Charles L. Byrne
Department of Mathematical Sciences
University of Massachusetts Lowell
March 1, 2017

The EM Algorithm: Theory, Applications and Related Methods

Contents

Preface

We shall not cease from exploration, and the end of all our exploring will be to arrive where we started and know the place for the first time.

T. S. Eliot

I have been trying to understand the EM algorithm for twenty-five years. I first encountered the EM algorithm in 1990, when I began collaborating with Mike King and members of his research group in the Department of Radiology, University of Massachusetts Medical School. They were interested in the application of the EM to image reconstruction in emission tomography. The particular case of the EM algorithm that they showed me was what I shall call here the EMML algorithm, also called the MLEM algorithm, which was derived from the Poisson statistics of emission tomography. Every time I visited UMassMed Bill Penney would give me a stack of papers to further my education. This is how I learned of the work of Vardi, Shepp, Kaufman, Fessler, Lange, Csiszár, and others [66, 68, 37, 53, 70, 54, 49, 1].

Around the same time, I became aware of the work of Gabor Herman and his colleagues and the ART, MART, and simultaneous MART (SMART) algorithms [45, 29]. Their approach to medical image reconstruction was more linear-algebraic than statistical. I recall an exchange of comments on the paper [70] in which Herman et al. suggested that the EMML algorithm might be usefully viewed in their linear-algebraic terms, a suggestion that met with vehement denial from the original authors. After I published [10], in which I showed a close connection between the EMML and the SMART, I was invited to speak to Herman's group at MIPG in Philadelphia. There I met Yair Censor and Paul Eggermont, with whom I have been collaborating ever since.

From the first, I was interested in the interplay between the statistical and the linear-algebraic approaches to image reconstruction. Through my study of the EMML and my collaboration with Censor I started to learn something about optimization and the role it could play in the reconstruction problem. Indeed, this paper continues that quest to understand this interplay.

Each time I wrote about the EMML algorithm and the more general

EM algorithm I felt that, although I was telling the truth, it was probably not the whole truth; I felt that there was always more to know about these methods and open questions to be answered. I was also bothered by what I perceived to be inadequacies in the standard treatment of the EM algorithm. In particular, the usual proof that any EM algorithm increases likelihood at each step is flawed. Because there is no all-encompassing proof of convergence for the EM algorithm, each algorithm in this class must be dealt with individually. A good illustration of this is found in the series of papers on the behavior of the EM algorithm for emission tomography [66, 68, 53, 70, 54, 10, 17]. I am sure that those who use the EM algorithms frequently have learned to live with this overall lack of rigor, and have focused on the particular EM algorithm they require. Nevertheless, it is of some interest to see how one might go about fixing the flaw. The current paper is just the latest in a series of attempts to understand what is going on.

As several authors have noted, the EM algorithm is not truly an algorithm, but a template or recipe for designing iterative algorithms. Nevertheless, I shall stick with tradition and refer here to "the EM algorithm". I discovered recently [26] that we can look at the EM algorithm from a nonstochastic perspective, which I call the "nonstochastic EM"(NSEM) template. Using the NSEM template we can derive an alternative to the usual EM template that I call the "statistical EM"(STEM) template, to distinguish it from the NSEM. I prove that any STEM iteration increases likelihood, and that most EM algorithms are in fact STEM algorithms.

It is helpful to view the NSEM and STEM as members of broader classes of templates, or recipes for iterative algorithms. The most inclusive of these are "auxiliary-function"(AF) algorithms. Contained within the AF class are three subclasses, "alternating minimization"(AM), "proximal minimization"algorithms (PMA), and "majorization minimization"(MM) methods, also known as "optimization transfer"or "surrogate-function"methods, in statistics. Each of these three subclasses has its own literature, but, as we shall show, all the algorithms in these three classes are equivalent to one another.

The EM algorithm is a particular case of PMA. Because the usual presentation of the EM algorithm involves conditional expectations, not a subject familiar to many of my students, I approach the EM algorithm through a more general nonstochastic EM algorithm (NSEM) that we can then use to derive the stochastic EM (STEM). The STEM avoids some difficulties with the traditional approach to the EM, but is equivalent in most cases. By deriving the STEM from the NSEM template we get that likelihood is increasing. But we want more than that; we want the likelihood to increase to its maximum value.

Auxiliary-function algorithms are formulated as minimization algorithms and it is guaranteed that the objective function is decreasing. The

SUMMA and the more general SUMMA2 algorithms are AF methods for which it is guaranteed that the iterative sequence of values of the objective function actually converges to the infimum of its values. Those STEM algorithms that can be reformulated as SUMMA2 algorithms are therefore guaranteed to maximize likelihood; the EMML is one such example. By deriving the STEM algorithms as NSEM algorithms we link STEM with other related optimization methods, including those for entropy maximization.

In Chapter 1 we consider some examples of problems that we shall solve using the STEM; we shall see more examples later. In Chapter 2 we define the EM algorithm, sketch the development of the algorithm for the case of discrete probabilities, and point out some difficulties we encounter when we apply the algorithm to probability density functions. In Chapter 3 we show how these difficulties can be avoided using the NSEM and STEM. In Chapter 4 we present several examples of applications of the EM algorithm.

Starting in Chapter 5 we place the EM algorithm within the broader context of PMA. We first consider the AF template and demonstrate the equivalence of its subclasses, AM, PMA and MM. The subclasses SUMMA and SUMMA2 of AF are discussed in Chapter 6. In a wide variety of applications the goal is to find an approximate or exact solution to a system of linear equations, often with certain constraints imposed. The remaining chapters deal with various aspects of the use of iterative algorithms to solve these problems.

Chapter 1

Introduction

In this introductory chapter we present three applications of the EM algorithm, first to a simple example involving marbles in bowls, and then to the reconstruction of images in single-photon emission tomography (SPECT) and list-mode positron emission tomography (List-mode PET). All three of these examples involve probabilistic mixtures. The version of the EM that we shall employ here is the standard version, which is suitable for the problems in this chapter, but, as we shall discuss later, not always suitable for other problems.

As has been pointed out many times, the EM algorithm is not a single algorithm but rather a template or framework for designing iterative methods for likelihood maximization. However, in keeping with tradition, we shall refer here to *the* EM algorithm. There is no general theory governing the behavior of EM algorithms. Consequently, features of individual EM algorithms are sometimes mistakenly attributed to all EM algorithms. In this chapter we itemize some of the myths and some of the truths, leaving details for later chapters.

1.1 Probabilistic Mixtures

The following example is simple, yet sufficient to illustrate many aspects of remote sensing. Imagine a box containing many slips of paper, on each of which is written one of the numbers $j = 1, 2, ..., J$. We have no access to the box. There are also J bowls of colored marbles. The colors of the marbles are denoted $i = 1, 2, ..., I$. We are allowed to examine the contents of each bowl, so we know precisely the probability $P_{i,j}$ that a marble with the color i will be drawn from bowl j. Out of my sight someone draws a slip of paper from the box and without saying anything extracts a marble from the indicated bowl. The color of the drawn marble is announced. This process happens N times, at the end of which I have a list $\mathbf{i} \doteq (i_1, i_2, ..., i_N)$, where i_n is the index of the color of the nth marble drawn. On the basis of this data and the probabilities $P_{i,j}$ I must estimate, for each j, the number θ_j, the proportion of slips of paper on which the number j is written, which is then also the probability of drawing a slip with the number j printed on it.

Let $f(i|\theta)$ be the probability that the color of the drawn marble is i, given θ. Then, for each i, we have

$$f(i|\theta) = \sum_{j=1}^{J} P_{i,j}\theta_j = (P\theta)_i, \tag{1.1}$$

where P is the matrix with entries $P_{i,j}$ and θ is the column vector with entries θ_j. This is a discrete probabilistic mixture, with parameter vector θ to be estimated.

Given the data $i_1, i_2, ..., i_N$, the likelihood function is

$$L(\theta) = \prod_{n=1}^{N} (P\theta)_{i_n},$$

and the log likelihood function is

$$LL(\theta) = \sum_{n=1}^{N} \log(P\theta)_{i_n}.$$

With N_i the number of times i appears on the list, that is, the number of indices n such that $i = i_n$, we can write

$$LL(\theta) = \sum_{i=1}^{I} N_i \log(P\theta)_i. \tag{1.2}$$

Maximizing $LL(\theta)$ over nonnegative vectors θ whose entries sum to one is equivalent to minimizing the Kullback–Leibler (KL) distance

$$KL(\alpha, P\theta) = \sum_{i=1}^{I} \alpha_i \log\left(\frac{\alpha_i}{(P\theta)_i}\right) + (P\theta)_i - \alpha_i, \tag{1.3}$$

where α is the column vector with entries $\alpha_i = N_i/N$. The KL distance will be defined and discussed in Chapter 2.

To employ the EM we postulate as the preferred data the list $\mathbf{j} = (j_1, j_2, ..., j_N)$, where j_n is the index of the bowl from which the nth marble was drawn. We prefer this data because, if we had the j_n, then our estimate of θ_j would simply be N_j/N, where N_j is the number of times the index j appears in the list. The log likelihood function for this preferred data is

$$LL_p(\theta) = \sum_{n=1}^{N} \log \theta_{j_n} = \sum_{j=1}^{J} N_j \log \theta_j. \tag{1.4}$$

Note that the observed data is not a function of the preferred data, so the relationship $\mathbf{i} = h(\mathbf{j})$ does not hold here. The probability of obtaining the list $\mathbf{i}$, given θ and the list $\mathbf{j}$, is $\prod_{n=1}^{N} P_{i_n, j_n}$, which is independent of θ; that is, this preferred data is *acceptable*, a term we shall discuss in detail later.

Denote by $f(\mathbf{j}|\theta)$ the probability of obtaining the list $\mathbf{j}$, given θ. In this particular instance there is an easy way to proceed; we can calculate the conditional expected value of $\log f(\mathbf{j}|\theta)$ directly. For each n and j let $X_{n,j}$ have the value one if $j_n = j$, and zero, otherwise. Then

$$\log f(\mathbf{j}|\theta) = \sum_{n=1}^{N} \sum_{j=1}^{J} X_{n,j} \log \theta_j.$$

Using

$$E(X_{n,j}|\mathbf{i}, \theta^{k-1}) = \theta_j^{k-1} \frac{P_{i_n,j}}{(P\theta^{k-1})_{i_n}}, \tag{1.5}$$

we have

$$E(\log f(\mathbf{j}|\theta)|\mathbf{i}, \theta^{k-1}) = \sum_{i=1}^{I} \sum_{j=1}^{J} \theta_j^{k-1} P_{i,j} \frac{N_i}{(P\theta^{k-1})_i} \log \theta_j. \tag{1.6}$$

Maximizing this with respect to θ, we have

$$\theta_j^k = \theta_j^{k-1} \sum_{i=1}^{I} P_{i,j} \frac{\alpha_i}{(P\theta^{k-1})_i}. \tag{1.7}$$

This iterative algorithm is well known and occurs as the EM algorithm in single-photon emission tomography (SPECT), as we shall see next. In that context it is often called the EMML algorithm, and we shall use that terminology here.

Note that the value of each $X_{n,j}$ is either zero or one so that, if we had the true values of the $X_{n,j}$, then we could determine the j_n as that value of j for which $X_{n,j} = 1$. In Equation (1.5) we estimate $X_{n,j}$ using its conditional expectation. However, this estimate does not have the values zero or one, so cannot be used directly to determine j_n. In fact, the preferred data is not estimated in this example.

1.2 SPECT

In single-photon emission tomography (SPECT) a radionuclide is injected into the body of the patient. Photons emitted by the radioactive decay are then detected by gamma cameras located outside the body of the patient. Typically, we discretize the body into a finite number of pixels (or voxels for three-dimensional processing), indexed by $j = 1, ..., J$. We want to estimate, for each j, the probability that a detected photon came from pixel (or voxel) j. We denote this probability by θ_j. We assume that θ_j is proportional to the relative concentration of radionuclide present within the jth pixel. Our estimates of the θ_j then form the image given to the doctor.

The detectors are numbered $i = 1, 2, ..., I$. We have as our observed data the list $i_1, i_2, ..., i_N$, where i_n denotes the detector at which the nth detection was made. What we wish we had, the preferred data, is the list $j_1, j_2, ..., j_N$, where j_n denotes the pixel from which the nth detected photon was emitted. As in the bowl example, we have a discrete probabilistic mixture. The probability of a detection at detector i, given $\theta = (\theta_1, \theta_2, ..., \theta_J)^T$, is

$$f(i|\theta) = \sum_{j=1}^{J} P_{i,j}\theta_j = (P\theta)_i, \tag{1.8}$$

where $P_{i,j}$ is the probability that a photon emitted from pixel j will be detected at detector i. We assume that these $P_{i,j}$ are known to us. Mathematically speaking, this problem is identical to the bowls problem. With N_i the number of photons detected at the ith detector, the iteration in Equation (1.7) solves the estimation problem. As we shall see later, the sequence $\{\theta^k\}$ converges to a maximizer of the likelihood.

Because this formulation of the SPECT problem is completely analogous to the bowls problem, here too the preferred data is not estimated.

This formulation of the SPECT problem is not the usual formulation. More commonly, one assumes that the random variables $Y_{i,j}$ are the number of photons emitted from j and detected at i, and that we have single realizations of the random variables $Y_i = \sum_{j=1}^{J} Y_{i,j}$. The $Y_{i,j}$ are assumed to be independent and $Y_{i,j}$ is $P_{i,j}\lambda_j$-Poisson, where λ_j is the expected number of photons emitted at the jth pixel during the scan. Then the Y_i are independent and Y_i is $(P\lambda)_i$-Poisson, where $\lambda \doteq (\lambda_1, ..., \lambda_J)^T$. The likelihood function for the observed data $y = (N_1, ..., N_I)$ is

$$f_Y(y|\lambda) = \prod_{i=1}^{I} e^{-(P\lambda)_i} (P\lambda)_i^{N_i} / N_i!. \tag{1.9}$$

Maximizing the likelihood function is equivalent to minimizing the Kullback–Leibler distance $KL(y, P\lambda)$. The EM iteration is once again that given in Equation (1.7).

Note that, in this second formulation of the SPECT problem, we do estimate the preferred data at each step and then use these estimates to obtain the next estimate of the vector λ.

Some of the early papers on the application of the EM algorithm to the SPECT problem credited desirable properties exhibited by the EMML algorithm to the use of the Poisson statistics and greater adherence to the actual physics of the situation. However, as we just saw, the same EMML algorithm can be derived for the SPECT problem simply by treating it as a probabilistic mixture problem, without assuming any Poisson statistics. The images produced by the EMML algorithm are not always good images and the convergence can be slow. What we can say is that the behavior of the EMML algorithm is unrelated to the use of Poisson statistics.

It is a mistake to give most of the credit for the behavior of an algorithm to the philosophical views that prompted its use. This happened in the 1980's when entropy maximization became increasingly popular for image reconstruction, and again with likelihood maximization. In [70] the EMML algorithm is discussed in terms of the statistics of the Poisson model for emission and it is suggested that its usefulness for tomographic image reconstruction lies in its explicit use of the physics inherent in the emission process. This article was published in a journal that invited and published commentary on the article from interested parties. Among those invited to comment were Gabor Herman and members of his group. In their work tomographic reconstruction had been treated as a linear algebra problem and their algorithms were iterative methods for solving large systems of linear equations and linear inequalities with constraints [45]. In their comments they offered the view that the EMML algorithm may well be viewed in linear algebraic terms. In rebuttal, the authors of the original paper asserted quite strongly that there was no connection between their statistical approach and that of Herman and his colleagues, claiming that likelihood maximization was a well studied part of statistical estimation theory and unrelated to solving linear equations.

In [12] I rederived the EMML algorithm in tandem with the *simultaneous multiplicative algebraic reconstruction technique* (SMART), a method developed by Herman's group and based on their linear-algebraic formulation, showing that these algorithms were closely related and that both algorithms could be viewed simply as iterative methods for solving systems of linear equations. This tandem development will be presented later in these notes. Whatever properties they may exhibit could not be attributed to adherence to the physics, nor to the theory of statistical likelihood maximization.

1.3 List-Mode PET

In positron-emission tomography a positron is emitted at some pixel (or voxel) and immediately encounters an electron. Their masses are annihilated and two gamma-ray photons head off in nearly opposite directions, along some line segment. When the detectors record two detections simultaneously, it is inferred that an emission occurred somewhere along the line segment, called the *line of response* (LOR), determined by the sites of the two detections. As the scanning proceeds, a list of the LOR involved is kept. It is convenient to assume that the collection of potential LOR forms a continuum, and that the probability that an LOR denoted by the variable v is on the list is given by

$$f(v|\theta) = \sum_{j=1}^{J} \theta_j f_j(v), \tag{1.10}$$

where θ_j is the probability that an emitted positron is emitted at pixel j and $f_j(v)$ is the pdf governing the distribution of the LOR resulting from emissions at the jth pixel. We have a probabilistic mixture once again, but it is not quite the same as in the previous subsection, since probability density functions are now involved.

We assume that our observed data is the vector $y = (v_1, v_2, ..., v_N)$, where v_n denotes the nth LOR on the list. As in the discrete case, we take as the preferred data the vector $x = (j_1, j_2, ..., j_N)$, where j_n denotes the pixel at which the nth positron was emitted. We do not have the relationship $Y = h(X)$. However, the preferred data is acceptable. Alternatively, we can take $w = (j_1, j_2, ..., j_N)$ and $x = (y, w)$, and use the missing-data model. Now we do have $Y = h(X)$. The W is acceptable, but that doesn't matter in the missing-data model.

It is shown in [17] that maximizing the likelihood in this case is equivalent to minimizing

$$F(\theta) = KL(u, P\theta) + \sum_{j=1}^{J} (1 - s_j)\theta_j,$$

over probability vectors θ, where P is the matrix with entries $P_{n,j} = f_j(v_n)$, $s_j = \sum_{n=1}^{N} f_j(v_n)$ and u is the vector whose entries are all $u_n = 1/N$. Since we are dealing with probability density functions now, the s_j can take on any positive value and $1 - s_j$ can be negative. It is easily shown that, if $\hat{\theta}$ minimizes $F(\theta)$ over all nonnegative vectors θ, then $\hat{\theta}$ is a probability vector. Therefore, we can obtain the maximum likelihood estimate of θ by minimizing $F(\theta)$ over nonnegative vectors θ.

The iterative step of the EM is now

$$\theta_j^k = \frac{1}{N}\theta_j^{k-1}\sum_{n=1}^{N}\frac{f_j(v_n)}{f(v_n|\theta^{k-1})}. \tag{1.11}$$

In previous articles this iterative algorithm was called the Mix-EM algorithm. As we shall discuss later, since the preferred data X is acceptable, likelihood is increasing for this algorithm. We shall go further now, and show that the sequence of probability vectors $\{\theta^k\}$ converges to a maximizer of the likelihood. The following theorem is found in [17].

Theorem 1.1 *Let u be any positive vector, P any nonnegative matrix with $s_j > 0$ for each j, and*

$$F(\theta) = KL(u, P\theta) + \sum_{j=1}^{J}\delta_j KL(\gamma_j, \theta_j).$$

If $s_j + \delta_j > 0$, $\alpha_j = s_j/(s_j + \delta_j)$, and $\delta_j\gamma_j \geq 0$, for all j, then the iterative sequence given by

$$\theta_j^{k+1} = \alpha_j s_j^{-1}\theta_j^k\Big(\sum_{n=1}^{N} P_{n,j}\frac{u_n}{(P\theta^k)_n}\Big) + (1-\alpha_j)\gamma_j \tag{1.12}$$

converges to a nonnegative minimizer of $F(\theta)$.

With the choices $u_n = 1/N$, $\gamma_j = 0$, and $\delta_j = 1 - s_j$, the iteration in Equation (1.12) becomes that of the Mix-EM algorithm. Therefore, the sequence $\{\theta^k\}$ converges to the maximum likelihood estimate of the mixing proportions.

1.4 The EMML and Gradient Descent

Previously, we saw that maximizing the likelihood in SPECT is equivalent to minimizing the function $f(\lambda) = KL(y, P\lambda)$ over all nonnegative λ.

It is interesting to place the EMML algorithm within the broader context of gradient-descent methods for minimization.

Let $f : \mathbb{R}^J \to \mathbb{R}$ be differentiable, with gradient ∇f. The goal is to minimize $f(x)$. A gradient descent algorithm (GDA) has the iterative step

$$x^k = x^{k-1} - \gamma_k \nabla f(x^{k-1}), \tag{1.13}$$

where the step-size parameter $\gamma_k > 0$ is chosen at each step to force $f(x^k) \leq f(x^{k-1})$.

1.4.1 Generalized Gradient Descent

A somewhat more general version of GDA, denoted GGDA, has the iterative step

$$x_j^k = x_j^{k-1} - \gamma_{k,j} \nabla f(x^{k-1})_j, \tag{1.14}$$

where now the step-size parameters are allowed to depend on j as well as on k. This would be helpful if we want to incorporate constraints such as nonnegativity. However, it is probably difficult, in general, to determine such parameters that will also guarantee that $f(x^k) \leq f(x^{k-1})$. As we shall see shortly, the EMML algorithm does achieve this dual objective.

1.4.2 The EMML Algorithm Revisited

The EMML algorithm minimizes the function $f(x) = KL(y, Px)$ over $x \geq 0$. We assume that $\sum_{i=1}^I P_{i,j} = 1$, for all j. The gradient of f has the entries

$$\nabla f(x)_j = \sum_{i=1}^I P_{i,j} \left(1 - \frac{y_i}{(Px)_i}\right) = 1 - \sum_{i=1}^I P_{i,j} \frac{y_i}{(Px)_i}. \tag{1.15}$$

Therefore, the GDA in this case is

$$x_j^k = x_j^{k-1} - \gamma_k \left(1 - \sum_{i=1}^I P_{i,j} \frac{y_i}{(Px^{k-1})_i}\right), \tag{1.16}$$

and the GGDA is

$$x_j^k = x_j^{k-1} - \gamma_{k,j} \left(1 - \sum_{i=1}^I P_{i,j} \frac{y_i}{(Px^{k-1})_i}\right). \tag{1.17}$$

Suppose now that we select $\gamma_{k,j} = x_j^{k-1}$. The GGDA is then

$$x_j^k = x_j^{k-1} \sum_{i=1}^I P_{i,j} \frac{y_i}{(Px^{k-1})_i} \tag{1.18}$$

which is the EMML algorithm. By selecting $\gamma_{k,j} = x_j^{k-1}$ we make the change in the jth entry small where the current value is already small, preventing the next value from becoming negative. When the current entry is not small, we allow the change to be greater. The EMML iterates are always positive vectors and the sequence $\{f(x^k)\}$ is decreasing as well.

1.5 Myths and Truths

In this section we repeat the traditional description of the EM algorithm and then itemize some myths and truths concerning the EM.

1.5.1 The Traditional Description

We assume that Y is a random vector-valued variable governed by the probability density function (pdf) or the discrete probability function (pf) $f_Y(y|\theta_{true})$, where θ_{true} is a member of the parameter space Θ. We have one realization y of Y and want to estimate θ_{true} by maximizing the likelihood function $f_Y(y|\theta)$ over $\theta \in \Theta$. In cases in which the EM algorithm is useful maximizing $f_Y(y|\theta)$ is difficult and requires iteration. The basis for the EM algorithm is the use of a second random vector-valued variable X, governed by the pdf or pf $f_X(x|\theta)$ and related to Y in some fashion, such that, if we had an instance x of X, maximizing $f_X(x|\theta)$ would be computationally simpler.

In [56] McLachlan and Krishnan introduce the EM algorithm by saying "The situations where the EM algorithm can be profitably applied can be described as *incomplete-data problems*, where ML estimation is made difficult by the absence of some part of data in a more familiar and simpler data structure." They go on to note, however, that the situations to which the EM algorithm can be used "include not only evidently incomplete-data situations, where there are missing data, truncated distributions, censored or grouped observations, but a whole variety of situations where the incompleteness of the data is not all that natural or evident." In other words, the so-called *complete data* X is often chosen just for convenience, not because the so-called *incomplete data* Y is actually incomplete in any obvious way. Clearly, the terms *incomplete data* and *complete data* can be misleading. We shall call Y the *observed* or *actual* data, and X the *preferred* or *virtual* data.

One might reasonably assume now that, at each step of the EM algorithm, we would use y and the current θ^k to calculate x^{k+1}, and then would maximize $f_X(x^{k+1}|\theta)$ to get θ^{k+1}. However, this is not quite what happens, generally. Since we want to maximize $f_X(x|\theta)$, or, equivalently, maximize $\log f_X(x|\theta)$, we first estimate $\log f_X(x|\theta)$ by calculating the conditional expected value of the random variable $\log f_X(X|\theta)$ using

$$E(\log f_X(X|\theta)|y, \theta^k) = \int f_{X|Y}(x|y, \theta^k) \log f_X(x|\theta) dx; \tag{1.19}$$

this is the so-called E-step of the algorithm. Then we maximize this conditional expected value as a function of θ to get the next iterate θ^{k+1}; this is the so-called M-step.

1.5.2 Some Myths and Truths

Because there is no rigorous general theory for the EM algorithm certain beliefs concerning the EM are the result of features observed in particular cases falsely attributed to all EM algorithms. Here are some of the myths about the EM algorithm.

- The preferred (or complete or virtual) data is always explicitly estimated at each step. This is false. It is done in the example in Section 4.2, but not in the example in Section 1.1. In the case of SPECT image reconstruction (Section 1.2) it is not done when the problem is presented as a probabilistic mixture, but is done when we assume the Poisson model. This means that, in situations in which the EM algorithm is to be used, not just to maximize likelihood, but to fill in missing data, additional calculation may be needed.

- There is always a function h such that $Y = h(X)$. This is false (see Section 1.1). We mentioned previously that the X is related to the Y in some fashion. It is common in the EM literature to assume that Y is a deterministic function of X; that is, there is a function h such that $Y = h(X)$. This is not always the case, as some of our examples in these notes demonstrate. In fact, when f_Y and f_X are pdf, the restriction to cases in which $Y = h(X)$ presents theoretical difficulties and should be replaced with another condition, called *acceptability* (see Section 3.7).

- It has been rigorously demonstrated that likelihood is always nondecreasing at each step of the iteration. This is false (see Section 2.5). The "proof" given in several places in the literature [39, 56] is flawed for the case in which f_Y and f_X are pdf.

- The sequence of iterates $\{\theta^k\}$ always converges to the maximum-likelihood estimate θ_{ML}. This is false (see Section 3.9). Obviously, for the sequence to converge some topology on Θ is necessary. Even then, convergence need not occur. Convergence must be demonstrated in each particular case.

- When the sequence $\{\theta^k\}$ converges the limit is the ML estimate. This is false (again, see Section 2.5). That the limit is the ML estimate must be demonstrated in each case.

Some things are true, nevertheless.

- Suppose that, instead of maximizing the integral in Equation (1.19), we maximize

  $$\int f_{X,Y}(x,y|\theta^k)\log f_{X,Y}(x,y|\theta)dx \tag{1.20}$$

  to obtain θ^{k+1}. Then likelihood is always non-decreasing.

- The virtual data X is said to be *acceptable* if the conditional pdf or pf $f_{Y|X}(y|x)$ is independent of the parameter θ.

- Whenever X is acceptable, maximizing the integral in Equation (1.19) is equivalent to maximizing the integral in Equation (1.20).

- In the discrete case of f_Y and f_X probability functions and $Y = h(X)$, maximizing the integral in Equation (1.19) is equivalent to maximizing the integral in Equation (1.20), so likelihood is always non-decreasing.

Chapter 2

The Expectation Maximization Maximum Likelihood Algorithm

2.1 Definition and Basic Properties

In this chapter we define the expectation maximization maximum likelihood (EM) algorithm, and present the standard proof, valid for the discrete case of finite or countably infinite probability functions, that likelihood is increasing. We then consider certain difficulties that arise when we attempt to extend the EM algorithm to the case of probability density functions.

2.2 What is the EM Algorithm?

In applications of the EM algorithm in statistics, Y is a random vector taking values in $\mathbb{R}^M$ and governed by the probability density function (pdf) or probability function (pf) $f_Y(y|\theta_{\text{true}})$. The θ_{true} is a parameter, or vector of parameters, to be estimated; the set Θ is the collection of all potential values of θ_{true}. We have one realization, y, of Y, and we will estimate θ_{true} by maximizing the likelihood function of θ, given by $L(\theta) = f_Y(y|\theta)$, over $\theta \in \Theta$, to get θ_{ML}, a maximum-likelihood estimate of θ_{true}.

In the EM approach it is postulated that there is a second random vector, X, taking values in $\mathbb{R}^N$, such that, had we obtained an instance x of X, maximizing the function $L_x(\theta) = f_X(x|\theta)$ would have been computationally simpler than maximizing $L(\theta) = f_Y(y|\theta)$. Clearly, maximizing $L_x(\theta)$ is equivalent to maximizing $LL_x(\theta) = \log f_X(x|\theta)$. In most discussions of the

EM algorithm the vector y is called the "incomplete" data, while the x is the "complete" data and the situation is described by saying that there is "missing" data. In many applications of the EM algorithm this is suitable terminology. However, any data that we do not have but wish that we did have can be called "missing". I will call the vector y the "observed" data and the x the "preferred" data.

It would be reasonable to estimate x, using the current estimate θ^{k-1} and the data y, and then to use this estimate of x to get the next estimate θ^k. Since it is $LL_x(\theta)$ that we want to maximize, we estimate $\log f_X(x|\theta)$, rather than x itself. The EM algorithm estimates $LL_x(\theta)$ as

$$E(\log f_X(X|\theta)|y, \theta^{k-1}) = \int f_{X|Y}(x|y, \theta^{k-1}) \log f_X(x|\theta) dx, \tag{2.1}$$

the conditional expected value of the random function $\log f_X(X|\theta)$, conditioned on the data y and the current estimate θ^{k-1}. This is the so-called E-step of the EM algorithm. It is convenient to define

$$Q(\theta|\theta^{k-1}) \doteq \int f_{X|Y}(x|y, \theta^{k-1}) \log f_X(x|\theta) dx. \tag{2.2}$$

The M-step is to maximize $Q(\theta|\theta^{k-1})$ to get θ^k. For the case of probability functions we replace the integral with summation.

An EM algorithm generates a sequence $\{\theta^k\}$ of estimates of θ_{true}. There are several objectives that we may consider:

1. the sequence $\{L(\theta^k)\}$ should be increasing;
2. the sequence $\{L(\theta^k)\}$ should converge to $L(\theta_{ML})$;
3. the sequence $\{\theta^k\}$ should converge to θ_{ML}.

In these notes we shall focus primarily on the first two objectives. Clearly, in order to achieve the third objective it is necessary to have a topology on the set Θ of potential parameter values. There are no general results concerning the third objective, which must be handled on a case-by-case basis.

2.3 The Kullback–Leibler or Cross-Entropy Distance

The Kullback–Leibler distance is quite useful in the discussions that follow. For positive numbers s and t, the Kullback–Leibler distance from s to t is

$$KL(s, t) = s \log \frac{s}{t} + t - s. \tag{2.3}$$

Since, for $x > 0$ we have

$$x - 1 - \log x \geq 0$$

and equal to zero if and only if $x = 1$, it follows that

$$KL(s,t) \geq 0,$$

and $KL(s,s) = 0$. We use limits to define $KL(0,t) = t$ and $KL(s,0) = +\infty$. Now we extend the KL distance to nonnegative vectors component-wise. The following lemma is easy to prove.

Lemma 2.1 *For any nonnegative vectors x and z, with $z_+ = \sum_{j=1}^{J} z_j > 0$, we have*

$$KL(x,z) = KL(x_+, z_+) + KL(x, \frac{x_+}{z_+} z). \tag{2.4}$$

We can extend the KL distance in the obvious way to infinite sequences with nonnegative terms, as well as to nonnegative functions of continuous variables.

2.4 The Discrete Case

We assume now that our actual data is y, one realization of Y, a discrete random vector taking values in some finite or countably infinite set A and governed by the probability function $f_Y(y|\theta_{\text{true}})$. It may seem odd that we assume that we have only a single realization of Y, since, in most statistical estimation problems we have many independent samples, say $z_1, ..., z_N$, of a single random variable Z. Note, however, that we can then define $y = (z_1, ..., z_N)^T$ as a single realization of the random vector $Y = (Z_1, ..., Z_N)^T$, where the Z_n are independent and with the same distribution as Z.

We postulate a second random vector X taking values in a finite or countably infinite set B and a function $h : B \to A$, such that $Y = h(X)$. Then

$$f_Y(y|\theta) = \sum_{x \in h^{-1}\{y\}} f_X(x|\theta), \tag{2.5}$$

where

$$h^{-1}\{y\} = \{x | h(x) = y\}.$$

Consequently,

$$f_{X|Y}(x|y,\theta) = \begin{cases} f_X(x|\theta)/f_Y(y|\theta), & \text{if } x \in h^{-1}\{y\}; \\ 0, & \text{if } x \notin h^{-1}\{y\}. \end{cases} \tag{2.6}$$

Let $c(x) = \chi_{h^{-1}(y)}(x)$ have the value one, for $x \in h^{-1}(y)$ and zero, otherwise. Then

$$f_X(x|\theta)c(x) = f_{X|Y}(x|y,\theta)f_Y(y|\theta). \tag{2.7}$$

To get θ^k we maximize

$$\sum_x f_{X|Y}(x|y,\theta^{k-1}) \log f_X(x|\theta). \tag{2.8}$$

We show now that the sequence $\{f_Y(y|\theta^k)\}$ is increasing.

If x is not in $h^{-1}(y)$ then the term $f_{X|Y}(x|y,\theta^{k-1}) \log f_X(x|\theta)$ is either zero times a finite number, or is zero times the log of zero, which, by taking limits, we equate to zero as well. Therefore, we can replace the sum in Equation (2.8) with

$$\sum_{x \in h^{-1}(y)} f_{X|Y}(x|y,\theta^{k-1}) \log f_X(x|\theta). \tag{2.9}$$

Use Equation (2.7) to get

$$\log f_X(x|\theta) = \log f_{X|Y}(x|y,\theta) + \log f_Y(y|\theta), \tag{2.10}$$

for $x \in h^{-1}(y)$. Then maximizing

$$\sum_{x \in h^{-1}(y)} f_{X|Y}(x|y,\theta^{k-1}) \log f_X(x|\theta)$$

is equivalent to maximizing

$$\sum_x f_{X|Y}(x|y,\theta^{k-1}) \log f_{X|Y}(x|y,\theta) + \log f_Y(y|\theta).$$

Since

$$\sum_x f_{X|Y}(x|y,\theta^{k-1}) \log f_{X|Y}(x|y,\theta^k) + \log f_Y(y|\theta^k) \geq$$

$$\sum_x f_{X|Y}(x|y,\theta^{k-1}) \log f_{X|Y}(x|y,\theta^{k-1}) + \log f_Y(y|\theta^{k-1}),$$

we have

$$\log f_Y(y|\theta^k) - \log f_Y(y|\theta^{k-1}) \geq$$

$$KL(f_{X|Y}(x|y,\theta^{k-1}), f_{X|Y}(x|y,\theta^k)) \geq 0.$$

Therefore, the likelihood is increasing.

2.5 Some Difficulties

When the probability functions are replaced by probability density functions, some difficulties arise. In [39, 56] and elsewhere we are told that

$$f_{X|Y}(x|y,\theta) = f_X(x|\theta)/f_Y(y|\theta). \tag{2.11}$$

This is false; integrating with respect to x gives one on the left side and $1/f_Y(y|\theta)$ on the right side. Perhaps the equation is not meant to hold for all x, but just for some x. In fact, if there is a function h such that $Y = h(X)$, then Equation (2.11) might hold just for those x such that $h(x) = y$, as in the discrete case. However, this modification of Equation (2.11) fails in the continuous case of probability density functions, since $h^{-1}\{y\}$ is often a subset of zero measure. Even if the set $h^{-1}\{y\}$ has positive measure, integrating both sides of Equation (2.11) over $x \in h^{-1}\{y\}$ tells us that $f_Y(y|\theta) \leq 1$, which need not hold for probability density functions.

Everyone who works with the EM algorithm will say that the likelihood is increasing for the EM algorithm. This is true for the discrete case, as we just saw. The proof breaks down for probability density functions, however.

The "proof"in [39] and reproduced in [56] proceeds as follows. Use Equation (2.11) to get

$$\log f_X(x|\theta) = \log f_{X|Y}(x|y,\theta) + \log f_Y(y|\theta). \tag{2.12}$$

Then replace the term $\log f_X(x|\theta)$ in Equation (2.1) with the right side of Equation (2.12), obtaining

$$\log f_Y(y|\theta) - Q(\theta|\theta^{k-1}) = -\int f_{X|Y}(x|y,\theta^{k-1}) \log f_{X|Y}(x|y,\theta)dx. \tag{2.13}$$

Jensen's Inequality tells us that

$$\int u(x) \log u(x) dx \geq \int u(x) \log v(x) dx, \tag{2.14}$$

for any probability density functions $u(x)$ and $v(x)$. Since $f_{X|Y}(x|y,\theta)$ is a probability density function, we have

$$\int f_{X|Y}(x|y,\theta^{k-1}) \log f_{X|Y}(x|y,\theta)dx \leq$$
$$\int f_{X|Y}(x|y,\theta^{k-1}) \log f_{X|Y}(x|y,\theta^{k-1})dx. \tag{2.15}$$

We conclude, therefore, that $\log f_Y(y|\theta) - Q(\theta|\theta^{k-1})$ attains its minimum value at $\theta = \theta^{k-1}$. Then we have

$$\log f_Y(y|\theta^k) - \log f_Y(y|\theta^{k-1}) \geq Q(\theta^k|\theta^{k-1}) - Q(\theta^{k-1}|\theta^{k-1}) \geq 0. \tag{2.16}$$

From (2.16) we have

$$Q(\theta|\theta^{k-1}) + \Big(LL(\theta^{k-1}) - Q(\theta^{k-1}|\theta^{k-1})\Big) \leq LL(\theta), \tag{2.17}$$

which is sometimes described, in the optimization-tranfer context, by saying that, except for a constant, $Q(\theta|\theta^{k-1})$ is a “minorization”of $LL(\theta)$.

This “proof”is incorrect; clearly it rests on the validity of Equation (2.11), which is generally false. How we may go about correcting this flaw in the formulation of the EM algorithm is the topic of Chapter 3.

Chapter 3

Nonstochastic EM and STEM

The notion of conditional expectation is not one commonly found in the toolbox of the average graduate student. For that reason, I found it a bit difficult to introduce the EM algorithm to my students. In my search for an alternative approach I discovered what I call the *nonstochastic EM* (NSEM) algorithm. In this chapter we present the nonstochastic EM template for optimization and define the STEM template in terms of NSEM. It will follow from results concerning NSEM that likelihood is always increasing for STEM algorithms.

3.1 NSEM

We assume that there is a function $b : \Theta \times \Omega \to \mathbb{R}_+$, where (Ω, μ) is a measure space and

$$a(\theta) = -f(\theta) = \int_\Omega b(\theta, \omega) d\mu(\omega). \tag{3.1}$$

Let θ^0 be arbitrary. For $k = 1, 2, ...$, we maximize

$$\int_\Omega b(\theta^{k-1}, \omega) \log b(\theta, \omega) d\mu(\omega) \tag{3.2}$$

to get θ^k. Note that the integration may be replaced by summation, as needed. Using the Kullback–Leibler distance, we can reformulate the NSEM.

With the shorthand notation $b(\theta) = b(\theta, \omega)$ we define

$$KL\left(b(\theta), b(\gamma)\right) = \int_\Omega KL\left(b(\theta,\omega), b(\gamma,\omega)\right) d\mu(\omega).$$

Proposition 3.1 *The sequence $\{a(\theta^k)\}$ is increasing.*

Proof: We have

$$a(\theta^{k-1}) = a(\theta^{k-1}) - KL\left(b(\theta^{k-1}), b(\theta^{k-1})\right) \leq a(\theta^k) - KL\left(b(\theta^{k-1}), b(\theta^k)\right).$$

Therefore,

$$a(\theta^k) - a(\theta^{k-1}) \geq KL\left(b(\theta^{k-1}), b(\theta^k)\right).$$

■

We see easily that θ^k minimizes

$$G_k(\theta) = KL\left(b(\theta^{k-1}), b(\theta)\right) - a(\theta) = f(\theta) + d(\theta, \theta^{k-1}), \tag{3.3}$$

for

$$d(\theta, \gamma) = KL\left(b(\gamma), b(\theta)\right).$$

Consequently, the NSEM is an auxiliary-function method.

3.2 STEM

Now we define the STEM class of iterative algorithms as a subclass of the NSEM. For any random vectors X and Y governed by the joint probability density function or joint probability function $f_{X,Y}(x, y|\theta)$ we have

$$f_Y(y|\theta) = \int f_{X,Y}(x, y|\theta) dx. \tag{3.4}$$

With $a(\theta) = f_Y(y|\theta)$ and $b(\theta, \omega) = f_{X,Y}(x, y|\theta)$ we see that Equation (3.4) becomes Equation (3.1). For the case of probability functions, the integration is replaced by summation. So our STEM template fits into that of the NSEM. The iterative step is then to find θ^k by maximizing the function

$$\int f_{X,Y}(x, y|\theta^{k-1}) \log f_{X,Y}(x, y|\theta) dx.$$

It follows from our discussion of the NSEM that the sequence $\{f_Y(y|\theta^k)\}$ is increasing. Although the STEM approach can be viewed as an alternative to the usual EM method, the two are the same in many important cases, as we shall see now.

3.3 The Discrete Case, with $Y = h(X)$

In many applications of the EM algorithm Y takes values in $\mathbb{R}^M$, X takes values in $\mathbb{R}^N$, with $M \leq N$, and there is a function $h : \mathbb{R}^N \to \mathbb{R}^M$ with $Y = h(X)$. In the case of discrete Y and X and probability functions, we have

$$f_Y(y|\theta) = \sum_{x \in h^{-1}(y)} f_X(x|\theta), \tag{3.5}$$

where $h^{-1}(y)$ denotes the set of all x for which $y = h(x)$. The joint probability function is

$$f_{X,Y}(x, y|\theta) = f_X(x|\theta)c(x). \tag{3.6}$$

Therefore,

$$f_Y(y|\theta) = \sum_x f_{X,Y}(x, y|\theta),$$

so that the usual EM formulation matches that of the STEM. Consequently, the sequence $\{f_Y(y|\theta^k)\}$ is increasing.

3.4 The Continuous Case, with $Y = h(X)$

We suppose now that X and Y are no longer discrete and probability density functions replace the probability functions in the previous subsection. When we mimic Equation (3.5) with

$$f_Y(y|\theta) = \int_{x \in h^{-1}(y)} f_X(x|\theta)dx, \tag{3.7}$$

we run into a problem; the set $h^{-1}(y)$ often has measure zero, so this relationship does not hold. We cannot say that

$$f_{X,Y}(x, y|\theta) = f_X(x|\theta)c(x).$$

Later in these notes we shall consider some particular cases in which more can be said, and show how the STEM approach gives us a way out of this difficulty.

3.5 The Missing-Data Model

Most discussions of the EM algorithm refer to the data vector y as the *incomplete* data, the desired vector x as the *complete* data, and describe the situation by saying that there is *missing* data. As we shall see later, there certainly are examples in which this terminology is reasonable. One example to which we shall return later is that of censored exponential data.

As an illustration of censored exponential data, one often considers the problem of estimating the average lifetime of lightbulbs. A collection of bulbs are observed and their times-to-failure recorded. Perhaps, during the limited observation time, not all the bulbs failed. The *missing* data is then the times-to-failure of all the bulbs that failed to fail.

For the missing-data model the random variable Y is the observed data, the random variable W is the missing data, and $X = (Y, W)$ is the complete or preferred data. Then

$$f_Y(y|\theta) = \int f_{Y,W}(y, w)dw, \tag{3.8}$$

which fits into the STEM formulation. Once again, we can replace the integral with summation if necessary. Also

$$E(\log f_X(X|\theta)|y, \theta^{k-1}) = E(\log f_{Y,W}(y, W|\theta)|y, \theta^{k-1}), \tag{3.9}$$

so that

$$E(\log f_X(X|\theta)|y, \theta^{k-1}) = \int f_{W|Y}(w|y, \theta^{k-1}) \log f_{Y,W}(y, w|\theta^{k-1})dw. \tag{3.10}$$

Therefore, since

$$f_{W|Y}(w|y, \theta^{k-1}) = f_{Y,W}(y, w|\theta^{k-1})/f_Y(y|\theta^{k-1}),$$

the M-step of the EM algorithm is equivalent to maximizing

$$\int f_{Y,W}(y, w|\theta^{k-1}) \log f_{Y,W}(y, w|\theta)dw. \tag{3.11}$$

This is the iterative step of the STEM. Therefore, likelihood is increasing. This version of the EM algorithm is used in [47].

3.6 Another Approach

We suppose that there is a second function $k : R^N \to R^{N-M}$ such that the function $G : R^N \to R^N$ given by

$$G(x) = (h(x), k(x)) = (y, w) = u$$

is invertible, with inverse H and determinant of the Jacobian matrix denoted by $J(y, w)$. For any measurable set A in R^M we have

$$\int_A f_Y(y|\theta)dy = \int_{y\in A}\int_{w\in\mathcal{W}(y)} f_X(H(y,w)|\theta)J(y,w)dw,$$

where

$$\mathcal{W}(y) = \{w|w = k(x), y = h(x)\}.$$

It then follows that

$$f_Y(y|\theta) = \int_{w\in\mathcal{W}(y)} f_X(H(y,w)|\theta)J(y,w)dw,$$

so that, for $x \in h^{-1}(y)$,

$$b(x|y,\theta) = b(H(y,k(x))|y,\theta) = f_X(H(y,k(x))|\theta)J(y,k(x))/f_Y(y|\theta)$$

defines a probability density function on $h^{-1}(y)$.

For example, suppose that $X = (Z_1, Z_2)$, where Z_1 and Z_2 are independent and uniformly distributed on the interval $[0, \theta]$. Suppose that $Y = Z_1 + Z_2 = h(X)$. The set $h^{-1}(y)$ is the set of all points (z_1, z_2) for which $h(z_1, z_2) = z_1 + z_2 = y$, which is a set of planar measure zero. The function $f_Y(y|\theta)$ is

$$f_Y(y|\theta) = \begin{cases} y/\theta^2, & 0 \le y \le \theta; \\ (2\theta - y)/\theta^2, & \theta \le y \le 2\theta. \end{cases} \tag{3.12}$$

In our example, we have $N = 2$ and $M = 1$. Let $k(z_1, z_2) = z_1 - z_2$. Then

$$G(z_1, z_2) = (z_1 + z_2, z_1 - z_2),$$

$$H(y, w) = (\frac{y+w}{2}, \frac{y-w}{2}),$$

and

$$J(y, w) = 1/2.$$

The set $\mathcal{W}(y)$ is the entire real line.

The pdf for X is

$$f_X(z_1, z_2) = \frac{1}{\theta^2}\chi_{[0,\theta]}(z_1)\chi_{[0,\theta]}(z_2)$$

so the pdf for the random variable Y is

$$f_Y(y) = \frac{1}{2}\int_R f(H(y,w))dw = \frac{1}{2\theta^2}\int_R \chi_{[0,\theta]}(\frac{y+w}{2})\chi_{[0,\theta]}(\frac{y-w}{2})dw.$$

This is easily seen to be $\frac{y}{\theta^2}$, for $0 \leq y \leq \theta$ and $\frac{2\theta - y}{\theta^2}$, for $1 \leq y \leq 2\theta$, which is the pdf in Equation (3.12). Related ideas are discussed in [34].

3.7 Acceptable Data

As we discussed, the relationship $Y = h(X)$ is problematic when probability density functions are involved. In this section we describe a condition that we can use as an alternative to $Y = h(X)$.

We say that the random vector X is *acceptable* if the conditional pdf or pf $f_{Y|X}(y|x,\theta)$ is independent of θ, that is

$$f_{Y|X}(y|x,\theta) = f_{Y|X}(y|x). \tag{3.13}$$

Let X be acceptable. Using

$$f_{X,Y}(x,y|\theta^{k-1}) = f_{X|Y}(x|y,\theta^{k-1})f_Y(y|\theta^{k-1})$$

and

$$\log f_X(x|\theta) = \log f_{X,Y}(x,y|\theta) - \log f_{Y|X}(y|x)$$

we find that maximizing $E(\log f_X(X|\theta)|y,\theta^{k-1})$ is equivalent to maximizing the function

$$\int f_{X,Y}(x,y|\theta^{k-1}) \log f_{X,Y}(x,y|\theta)dx,$$

which is the iterative step of the STEM. Therefore, once again, the likelihood is increasing.

In Chapter 1 we encountered the problem of estimating the mixing proportions for a mixture of pdf's that arises in list-mode PET. We had

$$f(v|\theta) = \sum_{j=1}^{J} \theta_j f_j(v), \tag{3.14}$$

where θ_j is the probability that an emitted positron is emitted at pixel j and $f_j(v)$ is the pdf governing the distribution of the LOR resulting from emissions at the jth pixel. We assumed that our observed data is the vector $y = (v_1, v_2, ..., v_N)$, where v_n denotes the nth LOR on the list. The preferred data is the vector $x = (j_1, j_2, ..., j_N)$, where j_n denotes the pixel at which the nth positron was emitted. We do not have the relationship $Y = h(X)$; there is no algebraic formula that gives the value v_n from j_n. However, the preferred data is acceptable, since

$$f_{Y|X}(y|x, \theta) = \prod_{n=1}^{N} f_{j_n}(v_n)$$

does not involve θ.

3.8 Using X as Missing Data

As we have just seen, the missing-data model, in which the integral in (3.11) is maximized, is guaranteed to increase likelihood. Suppose that, having selected our preferred data X, we use the missing-data model, with $W = X$. Then the likelihood would always be increasing. Why not do this in every case and not worry about acceptable data? The answer is that the original EM algorithm has us maximizing

$$\int f_{X|Y}(x|y, \theta^{k-1}) \log f_X(x|\theta) dx,$$

while the missing-data model has us maximizing

$$\int f_{Y,X}(y, x|\theta^{k-1}) \log f_{Y,X}(y, x|\theta) dx.$$

These two approaches produce the same sequence of iterates whenever the preferred data X is acceptable.

3.9 A Counterexample

In this section we give an example that shows that the sequence of parameter estimates generated by an EM algorithm may converge to something other than the maximum-likelihood estimate.

Suppose that X_1 and X_2 are independent random variables uniformly distributed on the interval $[0, \theta_{true}]$, our actual, or observed, data is one realization y of the random variable $Y = X_1 + X_2$, and we want to estimate θ_{true}. We take as the virtual, or preferred, data the random vector $X = (X_1, X_2)$. The pdf for $X = (X_1, X_2)$ is

$$f_X(x_1, x_2|\theta) = \frac{1}{\theta^2}\chi_{[0,\theta]}(x_1)\chi_{[0,\theta]}(x_2).$$

The pdf for the random variable Y is the function

$$f_Y(y|\theta) = \begin{cases} y/\theta^2, & \text{for } 0 \leq y \leq \theta, \\ \\ (2\theta - y)/\theta^2, & \text{for } \theta \leq y \leq 2\theta. \end{cases}$$

If we had been given the data values x_1 and x_2, and not just $y = x_1 + x_2$, the maximum likelihood estimate of θ_{true} would have been the maximum of x_1 and x_2. Given only $y = x_1 + x_2$, the maximum-likelihood estimate of θ_{true} is $\theta_{ML} = y$.

Suppose we have an initial estimate θ^0 of the parameter θ_{true}. Since $y = x_1 + x_2$, it makes no sense to select a value of θ^0 less than $y/2$; therefore, let us assume that $\theta^0 \geq y/2$. The (E) step is to calculate the conditional expected value of the random variable

$$\log f_X(X|\theta) = \log \chi_{[0,\theta]}(X_1) + \log \chi_{[0,\theta]}(X_2) - 2\log\theta, \tag{3.15}$$

conditioned on θ^0 and y. For any θ in the interval $[y/2, \theta^0)$, there will be a positive conditional probability that one or both of X_1 or X_2 will exceed θ, so, in order for the conditional expected value to be finite, we must restrict θ to the closed ray $[\theta^0, +\infty)$. The conditional expected value of $f_X(X|\theta)$ is then $-2\log\theta$. The maximum of $-2\log\theta$ over the ray $[\theta^0, +\infty)$ occurs at $\theta = \theta^0$, so $\theta^1 = \theta^0$. Therefore, beginning with $\theta^0 \geq y/2$, we have $\theta^k = \theta^0$ for all $k = 1, 2, ...$, and so the sequence $\{\theta^k\}$ need not converge to $\theta_{ML} = y$, and $\{f(y|\theta^k)\}$ need not converge to $f(y|\theta_{ML})$.

3.10 Regular Exponential Families

The preferred data is said to come from an *exponential family* [56] if $\theta = (\theta_1, ..., \theta_J)^T$ and

$$f_X(x|\theta) = b(x)\exp\left(c(\theta)^T t(x)\right)/a(\theta), \tag{3.16}$$

where $K \geq J$, $b(x)$ and $a(\theta)$ are real-valued functions, $c(\theta) = (c_1(\theta), ..., c_K(\theta))^T$ is a vector function of the parameters, and $t(x) = (t_1(x), ..., t_K(x))^T$ is a vector function of the preferred data vector x and is a sufficient statistic for the estimation of θ. If $K = J$ and the Jacobian of $c(\theta)$ is invertible, the family is said to be *regular*. The mixture problem provides a good example of a regular exponential family.

In the mixture problem our preferred data is $x = (j_1, ..., j_N)^T$. Let $c(\theta)_j = \log \theta_j$, and $t(x)_j$ the cardinality of the set of all n such that $j_n = j$. Then $t(x)$ is a sufficient statistic for the estimation of θ. With $b(x) = 1$ and $a(\theta) = 1$, we see that the pdf $f_X(x)$ is described by Equation (3.16).

3.11 Our Other Objectives

Every EM algorithm that can be viewed as an NSEM algorithm satisfies the first of our three objectives listed in Section 2.2. To satisfy the second objective, that is, to have the sequence $\{L(\theta^k)\}$ converge to $L(\theta_{ML})$, we need additional structure. The theory of the SUMMA and SUMMA2 subclasses of AF templates provides conditions sufficient for the second objective to hold, which is the main reason for embedding the NSEM in these larger templates. The third objective is achieved in some particular cases, as we shall see, but there is no generally applicable theory to guarantee this. We turn now to the more general templates for optimization.

Chapter 4

Examples

In this chapter we present several examples of the use of STEM.

4.1 A Multinomial Example

In many applications, the entries of the vector y are independent realizations of a single real-valued or vector-valued random variable V, as they are, at least initially, for finite mixture problems. This is not always the case, however, as the following example shows.

A well known example that was used in [39] and again in [56] to illustrate the EM algorithm concerns a multinomial model taken from genetics. Here there are four cells, with cell probabilities $\frac{1}{2} + \frac{1}{4}\theta_0$, $\frac{1}{4}(1 - \theta_0)$, $\frac{1}{4}(1 - \theta_0)$, and $\frac{1}{4}\theta_0$, for some $\theta_0 \in \Theta = [0, 1]$ to be estimated. The entries of y are the frequencies from a sample size of 197. We then have

$$f_Y(y|\theta) = \frac{197!}{y_1!y_2!y_3!y_4!}(\frac{1}{2} + \frac{1}{4}\theta)^{y_1}(\frac{1}{4}(1-\theta))^{y_2}(\frac{1}{4}(1-\theta))^{y_3}(\frac{1}{4}\theta)^{y_4}. \quad (4.1)$$

It is then supposed that the first of the original four cells can be split into two sub-cells, with probabilities $\frac{1}{2}$ and $\frac{1}{4}\theta_0$. We then write $y_1 = y_{11} + y_{12}$, and let

$$X = (Y_{11}, Y_{12}, Y_2, Y_3, Y_4), \quad (4.2)$$

where X has a multinomial distribution with five cells. Note that we do now have $Y = h(X)$.

This example is a popular one in the literature on the EM algorithm (see [39] for citations). It is never suggested that the splitting of the first group into two subgroups is motivated by the demands of the genetics theory itself. As stated in [56], the motivation for the splitting is to allow us to view the two random variables $Y_{12}+Y_4$ and Y_2+Y_3 as governed by a binomial distribution; that is, we can view the value of $y_{12}+y_4$ as the number of heads, and the value y_2+y_3 as the number of tails that occur in the flipping of a biased coin $y_{12}+y_4+y_2+y_3$ times. This simplifies the calculation of the likelihood maximizer.

4.2 Censored Exponential Data

McLachlan and Krishnan [56] give the following example of a likelihood maximization problem involving probability density functions. This example provides a good illustration of the usefulness of the missing-data model.

Suppose that Z is the time until failure of a component, which we assume is governed by the exponential distribution

$$f(z|\theta)=\frac{1}{\theta}e^{-z/\theta}, \tag{4.3}$$

where the parameter $\theta>0$ is the expected time until failure. We observe a random sample of N components and record their failure times, z_n. On the basis of this data, we must estimate θ, the mean time until failure.

It may well happen, however, that during the time allotted for observing the components, only r of the N components fail, which, for convenience, are taken to be the first r items in the record. Rather than wait longer, we record the failure times of those that failed, and record the elapsed time for the experiment, say T, for those that had not yet failed. The *censored data* is then $y=(y_1,...,y_N)$, where $y_n=z_n$ is the time until failure for $n=1,...,r$, and $y_n=T$ for $n=r+1,...,N$. The censored data is reasonably viewed as *incomplete*, relative to the *complete* data we would have had, had the trial lasted until all the components had failed.

Since the probability that a component will survive until time T is $e^{-T/\theta}$, the pdf for the vector y is

$$f_Y(y|\theta)=\Big(\prod_{n=1}^{r}\frac{1}{\theta}e^{-y_n/\theta}\Big)\,e^{-(N-r)T/\theta}, \tag{4.4}$$

and the log likelihood function for the censored, or incomplete, data is

$$\log f_Y(y|\theta)=-r\log\theta-\frac{1}{\theta}\sum_{n=1}^{N}y_n. \tag{4.5}$$

In this particular example we are fortunate, in that we can maximize

$f_Y(y|\theta)$ easily, and find that the ML solution based on the incomplete, censored data is

$$\theta_{MLi} = \frac{1}{r}\sum_{n=1}^{N} y_n = \frac{1}{r}\sum_{n=1}^{r} y_n + \frac{N-r}{r}T. \tag{4.6}$$

In most cases in which our data is incomplete, finding the ML estimate from the incomplete data is difficult, while finding it for the complete data is relatively easy.

We say that the missing data are the times until failure of those components that did not fail during the observation time. The preferred data is the complete data $x = (z_1, ..., z_N)$ of actual times until failure. The pdf for the preferred data X is

$$f_X(x|\theta) = \prod_{n=1}^{N} \frac{1}{\theta} e^{-z_n/\theta}, \tag{4.7}$$

and the log likelihood function based on the complete data is

$$\log f_X(x|\theta) = -N\log\theta - \frac{1}{\theta}\sum_{n=1}^{N} z_n. \tag{4.8}$$

The ML estimate of θ from the complete data is easily seen to be

$$\theta_{MLc} = \frac{1}{N}\sum_{n=1}^{N} z_n. \tag{4.9}$$

In this example, both the incomplete-data vector y and the preferred-data vector x lie in $\mathbb{R}^N$. We have $y = h(x)$ where the function h operates by setting to T any component of x that exceeds T. Clearly, for a given y, the set $h^{-1}\{y\}$ consists of all vectors x with entries $x_n \geq T$ or $x_n = y_n < T$. For example, suppose that $N = 2$, and $y = (y_1, T)$, where $y_1 < T$. Then $h^{-1}\{y\}$ is the one-dimensional ray

$$h^{-1}\{y\} = \{x = (y_1, x_2)|\, x_2 \geq T\}.$$

Because this set has measure zero in $\mathbb{R}^2$, Equation (3.7) does not make sense in this case.

We need to calculate $E(\log f_X(X|\theta)|y, \theta^k)$. Following McLachlan and Krishnan [56], we note that since $\log f_X(x|\theta)$ is linear in the unobserved data Z_n, $n = r+1, ..., N$, to calculate $E(\log f_X(X|\theta)|y, \theta^k)$ we need only replace the unobserved values with their conditional expected values, given y and θ^k. The conditional distribution of $Z_n - T$, given that $Z_n > T$, is still exponential, with mean θ. Therefore, we replace the unobserved values, that is, all the Z_n for $n = r+1, ..., N$, with $T + \theta^k$. Therefore, at the E-step we have

$$E(\log f_X(X|\theta)|y, \theta^k) = -N \log \theta - \frac{1}{\theta}\Bigg(\Big(\sum_{n=1}^{N} y_n\Big) + (N-r)\theta^k\Bigg). \quad (4.10)$$

The M-step is to maximize this function of θ, which leads to

$$\theta^{k+1} = \Bigg(\Big(\sum_{n=1}^{N} y_n\Big) + (N-r)\theta^k\Bigg)/N. \quad (4.11)$$

Let θ^* be a fixed point of this iteration. Then we have

$$\theta^* = \Bigg(\Big(\sum_{n=1}^{N} y_n\Big) + (N-r)\theta^*\Bigg)/N,$$

so that

$$\theta^* = \frac{1}{r}\sum_{n=1}^{N} y_n,$$

which, as we have seen, is the likelihood maximizer. From

$$\theta^k - \theta^* = (1 - \frac{r}{N})(\theta^{k-1} - \theta^*)$$

it follows that the sequence $\{\theta^k\}$ converges to θ^*.

We show now that likelihood is non-decreasing in this example. We have

$$LL(\theta) = -r \log \theta - \frac{1}{\theta}\sum_{n=1}^{N} y_n. \quad (4.12)$$

We know that

$$\left(-r \log \theta^{k+1} - \frac{1}{\theta^{k+1}}\sum_{n=1}^{N} y_n\right) - (N-r)\left(\log \theta^{k+1} + \frac{\theta^k}{\theta^{k+1}}\right)$$

$$\geq \left(-r \log \theta^k - \frac{1}{\theta^k}\sum_{n=1}^{N} y_n\right) - (N-r)\left(\log \theta^k + 1\right).$$

Therefore,

$$LL(\theta^{k+1}) - LL(\theta^k) = \left(-r \log \theta^{k+1} - \frac{1}{\theta^{k+1}}\sum_{n=1}^{N} y_n\right) - \left(-r \log \theta^k - \frac{1}{\theta^k}\sum_{n=1}^{N} y_n\right)$$

$$\geq (N-r)\left(\log \theta^{k+1} + \frac{\theta^k}{\theta^{k+1}} - \log \theta^k + 1\right) = (N-r)\left(\frac{\theta^k}{\theta^{k+1}} - 1 - \log \frac{\theta^k}{\theta^{k+1}}\right) \geq 0.$$

4.3 An Example from Genetics

The blood groups for human beings are O, A, B, and AB. To which class a particular human belongs is determined by genes O, A, and B, with O recessive to A and B. With p (respectively, q and r) the probability of receiving gene A (respectively, B and O) from one parent, the probability of being in blood group O is r^2, in blood group A is $p^2 + 2pr$, in blood group B is $q^2 + 2qr$, and in blood group AB is $2pq$. Our data are the group frequencies in our sample, $y = (n_O, n_A, n_B, n_{AB})$. Our goal is to estimate $\theta = (p, q, r)$. To simplify the calculations it is proposed in [56] that we take as our preferred data $x = (n_O, n_{AA}, n_{AO}, n_{BB}, n_{BO})$. The class A is made up of individuals who received either two A genes or one A and one O, with $n_A = n_{AA} + n_{AO}$. In designing the preferred data we imagine that we have the frequencies for these two subgroups, not just their sum.

With

$$m_A = n_{AA} + \frac{1}{2} n_{AO} + \frac{1}{2} n_{AB},$$

$$m_B = n_{BB} + \frac{1}{2} n_{BO} + \frac{1}{2} n_{AB},$$

and

$$m_O = n_O + \frac{1}{2} n_{AO} + \frac{1}{2} n_{BO},$$

we find that

$$\log f_X(x|\theta) = 2m_A \log p + 2m_B \log q + 2m_O \log r.$$

This is the log likelihood function for a multinomial distribution with frequencies m_A, m_B, and m_O and probabilities p, q, and r. The problem can now be solved using the iterative algorithm in Equation (1.11).

Chapter 5

Templates for Iterative Optimization

5.1 Definition and Basic Properties

In this chapter we discuss in some detail several templates for (or classes of) iterative optimization algorithms. The most general class consists of *auxiliary-function* (AF) methods. We show that three well known classes of iterative methods, *alternating minimization* (AM), *proximity-function* algorithms (PMA), and *majorization minimization* (MM) algorithms, are subclasses of AF and are equivalent to one another.

In the interest of consistent notation we shall describe the basic problem as follows. Let $f : \Theta \to \mathbb{R}$, where Θ is an arbitrary nonempty set. The problem is to minimize $f(\theta)$ over θ in the set Θ. When we apply these templates to the likelihood maximization problem the function $f(\theta)$ will be the negative of the log likelihood function $LL(\theta) = \log f_Y(y|\theta)$.

5.2 Auxiliary-Function Methods

The most general template that we consider here is the auxiliary-function (AF) template. A wide variety of iterative optimization methods are particular cases of AF algorithms [21]. Let θ^0 be arbitrary. For $k = 1, 2, ...$, we minimize the function

$$G_k(\theta) = f(\theta) + g_k(\theta), \tag{5.1}$$

to get θ^k, where $g_k : \Theta \to [0, +\infty]$. If the objective is to minimize $f(\theta)$ over some subset $\Gamma \subseteq \Theta$, we have a choice: we can augment $f(\theta)$ by adding a function that is zero within Γ and $+\infty$ outside Γ, or we can select the functions $g_k(\theta)$ to take the value $+\infty$ outside Γ. In any case, we will always select $g_k(\theta)$ to be finite whenever $f(\theta)$ is finite.

For this to be an AF method we require that the auxiliary functions $g_k(\theta)$ be nonnegative and $g_k(\theta^{k-1}) = 0$.

Lemma 5.1 *For any AF algorithm the sequence $\{f(\theta^k)\}$ is decreasing and converges to some $\beta^* \geq -\infty$. If the function f is bounded below, then the sequence $\{g_k(\theta^k)\}$ converges to zero.*

Proof: We have

$$f(\theta^{k-1}) = G_k(\theta^{k-1}) \geq G_k(\theta^k) = f(\theta^k) + g_k(\theta^k),$$

so that

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

∎

Let $\beta = \inf_{\theta\in\Theta} f(\theta)$. Later in these notes we shall consider conditions under which we can assert that $\beta^* = \beta$.

5.3 Alternating Minimization

Although it may not be immediately obvious, the alternating-minimization (AM) template of Csiszár and Tusnády [37] can be shown to be contained in the AF template.

Let $\Phi : P \times Q \to (-\infty, +\infty]$, where P and Q are arbitrary nonempty sets. In the AM approach we minimize $\Phi(p, q^{k-1})$ over $p \in P$ to get p^k and then minimize $\Phi(p^k, q)$ over $q \in Q$ to get q^k. It follows immediately that the sequence $\{\Phi(p^k, q^k)\}$ is decreasing. We want

$$\{\Phi(p^k, q^k)\} \downarrow \beta = \inf\{\Phi(p, q)|p \in P, q \in Q\}. \tag{5.2}$$

In [37] Csiszár and Tusnády show that, if the function Φ possesses what they call the *five-point property* (5PP),

$$\Phi(p, q) + \Phi(p, q^{k-1}) \geq \Phi(p, q^k) + \Phi(p^k, q^{k-1}), \tag{5.3}$$

for all p, q, and k, then (5.2) holds. There seemed to be no convincing explanation of why the five-point property should be used, except that it works. I was quite surprised when I discovered that the AM method can be reformulated as an AF method to minimize a function of the single variable p, and that the five-point property for AM is precisely the SUMMA Inequality [23] to be discussed later. For each p select $q(p)$ for which $\Phi(p, q(p)) \leq \Phi(p, q)$ for all $q \in Q$. Then define $f(p) \doteq \Phi(p, q(p))$. Since $q^{k-1} = q(p^{k-1})$, we have

$$\Phi(p, q^{k-1}) = \Phi(p, q(p^{k-1})).$$

Minimizing $\Phi(p, q^{k-1})$ to get p^k is equivalent to minimizing

$$G_k(p) = \Phi(p, q(p)) + \Phi(p, q(p^{k-1})) - \Phi(p, q(p)) = f(p) + g_k(p), \tag{5.4}$$

where

$$g_k(p) = \Phi(p, q(p^{k-1})) - \Phi(p, q(p)).$$

Clearly, $g_k(p) \geq 0$ and $g_k(p^{k-1}) = 0$. With p and P replaced by θ and Θ, respectively, Equation (5.4) becomes Equation (5.1). Therefore, every AM algorithm is also an AF algorithm.

We define a "distance"$d(p, p')$ on the set $P \times P$ by

$$d(p, p') \doteq \Phi(p, q(p')) - \Phi(p, q(p)). \tag{5.5}$$

Then we see that p^k is obtained by minimizing $f(p) + d(p, p^{k-1})$.

5.3.1 The Three- and Four-Point Properties

It is often the case that AM methods are described using the *three-* and *four-point properties* (3PP and 4PP). The 3PP is

$$\Phi(p, q^{k-1}) - \Phi(p^k, q^{k-1}) \geq \Delta(p, p^k) \geq 0, \tag{5.6}$$

where $\Delta : P \times P \to \mathbb{R}_+$ and $\Delta(p, p) = 0$, for all $p \in P$. The 4PP is the following:

$$\Delta(p, p^k) \geq \Phi(p, q^k) - \Phi(p, q), \tag{5.7}$$

for all p, q, and k. Clearly, the 3PP and 4PP together imply the 5PP.

When the 3PP and 4PP hold we have

$$\Delta(p,p') \geq d(p,p') = \Phi(p,q(p')) - \Phi(p,q(p)).$$

If we redefine Δ by $\Delta(p,p') \doteq d(p,p')$, then the 4PP is automatically true and the 3PP becomes equivalent to the 5PP. The 3PP is now

$$\Phi(p,q^{k-1}) - \Phi(p^k,q^{k-1}) \geq d(p,p^k). \tag{5.8}$$

The weak 3PP (w3PP), defined by

$$\Phi(p,q^{k-1}) - \Phi(p^k,q^k) \geq dp,(p^k), \tag{5.9}$$

will play a role in Chapter 6.

5.3.2 Alternating Bregman Distance Minimization

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

5.3.3 Bregman Distances

Let $f : \mathbb{R}^J \to \mathbb{R}$ be a Bregman function [7, 31, 9], and so $f(x)$ is convex on its domain and differentiable in the interior of its domain. Then, for x in the domain and z in the interior, we define the Bregman distance $D_f(x,z)$ by

$$D_f(x,z) = f(x) - f(z) - \langle \nabla f(z), x - z \rangle. \tag{5.10}$$

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

$$f(x) = \sum_{j=1}^{J} x_j \log x_j \, - x_j. \tag{5.11}$$

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

$$\langle \nabla f(p^{n+1}) - \nabla f(q^n), p - p^{n+1} \rangle \geq 0, \tag{5.12}$$

for all $p \in P$. Since

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

$$D_f(p, p^{n+1}) + \langle \nabla f(p^{n+1}) - \nabla f(q^n), p - p^{n+1} \rangle, \tag{5.13}$$

it follows that the three-point property holds, with

$$\Phi(p, q) = D_f(p, q), \tag{5.14}$$

and

$$\Delta(p, p') = D_f(p, p'). \tag{5.15}$$

To get the four-point property we need to restrict D_f somewhat; we assume from now on that $D_f(p, q)$ is jointly convex, that is, it is convex in the combined vector variable (p, q) (see [3]). Now we can invoke a lemma due to Eggermont and LaRiccia [42].

5.3.4 The Eggermont–LaRiccia Lemma

Lemma 5.2 *Suppose that the Bregman distance $D_f(p, q)$ is jointly convex. Then it has the four-point property.*

Proof: By joint convexity we have

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

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

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

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

for all q. Also,

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

It follows that

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

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

Therefore, we have

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

This is the four-point property. ∎

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

5.3.5 The Bauschke–Combettes–Noll Problem

In [4] Bauschke, Combettes and Noll consider the following problem: minimize the function

$$\Lambda(p,q) = \phi(p) + \psi(q) + D_f(p,q), \tag{5.16}$$

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

$$\beta = \inf_{(p,q)} \Lambda(p,q) > -\infty, \tag{5.17}$$

and seek a sequence $\{(p^n,q^n)\}$ such that $\{\Lambda(p^n,q^n)\}$ converges to β. The sequence is obtained by the AM method, as in our previous discussion. They prove that, if the Bregman distance is jointly convex, then $\{\Lambda(p^n,q^n)\} \downarrow \beta$. In this subsection we obtain this result by showing that $\Lambda(p,q)$ has the five-point property whenever $D = D_f$ is jointly convex. Our proof is loosely based on the proof of the Eggermont-LaRiccia lemma.

The five-point property for $\Lambda(p,q)$ is

$$\Lambda(p,q^{n-1}) - \Lambda(p^n,q^{n-1}) \geq \Lambda(p,q^n) - \Lambda(p,q). \tag{5.18}$$

Lemma 5.3 *The inequality in (5.18) is equivalent to*

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

$$D(p,q^n) + D(p^n,q^{n-1}) - D(p,q^{n-1}) - D(p^n,q^n). \tag{5.19}$$

By the joint convexity of $D(p,q)$ and the convexity of ϕ and ψ we have

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

$$\langle \nabla_p \Lambda(p^n,q^n), p-p^n\rangle + \langle \nabla_q \Lambda(p^n,q^n), q-q^n\rangle, \tag{5.20}$$

where $\nabla_p \Lambda(p^n, q^n)$ denotes the gradient of $\Lambda(p,q)$, with respect to p, evaluated at (p^n, q^n).

Since q^n minimizes $\Lambda(p^n, q)$, it follows that

$$\langle \nabla_q \Lambda(p^n, q^n), q - q^n \rangle = 0, \tag{5.21}$$

for all q. Therefore,

$$\Lambda(p,q) - \Lambda(p^n, q^n) \geq \langle \nabla_p \Lambda(p^n, q^n), p - p^n \rangle . \tag{5.22}$$

We have

$$\langle \nabla_p \Lambda(p^n, q^n), p - p^n \rangle =$$

$$\langle \nabla f(p^n) - \nabla f(q^n), p - p^n \rangle + \langle \nabla \phi(p^n), p - p^n \rangle. \tag{5.23}$$

Since p^n minimizes $\Lambda(p, q^{n-1})$, we have

$$\nabla_p \Lambda(p^n, q^{n-1}) = 0, \tag{5.24}$$

or

$$\nabla \phi(p^n) = \nabla f(q^{n-1}) - \nabla f(p^n), \tag{5.25}$$

so that

$$\langle \nabla_p \Lambda(p^n, q^n), p - p^n \rangle = \langle \nabla f(q^{n-1}) - \nabla f(q^n), p - p^n \rangle$$

$$= D(p, q^n) + D(p^n, q^{n-1}) - D(p, q^{n-1}) - D(p^n, q^n). \tag{5.26}$$

Using (5.22) we obtain the inequality in (5.19). This shows that $\Lambda(p,q)$ has the five-point property whenever the Bregman distance $D = D_f$ is jointly convex. From our previous discussion of AM, we conclude that the sequence $\{\Lambda(p^n, q^n)\}$ converges to β; this is Corollary 4.3 of [4].

5.4 Proximal Minimization

Proximal minimization algorithms (PMA) can be shown to be particular cases of AM algorithms, and, more directly, of AF algorithms. Let $d : \Theta \times \Theta \to \mathbb{R}_+$ be an arbitrary "distance", with $d(\theta, \theta) = 0$, for all θ. Let θ^0 be arbitrary in Θ. For each $k = 1, 2, ...$ we minimize the function

$$G_k(\theta) = f(\theta) + d(\theta, \theta^{k-1}) \tag{5.27}$$

to get θ^k. Clearly, since

$$g_k(\theta) = d(\theta, \theta^{k-1})$$

is nonnegative and $g_k(\theta^{k-1}) = 0$, any PMA is an AF algorithm. Using the distance defined in Equation (5.5), we see that all AM algorithms can be reformulated as PMA. It is clear from our previous discussion of the NSEM template that every NSEM algorithm is also in the PMA class. In Chapter 7 we consider PMA algorithms for which the distance function is a Bregman distance. We shall call such algorithms members of the PMAB class.

With $P = Q = \Theta$ and

$$\Phi(p, q) = \Phi(\theta, \gamma) = f(\theta) + d(\theta, \gamma) \tag{5.28}$$

we see that minimizing $G_k(\theta)$ in Equation (5.27) to get θ^k is equivalent to minimizing $\Phi(\theta, \theta^{k-1})$ and that minimizing $\Phi(\theta^k, \gamma)$ gives $\gamma = \theta^k$ again. Therefore, any PMA is also an AM algorithm.

5.5 Majorization Minimization

In [35] the authors review the use, in statistics, of “majorization minimization” (MM), also called “optimization transfer”. The objective is to minimize $f : \Theta \to \mathbb{R}$. In MM methods a second “majorizing” function $g(\theta|\gamma)$ is postulated, with the properties $g(\theta|\gamma) \geq f(\theta)$ and $g(\theta|\theta) = f(\theta)$. We then minimize $g(\theta|\theta^{k-1})$ to get θ^k. With

$$d(\theta, \gamma) \doteq g(\theta|\gamma) - f(\theta),$$

it is clear that MM methods are equivalent to PMA. In numerous papers [43, 1] Jeff Fessler and his colleagues use the terminology “surrogate-function minimization” to describe optimization transfer.

Chapter 6

SUMMA

6.1 Definition and Basic Properties

As we have seen, for any AF algorithm the sequence $\{f(\theta^k)\}$ is decreasing and so converges to some $\beta^* \geq -\infty$.We want more, however; we want $\beta^* = \beta \doteq \inf_\theta f(\theta)$. To have this we need to impose an additional condition on the auxiliary functions $g_k(\theta)$; the SUMMA Inequality is one such additional condition. To motivate our definition of the SUMMA Inequality we consider briefly barrier-function algorithms for constrained optimization.

6.2 Barrier-Function Algorithms

The problem is to minimize $f : \Theta \to \mathbb{R}$, subject to $\theta \in \Gamma$, where Γ is a nonempty subset of an arbitary set Θ. We select $b : \Theta \to (0, +\infty]$ with $\Gamma = \{\theta | 0 < b(\theta) < +\infty\}$. For each k we minimize $B_k(\theta) = f(\theta) + \frac{1}{k} b(\theta)$ over all $\theta \in \Theta$ to get θ^k, which must necessarily lie in Γ. Formulated this way, the method is not yet in AF form. Nevertheless, we have the following proposition.

Proposition 6.1 *The sequence $\{b(\theta^k)\}$ is increasing, and the sequence $\{f(\theta^k)\}$ is decreasing and converges to $\beta = \inf_{\theta \in \Gamma} f(\theta)$.*

Proof: From $B_k(\theta^{k-1}) \geq B_k(\theta^k)$ and $B_{k-1}(\theta^k) \geq B_{k-1}(\theta^{k-1})$, for $k = 2, 3, ...$, it follows easily that

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

Suppose that $\{f(\theta^k)\} \downarrow \beta^* > \beta$. Then there is $\gamma \in \Gamma$ with

$$f(\theta^k) \geq \beta^* > f(\gamma) \geq \beta,$$

for all k. Then

$$\frac{1}{k}(b(\gamma) - b(\theta^k)) \geq f(\theta^k) - f(\gamma) \geq \beta^* - f(\gamma) > 0,$$

for all k. But the sequence $\{\frac{1}{k}(b(\gamma) - b(\theta^k))\}$ converges to zero, which contradicts the assumption that $\beta^* > \beta$. ▮

The proof of Proposition 6.1 depended heavily on the details of the barrier-function method. Now we reformulate the barrier-function method as an AF method.

Minimizing $B_k(\theta) = f(\theta) + \frac{1}{k}b(\theta)$ to get θ^k is equivalent to minimizing $kf(\theta) + b(\theta)$, which, in turn, is equivalent to minimizing

$$G_k(\theta) = f(\theta) + g_k(\theta),$$

where

$$g_k(\theta) = [(k-1)f(\theta) + b(\theta)] - [(k-1)f(\theta^{k-1}) + b(\theta^{k-1})].$$

Clearly, $g_k(\theta) \geq 0$ and $g_k(\theta^{k-1}) = 0$. Now we have the AF form of the method. A simple calculation shows that

$$G_k(\theta) - G_k(\theta^k) = g_{k+1}(\theta), \tag{6.1}$$

for all $\theta \in \Theta$. Equation (6.1) serves to motivate our definition of the SUMMA Inequality.

6.3 The SUMMA Inequality

We say that an AF algorithm is in the SUMMA class if the SUMMA Inequality holds for all θ in Θ:

$$G_k(\theta) - G_k(\theta^k) \geq g_{k+1}(\theta). \tag{6.2}$$

One consequence of the SUMMA Inequality is

$$g_k(\theta) + f(\theta) \geq g_{k+1}(\theta) + f(\theta^k), \tag{6.3}$$

for all $\theta \in \Theta$. It follows from this that $\beta^* = \beta$. If this were not the case, then there would be $\phi \in \Theta$ with

$$f(\theta^k) \geq \beta^* > f(\phi)$$

for all k. The sequence $\{g_k(\phi)\}$ would then be a decreasing sequence of nonnegative terms with the sequence of its successive differences bounded below by $\beta^* - f(\phi) > 0$.

As we shall discuss, there are many iterative algorithms that satisfy the SUMMA Inequality, and are therefore in the SUMMA class. However, some important methods that are not in this class still have $\beta^* = \beta$; one example is the proximal minimization method of Auslender and Teboulle [2]. This suggests that the SUMMA class, large as it is, is still unnecessarily restrictive. This leads us to the definition of the SUMMA2 class.

6.4 The SUMMA2 Class

An AF algorithm is said to be in the SUMMA2 class if, for each sequence $\{\theta^k\}$ generated by the algorithm, there are functions $h_k : \Theta \to \mathbb{R}_+$ such that, for all $\theta \in \Theta$, we have

$$h_k(\theta) + f(\theta) \geq h_{k+1}(\theta) + f(\theta^k). \tag{6.4}$$

Any algorithm in the SUMMA class is in the SUMMA2 class; use $h_k = g_k$. As in the SUMMA case, we must have $\beta^* = \beta$, since otherwise the successive differences of the sequence $\{h_k(\phi)\}$ would be bounded below by $\beta^* - f(\phi) > 0$. It is helpful to note that the functions h_k need not be the g_k, and we do not require that $h_k(\theta^{k-1}) = 0$.

The PMA of Auslender and Teboulle [2] is in the SUMMA2 class. It is natural to ask if there are algorithms in the SUMMA2 class that are not in SUMMA and are not in the class defined by Auslender and Teboulle. There are such algorithms. As we shall discuss later, the *expectation maximization maximum likelihood* (EMML) [70, 10, 11, 12], as it is usually formulated, is such an algorithm.

6.5 AM and SUMMA

Let $\Phi : P \times Q \rightarrow (-\infty, +\infty]$, where P and Q are arbitrary nonempty sets. In the AM approach we minimize $\Phi(p, q^{k-1})$ over $p \in P$ to get p^k and then minimize $\Phi(p^k, q)$ over $q \in Q$ to get q^k. It follows immediately that the sequence $\{\Phi(p^k, q^k)\}$ is decreasing. The AM method can be reformulated as an AF method to minimize a function of the single variable p, and the five-point property for AM is precisely the SUMMA Inequality. For each p select $q(p)$ for which $\Phi(p, q(p)) \leq \Phi(p, q)$ for all $q \in Q$. Then let $f(p) = \Phi(p, q(p))$. Then, since $q^{k-1} = q(p^{k-1})$, we have

$$\Phi(p, q^{k-1}) = \Phi(p, q(p^{k-1})).$$

Minimizing $\Phi(p, q^{k-1})$ to get p^k is equivalent to minimizing

$$G_k(p) = \Phi(p, q(p)) + \Phi(p, q(p^{k-1})) - \Phi(p, q(p)) = f(p) + g_k(p), \tag{6.5}$$

where

$$g_k(p) = \Phi(p, q(p^{k-1})) - \Phi(p, q(p)). \tag{6.6}$$

Clearly, $g_k(p) \geq 0$ and $g_k(p^{k-1}) = 0$, so every AM algorithm is also an AF algorithm.

We want

$$\{\Phi(p^k, q^k)\} \downarrow \beta = \inf\{\Phi(p, q) | p \in P, q \in Q\}. \tag{6.7}$$

In [37] Csiszár and Tusnády show that, if the function Φ possesses what they call the *five-point property*,

$$\Phi(p, q) + \Phi(p, q^{k-1}) \geq \Phi(p, q^k) + \Phi(p^k, q^{k-1}), \tag{6.8}$$

for all p, q, and k, then (6.7) holds. With $g_k(p)$ as in Equation (6.6) we can easily show that the five-point property is precisely the SUMMA Inequality; every AM algorithm with the five-point property is in the SUMMA class.

As we saw in Chapter 5, when we define the distance Δ by $\Delta(p, p') \doteq d(p, p')$ the 4PP is automatically true and the 3PP and the 5PP become equivalent. Therefore, we need only focus on the 3PP, which, because it is equivalent to the 5PP, is now equivalent to the SUMMA inequality. The weak 3PP (w3PP) is

$$\Phi(p, q^{k-1}) - \Phi(p^k, q^k) \geq d(p, p^k), \tag{6.9}$$

or

$$\Phi(p, q^{k-1}) - \Phi(p, q^k) \geq f(p^k) - f(p). \tag{6.10}$$

Since the inequality in (6.10) is equivalent to

$$d(p, p^{k-1}) - d(p, p^k) \geq f(p^k) - f(p), \tag{6.11}$$

for all p, we see that every AM algorithm with the w3PP is in the SUMMA2 class.

6.6 The Bauschke–Combettes–Noll Problem Revisited

The BCN problem concerns the use of AM on the function $\Lambda(p, q)$ given by

$$\Lambda(p, q) = \phi(p) + \psi(q) + D_f(p, q), \tag{6.12}$$

where ϕ and ψ are convex on $\mathbb{R}^J$, D_f is a Bregman distance, and $P = Q$ is the interior of the domain of f. Their iterative steps are to minimize $\Lambda(p^{k-1}, q)$ to get q^k and then to minimize $\Lambda(p, q^k)$ to get p^k. From [4] we know that the five-point property (5PP) holds whenever the Bregman function is jointly convex.

We consider now the particular case in which the function $\psi(q) = 0$, for all q. Then we minimize $\phi(p^{k-1}) + D_f(p^{k-1}, q)$ to get $q^k = p^{k-1}$ and then minimize

$$G_k(p) = \phi(p) + D_f(p, p^{k-1})$$

to get p^k. This iterative algorithm is in the PMAB class. As we shall show in Chapter 7, all PMAB algorithms are in the SUMMA class.

In the previous subsection we learned that the function $\Phi(p, q) = \psi(p) + D_f(p, q)$ has the 5PP if and only if it can be reformulated as a SUMMA algorithm for minimizing the function $\Phi(p, q(p))$. In this case $q(p) = p$ and $\Phi(p, q(p)) = \psi(p)$. Therefore, since the iterative algorithm obtained by minimizing $\Phi(p, p^{k-1}) = \psi(p) + D_f(p, p^{k-1})$ to get p^k is in the SUMMA class, the function $\Phi(p, q) = \psi(p) + D_f(p, q)$ has the 5PP for all Bregman distances $D_f(p, q)$; we do not need that the Bregman distance be jointly convex.

6.7 The PMA of Auslender and Teboulle

In [2] Auslender and Teboulle take C to be a closed, nonempty, convex subset of $\mathbb{R}^J$, with interior U. At the kth step of their method one minimizes a function

$$G_k(x) = f(x) + d(x, x^{k-1}) \tag{6.13}$$

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

$$\langle \nabla_1 d(b, a), c - b \rangle \leq H(c, a) - H(c, b), \tag{6.14}$$

for all c in U.

6.7.1 Bregman Distances

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

$$\langle \nabla_1 d(b, a), c - b \rangle = D_h(c, a) - D_h(c, b) - D_h(b, a) \tag{6.15}$$

we see that D_h has $H = D_h$ for its associated induced proximal distance, so D_h is *self-proximal*, in the terminology of [2]. The method of Auslender and Teboulle seems not to be a particular case of SUMMA. However, it is in the SUMMA2 class, as we now show.

Denote by

$$\partial f(x) = \{u | f(y) - f(x) - \langle \nabla u, y - x \rangle \geq 0, \text{for all}\, y\}$$

the subdifferential of f at x. Since x^k minimizes $f(x) + d(x, x^{k-1})$, it follows that

$$0 \in \partial f(x^k) + \nabla_1 d(x^k, x^{k-1}),$$

so that

$$-\nabla_1 d(x^k, x^{k-1}) \in \partial f(x^k).$$

We then have

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

Using the associated induced proximal distance H, we find that, for all x,

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

Therefore, this method is in the SUMMA2 class, with the choice of $h_k(x) = H(x, x^{k-1})$. Consequently, we have $\beta^* = \beta$ for these algorithms.

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

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

which involves the $f(x)$ being minimized, but does not require that $f(x)$ be convex; it does not even require any structure on the domain of f. The SUMMA2 approach is general enough to include both classes.

6.7.2 D_ϕ Distances

Auslender and Teboulle consider two types of distances d for which there are induced proximal distances h: the first type are the Bregman distances, which are self-proximal in the sense that $d = H$; the second type are those having the form

$$d(x, z) = d_\phi(x, z) \doteq \sum_{j=1}^{J} z_j \phi(\frac{x_j}{z_j}), \tag{6.16}$$

for functions ϕ having certain properties to be discussed below. In such cases the induced proximal distance is $h(x, z) = \phi''(1)KL(x, z)$, where $KL(x, z)$ is the Kullback–Leibler distance,

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

Then we have

$$\phi''(1)\left(KL(\hat{x}, x^k) - KL(\hat{x}, x^{k+1})\right) \geq f(x^k) - f(\hat{x}). \tag{6.17}$$

The Hellinger distance,

$$d(x, z) = H(x, z) = \sum_{j=1}^{J} (\sqrt{x_j} - \sqrt{z_j})^2,$$

fits into this framework, as does the reversed KL distance,

$$d(x, z) = KL(z, x). \tag{6.18}$$

6.7.3 Conditions on $\phi(t)$

The required conditions on the function $\phi(t)$ are as follows: $\phi : \mathbb{R} \to (-\infty, +\infty]$ is lower semi-continuous, proper and convex, with dom $\phi \subseteq \mathbb{R}_+$, and dom $\partial\phi = \mathbb{R}_{++}$. In addition, the function ϕ is C^2, strictly convex, and nonnegative on $\mathbb{R}_{++}$, with $\phi(1) = \phi'(1) = 0$, and

$$\phi''(1)\left(1 - \frac{1}{t}\right) \leq \phi'(t) \leq \phi''(1)\log(t). \tag{6.19}$$

For the Hellinger case we have $\phi(t) = (\sqrt{t} - 1)^2$, so that these conditions are satisfied and we have

$$KL(\hat{x}, x^k) - KL(\hat{x}, x^{k+1}) \geq 2\left(f(x^k) - f(\hat{x})\right). \tag{6.20}$$

For the reversed KL distance we have $\phi(t) = t - 1 - \log t$, which also satisfies the required conditions.

6.8 PMA with the Hellinger Distance

We consider now the PMA with the Hellinger distance. According to [2] the Hellinger distance has an induced proximal distance, which turns out to be half the KL distance.

For $s > 0$ and $t > 0$ the Hellinger distance from s to t is

$$h(s,t) = (\sqrt{s} - \sqrt{t})^2. \tag{6.21}$$

With

$$\phi(x) = (\sqrt{x} - 1)^2, \tag{6.22}$$

we have

$$h(s,t) = t\phi(s/t). \tag{6.23}$$

Since, for all $c > 0$, we have

$$2(c - b)(1 - \sqrt{a/b}) \leq KL(c,a) - KL(c,b) = c\log\frac{b}{a} + a - b, \tag{6.24}$$

it follows from the theory in [2] that the Hellinger distance has half the KL distance as its induced proximal distance. Therefore, the PMA with the Hellinger distance is in the SUMMA2 class.

Chapter 7

PMA with Bregman Distances (PMAB)

Let $\mathcal{H}$ be a Hilbert space, and $h : \mathcal{H} \to \mathbb{R}$ strictly convex and Gâteaux differentiable. The *Bregman distance* associated with h is

$$D_h(x, y) = h(x) - h(y) - \langle \nabla h(y), x - y \rangle. \tag{7.1}$$

Proximal minimization with Bregman distances (PMAB) applies to the minimization of a convex function $f : \mathcal{H} \to \mathbb{R}$. In [30, 31] Censor and Zenios discuss in detail the PMAB methods, which they call proximal minimization with D-functions.

7.1 All PMAB are in SUMMA

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

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

where

$$\partial f(x) = \{u | f(y) - f(x) - \langle \nabla u, y - x \rangle \geq 0, \text{for all}\, y\}$$

is the subdifferential of f at x. In [21] it was shown that for the PMAB methods we have $u^k \in \partial f(x^k)$ such that

$$G_k(x) - G_k(x^k) = f(x) - f(x^k) - \langle u^k, x - x^k \rangle + D_h(x, x^k) \geq g_{k+1}(x), \tag{7.2}$$

for all x. Consequently, the SUMMA Inequality holds and all PMAB algorithms are in the SUMMA class. Since the KL distance is a Bregman distance, the iterative algorithm in which we obtain x^k by minimizing $f(x) + KL(x, x^{k-1})$ is in the SUMMA class.

Notice, however, that the algorithm in which we obtain x^k by minimizing $f(x) + KL(x^{k-1}, x)$ is not in the PMAB class; the order of the entries matters here. Nevertheless, since the distance $d(x, x^{k-1}) = KL(x^{k-1}, x)$ is of the form given by Equation (6.16), for the function

$$\phi(t) = t - 1 - \log t,$$

which does satisfy the conditions in Subsection 6.7.3, this algorithm is in the SUMMA2 class.

7.2 Convergence of the PMAB

Because all PMAB algorithms are in the SUMMA class, we know that the sequence $\{f(x^k)\} \downarrow \beta = \inf_x f(x)$. From the inequality in (6.3) we have

$$D_h(x, x^{k-1}) - D_h(x, x^k) \geq f(x^k) - f(x), \tag{7.3}$$

for all x. If there is $\hat{x}$ such that $f(x) \geq f(\hat{x})$, for all x, then

$$D_h(\hat{x}, x^{k-1}) - D_h(\hat{x}, x^k) \geq f(x^k) - f(\hat{x}) \geq 0, \tag{7.4}$$

for all k. Therefore, the sequence $\{D_h(\hat{x}, x^k)\}$ is decreasing. If the Bregman distance $D_h(z, \cdot)$ has bounded level sets, then the sequence $\{x^k\}$ is bounded, there is a cluster point of the sequence, call it x^*, and $f(x^*) = f(\hat{x})$. Replacing $\hat{x}$ with x^*, we find that the sequence $\{D_h(x^*, x^k)\}$ is decreasing. Under reasonable assumptions on D_h [31, 25] it will follow that a subsequence converges to zero, the entire sequence converges to zero, and the sequence $\{x^k\}$ converges to x^*.

The following will be of some interest later. Summing over $1 \leq k \leq N$ on both sides of (7.4), we get

$$D_h(\hat{x}, x^0) - D_h(\hat{x}, x^N) \geq N \left(\min_{1 \leq k \leq N} \{f(x^k)\} - f(\hat{x}) \right), \tag{7.5}$$

so that

$$\frac{1}{N} D_h(\hat{x}, x^0) \geq \min_{1 \leq k \leq N} \{f(x^k)\} - f(\hat{x}). \tag{7.6}$$

7.3 Simplifying the Calculations in PMAB

The iterative step of a PMAB algorithm is to minimize $f(x) + D_h(x, x^{k-1})$ to get x^k. We then have to solve the equation

$$\nabla(f + h)(x^k) = \nabla h(x^{k-1}) \tag{7.7}$$

for x^k. Unless h is selected with some care, solving Equation (7.7) can be difficult. Here is a “trick” to simplify the calculation.

Suppose that g and $h = g - f$ are such that h is convex and the equation

$$\nabla g(x^k) = \nabla h(x^{k-1}) = \nabla g(x^{k-1}) - \nabla f(x^{k-1}) \tag{7.8}$$

is easily solved. Said another way, we minimize

$$f(x) + D_g(x, x^{k-1}) - D_f(x, x^{k-1}) \tag{7.9}$$

to get x^k. In the next few subsections we give several examples of the use of this “trick” . Later, in our discussion of the SMART algorithm, we will show that it too is an example of this “trick” .

7.4 The Quadratic Upper Bound Principle

In [6] the authors introduce the *quadratic upper bound principle* as a method for obtaining a majorizing function in optimization transfer. The objective is to minimize the function $f : \mathbb{R}^J \to \mathbb{R}$. If f is twice continuously differentiable, then, for any x and z, we have, according to the extended Mean Value Theorem,

$$f(x) = f(z) + \langle \nabla f(z), x - z \rangle + \frac{1}{2}(x - z)^T \nabla^2 f(w)(x - z), \tag{7.10}$$

for some w on the line segment connecting x and z. If there is a positive-definite matrix B such that $B - \nabla^2 f(w)$ is positive-definite for all w, then we have

$$f(x) \leq f(z) + \langle \nabla f(z), x - z \rangle + \frac{1}{2}(x - z)^T B(x - z). \tag{7.11}$$

Then we have $g(x|z) \geq f(x)$, for all x and z, where

$$g(x|z) \doteq f(z) + \langle \nabla f(z), x - z \rangle + \frac{1}{2}(x - z)^T B(x - z). \tag{7.12}$$

The iterative step is now to minimize $g(x|x^{k-1})$ to get x^k.

The iterative step is equivalent to minimizing

$$G_k(x) = f(x) + \frac{1}{2}(x - x^{k-1})^T B(x - x^{k-1}) - D_f(x, x^{k-1}), \tag{7.13}$$

which is quite similar to the "trick"introduced in the previous section. However, it is not precisely the same, since the authors of [6] do not assume that f is convex, so this is not a particular case of PMAB. Unless f is convex, we cannot assert that this iteration is in the SUMMA class, so we cannot be sure that the iteration reduces $\{f(x^k)\}$ to the infimal value β. This approach also relies on the extended mean value theorem, while our "trick" permits us considerable freeedom in the selection of the function g.

7.5 Gradient Descent

Say that the operator ∇f is L-Lipschitz continuous if, for all x and z, we have

$$\|\nabla f(x) - \nabla f(z)\| \leq L\|x - z\|. \tag{7.14}$$

If $0 < \gamma < \frac{1}{L}$, then the function $g(x) = \frac{1}{2\gamma}\|x\|^2 - f(x)$ is convex. Having found x^{k-1}, we minimize

$$f(x) + \frac{1}{2\gamma}\|x - x^{k-1}\|^2 - D_f(x, x^{k-1}) \tag{7.15}$$

to get

$$x^k = x^{k-1} - \gamma\nabla f(x^{k-1}), \tag{7.16}$$

which is a version of the gradient descent algorithm.

7.6 The Landweber Algorithm

We want to minimize $f(x) = \frac{1}{2}\|Ax - b\|^2$. This function is L-Lipschitz continuous for $L = \rho(A^TA)$, the largest eigenvalue of the matrix A^TA. Therefore, the function $g(x) = \frac{1}{2\gamma}\|x\|^2 - f(x)$ is convex, for $0 < \gamma < \frac{1}{L}$. Having calculated x^{k-1}, we minimize

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

$$= f(x) + \frac{1}{2\gamma}\|x - x^{k-1}\|^2 - \frac{1}{2}\|Ax - Ax^{k-1}\|^2 \tag{7.17}$$

to get

$$x^k = x^{k-1} - \gamma A^T(Ax^{k-1} - b), \tag{7.18}$$

which is the Landweber algorithm.

7.7 B-SMART

In [62] the authors consider the problem of minimizing $f(x) = D_\phi(Px, y)$, where P is a nonnegative matrix, y a positive vector, and $\sigma_j = \sum_{i=1}^{I} P_{i,j} > 0$, for all j. Their algorithm is called the B-SMART algorithm.

They assume that there is a second Bregman distance D_φ such that $cD_\varphi(x, z) \geq D_\phi(Px, Pz)$, for all x and z. Having calculated x^{k-1}, they minimize

$$f(x^{k-1}) + \langle \nabla f(x^{k-1}), x - x^{k-1}\rangle + \frac{c}{t}D_\varphi(x, x^{k-1}) \tag{7.19}$$

to get x^k, with $0 < t \leq 1$. This is equivalent to minimizing

$$f(x) + \frac{c}{t}D_\varphi(x, x^{k-1}) - D_f(x, x^{k-1}). \tag{7.20}$$

Noting that

$$D_f(x, z) = D_\phi(Px, Pz), \tag{7.21}$$

this is equivalent to minimizing

$$f(x) + \frac{c}{t}D_\varphi(x, x^{k-1}) - D_\phi(Px, Px^{k-1}). \tag{7.22}$$

Since this method is in the PMAB class, Equation (14) of [62] follows immediately from the inequality in (7.6) above.

7.8 A Question

Suppose that we obtain x^k by minimizing

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

where $D_h(x, z)$ is some Bregman distance. If $\{x^k\}$ converges to some x^*, then x^* minimizes $f(x)$ over all x in the closure of the domain of h. Let M be the set of all x that minimize $f(x)$ over the closure of the domain of h; then x^* is a member of M. Does x^* minimize $h(x)$ over x in M? Probably not, since D_h does not determine a unique h. However, it may happen that x^* minimizes $D_h(x, x^0)$ over all x in M. For which Bregman distances does this hold?

Chapter 8

Incorporating Constraints

In this chapter we consider the various ways in which iterative algorithms can be modified to incorporate constraints.

8.1 AF Methods with Constraints

We assume now that $C \subseteq X$ is a nonempty subset of an arbitrary set X, that $f : X \to \mathbb{R}$, and we want to minimize $f(x)$ over x in C. As discussed previously in Section 5.2, the iterative step of a general AF algorithm is to minimize $f(x) + g_k(x)$ over x in X to get x^k in X. There are several ways to impose the constraint:

1. simply to minimize $f(x) + g_k(x)$ over x in C;

2. select as the auxiliary functions $g_k(x)$ functions defined only over x in C;

3. replace $f(x)$ with $f(x) + \iota_C(x)$, where $\iota_C(x) = 0$ if x is in C, and $\iota_C(x) = +\infty$ if x is not in C;

4. replace $g_k(x)$ with $g_k(x) + \iota_C(x)$.

When $X = \mathbb{R}^J$ and f is differentiable, replacing $f(x)$ with $f(x)+\iota_C(x)$ destroys differentiability. In the next section we consider a method to deal with this situation.

8.2 The Forward-Backward Splitting Methods

The *forward-backward splitting* (FBS) methods discussed by Combettes and Wajs [36] form a particular subclass of the PMAB methods. The problem now is to minimize the function $f(x) = f_1(x) + f_2(x)$, where both $f_1 : \mathcal{H} \to (-\infty, +\infty]$ and $f_2 : \mathcal{H} \to (-\infty, +\infty]$ are lower semicontinuous, proper and convex, and f_2 is Gâteaux differentiable, with L-Lipschitz continuous gradient. Before we describe the FBS algorithm we need to recall Moreau's proximity operators.

Following Combettes and Wajs [36], we say that the *Moreau envelope* of index $\gamma > 0$ of the closed, proper, convex function $f : \mathcal{H} \to (-\infty, \infty]$, or the Moreau envelope of the function γf, is the continuous, convex function

$$\mathrm{env}_{\gamma f}(x) = \inf_{y \in \mathcal{H}} \{f(y) + \frac{1}{2\gamma}||x - y||^2\}; \tag{8.1}$$

see also Moreau [57, 58, 59]. In Rockafellar's book [64] and elsewhere, it is shown that the infimum is attained at a unique y, usually denoted $\mathrm{prox}_{\gamma f}(x)$. Proximity operators generalize the orthogonal projections onto closed, convex sets. Consider the function $f(x) = \iota_C(x)$, the *indicator function* of the closed, convex set C, taking the value zero for x in C, and $+\infty$ otherwise. Then $\mathrm{prox}_{\gamma f}(x) = P_C(x)$, the orthogonal projection of x onto C. The following characterization of $x = \mathrm{prox}_f(z)$ is quite useful: $x = \mathrm{prox}_f(z)$ if and only if $z - x \in \partial f(x)$.

In [36] the authors show, using the characterization of $\mathrm{prox}_{\gamma f}$ given above, that x is a solution of this minimization problem if and only if

$$x = \mathrm{prox}_{\gamma f_1}(x - \gamma \nabla f_2(x)). \tag{8.2}$$

This suggests to them the following FBS iterative scheme:

$$x^k = \mathrm{prox}_{\gamma f_1}(x^{k-1} - \gamma \nabla f_2(x^{k-1})). \tag{8.3}$$

Basic properties and convergence of the FBS algorithm are then developed in [36].

8.3 Convergence of the FBS algorithm

Let $f : \mathbb{R}^J \to \mathbb{R}$ be convex, with $f = f_1 + f_2$, both convex, f_2 differentiable, and ∇f_2 L-Lipschitz continuous. Let $\{x^k\}$ be defined by Equation (8.3) and let $0 < \gamma \leq 1/L$.

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

$$G_k(x) = f(x) + \frac{1}{2\gamma}\|x - x^{k-1}\|_2^2 - D_{f_2}(x, x^{k-1}), \tag{8.4}$$

where

$$D_{f_2}(x, x^{k-1}) = f_2(x) - f_2(x^{k-1}) - \langle \nabla f_2(x^{k-1}), x - x^{k-1} \rangle. \tag{8.5}$$

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

The auxiliary function

$$g_k(x) = \frac{1}{2\gamma}\|x - x^{k-1}\|_2^2 - D_{f_2}(x, x^{k-1}) \tag{8.6}$$

can be rewritten as

$$g_k(x) = D_h(x, x^{k-1}), \tag{8.7}$$

where

$$h(x) = \frac{1}{2\gamma}\|x\|_2^2 - f_2(x). \tag{8.8}$$

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

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

$$\langle \nabla h(x) - \nabla h(y), x - y \rangle \geq 0, \tag{8.9}$$

for all x and y. This is equivalent to

$$\frac{1}{\gamma}\|x - y\|_2^2 - \langle \nabla f_2(x) - \nabla f_2(y), x - y \rangle \geq 0. \tag{8.10}$$

Since ∇f_2 is L-Lipschitz, the inequality (8.10) holds for $0 < \gamma \leq 1/L$.

Lemma 8.1 *The x^k that minimizes $G_k(x)$ over x is given by Equation (8.3).*

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

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

or, equivalently,

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

Consequently,

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

■

Theorem 8.1 *The sequence $\{x^k\}$ converges to a minimizer of the function $f(x)$, whenever minimizers exist.*

Proof: A relatively simple calculation shows that

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

$$\Big(f_1(x) - f_1(x^k) - \frac{1}{\gamma}\langle(x^{k-1} - \gamma\nabla f_2(x^{k-1})) - x^k, x - x^k\rangle\Big). \qquad (8.11)$$

Since

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

it follows that

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

Therefore,

$$G_k(x) - G_k(x^k) \geq \frac{1}{2\gamma}\|x - x^k\|_2^2 \geq g_{k+1}(x). \qquad (8.12)$$

Therefore, the inequality in (6.2) holds and the iteration fits into the SUMMA class.

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

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

so that

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

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

From

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

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

Replacing the generic $\hat{x}$ with x^{**}, we find that $\{G_k(x^{**}) - G_k(x^k)\}$ is decreasing to zero. From the inequality in (8.12), we conclude that the sequence $\{\|x^* - x^k\|_2^2\}$ converges to zero, and so $\{x^k\}$ converges to x^*. This completes the proof of the theorem. ∎

A number of well known iterative algorithms are particular cases of the FBS. We consider now some of these algorithms.

8.4 Projected Gradient Descent

Let C be a nonempty, closed convex subset of $\mathbb{R}^J$ and $f_1(x) = \iota_C(x)$, the function that is $+\infty$ for x not in C and zero for x in C. Then $\iota_C(x)$ is convex, but not differentiable. We have $\text{prox}_{\gamma f_1} = P_C$, the orthogonal projection onto C. The iteration in Equation (8.3) becomes

$$x^k = P_C\left(x^{k-1} - \gamma \nabla f_2(x^{k-1})\right). \tag{8.13}$$

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

8.5 The CQ Algorithm and Split Feasibility

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

$$f_2(x) = \frac{1}{2}\|P_Q Ax - Ax\|^2 \tag{8.14}$$

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

$$\nabla f_2(x) = A^T(I - P_Q)Ax. \tag{8.15}$$

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

$$x^k = P_C\left(x^{k-1} - \gamma A^T(I - P_Q)Ax^{k-1}\right); \tag{8.16}$$

this iterative method was called the CQ-algorithm in [18, 19]. The sequence $\{x^k\}$ converges to a solution whenever f_2 has a minimum on the set C, for $0 < \gamma \leq 1/L$.

If $Q = \{b\}$, then the CQ algorithm becomes the *projected Landweber* algorithm [5]. If, in addition, $C = \mathbb{R}^J$, then we get the Landweber algorithm [52]. In [32, 33] Yair Censor and his colleagues modified the CQ algorithm and applied it to derive protocols for intensity-modulated radiation therapy (IMRT). More recently, the CQ algorithm has been modified and applied to proton-beam therapy [61].

Chapter 9

AM with the Euclidean Distance

9.1 Definitions

In this chapter we illustrate the use of AM to derive an iterative algorithm to minimize the function $f(x) = \|b - Ax\|^2$, where A is an I by J real matrix and b an I by 1 real vector. Let R be the set of all I by J arrays r with entries $r_{i,j}$ such that $\sum_{j=1}^{J} r_{i,j} = b_i$, for each i. Let Q be the set of all I by J arrays of the form $q(x)$, where $q(x)_{i,j} = A_{i,j}x_j$. For any vectors u and v with the same size define

$$E(u,v) = \sum_n (u_n - v_n)^2. \tag{9.1}$$

9.2 Pythagorean Identities

We begin by minimizing $E(r, q(x))$ over all $r \in R$. We have the following proposition.

Proposition 9.1 *For all x and r we have*

$$E(r, q(x)) = E(r(x), q(x)) + E(r, r(x)), \tag{9.2}$$

where

$$r(x)_{i,j} = A_{i,j}x_j + \frac{1}{J}(b_i - Ax_i). \tag{9.3}$$

Therefore, $r = r(x)$ is the minimizer of $E(r, q(x))$.

Now we minimize $E(r(x), q(z))$ over z. We have the following proposition.

Proposition 9.2 *For all x and z we have*

$$E(r(x), q(z)) = E(r(x), q(Lx)) + \sum_{j=1}^{J} c_j (Lx_j - z_j)^2, \tag{9.4}$$

where $c_j = \sum_{i=1}^{I} A_{i,j}^2$ and

$$(Lx)_j = Lx_j \doteq x_j + \frac{1}{Jc_j} \sum_{i=1}^{I} A_{i,j}(b_i - Ax_i). \tag{9.5}$$

We omit the proofs of these propositions, which are not deep, but involve messy calculations. Note that

$$\|b - Ax\|^2 = f(x) = JE(r(x), q(x)). \tag{9.6}$$

9.3 The AM Iteration

The iterative step of the algorithm is then

$$x_j^k = Lx_j^{k-1} = x_j^{k-1} + \frac{1}{Jc_j} \sum_{i=1}^{I} A_{i,j}(b_i - Ax_i^{k-1}). \tag{9.7}$$

Applying (9.2) and (9.4) we obtain

$$f(x^{k-1}) = JE(r(x^{k-1}), q(x^{k-1})) = JE(r(x^{k-1}), q(x^k)) + J \sum_{j=1}^{J} c_j (x_j^k - x_j^{k-1})^2$$

$$= JE(r(x^k), q(x^k)) + JE(r(x^{k-1}), r(x^k)) + J \sum_{j=1}^{J} c_j (x_j^k - x_j^{k-1})^2$$

$$= f(x^k) + JE(r(x^{k-1}), r(x^k)) + J \sum_{j=1}^{J} c_j (x_j^k - x_j^{k-1})^2.$$

Therefore,

$$f(x^{k-1}) - f(x^k) = JE(r(x^{k-1}), r(x^k)) + J \sum_{j=1}^{J} c_j (x_j^k - x_j^{k-1})^2 \geq 0,$$

or

$$f(x^{k-1}) - f(x^k) \geq J \sum_{j=1}^{J} c_j (x_j^k - x_j^{k-1})^2 \geq 0, \tag{9.8}$$

from which it follows that the sequence $\{f(x^k)\}$ is decreasing and the sequence $\{\sum_{j=1}^{J} c_j (x_j^k - x_j^{k-1})^2\}$ converges to zero.

The inequality in (9.8) is the *First Monotonicity Property* for the Euclidean case. Since the sequence $\{E(b, Ax^k)\}$ is decreasing, the sequences $\{Ax^k\}$ and $\{x^k\}$ are bounded; let x^* be a cluster point of the sequence $\{x^k\}$. Since the sequence $\{\sum_{j=1}^{J} c_j (x_j^k - x_j^{k-1})^2\}$ converges to zero, it follows that $x^* = Lx^*$.

9.4 Useful Lemmas

We now present several useful lemmas.

Lemma 9.1 *For all x and z we have*

$$E(r(x), r(z)) = \sum_{j=1}^{J} c_j (x_j - z_j)^2 - \frac{1}{J} \sum_{i=1}^{I} (Ax_i - Az_i)^2. \tag{9.9}$$

Lemma 9.2 *For all x and z we have*

$$\frac{1}{J} \sum_{i=1}^{I} (Ax_i - Az_i)^2 \geq \frac{1}{J^2} \sum_{j=1}^{J} \frac{1}{c_j} \left(\sum_{i=1}^{I} A_{i,j} (Ax_i - Az_i) \right)^2. \tag{9.10}$$

Proof: Use Cauchy's Inequality. ∎

Lemma 9.3 *For all x and z we have*

$$E(r(x), r(z)) \geq \sum_{j=1}^{J} c_j (Lx_j - Lz_j)^2. \tag{9.11}$$

It follows from these lemmas that this iterative algorithm is in the SUMMA2 class; for any x we have

$$J \sum_{j=1}^{J} c_j (Lx_j - x_j^k)^2 - J \sum_{j=1}^{J} c_j (Lx_j - x_j^{k+1})^2$$

$$\geq f(x^k) - f(x) + J\sum_{j=1}^{J} c_j(Lx_j - x_j)^2. \tag{9.12}$$

Consequently, the sequence $f(x^k)\}$ converges to the minimum of the function $f(x)$, which must then be $f(x^*)$, and $\{x^k\}$ must converge to x^*.

9.5 Characterizing the Limit

The following proposition characterizes the limit x^*.

Proposition 9.3 *The choice of $\hat{x} = x^*$ minimizes the distance $\sum_{j=1}^{J} c_j(\hat{x}_j - x_j^0)^2$ over all minimizers $\hat{x}$ of $f(x) = \|b - Ax\|^2$.*

Proof: Let $\hat{x}$ be an arbitrary minimizer of $f(x)$. Using the Pythagorean identities we find that

$$JE(r(x^k), q(\hat{x})) = f(\hat{x}) + J\sum_{j=1}^{J} c_j(A\hat{x}_i - Ax^k)_i)^2 - \sum_{i=1}^{I}(A\hat{x}_i - Ax_i^k)^2,$$

and

$$JE(r(x^k), q(\hat{x})) = f(x^{k+1}) + JE(r(x^k), r(x^{k+1})) + J\sum_{j=1}^{J} c_j(\hat{x}_j - x_j^{k+1})^2.$$

Therefore,

$$J\sum_{j=1}^{J} c_j(\hat{x}_j - x_j^k)^2 - J\sum_{j=1}^{J} c_j(\hat{x}_j - x_j^{k+1})^2$$

$$= f(x^{k+1}) - f(\hat{x}) + JE(r(x^k), r(x^{k+1})) + \sum_{i=1}^{I}(A\hat{x}_i - Ax_i^k)^2.$$

Note that the right side of the last equation depends only on $A\hat{x}$ and not directly on $\hat{x}$ itself; therefore the same is true of the left side. Now we sum both sides over the index k to find that $\sum_{j=1}^{J} c_j(\hat{x}_j - x_j^0)^2 - \sum_{j=1}^{J} c_j(\hat{x}_j - x_j^*)^2$ does not depend directly on the choice of $\hat{x}$. The assertion of the proposition follows. ∎

9.6 SUMMA for the Euclidean Case

To get x^k we minimize

$$G_k(x) = JE(r(x^{k-1}), q(x))$$
$$= JE(r(x), q(x)) + \big(JE(r(x^{k-1}), q(x)) - JE(r(x), q(x))\big)$$
$$= f(x) + g_k(x),$$

where

$$g_k(x) = \big(JE(r(x^{k-1}), q(x)) - JE(r(x), q(x))\big) = JE(r(x^{k-1}), r(x)).$$

From (9.9) we have

$$g_k(x) = J\sum_{j=1}^{J} c_j(x_j^{k-1} - x_j)^2 - \sum_{i=1}^{I}(Ax_i^{k-1} - Ax_i)^2. \tag{9.13}$$

From

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

$$JE(r(x^{k-1}), q(x)) - JE(r(x^{k-1}), q(x^k)) = J\sum_{j=1}^{J} c_j(x_j^k - x_j)^2, \tag{9.14}$$

we see that

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

for all x, so that the SUMMA Inequality holds in this case. Therefore, we have

$$g_k(x) - g_{k+1}(x) \geq f(x^k) - f(x),$$

for all x, and so

$$g_k(\hat{x}) - g_{k+1}(\hat{x}) \geq f(x^k) - f(\hat{x}) \geq f(x^k) - f(x^{k+1}). \tag{9.15}$$

This is the *Second Monotonicity Property* for the Euclidean case.

9.7 Using the Landweber Algorithm

It is of some interest to consider an alternative approach, using the Landweber (LW) algorithm. The iterative step of the LW algorithm is

$$x_j^k = x_j^{k-1} + \gamma\sum_{i=1}^{I} A_{i,j}(b_i - Ax_i^{k-1}), \tag{9.16}$$

where $0 < \gamma < \frac{2}{\rho(A^TA)}$. We define $\beta_j = \frac{1}{Jc_j}$, $B_{i,j} = \sqrt{\beta_j}A_{i,j}$, and $z_j = x_j/\sqrt{\beta_j}$. Then $Bz = Ax$. The LW algorithm, applied to $Bz = b$ and with $\gamma = 1$, is

$$z^k = z^{k-1} + B^T(b - Bz^{k-1}). \tag{9.17}$$

Since the trace of B^TB is one, the choice of $\gamma = 1$ is allowed. It is known that the LW algorithm converges to the minimizer of $\|b - Bz\|$ for which $\|z - z^0\|$ is minimized. Converting back to the original x^k, we find that we get the same iterative sequence that we got using the AM method. Moreover, we find once again that the sequence $\{x^k\}$ converges to the minimizer x^* of $f(x)$ for which the distance $\sum_{j=1}^J c_j(\hat{x}_j - x_j^0)^2$ is minimized over all minimizers $\hat{x}$ of $f(x)$.

The Landweber algorithm applied to the original problem of minimizing $f(x) = \|Ax - b\|^2$ has the iterative step

$$x^k = x^{k-1} - \gamma A^T(Ax^{k-1} - b), \tag{9.18}$$

where $0 < \gamma < \frac{2}{\rho(A^TA)}$. The sequence $\{x^k\}$ converges to the minimizer x^* of $f(x)$ that minimizes $\|\hat{x} - x^0\|$ over all minimizers $\hat{x}$ of $f(x)$.

Chapter 10

The SMART and the EMML Algorithms

In this chapter we present the tandem development of the SMART and the EMML algorithms, as originally published in [12].

10.1 The Problem to be Solved

We assume that y is a positive vector in $\mathbb{R}^I$, P an I by J matrix with nonnegative entries $P_{i,j}$, $s_j = \sum_{i=1}^I P_{i,j} > 0$, and we want to find a nonnegative solution or approximate solution x for the linear system of equations $y = Px$. The EMML algorithm will minimize $KL(y, Px)$, while the SMART will minimize $KL(Px, y)$, over $x \geq 0$. For notational simplicity we shall assume that the system has been normalized so that $s_j = 1$ for each j.

10.2 The SMART Iteration

The SMART algorithm [38, 67, 29, 10, 12] minimizes the function $f(x) = KL(Px, y)$, over nonnegative vectors x. Having found the vector

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

$$x_j^k = x_j^{k-1} \exp\Big(\sum_{i=1}^{I} P_{ij} \log(y_i/(Px^{k-1})_i)\Big). \tag{10.1}$$

The iterative step of the SMART can be decsribed as $x^k = Sx^{k-1}$, where S is the operator defined by

$$(Sx)_j = x_j \exp\Big(\sum_{i=1}^{I} P_{ij} \log(y_i/(Px)_i)\Big). \tag{10.2}$$

In our proof of convergence of the SMART we will show that any cluster point x^* of the SMART sequence $\{x^k\}$ is a fixed point of the operator S. To avoid pathological cases in which $Px_i^* = 0$ for some index i, we can assume, at the outset, that all the entries of P are positive. This is wise, in any case, since the model of $y = Px$ is unlikely to be exactly accurate in applications.

10.3 The EMML Iteration

The EMML algorithm minimizes the function $f(x) = KL(y, Px)$, over nonnegative vectors x. Having found the vector x^{k-1}, the next vector in the EMML sequence is x^k, with entries given by

$$x_j^k = x_j^{k-1}\Big(\sum_{i=1}^{I} P_{ij}(y_i/(Px^{k-1})_i)\Big). \tag{10.3}$$

The iterative step of the EMML algorithm can be described as $x^k = Mx^{k-1}$, where M is the operator defined by

$$(Mx)_j = x_j\Big(\sum_{i=1}^{I} P_{ij}(y_i/(Px)_i)\Big). \tag{10.4}$$

As we shall see, the EMML algorithm forces the sequence $\{KL(y, Px^k)\}$ to be decreasing. It follows that $(Px^*)_i > 0$, for any cluster point x^* and for all i.

10.4 The SMART as AM

In [10] the SMART was derived using the following alternating minimization (AM) approach. Let $\mathcal{X}$ be the set of all nonnegative x for which Px has only positive entries; all positive x are in $\mathcal{X}$.

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

$$r(x)_{ij} = x_j P_{ij} y_i / (Px)_i, \tag{10.5}$$

and

$$q(x)_{ij} = x_j P_{ij}. \tag{10.6}$$

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

$$G_k(x) = KL(q(x), r(x^{k-1})) = \sum_{i=1}^{I} \sum_{j=1}^{J} KL(q(x)_{ij}, r(x^{k-1})_{ij}) \tag{10.7}$$

over $x \geq 0$. Note that $f(x) = KL(Px, y) = KL(q(x), r(x))$. We have the following helpful *Pythagorean identities*:

$$KL(q(x), r(z)) = KL(q(x), r(x)) + KL(x, z) - KL(Px, Pz); \tag{10.8}$$

and

$$KL(q(x), r(z)) = KL(q(Sz), r(z)) + KL(x, Sz). \tag{10.9}$$

Note that it follows from Equation (2.4) that $KL(x, z) - KL(Px, Pz) \geq 0$.

From the Pythagorean identities we find that x^k is obtained by minimizing

$$G_k(x) = KL\left(q(x), r(x^{k-1})\right) =$$

$$KL(Px, y) + KL(x, x^{k-1}) - KL(Px, Px^{k-1}), \tag{10.10}$$

so that

$$g_k(x) = KL(x, x^{k-1}) - KL(Px, Px^{k-1}). \tag{10.11}$$

Then

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

Therefore, the SMART is in the SUMMA class. It follows from our discussion of the SUMMA Inequality that, for all $x \geq 0$,

$$g_k(x) + f(x) \geq g_{k+1}(x) + f(x^k). \tag{10.12}$$

Since

$$\sum_{j=1}^{J} x_j^k \leq \sum_{i=1}^{I} y_i,$$

the sequence $\{x^k\}$ is bounded and has a cluster point, x^*, with $f(x^k) \geq f(x^*)$ for all k. With $x = x^*$ in (10.12), we obtain

$$D_h(x^*, x^{k-1}) - D_h(x^*, x^k) \geq f(x^k) - f(x^*) \geq 0.$$

Therefore, the sequence $\{f(x^k)\}$ converges to $f(x^*)$. Since the SMART is in SUMMA, we know that $f(x^*)$ must be the minimum of $f(x)$. Since a subsequence of $\{D_h(x^*, x^k)\}$ converges to zero, it follows that $\{x^k\}$ converges to x^*.

Let $\hat{x}$ be any minimizer of $KL(Px, y)$. Using the Pythagorean identites we find that

$$KL(\hat{x}, x^k) - KL(\hat{x}, x^{k+1}) = KL(Px^{k+1}, y) - KL(P\hat{x}, y) + KL(P\hat{x}, Px^k) + KL(x^{k+1}, x^k) - KL(Px^{k+1}, Px^k). \quad (10.13)$$

From Equation (10.13) we see that the difference $KL(\hat{x}, x^k) - KL(\hat{x}, x^{k+1})$ depends only on $P\hat{x}$, and not on $\hat{x}$ itself. Summing over the index k on both sides and "telescoping" , we find that the difference $KL(\hat{x}, x^0) - KL(\hat{x}, x^*)$ also depends only on $P\hat{x}$, and not on $\hat{x}$ itself. It follows that $\hat{x} = x^*$ is the minimizer of $f(x)$ for which $KL(\hat{x}, x^0)$ is minimized. If $y = Px$ has nonnegative solutions, and the entries of x^0 are all equal to one, then x^* maximizes the Shannon entropy over all nonnegative solutions of $y = Px$.

With $f(x) = KL(Px, y)$, we have $D_f(x, z) = KL(Px, Pz)$. Therefore, we obtain the next iterate x^k by minimizing

$$G_k(x) = KL\left(q(x), r(x^{k-1})\right) = f(x) + KL(x, x^{k-1}) - D_f(x, x^{k-1}) (10.14)$$

This shows that the SMART is yet another example of the "trick" used to obtain PMAB algorithms with iterates that can be simply calculated.

The following theorem summarizes the situation with regard to the SMART [10, 11, 12].

Theorem 10.1 *In the consistent case, in which the system $y = Px$ has nonnegative solutions, the sequence of iterates of SMART converges to the unique nonnegative solution of $y = Px$ for which the distance $KL(x, x^0)$ is minimized. In the inconsistent case it converges to the unique nonnegative minimizer of the distance $KL(Px, y)$ for which $KL(x, x^0)$ is minimized. In the inconsistent case, if P and every matrix derived from P by deleting columns has full rank then there is a unique nonnegative minimizer of $KL(Px, y)$ and at most $I - 1$ of its entries are nonzero.*

10.5 The EMML as AM

Now we want to minimize $f(x) = KL(y, Px)$. The iterative step of the EMML algorithm is obtained by minimizing

$$G_k(x) = KL(r(x^{k-1}), q(x)) \tag{10.15}$$

to get x^k. We have the following helpful *Pythagorean identities*:

$$KL(r(x), q(z)) = KL(r(z), q(z)) + KL(r(x), r(z)); \tag{10.16}$$

and

$$KL(r(x), q(z)) = KL(r(x), q(Mx)) + KL(Mx, z). \tag{10.17}$$

From the Pythagorean identities we have

$$KL(y, Px^k) - KL(y, Px^{k+1}) =$$

$$KL(r(x^k), r(x^{k+1})) + KL(x^{k+1}, x^k), \tag{10.18}$$

so that

$$KL(y, Px^k) - KL(y, Px^{k+1}) \geq KL(x^{k+1}, x^k). \tag{10.19}$$

The inequality in (10.19) is called the *First Monotonicity Property* in [41]. We also have

$$G_k(x) = KL(r(x), q(x)) + KL(r(x^{k-1}, r(x)) = f(x) + d(x, x^{k-1}), \tag{10.20}$$

and

$$G_k(x) = f(x) + g_k(x), \tag{10.21}$$

with

$$d(x, x^{k-1}) = g_k(x) = KL(r(x^{k-1}), q(x)) - KL(r(x), q(x)). \tag{10.22}$$

Therefore, the EMML algorithm is an AF algorithm, so that $\{f(x^k)\}$ is decreasing. The EMML algorithm appears not to be a member of the SUMMA subclass; however, as we shall see shortly, it is a member of the SUMMA2 subclass.

Lemma 10.1 *For $\{x^k\}$ given by Equation (10.3), the sequence $\{KL(y, Px^k)\}$ is decreasing and the sequences $\{KL(x^{k+1}, x^k)\}$ and $\{KL(r(x^k), r(x^{k+1}))\}$ converge to zero.*

Lemma 10.2 *The EMML sequence $\{x^k\}$ is bounded; for $k \geq 1$ we have*

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

Using (2.4) we obtain the following useful inequality:

$$KL(r(x), r(z)) \geq KL(Mx, Mz). \tag{10.23}$$

From

$$\begin{aligned} KL(r(x), q(x^k)) &= KL(r(x^k), q(x^k)) + KL(r(x), r(x^k)) \\ &\geq f(x^k) + KL(Mx, x^{k+1}), \end{aligned}$$

and

$$\begin{gathered} KL(r(x), q(x^k)) = KL(r(x), q(Mx)) + KL(Mx, x^k) = \\ f(x) - KL(Mx, x) + KL(Mx, x^k) \end{gathered}$$

we have

$$KL(Mx, x^k) - KL(Mx, x^{k+1}) \geq f(x^k) - f(x) + KL(Mx, x). \tag{10.24}$$

Note that we have used (10.23) here. Therefore, the EMML is in the SUMMA2 class. With x^* a cluster point, we have

$$KL(Mx^*, x^k) - KL(Mx^*, x^{k+1}) \geq f(x^k) - f(x^*) \geq 0. \tag{10.25}$$

Therefore, the sequence $\{KL(Mx^*, x^k)\}$ is decreasing, and the sequence $\{f(x^k)\}$ converges to $f(x^*)$. Since the EMML is in the SUMMA2 class, we know that $f(x^*)$ is the minimum value of $f(x)$ and $Mx^* = x^*$.

Let $\hat{x}$ be a minimizer of $f(x) = KL(y, Px)$. Inserting $x = \hat{x}$ into Equation (10.24), we obtain

$$KL(\hat{x}, x^k) - KL(\hat{x}, x^{k+1}) \geq KL(y, Px^k) - KL(y, Px^{k+1}). \tag{10.26}$$

The inequality in (10.26) is called the *Second Monotonicity Property* in [41].

The following theorem summarizes the situation with regard to the EMML algorithm [10, 11, 12].

Theorem 10.2 *In the consistent case, in which the system $y = Px$ has nonnegative solutions, the sequence of EMML iterates converges to a nonnegative solution of $y = Px$. In the inconsistent case it converges to a nonnegative minimizer of the distance $KL(y, Px)$. In the inconsistent case, if P and every matrix derived from P by deleting columns has full rank then there is a unique nonnegative minimizer of $KL(y, Px)$ and at most $I - 1$ of its entries are nonzero.*

In contrast to the SMART, we cannot characterize the limit in terms of the starting vector x^0.

Chapter 11

Acceleration Using KL Projections

For large values of I and J convergence of the SMART and EMML algorithms, as well as other simultaneous algorithms, can be quite slow. In this chapter we consider the use of block-iterative and sequential methods to accelerate the creation of useful images. Our experience with the ART and MART algorithms tells us that these block-iterative algorithms should converge in the consistent case, that is, when there is a nonnegative solution of $y = Px$, but when no such solution exists, the iterates should cycle among the vectors of a limit cycle.

11.1 Rescaled Block-Iterative SMART (RBI-SMART)

In the unnormalized case, in which $s_j = \sum_{i=1}^{I} P_{i,j}$ is positive, but not necessarily equal to one, the iterative step for SMART given in Equation (10.1) becomes

$$x_j^k = x_j^{k-1} \exp\left(s_j^{-1} \sum_{i=1}^{I} P_{ij} \log\left(\frac{y_i}{(Px^{k-1})_i} \right) \right), \tag{11.1}$$

or,

$$x_j^k = x_j^{k-1} \prod_{i=1}^{I} \left(\frac{y_i}{(Px^{k-1})_i} \right)^{s_j^{-1} P_{i,j}}. \tag{11.2}$$

We can also write SMART as

$$\log x_j^k = \log x_j^{k-1} + \left(s_j^{-1} \sum_{i=1}^{I} P_{ij} \log \left(\frac{y_i}{(Px^{k-1})_i} \right) \right), \tag{11.3}$$

The iterative step of the MART is

$$x_j^k = x_j^{k-1} \left(\frac{y_i}{(Px^{k-1})_i} \right)^{m_i^{-1} P_{i,j}}, \tag{11.4}$$

where $i = (k-1)(\bmod I) + 1$ and

$$m_i = \max\{P_{i,j} | j = 1, ..., J\}, \tag{11.5}$$

which we can also write as

$$\log x_j^k = \log x_j^{k-1} + m_i^{-1} P_{i,j} \log \left(\frac{y_i}{(Px^{k-1})_i} \right). \tag{11.6}$$

In [29] the authors offer a block-iterative variant of SMART and MART. The idea here is to decompose the set $\{i = 1, 2, ..., I\}$ into the union of N not necessarily disjoint subsets, $B_1, ..., B_N$, and then to mimic Equation (11.2), but to multiply only over the indices in the current subset. The *rescaled block-iterative* SMART (RBI-SMART) is a slightly modified version of the block-iterative algorithm in [29]. With $n = (k-1)(\bmod N) + 1$, $s_{n,j} = \sum_{i \in B_n} P_{i.j}$ and

$$m_n = \max\{s_{n,j} s_j^{-1}\}, \tag{11.7}$$

the iterative step of the RBI-SMART is

$$x_j^k = x_j^{k-1} \prod_{i \in B_n} \left(\frac{y_i}{(Px^{k-1})_i} \right)^{m_n^{-1} s_j^{-1} P_{i,j}}, \tag{11.8}$$

which we can write as

$$\log x_j^k = \log x_j^{k-1} + \left(m_n^{-1} s_j^{-1} \sum_{i \in B_n} P_{ij} \log \left(\frac{y_i}{(Px^{k-1})_i} \right) \right). \tag{11.9}$$

The objective now is to define analogous block-iterative variants of the EMML algorithm.

11.2 The Rescaled Block-Iterative EMML (RBI-EMML)

In the unnormalized case, in which $s_j = \sum_{i=1}^{I} P_{i,j}$ is positive, but not necessarily equal to one, the iterative step for the EMML algorithm given in Equation (10.3) becomes

$$x_j^k = x_j^{k-1}\left(s_j^{-1}\sum_{i=1}^{I} P_{ij}\left(\frac{y_i}{(Px^{k-1})_i}\right)\right). \tag{11.10}$$

In [48] the authors offered an accelerated variant of the EMML called the “ordered-subset”EM (OSEM) algorithm. The idea here is to decompose the set $\{i = 1, 2, ..., I\}$ into the union of N not necessarily disjoint subsets, $B_1, ..., B_N$, and then to mimic Equation (11.10), but to sum only over the indices in the current subset. At the kth step of the OSEM we have

$$x_j^k = x_j^{k-1}\left(s_{n,j}^{-1}\sum_{i\in B_n} P_{ij}\left(\frac{y_i}{(Px^{k-1})_i}\right)\right). \tag{11.11}$$

At first glance the OSEM seems to be the proper generalization of the EMML; in the RBI-SMART case we multiplied only over the indices in B_n and now we add only over the indices in B_n. It was observed that the OSEM produced useful images much quicker than did the EMML. It was to be expected that, for $N > 1$, the OSEM algorithm would not converge to a single image when the system $y = Px$ is inconsistent. However, it was observed that the OSEM could also fail to converge in the consistent case; the authors of [48] proved convergence for the consistent case only under a quite restrictive condition, called “subset balance”. It turned out that this behavior of the OSEM was due to the absence in OSEM of a second term [13].

The correct algorithm, called the “rescaled block-iterative” EMML (RBI-EMML) [13], has the iterative step

$$x_j^k = \left(1 - \frac{s_{n,j}}{m_n s_j}\right)x_j^{k-1} + \frac{1}{m_n s_j}x_j^{k-1}\left(\sum_{i\in B_n} P_{i,j}\left(\frac{y_i}{(Px^{k-1})_i}\right)\right). \tag{11.12}$$

The RBI-EMML converges to a solution in the consistent case, for any choice of blocks and for any starting vector.

Note that, if $s_{n,j} = t_j u_n$, then $m_n = u_n/u_+$, for $u_+ = \sum_{n=1}^N u_n$, $s_j = t_j u_+$, $m_n^{-1}s_j^{-1}s_{n,j} = 1$, and the RBI-EMML reduces to OSEM. In [48] it was shown that OSEM converges to a solution in the consistent case whenever the “subset balance” condition, $s_{n,j} = t_j$ for all n, holds, which means, in effect, whenever it is an RBI-EMML iteration. Subset balance is highly unlikely and almost impossible to achieve in practice; in particular, it would almost certainly force all the subsets to have the same number of indices, which is not necessarily desirable.

It may seem that the new term in the RBI-EMML is simply pulled out of a hat. After all, how can you know what should be there when it isn’t yet there? In fact, the added term appears quite naturally when the close connection between the EMML and SMART algorithms is considered. The key is KL projection onto hyperplanes.

11.3 KL Projections onto Hyperplanes

For notational simplicity, we shall assume once again that $s_j = 1$, for all j. For each $i = 1, 2, ..., I$, let H_i be the hyperplane

$$H_i = \{z|(Pz)_i = y_i\}. \tag{11.13}$$

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

$$\sum_{j=1}^{J} P_{ij} KL(z_j, x_j) \tag{11.14}$$

is $T_i(x)$ given by

$$T_i(x)_j = x_j \left(\frac{y_i}{(Px)_i} \right). \tag{11.15}$$

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

11.4 Reformulating SMART and EMML

The iterative step of the SMART algorithm,as given in Equation (10.1), can be expressed as

$$x_j^k = \prod_{i=1}^{I} (T_i(x^{k-1})_j)^{P_{ij}}. \tag{11.16}$$

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

The iterative step of the EMML algorithm, as given in Equation (10.3), can be expressed as

$$x_j^k = \sum_{i=1}^{I} P_{ij} T_i(x^{k-1})_j. \tag{11.17}$$

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

A correct block-iterative variant of the EMML was presented in [13]; block-iterative variants of the SMART, such as the MART, were already known [45, 29].

11.5 The MART and EMART Algorithms

The MART algorithm has the iterative step

$$x_j^k = x_j^{k-1}(y_i/(Px^{k-1})_i)^{P_{ij}m_i^{-1}}, \tag{11.18}$$

where $i = (k-1)(\bmod I) + 1$ and

$$m_i = \max\{P_{ij}|j = 1, 2, ..., J\}. \tag{11.19}$$

When there are nonnegative solutions of the system $y = Px$, the sequence $\{x^k\}$ converges to the solution x that minimizes $KL(x, x^0)$ [13, 14, 15]. We can express the MART in terms of the weighted KL projections $T_i(x^{k-1})$;

$$x_j^k = (x_j^{k-1})^{1-P_{ij}m_i^{-1}}(T_i(x^{k-1})_j)^{P_{ij}m_i^{-1}}. \tag{11.20}$$

We see then that the iterative step of the MART is a relaxed weighted KL projection onto H_i, and a weighted geometric mean of the current x_j^{k-1} and $T_i(x^{k-1})_j$. The expression for the MART in Equation (11.20) suggests a somewhat simpler iterative algorithm involving a weighted arithmetic mean of the current x_j^{k-1} and $T_i(x^{k-1})_j$; this is the EMART algorithm.

The iterative step of the EMART algorithm is

$$x_j^k = (1 - P_{ij}m_i^{-1})x_j^{k-1} + P_{ij}m_i^{-1}T_i(x^{k-1})_j. \tag{11.21}$$

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

11.6 RBI-SMART and RBI-EMML

As we just saw, the MART and EMART involve either weighted geometric or weighted arithmetic relaxation. The iterative step of the RBI-SMART can be expressed as

$$\log x_j^k = \left(1 - \frac{s_{n,j}}{m_n s_j}\right) \log x_j^{k-1} + \frac{1}{m_n s_j} \sum_{i \in B_n} P_{i,j} \log T_i(x^{k-1})_j. \quad (11.22)$$

This suggests that the block-iterative variant of the EMML should be

$$x_j^k = \left(1 - \frac{s_{n,j}}{m_n s_j}\right) x_j^{k-1} + \frac{1}{m_n s_j} \sum_{i \in B_n} P_{i,j} T_i(x^{k-1})_j. \quad (11.23)$$

Both the RBI-SMART and the RBI-EMML converge to a nonnegative solution of $y = Px$, not necessarily the same solution, whenever such solutions exist, for any choice of blocks.

Chapter 12

Why Are Block-Iterative Methods Faster?

We have made the claim, and experience has shown, that in the consistent case, block-iterative methods can converge significantly faster than their simultaneous relatives. In this chapter we investigate this claim a bit more theoretically. The arguments given here are not completely rigorous, but will give some idea of the source of the acceleration. Our goal is to get orders-of-magnitude estimates, not precise values. We begin by comparing the simultaneous Landweber algorithm with the sequential ART algorithm for solving the general system of linear equations $Ax = b$. Then we compare the simultaneous SMART with the sequential MART for solving the nonnegative system $Px = y$.

12.1 The Landweber and Cimmino Algorithms

Let $Az = b$ be a consistent system of linear equations, with $\sum_{j=1}^J A_{i,j}^2 = 1$, for each $j = 1, ..., J$. The iterative step of the Landweber algorithm is

$$x^{k+1} = x^k + \gamma A^T(b - Ax^k), \tag{12.1}$$

where $0 < \gamma < \frac{2}{L}$ for $L = \rho(A^TA)$, the largest eigenvalue of the matrix A^TA. We know that $1 \leq L \leq I$.

Simple calculations show that, for any z with $Az = b$,

$$\|z - x^k\|^2 - \|z - x^{k+1}\|^2 \geq (2\gamma - L\gamma^2)\|b - Ax^k\|^2. \tag{12.2}$$

The trace of A^TA is I, so the choice of $\gamma = \frac{1}{I}$ is acceptable. With this choice of γ we get Cimmino's algorithm:

$$x^{k+1} = x^k + \frac{1}{I} A^T (b - Ax^k), \tag{12.3}$$

and

$$\|z - x^k\|^2 - \|z - x^{k+1}\|^2 \geq (2/I - L/I^2)\|b - Ax^k\|^2. \tag{12.4}$$

The improvement we obtain in Equations (12.2) and (12.4) will depend L, and the choice of γ.

If we know L, which is probably not the case, especially for large systems, we may select $\gamma = \frac{1}{I}$, just to be safe; this is Cimmino's choice. If we have a better upper bound for L than just I, then we can use it in the choice of γ. For example, it was shown in [22] that, whenever the rows of A are normalized to length one, L cannot be larger than the maximum number of nonzero entries in any column of A. This is useful in the case of sparse A. In transmission tomography there are typically about $\sqrt{I}$ nonzero entries in a column, so the estimate $L \leq \sqrt{I}$ is usually acceptable. If $L = 1$ and we choose $\gamma = 1$, then Equation (12.2) becomes

$$\|z - x^k\|^2 - \|z - x^{k+1}\|^2 \geq \|b - Ax^k\|^2. \tag{12.5}$$

However, if L is closer to I than to 1 the choice of $\gamma = \frac{1}{I}$ will give us something more like

$$\|z - x^k\|^2 - \|z - x^{k+1}\|^2 \geq \frac{1}{I}\|b - Ax^k\|^2. \tag{12.6}$$

12.2 The ART

The iterative step of the ART is

$$x_j^{k+1} = x_j^k + A_{i,j}(b_i - (Ax^k)_i), \tag{12.7}$$

where $i = (k-1)(\text{mod}\, I) + 1$. We consider the improvement we obtain after one pass through all the data. For any z with $Az = b$ we have

$$\|z - x^0\|^2 - \|z - x^I\|^2 = \sum_{i=1}^{I} (b_i - (Ax^{i-1})_i)^2. \tag{12.8}$$

This is, very roughly, about I times the improvement in Equation (12.6).

12.3 The SMART

For SMART we assume that $s_j = \sum_{i=1}^{I} P_{i,j} = 1$, for each j. Then, with $y = Pz$, Equation (10.13) tells us that

$$KL(z, x^k) - KL(z, x^{k+1}) \approx KL(Px^{k+1}, y). \tag{12.9}$$

12.4 The MART

With $m_i = \max\{P_{i,j} | j = 1, ..., J\}$, and $y = Pz$ we have

$$KL(z, x^0) - KL(z, x^1) \approx m_1^{-1} KL(y_1, (Px^0)_1). \tag{12.10}$$

Since $s_j = 1$, we might estimate $m_1 \approx \frac{1}{I}$. Therefore, after one pass through all the data, we have

$$KL(z, x^0) - KL(z, x^I) \approx I\, KL(y, Px^{i-1}), \tag{12.11}$$

for some representative i. The point is that the improvement we may expect after one pass through the data may well be a factor of I larger than that obtained by one SMART iteration. Of course, if the entries of P are not more or less uniformly distributed, the m_i may well be greater than $\frac{1}{I}$ and the improvement after one pass through the data may well be somewhat less than before. In the sparse case, in which there are, say, only $\sqrt{I}$ nonnegative entries in any column, the m_i will be more like $\frac{1}{\sqrt{I}}$ and the improvement will be only a factor of $\sqrt{I}$ better than SMART. Since, in many applications, I is in the thousands, even this reduced improvement is significant.

Chapter 13

Regularization

The "night sky" phenomenon that occurs in nonnegatively constrained least-squares also happens with methods based on the Kullback-Leibler distance, such as MART, EMML and SMART, requiring some sort of regularization.

13.1 The "Night-Sky" Problem

As we saw previously, the sequence $\{x^k\}$ generated by the EMML iterative step in Equation (10.3) converges to a nonnegative minimizer $\hat{x}$ of $f(x) = KL(y, Px)$, and we have

$$\hat{x}_j = \hat{x}_j \sum_{i=1}^{I} P_{ij} \frac{y_i}{(P\hat{x})_i}, \tag{13.1}$$

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

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

$$1 = \sum_{i=1}^{I} P_{ij} = \sum_{i=1}^{I} P_{ij} \frac{y_i}{(P\hat{x})_i}. \tag{13.2}$$

Now let Q be the I by K matrix obtained from P by deleting rows j for which $\hat{x}_j = 0$. If Q has full rank and $K \geq I$, then Q^T is one-to-one, so that $1 = \frac{y_i}{(P\hat{x})_i}$ for all i, or $y = P\hat{x}$. But we are assuming that there is no nonnegative solution of $y = Px$. Consequently, we must have $K < I$ and $I - K$ of the entries of $\hat{x}$ are zero. This behavior is not restricted to the KL distance and occurs also in nonnegative least squares.

A simple picture helps to give a feel for what is going on here. Imagine an unopened umbrella. The metal ribs of the umbrella are the columns of the matrix P. Any vector of the form Px, for nonnegative x, is a nonnegative linear combination of the columns of P, so is on the surface of the umbrella or inside the umbrella. If the vector y is not on or inside the umbrella, then when we find the closest vector on or inside the umbrella, that closest vector to y cannot be inside; it must be on the surface of the umbrella. The vectors on the surface of the umbrella are linear combinations of just a few columns of P, that is, they lie on a face of the surface formed by just a few of the metal ribs. Therefore, when we write this closest vector as Px, the only x_j that are positive are those whose index j corresponds to those columns of P that we view as the ribs that form that face.

13.2 Regularizing SMART and EMML

As discussed in [10, 11], we can regularize the SMART algorithm by minimizing the function

$$(1-\alpha)KL(q(x), r(x^{k-1})) + \alpha KL(x, p), \tag{13.3}$$

where $p \geq 0$ is chosen a priori, perhaps as a prior estimate of the desired x, and $0 < \alpha < 1$. The resulting iterative step is

$$x_j^k = (Sx_j^{k-1})^{1-\alpha} p_j^\alpha. \tag{13.4}$$

We regularize EMML by minimizing

$$(1-\alpha)KL(r(x^{k-1}), q(x)) + \alpha KL(p, x). \tag{13.5}$$

The resulting iterative step is

$$x_j^k = (1-\alpha)(Mx^{k-1})_j + \alpha p_j. \tag{13.6}$$

By placing the variable x in the same position in both terms we are able to obtain a closed-form expression for the iterative step in each case.

13.3 More on Regularization

Simultaneous iterative methods such as the Landweber algorithm converge to a least squares solution when applied to an inconsistent system, that is, they minimize $\|Ax-b\|$, which means solving $A^TAx = A^Tb$. When the matrix A is ill-conditioned the resulting least-squares apporoximate solution may not be suitable. A better approximate solution can be found by using regularization. When A is ill-conditioned the least-squares solution may have an unrealistically large norm, prompting the introduction of some form of norm constraint. For example, we can minimize

$$\|Ax-b\|^2 + \gamma^2\|x\|^2.$$

The system to be solved now is $(A^TA + \gamma^2 I)x = A^Tb$, which is consistent.

Sequential iterative algorithms, such as ART and the various block-iterative variants, cannot converge to a single vector when the system is inconsistent. Instead, they exhibit subsequential convergence to a limit cycle (LC) consisting of (typically) as many distinct vectors as there are blocks. In [28] it was shown that the LC can be avoided and the least-squares solution approximated through the use of a small relaxation parameter. This suggests the use of updating of the relaxation parameter as the iteration proceeds. However, as noted in [69], convergence to the least-squares solution can be quite slow. In [46] it is mentioned that selecting the "right"update can be challenging. It is our objective now to provide methods for selecting the updates, based on previous work on how particular relaxation parameters affect the data error.

As we noted previously, the system to be solved when we regularize is the consistent system

$$(A^TA + \gamma^2 I)x = A^Tb.$$

We denote by $\hat{x}_\gamma$ the regularized solution. When the system is large, we want to avoid having to calculate A^TA and we want to use iterative methods. We discuss two methods for using ART to obtain regularized solutions of $Ax = b$. The first one is presented in [20], while the second one is due to Eggermont, Herman, and Lent [40]. It would be of some interest to find a similar approach for regularizing MART.

Both methods rely on the fact that when the ART is applied to a consistent system $Ax = b$ it converges to the solution of that system closest to where we began the iteration.

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

$$B^Tz = \begin{bmatrix} A^T & \gamma I \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = 0. \tag{13.7}$$

We begin with $u^0 = b$ and $v^0 = 0$. This system is consistent. Then, the lower component of the limit vector is $v^\infty = -\gamma\hat{x}_\gamma$. We know that with $c = \begin{bmatrix} b \\ 0 \end{bmatrix}$, we have

$$c = B\hat{x}_\gamma + z,$$

where $B^T z = 0$ and z is the vector in the null space of B^T closest to c. If we had tried to solve the inconsistent system $Bx = c$ we would get a limit cycle.

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

$$\begin{bmatrix} A & \gamma I \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} = b. \tag{13.8}$$

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

We know that $b = A\hat{x} + \hat{w}$, where $\hat{w}$ is the member of the null space of A^T closest to b. One way to avoid the limit cycle in ART in the inconsistent case is to apply ART twice. First, we solve the consistent system $A^T w = 0$, beginning at $w^0 = b$, to get $\hat{w}$. Then we solve the consistent system $Ax = b - \hat{w}$ to get $A\hat{x}$. It would also be of interest to find a similar approach for avoiding the limit cycle in MART.

Chapter 14

Modifying the KL Distance

The SMART, EMML and their block-iterative versions are based on the Kullback-Leibler distance between nonnegative vectors and require that the solution sought be a nonnegative vector. To impose more general constraints on the entries of x we derive algorithms based on shifted KL distances, also called Fermi-Dirac generalized entropies.

14.1 Fermi–Dirac Entropies

For a fixed real vector u, the shifted KL distance $KL(x-u, z-u)$ is defined for vectors x and z having $x_j \geq u_j$ and $z_j \geq u_j$. Similarly, the shifted distance $KL(v-x, v-z)$ applies only to those vectors x and z for which $x_j \leq v_j$ and $z_j \leq v_j$. For $u_j \leq v_j$, the combined distance

$$KL(x-u, z-u) + KL(v-x, v-z)$$

is restricted to those x and z whose entries x_j and z_j lie in the interval $[u_j, v_j]$. Our objective is to mimic the derivation of the SMART, EMML and BI methods, replacing KL distances with shifted KL distances, to obtain algorithms that enforce the constraints $u_j \leq x_j \leq v_j$, for each j. The algorithms that result are the ABMART and ABEMML block-iterative methods. These algorithms were originally presented in [16], in which the vectors u and v were called a and b, hence the names of the algorithms. As previously, we shall assume that the entries of the matrix P are nonnegative. We shall denote by B_n, $n = 1, ..., N$ a partition of the index set $\{i = 1, ..., I\}$ into blocks. For $k = 0, 1, ...$ let $n(k) = k(\text{mod}\, N) + 1$.

No iterates of the EMML and SMART algorithms can have $x_j = 0$; zero values of x_j can only occur in the limit. In certain medical imaging problems we are interested in locating “cold spots” with no uptake of radionuclide. It is helpful, in such cases, to modify the EMML and SMART to permit $x_j = 0$ prior to the limit. The algorithms described in this section were used in [60] to solve this kind of imaging problem.

14.2 Using Prior Bounds on x_j

For a fixed real vector u, the shifted KL distance $KL(x - u, z - u)$ is defined for vectors x and z having $x_j \geq u_j$ and $z_j \geq u_j$. Similarly, the shifted distance $KL(v - x, v - z)$ applies only to those vectors x and z for which $x_j \leq v_j$ and $z_j \leq v_j$. For $u_j \leq v_j$, the combined distance

$$KL(x - u, z - u) + KL(v - x, v - z)$$

is restricted to those x and z whose entries x_j and z_j lie in the interval $[u_j, v_j]$. Our objective is to mimic the derivation of the SMART and EMML methods, replacing KL distances with shifted KL distances, to obtain algorithms that enforce the constraints $u_j \leq \lambda_j \leq v_j$, for each j. The algorithms that result are the ABMART and ABEMML block-iterative methods. These algorithms were originally presented in [16], in which the vectors u and v were called a and b, hence the names of the algorithms. We shall assume that the entries of the matrix P are nonnegative. We shall denote by B_n, $n = 1, ..., N$ a partition of the index set $\{i = 1, ..., I\}$ into blocks. For $k = 0, 1, ...$ let $n = n(k) = k(\text{mod}\, N) + 1$.

14.2.1 The ABMART Algorithm

We assume that $(Pu)_i \leq y_i \leq (Pv)_i$ and seek a solution of $Px = y$ with $u_j \leq x_j \leq v_j$, for each j. The algorithm begins with an initial vector x^0 satisfying $u_j \leq x_j^0 \leq v_j$, for each j. Having calculated x^k, we take

$$x_j^{k+1} = \alpha_j^k v_j + (1 - \alpha_j^k) u_j, \tag{14.1}$$

with $n = n(k)$,

$$\alpha_j^k = \frac{c_j^k \prod^n (d_i^k)^{P_{ij}}}{1 + c_j^k \prod^n (d_i^k)^{A_{ij}}}, \tag{14.2}$$

$$c_j^k = \frac{(x_j^k - u_j)}{(v_j - x_j^k)}, \tag{14.3}$$

and

$$d_j^k = \frac{(y_i - (Pu)_i)((Pv)_i - (Px^k)_i)}{((Pv)_i - y_i)((Px^k)_i - (Pu)_i)}, \tag{14.4}$$

where $\prod^n$ denotes the product over those indices i in $B_{n(k)}$. Notice that, at each step of the iteration, x_j^k is a convex combination of the endpoints u_j and v_j, so that x_j^k always lies in the interval $[u_j, v_j]$.

We have the following theorem concerning the convergence of the ABMART algorithm:

Theorem 14.1 *If there is a solution of the system $Px = y$ that satisfies the constraints $u_j \leq x_j \leq v_j$ for each j, then, for any N and any choice of the blocks B_n, the ABMART sequence converges to that constrained solution of $Px = y$ for which the Fermi-Dirac generalized entropic distance from x to x^0, given by*

$$KL(x - u, x^0 - u) + KL(v - x, v - x^0),$$

is minimized. If there is no constrained solution of $Px = y$, then, for $N = 1$, the ABMART sequence converges to the minimizer of

$$KL(Px - Pu, y - Pu) + KL(Pv - Px, Pv - y)$$

for which

$$KL(x - u, x^0 - u) + KL(v - x, v - x^0)$$

is minimized.

The proof is in [16].

14.2.2 The ABEMML Algorithm

We make the same assumptions as previously. The iterative step of the ABEMML algorithm is

$$x_j^{k+1} = \alpha_j^k v_j + (1 - \alpha_j^k) u_j, \tag{14.5}$$

where

$$\alpha_j^k = \gamma_j^k / d_j^k, \tag{14.6}$$

$$\gamma_j^k = (x_j^k - u_j) e_j^k, \tag{14.7}$$

$$\beta_j^k = (v_j - x_j^k) f_j^k, \tag{14.8}$$

$$d_j^k = \gamma_j^k + \beta_j^k, \tag{14.9}$$

$$e_j^k = \left(1 - \sum_{i \in B_n} P_{ij}\right) + \sum_{i \in B_n} P_{ij} \left(\frac{y_i - (Pu)_i}{(Px^k)_i - (Pu)_i}\right), \tag{14.10}$$

and

$$f_j^k = \left(1 - \sum_{i \in B_n} P_{ij}\right) + \sum_{i \in B_n} P_{ij} \left(\frac{(Pv)_i - y_i}{(Pv)_i - (Px^k)_i}\right). \tag{14.11}$$

The following theorem concerns the convergence of the ABEMML algorithm:

Theorem 14.2 *If there is a solution of the system $Px = y$ that satisfies the constraints $u_j \leq x_j \leq v_j$ for each j, then, for any N and any choice of the blocks B_n, the ABEMML sequence converges to such a constrained solution of $Px = y$. If there is no constrained solution of $Px = y$, then, for $N = 1$, the ABEMML sequence converges to a constrained minimizer of*

$$KL(y - Pu, Px - Pu) + KL(Pv - y, Pv - Px).$$

The proof is found in [16]. In contrast to the ABMART theorem, this is all we can say about the limits of the ABEMML sequences.

The projected Landweber algorithm can also be used to impose the restrictions $u_j \leq x_j \leq v_j$; however, the projection step in that algorithm is implemented by clipping, or setting equal to u_j or v_j values of x_j that would otherwise fall outside the desired range. The result is that the values u_j and v_j can occur more frequently than may be desired. One advantage of the AB methods is that the values u_j and v_j represent barriers that can only be reached in the limit and are never taken on at any step of the iteration.

Bibliography

[1] Ahn, S., Fessler, J., Blatt, D., and Hero, A. (2006) "Convergent incremental optimization transfer algorithms: application to tomography."*IEEE Transactions on Medical Imaging*, **25(3)**, pp. 283–296.

[2] Auslender, 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] Bauschke, H., and Borwein, J. (2001) "Joint and separate convexity of the Bregman distance." in [8], pp. 23–36.

[4] Bauschke, H., Combettes, P., and Noll, D. (2006) "Joint minimization with alternating Bregman proximity operators." *Pacific Journal of Optimization*, **2**, pp. 401–424.

[5] Bertero, M., and Boccacci, P. (1998) *Introduction to Inverse Problems in Imaging*, Bristol, UK: Institute of Physics Publishing.

[6] Böhning, D., and Lindsey, B.G. (1988) "Monotonicity of quadratic approximation algorithms." *Ann Instit Stat Math*, **40**, pp. 641–663.

[7] 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.

[8] Butnariu, D., Censor, Y., and Reich, S. (eds.) (2001) *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications*, Studies in Computational Mathematics 8. Amsterdam: Elsevier Publ.

[9] Butnariu, D., Byrne, C., and Censor, Y. (2003) "Redundant axioms in the definition of Bregman functions." *Journal of Convex Analysis*, **10**, pp. 245–254.

[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. (1995) “Erratum and addendum to ‘Iterative image reconstruction algorithms based on cross-entropy minimization’.”*IEEE Transactions on Image Processing* **IP-4**, pp. 225–226.

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

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

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

[15] Byrne, C. (1998) “Accelerating the EMML algorithm and related iterative algorithms by rescaled block-iterative (RBI) methods.”*IEEE Transactions on Image Processing* **IP-7**, pp. 100–109.

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

[17] Byrne, C. (2001) “Likelihood maximization for list-mode emission tomographic image processing.” *IEEE Transactions on Medical Imaging*, **TMI-20(10)**, pp. 1084–1092.

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

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

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

[21] Byrne, C. (2008) “Sequential unconstrained minimization algorithms for constrained optimization.” *Inverse Problems*, **24(1)**, article no. 015013.

[22] Byrne, C. (2009) “Bounds on the largest singular value of a matrix and the convergence of simultaneous and block-iterative algorithms for sparse linear systems.” *International Transactions in Operations Research*, **16(4)**, pp. 465–479.

[23] Byrne, C. (2013) "Alternating minimization as sequential unconstrained minimization: a survey." *Journal of Optimization Theory and Applications*, electronic **154(3)**, DOI 10.1007/s1090134-2, (2012), and hardcopy **156(3)**, February, 2013, pp. 554–566.

[24] Byrne, C. (2014) "An elementary proof of convergence of the forward-backward splitting algorithm." *Journal of Nonlinear and Convex Analysis* **15(4)**, pp. 681–691.

[25] Byrne, C. (2014) *Iterative Optimization in Inverse Problems.* Boca Raton, FL: CRC Press.

[26] Byrne, C. (2015) "Non-stochastic EM algorithms in optimization." to appear in the *Journal of Nonlinear and Convex Analysis.*

[27] Byrne, C., and Lee, J-S. (2015) "Alternating minimization, proximal minimization, and optimization transfer are equivalent."to appear in the *Journal of Nonlinear and Convex Analysis.*

[28] Censor, Y., Eggermont, P., and Gordon, D. (1983) "Strong underrelaxation in Kaczmarz's method for inconsistent systems." *Numer. Math.*, **41**, pp. 83–92.

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

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

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

[32] Censor, Y., Bortfeld, T., Martin, B., and Trofimov, A. "A unified approach for inversion problems in intensity-modulated radiation therapy." *Physics in Medicine and Biology* 51 (2006), 2353–2365.

[33] Censor, Y., Elfving, T., Kopf, N., and Bortfeld, T. (2005) "The multiple-sets split feasibility problem and its application for inverse problems." *Inverse Problems*, **21** , pp. 2071–2084.

[34] Chang, J.T., and Pollard, D. (1997) "Conditioning as disintegration." *Statistica Neerlandia*, **51(3)**, pp. 287–317.

[35] Chi, E., Zhou, H., and Lange, K. (2014) "Distance Majorization and Its Applications." Mathematical Programming, **146 (1-2)**, pp. 409–436.

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

[37] Csiszár, I. and Tusnády, G. (1984) “Information geometry and alternating minimization procedures.”*Statistics and Decisions* **Supp. 1**, pp. 205–237.

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

[39] Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977) “Maximum likelihood from incomplete data via the EM algorithm.”*Journal of the Royal Statistical Society, Series B* **37**, pp. 1–38.

[40] Eggermont, P., Herman, G.T., and Lent, A. (1981) “Iterative algorithms for large partitioned linear systems, with applications to image reconstruction.” *Linear Algebra Appl.*, **40**, pp. 37–67.

[41] Eggermont, P., and LaRiccia, V. (1998) “On EM-like algorithms for minimum distance estimation.”unpublished notes.

[42] Eggermont, P., and LaRiccia, V. (2001) *Maximum Penalized Likelihood Estimation.* New York: Springer.

[43] Erdogan, H., and Fessler, J. (1999) “Monotonic algorithms for transmission tomography.” *IEEE Transactions on Medical Imaging*, **18(9)**, pp. 801–814.

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

[45] Herman, G.T. (1980) *Image Reconstruction from Projections: The Fundamentals of Computed Tomography.* New York: Academic Press.

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

[47] Hogg, R., McKean, J., and Craig, A. (2004) *Introduction to Mathematical Statistics*, 6th edition, Prentice Hall.

[48] 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.

[49] Kaufman, L. (1987) "Implementing and accelerating the EM algorithm for positron emission tomography."*IEEE Transactions on Medical Imaging*, **6(1)**, pp. 37–51.

[50] Kuiper, A., Bredies, K., Pock, Th., and Bischof, H. (eds.) (2013) *Scale Space and Variational Methods in Computer Vision: Proceedings of the 4th International Conference SSVM 2013, Graz, Austria, June 2–6, 2013*, Lecture Notes in Computer Science, Vol. 7893, Springer Verlag.

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

[52] Landweber, L. (1951) "An iterative formula for Fredholm integral equations of the first kind."*Amer. J. of Math.* **73**, pp. 615–624.

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

[54] 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.

[55] Lange, K., Hunter, D., and Yang, I. (2000) "Optimization transfer using surrogate objective functions (with discussion)." *J. Comput. Graph. Statist.*, **9**, pp. 1–20.

[56] McLachlan, G.J. and Krishnan, T. (1997) *The EM Algorithm and Extensions*. New York: John Wiley and Sons, Inc.

[57] Moreau, J.-J. (1962) "Fonctions convexes duales et points proximaux dans un espace hilbertien." *C.R. Acad. Sci. Paris Sér. A Math.*, **255**, pp. 2897–2899.

[58] Moreau, J.-J. (1963) "Propriétés des applications 'prox'." *C.R. Acad. Sci. Paris Sér. A Math.*, **256**, pp. 1069–1071.

[59] Moreau, J.-J. (1965) "Proximité et dualité dans un espace hilbertien." *Bull. Soc. Math. France*, **93**, pp. 273–299.

[60] 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.

[61] Penfold, S., Zalas, R., Casiraghi, M., Brooke, M., Censor, Y., and Schulte, R. (2017) "Sparsity constrained split feasibility for dose-volume constraints in inverse planning of intensity-modulated photon or proton therapy." to appear.

[62] Petra, S., Schnörr, C., Becker, F., and Lenzen, F. (2013) "B-SMART: Bregman-based first-order algorithms for non-negative compressed sensing problems." in [50], pp. 110–124.

[63] Rao, C.R. (1965) *Linear Statistical Inference and its Applications.* New York: John Wiley and Sons.

[64] Rockafellar, R. (1970) *Convex Analysis.* Princeton, NJ: Princeton University Press.

[65] Rockafellar, R.T. and Wets, R. J-B. (2009) *Variational Analysis* (3rd printing), Berlin: Springer-Verlag.

[66] Rockmore, A., and Macovski, A. (1976) "A maximum likelihood approach to emission image reconstruction from projections." *IEEE Transactions on Nuclear Science*, **NS-23**, pp. 1428–1432.

[67] Schmidlin, P. (1972) "Iterative separation of sections in tomographic scintigrams." *Nuklearmedizin* **11**, pp. 1–16.

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

[69] Trummer, M. (1983) "SMART- an algorithm for reconstructing pictures from projections." *J. of Applied Mathematics and Physics (ZAMP)*, **34**, pp. 746–753.

[70] 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.

Index