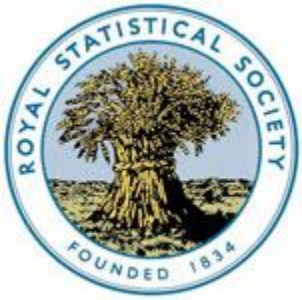

J. R. Statist. Soc. B (2005)
67, *Part* 2, *pp.* 235–251

Ascent-based Monte Carlo expectation–maximization

Brian S. Caffo,

Johns Hopkins University, Baltimore, USA

Wolfgang Jank

University of Maryland, College Park, USA

and Galin L. Jones

University of Minnesota, Minneapolis, USA

[Received October 2003. Revised July 2004]

Summary. The expectation–maximization (EM) algorithm is a popular tool for maximizing likelihood functions in the presence of missing data. Unfortunately, EM often requires the evaluation of analytically intractable and high dimensional integrals. The Monte Carlo EM (MCEM) algorithm is the natural extension of EM that employs Monte Carlo methods to estimate the relevant integrals. Typically, a very large Monte Carlo sample size is required to estimate these integrals within an acceptable tolerance when the algorithm is near convergence. Even if this sample size were known at the onset of implementation of MCEM, its use throughout all iterations is wasteful, especially when accurate starting values are not available. We propose a data-driven strategy for controlling Monte Carlo resources in MCEM. The algorithm proposed improves on similar existing methods by recovering EM's ascent (i.e. likelihood increasing) property with high probability, being more robust to the effect of user-defined inputs and handling classical Monte Carlo and Markov chain Monte Carlo methods within a common framework. Because of the first of these properties we refer to the algorithm as 'ascent-based MCEM'. We apply ascent-based MCEM to a variety of examples, including one where it is used to accelerate the convergence of deterministic EM dramatically.

Keywords: EM algorithm; Empirical Bayes estimates; Generalized linear mixed models; Importance sampling; Markov chain; Monte Carlo methods

1. Introduction

Since the seminal article of Dempster *et al.* (1977), the expectation–maximization (EM) algorithm has become a highly appreciated tool for maximizing probability models in the presence of missing data. Each iteration of an EM algorithm formally consists of an E-step and a separate M-step. The E-step calculates a conditional expectation whereas the M-step maximizes this expectation. Often, at least one of these steps is analytically intractable. Many researchers have suggested that a troublesome E-step may be overcome by approximating the expectation with Monte Carlo methods (see, for example, Booth and Hobert (1999), McCulloch (1994, 1997), Shi and Copas (2002) and Wei and Tanner (1990)). This is the Monte Carlo EM (MCEM) algorithm.

Address for correspondence: Brian S. Caffo, Department of Biostatistics, Johns Hopkins University, 615 Wolfe Street, Baltimore, MD 21205, USA.
E-mail: bcaffo@jhsph.edu

 1369–7412/05/67235

Despite their popularity, current implementations of MCEM have several drawbacks. First, these methods do not typically admit both independent sampling and Markov chain Monte Carlo (MCMC) techniques within a common framework. Second, they do not attempt to mimic some of the fundamentally appealing properties of the underlying EM algorithm. Third, many versions, e.g. some schemes that average over some part of the sequence of parameter estimates (Polyak and Juditsky, 1992; Shi and Copas, 2002), appear difficult to automate. Fourth, their behaviour depends on the parameterization of the model. Finally, the focus is on the convergence of the parameter estimates and hence the resulting Monte Carlo sample sizes are often too small to be of use for inferential or other purposes after completion of the algorithm. For example, additional simulation is typically required to obtain a good estimate of the observed information. Our intent is to build on the work of McCulloch (1994, 1997) and Booth and Hobert (1999) by studying a data-driven automated MCEM algorithm that seeks to overcome these difficulties.

Let Y denote a vector of observed data and U denote a vector of missing data and let λ be a vector of unknown parameters. Finally, $f_{Y,U}(y, u; \lambda)$ denotes the probability model of the complete data, (Y, U). Our objective is to obtain the maximizer $\hat{\lambda}$ of

$$L(\lambda; y) = \int f_{Y,U}(y, u; \lambda)\, \mathrm{d}u. \tag{1}$$

Instead of directly maximizing equation (1), the EM algorithm operates on the so-called Q-function. Let $\lambda^{(t-1)}$ be the current estimate of $\hat{\lambda}$. Then the tth E-step calculates

$$Q(\lambda, \lambda^{(t-1)}) = E[\log\{f_{Y,U}(y, u; \lambda)\} | y, \lambda^{(t-1)}]; \tag{2}$$

then in the tth M-step we require a value $\lambda^{(t)}$ that satisfies $Q(\lambda^{(t)}, \lambda^{(t-1)}) \geqslant Q(\lambda, \lambda^{(t-1)})$ for all λ in the parameter space. It is possible (Wu, 1983) to implement an incomplete M-step that only requires $\lambda^{(t)}$ to satisfy

$$Q(\lambda^{(t)}, \lambda^{(t-1)}) \geqslant Q(\lambda^{(t-1)}, \lambda^{(t-1)}) \tag{3}$$

which yields a *generalized EM* algorithm. The *ascent property* is obtained with an application of Jensen's inequality to expression (3), i.e.

$$L(\lambda^{(t)}; y) \geqslant L(\lambda^{(t-1)}; y). \tag{4}$$

Thus, at worst, each iteration of a generalized EM algorithm yields a better estimate of $\hat{\lambda}$. In fact, given an initial value $\lambda^{(0)}$, a generalized EM algorithm produces a sequence $\{\lambda^{(0)}, \lambda^{(1)}, \lambda^{(2)}, \ldots\}$ that, under regularity conditions, converges to $\hat{\lambda}$.

When the integral in equation (2) is analytically intractable or very high dimensional the MCEM algorithm approximates it with either classical Monte Carlo or MCMC methods. (See Lange (1999) or Robert and Casella (1999) for an introduction to the sampling methods that are used in this paper.) Let $\tilde{\lambda}^{(t-1)}$ denote the current MCEM estimate of $\hat{\lambda}$. Throughout we assume that $\{u^{(t,1)}, \ldots, u^{(t,m_t)}\}$ is either

(a) a random sample from $f_{U|Y}(u|y, \tilde{\lambda}^{(t-1)})$,
(b) a sample that is obtained from a candidate $h(u)$ with a set of associated importance weights $\{w(u^{(t,j)})\}$ or
(c) obtained by simulating an ergodic Markov chain with invariant density $f_{U|Y}(u|y, \tilde{\lambda}^{(t-1)})$.

Then, appealing to the appropriate strong law, we can approximate the expectation in equation (2) via

$$\tilde{Q}(\lambda, \tilde{\lambda}^{(t-1)}) = \frac{\sum_{j=1}^{m_t} w(u^{(t,j)}) \log\{f_{Y,U}(y, u^{(t,j)}; \lambda)\}}{\sum_{j=1}^{m_t} w(u^{(t,j)})} \tag{5}$$

where the importance weights are set to 1 in settings (a) and (c). The MCEM algorithm uses $\tilde{Q}$ in place of Q, i.e. the tth M-step consists of finding a value $\tilde{\lambda}^{(t)}$ such that

$$\tilde{Q}(\tilde{\lambda}^{(t)}, \tilde{\lambda}^{(t-1)}) \geqslant \tilde{Q}(\tilde{\lambda}^{(t-1)}, \tilde{\lambda}^{(t-1)}).$$

If the Monte Carlo sample size is constant across MCEM steps, i.e. $m_t = m$ for all t, an MCEM algorithm will not converge because of a persistent Monte Carlo error. This can sometimes be overcome by deterministically increasing m_t with t. However, automated data-driven strategies are needed to make efficient use of Monte Carlo resources across EM iterations. It is clear that a method for assessing the Monte Carlo error at each step is required for automated MCEM. Booth and Hobert (1999) made the first serious attempt (which was later extended by Levine and Casella (2001)) in this direction. We compare Booth and Hobert's (1999) method with ascent-based MCEM in Section 3.1.

The basic approach of ascent-based MCEM follows. Within each MCEM iteration an asymptotic lower bound is calculated for

$$Q(\tilde{\lambda}^{(t)}, \tilde{\lambda}^{(t-1)}) - Q(\tilde{\lambda}^{(t-1)}, \tilde{\lambda}^{(t-1)}).$$

If the lower bound is positive, the new parameter estimate is accepted and the algorithm moves on. If the lower bound is negative, this estimate of $\hat{\lambda}$ is rejected. We then generate another Monte Carlo sample, append it to the existing sample and obtain a new parameter estimate by using the larger Monte Carlo sample. This process is repeated until the lower bound is positive. A standard sample size calculation is used to determine the starting sample size for the next step.

Our empirical investigations suggest that ascent-based MCEM has several desirable features over existing MCEM algorithms. First, because the focus is on the Q-function, a univariate quantity on the log-scale, many of the required calculations are simple. In particular, this makes the use of MCMC sampling more straightforward than many existing approaches. Also, counterproductive use of the simulated data is rare since the ascent property holds with high probability. This, in turn, makes the algorithm stable in the sense that parameter estimates often follow a fairly smooth path to their limit. Also, ascent-based MCEM produces an appropriately large Monte Carlo sample in the final EM iteration to obtain a stable approximation of the asymptotic variance–covariance matrix of the parameter estimates by using standard methods: a clear requirement for the algorithm to be useful. This is in contrast with our experience with other MCEM algorithms that require additional simulation after completion of the MCEM algorithm to obtain a good estimate of the information matrix. (See Booth and Hobert (1999) and Gueorguieva and Agresti (2001) for some additional discussion.) Finally, since it is based on the Q-function, ascent-based MCEM is invariant to model reparameterizations. The following toy example illustrates some of these features.

1.1. Example 1

Consider the following conditionally independent model: suppose that $Y_i|u_i \sim N(u_i, 1)$ and $U_i \sim N(0, \lambda)$ for $i = 1, \ldots, n$. Table 1 displays data simulated according to this model with $n = 5$ and $\lambda = 1$. The maxmimum likelihood estimate (MLE) is $\hat{\lambda} = 1.3183$. Consider Figs 1(c) and 1(d) (we shall return to Fig. 1 later), which contain plots of the marginal log-likelihood paths for EM

Table 1. Simulated data for example 1

y_i	0.3364675	−2.6338934	0.9080410	1.8897579	−0.3811235

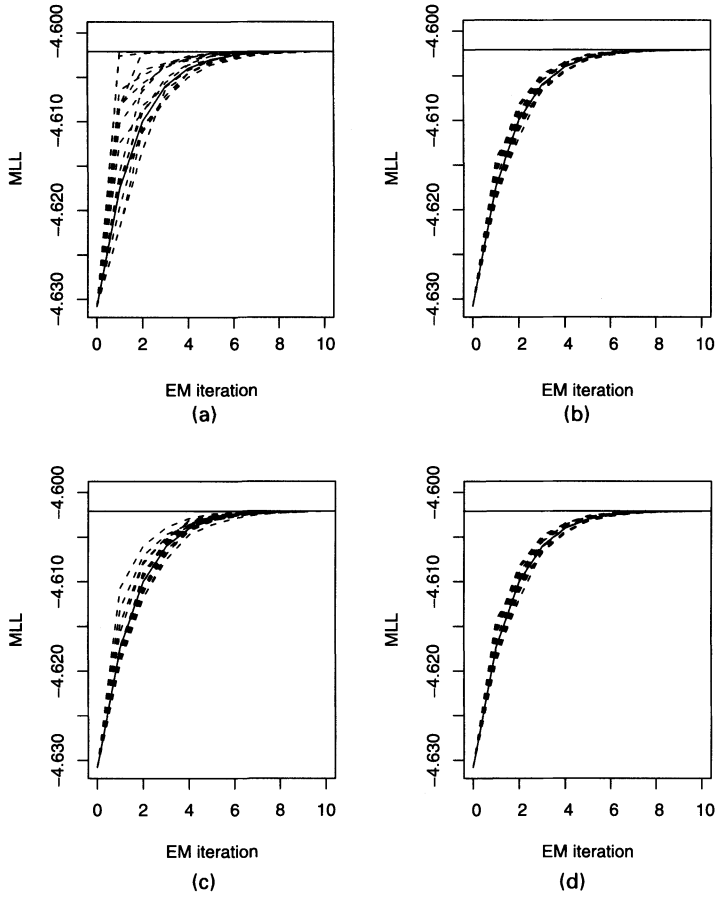

Fig. 1. Marginal log-likelihood path plots for ascent-based MCEM applied to Gaussian data for (a) $\alpha = 0.3$, $\beta = 0.25$ and rejection sampling, (b) $\alpha = 0.3$, $\beta = 0.25$ and an MCMC independence sampler, (c) $\alpha = 0.1$, $\beta = 0.25$ and rejection sampling and (d) $\alpha = 0.1$, $\beta = 0.25$ and an MCMC independence sampler (———, maximum log-likelihood; ⌒, marginal log-likelihood path of deterministic EM)

and 15 replications of ascent-based MCEM under two sampling schemes. In Fig. 1(c) we used rejection sampling to draw a random sample from $f_{U|Y}(u|y; \tilde{\lambda}^{(t-1)})$ whereas in Fig. 1(d) we used a Metropolis–Hastings independence sampler having invariant density $f_{U|Y}(u|y; \tilde{\lambda}^{(t-1)})$ and an $N(0, \tilde{\lambda}^{(t-1)})$ candidate density. We started all three algorithms at 1, i.e. $\lambda^{(0)} = \tilde{\lambda}^{(0)} = 1$. Fig. 1 indicates that, at least in this example, ascent-based MCEM mimics EM well and appears to recover the ascent property.

The rest of the paper is organized as follows. Ascent-based MCEM is developed in Section 2. In Section 3 we examine the performance of ascent-based MCEM in several examples.

2. Ascent-based Monte Carlo expectation–maximization

2.1. Recovering the ascent property

Recall that $\tilde{\lambda}^{(t-1)}$ denotes the current MCEM parameter estimate and that $\{u^{(t,j)}\}_{j=1}^{m_t}$ is the Monte Carlo sample. Let $\tilde{\lambda}^{(t,m_t)}$ be the corresponding maximizer of $\tilde{Q}(\lambda, \tilde{\lambda}^{(t-1)})$. Since an increase in the Q-function implies the ascent property, we shall check inequality (4) by checking inequality (3), i.e. the algorithm requires evidence that

$$\Delta Q(\tilde{\lambda}^{(t,m_t)}, \tilde{\lambda}^{(t-1)}) \equiv Q(\tilde{\lambda}^{(t,m_t)}, \tilde{\lambda}^{(t-1)}) - Q(\tilde{\lambda}^{(t-1)}, \tilde{\lambda}^{(t-1)}) > 0 \tag{6}$$

before accepting $\tilde{\lambda}^{(t,m_t)}$ and proceeding to the next MCEM step. We can appeal to the appropriate version of the strong law and estimate $\Delta Q(\tilde{\lambda}^{(t,m_t)}, \tilde{\lambda}^{(t-1)})$ consistently with

$$\begin{aligned}\Delta \tilde{Q}(\tilde{\lambda}^{(t,m_t)}, \tilde{\lambda}^{(t-1)}) &\equiv \tilde{Q}(\tilde{\lambda}^{(t,m_t)}, \tilde{\lambda}^{(t-1)}) - \tilde{Q}(\tilde{\lambda}^{(t-1)}, \tilde{\lambda}^{(t-1)}) \\ &= \frac{\sum_{j=1}^{m_t} w(u^{(t,j)}) \log\{f_{Y,U}(y, u^{(t,j)}; \tilde{\lambda}^{(t,m_t)}) / f_{Y,U}(y, u^{(t,j)}; \tilde{\lambda}^{(t-1)})\}}{\sum_{j=1}^{m_t} w(u^{(t,j)})},\end{aligned} \tag{7}$$

where $w(u^{(t,k)})$ denote the importance weights which are set equal to 1 if sampling directly from $f_{U|Y}$ or MCMC sampling is employed.

We shall now argue that, when suitably normalized,

$$\Delta \tilde{Q}(\tilde{\lambda}^{(t,m_t)}, \tilde{\lambda}^{(t-1)}) - \Delta Q(\tilde{\lambda}^{(t,m_t)}, \tilde{\lambda}^{(t-1)}) \tag{8}$$

has a limiting normal distribution with mean 0 and a variance σ^2 that depends on the sampling mechanism employed. For simplicity we shall assume that the sampling mechanism produces independent and identically distributed (IID) samples from $f_{U|Y}(u|y, \tilde{\lambda}^{(t-1)})$. However, all these arguments will go through more generally. For example, if MCMC sampling is employed then use of the split chain (Meyn and Tweedie, 1993; Nummelin, 1984) will be required to extend our argument. Recall that Booth and Hobert (1999), page 272, proved the asymptotic normality of $\sqrt{m_t}(\tilde{\lambda}^{(t)} - \lambda^{(t)})$ and consider

$$\begin{aligned}\sqrt{m_t}\{\Delta \tilde{Q}(\tilde{\lambda}^{(t)}; \tilde{\lambda}^{(t-1)}) - \Delta Q(\tilde{\lambda}^{(t)}; \tilde{\lambda}^{(t-1)})\} &= \sqrt{m_t}\{\Delta \tilde{Q}(\lambda^{(t)}; \tilde{\lambda}^{(t-1)}) - \Delta Q(\lambda^{(t)}; \tilde{\lambda}^{(t-1)})\} && (9) \\ &+ \sqrt{m_t}\{\Delta \tilde{Q}(\tilde{\lambda}^{(t)}; \tilde{\lambda}^{(t-1)}) - \Delta \tilde{Q}(\lambda^{(t)}; \tilde{\lambda}^{(t-1)})\} && (10) \\ &+ \sqrt{m_t}\{\Delta Q(\lambda^{(t)}; \tilde{\lambda}^{(t-1)}) - \Delta Q(\tilde{\lambda}^{(t)}; \tilde{\lambda}^{(t-1)})\}. && (11)\end{aligned}$$

Note that term (9) is asymptotically $N(0, \sigma^2)$ and standard Taylor series arguments show that terms (10) and (11) converge in probability to 0.

Given an estimate, $\hat{\sigma}^2$ say, of σ^2 we can calculate an asymptotic standard error (ASE) for expression (8). (Calculation of ASE is deferred until Section 2.2.) Let z_α be such that $\Pr(Z > z_\alpha) = \alpha$ where Z is a standard normal random variable. Then

$$\Delta \tilde{Q}(\tilde{\lambda}^{(t,m_t)}, \tilde{\lambda}^{(t-1)}) - z_\alpha \text{ASE} \tag{12}$$

will be smaller than $\Delta Q(\tilde{\lambda}^{(t,m_t)}, \tilde{\lambda}^{(t-1)})$ with probability $1 - \alpha$ as $m_t \to \infty$.

If the asymptotic lower bound (12) is positive, there is sufficient evidence to conclude that $\tilde{\lambda}^{(t,m_t)}$ increases the likelihood. Thus, $\tilde{\lambda}^{(t,m_t)}$ is accepted as the tth parameter update, i.e. $\tilde{\lambda}^{(t)} = \tilde{\lambda}^{(t,m_t)}$, and $t \to t+1$. If the lower bound is negative, then the estimate of Q is deemed swamped with Monte Carlo error and a larger sample size is needed to estimate Q accurately. In this

case, the tth iteration is repeated with a larger sample size. We discuss our rule for updating the sample size in Section 2.3.

This framework can be used to develop a rule that we have found useful for stopping the iterative procedure. Define z_γ similarly to z_α and note that, as in expression (12), we also have that

$$\Delta\tilde{Q}(\tilde{\lambda}^{(t,m_t)}, \tilde{\lambda}^{(t-1)}) + z_\gamma \mathrm{ASE} \tag{13}$$

will be larger than $\Delta Q(\tilde{\lambda}^{(t,m_t)}, \tilde{\lambda}^{(t-1)})$ with probability $1-\gamma$ as $m_t \to \infty$. Thus, if the interest lies in stopping MCEM when the marginal likelihood stabilizes, then waiting until expression (13) is less than some specified constant is a convenient stopping rule. Note that this is essentially a criterion that determines when too much simulation effort will be required, i.e. when the change in the Q-function is too small to be easily detected. Thus, as with any other version of MCEM, it may be useful to examine plots of the sequence of parameter estimates *versus* MCEM iteration. Another potentially useful plot would be of the Monte Carlo estimate of the likelihood obtained via Monte Carlo maximum likelihood (MCML) (Geyer, 1994). However, it is not clear that performing additional rounds of MCML would improve the estimate of $\hat{\lambda}$. Moreover, in our experience MCML often requires Monte Carlo sample sizes that are of the same order of magnitude as MCEM, so if MCEM is computationally infeasible then MCML may be as well. For more on these issues the reader is directed to the comparisons of MCML and MCEM in McCulloch (1997), Jank and Booth (2003) and Booth *et al.* (2001).

In contrast with expression (13), a standard stopping rule (Booth and Hobert, 1999; Booth *et al.*, 2001; Searle *et al.*, 1992; Shi and Copas, 2002) is based on either a small absolute or a small relative change in the parameter estimates in consecutive iterations. Typically, this criterion must be satisfied for several consecutive iterations to guard against prematurely claiming convergence. We believe that this approach is understandable but somewhat misguided since it places the emphasis on the behaviour of the parameter estimates with little or no regard to the estimation of the information. We address this issue in the context of a bench-mark example in Section 3.1.

2.2. Monte Carlo standard errors

2.2.1. Independent sampling

If importance sampling is employed an estimate of σ^2 is given by

$$\hat{\sigma}^2 = m_t \left\{ \frac{\sum w(u^{(t,j)})\,\Lambda(u^{(t,j)})}{\sum w(u^{(t,j)})} \right\}^2 \left[\frac{\sum \{w(u^{(t,j)})\,\Lambda(u^{(t,j)})\}^2}{\{\sum w(u^{(t,j)})\,\Lambda(u^{(t,j)})\}^2} - 2\frac{\sum w^2(u^{(t,j)})\,\Lambda(u^{(t,j)})}{\{\sum w(u^{(t,j)})\,\Lambda(u^{(t,j)})\}\sum w(u^{(t,j)})} + \frac{\sum w^2(u^{(t,j)})}{\{\sum w(u^{(t,j)})^2\}} \right] \tag{14}$$

where the sums all range from $j = 1, \ldots, m_t$ and

$$\Lambda(u^{(t,j)}) = \log\left\{ \frac{f_{Y,U}(y, u^{(t,j)}; \tilde{\lambda}^{(t,m_t)})}{f_{Y,U}(y, u^{(t,j)}; \tilde{\lambda}^{(t-1)})} \right\}.$$

This is a standard formula for the variance of the ratio of two means (Kendall and Stuart (1958), page 232) applied to the importance sampling ratio estimator. With rejection sampling the importance weights are set to 1 and equation (14) reduces to the (biased) sample variance of the m_t independent terms $\Lambda(u^{(t,j)})$. Thus, in either case, it is easy to estimate ASE with $\hat{\sigma}/\sqrt{m_t}$.

2.2.2. Markov chain Monte Carlo sampling

Calculating a reasonable Monte Carlo standard error is more difficult when we are forced to employ MCMC sampling. There are several different methods for doing this, including regenerative simulation (RS) and batch means. Here we investigate the use of RS instead of batch means since it produces a strongly consistent estimate of σ^2 under weaker regularity conditions (Jones *et al.*, 2004). However, there are settings where a method such as batch means will be preferred.

We shall give only a sketch of our implementation of RS as the details have appeared elsewhere; see for example Hobert *et al.* (2002), Jones *et al.* (2004), Jones and Hobert (2001) and Mykland *et al.* (1995). The basic idea is that we simulate a Markov chain in such a way that it is possible to identify regeneration times that break the chain into tours that are IID.

Assume that the simulation is started with a regeneration (this is often easy to do; see Mykland *et al.* (1995) for some examples) and hence we do not require any burn-in. Let $0 = \tau_0 < \tau_1 < \tau_2 < \ldots$ be the regeneration times and suppose that the simulation is run for a fixed number R_t of tours. Then the total length of the simulation τ_{R_t} is random. Let $N_r = \tau_r - \tau_{r-1}$ and define

$$S_r(\lambda, \tilde{\lambda}^{(t-1)}) = \sum_{j=\tau_{r-1}}^{\tau_r - 1} \log\left\{ \frac{f_{Y,U}(y, u^{(t,j)}; \lambda)}{f_{Y,U}(y, u^{(t,j)}; \tilde{\lambda}^{(t-1)})} \right\}.$$

The $(S_r(\lambda, \tilde{\lambda}^{(t-1)}), N_r)$, $r = 1, \ldots, R_t$, pairs are IID since each is based on a different tour.

Now, under regularity conditions (Jones *et al.* 2004), a consistent estimate of the desired asymptotic variance is given by

$$\hat{\gamma}^2(\lambda, \tilde{\lambda}^{(t-1)}) = \frac{\sum_{r=1}^{R_t} [S_r(\lambda, \tilde{\lambda}^{(t-1)}) - \{\bar{S}(\lambda, \tilde{\lambda}^{(t-1)})/\bar{N}\} N_r]^2}{R_t \bar{N}^2}.$$

Since $\lambda^{(t)}$ is unknown, we estimate ASE with $\hat{\gamma}(\tilde{\lambda}^{(t,\tau_{R_t})}, \tilde{\lambda}^{(t-1)})/\sqrt{R_t}$, where $\tilde{\lambda}^{(t,\tau_{R_t})}$ denotes the maximizer of equation (5) based on R_t regenerations of the chain. Therefore, an asymptotic lower bound for $\Delta Q(\tilde{\lambda}^{(t,m_t)}, \tilde{\lambda}^{(t-1)})$ is given by

$$\Delta \tilde{Q}(\tilde{\lambda}^{(t,\tau_{R_t})}, \tilde{\lambda}^{(t-1)}) - z_\alpha \frac{\hat{\gamma}(\tilde{\lambda}^{(t,\tau_{R_t})}, \tilde{\lambda}^{(t-1)})}{\sqrt{R_t}}.$$

2.3. Updating the Monte Carlo sample size

Because the initial jumps in the EM algorithm are typically large, it has become conventional wisdom that smaller Monte Carlo sample sizes can be tolerated in the initial stages of MCEM. However, larger sample sizes will be required later to decrease the variability of $\tilde{\lambda}^{(t)}$. Therefore, we need a rule for increasing the sample size as the computation progresses.

To motivate our solution we digress and briefly consider a general MCEM algorithm, i.e. not necessarily ascent-based MCEM. The probability that the ascent property does *not* hold between two consecutive iterations is $\Pr\{Q(\tilde{\lambda}^{(t)}, \tilde{\lambda}^{(t-1)}) \leqslant Q(\tilde{\lambda}^{(t-1)}, \tilde{\lambda}^{(t-1)})\}$. Note that this probability does not correspond to α from expression (12) and if $m_t \to \infty$ then

$$\Pr\{Q(\tilde{\lambda}^{(t)}, \tilde{\lambda}^{(t-1)}) \leqslant Q(\tilde{\lambda}^{(t-1)}, \tilde{\lambda}^{(t-1)})\} \to 0.$$

Thus, regardless of the initial Monte Carlo sample size, by increasing the sample size within an E-step the algorithm will reach a point where the ascent property very probably holds.

Another interesting observation follows if we assume that the Monte Carlo sample sizes are increased between MCEM steps sufficiently fast that

$$\sum_{t=1}^{\infty} \Pr\{Q(\tilde{\lambda}^{(t)}, \tilde{\lambda}^{(t-1)}) \leqslant Q(\tilde{\lambda}^{(t-1)}, \tilde{\lambda}^{(t-1)})\} < \infty.$$

Then the Borel–Cantelli theorem (Billingsley (1995), pages 59–60) says that, with probability 1, the sequence $\{\tilde{\lambda}^{(t)}\}$ has the ascent property except for perhaps finitely many iterations. Therefore, this realization of an MCEM algorithm defines an eventual (i.e., for all MCEM iterations, $t > N$, where $N < \infty$) realization of a deterministic generalized EM algorithm.

We now return to ascent-based MCEM. Recall that the current Monte Carlo sample size m_t is repeatedly increased within the tth MCEM step until the asymptotic lower bound (12) is positive. We propose a geometric rate of increase; specifically, we set the next Monte Carlo sample size to be $m_t + m_t/k$ for some $k = 2, 3, \ldots$. The m_t/k additional samples are drawn and appended to the current sample. This process clearly trades computing time for stability. However, for automated algorithms, we believe that computing time is of less concern than confidence in the output of the algorithm. On a related note, it is possible to use an importance weighting scheme to use samples from all the previous simulations as in Booth and Hobert (1999) and Quintana *et al.* (1999). However, researchers such as Booth and Hobert (1999) found little advantage in this approach and in our experience with ascent-based MCEM it often requires an enormous amount of storage and hence negatively affects computational efficiency.

In the interest of obtaining computational efficiency and avoiding severe inflation of the type 1 error rate the starting sample size for each MCEM iteration should be chosen so that we go through the appending process infrequently. For MCEM iteration t, let $m_{t,\text{start}}$ be the starting Monte Carlo sample size and $m_{t,\text{end}}$ be the ending Monte Carlo sample size. To force an increase (on average) in the Monte Carlo sample sizes across MCEM iterations we take $m_{t+1,\text{start}} \geqslant m_{t,\text{start}}$. If we assume that

$$\Delta\tilde{Q}(\tilde{\lambda}^{(t+1)}, \tilde{\lambda}^{(t)}) \sim N\left\{\Delta Q(\lambda^{(t+1)}, \tilde{\lambda}^{(t)}), \frac{\hat{\sigma}^2}{m_{t+1}}\right\}$$

then we can use standard sample size calculations to determine the value of $m_{t+1,\text{start}}$. In particular, under independent sampling we set

$$m_{t+1,\text{start}} = \max[m_{t,\text{start}}, \hat{\sigma}^2(z_\alpha + z_\beta)^2/\{\Delta\tilde{Q}(\tilde{\lambda}^{(t)}, \tilde{\lambda}^{(t-1)})\}^2], \tag{15}$$

where β is a specified type 2 error rate and $\hat{\sigma}^2$ is an estimate of the variance of $\Delta\tilde{Q}$ that was developed in Section 2.2. Since $\Delta Q(\lambda^{(t+1)}, \tilde{\lambda}^{(t)})$ depends on unknown quantities we have replaced it with an estimate from the previous iteration. Recall that under RS we do not choose the Monte Carlo sample size but instead we fix the number of regenerations, R_t. Since these R_t tours are IID the above sample size calculation is applied to the number of regenerations.

The validity of equation (15) clearly relies on the quality of the normal approximation. A poor approximation primarily results in an inflated type 1 error rate for the lower bound. To illustrate the effect of inflating α we took the setting of example 1 and performed ascent-based MCEM for two levels of α and both sampling mechanisms. The results are given in Fig. 1. The plots indicate that, at least in this example, ascent-based MCEM recovers the ascent property even when $\alpha = 0.3$. Also, α apparently controls how tightly MCEM mimics EM. Finally, we note that a type 2 error means that $\Delta Q(\tilde{\lambda}^{(t)}, \tilde{\lambda}^{(t-1)}) > 0$ when, on the basis of expression (12), we have concluded otherwise. Hence unnecessary additional simulation will be performed within the Monte Carlo E-step. Thus, β will affect the total simulation effort.

3. Applications

3.1. A bench-mark example

Let $y_{ij}|u_i \sim \text{Bernoulli}(\pi_{ij})$ independently for $i = 1, \ldots, 10$ and $j = 1, \ldots, 15$ where

$$\text{logit}(\pi_{ij}) = \lambda_1(j/15) + u_i$$

and the u_i are IID $N(0, \lambda_2)$. This model has been examined in the context of MCEM by several researchers including Booth and Hobert (1999), Levine and Casella (2001), McCulloch (1997) and Quintana *et al.* (1999). In particular, Booth and Hobert (1999) reported data that were simulated according to this model with $\lambda_1 = 5$ and $\lambda_2 = 0.5$. Moreover, they used numerical integration to obtain the MLE $(\hat{\lambda}_1, \hat{\lambda}_2) = (6.132, 1.766)$. We used numerical integration to estimate the inverse information and obtained (1.80, 1.13, 2.55) for the three components corresponding to $\text{var}(\hat{\lambda}_1)$, $\text{cov}(\hat{\lambda}_1, \hat{\lambda}_2)$ and $\text{var}(\hat{\lambda}_2)$ respectively. Our goal in the rest of this subsection is to compare the performance of Booth and Hobert's (1999) algorithm (BH-MCEM) with ascent-based MCEM in terms of convergence of the parameter estimates and also estimation of the inverse information. More specifically, we shall evaluate the performance of both algorithms when implementing the standard practice of approximating the inverse observed information matrix by using the Monte Carlo sample from the final MCEM iteration via Louis's (1982) method.

In the BH-MCEM algorithm a $100(1-\alpha_1)\%$ confidence ellipsoid, centred at the current parameter estimate, for the true EM update is computed at the conclusion of each MCEM step. If this ellipsoid contains the parameter estimate from the previous iteration then the sample size is increased by a user-defined multiple $1/k_1$ for some $k_1 > 0$, i.e. $m_{t+1} = m_t + m_t/k_1$. However, the BH-MCEM algorithm does not reject the current estimate of the MLE which, in our experience, means that this algorithm tends to produce a much more volatile sequence of estimates than ascent-based MCEM.

We performed 10000 independent replications each of the BH-MCEM and ascent-based MCEM algorithms. For both algorithms, the simulations were drawn from the conditional distribution of $U|Y$ using an accept–reject sampler. Each replication was started at $(\tilde{\lambda}_1^{(0)}, \tilde{\lambda}_2^{(0)}) = (0, 1)$ and the initial sample size was $m_0 = 10$. Each replication was terminated when the relative change in two successive parameter updates was less than 2% for $C \geqslant 1$ consecutive MCEM steps where $C = 1$ for the ascent-based MCEM and $C = 1, 2, 3, 4$ for the BH-MCEM algorithm. With the ascent-based MCEM algorithm we used $\alpha = 0.25$ and $\beta = 0.25$ and set $k = 3$ whereas for the BH-MCEM algorithm we used $\alpha_1 = 0.25$ and $k_1 = 3$.

The results are summarized in Table 2. Reported are the mean, standard error and the median of the total simulation effort, the percentage of simulation spent on the final iteration and the relative errors in the approximations to the parameter estimates and inverse information components. For example, if $\tilde{\lambda}_{1,i}$ is the estimate of $\hat{\lambda}_1$ for ascent-based MCEM from replication i, then 0.0189 is the mean of the relative errors, $|\tilde{\lambda}_{1,i} - \hat{\lambda}_1|/|\hat{\lambda}_1|$, over the 10000 replications.

From Table 2 we notice that ascent-based MCEM tends to devote a much higher percentage of its simulation effort to the last MCEM step than does the BH-MCEM algorithm for any value of C. It is worth emphasizing this aspect of the comparison because a large final Monte Carlo sample is also useful in many problems such as prediction of random effects in mixed model applications or estimating posterior quantities in empirical Bayesian settings. Secondly, in terms of parameter estimation it appears that the BH-MCEM algorithm with $C = 2$ has roughly the same relative error as ascent-based MCEM but with less simulation effort. However, the BH-MCEM algorithm with either $C = 1$ or $C = 2$ results in substantially larger average relative errors in terms of estimating the inverse information. In contrast, if $C = 3$ then, in terms of mean and median relative error, the ascent-based MCEM and BH-MCEM algorithms estimate

Table 2. Comparison of the ascent-based MCEM algorithm with the BH-MCEM algorithm†

Method		*TotSim*	*PerSim (%)*	$RE(\hat{\lambda}_1)$	$RE(\hat{\lambda}_2)$	$RE\{var(\hat{\lambda}_1)\}$	$RE\{var(\hat{\lambda}_2)\}$	$RE\{cov(\hat{\lambda}_1, \hat{\lambda}_2)\}$
Ascent-based	Mean	2009	55	0.0189	0.0866	0.3458	0.5000	0.7234
MCEM	SE	34	0.2	0.0001	0.0006	0.0318	0.0390	0.0653
($C=1$)	Median	1131	55	0.0162	0.0761	0.1390	0.2152	0.2649
BH-MCEM	Mean	292	18	0.0257	0.1067	2.4000	2.5186	4.1203
($C=1$)	SE	2	0.4	0.0002	0.0008	0.4200	0.3002	0.5863
	Median	216	18	0.0198	0.0860	0.3646	0.5168	0.7298
BH-MCEM	Mean	1277	18	0.0120	0.0549	0.9825	1.3377	2.0687
($C=2$)	SE	11	0.1	0.0001	0.0005	0.0871	0.1148	0.1845
	Median	960	18	0.0090	0.0403	0.2099	0.2909	0.4296
BH-MCEM	Mean	2970	19	0.0076	0.0344	0.3494	0.4969	0.7528
($C=3$)	SE	23	0.1	0.0001	0.0003	0.0313	0.0548	0.0781
	Median	2419	19	0.0057	0.0257	0.1337	0.1872	0.2766
BH-MCEM	Mean	5113	19	0.0054	0.0248	0.2823	0.3584	0.5798
($C=4$)	SE	37	0.1	0.0001	0.0002	0.0636	0.0632	0.1206
	Median	4270	19	0.0042	0.0191	0.1029	0.1409	0.2112

†TotSim is the total number of simulated vectors for a single replication of MCEM whereas PerSim is the percentage of the total simulation effort that is used in the final MCEM step. $RE(\hat{\lambda}_1)$, $RE(\hat{\lambda}_2)$, $RE\{var(\hat{\lambda}_1)\}$, $RE\{var(\hat{\lambda}_2)\}$ and $RE\{cov(\hat{\lambda}_1, \hat{\lambda}_2)\}$ denote the relative errors in the approximations to the MLEs and the three inverse information components. We report the mean, standard error of the mean, SE, and the median of the 10000 replications.

the inverse information with about the same quality. Also, note that the standard errors of the mean corresponding to the inverse information estimates are larger for the BH-MCEM algorithm than those for ascent-based MCEM. This is also the case for the BH-MCEM algorithm with $C=4$. Thus, the volatility of the estimates produced by the BH-MCEM algorithm is apparent in spite of the fact that it (with $C=3,4$) employed a substantially larger simulation effort than ascent-based MCEM. This example suggests that, in terms of producing stable estimates of parameters and the inverse information, the ascent-based MCEM algorithm compares favourably with the BH-MCEM algorithm.

3.2. *A hybrid expectation–maximization–Monte Carlo expectation–maximization algorithm*

Fig. 1 indicates that, especially in its early stages, ascent-based MCEM can result in step sizes that are larger than EM. This suggests that a hybrid algorithm that starts out with MCEM and eventually switches to EM can be superior (in terms of run time) to ordinary EM. We shall use such an algorithm to obtain empirical Bayes (EB) estimates of the hyperparameters for a hierarchical model. This example illustrates how the judicious use of ascent-based MCEM can accelerate the convergence of deterministic EM.

This example is motivated by a microarray experiment which attempted to identify genes that behave differently across tissue types for subjects with varying ages. In particular, levels of 'expression' for several genes were measured across experimental conditions hoping to locate candidate genes that are differentially expressed across tissue types and ages.

Let Y_i be a vector of J responses for $i=1,\ldots,n$. Here, i represents the gene index and n represents the number of genes (n may be of the order of 10000–20000 in microarray experiments). Let x be a $J \times p$ matrix of covariates (e.g. tissue type and age of subject). Suppose that conditional on u_{1i} and u_{2i} the data Y_i are independent with

$$Y_i|u_{1i}, u_{2i} \sim N(xu_{1i}, Iu_{2i}^{-1}),$$

where I is an identity matrix. At the second stage, conditional on u_{2i} the U_{1i} are independent with

$$U_{1i}|u_{2i} \sim N(0, \lambda_1 u_{2i}^{-1}),$$

where each u_{1i} is a $p \times 1$ vector and λ_1 is a $p \times p$ matrix. Finally, the U_{2i} are assumed independent with

$$U_{2i} \sim \text{gamma}(\lambda_2, \lambda_3),$$

where λ_2 is the gamma shape parameter and λ_3 is the gamma rate parameter (then the gamma mean is λ_2/λ_3). Note that U_{1i} may be viewed as the gene-specific random mean whereas U_{2i} is the gene-specific random precision. Throughout we assume that λ_1, λ_2 and λ_3 are strictly positive so that the priors are all proper. The EB estimate of $\lambda = (\lambda_1, \lambda_2, \lambda_3)^T$ maximizes equation (1), where y is the observed data and U contains the U_{1i} and U_{2i}. Therefore, EM may be used to calculate the EB estimate of λ.

A convenient feature of EM is that $\lambda_1^{(t)}$ has a closed form and can be calculated independently of $\lambda_2^{(t)}$ and $\lambda_3^{(t)}$. However, EM is computationally burdensome in this setting since at least one complete pass through the gene index i is required at each EM iteration. This burden may be lessened by using MCEM. The basic idea is to sum over only a random subset of the gene index in the early stages of the hybrid algorithm. When the Monte Carlo sample sizes for MCEM become too large, the algorithm then switches to EM.

The Q-function (up to a scalar constant) is given by

$$Q(\lambda, \lambda^{(t-1)}) = \frac{1}{n} \sum_{i=1}^{n} E[\log\{f_{Y,U}(y_i, U_{1i}, U_{2i}; \lambda)|y_i, \lambda^{(t-1)}\}]. \tag{16}$$

Let U_3 be uniformly distributed on the integers $1, \ldots, n$, independently of y. Using U_3, we can rewrite equation (16) as

$$Q(\lambda, \lambda^{(t-1)}) = E[\log\{f_{Y,U}(y_{U_3}, U_{1U_3}, U_{2U_3}; \lambda)|y, \lambda^{(t-1)}\}], \tag{17}$$

where, for example, the notation y_{U_3} refers to the vector y_i with i evaluated at the random index U_3. It is easy to sample directly from $f_{U|Y}$ via sequential sampling as we now describe.

Let shape(i) and rate(i) be the shape and rate parameters for the gamma density of $U_{2i}|y_i, \tilde{\lambda}^{(t-1)}$. Similarly, let mean($i, u_{2i}$) and var($i, u_{2i}$) be the mean and variance corresponding to the normal density of $U_{1i}|u_{2i}, y_i, \tilde{\lambda}^{(t-1)}$. Formulae for these quantities are given in Table 3. The sequential sampling mechanism for generating from $f_{U|Y}$ is as follows.

Step 1: simulate $u_3^{(t,j)}$ uniformly on the integers $1, \ldots, n$.
Step 2: simulate $u_2^{(t,j)}$ as gamma$\{$shape$(u_3^{(t,j)})$, rate$(u_3^{(t,j)})\}$.
Step 3: simulate $u_1^{(t,j)}$ as $N\{$mean$(u_3^{(t,j)}, u_2^{(t,j)})$, var$(u_3^{(t,j)}, u_2^{(t,j)})\}$.

Table 3. Posterior parameters for the model from Section 3.2

shape$(i) = 0.5J + \tilde{\lambda}_2^{(t-1)}$
rate$(i) = 0.5y_i^T[I - X\{X^TX + (\tilde{\lambda}_1^{(t-1)})^{-1}\}^{-1}X^T]y_i + \tilde{\lambda}_2^{(t-1)}$
mean$(i, u_{2i}) = \{X^TX + (\tilde{\lambda}_1^{(t-1)})^{-1}\}^{-1}X^Ty_i$
var$(i, u_{2i}) = \{X^TX + (\tilde{\lambda}_1^{(t-1)})^{-1}\}^{-1}u_{2i}^{-1}$

With these simulated variables it is easy to form a Monte Carlo approximation to equation (17).

The algorithm proceeds by using our MCEM algorithm in the early stages and then in the later stages EM based on equation (16). A rough estimate of the computational effort, which was obtained by inspecting the computer code, suggests that at least $2m_t$ non-trivial computations are required for one MCEM iteration. In contrast, one EM iteration needs n non-trivial computations. Therefore, we switch from MCEM to EM when the ending Monte Carlo sample size for the current MCEM iteration is larger than $n/2$. Finally, note that this hybrid algorithm does not require a stochastic stopping criterion since EM is used in the final stages of the algorithm.

3.2.1. *A numerical example*

We simulated data from the assumed hierarchical model for $n = 20000$, $n = 60000$ and $n = 100000$ with λ_1, λ_2 and x (20×5) obtained from a specific microarray experiment. The simulated data are available on request.

We set $\tilde{\lambda}_1^{(0)}$ equal to the identity matrix, $\tilde{\lambda}_2^{(0)} = \tilde{\lambda}_3^{(0)} = 1$, $\alpha = 0.3$, $\beta = 0.05$ and $k = 2$ and performed 100 independent runs of the hybrid algorithm. This was repeated for each of the three values of n. We also ran an EM algorithm using the same starting values. All programs were run on the same computer and the same code was used for both EM and the EM portion of the hybrid algorithm.

The run times for the hybrid algorithm were substantially better than those for EM. In particular, when $n = 20000$, it took EM 6 min 17 s to converge. In contrast the maximum run time for the hybrid algorithm was 4 min 56 s, an improvement of 22%. The minimum, 25th, 50th and 75th percentiles of the run times for the hybrid algorithm were 1 min 29 s, 3 min 24 s, 3 min 47 s and 4 min 13 s respectively. This clearly indicates that the hybrid algorithm performs much better than deterministic EM in this example. Moreover, the improvement in run time can be even more substantial as n increases. For example, with $n = 60000$ the five-number summary for the run times for the hybrid algorithm was (3 min 25 s, 8 min 55 s, 9 min 44 s, 11 min 00 s, 13 min 23 s) whereas the EM algorithm took 19 min 00 s, i.e. the improvement ranged from 30% to 82%. When $n = 100000$ the five-number summary of the run times for the hybrid algorithm was (5 min 35 s, 13 min 21 s, 15 min 27 s, 16 min 42 s, 21 min 20 s) and EM required 31 min 46 s. There is one important *caveat* to our results: the hybrid algorithm may be less efficient than EM if the starting values are chosen very close to the (unknown) MLE.

3.3. *Empirical Bayes estimates for a hierarchical model*

3.3.1. *The model and Gibbs sampler*

Suppose that, conditional on $\theta = (\theta_1, \ldots, \theta_A)^{\mathrm{T}}$ and ν_e, the data Y_{ij} are independent with

$$Y_{ij}|\theta_i, \nu_e \sim N(\theta_i, \nu_e^{-1})$$

where $i = 1, \ldots, A$ and $j = 1, \ldots, n_i$. At the second stage, conditional on μ and ν_θ, $\theta_1, \ldots, \theta_A$ and ν_e are independent with

$$\theta_i|\mu, \nu_\theta \sim N(\mu, \nu_\theta^{-1}),$$
$$\nu_e \sim \text{gamma}(a_2, \lambda_2).$$

Finally, at the third stage, μ and ν_θ are assumed independent with

$$\mu \sim N(\mu_0, \nu_0^{-1}),$$
$$\nu_\theta \sim \text{gamma}(a_1, \lambda_1)$$

where $\mu_0, \nu_0, a_1, \lambda_1, a_2$ and λ_2 are constants and all except μ_0 are assumed to be strictly positive; hence all the priors are proper. Regardless of the prior specification on the variance components, the EB estimate of μ_0 is the overall mean $\bar{y} = \Sigma_{ij}\, y_{ij}/n$ where $n = \Sigma_i\, n_i$. Also, for reasons discussed in Section 3.3.2, we focus on estimating λ_1 and λ_2 for fixed a_1, a_2 and ν_0.

In terms of the EM notation, we have $U = (\theta, \mu, \nu_\theta, \nu_e)^{\mathrm{T}}$, and $\lambda = (\lambda_1, \lambda_2)^{\mathrm{T}}$. Then the Q-function is an expectation with respect to the posterior density. Thus, a consequence of MCEM is that the final Monte Carlo sample that is used to approximate the Q-function is a sample from the posterior required for EB inference. We shall use the block Gibbs sampler, introduced by Hobert and Geyer (1998), to sample (approximately) from the posterior.

3.3.2. A numerical example

Littell *et al.* (1996), page 141, gave data arising from an experiment in which six randomly chosen influents for the Mississippi River are used to monitor the nitrogen concentration in parts per million.

The EB estimate of μ_0 is $\bar{y}$ and of the remaining parameters we choose to estimate only λ_1 and λ_2. One reason for this choice is that the EB estimate of ν_0 is ∞, i.e. the EB estimate of the prior on μ is a point mass at the overall mean. Therefore, we fix ν_0 at an *a priori* specified value; in this example we chose 0.1. Furthermore, maximizing the posterior with respect to all of a_1, λ_1 and a_2 and λ_2 is unstable, since it is flat in the direction of fixed means, a_1/λ_1 and a_2/λ_2. Therefore, we chose to fix the prior variances, a_1/λ_1^2 and a_2/λ_2^2, at 0.1, which leaves only λ_1 and λ_2 identifiable.

For starting values we used a method-of-moments approach. Specifically, we set the prior expectations for ν_θ and ν_e equal to the obvious values

$$E(\nu_\theta) = \frac{a_1}{\lambda_1} = \frac{n}{\mathrm{MSTR} - \mathrm{MSE}} = 0.108,$$

$$E(\nu_e) = \frac{a_2}{\lambda_2} = \frac{1}{\mathrm{MSE}} = 0.0235$$

and solved for a_1, λ_1, a_2 and λ_2. This yields $a_1 = 1.17$, $\lambda_1 = 1.08$, $a_2 = 0.006$ and $\lambda_2 = 0.235$. We shall retain the values for a_1 and a_2 and use the estimates of λ_1 and λ_2 as starting values in our MCEM program, i.e. $\tilde{\lambda}^{(0)} = (1.08, 0.235)^{\mathrm{T}}$.

For the MCEM algorithm we set $k = 2$, $\alpha = 0.3$ and $\beta = 0.25$. In each MCEM step the starting value for the block Gibbs sampler was $(\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_6, \bar{y})^{\mathrm{T}}$. The use of RS in this setting was described in Jones and Hobert (2001). We set $\gamma = 0.05$ and declared convergence when expression (13) fell below 10^{-5}. The computation terminated after eight MCEM iterations. The EB estimates are $\tilde{\lambda}^{(8)} = (59.618, 0.258)^{\mathrm{T}}$. Table 4 reports the number of regenerations and the average number of iterations of the block Gibbs sampler per regeneration for each MCEM step. The initial number of regenerations was specified to be 5.

3.4. An application to model-based spatial statistics

Fig. 2 depicts counts of automobile thefts and larcenies for a grid of city streets in south-east Baltimore over a period of 70 days. Each individual circle in Fig. 2 represents an automobile theft or a larceny at or near that intersection. Although it is reasonable to model these counts as realizations of Poisson random variables, assuming independence for counts in geographical proximity is not. Thus, we propose a generalized linear mixed model that uses the random effects to account for the spatial correlation (Diggle *et al.*, 1998). Note that fitting this model involves approximating an intractable integral whose dimension is the total number of responses.

Table 4. Regenerations and average number of iterations

MCEM iteration	*Number of regenerations*	*Average Gibbs iterations per regeneration*
1	8	45.0
2	8	4.50
3	93	6.01
4	473	5.67
5	5394	5.85
6	27309	5.76
7	6.9991×10^5	5.74
8	2.3622×10^6	5.73

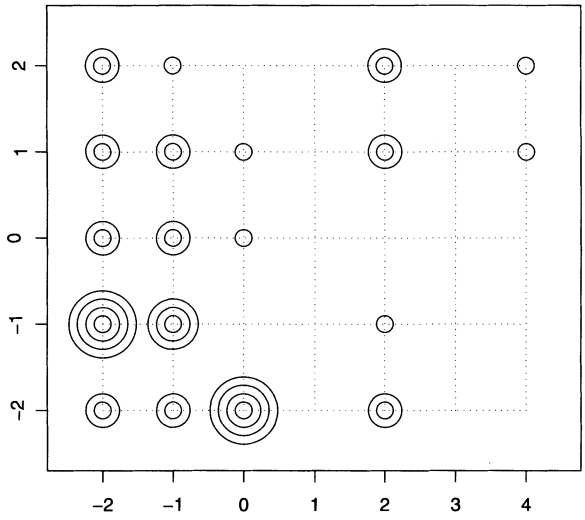

Fig. 2. Car thefts and larcenies by intersection (source, `http://www.ci.baltimore.md.us/government/police/`): ○, one theft or larceny at the intersection

Let x_i, $i = 1, \ldots, n$, be bivariate data points representing the locations of the 35 observed intersections, and let y_i denote the number of automobile thefts and larcenies at intersection x_i. Let $U = (U_1, \ldots, U_n)^{\mathrm{T}}$ be a vector of random effects. Conditional on u, we assume that y_i follows a Poisson density with mean μ_i satisfying

$$\log(\mu_i) = \lambda_1 + u_i. \tag{18}$$

Finally, we assume that U follows a multivariate normal distribution with mean 0 and variance–covariance matrix Σ, whose (i, j)-element is given by

$$\Sigma_{ij} = \lambda_2 \exp(-\lambda_3 \|x_i - x_j\|), \tag{19}$$

i.e. the correlation between any two random effects decays exponentially with the geographic distance between the associated observations.

Unlike applications of generalized linear mixed models with subject or strata level random effects, the marginal likelihood does not factor into the product of several smaller dimensional

integrals. Therefore, numerical integration is not an option, even for relatively small values of n. The same problem occurs with the integrals that are required for performing EM. However, fitting via the EM algorithm is appealing in this setting because considerable simplifications occur in the logarithm of the complete-data likelihood.

Unfortunately, direct simulation from $f_{U|Y}(u|y;\lambda^{(t-1)})$ is not possible. We shall investigate the use of two different sampling mechanisms; the Esup rejection sampler that was proposed by Caffo *et al.* (2002) and importance sampling, both using a candidate density that was obtained by shifting and scaling Student's t-density by the Laplace approximation to the mean and standard deviation of the distribution of $U|y;\lambda^{(t-1)}$. The formulae for these approximations are given in Appendix A.

Fig. 3 shows the path plots of the parameter estimates and the Monte Carlo sample sizes for the two versions of our algorithm. We set $\alpha = 0.15$, $\beta = 0.3$ and $k = 2$. The starting values $\tilde{\lambda}^{(0)} = (-0.3, 0.92, 1)^{\mathrm{T}}$ were obtained as the posterior modes from the R (Ihaka and Gentleman, 1996) contributed software package geoRglm (Christensen and Ribeiro, 2002) for a fixed value of $\lambda_3 = 1$. The algorithm converged in 56 MCEM steps for rejection sampling and 44 MCEM steps for importance sampling. We set $\gamma = 0.05$ and declared convergence when expression (13) fell below 10^{-4}. The final estimates were $\tilde{\lambda}^{(44)} = (-0.087, 0.776, 4.00)^{\mathrm{T}}$ for importance sampling and $\tilde{\lambda}^{(56)} = (-0.082, 0.775, 3.99)^{\mathrm{T}}$ for rejection sampling. The computing time for both algorithms was similar.

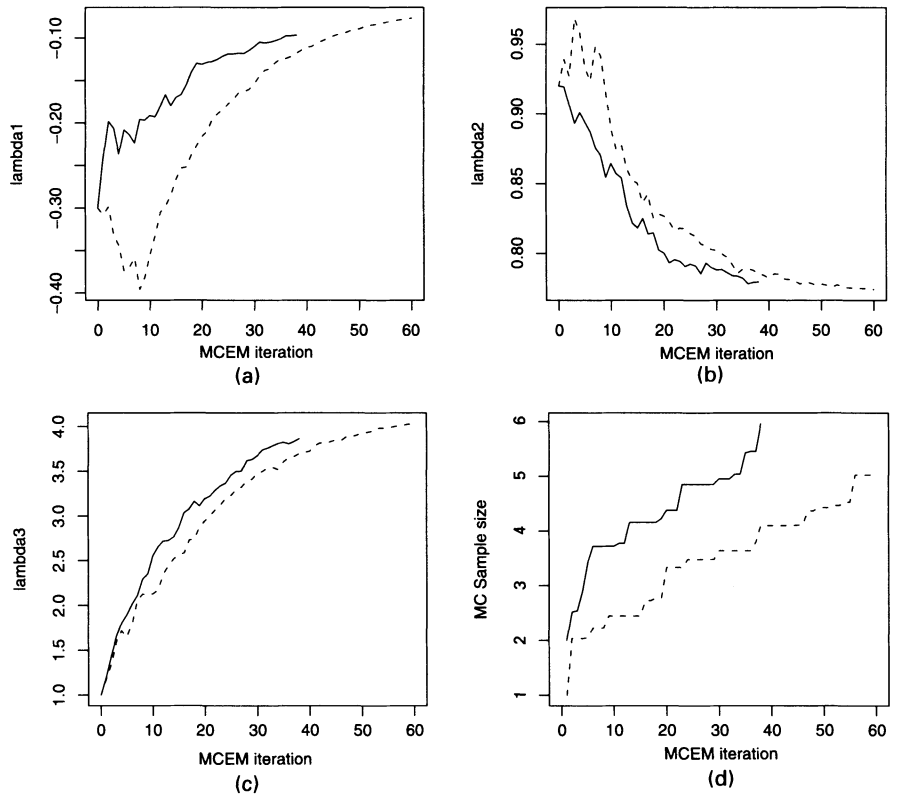

Fig. 3. Convergence of parameter estimates for MCEM applied to a spatial application of generalized linear mixed models (———, importance sampling; -------, rejection sampling): (a) convergence of λ_1; (b) convergence of λ_2; (c) convergence of λ_3; (d) common logarithm of the ending Monte Carlo sample sizes

Acknowledgements

The authors are grateful to Jim Booth, Charlie Geyer, Jim Hobert and Tom Louis for many helpful comments and suggestions. The authors also thank the Associate Editor and two referees for their many constructive comments.

Appendix A: Laplace approximation for Section 3.4

The Laplace approximation μ^* to $E(U|y;\tilde{\lambda}^{(t-1)})$ is the solution to

$$\frac{\partial}{\partial u}\log\{f_{Y,U}(y,u;\lambda^{(t-1)})\} = y - \mu - \Sigma^{-1}u = 0$$

where $\mu = \exp(\lambda_1^{(t-1)} + u)$ and Σ is the variance–covariance matrix that is defined by equation (19) evaluated at $\lambda_2^{(t-1)}$ and $\lambda_3^{(t-1)}$. The Laplace approximation to $\mathrm{var}(U|y;\tilde{\lambda}^{(t-1)})$ is

$$\left[\frac{\partial^2}{\partial u\,\partial u^{\mathrm{T}}}\log\{f_{Y,U}(y,u;\lambda^{(t-1)})\}\right]^{-1} = \{\mathrm{diag}(\mu^*) + \Sigma^{-1}\}^{-1}$$

where $\mathrm{diag}(\mu^*)$ is a diagonal matrix with μ^* along the main diagonal.

References

Billingsley, P. (1995) *Probability and Measure*, 3rd edn. New York: Wiley.
Booth, J. G. and Hobert, J. P. (1999) Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. *J. R. Statist. Soc.* B, **61**, 265–285.
Booth, J. G., Hobert, J. P. and Jank, W. S. (2001) A survey of Monte Carlo algorithms for maximizing the likelihood of a two-stage hierarchical model. *Statist. Modllng*, **1**, 333–349.
Caffo, B. S., Booth, J. G. and Davison, A. C. (2002) Empirical sup rejection sampling. *Biometrika*, **89**, 745–754
Christensen, O. F. and Ribeiro, Jr, P. J. (2002) geoRglm—a package for generalised linear spatial models. *R News*, **2**, 26–28.
Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc.* B, **39**, 1–22.
Diggle, P. J., Tawn, J. A. and Moyeed, R. A. (1998) Model-based geostatistics. *Appl. Statist.*, **47**, 299–326.
Geyer, C. J. (1994) On the convergence of Monte Carlo maximum likelihood calculations. *J. R. Statist. Soc.* B, **56**, 261–274.
Gueorguieva, R. and Agresti, A. (2001) A correlated probit model for joint modeling of clustered binary and continuous responses. *J. Am. Statist. Ass.*, **96**, 1102–1112.
Hobert, J. P. and Geyer, C. J. (1998) Geometric ergodicity of Gibbs and block Gibbs samplers for a hierarchical random effects model. *J. Multiv. Anal.*, **67**, 414–430.
Hobert, J. P., Jones, G. L., Presnell, B. and Rosenthal, J. S. (2002) On the applicability of regenerative simulation in Markov chain Monte Carlo. *Biometrika*, **89**, 731–743.
Ihaka, R. and Gentleman R. (1996) R: a language for data analysis and graphics. *J. Comput. Graph. Statist.*, **5**, 299–314.
Jank, W. and Booth J. G. (2003) Efficiency of Monte Carlo EM and simulated maximum likelihood in two-stage hierarchical models. *J. Comput. Graph. Statist.*, **12**, 214–229.
Jones, G. L., Haran, M. and Caffo, B. S. (2004) Output analysis for Markov chain Monte Carlo simulations. *Technical Report*. School of Statistics, University of Minnesota, Minneapolis.
Jones, G. L. and Hobert, J. P. (2001) Honest exploration of intractable probability distributions via Markov chain Monte Carlo. *Statist. Sci.*, **16**, 312–334.
Kendall, M. G. and Stuart, A. (1958) *The Advanced Theory of Statistics*, vol. 1. London: Griffin.
Lange, K. (1999) *Numerical Analysis for Statisticians*. New York: Springer.
Levine, R. A. and Casella, G. (2001) Implementations of the Monte Carlo EM algorithm. *J. Comput. Graph. Statist.*, **10**, 422–439.
Littell, R. C., Milliken, G. A., Stroup, W. W. and Wolfinger, R. D. (1996) *SAS System for Mixed Models*. Cary: SAS Institute.
Louis, T. A. (1982) Finding the observed information matrix when using the EM algorithm. *J. R. Statist. Soc.* B, **44**, 226–233.
McCulloch, C. E. (1994) Maximum likelihood variance components estimation for binary data. *J. Am. Statist. Ass.*, **89**, 330–335.
McCulloch, C. E. (1997) Maximum likelihood algorithms for generalized linear mixed models. *J. Am. Statist. Ass.*, **92**, 162–170.

Meyn, S. P. and Tweedie, R. L. (1993) *Markov Chains and Stochastic Stability*. London: Springer.
Mykland, P., Tierney, L. and Yu, B. (1995) Regeneration in Markov chain samplers. *J. Am. Statist. Ass.*, **90**, 233–241.
Nummelin, E. (1984) *General Irreducible Markov Chains and Non-negative Operators*. London: Cambridge University Press.
Polyak, B. T. and Juditsky, A. B. (1992) Acceleration of stochastic-approximation by averaging. *SIAM J. Control Optimzn*, **30**, 838–855.
Quintana, F. A., Liu, J. S. and del Pino, G. E. (1999) Monte Carlo EM with importance reweighting and its applications in random effects models. *Comput. Statist. Data Anal.*, **29**, 429–444.
Robert, C. P. and Casella, G. (1999) *Monte Carlo Statistical Methods*. New York: Springer.
Searle, S. R., Casella, G. and McCulloch, C. E. (1992) *Variance Components*. New York: Wiley.
Shi, J. Q. and Copas, J. (2002) Publication bias and meta-analysis for 2×2 tables: an average Markov chain Monte Carlo EM algorithm. *J. R. Statist. Soc.* B, **64**, 221–236.
Wei, G. C. G. and Tanner, M. A. (1990) A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *J. Am. Statist. Ass.*, **85**, 699–704.
Wu, C. F. J. (1983) On the convergence properties of the EM algorithm. *Ann. Statist.*, **11**, 95–103.