

Monotonically Overrelaxed EM Algorithms

Yaming Yu

Department of Statistics, University of California, Irvine

June 28, 2011

Abstract

We explore the idea of overrelaxation for accelerating the expectation-maximization (EM) algorithm, focusing on preserving its simplicity and monotonic convergence properties. It is shown that in many cases a trivial modification in the M-step results in an algorithm that maintains monotonic increase in the log-likelihood, but can have an appreciably faster convergence rate, especially when EM is very slow. The method is applicable to more general fixed point algorithms. Its simplicity and effectiveness are illustrated with several statistical problems, including probit regression, least absolute deviations regression, Poisson inverse problems, and finite mixtures. This paper has supplemental materials online.

Keywords: AECM; bound optimization algorithms; ECM; ECME; majorization-minimization; PX-EM; robust regression; SAGE; successive overrelaxation.

1 Introduction

The expectation-maximization (EM) algorithm (Dempster et al., 1977; McLachlan and Krishnan, 1997), with its many extensions such as the ECM (Meng and Rubin, 1993) and ECME (Liu and Rubin, 1994) algorithms, is a powerful tool for maximum likelihood computation in a missing data context, and even in some situations involving no apparent missing data. Part of its appeal lies in the simplicity of the E- and M-steps, and in its monotonic convergence property, i.e., each iteration never decreases the observed data log-likelihood. However, EM is known to be slow sometimes, and various methods have been proposed to accelerate its convergence. These may be roughly classified into two groups, (i) numerical analysis tools including multivariate versions of Aitken's acceleration (Laird, Lange and Stram, 1987), conjugate gradient acceleration (Jamshidian and Jennrich, 1993), quasi-Newton acceleration (Lange, 1995b; Jamshidian and Jennrich, 1997),

and other extrapolation methods (Varadhan and Roland, 2004; Kuroda and Sakakihara, 2006), and (ii) methods derived within the EM framework, including the space-alternating generalized EM (SAGE; Fessler and Hero, 1994), the efficient data augmentation algorithms (Meng and van Dyk, 1997, 1998), and the parameter-expanded EM (PX-EM; Liu, Rubin and Wu, 1998). The former group work for more general iterative algorithms, but do not automatically preserve the monotonic convergence property; the latter involve in-depth inspection of the model structure, and may not apply to certain models. When applicable, however, algorithms based on efficient data augmentation or parameter expansion are often as simple as the original EM, have better rates of convergence, and preserve the monotonicity automatically.

This paper studies a simple acceleration strategy, called *monotonic overrelaxation*, which falls in between these two groups. We employ the successive overrelaxation (SOR) idea from numerical analysis (Young 1971), focusing on preserving the simplicity and stability of EM. The basic idea is to modify the M-step in a way that moves the parameter estimate further away from the current estimate than prescribed by EM, but still increases the expected complete-data log-likelihood (i.e., the Q function). The resulting algorithm is therefore a general EM algorithm (GEM; Dempster et al., 1977), and automatically maintains monotonicity. We show how to design simple overrelaxation schemes with appreciably faster rates of convergence. Although SOR has been considered for accelerating EM in the literature (see, e.g., Lange, 1995a, and Salakhutdinov and Roweis, 2003), a straightforward implementation does not automatically preserve monotonicity. A common recipe is to monitor the log-likelihood and to backtrack when the algorithm overshoots. Our strategies, which often amount to a trivial modification in the M-step, can eliminate such complications, and save both programming efforts and computer time. While the increase in speed may not be as dramatic as when some of the more aggressive accelerators are used, it comes with little extra cost.

After a simple illustration using probit regression (Section 2), we formally define monotonic overrelaxation in the EM framework (Section 3). Theoretical properties, such as monotonicity and convergence rate, are also analyzed in Section 3. We also discuss how it can be applied to more general algorithms such as the minorization-maximization (or majorization-minimization) algorithm (Lange et al., 2000). Section 4 applies monotonic overrelaxation to several examples including least absolute deviations regression, Poisson inverse problems, and finite mixtures. Both

simulations and real data examples are used to demonstrate the effectiveness of monotonic over-relaxation. Section 5 concludes with some general remarks and directions for future work.

2 An illustrative example

Consider the probit regression model

$$y_i = \text{sgn}(\eta_i), \quad \eta_i | (\theta, X) \stackrel{\text{ind}}{\sim} N(X_i \theta, 1),$$

where $X = (X_1^\top, \dots, X_n^\top)^\top$ ($n \times p$) is a full-rank design matrix, θ ($p \times 1$) is the parameter of interest, and we observe the ± 1 vector $y = (y_1, \dots, y_n)^\top$, the signs of the latent variables $\eta = (\eta_1, \dots, \eta_n)^\top$. This model formulation leads to a straightforward EM algorithm, which we call standard EM. The complete-data log-likelihood is $l_c(\theta; X, y, \eta) = -\sum_{i=1}^n (\eta_i - X_i \theta)^2 / 2$. At iteration t , the E-step computes the conditional expectation of $l_c(\theta; X, y, \eta)$ given the current parameter estimate $\theta^{(t)}$ and observed data. That is,

$$Q(\theta | \theta^{(t)}) \equiv E[l_c(\theta; X, y, \eta) | \theta^{(t)}, X, y] = -\sum_{i=1}^n \frac{(\tilde{\eta}_i - X_i \theta)^2 + v_i}{2}, \quad (2.1)$$

where

$$\tilde{\eta}_i \equiv E[\eta_i | \theta^{(t)}, X, y] = \mu_i + \frac{y_i \phi(\mu_i)}{\Phi(y_i \mu_i)}, \quad \mu_i = X_i \theta^{(t)}, \quad v_i = \text{var}(\eta_i | \theta^{(t)}, X, y), \quad (2.2)$$

and ϕ and Φ denote the standard normal density and cumulative distribution function (CDF), respectively. The M-step maximizes $Q(\theta | \theta^{(t)})$ with respect to θ to obtain

$$\theta_{\text{EM}}^{(t+1)} = (X^\top X)^{-1} X^\top \tilde{\eta},$$

where $\tilde{\eta} = (\tilde{\eta}_1, \dots, \tilde{\eta}_n)^\top$ is given by (2.2). The M-step is especially simple, and the observed log-likelihood increases at each iteration.

Monotonically overrelaxed EM (MOEM) performs the same E-step as above. At the M-step, however, we update $\theta^{(t)}$ as

$$\theta_{\text{MOEM}}^{(t+1)} = (1 + w)\theta_{\text{EM}}^{(t+1)} - w\theta^{(t)}, \quad (2.3)$$

where w is an *overrelaxation parameter*. Because (2.1) is a quadratic function of θ , we know that any $w \in [0, 1]$ satisfies

$$Q\left(\theta_{\text{MOEM}}^{(t+1)} \mid \theta^{(t)}\right) \geq Q\left(\theta^{(t)} \mid \theta^{(t)}\right).$$

Basic EM theory says that any θ that increases the Q function also increases the observed data log-likelihood. Thus the iteration $\theta^{(t)} \rightarrow \theta_{\text{MOEM}}^{(t+1)}$ maintains monotonicity of the observed data log-likelihood. Also note that computing $\theta_{\text{MOEM}}^{(t+1)}$ from $\theta_{\text{EM}}^{(t+1)}$ requires minimal additional programming effort and computer time.

Equation (2.3) is the classical successive overrelaxation (SOR) update, originally used to accelerate the Gauss-Seidel algorithm for solving linear systems of equations (Young 1971). The rationale is that EM iterations are conservative, and larger steps at the M-step may increase the speed; see Section 3 for more discussion. We emphasize how easy it is to implement SOR and maintain monotonic increase in the observed log-likelihood in this example.

To illustrate the effect of monotonic overrelaxation, we use real data from van Dyk and Meng (2001) (see their Table 1), which concern two clinical measurements (i.e., covariates) that are used to predict the occurrence of latent membranous lupus nephritis. We fit a probit regression model including both covariates and an intercept (i.e., $p = 3$). There are a total of $n = 55$ subjects. Table 1 records the performance of monotonic overrelaxation with various choices of w . Each algorithm is started at $\theta^{(0)} = (0, 0, 0)^\top$, and is terminated when the maximum absolute value of $\theta^{(t+1)} - \theta^{(t)}$ falls below 10^{-8} . Each algorithm pre-computes the matrix $(X^\top X)^{-1} X^\top$. All calculations are performed on the same Sun Solaris 10 machine running R. Table 1 shows that the number of iterations decreases steadily as w increases. Further reduction (not shown) is even observed when we experiment with $w > 1$, although it is intuitive that setting w too high would overshoot and no longer maintain monotonicity. We also record the computer time as measured by the R function `system.time()`. Not surprisingly, the computer time comparison is similar to comparing iteration counts, since the overrelaxation step is a trivial modification of the M-step. Figure 1 displays the observed log-likelihood for all five algorithms; it is clear that they all maintain monotonicity.

3 Monotonic overrelaxation: the method

We give a formal definition of monotonic overrelaxation in the EM framework (Section 3.1), and investigate its theoretical properties (Section 3.2). Possible extensions to many extensions of EM are outlined in Section 3.3.

Table 1: Iteration count and computer time (in seconds) for monotonically overrelaxed EM for fitting the lupus nephritis data. Standard EM corresponds to $w = 0$.

w	0.0	0.25	0.5	0.75	1.0
iteration count	9642	7857	6645	5766	5098
computer time	1.95	1.76	1.49	1.29	1.14

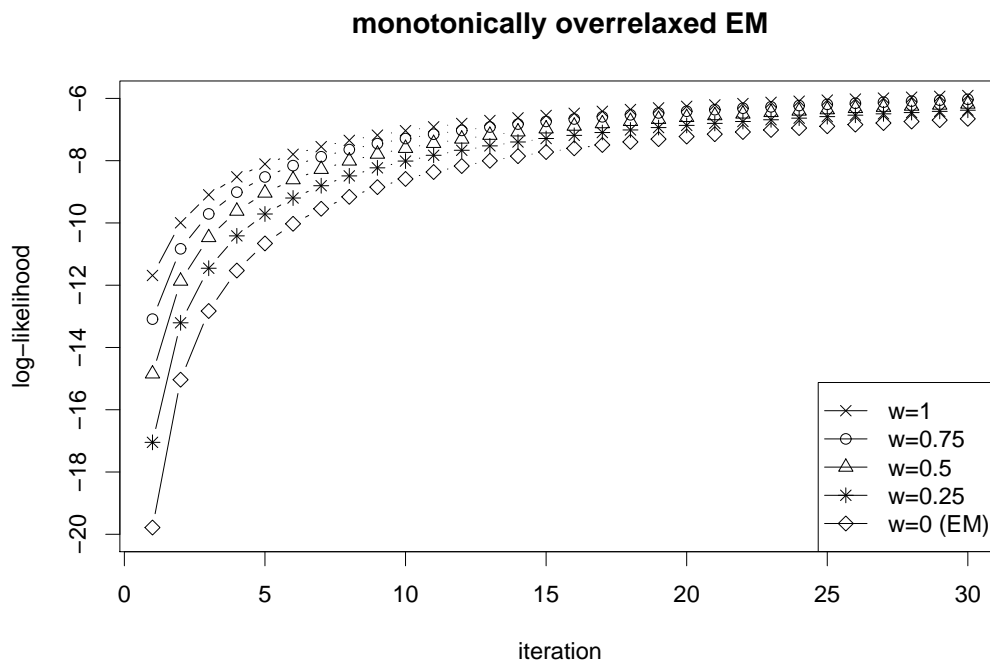


Figure 1: Progression of the log-likelihood for MOEM for fitting the lupus nephritis data.

3.1 A general prescription

Consider a missing data problem where the complete data are denoted by Y , and only a function of Y , denoted by Y_{obs} , is observed. The goal is to estimate the $p \times 1$ parameter θ by maximizing the observed data likelihood $L_o(\theta; Y_{\text{obs}})$. Denote the complete-data likelihood as $L_c(\theta; Y)$. Monotonically overrelaxed expectation-maximization (MOEM) achieves this by iterating between two steps.

Monotonically Overrelaxed EM

E-step This is the same as the E-step of standard EM (Dempster et al., 1977). At iteration t , we take the conditional expectation of $\log L_c(\theta; Y)$ given the current parameter estimate and observed data:

$$Q(\theta | \theta^{(t)}) = E[\log L_c(\theta; Y) | \theta^{(t)}, Y_{\text{obs}}]. \quad (3.1)$$

M-step Find $\theta^{(t+1)}$ such that

$$Q(\theta^{(t+1)} | \theta^{(t)}) \geq Q(\theta^{(t)} | \theta^{(t)}) \quad (3.2)$$

and

$$\theta_i^{(t+1)} = (1 + w_i^{(t)}) \theta_{i,\text{EM}}^{(t+1)} - w_i^{(t)} \theta_i^{(t)}, \quad i = 1, \dots, p, \quad (3.3)$$

where $\theta_{\text{EM}}^{(t+1)}$ is the standard EM update, i.e., the maximizer of $Q(\theta | \theta^{(t)})$ with respect to θ , and $w_i^{(t)} \geq 0$ are overrelaxation parameters that may depend on $\theta^{(t)}$.

Basically, (3.3) requires that $\theta^{(t+1)}$ moves further away from $\theta^{(t)}$ (coordinate-wise) than does the standard EM update $\theta_{\text{EM}}^{(t+1)}$. Standard EM corresponds to $w_i^{(t)} \equiv 0$.

One advantage of EM is that the maximizer of $Q(\theta | \theta^{(t)})$ is often in closed form. Our MOEM generalization may seem to have destroyed this benefit. However, as we shall illustrate with several examples, often finding a family of overrelaxed $\theta^{(t+1)}$ satisfying both (3.2) and (3.3) is as easy as finding the exact maximizer $\theta_{\text{EM}}^{(t+1)}$. Similar to efficient data augmentation or PX-EM, MOEM requires an in-depth inspection of the model structure, but the resulting algorithm is usually a simple modification of standard EM.

3.2 Monotonicity and convergence rate

The theoretical properties of MOEM follow closely those of standard EM. Specifically, (3.2) says that MOEM is a general EM algorithm, each iteration of which maintains monotonicity of the observed data likelihood, i.e.,

$$L_o(\theta^{(t+1)}; Y_{\text{obs}}) \geq L_o(\theta^{(t)}; Y_{\text{obs}}).$$

Proposition 1 summarizes our convergence results concerning MOEM. Basically, under mild conditions, the limiting points of MOEM coincide with those of standard EM. Thus, all results concerning whether standard EM converges to a local maximum or a stationary point can be transferred to MOEM directly from Wu (1983). Let us denote the MOEM mapping by $\theta^{(t+1)} = M(\theta^{(t)})$, and define $\Theta_0 = \{\theta : l_o(\theta) \geq l_o(\theta^{(0)})\}$, where $l_o(\theta) \equiv \log L_o(\theta; Y_{\text{obs}})$ and $\theta^{(0)}$ is the starting value.

Proposition 1. *Assume (a) Θ_0 is a compact subset of the p -dimensional Euclidean space; (b) $l_o(\theta)$ is continuous on Θ_0 ; (c) M is continuous on Θ_0 ; (d) for any $\theta \in \Theta_0$ we have $Q(M(\theta)|\theta) > Q(\theta|\theta)$, unless $M(\theta) = \theta$. Let Γ denote the set of limit points of the MOEM iterations. Then (i) every $\theta^* \in \Gamma$ is a fixed point of the mapping M on Θ_0 ; (ii) every $\theta^* \in \Gamma$ is also a fixed point of the standard EM mapping; (iii) as $t \rightarrow \infty$, $l_o(\theta^{(t)})$ monotonically increases to $l_o(\theta^*)$ for some $\theta^* \in \Gamma$.*

Proof. Part (i) follows from standard arguments (see Wu 1983). To prove Part (ii), we note that, by (3.3), if $\theta^{(t)}$ is a fixed point of M , i.e., $\theta^{(t+1)} = \theta^{(t)}$, then $\theta_{\text{EM}}^{(t+1)} = \theta^{(t)}$, i.e., $\theta^{(t)}$ is also a fixed point of standard EM. Part (iii) holds by (3.2) as mentioned earlier. \square

A crucial requirement in Proposition 1 is condition (d), which is often easily verified. In the probit regression example of Section 2, condition (d) is satisfied if $0 \leq w < 1$. Similar analysis applies to subsequent examples in Section 4. Also note that Proposition 1 only claims convergence of the log-likelihood; convergence of the sequence $\theta^{(t)}$ itself requires further regularity conditions (see Vaida 2005).

We now analyze the convergence rates. Let θ^* be a local maximum of $l_o(\theta)$ in the interior of the parameter space. It is well known that the convergence rate of standard EM is the fraction of missing information

$$R_{\text{EM}} = I_{\text{com}}^{-1} I_{\text{mis}}$$

where

$$I_{\text{com}} = - \left. \frac{\partial^2 Q(\theta|\theta^*)}{\partial\theta\partial\theta^\top} \right|_{\theta=\theta^*}, \quad I_{\text{mis}} = I_{\text{com}} - I_{\text{obs}}, \quad I_{\text{obs}} = - \frac{\partial^2 l_o(\theta^*)}{\partial\theta\partial\theta^\top}.$$

These formulas imply that all eigenvalues of R_{EM} are real and lie in the interval $[0, 1]$. The *global rate* of EM, defined as the spectral radius of R_{EM} , is therefore its maximum eigenvalue. That the eigenvalues are nonnegative can be interpreted as saying that the EM iterations are *conservative*; it is possible to improve its speed by taking larger steps at the M-step, as prescribed by MOEM.

In the definition of MOEM, i.e., equation (3.3), let us assume that the overrelaxation parameters $w_i^{(t)} = w_i(\theta^{(t)})$ are differentiable in $\theta^{(t)}$. Near θ^* , the iteration (3.3) behaves as if $w_i^{(t)}$ is fixed at $w_i^* \equiv w_i(\theta^*)$, and we may calculate the matrix rate of MOEM as

$$R_{\text{MOEM}} \equiv \frac{\partial M(\theta^*)}{\partial\theta} = R_{\text{EM}} + DR_{\text{EM}} - D, \quad D = \text{Diag}(w_1^*, \dots, w_p^*).$$

For simplicity, we assume that $w_i^* \equiv w$ for all i , i.e., D is proportional to the identity. This is not as restrictive as it may seem, because we still allow $w_i^{(t)}$ to depend on i for finite t ; one can often construct overrelaxation schemes such that, in the limit, w_i^* , $i = 1, \dots, p$, are all equal to a pre-specified constant (see Section 4). If D is proportional to the identity, then we can appeal to standard results from the literature of SOR. In particular, every eigenvalue λ of R_{EM} corresponds to an eigenvalue $(1 + w)\lambda - w$ of R_{MOEM} . The global rate of MOEM is therefore

$$\rho(R_{\text{MOEM}}) = \max\{|(1 + w)\lambda_{\#} - w|, |(1 + w)\lambda^{\#} - w|\}, \quad (3.4)$$

where $\lambda_{\#}$ and $\lambda^{\#}$ denote the minimum and maximum eigenvalues of R_{EM} , respectively. It is well known that (3.4) is minimized when $w = (\lambda_{\#} + \lambda^{\#})/(2 - \lambda_{\#} - \lambda^{\#})$. Moreover, this optimal w is positive unless $\lambda_{\#} = \lambda^{\#} = 0$, showing that EM can always be accelerated by some degree of overrelaxation.

Because $\lambda^{\#}$ and $\lambda_{\#}$ are unknown, the practical value of the optimal w is somewhat limited. If standard EM is already fast, then a large w may slow it down due to the potential negative eigenvalues of R_{MOEM} (see Section 4.2 for an example). On the other hand, noting $\lambda_{\#} \geq 0$, we observe that, if $0 \leq w \leq \lambda^{\#}/(2 - \lambda^{\#})$, then $|(1 + w)\lambda_{\#} - w| \leq (1 + w)\lambda^{\#} - w$, and hence

$$1 - \rho(R_{\text{MOEM}}) = (1 + w)(1 - \lambda^{\#}).$$

That is, the *speed* (defined as one minus the rate) of MOEM is $1 + w$ times that of standard EM. The factor $1 + w$ sometimes manifests itself when comparing iteration counts. In Table 1, for example, MOEM with $w = 0.5$ takes about two-thirds as many iterations as standard EM.

If standard EM is very slow, i.e., $\lambda^\#$ is close to one, then the choice

$$w = \frac{\lambda^\#}{2 - \lambda^\#} \tag{3.5}$$

would lead to an algorithm that is about twice as fast as standard EM near the mode. To achieve (3.5), however, we need to know $\lambda^\#$. One possibility is to perform EM iterations at the initial stage of the program, and estimate $\lambda^\#$ by the ratio of successive differences, i.e., $\|\theta^{(t+1)} - \theta^{(t)}\| / \|\theta^{(t)} - \theta^{(t-1)}\|$, where $\|\cdot\|$ denotes the Euclidean norm. A strategy even simpler than (3.5) is to set w at a fixed constant slightly less than one. We expect the resulting algorithm to improve standard EM when standard EM is slow. Of course, we have assumed throughout that it is possible to pre-specify the overrelaxation parameter (at least in the limit). This is nontrivial because we also require that each iteration increases the Q function. We shall illustrate with concrete examples (Section 4) before discussing this further in Section 5.

3.3 Extensions

Minorization-maximization algorithms (MM; Lange et al., 2000), also known as bound optimization algorithms (see, e.g., Salakhutdinov and Roweis, 2003), are extensions of EM that do not require a missing data framework. To maximize $l(\theta)$ (e.g., the log-likelihood), we first find a function $Q(\theta|\tilde{\theta})$ such that $l(\theta) \geq Q(\theta|\tilde{\theta})$ for all θ and $\tilde{\theta}$, with equality when $\theta = \tilde{\theta}$. Given the current $\theta^{(t)}$, the MM algorithm sets $\theta^{(t+1)}$ as the maximizer of $Q(\theta|\theta^{(t)})$ with respect to θ . The iteration $\theta^{(t)} \rightarrow \theta^{(t+1)}$ increases the objective function because

$$l(\theta^{(t+1)}) \geq Q(\theta^{(t+1)}|\theta^{(t)}) \geq Q(\theta^{(t)}|\theta^{(t)}) = l(\theta^{(t)}). \tag{3.6}$$

Monotonic overrelaxation can be extended to MM algorithms. Once the function $Q(\theta|\tilde{\theta})$ is constructed, we use the same prescription, i.e., (3.2) and (3.3) (reading $\theta_{\text{EM}}^{(t+1)}$ for $\theta_{\text{MM}}^{(t+1)}$), to update θ . Because each iteration increases the Q function, monotonicity of $l(\theta)$ is guaranteed, again by (3.6). We refer to these algorithms as monotonically overrelaxed MM (MOMM) algorithms.

The expectation-conditional-maximization algorithm (ECM; Meng and Rubin, 1993) is an extension of EM that replaces the sometimes difficult M-step by a sequence of conditional maximization steps. We can apply monotonic overrelaxation to ECM, replacing each conditional maximization by an overrelaxation step. The expectation-conditional-maximization-either algorithm

(ECME; Liu and Rubin, 1994) replaces the final few conditional maximization steps of ECM by conditionally maximizing the observed data log-likelihood instead of the Q function. The alternating-expectation-conditional-maximization algorithm (AECM; Meng and van Dyk, 1997, 1998) is a further extension; it inserts certain E-steps between the CM steps in ECM so that different CM steps can correspond to different complete-data formulations. Monotonic overrelaxation can also be applied to these extensions of EM. We simply replace each conditional maximization step by a conditional update as prescribed by (3.2) and (3.3), where the Q function is now specific to the CM step. This component-wise strategy is especially attractive when it is difficult to apply overrelaxation to the whole parameter vector jointly.

The PX-EM of Liu et al. (1998) seeks to improve EM by introducing expansion parameters that are identifiable only from the complete data. We illustrate how to apply monotonic overrelaxation to this special EM algorithm in Section 4.1.

Finally, monotonic overrelaxation should be applicable to algorithms based purely on conditional maximization (i.e., no auxiliary variables). Examples include the iterative proportional fitting algorithm (IPF; Deming and Stephan, 1940) for ML estimation in log-linear models (Bishop et al., 1975), and the iterative conditional mode algorithm for Bayesian image restoration. Indeed, the successive overrelaxation idea was originally used to improve Gauss-Seidel, an archetypal conditional maximization algorithm. The behavior of Gauss-Seidel also suggests that component-wise overrelaxation works well when components of the parameter vector are heavily “positively correlated”.

4 Examples

4.1 Probit regression revisited

Liu et al. (1998) use probit regression as part of the illustration of their PX-EM algorithm. We show how to apply monotonic overrelaxation to further improve PX-EM. In the setting of Section 2, let us introduce an auxiliary parameter $\alpha > 0$ and reformulate the complete-data model as

$$y_i = \text{sgn}(\xi_i), \quad \xi_i | (\theta, X) \stackrel{\text{ind}}{\sim} N(\alpha X_i \theta, \alpha^2),$$

where $\xi = (\xi_1, \dots, \xi_n)^\top$ are the latent variables. The observed data model remains the same. PX-EM is simply EM for the expanded parameter (θ, α) under this new complete-data model. At iteration t , suppose the current parameter is $\theta^{(t)}$ and, without loss of generality, $\alpha^{(t)} = 1$. The E-step computes the Q function as

$$Q(\theta, \alpha | \theta^{(t)}, \alpha^{(t)} = 1) = -n \log \alpha - \sum_{i=1}^n \frac{(\tilde{\eta}_i - \alpha X_i \theta)^2 + v_i}{2\alpha^2}, \quad (4.1)$$

where $\tilde{\eta}$ and v_i are the same as in (2.2). We need the values of v_i , which are unnecessary when implementing standard EM.

$$v_i \equiv \text{var}(\xi_i | \theta^{(t)}, \alpha^{(t)} = 1, X, y) = 1 - \left(y_i \mu_i + \frac{\phi(\mu_i)}{\Phi(y_i \mu_i)} \right) \frac{\phi(\mu_i)}{\Phi(y_i \mu_i)}, \quad \mu_i = X_i \theta^{(t)}.$$

The M-step maximizes $Q(\theta, \alpha | \theta^{(t)}, \alpha^{(t)} = 1)$ with respect to (θ, α) , yielding

$$\begin{aligned} \hat{\alpha}^2 &= n^{-1} \sum_{i=1}^n (r_i^2 + v_i), \quad r_i = \tilde{\eta}_i - X_i \theta_{\text{EM}}^{(t+1)}, \quad \theta_{\text{EM}}^{(t+1)} = (X^\top X)^{-1} X^\top \tilde{\eta}, \\ \theta_{\text{PXEM}}^{(t+1)} &= \theta_{\text{EM}}^{(t+1)} / \hat{\alpha}. \end{aligned} \quad (4.2)$$

Note that $\theta_{\text{EM}}^{(t+1)}$ is the standard EM update. Hence PX-EM only involves minor modifications of the standard EM iteration. The convergence rate of PX-EM, however, is provably better (Liu et al., 1998).

It seems difficult to design an overrelaxation step for (θ, α) jointly. However, we can divide the M-step into two conditional maximization steps and apply overrelaxation to each. It helps to view (4.1) as a function of $\beta \equiv \alpha\theta$ and $\gamma \equiv \alpha^{-2}$. For fixed γ , (4.1) is a quadratic function of β . Thus, setting

$$\beta^{(t+1)} = (1 + w)\theta_{\text{EM}}^{(t+1)} - w\theta^{(t)}, \quad w \in [0, 1],$$

increases the Q function. When $\alpha\theta$ is fixed at $\beta^{(t+1)}$, the function (4.1) in terms of $\gamma = \alpha^{-2}$ becomes

$$\tilde{Q}(\gamma) = \frac{n}{2} \left(\log \gamma - \frac{\gamma}{\hat{\gamma}} \right), \quad (4.3)$$

where

$$\hat{\gamma} = \frac{n}{\sum_{i=1}^n (\tilde{r}_i^2 + v_i)}, \quad \tilde{r}_i = \tilde{\eta}_i - X_i \beta^{(t+1)}. \quad (4.4)$$

To design an overrelaxation step for γ , we exploit simple properties of (4.3). The function (4.3) has a unique mode at $\hat{\gamma}$. It is skewed to the right as a function of γ , but skewed to the left as a

Table 2: Iteration count and computer time (in seconds) for monotonically overrelaxed PX-EM for fitting the lupus nephritis data. PX-EM corresponds to $w = 0$.

w	0.0	0.25	0.5	0.75	1.0
iteration count	499	403	338	292	257
computer time	0.16	0.16	0.13	0.11	0.10

function of $\log \gamma$. Hence we perform overrelaxation on the linear scale when $\hat{\gamma} > \gamma^{(t)} = 1$, and on the log scale otherwise. Specifically, let us define ($w \geq 0$)

$$s(a, b, w) = \begin{cases} (1+w)b - wa, & b \geq a \geq 0, \\ b^{1+w}a^{-w}, & a > b \geq 0. \end{cases} \quad (4.5)$$

Then $s(a, b, w)$ and a lie on opposite sides of b . One can easily show

$$\tilde{Q}(s(1, \hat{\gamma}, w)) \geq \tilde{Q}(1), \quad w \in [0, 1].$$

That is, for any $w \in [0, 1]$ and fixed $\beta^{(t+1)}$, setting γ at $\gamma^{(t+1)} = s(1, \hat{\gamma}, w)$ increases the Q function. Since $\beta = \alpha\theta = \gamma^{-1/2}\theta$, the implied value of θ is $\theta^{(t+1)} = \beta^{(t+1)} (\gamma^{(t+1)})^{1/2}$. Overall, we obtain a monotonically overrelaxed PX-EM (MOPX) algorithm:

$$\theta_{\text{MOPX}}^{(t+1)} = \left((1+w)\theta_{\text{EM}}^{(t+1)} - w\theta^{(t)} \right) s^{1/2}(1, \hat{\gamma}, w), \quad (4.6)$$

where $\hat{\gamma}$ and $s(\dots)$ are given by (4.4) and (4.5), respectively.

The MOPX iteration (4.6) is only slightly more complicated than (4.2). We evaluate its effect on PX-EM under the same setting as in Section 2. Table 2 records the performance of MOPX for different choices of w . Compared with Table 1, Table 2 shows that PX-EM reduces the number of iterations and computer time of standard EM by large factors. It is therefore especially worth noting that MOPX leads to further reductions. Figure 2 shows that all five algorithms maintain monotonicity in the observed log-likelihood; it also reveals the steady improvement by using larger w .

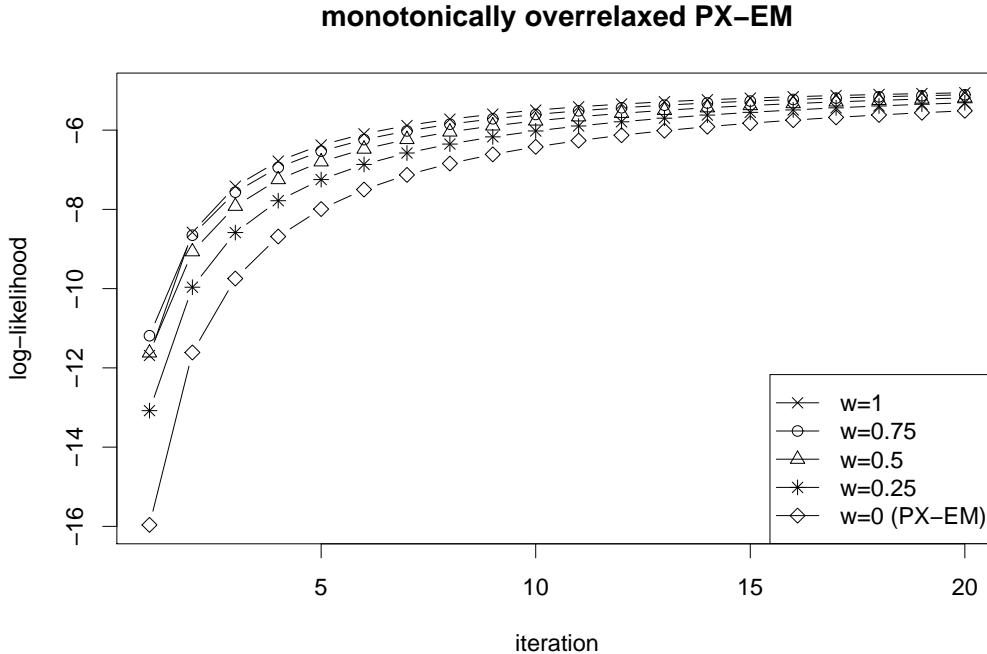


Figure 2: Progression of the log-likelihood for MOPX for fitting the lupus nephritis data.

4.2 Least absolute deviations regression

Robust regression problems, such as t -regression or least absolute deviations (LAD) regression, can often be handled by MM algorithms with a $Q(\theta|\tilde{\theta})$ function that is quadratic in θ . As in the case of EM for probit regression in Section 2, monotonic overrelaxation is easy to apply. We illustrate with LAD regression.

Consider maximizing the function

$$l(\theta) = - \sum_{i=1}^n |y_i - X_i\theta|$$

where $y = (y_1, \dots, y_n)^\top$ is the observed response, $X = (X_1^\top, \dots, X_n^\top)^\top$ ($n \times p$) is a full-rank matrix of covariates, and θ ($p \times 1$) is the parameter of interest. Schlossmacher (1973) proposes an iteratively reweighted least squares algorithm for solving this problem. As noted by Lange et al. (2000), Schlossmacher's algorithm can be derived as an MM algorithm. We choose

$$Q(\theta|\tilde{\theta}) = \frac{l(\tilde{\theta})}{2} - \frac{1}{2} \sum_{i=1}^n \frac{(y_i - X_i\theta)^2}{|y_i - X_i\tilde{\theta}|},$$

which satisfies $l(\theta) \geq Q(\theta|\tilde{\theta})$ for all θ and $\tilde{\theta}$ by the concavity of the function \sqrt{x} , $x \geq 0$. At iteration t , maximizing $Q(\theta|\theta^{(t)})$ with respect to θ yields

$$\theta_{\text{MM}}^{(t+1)} = (X^\top DX)^{-1} X^\top Dy, \quad D = \text{Diag}(1/|r_1|, \dots, 1/|r_n|), \quad r_i = y_i - X_i\theta^{(t)}. \quad (4.7)$$

This algorithm can also be interpreted as an EM algorithm (Lange et al., 2000).

Monotonic overrelaxation can be applied to Schlossmacher's algorithm. We simply set the next iteration as

$$\theta_{\text{MOMM}}^{(t+1)} = (1 + w)\theta_{\text{MM}}^{(t+1)} - w\theta^{(t)}.$$

Since $Q(\theta|\tilde{\theta})$ is a quadratic function of θ , any $w \in [0, 1]$ ensures monotonic increase in $l(\theta)$.

Schlossmacher's algorithm may run into numerical problems when some of the residuals r_i in (4.7) are close to zero. One remedy is to replace the weights $1/|r_i|$ by $(r_i^2 + \epsilon)^{-1/2}$, where $\epsilon > 0$. This corresponds to maximizing a slightly different objective function

$$\tilde{l}(\theta) = - \sum_{i=1}^n ((y_i - X_i\theta)^2 + \epsilon)^{1/2}.$$

We have observed that using a small but positive ϵ ($\epsilon = 10^{-8}$ is used in the simulation experiment below) can make both Schlossmacher's algorithm and the overrelaxed versions much more stable.

The simulation setting is as follows. We choose $n = 100$ and $p = 3$. The first column of X is a vector of ones; all other entries are independent standard normal variates. The response y is generated according to $y_i \sim X_i\theta + t_3$, i.e., a t -distribution with 3 degrees of freedom, and the true parameter value is $\theta = (1, 1, 1)^\top$. We use the same starting value and convergence criterion as in Section 2. The experiment is replicated 200 times.

This experiment serves to illustrate both the potential gain and some of the practical issues of using overrelaxation. In our simulations, Schlossmacher's algorithm often converges quickly, e.g., in less than 50 iterations. We have noticed that setting the overrelaxation parameter too high in these easy cases can slow down the algorithm considerably. As discussed in Section 3, one may estimate the convergence rate and then set the overrelaxation parameter. Here we adopt the simpler strategy of applying overrelaxation only after iteration 50. Table 3 records some summary statistics of the iteration counts of MOMM. Only the replications where Schlossmacher's algorithm takes more than 50 iterations are included in the comparisons (there are 155 such cases, including one where Schlossmacher's algorithm takes more than 10000 iterations). The

Table 3: Summary statistics of the iteration counts for monotonically overrelaxed MM for fitting LAD regression on simulated data. Schlossmacher’s algorithm corresponds to $w = 0$.

w	0.0	0.5	0.9	1.0
1st Quartile	80	69	68	69
Median	117	94	86	92
Mean	305+	236	200	198
3rd Quartile	207	157	135	138
Maximum	10000+	8910	7155	6823

improvements of using $w > 0$ are appreciable (albeit moderate). Figure 3 displays the acceleration ratios, calculated by dividing the iteration count of Schlossmacher’s algorithm by that of MOMM. The 155 replications are divided into fast MM (left panel) or slow MM (right panel) according to whether Schlossmacher’s algorithm takes fewer or more than the median of its iteration counts. Overrelaxation leads to more appreciable improvements when Schlossmacher’s algorithm is slower. Also, $w = 0.5$ and $w = 0.9$ appear more stable than $w = 1$, which occasionally slows down Schlossmacher’s algorithm by a significant factor. This again highlights the potential danger of selecting too large an overrelaxation parameter. Overall, $w = 0.5$ seems too conservative, and $w = 1$ too aggressive; $w = 0.9$ is a reasonable trade-off in this example.

4.3 Poisson inverse problems

EM is a valuable tool for image reconstruction using positron emission tomography (Vardi et al., 1985; Fessler and Hero, 1994). We review a simplified version of the SAGE algorithm of Fessler and Hero (1994), and show how monotonic overrelaxation can be applied.

Consider ML estimation in the Poisson model

$$y_i \stackrel{\text{ind}}{\sim} \text{Poi}(x_i \lambda + r_i), \quad i = 1, \dots, n, \quad (4.8)$$

where $\lambda \geq 0$ is the unknown parameter, y_1, \dots, y_n are observed counts, and $x_i, r_i, i = 1, \dots, n$, are nonnegative constants. To design a fast EM algorithm, let us define

$$r_0 = \min_{1 \leq i \leq n} \left\{ \frac{r_i}{x_i} \right\}, \quad \tilde{r}_i = r_i - x_i r_0,$$

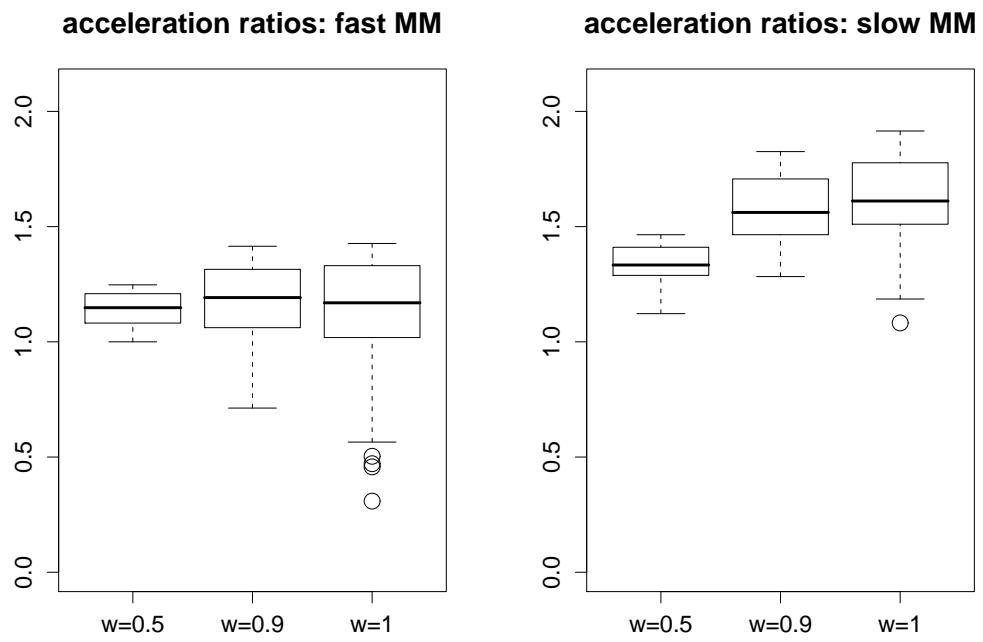


Figure 3: Acceleration ratios relative to Schlossmacher's algorithm for fitting LAD regression on simulated data.

and write the model as

$$y_i \stackrel{\text{ind}}{\sim} \text{Poi}(x_i(\lambda + r_0) + \tilde{r}_i), \quad i = 1, \dots, n. \quad (4.9)$$

Introduce mutually independent $z_{i1}, z_{i2}, i = 1, \dots, n$, such that

$$z_{i1} \sim \text{Poi}(x_i(\lambda + r_0)), \quad z_{i2} \sim \text{Poi}(\tilde{r}_i).$$

Suppose only $y_i = z_{i1} + z_{i2}, i = 1, \dots, n$, are observed. Then the observed data model is simply (4.9). Under this formulation, EM calculates the Q function, up to a constant, as

$$Q(\lambda | \lambda^{(t)}) = - \sum_{i=1}^n x_i(\lambda + r_0) + \sum_{i=1}^n \tilde{z}_{i1} \log(\lambda + r_0), \quad (4.10)$$

where

$$\tilde{z}_{i1} = E[z_{i1} | \lambda^{(t)}, y_i, x_i] = \frac{y_i x_i (\lambda^{(t)} + r_0)}{x_i (\lambda^{(t)} + r_0) + \tilde{r}_i}.$$

Maximizing (4.10) with respect to $\lambda \geq 0$ yields the EM iteration

$$\lambda_{\text{EM}}^{(t+1)} = \max \left\{ 0, \frac{\sum_{i=1}^n \tilde{z}_{i1}}{\sum_{i=1}^n x_i} - r_0 \right\}. \quad (4.11)$$

Monotonic overrelaxation is facilitated by the simple form of (4.10). The function (4.10) is a slight modification of (4.3) and can be handled in the same way. Define

$$\lambda_{\text{MOEM}}^{(t+1)} = \max \left\{ 0, s \left(\lambda^{(t)} + r_0, \lambda_{\text{EM}}^{(t+1)} + r_0, w \right) - r_0 \right\}, \quad w \in [0, 1], \quad (4.12)$$

where $s(\dots)$ is given by (4.5). Note that $\lambda_{\text{MOEM}}^{(t+1)}$ and $\lambda^{(t)}$ lie on opposite sides of $\lambda_{\text{EM}}^{(t+1)}$. We can show

$$Q \left(\lambda_{\text{MOEM}}^{(t+1)} | \lambda^{(t)} \right) \geq Q \left(\lambda^{(t)} | \lambda^{(t)} \right)$$

as we do for the parameter γ in the PX-EM example of Section 4.1. Thus, (4.12) satisfies the requirements of monotonic overrelaxation.

Iteration (4.11) serves as a building block for the space-alternating-generalized EM (SAGE) algorithm for ML fitting of the model

$$y_i \stackrel{\text{ind}}{\sim} \text{Poi}(X_i \theta), \quad i = 1, \dots, n, \quad (4.13)$$

where $X = (X_1^\top, \dots, X_n^\top)^\top \equiv (X_{ij})$ is a known $n \times p$ matrix, and $\theta = (\theta_1, \dots, \theta_p)^\top$ is the parameter of interest. In practice, penalized likelihood fitting has better statistical properties and is preferred.

We describe the algorithms for pure ML fitting for simplicity, although both the SAGE algorithm and our proposed monotonic overrelaxation scheme are easily extended.

Fessler and Hero (1994) design the SAGE algorithm as a conditional maximization scheme which updates one coordinate of θ at a time, given the others. Each conditional step seeks to maximize the log-likelihood with respect to a particular coordinate of θ . The iteration (4.11) can be used for such conditional steps because the model in terms of each coordinate of θ is in the form of (4.8). Monotonic overrelaxation is readily applied to the SAGE algorithm. Instead of using (4.11) at each conditional maximization step, we use (4.12). We refer to the resulting algorithm as monotonically overrelaxed SAGE (MO-SAGE).

A small simulation is conducted to illustrate MO-SAGE. We set $n = p = 100$, $X_{ij} = \phi(i - j)$ (the normal density), and generate y according to (4.13) with $\theta = (5, \dots, 5)^\top$. Based on 50 replications, Table 4 records the performance of MO-SAGE with different overrelaxation parameters, starting at the same $\theta^{(0)} = (1, \dots, 1)^\top$. We observe that the effect of overrelaxation becomes more pronounced for larger w . It seems safe to set $w = 1$, which yields substantial reductions in the mean iteration counts. Figure 4 displays the acceleration ratios. The 50 replications are divided into two groups by the median iteration count of SAGE. For each w , the effect of overrelaxation does not seem to depend much on whether SAGE is fast or slow. Note that the improvement can go far beyond a factor of two. This is possible because MO-SAGE is a component-wise strategy, whose convergence rate is more complicated than that of “global overrelaxation,” i.e., applying overrelaxation to the whole parameter vector jointly. It also illustrates the potential benefits of component-wise overrelaxation.

It would be worthwhile to develop MO-SAGE strategies for penalized ML fitting of large scale real data. We plan to report such findings in future works.

4.4 Finite mixtures

This section focuses on ML estimation of mixture proportions with known component densities. Several other statistical problems, including nonparametric estimation of the distribution function for censored data, lead to log-likelihood functions of the same form (Turnbull, 1976; Groeneboom and Wellner, 1992; Böhning et al., 1996). The EM algorithm is easy to implement, but can be extremely slow, and much effort is devoted to finding faster alternatives; see, e.g., Wellner and

Table 4: Summary statistics of the iteration counts for monotonically overrelaxed SAGE on simulated data. SAGE corresponds to $w = 0$.

w	0.0	0.25	0.5	0.75	1.0
1st Quartile	977	676	488	354	239
Median	1458	1020	700	514	359
Mean	1632	1150	804	588	392
3rd Quartile	2034	1495	1078	779	502
Maximum	4159	2798	1716	1501	1057

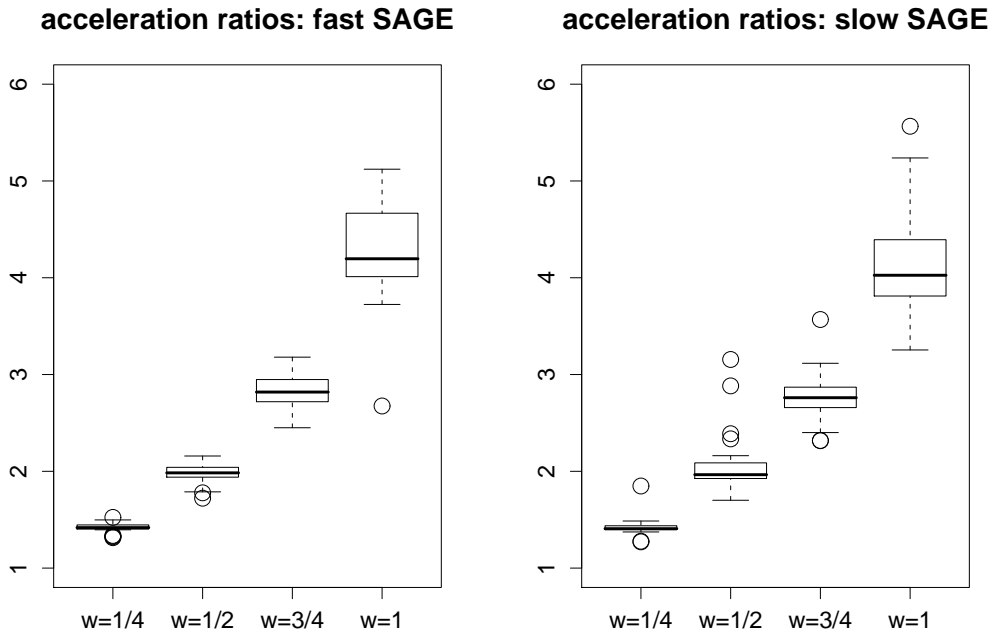


Figure 4: Acceleration ratios relative to SAGE for the Poisson inverse problem on simulated data.

Zhan (1997), Pilla and Lindsay (2001), and Yu (2010c). Here we derive a simple monotonic overrelaxation scheme which can conceivably generalize to problems such as mixture models with parametrized component densities.

Suppose samples $y = (y_1, \dots, y_n)$ are observed from a mixture of m known densities with unknown proportions $\theta_1, \dots, \theta_m$. Let f_{ij} be the j th component density evaluated at y_i . Then the log-likelihood for $\theta = (\theta_1, \dots, \theta_m)$ is

$$l(\theta) = \sum_{i=1}^n \log \left(\sum_{j=1}^m f_{ij} \theta_j \right). \quad (4.14)$$

Let I_{ij} be the latent component indicators, i.e., $I_{ij} = 1$ if the i th observation is from component j , and $I_{ij} = 0$ otherwise. Given the current $\theta^{(t)}$, standard EM computes the Q function as (up to a constant)

$$Q(\theta | \theta^{(t)}) = n \sum_{j=1}^m K_j \log \theta_j, \quad (4.15)$$

where

$$K_j = \frac{1}{n} \sum_{i=1}^n E(I_{ij} | y, \theta^{(t)}) = \frac{1}{n} \sum_{i=1}^n \frac{f_{ij} \theta_j^{(t)}}{\sum_{k=1}^m f_{ik} \theta_k^{(t)}}.$$

The M-step maximizes $Q(\theta | \theta^{(t)})$ with respect to θ , yielding $\theta_{j,\text{EM}}^{(t+1)} = K_j$. Overall each iteration is simply

$$\theta_{j,\text{EM}}^{(t+1)} = \frac{1}{n} \sum_{i=1}^n \left(\frac{f_{ij}}{\sum_{k=1}^m f_{ik} \theta_k^{(t)}} \right) \theta_j^{(t)}, \quad j = 1, \dots, m.$$

We can derive a family of overrelaxation schemes by modifying the above M-step as

$$\theta_j^{(t+1)} = (1 + w^{(t)}) K_j - w^{(t)} \theta_j^{(t)}, \quad (4.16)$$

where

$$w^{(t)} = \frac{w(\min_j r_j)}{1 + w - w(\min_j r_j)}, \quad r_j \equiv \frac{K_j}{\theta_j^{(t)}}, \quad w \in [0, 1]. \quad (4.17)$$

Obviously, this iteration satisfies (3.3). To show that it also satisfies (3.2), note that

$$\begin{aligned} Q(\theta^{(t+1)} | \theta^{(t)}) - Q(\theta^{(t)} | \theta^{(t)}) &= n \sum_{j=1}^m \theta_j^{(t)} r_j \log((1 + w^{(t)}) r_j - w^{(t)}) \\ &\geq n \bar{r} \log((1 + w^{(t)}) \bar{r} - w^{(t)}) \\ &= 0, \end{aligned}$$

where $\bar{r} = \sum_{j=1}^m \theta_j^{(t)} r_j = 1$ (hence the final equality), and the inequality follows by applying Jensen's inequality to the function $g(x) = x \log((1 + w^{(t)})x - w^{(t)})$, which is convex on $x \in [2w^{(t)}/(1 + w^{(t)}), \infty)$. The $w^{(t)}$ specified by (4.17) ensures that all r_j are within this convexity range. Thus (4.16) gives a family of MOEM updates. A similar technique is employed by Dette et al. (2008) to construct algorithms for finding D-optimal designs (approximate theory); see Yu (2010b) for a subsequent analysis of the convergence rates.

The iteration (4.16) is of the same form as the squeezed EM in Section 2.2 of Yu (2010c). The difference is that (4.16) is based on overrelaxation, whereas the squeezed EM of Yu (2010c) is based on efficient data augmentation. The two strategies lead to different allowable ranges for $w^{(t)}$. In (4.17), we have $\min_j r_j \leq 1$ because $\sum_j r_j \theta_j^{(t)} = 1$, and hence $w^{(t)} \leq w$. We impose $w \in [0, 1]$ to ensure monotonicity. In contrast, the $w^{(t)}$ implied by the squeezing strategy of Yu (2010c) can exceed one, if the component densities overlap heavily. This squeezing strategy is partly inspired by Fessler and Hero (1994) and is applicable to channel capacity calculations in Shannon theory (see Yu, 2010a).

If (4.16) converges to some θ^* with all positive coordinates, then we can show $\lim_{t \rightarrow \infty} w^{(t)} = w$. In general we have $\lim_{t \rightarrow \infty} w^{(t)} \leq w$. That is, there may be a discrepancy between the nominal amount of overrelaxation represented by w and the actual amount represented by $\lim_{t \rightarrow \infty} w^{(t)}$. We may not be able to obtain a large $\lim_{t \rightarrow \infty} w^{(t)}$ because of this discrepancy (see Section 4.5 for an example).

There is no conceptual problem extending the MOEM algorithm to ML fitting of mixture models with unknown parameters in the component densities. Suppose the component densities are parameterized as $f_j(\cdot; \xi)$, $j = 1, \dots, m$. Then, defining I_{ij} as the component indicators as before, we have the complete-data log-likelihood

$$\sum_{i,j} I_{ij} \log \theta_j + \sum_{i,j} I_{ij} \log f_j(y_i; \xi).$$

The Q function, modified from (4.15), becomes

$$Q(\theta, \xi | \theta^{(t)}, \xi^{(t)}) = \sum_{i,j} \tilde{I}_{ij} \log \theta_j + \sum_{i,j} \tilde{I}_{ij} \log f_j(y_i; \xi),$$

where

$$\tilde{I}_{ij} = \frac{\theta_j^{(t)} f_j(y_i; \xi^{(t)})}{\sum_k \theta_k^{(t)} f_k(y_i; \xi^{(t)})}.$$

Often the parameters θ and ξ are distinct. Then the M-step reduces to two separate maximizations, one for each of θ and ξ . Monotonic overrelaxation can be incorporated by modifying the maximization step for θ using the scheme developed in this section. Depending on the model structure, additional overrelaxation can be incorporated for ξ . For example, ξ could represent the component means in a Gaussian mixture problem. Then straightforward overrelaxation as in Section 2 or Section 4.2 can be applied to ξ .

4.5 Bivariate interval censoring

This section illustrates the monotonic overrelaxation scheme of Section 4.4 with a bivariate censoring problem. Let (Y, Z) be a pair of random variables whose joint distribution function is F . Suppose there is a censoring mechanism, independent of (Y, Z) , so that we only observe a two-dimensional rectangular region R that is known to contain (Y, Z) . Our goal is to find the nonparametric MLE \hat{F} given n independent and identically distributed observation rectangles R_i . It can be shown that \hat{F} only assigns probability to certain rectangular regions known as *maximal intersections* (see Betensky and Finkelstein, 1999, and Maathuis, 2005). The log-likelihood, written in terms of the probabilities $\theta_1, \dots, \theta_m$ assigned to these maximal intersections, is of the form of (4.14), the entry f_{ij} being the indicator of whether the i th observation rectangle includes maximal intersection j .

We use data from the AIDS Clinical Trials Group protocol ACTG 181 as analyzed by Betensky and Finkelstein (1999). This was a natural history substudy of a comparative trial of three anti-pneumocystis drugs. Patients were followed for two clinical events, shedding of cytomegalovirus (CMV) in the urine and blood, and colonization of mycobacterium avium complex (MAC) in the sputum and stool. Some patients missed several prescheduled clinical visits, creating interval censored observations for either or both of these events. The data set analyzed by Betensky and Finkelstein (1999) is relatively small, corresponding to $n = 204$ and $m = 32$ in our notation (i.e., 32 maximal intersections).

Starting from the uniform distribution $\theta_j^{(0)} = 1/m$, $j = 1, \dots, m$, standard EM converges in 709 iterations, whereas (4.16) with $w = 1$ in (4.17) converges in 584 iterations. The algorithms deliver the same nonparametric MLE as reported by Betensky and Finkelstein (1999). The improvement in speed by using MOEM is small, partly because we cannot obtain large values of the actual

overrelaxation parameter $w^{(t)}$ in (4.16). As mentioned earlier, the limit of $w^{(t)}$ would have been w if all coordinates of the MLE $\hat{\theta}$ were positive. In this example, however, the MLE assigns zero weight to several maximal intersections. A calculation reveals that $\lim_{t \rightarrow \infty} w^{(t)} = 0.240$, which is far below $w = 1$.

The bivariate censoring problem can be a computational challenge, especially when the dimension is high. Specialized hybrid algorithms as developed by Wellner and Zhan (1997) or Yu (2010c) may hold more promise than the overrelaxation scheme of Section 4.4, whose effect is more moderate. Nevertheless, the overrelaxation strategy is valuable as it can conceivably extend to more general mixture problems, as discussed near the end of Section 4.4.

5 Discussion

Overrelaxation is an old idea that can be used to accelerate a variety of fixed point algorithms. In this paper we apply overrelaxation to EM-type algorithms commonly used in statistical computing, with special attention to maintaining the monotonicity of the objective function. We have shown through several examples that maintaining monotonicity often amounts to a small modification in the M-step. In hindsight, this is not surprising, because one key idea of EM (or more generally, MM) is to maximize a surrogate function Q that is easier to handle than the objective function $l(\theta)$ itself. Often Q is quadratic in at least part of the parameters. When $\hat{\theta}$, the maximizer of Q , can be found easily, one could just as easily design an overrelaxation step, moving the current θ to a higher point on the opposite side of $\hat{\theta}$. The resulting monotonically overrelaxed algorithms inherit the simplicity and stability of EM, often with appreciably faster rates of convergence.

One practical issue is the choice of the overrelaxation parameter w , besides the issue of whether the actual amount of overrelaxation can be pre-set while maintaining monotonicity. Examples in Section 4 show that the best choice often depends on the underlying problem, as does the effectiveness of overrelaxation in general. There exist problems where EM is fast, and setting $w \approx 1$ slows it down. There also exist problems where EM is slow, and setting $w = 1$ or even $w > 1$ leads to considerable improvements, although monotonicity of the log-likelihood cannot be guaranteed if w is too large. The former scenario shows the danger of too much overrelaxation. To improve stability, one may consider applying overrelaxation only after a certain number of iterations of the basic algorithm. The latter scenario indicates that our monotonically overrelaxed

algorithms may be too conservative sometimes. If, in a certain problem, one can routinely set $w \gg 1$ without frequent overshooting, then it pays to simply monitor the objective function $l(\theta)$, and step back when $l(\theta)$ decreases, rather than to guarantee monotonicity by designing model-specific overrelaxation steps.

In Section 4 we have designed overrelaxation strategies for various parameters such as regression coefficients, Poisson intensities, and mixture proportions. Although the resulting strategies are quite simple, their derivations sometimes involve nontrivial inequalities; see Section 4.4. Designing such strategies resembles constructing efficient data augmentation schemes (Meng and van Dyk 1999) or finding hidden expansion parameters. All require close inspection of the model structure. Being model-dependent means there is much room for exploration. It is not obvious, for example, how to design a monotonic overrelaxation strategy for a variance-covariance matrix, which would help speeding up EM-type algorithms for mixed effects models (Liu and Rubin 1994; Meng and van Dyk 1998).

In some sense overrelaxation is analogous to the conditional augmentation of Meng and van Dyk (1999), with the overrelaxation parameter w playing the role of the working parameter. Therefore it would be interesting to see if there exists an analogue of marginal augmentation, which integrates out the working parameter. Note, however, that in overrelaxation different w 's correspond to the same missing data formulation. Only the M-step is changed.

At any rate, through the examples in Section 4 we hope to convey that (i) overrelaxation is applicable to many optimization problems in statistical computing; (ii) it is possible to apply overrelaxation with automatically guaranteed monotonicity. Naturally, it would be worthwhile to apply these ideas to more statistical problems and to evaluate the resulting algorithms in realistic settings.

Supplemental materials

R code for examples of monotonically overrelaxed EM algorithms in Section 4. (overrelax-code.tar; GNU tar file)

Acknowledgments

The author would like to thank Don Rubin, Xiao-Li Meng, and David van Dyk for introducing him to the field of statistical computing. He is also grateful to the Editor, the Associate Editor, and two referees for their valuable comments.

References

- [1] Betensky, R. A. and Finkelstein, D. M. (1999). A non-parametric maximum likelihood estimator for bivariate interval censored data. *Statistics in Medicine* **18**, 3089-3100.
- [2] Bishop, Y. M. M., Fienberg, S. E. and Holland, P. W. (1975). *Discrete Multivariate Analysis: Theory and Practice*. MIT Press.
- [3] Böhning, D., Schlattmann, P. and Dietz E. (1996). Interval censored data: a note on the nonparametric maximum likelihood estimator of the distribution function. *Biometrika* **83**, 462–466.
- [4] Deming, W. E. and Stephan, F. F. (1940). On a least squares adjustment of a sampled frequency table when the expected marginal totals are known, *Annals of Mathematical Statistics* **11**, 427-444.
- [5] Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood estimation from incomplete data via the EM algorithm (with discussion). *J. Roy. Statist. Soc. B* **39**, 1–38.
- [6] Dette, H., Pepelyshev, A. and Zhigljavsky, A. (2008). Improving updating rules in multiplicative algorithms for computing D-optimal designs, *Computational Statistics & Data Analysis* **53**, 312–320.
- [7] Fessler, J.A. and Hero, A.O. (1994). Space-alternating generalized expectation-maximisation algorithm, *IEEE Trans. Signal Processing* **42**, 2664–2677.
- [8] Groeneboom, P. and Wellner, J. (1992). *Information Bounds and Nonparametric Maximum Likelihood Estimation*. Basel: Birkhäuser.

- [9] Jamshidian, M. and Jennrich, R.I. (1993). Conjugate gradient acceleration of the EM algorithm. *J. Am. Statist. Assoc.* **88**, 221–228.
- [10] Jamshidian, M. and Jennrich, R.I. (1997). Acceleration of the EM algorithm by using quasi-Newton methods, *J. Roy. Statist. Soc. Series B* **59**, 569–587.
- [11] Kuroda, M. and Sakakihara, M. (2006). Accelerating the convergence of the EM algorithm using the vector epsilon algorithm, *Computational Statistics and Data Analysis* **51**, 1549–1561.
- [12] Laird, N., Lange, N. and Stram, D. (1987). Maximizing likelihood computations with repeated measures: application of the EM algorithm. *J. Am. Statist. Assoc.* **82**, 97–105.
- [13] Lange, K. (1995a). A gradient algorithm locally equivalent to the EM algorithm. *J. Roy. Statist. Soc. Series B* **57**, 425–438.
- [14] Lange, K. (1995b). A quasi-Newtonian acceleration of the EM algorithm. *Statist. Sinica* **5**, 1–18.
- [15] Lange, K., Hunter, D.R. and Yang, I. (2000). Optimization transfer using surrogate objective functions (with discussion), *Journal of Computational and Graphical Statistics* **9**, 1–59.
- [16] Liu, C. H. and Rubin, D. B. (1994). The ECME algorithm: a simple extension of EM and ECM with faster monotone convergence. *Biometrika* **81**, 633–648.
- [17] Liu, C. H., Rubin, D. B. and Wu, Y. N. (1998). Parameter expansion to accelerate EM—the PX-EM algorithm. *Biometrika* **85**, 755–770.
- [18] Louis, T.A. (1982). Finding the observed information matrix when using the EM algorithm. *J. Roy. Statist. Soc. Series B* **44**, 226–233.
- [19] Maathuis, M. (2005). Reduction algorithm for the NPMLE for the distribution function of bivariate interval-censored data. *J. Computational and Graphical Statistics* **14**, 352–362.
- [20] McLachlan, G.J. and Krishnan, T. (1997). *The EM Algorithm and Extensions*, Wiley and Sons.
- [21] Meng, X.-L. and Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: a general framework. *Biometrika* **80**, 267–278.

- [22] Meng, X.-L. and van Dyk, D.A. (1997). The EM algorithm – an old folk-song sung to a fast new tune (with discussion), *J. Roy. Statist. Soc. B* **59**, 511–567.
- [23] Meng, X.-L. and van Dyk, D.A. (1998). Fast EM-type implementations for mixed effects models, *J. Roy. Statist. Soc. B* **60**, 559–578.
- [24] Meng, X.-L. and van Dyk, D.A. (1999). Seeking efficient data augmentation schemes via conditional and marginal augmentation, *Biometrika* **86**, 301–320.
- [25] Pilla, R.S. and Lindsay, B.G. (2001). Alternative EM methods for nonparametric finite mixture models, *Biometrika* **88**, 535–550.
- [26] Salakhutdinov, R. and Roweis, S. (2003). Adaptive overrelaxed bound optimization methods, *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, Washington DC.
- [27] Schlossmacher, E.J. (1973). An iterative technique for absolute deviations curve fitting. *J. Am. Statist. Assoc.* **68**, 857-859.
- [28] Turnbull, B. W. (1976). The empirical distribution function with arbitrarily grouped, censored and truncated data. *J. Roy. Statist. Soc. B* **38**, 290–295.
- [29] Vaida, F. (2005). Parameter convergence for EM and MM algorithms, *Statistica Sinica* **15**, 831–840.
- [30] Van Dyk, D. A. and Meng, X.-L. (2001). The art of data augmentation (with discussion). *Journal of Computational and Graphical Statistics* **10**, 1–111.
- [31] Varadhan, R. and Roland, Ch. (2004). Squared extrapolation methods (SQUAREM): A new class of simple and efficient numerical schemes for accelerating the convergence of the EM algorithm. *Department of Biostatistics Working Paper*, Johns Hopkins University, **63**, 1-70.
- [32] Vardi, Y., Shepp, L.A. and Kaufman, L. (1985). A statistical model for positron emission tomography (with discussion), *J. Amer. Statist. Assoc.* **80**, 8–37.
- [33] Wellner, J. A. and Zhan, Y. (1997). A hybrid algorithm for computation of the nonparametric maximum likelihood estimator from censored data. *J. Amer. Statist. Assoc.* **92**, 945-959.

- [34] Wu, C. F. J. (1983). On the convergence properties of the EM algorithm. *Annals of Statistics* **11**, 95–103.
- [35] Young, D. (1971). *Iterative Solutions of Large Linear Systems*. New York: Academic Press.
- [36] Yu, Y. (2010a). Squeezing the Arimoto-Blahut algorithm for faster convergence, *IEEE Transactions on Information Theory* **56**, 3149–3157.
- [37] Yu, Y. (2010b). Strict monotonicity and convergence rate of Titterington’s algorithm for computing D-optimal designs. *Computational Statistics & Data Analysis* **54**, 1419–1425.
- [38] Yu, Y. (2010c). Improved EM for mixture proportions with applications to nonparametric ML estimation for censored data, *Preprint*, arXiv:1002.3640