{ij} = \beta(z)_j. \tag{6.14}$$

Now we define the distance function $G(u, z)$ for suitable u and z, that is, for u and z satisfying the constraints

$$G(u, z) = KL(a(z), q(u) - q(a)) + KL(b(z), q(b) - q(u)). \tag{6.15}$$

If we minimize $G(u, z)$ with respect to u we obtain the vector z' given by

$$z'_j = \frac{[b_j\gamma(z)_j + a_j\beta(z)_j]}{d(z)_j} \tag{6.16}$$

so we see that the minimizing u involves taking one step of our algorithm, starting at z. We make this more precise in the following proposition. The proof is a direct calculation.

Proposition 6.1. *For any suitable u , z and z′ given by (6.16) we have*

$$G(u, z) = G(z', z) + \delta(z, u) \tag{6.17}$$

where

$$\delta(z, u) = \sum_{j=1}^{J} w(z)_j[KL(z'_j - a_j, u_j - a_j) + KL(b_j - z'_j, b_j - u_j)] \tag{6.18}$$

and

$$w(z)_j = \frac{d(z)_j}{b_j - a_j}. \tag{6.19}$$

The following proposition, also proven by a direct calculation, relates $G(u, z)$ to our cost function $F(u) = KL(y - Pa, Pu - Pa) + KL(Pb - y, Pb - Pu)$.

Proposition 6.2. *For any suitable u and z we have* $G(u, u) = F(u)$ *and*

$$G(u, z) = G(u, u) + KL(a(z), a(u)) + KL(b(z), b(u)). \tag{6.20}$$

Our proof of convergence will follow almost immediately once we prove the double inequality (6.21) below. Note that it follows from (6.17) that $F(x^k) = G(x^k, x^k) = G(x^{k+1}, x^k) + \delta(x^k, x^k) \geqslant G(x^{k+1}, x^{k+1}) + \delta(x^k, x^k) = F(x^{k+1}) + \delta(x^k, x^k)$, so that the sequence $\{F(x^k)\}$ is decreasing and $\{\delta(x^k, x^k)\}$ converges to zero. Since $\{x^k\}$ is bounded, we again extract a subsequence converging to a cluster point x^*. Since $\delta(x^*, x^*) = 0$ it follows that $(x^*)' = x^*$; that is, x^* is a fixed point of the iteration. Similarly, we conclude that since $u = x$ minimizes $F(u)$, x also is a fixed point of the iteration.

Proposition 6.3. *For* $u = x$ *minimizing* $F(u)$ *over all* u *with* $a \leqslant u \leqslant b$, *we have the double inequality*

$$\delta(x, x^k) \geqslant KL(a(x), a(x^k)) + KL(b(x), b(x^k)) \geqslant \delta(x, x^{k+1}). \tag{6.21}$$

Proof. From (6.17) we have $G(x^k, x) = F(x) + \delta(x, x^k)$ and from (6.20) we have that $G(x^k, x) = F(x^k) + KL(a(x), a(x^k)) + KL(b(x), b(x^k))$. Since $F(x) \leqslant F(x^k)$, the first inequality in (6.21) follows. To prove the second one we begin by using proposition 3.1 above to conclude that

$$KL(a(x), a(x^k)) + KL(b(x), b(x^k)) \geqslant \sum_{j=1}^{J}\left[KL\left(\sum_{i=1}^{I} a(x)_{ij}, \sum_{i=1}^{I} a(x^k)_{ij}\right)\right.$$
$$\left.+KL\left(\sum_{i=1}^{I} b(x)_{ij}, \sum_{i=1}^{I} b(x^k)_{ij}\right)\right].$$

A calculation shows that

$$\sum_{j=1}^{J}\left[KL\left(\sum_{i=1}^{I}a(x)_{ij},\sum_{i=1}^{I}a(x^k)_{ij}\right)+KL\left(\sum_{i=1}^{I}b(x)_{ij},\sum_{i=1}^{I}b(x^k)_{ij}\right)\right]$$
$$=\delta(x,x^{k+1})+KL(d(x),d(x^k))$$

from which the second inequality follows. □

Note that it follows from (6.21) that the sequences$\{\delta(x,x^k)\}$ and $\{KL(a(x),a(x^k))+KL(b(x),b(x^k))\}$ are decreasing. Therefore $\delta(x,x^*)=KL(a(x),a(x^*))+KL(b(x),b(x^*))$ and is finite.

Proposition 6.4. *The vector x^* satisfies the constraints and $F(x)=F(x^*)$.*

Proof. We have $G(x^*,x)=F(x)+\delta(x,x^*)$ and $G(x^*,x)=F(x^*)+KL(a(x),a(x^*))+KL(b(x),b(x^*))$. From the comment just above we have $F(x)=F(x^*)$. □

Now we may use x^* in place of x to obtain that $\{\delta(x^*,x^k)\}$ is decreasing. But, since a subsequence of the $\{x^k\}$ converges to x^* the sequence $\{\delta(x^*,x^k)\}$ must converge to zero and therefore the whole sequence $\{x^k\}$ must converge to x^*. This completes the proof of convergence of ABEMML in the inconsistent case, for $N=1$.

7. Applications

The application of RBI-SMART and RBI-EMML to problems of medical image reconstruction has been discussed recently in the literature [27–30]. The ABMART and ABEMML methods enhance our ability to incorporate prior information in such reconstructions [26]. Simulations suggest that in emission tomography the speed and accuracy with which small 'hot' or 'cold' spots can be resolved may depend on the neighbouring background: the more contrast the better. We hope to use the vectors a and b to provide prior information about the background and the feasible deviations from that to be expected.

In applications in which there is insufficient data to specify with any precision the image to be reconstructed, such as limited-angle tomography or imaging from scattered field data, the use of prior information in the reconstruction is essential [31]. The ABEMML and ABMART should be helpful in such cases.

If $a_j=-\infty$ and $b_j=0$ for some j, while $a_j=0$ and $b_j=+\infty$ for the others, then we are saying that we know the signs of the x_j. There are applications in which such prior knowledge of the signs comes from the nature of the unknowns: in the emission tomography problem they are all intensities of radioactivity, hence non-negative.

8. Conclusions

We have extended the approach of Snyder *et al* for deblurring subject to non-negativity constraints to include more general constraints on the entries of the desired solution, while also permitting the blurring kernel to take on negative values. Our algorithms are generalizations of the EMML, MART and their rescaled block-iterative versions to include the constraints that the vector x we seek be found within a box in R^J, that is, $a_j\leqslant x_j\leqslant b_j$ for each j. If there are exact solutions of $y=Px$ satisfying these constraints, then the new algorithms ABMART and ABEMML converge to such solutions. When no such solutions exist the algorithms tell us that, either by failing to converge (in the case of more than one block we probably get a limit cycle) or by converging to an approximate solution (in the case of a single block of data).

Acknowledgments

This work was sponsored in part by the National Cancer Institute under grant CA-42165 (M King, principal investigator). Its contents are solely the responsibility of the author and do not necessarily represent the official views of the National Cancer Institute.

References

[1] Snyder D, Schulz T and O'Sullivan J 1992 Deblurring subject to non-negativity constraints *IEEE Trans. Sig. Proc.* **40** 1143–50

[2] Vardi Y and Lee D 1993 From image deblurring to optimal investments: maximum likelihood solutions for positive linear inverse problems *J. R. Stat. Soc.* B **55** 569–612

[3] Dempster A, Laird N and Rubin D 1977 Maximum likelihood from incomplete data via the EM algorithm *J. R. Stat. Soc.* B **39** 1–38

[4] Lindsay B G 1983 The geometry of mixture likelihoods: a general theory *Ann. Stat.* **11** 86–94

[5] Lindsay B G 1983 The geometry of mixture likelihoods part II: the exponential family *Ann. Stat.* **11** 783–92

[6] Lindsay B G and Roeder K 1993 Uniqueness of estimation and identifiability in mixture models *Can. J. Stat.* **21** 139–47

[7] Everitt B S and Hand D J 1981 Finite mixture distributions *Monographs on Applied Probability and Statistics* (New York: Chapman and Hall)

[8] Byrne C, Haughton D and Jiang T 1993 High resolution inversion of the Poisson and binomial transforms *Inverse Problems* **9** 39–56

[9] Vardi Y and Zhang C-H 1998 Estimating mixing probabilities subject to lower and upper bound constraints with applications *Technical Report 97-006* Department of Statistics, Rutgers University

[10] Shepp L and Vardi Y 1982 Maximum likelihood reconstruction for emission tomography *IEEE Trans. Med. Imaging* **MI-1** 113–22

[11] Lange K and Carson R 1984 EM reconstruction for emission and transmission tomography *J. Comput. Assist. Tomogr.* **8** 306–12

[12] Vardi Y, Shepp L and Kaufman L 1985 A statistical model for positron emission tomography *J. Am. Stat. Assoc.* **80** 8–37

[13] Lange K, Bahn M and Little R 1987 A theoretical study of some maximum likelihood algorithms for emission and transmission tomography *IEEE Trans. Med. Imaging* **MI-6** 106–14

[14] Byrne C L 1993 Iterative image reconstruction algorithms based on cross-entropy minimization *IEEE Trans. Image Process.* **IP-2** 96–103

[15] Byrne C L 1995 Erratum and addendum to 'Iterative image reconstruction algorithms based on cross-entropy minimization' *IEEE Trans. Image Process.* **IP-4** 225–6

[16] Byrne C L 1996 Iterative reconstruction algorithms based on cross-entropy minimization *Image Models (and their Speech Model Cousins) (IMA Volumes in Mathematics and its Applications 80)* ed S E Levinson and L Shepp (New York: Springer) pp 1–11

[17] Byrne C L 1996 Block-iterative methods for image reconstruction from projections *IEEE Trans. Image Process.* **IP-5** 792–4

[18] Byrne C L 1997 Convergent block-iterative algorithms for image reconstruction from inconsistent data *IEEE Trans. Image Process.* **IP-6** 1296–304

[19] Byrne C L 1998 Accelerating the EMML algorithm and related iterative algorithms by rescaled block-iterative (RBI) methods *IEEE Trans. Image Process.* **IP-7** 100–9

[20] Censor Y 1981 Row-action methods for huge and sparse systems and their applications *SIAM Rev.* **23** 444–64

[21] Censor Y 1988 Parallel application of block-iterative methods in medical imaging and radiation therapy *Math. Prog.* **42** 307–25

[22] Censor Y and Herman G T 1987 On some optimization techniques in image reconstruction from projections *Appl. Numer. Math.* **3** 365–91

[23] Censor Y and Segman J 1987 On block-iterative entropy minimization *J. Inf. Optim. Sci.* **8** 275–91

[24] Kullback S and Leibler R 1951 On information and sufficiency *Ann. Math. Stat.* **22(1)** 79–86

[25] Browne J and DePierro A 1996 A row-action alternative to the EM algorithm for maximizing likelihoods in emission tomography *IEEE Trans. Med. Imaging* **15** 687–99

[26] Narayanan M, Byrne C and King M 1998 AB-EMML—An iterative reconstruction algorithm incorporating upper and lower bounds, with application to SPECT transmission imaging, submitted

[27] Lalush D, Karimi S and Tsui B 1996 Convergence and resolution recovery of block-iterative EM algorithms modeling 3D detector response in SPECT *Conf. Record, IEEE Med. Imag. Conf. (Anaheim, CA)* pp 1618–22

[28] Xia W, Glick S, Pan T-S, Soares E and Luo D-S 1996 Region of interest evaluation of SPECT image reconstruction methods using a realistic brain phantom *Conf. Record, IEEE Med. Imag. Conf. (Anaheim, CA)* pp 1547–8

[29] Byrne C, Soares E, Pan T-S, Glick S and King M 1996 Accelerating the EM algorithm using rescaled block-iterative methods *Conf. Record, IEEE Med. Imag. Conf. (Anaheim, CA)* pp 1752–6

[30] Lalush D and Tsui B 1997 Block-iterative techniques for fast 4D reconstruction using *a priori* motion models in gated cardiac SPECT *Proc. 1997 Int. Meeting on Fully 3D Image Reconstruction (Pittsburgh, PA)*

[31] Pan T-S, Tsui B and Byrne C 1997 Choice of initial conditions in the ML reconstruction for transmission CT with truncated projection data *IEEE Trans. Med. Imaging* **MI-16** 426–38