Monte Carlo Local Likelihood for Estimating
Generalized Linear Mixed Models

M. Jeon*
Graduate School of Education, University of California, Berkeley, CA 94720, USA
*email: mjj@berkeley.edu

and

C. Kaufman*
Department of Statistics, University of California, Berkeley, CA 94720, USA
*email: cgk@berkeley.edu

and

S. Rabe-Hesketh*
Graduate School of Education, University of California, Berkeley, CA 94720, USA
*email: sophiarh@berkeley.edu

SUMMARY: We propose the Monte Carlo local likelihood (MCLL) method for estimating generalized linear mixed models (GLMMs) with crossed random effects. MCLL initially treats model parameters as random variables, sampling them from the posterior distribution in a Bayesian model. The likelihood function is then approximated up to a constant by fitting a density to the posterior samples and dividing it by the prior. In the MCLL algorithm, the posterior density is approximated using local likelihood density estimation (Loader, 1996), in which the log-density is locally approximated by a polynomial function. In his Monte Carlo kernel likelihood (MCKL) method, De Valpine (2004) proposed a similar approach using kernel density estimation instead of local likelihood density estimation. We also develop a novel method to compute standard errors and the Bayes factor. Using empirical and simulation examples, we show that our proposed method outperforms existing methods such as the Laplace approximation and Monte Carlo maximum likelihood estimation (MCMLE).

KEY WORDS: Bayes factor; Crossed random effects; Generalized liner mixed model; Local likelihood density estimation

This paper has been submitted for consideration for publication in Biometrics
1. Introduction

Maximum likelihood estimation for generalized linear mixed models (GLMMs) is hindered by high dimensional intractable integrals involved in the likelihood function. The problem is magnified when the random effects have a crossed design and thus the data cannot be reduced to small independent clusters. For instance, the simplest logistic mixed model for a binary outcome $y_{ij}$ can be written as

$$\text{logit}(p(y_{ij} = 1|z_i, u_j)) = \mu + z_i + u_j,$$

where $\mu$ is a fixed intercept and $z_i$ and $u_j$ are independent random effects that are crossed with each other, with $z_i \sim N(0, \sigma^2)$, $i = 1, ..., m$ and $u_j \sim N(0, \tau^2)$, $j = 1, ..., n$. If all combinations of $i$ and $j$ exist in the data, this likelihood function involves $m+n$ dimensional integrals.

Various methods have been proposed for approximating the intractable likelihood function. For instance, the Laplace approximation (Tierney and Kadane, 1986; Lindstrom and Bates, 1988; Wolfinger, 1993) and adaptive quadrature (Naylor and Smith, 1982; Pinheiro and Bates, 1995; Schilling and Bock, 2005; Rabe-Hesketh et al., 2005) have been widely used. The Laplace approximation and the similar penalized quasi-likelihood approximation (PQL; Breslow and Clayton, 1993) are known to perform poorly for small cluster sizes and for large variance components (Breslow and Lin, 1995; Rodriguez and Goldman, 1995; Joe, 2008). Adaptive quadrature is more accurate but computationally more demanding than Gaussian quadrature (Pinheiro and Bates, 1995; Rabe-Hesketh et al., 2005). For a review of other approaches, see e.g., Cho and Rabe-Hesketh (2011).

Monte Carlo (MC) methods have also been utilized in various ways in ML estimation. Most methods are based on sampling the random effects given current parameter estimates. These methods can be distinguished by whether a ‘single sample’ or ‘many samples’ are used per evaluation of the objective function (for this distinction, see Geyer, 1996). The ‘single sample’
method is computationally more efficient than the ‘many samples’ method because it uses the same samples for all evaluations of the objective function. For instance, Geyer and Thompson (1992), Geyer (1994), and Sung and Geyer (2007) used MC simulations of the random effects for an importance sampling approximation of the likelihood (or the likelihood ratio). The efficiency of the ‘single sample’ method highly depends on the importance sampling distribution. If the initial guess of parameters is far from the true parameter values, this method can perform poorly (Geyer, 1994; McCulloch, 1997). Geyer (1996) suggested iterating the procedure so that the objective function is maximized around a true parameter region. However, this approach requires many MC samples for each iteration of the algorithm.

MC expectation maximization (MCEM) is an example of a ‘many sample’ method. Several MCEM algorithms have been proposed using various sampling methods: e.g., Metropolis-Hastings (McCulloch, 1997), an independent sampler based on importance sampling or rejection sampling (Booth and Hobert, 1999), and a slice sampler (Vaida and Meng, 2005). The basic idea is to use MC samples to approximate the intractable conditional expectation. MCEM requires samples at each iteration of the algorithm. In addition, the algorithm needs a separate method for calculating standard errors because it does not evaluate the likelihood function or its derivatives. A method for monitoring convergence may also be required (e.g., Booth and Hobert, 1999).

Compared to the MC methods described above, the MC kernel likelihood (MCKL) algorithm (De Valpine, 2004) takes a unique position in that it jointly samples the parameters and random effects to approximate the likelihood function. MCKL initially treats parameters as having probability densities and samples them from a posterior density as in Bayesian methods. The likelihood is estimated up to a constant as a weighted kernel density estimate where the weights are obtained by considering the posterior as an importance sampling density. The likelihood can also be estimated up to a constant as an unweighted kernel
density estimated divided by the prior. MCKL is a ‘single sample’ method because once the posterior samples of the parameters (along with samples of the random effects) are obtained, they are used during all iterations of the algorithm. MCKL is different from the typical ‘single sample’ method that samples the random effects given particular parameter values. De Valpine demonstrated the efficiency of the MCKL method in estimating the parameters of population dynamic models. However, a method for standard errors has not yet been provided for MCKL.

In this paper, we propose a MC local likelihood (MCLL) method for estimating GLMMs. MCLL is similar to MCKL in spirit: the algorithm begins by treating the parameters as random variables and sampling them jointly with random effects from the posterior distribution corresponding to a particular prior distribution. (We discuss how to choose the prior later in this paper.) The likelihood function is then approximated up to a constant by estimating the density of the posterior samples of the parameters and dividing this by the prior. In contrast to MCKL, we approximate the posterior density using local likelihood density estimation (Hjort and Jones, 1996; Loader, 1996), in which the log-density is locally approximated by a polynomial. The unweighted version of MCKL can be seen as a special case of MCLL with a polynomial of degree zero. One motivation for MCLL is that the kernel density estimator usually shows a substantial bias near peaks whereas the local likelihood density estimator does not (Loader, 1999, Ch.2). Furthermore, MCLL can exploit the form of the local likelihood density estimate to provide estimates of standard errors and the Bayes factor that are accurate and easy to calculate.

The outline of this paper is as follows. In Section 2, we introduce the general idea of local likelihood density estimation. The MCLL algorithm is then described in detail including specific implementation issues. In Sections 3 and 4, we discuss computation of standard errors and the Bayes factor. Empirical and simulation studies are then provided in Sections
4 and 5 to evaluate the proposed MCLL algorithm. The paper ends with some concluding
remarks concerning open questions and the breadth of applications that can benefit from
the MCLL approach.

2. Monte Carlo Local Likelihood Method

2.1 Local Likelihood Density Estimation

Suppose we have a sample \( \theta^{(1)}, ..., \theta^{(m)} \) from an unknown density \( f(\theta) \) (in our context, this
is the posterior density). Define a localized log-likelihood for \( f(\theta) \) as

\[
\hat{l}(\theta) = \sum_{j=1}^{m} K_h(\theta(j) - \theta) \log f(\theta(j)) - m \int K_h(u - \theta)f(u)du,
\]

where \( K_h(\cdot) \) represents a symmetric unimodal density (or kernel function) whose argument is
divided by the corresponding element of \( h \), a vector of bandwidths. The local polynomial
approximation supposes that \( \log f(\theta(j)) \) can be well approximated by a low-degree polynomial
in a neighborhood of the fitting point \( \theta \) (Loader, 1996). That is,

\[
\log f(\theta(j)) \approx P_a(\theta(j) - \theta),
\]

where \( a \) represents the parameters of the local polynomial function. For example, in the
three dimensional case \( (d=3) \), \( \log f(\theta(j)) \) can be approximated by a quadratic function in
the vicinity of \( \theta \)

\[
P_a(\theta(j) - \theta) = a_0 + a_1(\theta(j) - \theta_1) + a_2(\theta(j) - \theta_2) + a_3(\theta(j) - \theta_3)
\]

\[
+ \frac{1}{2}a_4(\theta(j) - \theta_1)^2 + \frac{1}{2}a_5(\theta(j) - \theta_2)^2 + \frac{1}{2}a_6(\theta(j) - \theta_3)^2
\]

\[
+ a_7(\theta(j) - \theta_1)(\theta(j) - \theta_2) + a_8(\theta(j) - \theta_1)(\theta(j) - \theta_3)
\]

\[
+ a_9(\theta(j) - \theta_2)(\theta(j) - \theta_3). \tag{2}
\]

When the components of \( h \) go to infinity, maximizing \( \hat{l}(\theta) \) is equivalent to maximizing the
usual likelihood. With a moderate \( h \), \( \hat{l}(\theta) \) is a semi-parametric version of the likelihood.
The localized log-likelihood \( \hat{i}(\theta) \) can then be approximated as

\[
\hat{i}(\theta, a) = \sum_{j=1}^{m} K_h(\theta^{(j)} - \theta)P_a(\theta^{(j)} - \theta) - m \int K_h(u - \theta) \exp(P_a(u - \theta)) du,
\]

where the parameter space for \( a \) is assumed to be an open set which holds if \( K_h \) is continuous with bounded support.

2.2 MCLL Procedure

MCLL begins by generating Markov chain Monte Carlo (MCMC) samples of model parameters from the posterior for a particular prior distribution. Then the algorithm involves two nested maximization steps: maximizing an approximate likelihood \( \hat{L}(y|\theta) \) over \( \theta \), with each evaluation of \( \hat{L}(y|\theta) \) requiring maximization of the localized log-likelihood over the parameters \( a \) of the local polynomial function. These two nested maximization steps iterate until convergence.

Specifically, assuming a \( d \)-dimensional parameter space \( \theta \) with observed data vector \( y \), the MCLL algorithm proceeds as follows:

Step 1. Choose a prior \( p(\theta) \) and use an MCMC method to obtain samples from the posterior \( p(\theta|y) \)

\[
p(\theta|y) = \frac{L(y|\theta)p(\theta)}{C_s},
\]

where the normalizing constant is \( C_s = \int L(y|\theta)p(\theta)d\theta \).

Step 2. Maximize an approximate likelihood, defined up to the unknown constant \( C_s \) by

\[
\hat{L}(y|\theta) = \frac{1}{p(\theta)}P_{sp}(\theta|y),
\]

where \( P_{sp}(\theta|y) \) is the local likelihood estimate of the posterior density. Specifically, for a given value of \( \theta \), this is obtained by assuming that the log-posterior density can be locally approximated by a polynomial function \( P_a(\theta^{(j)} - \theta) \) with parameters \( a \). The \( a \) parameters
are estimated for a particular $\theta$ by maximizing a localized version of the log-likelihood in (3).

The MCLL algorithm is akin to an unweighted version of the MCKL method in that the approximate likelihood function (smoothed likelihood in the MCKL case) is obtained by dividing an estimate of the posterior density by the known prior density. With MCKL, a weighted version can be formulated which can be seen as an unnormalized, importance-sampled kernel estimate of the true likelihood $L(y|\theta)$. However, when local density estimation is used as in MCLL, there is no clear mechanism for creating a weighted version. Moreover, De Valpine (2004) noted that there is little difference in performance between the weighted and unweighted versions of MCKL when a diffuse prior distribution is used. Therefore, we adopt an unweighted version as the main device for MCLL.

2.3 Implementation Issues

There are several issues to be discussed in implementing the MCLL algorithm. First, the bandwidth is chosen in Step 2 by considering the bias-variance trade-off. We choose a bandwidth at each data point so that the local neighborhood contains a specified number of points. The degree of the local polynomial can also affect the bias-variance trade-off. We choose a quadratic function as a default following Loader (1999, Ch.2) who shows that a low degree polynomial is often sufficient. In addition, a tricube weight function, $K(u) = \frac{e^{-|u|}}{1+|u|^3}$, is chosen as a default.

Second, we use an orthogonal transformation of the posterior samples $\theta^{(j)}$, $\tilde{\theta}^{(j)} = L^{-1}(\theta^{(j)} - \bar{b})$, where $\bar{b}$ is the mean of the posterior samples $\theta^{(j)}$ and $L$ is the Cholesky decomposition of the empirical covariance matrix $\widehat{\text{Cov}}(\theta)$ of the posterior samples $\theta^{(j)}$ so that $\widehat{\text{Cov}}(\theta) = LL^T$. The transformed $\tilde{\theta}^{(j)}$ have an identity covariance matrix and a zero mean vector.

This orthogonal transformation is also called data preshifting (Wand and Jones, 1993; Duong and Hazelton, 2003). Preshifting posterior samples is useful in implementing MCLL.
because it simplifies the integral term in (3). Specifically, if the elements of \( \theta \) are approximately independent in the posterior, then interactions terms in \( P_a(\theta^{(j)} - \theta) \) can be dropped. In addition, a product kernel can be used, with

\[
K_n(\theta^{(j)} - \theta) = \prod_{i=1}^{d} K_0 \left( \frac{\theta^{(j)}_i - \theta_i}{h_i} \right),
\]

for a one-dimensional kernel \( K_0 \). With these two simplifications, the multidimensional integral can be factorized as a product of one-dimensional integrals. In addition, the orthogonal transformation standardizes the bandwidth choice by transforming the parameters to be on the same scale.

Third, we use a log-transformation of standard deviation parameters. This has several advantages: 1) it avoids need for a modified kernel in Step 2 to handle truncation of the density at zero, 2) the posterior distributions are closer to normal, so that the data preshielding operation produces approximately independent rather than only approximately uncorrelated samples, and 3) the log-posterior is better approximated by a quadratic function.

Fourth, diffuse priors can be chosen for the regression coefficients and log standard deviation parameters, in which case the posterior mean estimates (automatically obtained in Step 1) are also close to the MLEs. Note that even if informative priors are used, the MCLL algorithm provides results close to the MLEs, which will not necessarily be true for estimates of the posterior means. Informative priors may be useful for improving mixing in MCMC in Step 1 but some care is required. We illustrate the choice of priors for given problems in the empirical and simulation study sections.

We wrote an R package mcll that implements the MCLL algorithm (Step 2 maximizations). It is available at http://cran.r-project.org/web/packages/mcll/index.html, along with a vignette illustrating how we carried out the calculations for the example in Section 5.
3. Standard Errors

Asymptotic theory for the MLE suggests obtaining standard error estimates using the Hessian matrix of the log-likelihood function evaluated at the MLE. Analogously, one could calculate the Hessian matrix of \( \hat{L}(y|\theta) \) through numerical differentiation of the estimated log-likelihood function. However, curvatures obtained by numerical differentiation will be sensitive to the bandwidth choice.

Therefore, we derive an alternative way of computing the Hessian matrix for MCLL. First write down the log-likelihood function

\[
\log L(y|\theta) = \log p(\theta|y) - \log p(\theta) + C_s, \tag{6}
\]

where \( \log p(\theta|y) \) is the log-posterior, \( \log p(\theta) \) is the log-prior, and \( C_s \) is a constant.

The required Hessian is

\[
\frac{\partial^2}{\partial \theta^2} \log L(y|\theta) \bigg|_{\hat{\theta}} = \frac{\partial^2}{\partial \theta^2} \log p(\theta|y) \bigg|_{\hat{\theta}} - \frac{\partial^2}{\partial \theta^2} \log p(\theta) \bigg|_{\hat{\theta}}.
\]

Using \( \hat{L}(y|\theta) \), instead of \( L(y|\theta) \), we obtain

\[
H_L = H_{p_{\text{a}}} - H_{p_{\text{r}}},
\]

where \( H_L, H_{p_{\text{a}}}, \) and \( H_{p_{\text{r}}} \) are the Hessian matrices of the approximate log-likelihood \( \log \hat{L}(y|\theta) \), the log-posterior \( \log p(\theta|y) \), and the log-prior \( \log p(\theta) \), respectively.

Typically \( H_{p_{\text{r}}} \) can be solved analytically. To obtain \( H_{p_{\text{a}}} \), we use the quadratic approximation of the log-posterior obtained using local likelihood density estimation. For example, for the case \( d = 3 \) and a quadratic function as given in (2), the Hessian matrix of the approximation is

\[
H_{p_{\text{a}}} \approx \begin{bmatrix}
\hat{a}_4 & \hat{a}_7 & \hat{a}_8 \\
\hat{a}_7 & \hat{a}_5 & \hat{a}_9 \\
\hat{a}_8 & \hat{a}_9 & \hat{a}_6
\end{bmatrix}.
\]

The coefficients for the interaction terms in (2) correspond to the off-diagonal terms (\( \hat{a}_7 \)
to $\hat{a}_0$ in (7) and are zero if the elements of $\mathbf{\theta}$ are uncorrelated in the posterior. This will be approximately true if the orthogonal transformation has been used. Thus in practice, these off-diagonal terms are set to zero and not estimated. In addition, the Delta method is used to obtain the approximate standard errors for the back-transformed $\mathbf{\theta}$.

4. Bayes Factors

Estimation of Bayes factors is a difficult problem because the marginal likelihood integrated over the random effects and model parameters is not easily computed from the output of the MCMC algorithm. Although the main motivation for MCLL is to approximate the MLE, we now describe how the Bayes factor can readily be computed as a product of the MCLL algorithm.

A Bayes factor for model $M_1$ and $M_2$ can be defined as the ratio of the corresponding marginal likelihoods (integrated over the random effects and model parameters)

$$BF_{12} = \frac{p(y|M_1)}{p(y|M_2)},$$

where

$$p(y|M_k) = \int p(y|\mathbf{\theta}_k, M_k)p(\mathbf{\theta}_k|M_k)d\mathbf{\theta}_k.$$ 

Here $p(y|\mathbf{\theta}_k, M_k)$ is the joint density of the responses given model $M_k$ with parameter values $\mathbf{\theta}_k$ and $p(\mathbf{\theta}_k|M_k)$ is the prior density for the model parameters $\mathbf{\theta}_k$ in model $M_k$.

The posterior density of the model parameters for $M_k$ at the MCLL estimates $\hat{\mathbf{\theta}}_k$ is

$$p(\hat{\mathbf{\theta}}_k|y, M_k) = p(\hat{\mathbf{\theta}}_k|M_k)p(y|\hat{\mathbf{\theta}}_k, M_k)\frac{p(\mathbf{\theta}_k|M_k)}{p(y|M_k)}.$$

Dividing both sides by the prior $p(\hat{\mathbf{\theta}}_k|M_k)$, we derive

$$\hat{L}(y|\hat{\mathbf{\theta}}_k, M_k) = \frac{p(\hat{\mathbf{\theta}}_k|y, M_k)}{p(\mathbf{\theta}_k|M_k)} = \frac{p(y|\hat{\mathbf{\theta}}_k, M_k)}{p(y|M_k)}. \tag{8}$$
The ratio for models 2 and 1 is
\[
\frac{\hat{L}(y|\mathbf{\theta}_2, M_2)}{\hat{L}(y|\mathbf{\theta}_1, M_1)} = \frac{p(y|M_1)}{p(y|M_2)} \times \frac{p(y|\mathbf{\theta}_2, M_2)}{p(y|\mathbf{\theta}_1, M_1)}.
\]

The Bayes factor \(\hat{\text{BF}}_{12}\) is therefore obtained as
\[
\hat{\text{BF}}_{12} = \frac{\hat{L}(y|\mathbf{\theta}_2, M_2)}{\hat{L}(y|\mathbf{\theta}_1, M_1)} \times \frac{p(y|\mathbf{\theta}_1, M_1)}{p(y|\mathbf{\theta}_2, M_2)},
\]
where \(\hat{L}(y|\mathbf{\theta}_k, M_k)\) \((k = 1, 2)\) are by-products of the MCLL algorithm and the likelihood \(p(y|\mathbf{\theta}_k, M_k)\) can be obtained using importance sampling (e.g., Durbin and Koopman, 1997; Shephard and Pitt, 1997). Details on the importance sampling method are provided in the Web Appendix 1.

5. Empirical Study
To illustrate the proposed algorithm, we consider the salamander mating data first used by McCullagh and Nelder (1989). The dataset consists of three separate experiments, each involving matings among salamanders of two different populations, called Rough Butt (RB) and White Side (WS). Sixty females and sixty males of two populations of salamander were paired by a crossed, blocked, and incomplete design in an experiment to investigate whether the two populations have developed generic mechanisms which would prevent inter-breeding. The response is a binary variable indicating whether mating was successful between female \(i\) and male \(j\). We adopted model A used by Karim and Zeger (1992)
\[
\logit(p(y_{ij} = 1|z_i^f, z_j^m)) = \beta_1 + \beta_2 x_{1i} + \beta_3 x_{2j} + \beta_4 x_{1i} x_{2j} + z_i^f + z_j^m,
\]
where the covariates are dummy variables for White Side female \((x_i)\), White Side male \((x_j)\), and the interaction \((x_{1i} x_{2j})\). The two crossed random effects are random intercepts \(z_i^f \sim N(0, \sigma_f^2)\) for females and \(z_j^m \sim N(0, \sigma_m^2)\) for males. Each salamander participates in six matings, resulting in 360 matings in total. The two variance components were reparameterized as \(\tau_f = \log \sigma_f\) and \(\tau_m = \log \sigma_m\).
The M CLL parameter estimates were compared to the estimates based on the Laplace approximation and the posterior means under a vague prior. They were also compared with the parameter estimates reported in the literature from various estimators, such as MCEM (Booth and Hobert, 1999), PQL (Breslow and Clayton, 1993), and MCMLE (Sung and Geyer, 2007). In addition, the MCKL method (De Valpine, 2004) was implemented for a comparison with M CLL. In the Web Appendix 1, we also evaluated the likelihood ratio statistic and the Bayes factor for testing the interaction effect ($\beta_i$).

5.1 Implementation

Diffuse normal priors were specified for the fixed effect (regression coefficient) parameters, with mean 0 and standard deviation 100. The same log-normal prior was specified for both $\sigma_f$ and $\sigma_m$. Specifically, this distribution had mean 0.5 and variance 0.2, which corresponds to a mean of -0.987 and variance of 0.588 on the log scale. At the 0.05 quantile (0.11) of this prior for $\sigma_f$ and $\sigma_m$, the random effects have a relatively small impact on the resulting probabilities, whereas at the 0.95 quantile (1.3) of the prior for $\sigma_f$ and $\sigma_m$, the resulting probabilities can be shifted dramatically, particularly if they are close to 0.5 based on the fixed effects component.

The Bayesian software WinBUGS (Lunn et al., 2000) together with the R package R2WinBUGS (Sturtz et al., 2005) was used to obtain the posterior samples in Step 1. Three chains were used with relatively diffuse starting values. Each chain was run for 1,000 iterations after a 2,000 iteration burn-in period. For convergence assessment, the Gelman-Rubin statistic (Gelman and Rubin, 1992) was used in addition to graphical checks such as trace plots and autocorrelation plots. For Step 2, we used our own R package mclll which has an option of using the R package locfit (Loader, 2012). For the bandwidth selection in Step 2, we choose the bandwidth to be the $\alpha$ quantile of the distances between $\theta$ and the observations. Specifically, for $0 < \alpha < 1$, let $n$ be the sample size, $d(\theta, \theta^{(j)}) = |\theta - \theta^{(j)}|$, and $k = \lfloor n\alpha \rfloor$. 
Then take the bandwidth to be the $k$th smallest values of $d(\theta, \theta^{(j)})$. Note that this bandwidth choice ensures each evaluation of the kernel will incorporate the same number of observations. We chose $\alpha = 0.7$ as suggested by Loader (1999), although values of $\alpha$ between 0.5 and 0.8 produced similar results, with the average differences less than 10% in the parameter estimates.

To implement the Laplace approximation, we used the R function `lmer` in the `lme4` package (Bates and Maechler, 2009). For adaptive quadrature, `xtmologit` in Stata (StataCorp, 2009) was used. To implement the MCKL method, we followed the procedure taken by Joon (2011) using the same posterior samples as in the MCLL method. Specifically, for the bandwidth choice for MCKL, we took diagonal elements of the covariance matrix of the kernel density to be proportional to the marginal posterior variances in each dimension of the posterior space with a proportionality constant $q^2$. We used $q = 0.5$ although 10 different values (0.1 to 1.0) were also examined for $q$. We also adjusted for smoothing bias in the MCKL estimates using posterior cumulants as suggested by De Valpine (2004).

5.2 Results

Table 1 lists the parameter estimates for model (10) from a variety of estimators in the literature. Standard errors are included when they were reported in the original papers.

[Table 1 about here.]

Overall, the results from MCLL are comparable to the other estimators. The regression coefficient estimates are a little smaller than the other estimates except PQL and MCMLE. The standard deviation estimates are close to the estimates from adaptive quadrature with three quadrature points. The MCKL parameter estimates are a little smaller than the MCLL estimates. The MCKL estimates also vary somewhat with the choice of bandwidth.
6. Simulation Studies

Two simulation studies were conducted to evaluate the performance of the MCLL method using 1) simulated salamander mating data for a generalized linear mixed model with crossed random effects and 2) simulated birth weight data for a linear mixed model. This linear mixed model is considered to evaluate the MCLL method when MLEs can be calculated exactly.

6.1 Simulated Salamander Mating Data

The first example is closely related to the salamander mating data (McCullagh and Nelder, 1989). 100 datasets were generated based on model (10) using the same true parameter values considered by other researchers (e.g., Lin and Breslow, 1996), namely \( \beta = (1.06, -3.05, -0.72, 3.77)' \) and \( (\sigma_j^2, \sigma_m^2)' = (.50, .50)' \). To implement the MCLL method, the same settings were used as in the empirical study. The performance of MCLL was compared to the Laplace approximation using the \texttt{lme4} R package (Bates and Maechler, 2009), posterior mean estimates, and MCMLE using the \texttt{bmer} R package (Geyer and Sung, 2006) with 10,000 samples.

6.2 Simulated Birth Weight Data

For the second example, we use a linear mixed model because it allows us to compare our estimates with exact MLEs. In the original dataset, from Rabe-Hesketh et al. (2008), there was birth weight and covariate information on 1,000 families comprising mother, father, and one child. A two-level linear mixed model was formulated for family members \( i \) nested in families \( j \) with three uncorrelated random coefficients as

\[
y_{ij} = x'_{ij} \beta + \alpha_{1ij}^{(2)} [M_i + K_i/2] + \alpha_{2ij}^{(2)} [F_i + K_i/2] + \alpha_{3ij}^{(2)} [K_i/\sqrt{2}] + \epsilon_{ij},
\]

where \( x_{ij} \) is a vector of covariates with regression coefficients \( \beta \), and \( M_i, K_i, \) and \( F_i \) are dummy variables for mother, child, and father. The covariates include dummy variables for male, and mother being aged 20–35, younger than 35, or older than 35 at the time of the birth. The three random effects at level 2, \( \alpha_{1ij}^{(2)}, \alpha_{2ij}^{(2)}, \) and \( \alpha_{3ij}^{(2)} \) are i.i.d as \( \alpha_{kij}^{(2)} \sim N(0, \sigma_k^2) \)
with $k = 1, 2, 3$. The level-1 residuals have a normal distribution, $e_{ij}^{(2)} \sim N(0, \sigma_E^2)$ and are independent of the random effects. Here $\sigma_A$ can be interpreted as the additive genetic standard deviation and $\sigma_E$ as the unique environment standard deviation.

We generated 100 datasets based on model (11) using the MLEs of the original data as true values, which are $\beta = (3368.09, 155.34, 126.94, 213.43)'$ and $(\sigma_A, \sigma_E)' = (311.21, 374.66)'$.

Diffuse normal priors were chosen for the regression coefficients (mean 0, standard deviation 1,000). The same log-normal prior was specified for both $\sigma_A$ and $\sigma_E$, with mean 500 and variance 20,000, which corresponds to a mean of 6.17 and variance of 0.073 on the log scale. The 0.05 quantile of this distribution is 307 grams, which corresponds to random effects that with 95% probability fall between -614 and 614 grams. The 0.95 quantile is 745 grams. This choice of prior gives more weight to large values of $\sigma_A$ and $\sigma_E$ (compared with the true values) than would a noninformative prior and the posterior distributions for $\sigma_A$ and $\sigma_E$ are centered in the left tail of the prior. However, this choice allows us to see how well MCLL performs when an informative prior is used, even when that prior is poorly chosen.

To obtain the MLEs, we used the R function `lme` in the package `nlme` (Pinheiro et al., 2012).

6.3 Results

Table 2 presents estimated bias and mean squared error (MSE) from the first simulation study using the salamander mating data.

[Table 2 about here.]

The MCLL method performed well compared to the Bayesian and the Laplace approximation methods. For the regression parameters, the bias and MSE were quite similar between the methods except that MCMLE showed larger bias than the other methods. For the standard deviation parameters, MCLL showed smaller bias and MSE than the other methods.
Table 3 lists the average standard error estimates compared with the standard deviations of the parameter estimates (or the empirical standard errors) for the regression coefficients. Standard errors for the standard deviations $\sigma_f$ and $\sigma_m$ were not considered because the Wald-type tests and confidence intervals are inappropriate for these parameters (e.g., Stram and Lee, 1994).

[Table 3 about here.]

The results show that the means of the standard error estimates (SE) for MCLL were close to the empirical standard errors (SD) for all three methods. The MCLL standard errors tend to be more accurate than those from MCMLE and the Laplace approximation by looking at the SE to SD ratio. In addition, our standard error estimates tend to be more conservative than those from the Laplace approximation. Monte Carlo errors (MCEs) were about 10% of the means of the parameter estimates and less than 10% for the means of the standard errors in all methods.

For the linear mixed model example with the simulated birth weight datasets, we compared the MCLL estimates and posterior means with the MLEs. Figure 1 compares the distances from the MLEs between the two methods for each parameter.

[Figure 1 about here.]

Figure 1 shows that the posterior mean estimates display a marked bias (defined relative to the MLEs), which is evident in the point clouds being shifted away from zero on the $x$-axis. Although the MCLL estimates are more variable, the mean squared distances to the MLEs are smaller than for the posterior mean estimates except for $\beta_2$. The second step of the MCLL algorithm adjusts the estimates and we observe that they are no longer biased relative to the MLEs. This is strong evidence that MCLL estimates are closer to the MLEs than the posterior mean estimates. The MCEs were about 10% of the means of the parameter estimates for all methods.
Table 4 compares the average standard error estimates over replicates with the empirical standard errors.

[Table 4 about here.]

Table 4 shows that both the ML and MCLL standard errors are good approximations to the empirical standard deviations. The MCLL standard error estimates tend to be a bit more conservative than the ML standard errors. The MCEs for the means of the standard error estimates were less than 5% for MCLL and less than 10% for the ML method.

7. Concluding Remarks

The convergence of the MCKL estimator was shown in De Valpine (2004) based on proofs of convergence of kernel mode estimates. The proofs for MCKL cannot be directly applied to the MCLL method with a polynomial higher than degree zero. Unlike kernel density estimation, there has been no proof yet on the convergence of mode estimates in local density estimation, which would be an intermediate step in proving convergence of the MCLL estimator.

It is important to note that MCLL allows likelihood inference for any complex models in which ML estimation may be infeasible but MCMC methods are possible. For example, in addition to the GLMMs with crossed random effects considered here, the MCLL algorithm could be used to fit spatial models with higher dimensional latent variables such as for disease mapping (Lawson, 2008). In addition, the underlying idea of MCLL can be adapted to a purely Bayesian method. Specifically, one can estimate the posterior mode and Hessian at the mode by following the MCLL procedure but not dividing by the prior.

We have shown that the MCLL method provides results close to the MLEs even if priors are poorly chosen, whereas the posterior mean estimates are quite different. Therefore, when ML inference is desirable for highly complex models, the MCLL method is an effective and
practical choice. In addition, MCLL performs very favorably to existing methods such as the Laplace approximation and MCMLE.

Supplementary Materials

Web Appendix 1 referenced in Section 4 and 5 is available with this paper at the Biometrics website on Wiley Online Library.

References


StataCorp (2009). *Stata Statistical Software: Release 11*. StataCorp LP, TX.


791–795.
Figure 1. Distances from the MLEs for MCLL estimates (MCLL-MLE) and for posterior mean estimates (Post.m-MLE) for 100 simulated birth weight datasets
Table 1
Comparison of several estimators for the salamander mating data. Standard errors are given in parentheses if reported. MCEM: Booth and Hobert (1999); PQL: Breslow and Clayton (1993); Laplace: *lmer*; Adaptive quad(3): *zimlogit* with 3 quadrature points; MCMLE: Sung and Geyer (2007); MCMLE(bernor): *bernor* with $10^7$ samples; MCLL: MCLL method; Post.m: Posterior means (the posterior samples that were used for MCLL); MCKL: MCKL method after cumulant bias correction ($q = 0.5$).

<table>
<thead>
<tr>
<th>Method</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>$\beta_3$</th>
<th>$\beta_4$</th>
<th>$\sigma_m$</th>
<th>$\sigma_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCEM$^a$</td>
<td>1.02</td>
<td>-2.96</td>
<td>-0.69</td>
<td>3.63</td>
<td>1.18</td>
<td>1.12</td>
</tr>
<tr>
<td>PQL</td>
<td>0.79(0.32)</td>
<td>-2.29(0.43)</td>
<td>-0.54(0.39)</td>
<td>2.82(0.50)</td>
<td>0.79</td>
<td>0.72</td>
</tr>
<tr>
<td>Laplace</td>
<td>1.00(0.39)</td>
<td>-2.90(0.56)</td>
<td>-0.70(0.46)</td>
<td>3.59(0.64)</td>
<td>1.08</td>
<td>1.02</td>
</tr>
<tr>
<td>Adaptive quad(3)</td>
<td>1.01(0.41)</td>
<td>-2.95(0.58)</td>
<td>-0.70(0.48)</td>
<td>3.62(0.64)</td>
<td>1.16</td>
<td>1.10</td>
</tr>
<tr>
<td>MCMLE$^a$</td>
<td>1.00</td>
<td>-2.78</td>
<td>-0.47</td>
<td>3.52</td>
<td>1.17</td>
<td>1.10</td>
</tr>
<tr>
<td>MCMLE (bernor)$^b$</td>
<td>1.00 (0.37)</td>
<td>-2.79 (0.52)</td>
<td>-0.47 (0.50)</td>
<td>3.52 (1.00)</td>
<td>1.17</td>
<td>1.10</td>
</tr>
<tr>
<td>MCCL</td>
<td>0.93 (0.41)</td>
<td>-2.87(0.56)</td>
<td>-0.65(0.49)</td>
<td>3.59(0.67)</td>
<td>1.16</td>
<td>1.08</td>
</tr>
<tr>
<td>Post.m</td>
<td>1.01 (0.41)</td>
<td>-2.92 (0.58)</td>
<td>-0.69 (0.49)</td>
<td>3.58(0.66)</td>
<td>1.09</td>
<td>1.02</td>
</tr>
<tr>
<td>MCKL</td>
<td>0.84</td>
<td>-2.77</td>
<td>-0.54</td>
<td>3.47</td>
<td>1.13</td>
<td>1.08</td>
</tr>
</tbody>
</table>

$^a$ The parameter estimates for MCEM and MCMLE were obtained based on the estimates in the respective original papers that used a different parameterization of the same model. Standard errors for MCMLE could not be derived because the covariance matrix of the model parameters were not provided in the original paper.

$^b$ R package *bernor* was used to obtain MCMLE estimates and standard errors with $10^7$ samples as in Sung and Geyer (2007).
Table 2
Estimated bias and mean squared error (MSE) of the MCLL, Laplace approximation, posterior mean (Post.m), and MCMLE estimates for 100 simulated salamander datasets.

<table>
<thead>
<tr>
<th></th>
<th>True</th>
<th>Laplace</th>
<th>Post.m</th>
<th>MCMLE</th>
<th>MC LL</th>
<th>Laplace</th>
<th>Post.m</th>
<th>MCMLE</th>
<th>MC LL</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_1$</td>
<td>1.06</td>
<td>-0.03</td>
<td>-0.03</td>
<td>-0.07</td>
<td>-0.03</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>-3.05</td>
<td>0.04</td>
<td>0.01</td>
<td>0.12</td>
<td>0.01</td>
<td>0.22</td>
<td>0.21</td>
<td>0.20</td>
<td>0.21</td>
</tr>
<tr>
<td>$\beta_3$</td>
<td>-0.72</td>
<td>0.06</td>
<td>0.06</td>
<td>0.08</td>
<td>0.05</td>
<td>0.17</td>
<td>0.17</td>
<td>0.17</td>
<td>0.16</td>
</tr>
<tr>
<td>$\beta_4$</td>
<td>3.77</td>
<td>-0.04</td>
<td>-0.01</td>
<td>-0.14</td>
<td>-0.04</td>
<td>0.35</td>
<td>0.33</td>
<td>0.27</td>
<td>0.35</td>
</tr>
<tr>
<td>$\sigma_m$</td>
<td>0.71</td>
<td>-0.17</td>
<td>-0.21</td>
<td>-0.16</td>
<td>-0.13</td>
<td>0.10</td>
<td>0.09</td>
<td>0.08</td>
<td>0.07</td>
</tr>
<tr>
<td>$\sigma_f$</td>
<td>0.71</td>
<td>-0.15</td>
<td>-0.19</td>
<td>-0.20</td>
<td>-0.12</td>
<td>0.12</td>
<td>0.09</td>
<td>0.10</td>
<td>0.08</td>
</tr>
</tbody>
</table>
Table 3

Average standard error estimates for 100 simulated salamander datasets. SD is the empirical standard error (standard deviation of the parameter estimates). SE is the average of the standard error estimates, and SE/SD is the ratio of SE to SD.

<table>
<thead>
<tr>
<th></th>
<th>MCLL SE</th>
<th>MCLL SD</th>
<th>MCLL SE/SD</th>
<th>MCMLE SE</th>
<th>MCMLE SD</th>
<th>MCMLE SE/SD</th>
<th>Laplace SE</th>
<th>Laplace SD</th>
<th>Laplace SE/SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_1$</td>
<td>0.32</td>
<td>0.34</td>
<td>0.93</td>
<td>0.30</td>
<td>0.33</td>
<td>0.89</td>
<td>0.29</td>
<td>0.34</td>
<td>0.86</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>0.49</td>
<td>0.46</td>
<td>1.07</td>
<td>0.48</td>
<td>0.43</td>
<td>1.12</td>
<td>0.44</td>
<td>0.46</td>
<td>0.95</td>
</tr>
<tr>
<td>$\beta_3$</td>
<td>0.39</td>
<td>0.40</td>
<td>0.97</td>
<td>0.43</td>
<td>0.40</td>
<td>1.08</td>
<td>0.37</td>
<td>0.41</td>
<td>0.90</td>
</tr>
<tr>
<td>$\beta_4$</td>
<td>0.60</td>
<td>0.58</td>
<td>1.03</td>
<td>0.66</td>
<td>0.50</td>
<td>1.33</td>
<td>0.53</td>
<td>0.59</td>
<td>0.90</td>
</tr>
</tbody>
</table>
Table 4

Average standard error estimates for 100 simulated birth weight datasets. SD is the empirical standard error (standard deviation of the parameter estimates), SE is the average of the standard error estimates, and SE/SD is the ratio of SE to SD.

<table>
<thead>
<tr>
<th></th>
<th>MCLL</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SE</td>
<td>SD</td>
<td>SE/SD</td>
<td>SE</td>
<td>SD</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>31.56</td>
<td>31.70</td>
<td>1.00</td>
<td>31.11</td>
<td>31.49</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>17.79</td>
<td>15.35</td>
<td>1.16</td>
<td>17.52</td>
<td>15.31</td>
</tr>
<tr>
<td>$\beta_3$</td>
<td>31.36</td>
<td>32.00</td>
<td>0.98</td>
<td>30.94</td>
<td>32.00</td>
</tr>
<tr>
<td>$\beta_4$</td>
<td>53.34</td>
<td>58.05</td>
<td>0.92</td>
<td>52.58</td>
<td>57.99</td>
</tr>
</tbody>
</table>