J. R. Statist. Soc. B (2000)
62, *Part* 2, *pp.* 257–270

Standard errors for EM estimation

Mortaza Jamshidian

University of Central Florida, Orlando, USA

and Robert I. Jennrich

University of California at Los Angeles, USA

[Received May 1998. Final revision July 1999]

Summary. The EM algorithm is a popular method for computing maximum likelihood estimates. One of its drawbacks is that it does not produce standard errors as a by-product. We consider obtaining standard errors by numerical differentiation. Two approaches are considered. The first differentiates the Fisher score vector to yield the Hessian of the log-likelihood. The second differentiates the EM operator and uses an identity that relates its derivative to the Hessian of the log-likelihood. The well-known SEM algorithm uses the second approach. We consider three additional algorithms: one that uses the first approach and two that use the second. We evaluate the complexity and precision of these three and the SEM algorithm in seven examples. The first is a single-parameter example used to give insight. The others are three examples in each of two areas of EM application: Poisson mixture models and the estimation of covariance from incomplete data. The examples show that there are algorithms that are much simpler and more accurate than the SEM algorithm. Hopefully their simplicity will increase the availability of standard error estimates in EM applications. It is shown that, as previously conjectured, a symmetry diagnostic can accurately estimate errors arising from numerical differentiation. Some issues related to the speed of the EM algorithm and algorithms that differentiate the EM operator are identified.

Keywords: Asymptotic variance–covariance matrix; EM algorithm; Numerical differentiation; Observed information; Precision; SEM algorithm; Slow convergence

1. Introduction

The EM algorithm (Dempster *et al.*, 1977) is a method for computing maximum likelihood estimates. It tends to be numerically stable and is easy to implement in many applications. A drawback is that it does not produce standard errors as a by-product.

A review of methods for estimating standard errors when applying the EM algorithm appears in Baker (1992). Among the methods that he reviews, we shall focus on computing the observed information matrix. To be more specific, let $\mathbf{y}$ be an observed data vector obtained by sampling from one of a family of densities $f(\mathbf{y}|\boldsymbol{\theta})$. The maximum likelihood estimate $\hat{\boldsymbol{\theta}}$ of the parameter vector $\boldsymbol{\theta}$ is the value $\boldsymbol{\theta}$ that maximizes $l(\boldsymbol{\theta}) = \log\{f(\mathbf{y}|\boldsymbol{\theta})\}$. Let $S(\boldsymbol{\theta})$ and $H(\boldsymbol{\theta})$ denote the gradient and Hessian of $l(\boldsymbol{\theta})$. The vector $S(\boldsymbol{\theta})$ is called the Fisher score vector and $-H(\boldsymbol{\theta})$ is called the observed information matrix. Let $H = H(\hat{\boldsymbol{\theta}})$. The matrix $V = -H^{-1}$ estimates the covariance matrix of $\hat{\boldsymbol{\theta}}$. It is V, or equivalently H, that we wish to compute.

We consider two approaches; the first numerically differentiates the Fisher score vector $S(\boldsymbol{\theta})$ at $\hat{\boldsymbol{\theta}}$ to give H. We refer to this as the NDS approach. The second, introduced by Meng

Address for correspondence: Mortaza Jamshidian, Department of Statistics, University of Central Florida, Orlando, FL 32816-2370,USA.
E-mail: morij@ucf.edu

 1369–7412/00/62257

and Rubin (1991), numerically differentiates the EM operator $M(\boldsymbol{\theta})$ and uses an identity given by Dempster *et al.* (1977) that relates the Jacobian of $M(\boldsymbol{\theta})$ at $\hat{\boldsymbol{\theta}}$ to H. We refer to this as the NDM approach.

Meng and Rubin (1991) also introduced a specific NDM algorithm called the SEM (supplemented EM) algorithm. Although no method of standard error estimation in conjunction with the EM algorithm is extensively used, the SEM algorithm seems to be the best known (for applications and examples of implementations see McLachlan and Krishnan (1997), Belin and Rubin (1995), Kim and Taylor (1995), Little (1995), Rubin *et al.* (1995), McCulloch (1994), Pagano *et al.* (1994), Segal *et al.* (1994), Lindsey and Ryan (1993) and Tu *et al.* (1993)). Meng and Rubin (1991) discussed SEM estimates for three examples: a one-parameter multinomial example, a univariate contaminated normal example and a bivariate normal example. The numbers of parameters estimated in these examples were 1, 2 and 5. These examples represent simple applications of the EM and SEM algorithms, where the EM algorithm converges reasonably fast and the number of parameters is small. Important applications of the EM method can involve larger numbers of parameters, more complex problems and slow convergence.

Baker (1992) mentioned the possibility of numerical inaccuracies when using the SEM algorithm and a limitation in that

> 'it requires code for the complete-data-observed information matrix, which is not available for some complicated models'.

Segal *et al.* (1994) gave an implementation of the SEM algorithm in the context of maximum penalized likelihood estimation and reported their SEM estimates on the one-parameter multinomial example mentioned above. They expressed the desire for additional numerical studies of the SEM algorithm in high dimensional settings. Belin and Rubin (1995) gave an implementation of the SEM method in the context of calibrating false match rates in record linkage, and they applied it to a problem with eight parameters, using a stopping rule that was different from that proposed by Meng and Rubin (1991). They stated that for their eight-parameter example the SEM method

> 'yields acceptable results for practice in that off-diagonal elements of the resulting covariance matrix agree with one another to a few decimal places'.

They also explained that, because the SEM algorithm requires very accurate estimates of $\hat{\boldsymbol{\theta}}$, SEM estimates can be much more expensive to obtain than the EM estimates. Finally, in a book review, McCulloch (1998) pointed out that

> 'for many problems, the Meng and Rubin (1991) method of obtaining standard errors can be numerically unstable'.

Meng and Rubin (1991), section 3.3, called for further investigations of whether or not the SEM procedure is the best way to approximate the Jacobian of $M(\boldsymbol{\theta})$. For this, in this paper, we consider two alternative NDM algorithms called FDM and REM based on the forward difference and Richardson extrapolation methods of numerical differentiation. We also consider a Richardson extrapolation implementation of the NDS approach and call it RES.

In Section 2 we define the EM operator and identify some basic results including the relationship between the Jacobian of $M(\boldsymbol{\theta})$ at $\hat{\boldsymbol{\theta}}$ and H. We also describe the NDS and NDM methods. In Section 3 we present numerical differentiation methods. In particular the forward difference algorithm FDM and the Richardson extrapolation algorithms RES and REM are described in Section 3.1 and the SEM algorithm is described in Section 3.2. In Section 4 we look at problems with NDM algorithms when the EM algorithm converges too slowly. In

Section 5 we compare the RES, REM, FDM and SEM methods on a simple example, on three examples using Poisson mixture models and three examples involving the estimation of covariances from incomplete data. Finally, in Section 6 we discuss the algorithms considered and make some recommendations. Four technical results are given in Appendix A.

2. Some definitions and basic results

For EM estimation, suppose that the observed data vector $\mathbf{y}$ is a function of a vector $\mathbf{x}$ of complete data with density $g(\mathbf{x}|\boldsymbol{\theta})$. Let

$$Q(\boldsymbol{\theta}', \boldsymbol{\theta}) = E[\log\{g(\mathbf{x}|\boldsymbol{\theta}')\}|\mathbf{y}, \boldsymbol{\theta}],$$

and let $M(\boldsymbol{\theta})$ be the value of $\boldsymbol{\theta}'$ that maximizes $Q(\boldsymbol{\theta}', \boldsymbol{\theta})$ given $\boldsymbol{\theta}$. The function $M(\boldsymbol{\theta})$ is called the EM operator.

Let $\dot{Q}(\boldsymbol{\theta}', \boldsymbol{\theta})$ and $\ddot{Q}(\boldsymbol{\theta}', \boldsymbol{\theta})$ denote the gradient and Hessian of $Q(\boldsymbol{\theta}', \boldsymbol{\theta})$ viewed as a function of $\boldsymbol{\theta}'$ and let $\dot{M}(\boldsymbol{\theta})$ denote the Jacobian of $M(\boldsymbol{\theta})$. Dempster *et al.* (1977) have shown that

$$H = \ddot{Q}(I - \dot{M}), \tag{1}$$

where $\ddot{Q} = \ddot{Q}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}})$ and $\dot{M} = \dot{M}(\hat{\boldsymbol{\theta}})$. A simple derivation of equation (1) based on Fisher's (1925) result

$$S(\boldsymbol{\theta}) = \dot{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}) \tag{2}$$

is given in Appendix A.

For convenience, hereafter we shall denote the approximations to H, V and $\dot{M}$ by H^*, V^* and $\dot{M}^*$. The type of approximation will be clear from the context.

As mentioned, we consider two ways to approximate H. For the NDS approach we numerically differentiate $S(\boldsymbol{\theta})$ at $\hat{\boldsymbol{\theta}}$ to obtain H^*. For the NDM approach we numerically differentiate $M(\boldsymbol{\theta})$ at $\hat{\boldsymbol{\theta}}$ to obtain $\dot{M}^*$ and use $H^* = \ddot{Q}(I - \dot{M}^*)$ to obtain H^*. Note that the NDS approach requires the vector of partial derivatives $\dot{Q}(\boldsymbol{\theta}, \boldsymbol{\theta})$, and the NDM approach requires $M(\boldsymbol{\theta})$ and the matrix of second partial derivatives $\ddot{Q}$, where $\boldsymbol{\theta}$ is arbitrary. These are problem specific and usually the most difficult part of implementing NDS and NDM algorithms.

To define NDS and NDM algorithms, specific methods of numerical differentiation are required. These are discussed in the next section.

3. Numerical differentiation methods

3.1. The RES, FDM and REM algorithms

In the context of approximating the derivative of a scalar-valued function $f(x)$ of a scalar-valued variable x, the simplest method of numerical differentiation is the forward difference method. A significantly better method, when we can afford twice as many function evaluations, is the central difference method. The central difference method may be further improved by applying Richardson extrapolation when doubling the number of function evaluations is not of great concern. A first-order Richardson extrapolation of the central difference is

$$\frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h}, \tag{3}$$

where h is a small positive value. The truncation error for Richardson extrapolation is much smaller than that for forward difference, $\mathcal{O}(h^4)$ *versus* $\mathcal{O}(h)$ (see, for example, Conte and deBoor (1980), pages 333–339).

Generalizations of these methods can be used to approximate the Jacobian of a vector-valued function $\mathbf{f}(\mathbf{x})$ of a vector-valued variable $\mathbf{x}$. For example, in the forward difference method, the jth column of the Jacobian of $\mathbf{f}(\mathbf{x})$ at $\mathbf{x}$ may be approximated by

$$D_j(h) = \frac{\mathbf{f}(\mathbf{x} + h\mathbf{u}^{(j)}) - \mathbf{f}(\mathbf{x})}{h},$$

where $\mathbf{u}^{(j)}$ is the jth co-ordinate vector, a vector with its jth component equal to 1 and all others equal to 0 (see, for example, Dennis and Schnabel (1983), pages 94–97, and Press *et al.* (1992), program `fdjac`, page 381). Similarly $D_j(h)$ for the Richardson extrapolation method is defined by replacing f by $\mathbf{f}$ and h by $h\mathbf{u}^{(j)}$ in the numerator of expression (3). Let p be the number of components of $\boldsymbol{\theta}$, and let $\mathbf{h}$ be a p-vector of small positive values h_j. Then

$$D(\mathbf{h}) = (D_1(h_1), \ldots, D_p(h_p))$$

is the corresponding numerical approximation to the Jacobian of $\mathbf{f}$ at $\mathbf{x}$.

Let FDM be the NDM algorithm that uses forward differences with

$$h_j = \eta \max(|\boldsymbol{\theta}_j|, 1) \tag{4}$$

and relative increment $\eta = 10^{-7}$. Also let RES and REM be the NDS and NDM algorithms that use Richardson extrapolation with h_j defined in equation (4) and relative increment $\eta = 10^{-4}$. The choices of the relative increments η were motivated by some general theory (for example, see Press *et al.* (1992), pages 182–183, and Dennis and Schnabel (1983), pages 98–99 and 105) and experience with some preliminary examples. No effort, however, was made to optimize these choices for our specific applications. There are more sophisticated implementations of forward difference and Richardson extrapolation for which the choice of increment is adapted to the problem at hand (see, for example, Press *et al.* (1992), program `dfridr`, pages 182–183). In the interest of simplicity we made no attempt to utilize these because our simple choices gave accurate approximations.

3.2. The SEM algorithm

Starting with an initial value $\boldsymbol{\theta}^{(0)}$, the EM algorithm proceeds by generating a sequence of EM iterates $\boldsymbol{\theta}^{(n+1)} = M(\boldsymbol{\theta}^{(n)})$. Under fairly general conditions this converges to $\hat{\boldsymbol{\theta}}$ (Dempster *et al.*, 1977; Wu, 1983).

The SEM algorithm of Meng and Rubin (1991) is a forward difference NDM algorithm that computes $\dot{M}^*$ as follows: produce an EM sequence $\boldsymbol{\theta}^{(n)}$, and stop as soon as $\|\boldsymbol{\theta}^{(n+1)} - \boldsymbol{\theta}^{(n)}\| < \epsilon$, for some small ϵ. Set $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(n+1)}$. Using a possibly different EM sequence $\boldsymbol{\theta}^{(n)}$, let

$$r_{ij}^{(n)} = \frac{M_i\{\hat{\boldsymbol{\theta}} + (\boldsymbol{\theta}_j^{(n)} - \hat{\boldsymbol{\theta}}_j)\mathbf{u}^{(j)}\} - \hat{\boldsymbol{\theta}}_i}{\boldsymbol{\theta}_j^{(n)} - \hat{\boldsymbol{\theta}}_j},$$

where M_i is the ith component of M, and $\boldsymbol{\theta}_j$ is the jth component of $\boldsymbol{\theta}$. Stop each of the p^2 sequences $r_{ij}^{(n)}$ as soon as $|r_{ij}^{(n+1)} - r_{ij}^{(n)}| < \sqrt{\epsilon}$. Set $r_{ij}^* = r_{ij}^{(n+1)}$, and let $\dot{M}^*$ be the matrix of r_{ij}^*-values. The corresponding H^* is the SEM approximation to H. Meng and Rubin (1991) did not give specific convergence criteria, but those given here seem to be at least similar to what they had in mind. We shall use these criteria for our computations.

The SEM algorithm breaks down if some components of $\boldsymbol{\theta}^{(n)} - \hat{\boldsymbol{\theta}}$ are 0. Meng and Rubin (1991), section 3.4, explained how to avoid this problem, but the fix is not straightforward and is problem specific. This problem does not arise with the other algorithms that we consider.

As a starting value for the SEM iteration Meng and Rubin (1991) recommend the same starting value as is used for the EM iteration or, to save computation, $\boldsymbol{\theta}_i^{(0)} = \hat{\boldsymbol{\theta}}_i + 2\sqrt{-\ddot{Q}_{ii}^{-1}}$. This alternative was used for our examples in Sections 5.3 and 5.4. We consider three versions of the SEM algorithm denoted by SEM_4, SEM_8 and SEM_{12}. They correspond to using ϵ-values 10^{-4}, 10^{-8} and 10^{-12}. The same ϵ-values were used by Meng and Rubin (1991) in their examples.

4. Problems when the EM algorithm is slow

The speed of the EM algorithm can affect the performance of NDM algorithms. These include the FDM, REM and SEM algorithms. The largest eigenvalue $\lambda_{\max}$ of $\dot{M}$ determines the rate of convergence of the EM algorithm. As $\lambda_{\max}$ approaches 1 the EM algorithm slows. Let $\Delta V = V^* - V$ and $\Delta \dot{M} = \dot{M}^* - \dot{M}$ be the errors in V^* and $\dot{M}^*$. It is shown in Appendix A that

$$\Delta V \approx (I - \dot{M})^{-1} \Delta \dot{M} V \tag{5}$$

for small values of $\Delta \dot{M}$. As the EM algorithm slows, $(I - \dot{M})^{-1}$ becomes large. This multiplies the error in $\dot{M}^*$ by a large factor and substantially increases the error in V^*. We call this *error magnification* resulting from the use of a slow EM algorithm.

Meng and Rubin (1991) have stated that, when the EM algorithm is slow, the SEM algorithm is 'self-adjusting'. Although we are not sure what they had in mind, there is a self-adjustment phenomenon at work here. Consider the one-parameter problem and a plot of the EM operator $M(\boldsymbol{\theta})$ on $\boldsymbol{\theta}$. It and the 45° line through the origin pass through the point $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}})$. Moreover, because the EM algorithm converges to $\hat{\boldsymbol{\theta}}$, immediately to the left of $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}})$, $M(\boldsymbol{\theta})$ is above the 45° line and immediately to the right it is below. Fig. 1 shows several such plots that are based on our example in Section 5.2. Consider a sequence of EM operators whose slopes at $\hat{\boldsymbol{\theta}}$ approach 1. Assuming that it exists, the limit of such a sequence must have an inflection point at $\hat{\boldsymbol{\theta}}$ and hence, again assuming that it exists, a zero second derivative there. This means that, for a given increment h, the leading term in the truncation error for the forward difference method which is $\ddot{M}(\hat{\boldsymbol{\theta}})h/2$ disappears as we move to slower EM algorithms. We now have increasingly smaller truncation errors being magnified by increasingly larger

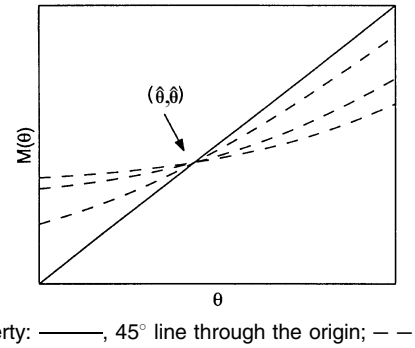

Fig. 1. Self-adjustment property: ———, 45° line through the origin; — — —, plots of three EM operators

factors. This is a self-adjusting phenomenon that can reduce the effect of slow convergence when using the forward difference method. This self-adjustment applies only to truncation errors and not to rounding errors, both of which will be magnified as the EM algorithm becomes slow. Note also that it only applies to forward difference methods.

The SEM algorithm has a problem in addition to error inflation. For a given ϵ the size of the increments h_j used increases as the speed of the EM algorithm decreases (see Appendix A.3) and this leads to larger truncation errors. As our examples will show, self-adjustment may not be sufficient to handle this additional problem and as a consequence the performance of the SEM algorithm often suffers when the corresponding EM algorithm is slow.

5. Comparison of algorithms

5.1. Complexity and precision

We shall compare the complexity and precision of the RES, FDM, REM and SEM algorithms for several examples. In each case the complexity has two aspects: a general aspect and a problem-specific aspect. Because it requires the construction and monitoring of p^2 sequences of approximate derivatives $r_{ij}^{(n)}$, the complexity of the general part of the SEM algorithm is greater than that of the other algorithms. The difference is even greater when one or several components of $\boldsymbol{\theta}^{(n)} - \hat{\boldsymbol{\theta}}$ become 0.

Turning to problem-specific complexity, the NDS algorithm requires formulae and code for $S(\boldsymbol{\theta}) = \dot{Q}(\boldsymbol{\theta}, \boldsymbol{\theta})$, and the NDM algorithms require these for $M(\boldsymbol{\theta})$ and $\ddot{Q}$. In each of our examples it was simpler to obtain the required formulae and code for the NDS algorithm than for the NDM algorithms.

With regard to the precision of the various methods, we compared the sizes of the errors ΔV. To obtain a natural index to measure these errors, note that the observed information matrix estimate of the variance of a linear combination $\mathbf{l}^{\mathrm{T}}\hat{\boldsymbol{\theta}}$ of the components of $\hat{\boldsymbol{\theta}}$ is $\mathbf{l}^{\mathrm{T}}V\mathbf{l}$. Let V^* be an approximation to V. We shall use the *maximum relative error* MRE in $\mathbf{l}^{\mathrm{T}}V^*\mathbf{l}$

$$\mathrm{MRE}(V^*) = \max_{\mathbf{l}} \left| \frac{\mathbf{l}^{\mathrm{T}}\Delta V\mathbf{l}}{\mathbf{l}^{\mathrm{T}}V\mathbf{l}} \right|$$

to measure the errors in V^*. Correspondingly, we shall use the *precision* PRE of V^*,

$$\mathrm{PRE}(V^*) = -\log_{10}\{\mathrm{MRE}(V^*)\}, \tag{6}$$

in our tables and graphs as a measure of performance of the algorithms. Roughly, PRE measures the number of leading digits in V^* that agree with those in V.

Let $C = \mathrm{sym}(V^*) = (V^* + V^{*\mathrm{T}})/2$ and $K = (V^* - V^{*\mathrm{T}})/2$ denote the symmetric and skew symmetric parts of V^*. Since V is symmetric, non-zero elements of K are pure error. Meng and Rubin (1991) suggested using the size of K to estimate the size of the errors in V^*. A basic question is whether this works in practice. We shall use our examples to evaluate the extent to which errors in the skew symmetric part of V^* can predict those in the symmetric part.

For this it is shown in Appendix A that

$$\mathrm{MRE}(V^*) = \|V^{-1/2}\,\mathrm{sym}(\Delta V)V^{-1/2}\| \tag{7}$$

where $\|A\|$ denotes the spectral norm of A. We shall estimate $\mathrm{MRE}(V^*)$ by

$$\widehat{\mathrm{MRE}}(V^*) = \|C^{-1/2}KC^{-1/2}\|$$

and $\mathrm{PRE}(V^*)$ by

$$\widehat{\mathrm{PRE}}(V^*) = -\log_{10}\{\widehat{\mathrm{MRE}}(V^*)\}. \tag{8}$$

For some of our examples the EM algorithm did not produce accurate values for $\hat{\boldsymbol{\theta}}$ even when ϵ was very small. Rather than compromise the use of equation (1) which holds only at $\hat{\boldsymbol{\theta}}$, we computed $\hat{\boldsymbol{\theta}}$ accurately by using Newton's method and used this value for all the methods compared.

As noted earlier we consider the SEM_4, SEM_8 and SEM_{12} algorithms in our examples. Since SEM_4 estimates were not accurate for many of these we do not report them in detail. Briefly, however, SEM_4 had zero precision (PRE) in two of the six examples of Sections 5.3 and 5.4 and its precision ranged from one to three digits in the remaining four examples.

5.2. A simple example

A simple artificial example may help to clarify ideas, especially those in Section 4. Let $y \sim N(e^{\theta}, 1)$ be observed data. Consider the EM algorithm with complete data $x = (y, u)$ where $u \sim N(e^{\theta}, \sigma^2)$ is unobserved and independent of y. The speed of this algorithm depends on the choice of the constant σ^2. This EM algorithm is sufficiently simple to consider in detail, but sufficiently rich to be interesting.

It is easy to show that for this example, as $\sigma^2 \to 0$, $\dot{M} \to 1$, so the EM algorithm converges arbitrarily slowly as $\sigma^2 \to 0$.

Let $y = 10$. Fig. 2(a) is a plot of V and its approximations based on the RES, REM, FDM and SEM_8 algorithms for values of σ^2 that vary from 0.001 to 0.2. These correspond to values of $\dot{M}$ that vary from 0.999 to 0.83. To the resolution of Fig. 2(a), the plots for the approximations RES, REM and FDM are indistinguishable from that for V. They are all represented by the upper curve. The lower curve is for SEM_8 which shows substantial error for small values of σ^2. In this example, as predicted in Section 4, self-adjustment does not protect the SEM algorithm when the convergence of the EM algorithm is slow.

The primary reason why the SEM_8 algorithm shows more error than the other methods is that the increments used by it are too large especially when the EM algorithm is slow. The relative increments used vary from about 0.565 when $\sigma^2 = 0.001$ to about 0.003 when $\sigma^2 = 0.2$. This suggests using a smaller ϵ so we shall switch to SEM_{12}.

Fig. 2(b) is a precision plot for the approximations in Fig. 2(a) with SEM_8 replaced by SEM_{12}. From the top down these plots are for the RES, REM, FDM and SEM_{12} algorithms. The precision in the RES plot is consistently about 12.5 significant digits and that in the

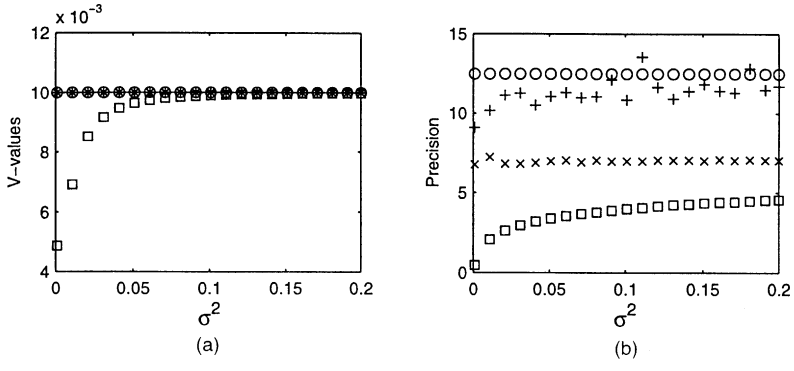

Fig. 2. (a) V and its approximations and (b) precisions of approximations to V: ———, V; □, SEM; ○, RES; +, REM; ×, FDM

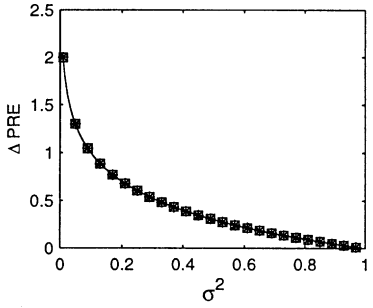

Fig. 3. Error magnification: ———, $-\log(1/\dot{M}-1)$; □, SEM; ×, FDM; +, REM

REM plot varies around 11 significant digits. The precision in the FDM plot is always about seven significant digits and that in the SEM plot varies from about 0.5 significant digits when σ^2 is small to about 4.5 significant digits when $\sigma^2 = 0.2$. The RES curve is constant because the RES method does not involve σ^2 or $M(\theta)$ for that matter. The jagged appearance of the REM curve results from the fact that Richardson extrapolation has eliminated almost all of the truncation error and what remains is primarily the rounding error which tends to display a random behaviour. The SEM algorithms for this example were started at $\boldsymbol{\theta}^{(0)} = 1$.

All three of the NDM algorithms FDM, REM and SEM face the error magnification problem associated with slow convergence, discussed in Section 4, but the forward difference method FDM appears to have been protected by self-adjustment. When $\boldsymbol{\theta}$ is a single parameter, it follows from approximation (5) that

$$\Delta\text{PRE} \approx -\log_{10}\left(\frac{1}{\dot{M}}-1\right), \tag{9}$$

where $\Delta\text{PRE} = \text{PRE}(\dot{M}^*) - \text{PRE}(V^*)$, the precision loss due to error magnification. This approximation expresses error magnification as a function of $\dot{M}$. As the EM algorithm slows, $\dot{M}$ approaches 1, and the precision loss ΔPRE increases. Fig. 3 shows how well approximation (9) holds for this example. The full curve in Fig. 3 is a plot of the right-hand side of approximation (9) on σ^2, for σ^2 ranging from 0.01 to 1. The other points shown are plots of the left-hand side of approximation (9) on σ^2 for REM, FDM and SEM_{12}. To the resolution of the plot all the values of ΔPRE lie on the full curve, indicating that expression (9) is a good approximation. From the plot, as σ^2 moves from 0.01 to 1, the loss of precision in V^* ranges from approximately two to zero digits.

Fig. 4 displays the self-adjustment of the FDM algorithm. It is a plot of precision of $\dot{M}^*$ on σ^2. As predicted in Section 4, the precision of $\dot{M}^*$ is higher when the EM algorithm is slow because of smaller truncation errors. Since the SEM algorithm uses a forward difference method, it also should be more accurate when the EM algorithm converges slowly. But, as discussed, the large increments used by the SEM algorithm in this case overtake the effect of self-adjustment as is shown in Fig. 2.

5.3. Poisson mixtures

Let $y_1, \ldots, y_n$ be a sample from a mixture

$$f(y|\boldsymbol{\theta}) = \gamma\frac{e^{-\theta_1}\theta_1^y}{y!} + (1-\gamma)\frac{e^{-\theta_2}\theta_2^y}{y!} \tag{10}$$

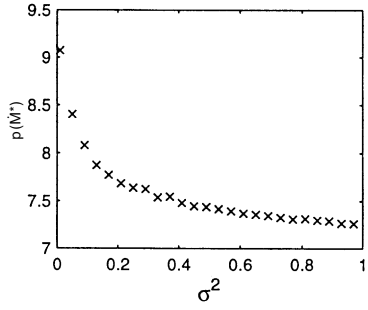

Fig. 4. FDM self-adjustment

of two Poisson densities. Lange (1995) described the standard EM algorithm for obtaining a maximum likelihood estimate of the parameter vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \gamma)^T$. Jamshidian and Jennrich (1997) gave the related functions $l(\boldsymbol{\theta})$, $M(\boldsymbol{\theta})$, $Q(\boldsymbol{\theta}', \boldsymbol{\theta})$, $\dot{Q}(\boldsymbol{\theta}', \boldsymbol{\theta})$ and $\ddot{Q}(\boldsymbol{\theta}', \boldsymbol{\theta})$.

We consider three examples in this area. The first two use $n = 2000$ data points generated artificially from Poisson mixture distributions with population parameter values $\boldsymbol{\theta} = (1, 15, 0.5)^T$ and $\boldsymbol{\theta} = (1, 5, 0.3)^T$. The third example uses a set of real data considered by Hasselblad (1969), Titterington *et al.* (1985) and Lange (1995). For this example, y_i is the number of death notices for women aged 80 years and over on each day i over a 3-year period.

Table 1 shows the comparisons for these three examples. From the $\lambda_{\max}$-values the EM algorithm converges very fast for the first example, moderately fast for the second example and slowly for the third example.

The precision PRE, defined by equation (6), and its estimate $\widehat{\text{PRE}}$, defined by equation (8), are reported in Table 1 for each example and method. In all cases $\widehat{\text{PRE}}$ approximates PRE very well. The maximum absolute error of $\widehat{\text{PRE}}$ in estimating PRE is 0.8 digits.

In all three examples the RES and REM algorithms estimate V with 10 or more digits of precision. The precision of FDM ranges from 5.4 to 8.0 in the three examples, which is satisfactory for many applications. The SEM_8 algorithm is less accurate than SEM_{12} in all three examples, and both SEM_8 and SEM_{12} are less accurate than the other three algorithms. In example 3 where the EM algorithm converges very slowly the SEM method is notably less accurate than the other algorithms with SEM_8 having essentially no precision.

As noted, both the FDM and the SEM algorithms use the forward difference method of differentiation. The main reason for the smaller precision of the SEM algorithm compared with that of FDM is the larger relative increments used by the SEM algorithm. The median

Table 1. PRE and $\widehat{\text{PRE}}$ (in parentheses) for five estimates of V for the Poisson mixture examples

Example	$\lambda_{\max}$	*PRE- and $\widehat{PRE}$-values for the following algorithms:*				
		RES	*REM*	*FDM*	SEM_8	SEM_{12}
1	0.0576	11.3	11.3	8.0	4.9	6.8
		(11.8)	(11.7)	(8.5)	(5.0)	(7.0)
2	0.8452	10.7	10.7	6.6	2.0	3.8
		(11.1)	(11.0)	(7.4)	(2.5)	(4.3)
3	0.9957	10.0	11.0	5.4	0.2	1.1
		(10.4)	(11.1)	(6.1)	(0.8)	(1.4)

Table 2. Median relative increments

Example	SEM_8 *increment*	SEM_{12} *increment*
1	0.00006	0.000002
2	0.007	0.00006
3	0.04	0.002

relative increments used by the SEM_8 and SEM_{12} algorithms in the three examples are shown in Table 2. These values are larger, and in example 3 much larger, than the relative increment 10^{-7} used by the FDM algorithm. As in our simple example, as the EM algorithm converges more slowly the SEM algorithm's relative increments grow larger and self-adjustment is not sufficient to offset the effect of truncation errors introduced by the large increments.

Error magnification, as discussed in Section 4, has also affected the performance of all the NDM algorithms in example 3. Define the precision of $\dot{M}^*$ as $p(\dot{M}^*) = -\log_{10}(\|\Delta\dot{M}\|/\|\dot{M}\|)$, and that of V^* similarly. The losses of precision $p(\dot{M}^*) - p(V^*)$ due to error magnification were 0.9, 2.2, 1.9 and 2.0 digits for the REM, FDM, SEM_8 and SEM_{12} algorithms in this example.

5.4. Estimation of covariance from incomplete data

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be a sample from an m-dimensional normal distribution with mean $\mathbf{0}$ and covariance matrix Σ. Suppose that a subvector of each $\mathbf{x}_i$, say $\mathbf{y}_i$, is observed and that the distribution of $\mathbf{y}_i$ is the corresponding marginal of the distribution of $\mathbf{x}_i$. The $\mathbf{y}_i$ might arise from a missing data mechanism such as missing completely at random or missing at random. The problem is to obtain a maximum likelihood estimate of Σ based on the $\mathbf{y}_i$. Jamshidian and Jennrich (1997) described the standard EM algorithm and gave the related functions $S(\boldsymbol{\theta})$, $M(\boldsymbol{\theta})$, $Q(\boldsymbol{\theta}', \boldsymbol{\theta})$, $\dot{Q}(\boldsymbol{\theta}', \boldsymbol{\theta})$ and $\ddot{Q}(\boldsymbol{\theta}', \boldsymbol{\theta})$ for this example. As in the previous subsection, we consider three examples in this area to evaluate the RES, REM, FDM and SEM approximations of V.

For the first example we artificially generated a sample of size 100 from a multivariate normal distribution with $m = 3$, mean $\mathbf{0}$, and covariances $\Sigma_{ij} = 0.2 + 0.8\delta_{ij}$, where δ_{ij} denotes the Kronecker delta. Then we discarded 20% of the data at random to obtain an incomplete data set. The second example uses incomplete real data from a rheumatoid arthritis study (Spiegel *et al.*, 1986) with 92 subjects and eight variables. The variables are four 50-ft walk times in seconds and the number of assistance devices, ranging from 0 to 16, that each patient used on each walk. Finally, the third example uses an artificial data set given by Jamshidian and Jennrich (1993) in their Table 2.

Table 3 shows $\lambda_{\max}$, PRE and $\widehat{\text{PRE}}$ for each of the three examples. As revealed by the $\lambda_{\max}$-values, the EM algorithm converges quite fast for the first example, moderately slowly for the second and very slowly for the third. Again $\widehat{\text{PRE}}$ estimates PRE very well with a maximum absolute error of 0.3 digits.

The RES and REM algorithms performed very well in all three examples with accuracies ranging from 8.9 to 11.3 digits. The FDM algorithm is also reasonably accurate with precisions ranging from 4.7 to 7.6. The SEM algorithms performed well on example 1, where the EM algorithm is fast. However, their precision decreased in examples 2 and 3 where the EM algorithm converged more slowly. The SEM estimates of V had zero precision in example 3.

Again, large increments are the main cause of the poor accuracy of the SEM method in

Table 3. PRE and $\widehat{\text{PRE}}$ (in parentheses) for five estimates of V for the estimation of covariance from incomplete-data examples

Example	$\lambda_{\max}$	*PRE- and $\widehat{PRE}$-values for the following algorithms:*				
		RES	*REM*	*FDM*	*SEM_8*	*SEM_{12}*
1	0.4505	11.3	11.2	7.6	4.1	5.6
		(11.5)	(11.5)	(7.6)	(4.2)	(5.9)
2	0.9488	9.6	9.6	5.6	1.1	2.0
		(9.6)	(9.6)	(5.6)	(1.2)	(2.0)
3	0.9994	8.9	8.9	4.7	0.0†	0.0
		(8.9)	(9.0)	(5.0)	(0.0†)	(0.0†)

†Negative precision is reported as 0.0.

Table 4. Median relative increments

Example	*SEM_8 increment*	*SEM_{12} increment*
1	0.0007	0.00001
2	0.004	0.001
3	0.03	0.002

examples 2 and 3. In these examples relative increments as large as 0.3 were used by both SEM_8 and SEM_{12}. The median relative increments used by the SEM_8 and SEM_{12} algorithms on the three examples are shown in Table 4. Finally, in example 3 the losses of precision due to error magnification were 2.9, 1.7, 1.0 and 2.4 digits for the REM, FDM, SEM_8 and SEM_{12} algorithms.

6. Discussion

Progress in producing standard errors when using the EM algorithm seems to have been hampered by a general reluctance to use numerical differentiation. We have shown that in a variety of applications accurate standard errors can be readily obtained by using very simple forms of numerical differentiation. We have also empirically verified a conjecture by Meng and Rubin (1991) that the precision in the skew symmetric part of a numerically computed covariance matrix accurately predicts the precision in the symmetric part.

Of the algorithms discussed, we in general prefer algorithms that numerically differentiate $S(\boldsymbol{\theta})$ to those that numerically differentiate $M(\boldsymbol{\theta})$ because this is the direct approach and it does not suffer from the error magnification problem when the EM algorithm is slow. Moreover differentiating $S(\boldsymbol{\theta})$ is appropriate for all maximum likelihood applications, not just EM applications. Exceptions to this general preference are applications where it is simpler to compute $\ddot{Q}$ than $\dot{Q}(\boldsymbol{\theta}, \boldsymbol{\theta})$. We know of few EM applications of this type, however. In general $\dot{Q}(\boldsymbol{\theta}, \boldsymbol{\theta})$ is simpler to compute than $\ddot{Q}$ and can be very much simpler. An example is factor analysis with incompletely observed responses (see for example Jamshidian and Bentler (1999)).

For numerical differentiation of $M(\boldsymbol{\theta})$, we prefer Richardson extrapolation to forward differences because a small increase in complexity gives a significant increase in precision. An exception is when computing speed is important, since the forward difference method is about four times as fast as Richardson extrapolation. We prefer algorithms that differentiate $M(\boldsymbol{\theta})$ directly to the iterative SEM algorithm because they are simpler and in our examples more precise. They can also be considerably faster, but this is usually of less importance.

The methods discussed should be applied in settings where the analytic calculation of derivatives is cumbersome or impossible. When feasible, analytic derivatives are more precise and avoid the inherent instability that is associated with numerical differentiation. One should also consider alternatives to the EM algorithm for maximum likelihood estimation. Many of these produce standard errors as a by-product, e.g. the Fisher scoring and Newton algorithms, and some accelerated forms of the EM algorithm (e.g. Jamshidian and Jennrich (1997)).

Acknowledgements

Mortaza Jamshidian's research was supported in part by grant DA01070 from the National Institute on Drug Abuse.

The authors would like to thank the Joint Editor, the Associate Editor and the referee for their very helpful comments that led to a much improved presentation.

Appendix A

A.1. Derivation of equation (1)

We shall use Fisher's result (2) to prove equation (1). Because $\boldsymbol{\theta}' = M(\boldsymbol{\theta})$ maximizes $Q(\boldsymbol{\theta}', \boldsymbol{\theta})$ given $\boldsymbol{\theta}$,

$$\dot{Q}\{M(\boldsymbol{\theta}), \boldsymbol{\theta}\} = 0. \tag{11}$$

Let $\ddot{Q}_{12}(\boldsymbol{\theta}', \boldsymbol{\theta}) = (\partial/\partial\boldsymbol{\theta})\dot{Q}(\boldsymbol{\theta}', \boldsymbol{\theta})$. Differentiating equations (2) and (11) gives

$$H(\boldsymbol{\theta}) = \ddot{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}) + \ddot{Q}_{12}(\boldsymbol{\theta}, \boldsymbol{\theta}),$$

$$0 = \ddot{Q}\{M(\boldsymbol{\theta}), \boldsymbol{\theta}\}\dot{M}(\boldsymbol{\theta}) + \ddot{Q}_{12}\{M(\boldsymbol{\theta}), \boldsymbol{\theta}\}.$$

Evaluating both of these equations at $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$, using the fact that $M(\hat{\boldsymbol{\theta}}) = \hat{\boldsymbol{\theta}}$, and eliminating $\ddot{Q}_{12}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}})$ by subtracting the second equation from the first gives equation (1).

A.2. Derivation of equation (5)

From equation (1), $V^* = -(I - \dot{M}^*)^{-1}\ddot{Q}^{-1}$. Viewing V^* as a function of $\dot{M}^*$, the differential of V^* at $\dot{M}$ is

$$\mathrm{d}V^* = (I - \dot{M})^{-1}\,\mathrm{d}\dot{M}^* V.$$

It follows from the definition of the differential that

$$\Delta V = (I - \dot{M})^{-1}\Delta\dot{M}V + o(\Delta\dot{M}).$$

A.3. Increment inflation

We have noted in our examples that when the EM algorithm converges slowly the SEM algorithm tends to use large increments in its forward difference approximations. We shall show why this may be the case in general.

Consider the one-parameter case and assume that θ^* actually equals $\hat{\theta}$. Then

$$r^{(n)} = \frac{M(\theta^{(n)}) - \hat{\theta}}{\theta^{(n)} - \hat{\theta}} \approx \dot{M} + \frac{1}{2}\ddot{M}(\theta^{(n)} - \hat{\theta}).$$

Thus

$$r^{(n+1)} - r^{(n)} \approx \tfrac{1}{2}\ddot{M}(\theta^{(n+1)} - \theta^{(n)}).$$

Since

$$\theta^{(n+1)} - \hat{\theta} = M(\theta^{(n)}) - M(\hat{\theta}) \approx \dot{M}(\theta^{(n)} - \hat{\theta}),$$
$$r^{(n+1)} - r^{(n)} \approx \tfrac{1}{2}\ddot{M}(\dot{M} - 1)(\theta^{(n)} - \hat{\theta}).$$

Using the convergence criterion in Section 3.1, when the $r^{(n)}$-sequence converges $|r^{(n+1)} - r^{(n)}| \approx \sqrt{\epsilon}$. Thus

$$|\theta^{(n)} - \hat{\theta}| \approx 2\sqrt{\epsilon}/|\ddot{M}(\dot{M} - 1)|.$$

When the EM algorithm is slow both $\ddot{M}$ and $\dot{M} - 1$ tend to be small and therefore the SEM increment $\boldsymbol{\theta}^{(n)} - \hat{\boldsymbol{\theta}}$ tends to be large.

A.4. Derivation of equation (7)

It is sufficient to show that, for any matrix A and positive definite matrix W,

$$\sup_{\mathbf{l}} \left| \frac{\mathbf{l}^{\mathrm{T}} A \mathbf{l}}{\mathbf{l}^{\mathrm{T}} W \mathbf{l}} \right| = \| W^{-1/2} \operatorname{sym}(A) W^{-1/2} \|.$$

Let $C = \operatorname{sym}(A)$. Since $\mathbf{l}^{\mathrm{T}} A \mathbf{l} = \mathbf{l}^{\mathrm{T}} C \mathbf{l}$,

$$\sup_{\mathbf{l}} \left| \frac{\mathbf{l}^{\mathrm{T}} A \mathbf{l}}{\mathbf{l}^{\mathrm{T}} W \mathbf{l}} \right| = \sup_{\mathbf{l}} \left| \frac{\mathbf{l}^{\mathrm{T}} W^{-1/2} C W^{-1/2} \mathbf{l}}{\mathbf{l}^{\mathrm{T}} \mathbf{l}} \right| = \| W^{-1/2} C W^{-1/2} \|.$$

References

Baker, S. G. (1992) A simple method for computing the observed information matrix when using the EM algorithm with categorical data. *J. Comput. Graph. Statist.*, **1**, 63–76.

Belin, T. R. and Rubin, D. B. (1995) A method for calibrating false-match rates in record linkage. *J. Am. Statist. Ass.*, **90**, 694–707.

Conte, S. D. and deBoor, C. (1980) *Elementary Numerical Analysis: an Algorithmic Approach*. New York: McGraw-Hill.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977) Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. R. Statist. Soc.* B, **39**, 1–38.

Dennis, J. E. and Schnabel, R. B. (1983) *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs: Prentice Hall.

Fisher, R. A. (1925) Theory of statistical estimation. *Proc. Camb. Phil. Soc.*, **22**, 700–725.

Hasselblad, V. (1969) Estimation of finite mixtures of distributions from the exponential family. *J. Am. Statist. Ass.*, **64**, 1459–1471.

Jamshidian, M. and Bentler, P. M. (1999) ML estimation of mean and covariance structures with missing data using complete data routines. *J. Educ. Behav. Statist.*, **23**, 21–41.

Jamshidian, M. and Jennrich, R. I. (1993) Conjugate gradient acceleration of the EM algorithm. *J. Am. Statist. Ass.*, **88**, 221–228.

——— (1997) Acceleration of the EM algorithm by using quasi-Newton methods. *J. R. Statist. Soc.* B, **59**, 569–587.

Kim, D. K. and Taylor, J. M. G. (1995) The restricted EM algorithm for maximum likelihood estimation under linear restrictions on the parameters. *J. Am. Statist. Ass.*, **90**, 708–716.

Lange, K. (1995) A quasi-Newton acceleration of the EM algorithm. *Statist. Sin.*, **5**, 1–18.

Lindsey, J. C. and Ryan, L. M. (1993) A three-state multiplicative model for rodent tumorigenicity experiments. *Appl. Statist.*, **42**, 283–300.

Little, R. J. A. (1995) Modeling the drop-out mechanism in repeated-measures studies. *J. Am. Statist. Ass.*, **90**, 1112–1121.

McCulloch, C. E. (1994) Maximum likelihood variance components estimation for binary data. *J. Am. Statist. Ass.*, **89**, 330–335.

——— (1998) Review of *The EM Algorithm and Its Extensions*. *J. Am. Statist. Ass.*, **93**, 403–404.

McLachlan, G. J. and Krishnan, T. (1997) *The EM Algorithm and Extensions*. New York: Wiley.

Meng, X. L. and Rubin, D. B. (1991) Using EM to obtain asymptotic variance-covariance matrices: the SEM algorithm. *J. Am. Statist. Ass.*, **86**, 899–909.

Pagano, M., Tu, X. M., Gruttola, V. D. and MaWhinney, S. (1994) Regression analysis of censored and truncated data: estimating reporting-delay distributions and AIDS incidence from surveillance data. *Biometrics*, **50**, 1203–1214.

Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling, W. T. (1992) *Numerical Recipes in FORTRAN: the Art of Scientific Computing*, 2nd edn. Cambridge: Cambridge University Press.

Rubin, D. B., Stern, H. S. and Vehovar, V. (1995) Handling 'don't know' survey responses: the case of the Slovenian Plebiscite. *J. Am. Statist. Ass.*, **90**, 822–828.

Segal, M. R., Bacchetti, P. and Jewell, N. P. (1994) Variance for maximum penalized likelihood estimates obtained via the EM algorithm. *J. R. Statist. Soc.* B, **56**, 345–352.

Spiegel, J. S., Spiegel, T. M., Ward, N. B., Paulus, H. E., Leake, B. and Kane, R. L. (1986) Rehabilitation for rheumatoid arthritis patients, a controlled trial. *Arth. Rheum.*, **29**, 628–637.

Titterington, D. M., Smith, A. F. M. and Makov, U. E. (1985) *Statistical Analysis of Finite Mixture Distributions*. New York: Wiley.

Tu, X. M., Meng, X. L. and Pagano, M. (1993) The AIDS epidemic: estimating survival after AIDS diagnosis from surveillance data. *J. Am. Statist. Ass.*, **88**, 26–36.

Wu, C. F. J. (1983) On the convergence properties of the EM algorithm. *Ann. Statist.*, **11**, 95–103.