Ascent-Based Monte Carlo EM

Brian S. Caffo*, Wolfgang Jank, Galin L. Jones

Department of Biostatistics

Johns Hopkins University

and

R.H. Smith School of Business

University of Maryland

and

School of Statistics

University of Minnesota

April 8, 2004

_____

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

Abstract

The 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 proposed algorithm improves on similar existing methods by: (i) recovering EM's ascent (i.e., likelihood-increasing) property with high probability, (ii) being more robust to the impact of user defined inputs, and (iii) handling classical Monte Carlo and Markov chain Monte Carlo within a common framework. Because of (i) 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 dramatically accelerate the convergence of deterministic EM.

Key words and phrases. EM algorithm, Empirical Bayes, Generalized Linear Mixed Models, Importance Sampling, Markov chain, Monte Carlo

1 INTRODUCTION

Since the seminal article of Dempster et al. (1977), the 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 while the M-step maximizes this expectation. Often, at least one of these steps is analytically intractable. Many authors have suggested that a troublesome E-step may be overcome by approximating the expectation with Monte Carlo methods (see, e.g, Booth and Hobert, 1999; McCulloch, 1994, 1997; Shi and Copas, 2002; Wei and Tanner, 1990). This is the Monte Carlo EM (MCEM) algorithm.

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 behavior 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) and study a data-driven automated MCEM algorithm that seeks to overcome these difficulties.

Let Y denote a vector of observed data, 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

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

Instead of directly maximizing (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

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

while in the tth M-step we require a value, $\lambda^{(t)}$, that satisfies $Q(\lambda^{(t)}, \lambda^{(t-1)}) \geq Q(\lambda, \lambda^{(t-1)})$ 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

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

which yields a *Generalized EM* (GEM) algorithm. The *ascent property* is obtained with an application of Jensen's inequality to (3), that is,

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

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

When the integral in (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 used in this paper.) Let $\tilde{\lambda}^{(t-1)}$ denote the current MCEM estimate of $\hat{\lambda}$. Throughout we assume that $\{u^{(t,1)}, \dots, u^{(t,m_t)}\}$ is either (i) a random sample from $f_{U|Y}(u|y, \tilde{\lambda}^{(t-1)})$; (ii) a sample obtained from a candidate $h(u)$ with a set of associated importance weights $\{w(u^{(t,j)})\}$; or (iii) 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 (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 under (i) and (iii). The MCEM algorithm uses $\tilde{Q}$ in place of Q; that is, the tth M-step consists of finding a value $\tilde{\lambda}^{(t)}$ such that

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

If the Monte Carlo sample size is constant across MCEM steps, i.e., $m_t = m$ for all t, MCEM will not converge due to 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 (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 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 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 using standard methods; a clear requirement for the algorithm to be useful. This is in contrast to 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.

Example 1. Consider the following conditionally independent model: Suppose $Y_i|u_i \sim \text{Normal}(u_i, 1)$ and $U_i \sim \text{Normal}(0, \lambda)$ for $i = 1, \ldots n$. Table 1 displays data simulated according to this model with $n = 5$ and $\lambda = 1$. The mle is $\hat{\lambda} = 1.3183$. Consider the bottom two plots in Figure 1 (We will return to this figure later.) which contain plots of the marginal log-likelihood paths for EM and 15 replications of Ascent-based MCEM under two sampling schemes: In the left-hand plot, we used rejection sampling to draw a random sample from $f_{U|Y}(u|y; \tilde{\lambda}^{(t-1)})$ while, in the right-hand plot, we used a Metropolis-Hastings independence sampler having invariant density $f_{U|Y}(u|y; \tilde{\lambda}^{(t-1)})$ and a $\text{Normal}(0, \tilde{\lambda}^{(t-1)})$ candidate density. We started all three algorithms at 1, i.e., $\lambda^{(0)} = \tilde{\lambda}^{(0)} = 1$. Figure 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 article 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 EM

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 will check (4) by checking (3). That is, the algorithm requires evidence that

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

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 (6) consistently with

$$\Delta \tilde{Q}(\tilde{\lambda}^{(t,m_t)}, \tilde{\lambda}^{(t-1)}) \equiv \tilde{Q}(\tilde{\lambda}^{(t,m_t)}, \tilde{\lambda}^{(t-1)}) - \tilde{Q}(\tilde{\lambda}^{(t-1)}, \tilde{\lambda}^{(t-1)})$$
$$= \frac{\sum_{j=1}^{m_t} w(u^{(t,j)}) \log \frac{f_{YU}(y, u^{(t,j)}; \tilde{\lambda}^{(t,m_t)})}{f_{YU}(y, u^{(t,j)}; \tilde{\lambda}^{(t-1)})}}{\sum_{j=1}^{m_t} w(u^{(t,j)})} , \qquad (7)$$

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

We will now argue that, when suitably normalized,

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

has a limiting normal distribution with mean zero and a variance, σ^2, that depends on the sampling mechanism employed. For the sake of simplicity we will assume that the sampling mechanism produces iid samples from $f_{U|Y}(u|y, \tilde{\lambda}^{(t-1)})$. However, all of these arguments will go through more generally. For example, if MCMC 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, p. 272) prove the asymptotic normality of $\sqrt{m_t}(\tilde{\lambda}^{(t)} - \lambda^{(t)})$ and consider

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

Note that (9) is asymptotically $N(0, \sigma^2)$ and standard Taylor series arguments show that (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 (8). (Calculation of ASE is deferred until subsection 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, that is, $\tilde{\lambda}^{(t)} = \tilde{\lambda}^{(t,m_t)}$, 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 accurately estimate Q. 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 (12), we also have that

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

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 (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; that is, 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 on 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 to (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 in order to guard against prematurely claiming convergence. We

believe this approach is understandable but somewhat misguided since it places the emphasis on the behavior of the parameter estimates with little or no regard to the estimation of the information. We address this issue in the context of a benchmark 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

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

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

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

This is a standard formula for the variance of the ratio of two means (Kendall and Stuart, 1958, p. 232) applied to the importance sampling ratio estimator . With rejection sampling the importance weights are set to 1 and (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

Calculating a reasonable Monte Carlo standard error is more difficult when we are forced to employ MCMC. 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 will give only a sketch of our implementation of RS as the details have appeared elsewhere; see e.g., Hobert et al. (2002); Jones et al. (2004); Jones and Hobert (2001); and Mykland et al. (1995). The basic idea is that one simulates 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 $0 = \tau_0 < \tau_1 < \tau_2 <$

$\cdots$ 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

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

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

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

Since $\lambda^{(t)}$ is unknown, we estimate the 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 (5) based on R_t regenerations of the chain. Therefore, an asymptotic lower bound for (7) 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 in order 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; that is, not necessarily Ascent-based MCEM. The probability that the ascent property does *not* hold between two consecutive iterations is $\Pr\{Q(\tilde{\lambda}^{(t)}, \tilde{\lambda}^{(t-1)}) \leq Q(\tilde{\lambda}^{(t-1)}, \tilde{\lambda}^{(t-1)})\}$. Note that this probability does not correspond to α from (12) and if $m_t \to \infty$ then $\Pr\{Q(\tilde{\lambda}^{(t)}, \tilde{\lambda}^{(t-1)}) \leq Q(\tilde{\lambda}^{(t-1)}, \tilde{\lambda}^{(t-1)})\} \to 0$. 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 likely holds.

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

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

Then Borel-Cantelli (Billingsley, 1995, pp 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 GEM 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 less of a concern than confidence in the output of the algorithm. On a related note, it is possible to use an importance weighting scheme and use samples from all of the previous simulations as in Booth and Hobert (1999) and Quintana et al. (1999). However, authors such as Booth and Hobert (1999) find little advantage in this approach and in our experience with Ascent-based MCEM it often requires an enormous amount of storage and hence negatively impacts 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 $m_{t,start}$ be the starting Monte Carlo sample size and $m_{t,end}$ be the ending Monte Carlo sample size. In order to force an increase (on average) in the Monte Carlo sample sizes across MCEM iterations we take $m_{t+1,start} \geq m_{t,start}$. If we assume that

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

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

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

where β is a specified type 2 error rate and $\hat{\sigma}^2$ is an estimate of the variance of $\Delta \tilde{Q}$ 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 (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 2 levels of α and both sampling mechanisms. The results are given in Figure 1. The plots indicate that,

10

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 (12) we have concluded otherwise. Hence unnecessary additional simulation will be performed within the Monte Carlo E-step. Thus, β will impact the total simulation effort.

3 APPLICATIONS

3.1 A Benchmark Example

Let $y_{ij}|u_i \sim$ Bernoulli (π_{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 authors including Booth and Hobert (1999), Levine and Casella (2001), McCulloch (1997) and Quintana et al. (1999). In particular, Booth and Hobert (1999) report data that was simulated according to this model with $\lambda_1 = 5$ and $\lambda_2 = .5$. 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 $\text{Var}(\hat{\lambda}_1)$, $\text{Var}(\hat{\lambda}_2)$ and $\text{Cov}(\hat{\lambda}_1, \hat{\lambda}_2)$, 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 will evaluate the performance of both algorithms when implementing the standard practice of approximating the inverse observed information matrix using the Monte Carlo sample from the final MCEM iteration via Louis'(1982) method.

In BH-MCEM a $100(1 - \alpha_1)\%$ confidence ellipsoid, centered 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, BH-MCEM does not reject the current estimate of the mle which, in our experience, means BH-MCEM tends to produce a much more volatile sequence of estimates than Ascent-based MCEM.

We performed 10,000 independent replications each of BH-MCEM and Ascent-based MCEM. 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 $C \geq 1$ consecutive MCEM steps where $C = 1$ for Ascent-based MCEM and $C = 1, 2, 3, 4$ for BH-MCEM. With Ascent-based MCEM we used $\alpha = 0.25$, $\beta = .25$ and set $k = 3$ while for BH-MCEM 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, percent 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 10,000 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 BH-MCEM 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 BH-MCEM with $C = 2$ has roughly the same relative error as Ascent-based MCEM but with less simulation effort. However, BH-MCEM with either $C = 1$ or 2 results in substantially larger average relative errors in terms of estimating the inverse information. On the other hand, if $C = 3$ then, in terms of mean and median relative error, Ascent-based MCEM and BH-MCEM estimate 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 BH-MCEM than those for Ascent-based MCEM. This is also the case for BH-MCEM with $C = 4$. Thus, the volatility of the estimates produced by BH-MCEM is apparent in spite of the fact that BH-MCEM $(C = 3, 4)$ 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, Ascent-based MCEM compares favorably to BH-MCEM.

3.2 A Hybrid EM/MCEM Algorithm

Figure 1 indicates that, especially in its early stages, Ascent-based MCEM can result in step sizes 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 will use such an algorithm to obtain 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 on the order of ten to twenty thousand in microarray experiments). Let x be a $J \times p$ matrix of covariates (e.g., tissue type and subject age). Suppose that conditional on u_{1i}, u_{2i} the data, Y_i, are independent with

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

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

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

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 while 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 (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 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 get too large, the algorithm then switches to EM.

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

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

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

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

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 $U_{1i}|u_{2i}, y_i, \tilde{\lambda}^{(t-1)}$. Formulas for these quantities are given in Table 3. The sequential sampling mechanism for generating from $f_{U|Y}$ follows:

1. simulate $u_3^{(t,j)}$ uniformly on the integers $1, \ldots, n$,

2. simulate $u_2^{(t,j)}$ as Gamma$\{shape(u_3^{(t,j)}), rate(u_3^{(t,j)})\}$,

3. simulate $u_1^{(t,j)}$ as Normal$\{mean(u_3^{(t,j)}, u_2^{(t,j)}), var(u_3^{(t,j)}, u_2^{(t,j)})\}$.

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

The algorithm proceeds by using our MCEM algorithm in the early stages and then in the later stages EM based on (16). A rough estimate of the computational effort, obtained by inspecting the program code, suggests that at least $2m_t$ nontrivial computations are required for one MCEM iteration. In contrast, one EM iteration needs n nontrivial 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, notice 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 $n = 20,000$, $n = 60,000$ and $n = 100,000$ with λ_1, λ_2 and x (20×5) obtained from a specific microarray experiment. The simulated data are available upon request.

We set $\tilde{\lambda}_1^{(0)}$ equal to the identity matrix, $\tilde{\lambda}_2^{(0)} = \tilde{\lambda}_3^{(0)} = 1$, $\alpha = 0.3$, $\beta = 0.05$, $k = 2$ and performed 100 independent runs of the hybrid algorithm. This was repeated for each of the 3 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 $n = 20,000$, it took EM 6 minutes and 17 seconds (6:17) to converge. On the other hand, the maximum run time for the hybrid algorithm was 4:56; an improvement of 22%. The minimum, 25th, 50th and 75th percentiles of the run times for the hybrid algorithm are 1:29, 3:24, 3:47, and 4:13 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 $n = 60,000$ the five number summary for the run times for the hybrid algorithm was (3:25, 8:55, 9:44, 11:00, 13:23) while EM took 18:60. That is, the improvement ranged from 30% to 82%. When $n = 100,000$ the five number summary of the run times for the hybrid algorithm was (5:35, 13:21, 15:27, 16:42, 21:20) and EM required 31:46. 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 $\theta = (\theta_1, \ldots, \theta_A)^T$ and ν_e, the data, Y_{ij}, are independent with

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

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

$$\theta_i|\mu, \nu_\theta \sim \text{Normal}(\mu, \nu_\theta^{-1}) \quad \text{and} \quad \nu_e \sim \text{Gamma}(a_2, \lambda_2).$$

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

$$\mu \sim \text{Normal}(\mu_0, \nu_0^{-1}) \quad \text{and} \quad \nu_\theta \sim \text{Gamma}(a_1, \lambda_1)$$

where $\mu_0, \nu_0, a_1, \lambda_1, a_2, \lambda_2$ are constants and all but μ_0 are assumed to be strictly positive; hence all of the priors are proper. Regardless of the prior specification on the variance components, the EB estimate of μ_0 is the overall mean, $\bar{y} = \sum_{ij} y_{ij}/n$ where $n = \sum_i n_i$. Also, for reasons discussed in Subsection 3.3.2, we focus on estimating λ_1 and λ_2 for fixed a_1, a_2 and ν_0.

Note that, in terms of the EM notation, we have $U = (\theta, \mu, \nu_\theta, \nu_e)^T$, and $\lambda = (\lambda_1, \lambda_2)^T$. 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 used to approximate the Q-function is a sample from

the posterior required for EB inference. We will 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, p.141) give 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 ∞. That is, 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, 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

$$E(\nu_\theta) = \frac{a_1}{\lambda_1} = \frac{n}{\text{MSTR} - \text{MSE}} = 0.108 \qquad \text{and} \qquad E(\nu_e) = \frac{a_2}{\lambda_2} = \frac{1}{\text{MSE}} = 0.0235$$

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 will 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, that is, $\tilde{\lambda}^{(0)} = (1.08, 0.235)^T$.

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 $(\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_6, \bar{y})^T$. The use of RS in this setting was described in Jones and Hobert (2001). We set $\gamma = 0.05$ and declared convergence when (13) fell below 10^{-5}. The computation terminated after 8 MCEM iterations. The EB estimates are $\tilde{\lambda}^{(8)} = (59.618, 0.258)^T$. 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

Figure 2 depicts counts of auto thefts and larcenies for a grid of city streets in southeast Baltimore over a period of 70 days. Each individual circle in the figure represents an auto theft or a larceny at or near that intersection. While it is reasonable to model these counts as realizations of Poisson

random variables, assuming independence for counts in close geographical proximity is not. Thus, we propose a generalized linear mixed model (GLMM) 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.

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 auto thefts and larcenies at intersection x_i. Let $U = (U_1, \ldots, U_n)^T$ 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 zero and variance-covariance matrix, Σ, whose i, j element is given by

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

That is, the correlation between any two random effects decays exponentially with the geographic distance between the associated observations.

Unlike applications of GLMMs 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 required for performing EM. However, fitting via the EM algorithm is appealing in this setting because considerable simplifications occur in the log of the complete data likelihood.

Unfortunately, direct simulation from $f_{U|Y}(u|y; \lambda^{(t-1)})$ is not possible. We will investigate the use of two different sampling mechanisms; the Esup rejection sampler proposed by Caffo et al. (2002) and importance sampling both using a candidate density 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 formulas for these approximations are given in Appendix A.

Figure 3 shows the path plots of the parameter estimates and the Monte Carlo sample sizes for the two versions of our algorithm. We set $\alpha = .15$, $\beta = .3$ and $k = 2$. The starting values $\tilde{\lambda}^{(0)} = (-.3, .92, 1)^T$ were obtained as the posterior modes from the R (Ihaka and Gentleman, 1996) contributed software package geoRglm (Christensen and Ribeiro Jr, 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 (13) fell below

10^{-4}. The final estimates were $\tilde{\lambda}^{(44)} = (-.087, .776, 4.00)^T$ for importance sampling and $\tilde{\lambda}^{(56)} = (-.082, .775, 3.99)^T$ for rejection sampling. The computing time for both algorithms was similar.

A   Laplace Approximation for Section 3.4

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

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

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

$$\left\{ \frac{\partial^2}{\partial u \partial u^T} \log f_{YU}(y, u; \lambda^{(t-1)}) \right\}^{-1} = \left\{ \text{Diag}(\mu^*) + \Sigma^{-1} \right\}^{-1}$$

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

References

Billingsley, P. (1995). *Probability and Measure*. Wiley, New York, third edition.

Booth, J. G. and Hobert, J. P. (1999). Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. *Journal of the Royal Statistical Society,* Series 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. *Statistical Modelling*, 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. *Journal of the Royal Statistical Society,* Series B, 39:1–22.

Diggle, P. J., Tawn, J. A., and Moyeed, R. A. (1998). Model-based geostatistics (with discussion). *Applied Statistics*, 47:299–326.

Geyer, C. J. (1994). On the convergence of Monte Carlo maximum likelihood calculations. *Journal of the Royal Statistical Society,* Series B, 56:261–274.

Gueorguieva, R. and Agresti, A. (2001). A correlated probit model for joint modeling of clustered binary and continuous responses. *Journal of the American Statistical Association*, 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. *Journal of Multivariate Analysis*, 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. *Journal of Computational and Graphical Statistics*, 5:299–314.

Jank, W. and Booth, J. G. (2003). Efficiency of Monte Carlo EM and simulated maximum likelihood in two-stage hierarchical models. *Journal of Computational and Graphical Statistics*, 12:214–229.

Jones, G. L., Haran, M., and Caffo, B. S. (2004). Output analysis for Markov chain Monte Carlo simulations. Technical report, University of Minnesota, School of Statistics.

Jones, G. L. and Hobert, J. P. (2001). Honest exploration of intractable probability distributions via Markov chain Monte Carlo. *Statistical Science*, 16:312–334.

Kendall, M. G. and Stuart, A. (1958). *The Advanced Theory of Statistics*, volume 1. Charles Griffen & Company Limited, London.

Lange, K. (1999). *Numerical Analysis for Statisticians.* Springer–Verlag, New York.

Levine, R. A. and Casella, G. (2001). Implementations of the Monte Carlo EM algorithm. *Journal of Computational and Graphical Statistics*, 10:422–439.

Littell, R. C., Milliken, G. A., Stroup, W. W., and Wolfinger, R. D. (1996). *SAS system for Mixed Models.* SAS Institute Inc., Cary, NC.

Louis, T. A. (1982). Finding the Observed Information Matrix when using the EM algorithm. *Journal of the Royal Statistical Society,* Series B, 44:226–233.

McCulloch, C. E. (1994). Maximum likelihood variance components estimation for binary data. *Journal of the American Statistical Association*, 89:330–335.

McCulloch, C. E. (1997). Maximum likelihood algorithms for generalized linear mixed models. *Journal of the American Statistical Association*, 92:162–170.

Meyn, S. P. and Tweedie, R. L. (1993). *Markov Chains and Stochastic Stability.* Springer-Verlag, London.

Mykland, P., Tierney, L., and Yu, B. (1995). Regeneration in Markov chain samplers. *Journal of the American Statistical Association*, 90:233–241.

Nummelin, E. (1984). *General Irreducible Markov Chains and Non-negative Operators.* Cambridge University Press, London.

Polyak, B. T. and Juditsky, A. B. (1992). Acceleration of stochastic–approximation by averaging. *SIAM Journal on Control and Optimization*, 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. *Computational Statistics and Data Analysis*, 29:429–444.

Robert, C. P. and Casella, G. (1999). *Monte Carlo Statistical Methods.* Springer, New York.

Searle, S. R., Casella, G., and McCulloch, C. E. (1992). *Variance Components.* John Wiley and Sons, New York.

Shi, J. Q. and Copas, J. (2002). Publication bias and meta-analysis for 2×2 tables: an average Markov chain Monte Carlo EM algorithm. *Journal of the Royal Statistical Society,* Series 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. *Journal of the American Statistical Association*, 85:699–704.

Wu, C. F. J. (1983). On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11:95–103.

Table 1: Simulated data for Example 1.

y_i	0.3364675	-2.6338934	0.9080410	1.8897579	-0.3811235

Table 2: Comparison of Ascent-based MCEM with BH-MCEM. TotSim is the total number of simulated vectors for a single replication of MCEM while PerSim is the percentage of the total simulation effort 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 and the median of the 10,000 replications.

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	Mean	2009	55	.0189	.0866	.3458	.5000	.7234
$(C = 1)$	SE	34	0.2	.0001	.0006	.0318	.0390	.0653
	Med	1131	55	.0162	.0761	.1390	.2152	.2649
BH	Mean	292	18	.0257	.1067	2.4000	2.5186	4.1203
$(C = 1)$	SE	2	0.4	.0002	.0008	.4200	.3002	.5863
	Med	216	18	.0198	.0860	.3646	.5168	.7298
BH	Mean	1277	18	.0120	.0549	.9825	1.3377	2.0687
$(C = 2)$	SE	11	0.1	.0001	.0005	.0871	.1148	.1845
	Med	960	18	.0090	.0403	.2099	.2909	.4296
BH	Mean	2970	19	.0076	.0344	.3494	.4969	.7528
$(C = 3)$	SE	23	0.1	.0001	.0003	.0313	.0548	.0781
	Med	2419	19	.0057	.0257	.1337	.1872	.2766
BH	Mean	5113	19	.0054	.0248	.2823	.3584	.5798
$(C = 4)$	SE	37	0.1	.0001	.0002	.0636	.0632	.1206
	Med	4270	19	.0042	.0191	.1029	.1409	.2112

Table 3: Posterior Parameters for the Model from Section 3.2

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

Table 4: Regenerations and Average Number of Iterations

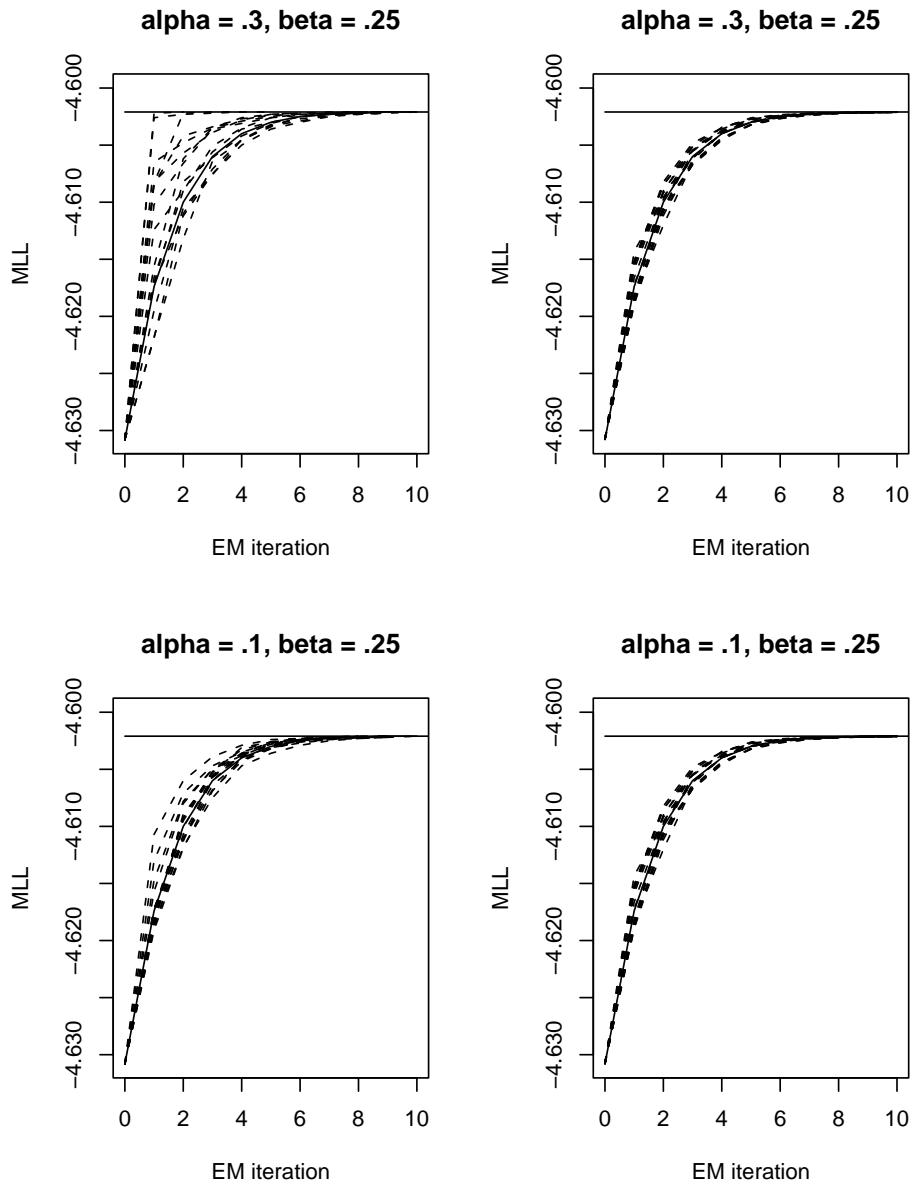| MCEM Iteration	Number of Regenerations	Average Gibbs Iterations per Regeneration
1	8	45.0
2	8	4.50
3	93	6.01
4	473	5.67
5	5394	5.85
6	27309	5.76
7	6.9991×10^5	5.74
8	2.3622×10^6	5.73

Figure 1: Marginal log-likelihood path plots for Ascent-based MCEM applied to Gaussian data for two values of α. The left-hand plots correspond to using rejection sampling while the right-hand plots are based on an MCMC independence sampler. The solid line is the maximum log-likelihood and the solid curve is the marginal log-likelihood path of deterministic EM.
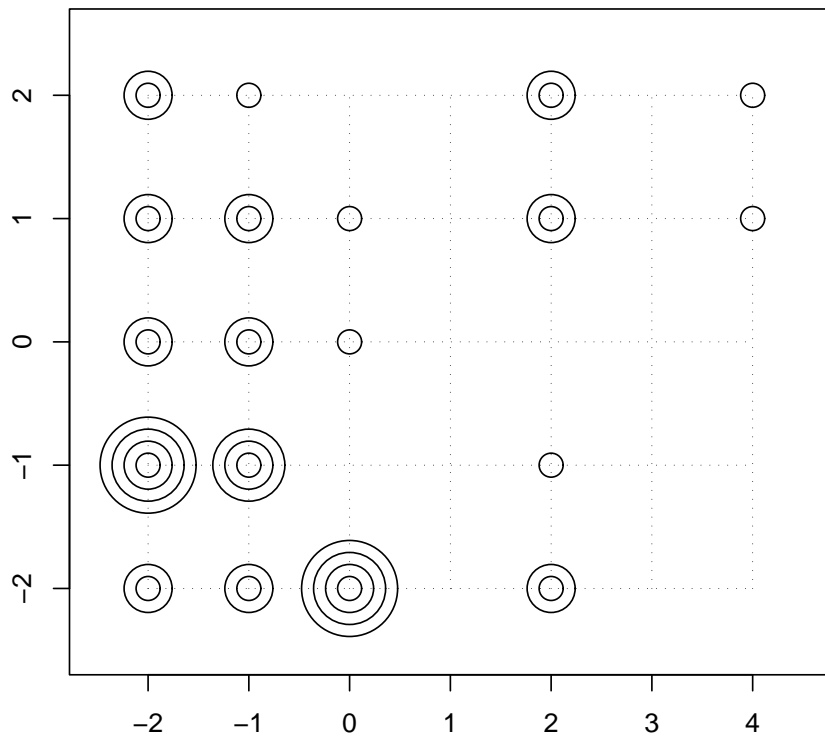
Figure 2: Car thefts and larcenies by intersection. Individual circles represent one theft or larceny at that intersection. Source: http://www.ci.baltimore.md.us/government/police/.
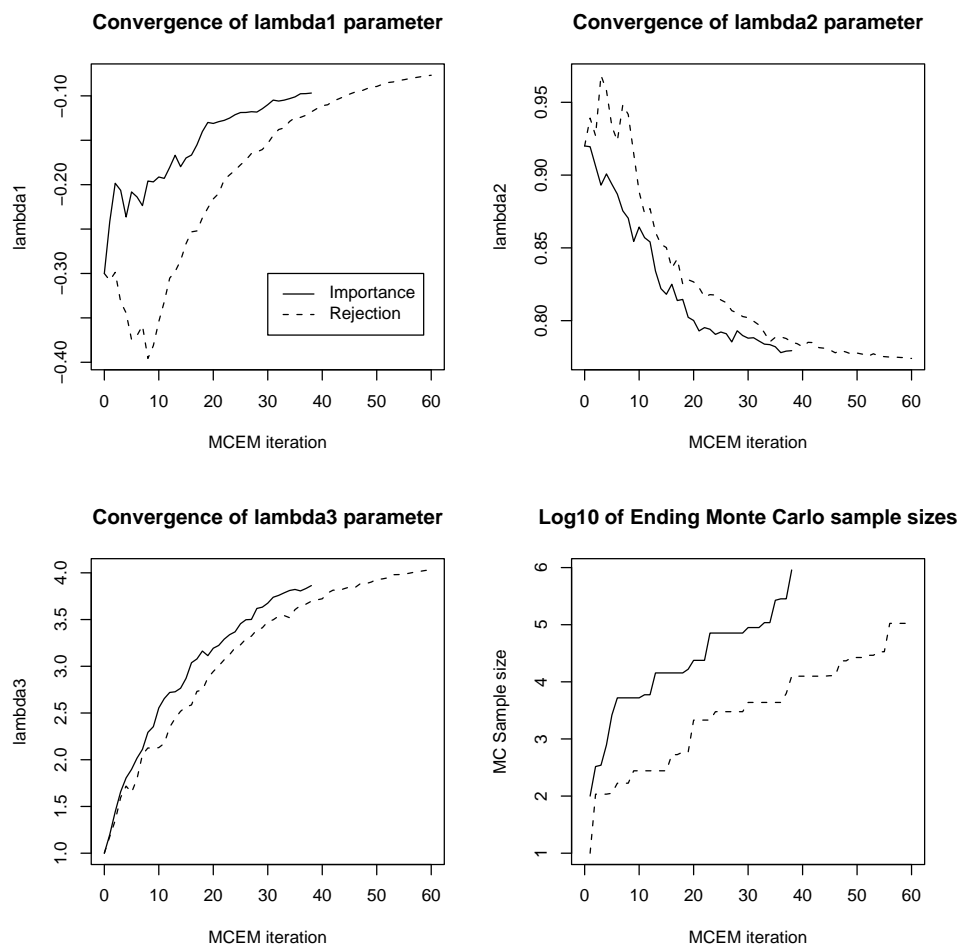
Figure 3: Convergence of parameter estimates for MCEM applied to a spatial application of generalized linear mixed models.