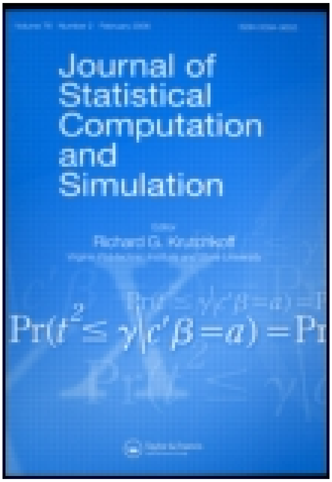

Journal of Statistical Computation and Simulation
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gscs20

The monte carlo newton-raphson algorithm

Anthony Y. C. Kuk [a] & Yuk W. Cheng [a]
[a] Department of Statistics , The University of New South Wales , Sydney, NSW 2052, Australia

Published online: 04 Mar 2011.

To cite this article: Anthony Y. C. Kuk & Yuk W. Cheng (1997) The monte carlo newton-raphson algorithm, Journal of Statistical Computation and Simulation, 59:3, 233-250, DOI: 10.1080/00949657708811858

To link to this article: http://dx.doi.org/10.1080/00949657708811858

J. Statist. Comput. Simul., 1997, Vol. 59, pp. 233–250
Reprints available directly from the publisher
Photocopying permitted by license only

THE MONTE CARLO NEWTON-RAPHSON ALGORITHM

ANTHONY Y. C. KUK and YUK W. CHENG

*Department of Statistics, The University of New South Wales,
Sydney, NSW 2052, Australia*

(Received 11 November 1994; In final form 2 June 1997)

It is shown that the Monte Carlo Newton-Raphson algorithm is a viable alternative to the Monte Carlo EM algorithm for finding maximum likelihood estimates based on incomplete data. Both Monte Carlo procedures require simulations from the conditional distribution of the missing data given the observed data with the aid of methods like Gibbs sampling and rejective sampling. The Newton-Raphson algorithm is computationally more efficient than the EM algorithm as it converges faster. We further refine the procedure to make it more stable numerically. Our stopping criterion is based on a chi-square test for zero gradient. We control the type II error by working out the number of Monte Carlo replications required to make the non-centrality parameter sufficiently large. The procedure is validated and illustrated using three examples involving binary, survival and count data. In the last example, the Monte Carlo Newton-Raphson procedure is eight times faster than a modified version of the Monte Carlo EM algorithm.

Keywords: EM algorithm; Gibbs sampling; incomplete data; random effects; rejective sampling

1. INTRODUCTION

The Newton-Raphson and the EM (Dempster, Laird and Rubin, 1977) algorithms are two popular methods for finding maximum likelihood estimates with the latter particularly suited for incomplete data. Let $\boldsymbol{Y}$ denote the observed incomplete data, $\boldsymbol{Z}$ the missing or latent data and $\boldsymbol{X}=(\boldsymbol{Y}, \boldsymbol{Z})$, the complete data. The EM algorithm is most useful in situations where it is difficult to maximise the observed log-likelihood function $l(\theta;\boldsymbol{y})=\log f(\boldsymbol{y};\theta)$ directly whereas the log-likelihood function $l(\theta;\boldsymbol{x})=\log f(\boldsymbol{y},\boldsymbol{z};\theta)$ based on the complete data can be maximised easily. The idea behind the EM algorithm is to solve the difficult incomplete data problem via solving the easier complete data problem. Given $\theta^{(k)}$, the current estimate of the parameter θ, an approximation of the complete data log-likelihood function is obtained by taking conditional expectation. This is the so called E-step:

$$Q(\theta, \theta^{(k)}) = E\{l(\theta; X) | y; \theta^{(k)}\} = \int l(\theta; y, z) f(z|y; \theta^{(k)}) dz. \quad (1)$$

In the M-step of the algorithm, we maximise $Q(\theta,\theta^{(k)})$ as a function of θ to obtain the updated estimate $\theta^{(k+1)}$. The algorithm is then iterated until convergence.

Ever since its proposal, the EM algorithm has been applied successfully to solve a wide range of problems. To a certain extent, the enormous success and popularity of the method have diverted attention away from its limitations. A point which we feel has not been emphasised enough in the literature is that the EM algorithm is appealing only when both the E-step and the M-step of the algorithm can be implemented easily. When it is not possible to carry out the E-step analytically, a number of authors (Wei and Tanner, 1990; Sinha, Tanner and Hall, 1994; Chan and Ledolter, 1995) have proposed the Monte Carlo implementation of the E-Step by simulating $z_1, z_2, \ldots, z_M$ from the conditional distribution $f(z|y; \theta^{(k)})$. A Monte Carlo approximation of $Q(\theta, \theta^{(k)})$ is

$$Q_M(\theta, \theta^{(k)}) = \frac{1}{M} \sum_{i=1}^{M} l(\theta; y, z_i).$$

The subsequent M-step to obtain $\theta^{(k+1)}$ usually requires iterations unless closed form formula exists for the maximiser of $Q_M(\theta, \theta^{(k)})$. More importantly, the EM algorithm converges at a rather slow linear rate. The Monte Carlo version of the EM algorithm will suffer even more from this slow convergence as it requires a fresh set of simulations from a new distribution at each iteration. This is particularly so if a time consuming method like Gibbs sampling is used at each iteration. The purpose of this paper is to point out that a Monte Carlo implementation of the Newton-Raphson algorithm is a viable alternative to the Monte Carlo EM algorithm. As the Newton Raphson algorithm has a quadratic rate of convergence, we have good reason to believe that the Monte Carlo Newton Raphson procedure will converge faster than the Monte Carlo EM algorithm. The savings in computing time can be quite substantial particularly if methods like Gibbs sampling are used. Moreover, the Monte Carlo Newton-Raphson procedure produces standard error estimates as a by product. The details including the choice of stopping criterion are given in Section 2. In Section 3, we describe a refinement of the Monte Carlo Newton-Raphson algorithm to make it more stable numerically as well as a Monte Carlo version of the EM gradient algorithm due to Lange (1995). We illustrate the algorithm and compare it with the Monte Carlo EM algorithm using three data sets involving binary, survival and count data. The first two data sets serve as test cases because for them we can actually write down in closed form the likelihood function to be maximised without resorting to Monte Carlo method. Results from these test cases lend support to the validity of the Monte Carlo Newton-Raphson procedure. In terms of computing time, the Monte Carlo Newton-Raphson procedure is eight times faster than the Monte Carlo EM gradient algorithm in our last example.

2. THE MONTE CARLO NEWTON-RAPHSON METHOD

Let $l(\theta;y)=\log f(y;\theta)$ denote the log-likelihood function based on the observed data y, $l'(\theta;y)$ the $p\times 1$ vector of the first derivatives of $l(\theta;y)$ with respect to the components of $\theta=(\theta_1,\theta_2,\ldots,\theta_p)$ and $l''(\theta;y)$ the $p\times p$ matrix of the second derivatives of $l(\theta;y)$ with respect to the components of θ. The Newton-Raphson iteration is given by

$$\theta^{(k+1)}=\theta^{(k)}-\{l''(\theta^{(k)};y)\}^{-1}l'(\theta^{(k)};y). \tag{2}$$

For incomplete data problems, it is not always possible to obtain closed from expressions for $l'(\theta;y)$ and $l''(\theta;y)$. Louis (1982) expressed $l'(\theta;y)$ and $l''(\theta;y)$ in terms of the conditional expectations of certain functions of the complete data $X=(Y,Z)$ given the observed data y. To be specific

$$\begin{aligned}
l'(\theta;y) &= E\{l'(\theta;y,Z)\,|\,y;\theta\},\\
l''(\theta;y) &= E\{l''(\theta;y,Z)\,|\,y;\theta\}\\
&\quad + E\{l'(\theta;y,Z)l'^{T}(\theta;y,Z)\,|\,y;\theta\}\\
&\quad - l'(\theta;y)l'^{T}(\theta;y).
\end{aligned}$$

If the above conditional expectations cannot be performed analytically, we propose approximating them by simulating $z_1, z_2, \ldots, z_M$ from the conditional distribution of $\boldsymbol{Z}$ given y with θ set to the current estimate $\theta^{(k)}$. Note that as k changes, so is $\theta^{(k)}$ and a new set of $z_1, \ldots, z_M$ has to be simulated. The Monte Carlo approximations of $l'(\theta^{(k)}; y)$ and $l''(\theta^{(k)}; y)$ are given by

$$l'_M(\theta^{(k)}; y) = \frac{1}{M}\sum_{i=1}^{M} l'(\theta^{(k)}; y, z_i), \tag{3}$$

$$l''_M(\theta^{(k)}; y) = \frac{1}{M}\sum_{i=1}^{M} l''(\theta^{(k)}; y, z_i) + \left\{ \frac{1}{M}\sum_{i=1}^{M} l'(\theta^{(k)}; y, z_i) l'^T (\theta^{(k)}; y, z_i) - l'_M(\theta^{(k)}; y) l'^T_M(\theta^{(k)}; y) \right\}. \tag{4}$$

Substituting l'_M and l''_M for l' and l'' in (2), we obtain the Monte Carlo Newton-Raphson iterative procedure

$$\text{algorithm 1}: \qquad \theta^{(k+1)} = \theta^{(k)} - \{l''_M(\theta^{(k)}; y)\}^{-1} l'_M(\theta^{(k)}; y). \tag{5}$$

We suggest iterating (5) until it converges to say $\hat{\theta}_M$ which for large M should be a good approximation to $\hat{\theta}$, the maximum likelihood estimate. As a by product, we can use $-\{l''_M(\hat{\theta}_M; y)\}^{-1}$ to estimate the variance-covariance matrix of $\hat{\theta}_M$.

The stopping criterion used for the Monte Carlo Newton-Raphson procedure (5) should be more lenient that that of the analytic Newton-Raphson procedure due to the noise induced by the Monte Carlo sampling. One possibility is to stop the iterations when $\theta^{(k)}$ converges to say 2 or 3 significant digits. It is also unreasonable to assume that $l'_M(\theta^{(k)})$ will converge to 0 exactly due to noise. We should rather standardise the Monte Carlo gradient vector $l'_M(\theta^{(k)})$ by dividing each of its component by a suitable estimate of its standard error. Alternatively, we can construct an overall statistic

$$W(\theta^{(k)}) = \{l'_M(\theta^{(k)})\}^T \hat{\sum}^{-1} l'_M(\theta^{(k)}), \tag{6}$$

where $\hat{\Sigma}$ is a suitable estimate of the variance-covariance matrix Σ of $l'_M(\theta^{(k)})$. The distribution of W under the hypothesis that the analytic gradient $l'(\theta^{(k)})$ is 0 is a chi-square distribution with p degrees of freedom since $l'_M(\theta^{(k)})$ defined by (3) is an average over Monte Carlo replicates and so is asymptotically normal for large M. This suggests that we can use the following level α test

$$W(\theta^{(k)}) < \chi^2_p(\alpha) \tag{7}$$

as a stopping criterion, where $\chi^2_p(\alpha)$ is the upper 100α-percentile of a chi-square distribution with p degrees of freedom. If $z_1,\ldots,z_M$ are independent and identically distributed, we can obviously let $\hat{\Sigma} = S/M$ where S is the sample covariance matrix of the Monte Carlo replicates $l'(\theta^{(k)};y,z_1),\ldots,l'(\theta^{(k)};y,z_M)$. If Gibbs sampling or some other Markov Chain sampling methods are used, the resulting $z_1,\ldots,z_M$ are dependent and S/M is not a suitable candidate for $\hat{\Sigma}$. A simple way to account for the dependence of $z_1,\ldots,z_M$ is to use the method of batching (Geyer, 1992, P.476). The idea is to group $z_1,\ldots,z_M$ into $G = M/L$ groups of size L each and rewrite (3) as

$$l'_M(\theta^{(k)}) = \frac{1}{G}\sum_{g=1}^{G} \boldsymbol{b}_g,$$

where

$$\boldsymbol{b}_g = \frac{1}{L}\sum_{i=(g-1)L+1}^{gL} l'(\theta^{(k)};y,z_i)$$

is the batched mean of the g^{th} group. By treating the batched means as independent, we can estimate Σ by S_b/G, where S_b is the sample covariance matrix of the G vectors of batched means $\boldsymbol{b}_1,\ldots,\boldsymbol{b}_G$. In our last example (Section 7), we use $M = 20000$ and group $z_1,\ldots,z_M$ into $G = 800$ groups of size $L = 25$ each for the purpose of batching.

The W criterion given by (7) only controls the type I error which in the present context is the error of not stopping when the true gradient $l'(\theta^{(k)})$, which we try to estimate by $l'_M(\theta^{(k)})$, is already zero. In other words, criterion (7) protects us from stopping too late. To avoid stopping too early, we need to control the type II error which in this case means stopping prematurely when the gradient vector $l'(\theta^{(k)})$ is not yet zero. For the purpose of calculating type II error, we note that when $l'(\theta^{(k)})\neq 0$, the statistic $W(\theta^{(k)})$ has asymptotically a non-central χ_p^2 distribution with non-centrality parameter

$$\xi = l'^T \textstyle\sum^{-1} l',$$

where

$$l' = l'(\theta^{(k)}) = E\{l'_M(\theta^{(k)})\}.$$

Using the method of batching with G groups of size L each, we can approximate Σ by Σ_b/G where Σ_b is the population covariance matrix of the batched mean and thus the non-centrality parameter becomes

$$\xi = G\delta^2$$

where

$$\delta^2 = l'^T \textstyle\sum_b^{-1} l'$$

can be interpreted as a square norm for the gradient vector l'. It follows that by increasing G, we can increase the non-centrality parameter $\xi = G\delta^2$ and hence control the type II error. In our last example in Section 7, there are $p=8$ parameters and we fix the level of the type I error at $\alpha=0.1$ so that the critical value is $\chi_8^2(0.1) = 13.36$. Fixing the batch size at $L=25$, we want to choose G large enough so as to control the type II error at the same level 0.1 when the true gradient l' has square norm

$$\delta^2 = 0.02 = 8(0.05)^2$$

or just 0.05^2 when averaged over the 8 components of l'. Using the NAG subroutine G01GCF for calculating non-central χ_8^2 distributions, we find that the type II error equals 0.1 if the non-centrality parameter

$$\xi = G\delta^2 = 16.$$

Thus $G=16/\delta^2=16/0.02=800$ and so $M=GL=20000$ Monte Carlo replications are required.

3. TWO VARIANTS OF THE METHOD

In carrying out algorithm 1 as defined by (5), a legitimate concern is about the positive definiteness of $-l''_M(\hat{\theta}_M; y)$. For regular problems, $E_\theta\{-l''(\theta; Y)\}$ is the covariance matrix of $l'(\theta; Y)$ and thus is positive definite. If follows from the law of large numbers and the asymptotic consistency of maximum likelihood estimator that the observed information matrix $-l''(\hat{\theta}; y)$ should also be positive definite. Provided that M is large, we can also expect $-l''_M(\hat{\theta}_M; y)$, the Monte Carlo approximation of $-l''(\hat{\theta}; y)$ to be positive definite. However, it is possible that $-l''_M(\theta^{(k)}; y)$ is not positive definite if $\theta^{(k)}$ is far away from $\hat{\theta}$. Our experience suggested that the non-positive definiteness of $-l''_M(\theta^{(k)}; y)$ can lead to the non-convergence of (5). If this is the case, we suggest the following variation of (5)

algorithm 2 : $$\theta^{(k+1)} = \theta^{(k)} - \{l''_{M1}(\theta^{(k)}; y)\}^{-1} l'_M(\theta^{(k)}; y), \qquad (8)$$

where

$$l''_{M1}(\theta^{(k)}; y) = \frac{1}{M} \sum_{i=1}^{M} l''(\theta^{(k)}; y, z_i) \qquad (9)$$

is just the first summation appearing on the right hand side of (4) and can be interpreted as the Monte Carlo approximation of the so called complete information. Upon convergence of (8), we still use $-l''_M(\hat{\theta}_M; y)^{-1}$ to estimate the covariance matrix of $\hat{\theta}_M$.

There is an interesting connection between algorithm 2 and the Monte Carlo EM algorithm. Recall that at the M-step of the EM procedure, one has to maximise $Q(\theta; \theta^{(k)})$ defined by (1) with respect to θ to obtain $\theta^{(k+1)}$. In general, this maximisation is carried out iteratively using, say, Newton's method

$$\theta_{\text{new}} = \theta_{\text{old}} - \{Q''(\theta_{\text{old}})\}^{-1} Q'(\theta_{\text{old}}),$$

where $Q'(\theta)$, $Q''(\theta)$ are the matrices of first and second order derivatives of $Q(\theta; \theta^{(k)})$ with respect to θ. It is clear from (1) that

$$Q'(\theta) = E\{l'(\theta; y, Z) \,|\, y; \theta^{(k)}\}$$

and

$$Q''(\theta) = E\{l''(\theta; y, \boldsymbol{Z}) | y; \theta^{(k)}\}.$$

Thus $l'_M(\theta^{(k)})$ given by (3) and $l''_{M1}(\theta^{(k)})$ given by (9) are just Monte Carlo estimates of $Q'(\theta^{(k)})$ and $Q''(\theta^{(k)})$ respectively. Hence algorithm 2 is related to the EM algorithm in that it replaces the M-step by a single iteration of Newton's method. Such a method has been proposed by Lange (1995) who also proved that his algorithm is locally equivalent to the EM algorithm. The only difference between our algorithm 2 and Lange's algortihm is that we are using Monte Carlo methods to approximate $Q'(\theta^{(k)})$ and $Q''(\theta^{(k)})$.

Based on Lagne's results,we expect algorithm 2 to behave very much like the Monte Carlo EM algorithm. In particular, the rate of convergence is linear which is infeasible if the simulation of $z_1, \ldots, z_M$ at each iteration is time consuming. The Newton Raphson procedure enjoys a quadratic rate of convergence. However, a naive Monte Carlo implementation of the Newton Raphson procedure as defined previously by (5) is found to be erratic and may even diverge particularly when $-l''_M(\theta^{(k)})$ is non-positive definite. Thus there is a need to refine (5) so that it behaves properly. The refinements that we propose (algorithm 3) are

(i) Halve the bracketed term on the right hand side of (4) s times ($s = 0, 1 \ldots$) until the resulting matrix

$$H_s = \frac{1}{M}\sum_{i=1}^{M} l''(\theta^{(k)}; y, z_i) + \frac{1}{2^s}\left\{\frac{1}{M}\sum_{i=1}^{M} l'(\theta^{(k)}; y, z_i) l'^T(\theta^{(k)}; y, z_i) - l'_M(\theta^{(k)}; y) l_M(\theta^{(k)}; y)\right\}$$

is negative definite.

(ii) Compute

$$\theta_0^{(k+1)} = \theta^{(k)} - H_s^{-1} l'_M(\theta^{(k)}; y).$$

(iii) Halve the step length t times ($t = 0, 1, \ldots$) to obtain

$$\theta_t^{(k+1)} = \theta^{(k)} + \frac{1}{2^t}(\theta_0^{(k+1)} - \theta^{(k)})$$

and stop when $W(\theta_t^{(k+1)}) < W(\theta^{(k)})$. Update the estimate from $\theta^{(k)}$ to $\theta^{(k+1)} = \theta_t^{(k+1)}$.

The rationale behind (iii) is that $W(\theta^{(k)})$ as defined by (6) is a statistic for testing whether the analytic gradient $l'(\theta^{(k)}) = 0$. As the likelihood function is usually intractable for problems which requires Monte Carlo method of inference, we cannot check whether each iteration increases the likelihood. We impose $W(\theta_t^{(k+1)}) < W(\theta^{(k)})$ instead which suggests that $l'(\theta_t^{(k+1)})$ is "closer" to 0 than $l'(\theta^{(k)})$. With the above refinements, the resulting method which we call algorithm 3 is found to perform satisfactorily. When appplied to the polio incidence data in Section 7, algorithm 1 diverges. Both algorithms 2 and 3 converge but algorithm 2 requires eight times more computing time.

4. MULTIPLE RUNS

To assess the extra-variation due to Monte Carlo sampling and to account for such variation in the reported standard errors, we suggest running the Monte Carlo Newton Raphson procedure independently r times using different initial seeds. Let $\hat{\theta}_1, \ldots, \hat{\theta}_r$ denote the estimates obtained from the r runs. Following Kuk and Chen (1992), we suggest using

$$\bar{\theta} = \frac{1}{r}\sum_{i=1}^{r} \hat{\theta}_i$$

as the final estimate of θ. The asymptotic covariance matrix of $\bar{\theta}$ can be estimated by

$$V = V_1 + V_2$$

where

$$V_1 = -\{l''_M(\bar{\theta}; y)\}^{-1},$$
$$V_2 = \frac{1}{r} S_\theta$$

and S_θ is the sample covariance matrix of $\hat{\theta}_1, \ldots, \hat{\theta}_r$. Note that we are explicitly accounting for the extra variation due to Monte Carlo sampling by V_2. If this component of variance accounts for only a small portion of the total variance, we can conclude that the Monte Carlo variation is negligible.

In our last example to be discussed in section 7, we perform 5 independent runs of algorithms 2 and 3 with $M=20000$ as determined at the end of Section 2. There are almost no between-run variation in the parameter estimates which convince us further that $M=20000$ is sufficient.

5. A BETA-BINOMIAL EXAMPLE

The beta-binomial distribution is commonly used to model clustered binary data to account for over dispersion and intra-cluster correlation. To cast the problem into the framework of this paper, we define the complete data as (n_i, y_i, z_i), $i=1,2,\ldots,m$ and the observed data as (n_i, y_i), $i=1,2,\ldots,m$. The latent variables $z_1, z_2, \ldots, z_m$ are assumed to be independently distributed according to a *beta*(α,β) distribution and given the z_i, the observed data y_i are independently distributed as *binomial* (n_i, z_i). Since the beta distribution is the conjugate prior distribution of the binomial distribution, we can write down the likelihood function in closed form as

$$lik(\alpha,\beta;y_1,y_2,\ldots,y_m)=\prod_{i=1}^{m}\left\{\binom{n_i}{y_i}\frac{\prod_{j=1}^{y_i}(\alpha+j-1)\prod_{j=1}^{n_i-y_i}(\beta+j-1)}{\prod_{j=1}^{n_i}(\alpha+\beta+j-1)}\right\}.$$

The first and second derivatives of the log likelihood function with respect to α and β can be obtained analytically and are given in Kleinman (1973). It follows that the Newton-Raphson procedure (2) can be implemented analytically (see Smith (1983) for a set of fortran codes) without the need to resort to Monte Carlo methods.

The particular data set that we are going to use in the following illustration is the treatment group data reported in Weil (1970). Using the moment estimates as the initial estimates, we carry out the Newton-Raphson procedure analytically which yields the maximum likelihood estimates $\hat{\alpha}=1.591$ (standard error $=0.894$), $\hat{\beta}=0.559$ ($s.e.=0.267$). The maximised log likelihood is -64.990. When reparametrized to $\mu=\alpha/(\alpha+\beta), \gamma=1/(\alpha+\beta)$, the estimates are

$\hat{\mu}=0.740$ (*s.e.* $=0.069$) and $\hat{\gamma}=0.465$ (*s.e.* $=0.241$) which agree with the estimates obtained by Williams (1975) to three diecimal places. A summary of the Newton-Raphson iterations is given in Table I.

Even though the Monte Carlo Newton-Rapshon and the Monte Carlo EM algorithm are really not necessary for this problem, we will carry them out anyway in order to gain insights into the way they work and to see how well they approximate the "true" answer. To obtain the Monte Carlo approximations (3) and (4), we need to simulate from the conditional distribution of $\boldsymbol{z}$ given $\boldsymbol{y}$. This can be done easily as the conditional distribution of z_i given y_i is $beta(\alpha+y_i, \beta+n_i-y_i)$. Based on $M=1000$, the Monte Carlo Newton-Raphson iterations are given in Table II. We stop at step 8 because the $\alpha^{(k)}$ and $\beta^{(k)}$ have stabilised to 2 decimal places and for the first time the 95% confidence interval (C. I.) for both gradients contain 0. Thus using

TABLE I A summary of the analytic Newton-Raphson iterations for fitting a beta-binomial distribution to the data of Weil (1970)

k	$\alpha^{(k)}$	$\beta^{(k)}$	$\partial l/\partial\alpha$	$\partial l/\partial\beta$	l
0	1.225	0.361	−0.643	7.085	−65.457
1	1.392	0.469	−0.117	1.950	−65.060
2	1.544	0.539	−0.014	0.337	−64.993
3	1.588	0.558	−0.000	0.017	−64.990
4	1.591	0.559	−0.000	0.000	−64.990
5	1.591	0.559	0.000	0.000	−64.990
standard error	0.894	0.267			

TABLE II A summary of the Monte Carlo Newton-Raphson iterations for fitting a beta-binomial distribution to the data of Weil (1970)

			Monte Carlo gradient		*95% C.I.*	
k	$\alpha^{(k)}$	$\beta^{(k)}$	$\partial l/\partial\alpha$	$\partial l/\partial\beta$	$\partial l/\partial\alpha$	$\partial l/\partial\beta$
0	1.225	0.361	−0.653	7.519	(−0.74,−0.57)	(6.78, 7.54)
1	1.361	0.461	−0.082	2.201	(−0.16,−0.01)	(1.86, 2.54)
2	1.549	0.543	−0.017	0.410	(−0.09, 0.06)	(0.12, 0.70)
3	1.605	0.567	0.099	−0.026	(0.03, 0.17)	(−0.31, 0.26)
4	1.672	0.581	−0.150	0.026	(−0.22, −0.08)	(−0.25, 0.30)
5	1.535	0.550	0.157	−0.015	(0.09, 0.23)	(−0.29, 0.26)
6	1.631	0.571	−0.102	0.209	(−0.17, −0.03)	(−0.08, 0.49)
7	1.582	0.564	0.089	−0.368	(0.02, 0.16)	(−0.65, −0.09)
8	1.583	0.555	−0.035	−0.021	(−0.11, 0.04)	(−0.28, 0.32)
standard error	0.906	0.276				

Monte Carlo Newton-Raphson algorithm 1 with $M = 1000$, we obtain $\hat{\alpha} = 1.583$ and $\hat{\beta} = 0.555$ as opposed to the true answer of 1.591 and 0.559. The estimated standard errors obtained by inverting l''_M are 0.906 and 0.276 respectively. Thus the Monte Carlo Newton-Raphson procedure seems to work quite well for this data set. After 8 steps of the Monte Carlo EM procedure, we obtain almost the same answer $\hat{\alpha} = 1.578(s.e. = 0.872)$ and $\hat{\beta} = 0.553(s.e. = 0.267)$. However the computing time for the Monte Carlo EM is about five times that of the Monte Carlo Newton-Raphson procedure as it requires iterations for each M-step.

6. AN EXAMPLE WITH GROUPED SURVIVAL DATA

Consider the proportional hazards model

$$h(t; \boldsymbol{x}) = e^{\beta x} h_0(t),$$

where $h(t; \boldsymbol{x})$ denote the hazard function given $\boldsymbol{x}$, a vector of covariates; β a vector of parameters of matching dimension and $h_0(t)$ an arbitrary baseline function. To estimate β based on an uncensored sample $(t_1, t_2, \ldots, t_n)$ with no ties, it is customary to eliminate the baseline hazard function $h_0(t)$ as nuisance parameters by defining the partial likelihood which is just the likelihood based on the rank vector $\boldsymbol{r} = (r_1, r_2, \ldots, r_n)$,

$$lik(\beta; \boldsymbol{r}) = \prod_{i=1}^{n} \frac{e^{\beta x_i}}{\sum_{j: r_j \geq r_i} e^{\beta x_j}}. \tag{10}$$

When the failure times are grouped into L intervals, we only observe $\boldsymbol{D} = (D_1, D_2, \ldots, D_L)$, where D_l with cardinality d_l denotes the set of subjects with failure times in the lth interval. If follows that the rank vector $\boldsymbol{r}$ cannot be determined completely and the appropriate partial likelihood for such grouped data is

$$lik(\beta; \boldsymbol{D}) = \sum_{\boldsymbol{r} \varepsilon S(\boldsymbol{D})} lik(\beta; \boldsymbol{r}), \tag{11}$$

where $S(\boldsymbol{D})$ denotes the set of all rank vectors consistent with the observed $\boldsymbol{D}$. Note that (11) is defined as the sum of $\prod_{l=1}^{L} d_l!$ ordinary partial likelihoods and so unless the d_l are all very small. The maximisation of (11) is a formidable task. Sinha, Tanner and Hall (1994) proposed the use of Monte Carlo EM algorithm to find the maximum likelihood estimate of β treating the rank vector $\boldsymbol{r}$ as the complete data and $\boldsymbol{D}$ as the observed data. The rank vector $\boldsymbol{r}$ can be partitioned according to $\boldsymbol{D}$ into L sub-vectors $\boldsymbol{r}^{(1)},\boldsymbol{r}^{(2)},\ldots,\boldsymbol{r}^{(L)}$, where $\boldsymbol{r}^{(l)}$ consists of the ranks r_i for those subjects $i \varepsilon D_l$. Sinha, Tanner and hall (1994) showed that given $\boldsymbol{D}$, the sub-vectors $\boldsymbol{r}^{(1)},\boldsymbol{r}^{(2)},\ldots,\boldsymbol{r}^{(L)}$ are mutually independent and

$$Pr(\boldsymbol{r}^{(l)}|\boldsymbol{D}) \propto \prod_{i \varepsilon D_l} \frac{e^{\beta x_i}}{\sum_{j \varepsilon D_l, r_j \geq r_i} e^{\beta x_j} + \sum_{j \varepsilon R_{l+1}} e^{\beta x_j}}, \tag{12}$$

where $R_{l+1} = D_{l+1} \cup D_{l+2} \cup \ldots \cup D_L$. By using (12), we can simulate $\boldsymbol{r}_1,\boldsymbol{r}_2,\ldots,\boldsymbol{r}_M$ from the conditional distribution of $\boldsymbol{r}$ given $\boldsymbol{D}$ and the current $\beta^{(k)}$. We are now in a position to implement the Monte Carlo Newton-Raphson algorithm 1 by substituting into (3), (4) and (5), $\theta = \beta$, $\boldsymbol{y} = \boldsymbol{D}$, $z_i = \boldsymbol{r}_i$ and $l(\theta^{(k)};\ y,\ z_i) = l(\beta^{(k)}; \boldsymbol{D}, \boldsymbol{r}_i) = l(\beta^{(k)}; \boldsymbol{r}_i)$, where $l(\beta; \boldsymbol{r})$ is the logarithm of the complete data likelihood (10).

For the purpose of illustration, we use the same data set that Sinha, Tanner and Hall (1994) used in Section 5 of their paper. The data consists of 30 failure times grouped into 10 groups of 3 each and there is a single binary covariate x. Sinha, Tanner and Hall (1994) used the Monte Carlo EM algorithm to obtain the maximum likelihood estimate and claimed that it is not feasible to maximise the analytic likelihood (11) directly as it is a sum of $(3!)^{10} = 6^{10}$ partial likelihoods. Actually, the fact that x is binary allows us to reduce the number of terms in (11) drastically. If there are m 1's and $(d-m)$ 0's in a group; we only need to consider the ${}_dC_m$ ways of arranging the 0's and 1's rather than all $d!$ permutations. For the present data set, all 10 groups are of size 3 and among them 1 group has no 1's, 5 groups have one 1, 2 groups have two 1's and 2 groups have three 1's. It follows that we can reduce (11) to a sum of $({}_3C_0)^1({}_3C_1)^5({}_3C_2)^2({}_3C_3)^2 = 3^7 = 2187$ terms which is a lot more manageable. We have actually maximised (11) directly using Newton Raphson method to obtain $\hat{\beta} = 0.4754$ (*s.e.* $= 0.394$) as the "true" answer which the Monte Carlo methods seek to approximate. Sinha, Tanner and Hall (1994) obtained $\hat{\beta} = 0.329$ using Breslow's approximation and $\hat{\beta} = 0.385$ using the

Monte Carlo EM algorithm with $M = 500$. Both estimates are too small. Using Monte Carlo Newton-Raphson algorithm 1 with $M = 500$, we obtain $\hat{\beta} = 0.4752$ ($s.e. = 0.394$) which agrees with the true answer to 3 decimal places. For the sake of comparison, we also consider the joint estimation of β together with the baseline probabilities for the 10 time-intervals. This is equivalent to fitting an ordinal data regression model using the complementary log-log link. Using the *SAS* package, we obtain $\hat{\beta} = 0.484$ ($s.e. = 0.392$). For grouped survival data, there is some debate as to whether we should eliminate the baseline parameters from the likelihood function as in (11) or whether we should estimate the regression parameters and the baseline parameters simultaneously. For the present data set, it is reassuring to know that the two approaches give similar answers.

7. POLIO INCIDENCE DATA

As our final example, we consider the data reported by Zeger (1988) concerning the monthly number of cases of poliomyelitis reported by the U.S. Centers for Disease Control for the years 1970 to 1983. We index the $n = 168$ observations by t. To study time trend and seasonality, Zeger (1988) introduced the covariate vector $\boldsymbol{x}_t = (1, \frac{t}{1000}, \cos(\frac{2\pi t}{12}), \sin(\frac{2\pi t}{12}), \cos(\frac{2\pi t}{6}), \sin(\frac{2\pi t}{6}))^T$. To account for over dispersion and autocorrelation, Chan and ledolter (1995) proposed the following latent process model. It is assumed that there is an unobserved Gaussian $AR(1)$ process $\{z_t\}$ staisfying $z_t = \rho z_{t-1} + \varepsilon_t$ where the ε_t are i.i.d $N(0,\sigma^2)$. Given the latent process $\{z_t\}$, the observations y_t are independently Poisson distributed with mean λ_t satisfying

$$\log \lambda_t = \beta \boldsymbol{x}_t + z_t,$$

where β is a row vector of regression parameters. As noted by Chan and Ledolter (1995), the likelihood of the observed data does not have a simple closed form and maximum likelihood estimation is intractable. To implement the Monte Carlo EM or the Monte Carlo Newton-Raphson procedure, we need to simulate $z_1, z_2, \ldots, z_M$ from the conditional distribution of z given y and $\theta = (\beta, \rho, \sigma^2)$. With fourteen years of monthly data, z is a 168×1 vector and it is extremely difficult to simulate directly from their joint conditional distribution. To overcome this problem, Chan and Ledolter (1995) suggest using Gibbs sampling which is a Markov chain Monte Carlo method. Beginning with some initial values $z^{(0)} = (z_1^{(0)}, z_2^{(0)}, \ldots, z_n^{(0)})$, we proceed to generate $z^{(b)} = (z_1^{(b)}, z_2^{(b)}, \ldots, z_n^{(b)}), b = 1, 2, \ldots$ sequentially in the following manner. Given $z^{(b)}$ and the current estimate $\theta^{(k)}$, we simulate

$$z_1^{(b+1)} \text{ from } f(z_1 | z_2^{(b)}, z_3^{(b)}, \ldots, z_n^{(b)}, y; \theta^{(k)}),$$

$$\vdots$$

$$z_t^{(b+1)} \text{ from } f(z_t | z_1^{(b+1)}, z_2^{(b+1)}, \ldots, z_{t-1}^{(b+1)}, z_{t+1}^{(b)}, \ldots, z_n^{(b)}, y; \theta^{(k)}),$$

$$\vdots$$

$$z_n^{(b+1)} \text{ from } f(z_n | z_1^{(b+1)}, z_2^{(b+1)}, \ldots, z_{n-1}^{(b+1)}, y; \theta^{(k)}).$$

It can be shown that $\{z^{(b)}\}$ is a Markov chain and the conditional distribution of z given y is the stationary distribution of this Markov chain. In our analysis, we discard the first $T = 200$ z's of the Markov chain as transient values and treat the following $M = 20000$ z's (see section 2 for the rationale behind this choice) as realizations from the conditional distribution of z given y which can then be substituted into Q_M for the implementation of the Monte Carlo EM algorithm and into (3) and (4) for the Monte Carlo Newton-Raphson procedure. The operational details for simulating z_t from the conditional distribution given $z^{-t} = (z_1, z_2, \ldots, z_{t-1}, z_{t+1}, \ldots, z_n)$ and y are described in Chan and Ledolter (1995). Specifically, the conditional density of $z_t + \beta x_t$ given z^{-t} and y is given by equation (3.4) of their paper and to sample from this log concave density, we use the rejective algorithm described in section 2.6 of Devorye (1986).

With the aid of Gibbs sampling as described above, we carry out algorithms 2 and 3 with starting values $\beta_1^{(0)} = 0.557, \beta_2^{(0)} = -4.799$, $\beta_3^{(0)} = 0.137, \beta_4^{(0)} = -0.535, \beta_5^{(0)} = 0.459$ and$\beta_6^{(0)} = -0.0696$ obtained by fitting an ordinary Poisson regression and $\rho^{(0)} = 0, \sigma^{2^{(0)}} = 1.0$. We use (7) with $\alpha = 0.1$ as the convergence criterion. The resulting estimates are reported in Table III together with standard errors obtained by inverting l''_M. Also reported in Table III are the parameter

TABLE III Results of five runs of algorithms 2 and 3 with $M=20000$ for fitting a latent process model to the polio incidence data

Algorithm 2

Run	*No. of iterations*	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\beta}_5$	$\hat{\beta}_6$	$\hat{\rho}$	$\hat{\sigma}^2$
1	65	0.239	−3.81	0.162	−0.479	0.412	−0.0099	0.644	0.289
		*±0.294**	*±2.94*	*±0.147*	*±0.166*	*±0.128*	*±0.128*	*±0.210*	*±0.616*
2	61	0.236	−3.78	0.161	−0.482	0.413	−0.0100	0.641	0.290
3	73	0.237	−3.74	0.162	−0.481	0.413	−0.0104	0.649	0.283
4	65	0.237	−3.79	0.164	−0.482	0.413	−0.0097	0.645	0.287
5	67	0.232	−3.73	0.162	−0.481	0.413	−0.0097	0.647	0.285
Average	66	0.236	−3.77	0.162	−0.481	0.413	−0.0099	0.645	0.287

Algorithm 3

Run	*No. of iterations*	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\beta}_5$	$\hat{\beta}_6$	$\hat{\rho}$	$\hat{\sigma}^2$
1	10	0.243	−3.81	0.161	−0.481	0.413	−0.0108	0.661	0.272
		±0.278	*±2.83*	*±0.145*	*±0.165*	*±0.127*	*±0.125*	*±0.218*	*±0.627*
2	7	0.235	−3.73	0.160	−0.480	0.413	−0.0098	0.660	0.275
3	6	0.237	−3.71	0.161	−0.482	0.414	−0.0098	0.656	0.274
4	9	0.234	−3.66	0.161	−0.480	0.414	−0.0110	0.663	0.270
5	8	0.240	−3.79	0.160	−0.481	0.414	−0.0110	0.669	0.266
Average	8	0.238	−3.74	0.161	−0.481	0.414	−0.0105	0.662	0.271

*The figures in italics are standard errors.

estimates obtained from four additional runs of each algorithm. As there are almost no variation in the estimates between runs, we can confidently use the estimates and standard errors produced from the first run.

Note that the two algorithms give almost the same estimate. However, it takes algorithm 2 66 iterations to converge on the average. The estimated standard errors are also similar. With $M=20000$, each iteration takes roughly 6 minutes of CPU time on a DEC8200 computer. Algorithm 3, on the average, requires 8 iterations including the half-stepping. Thus on the average, there is a saving of 58 iterations or 348 minutes of CPU time.

8. CONCLUSION

We show that the Monte Carlo Newton-Raphson algorithm is a viable alternative to the Monte Carlo EM algorithm. Both Monte Carlo algorithms involve the simulations of z from the conditional distribution of z given the observed data y and the current estimate $\theta^{(k)}$. The EM algorithm converges at a linear rate and generally requires iterations within each M-step unless there exists explicit formula for finding the maximizer at the M-step. The Newton-Raphson procedure converges at a quadratic rate and is computationally more efficient. A naive Monte Carlo implementaion of the Newton-Raphson procedures (algorithm 1) is, however, quite erratic and so we refine it to algorithm 3. We also suggest a stopping criterion based on a chi-square test for zero gradient and demonstrate the calculation of type II error with appliaction to the determination of the number of Monte Carlo replications required. A major appliaction of the Monte Carlo Newton-Raphson algorithm is in the fitting of generalized linear models with random effects (Karim and Zeger, 1992; Kuk, 1995) to clustered binary or count data and the so called random frailty models (Clayton, 1991) for survival data. The likelihood functions for some of these models involve integrals of dimension as high as 20 or even 40 (Karim and Zeger, 1992). For such high-dimensional integrals, numerical integration is no longer feasible or reliable and Monte Carlo methods are more appealing.

Acknowledgment

The authors wish to thank the referee for helpful comments and Dr. K.S. Chan for providing a preprint of his joint paper with Dr. J. Ledolter. This work was partially supported by the Australian Research Council.

References

Chan, K. S. and Ledolter, J. (1995) Monte Carlo EM estimation for time series models involving counts. *J. Amer. Statist. Assoc.*, **90**, 242–252.

Clayton, D. G. (1992) A Monte Carlo method for Bayesian inference in frailty models. *Biometrics*, **47**, 467–485.

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

Devorye, L. (1986) *Non-uniform Random Variate Generation*. New York : Springer.

Geyer, C. J. (1992) Practical Markov Chain Monte Carlo. *Statistical Science*, **7**, 473–511.

Karim, M. R. and Zeger, S. L. (1992) Generalized linear models with random effects; salamander mating revisited. *Biometrics*, **48**, 631–644.

Kleinman, J. C. (1973) Proportions with extraneous variance : single and independent samples. *J. Amer. Statist. Assoc.*, **68**, 46 – 54.

Kuk, A. Y. C. (1995) Asymptotically unbiased estimation in generalized linear models with random effects. *J. R. Statist. Soc. B*, **57**, 395 – 407.

Kuk, A. Y. C. and Chen, C. H. (1992) A mixture model combining logistic regression with proportional hazards regression. *Biometrika*, **79**, 531 – 541.

Lange, K. (1995) A gradient algorithm locally equivalent to the EM algorithm. *J. R. Statist. Soc. B*, **57**, 425 – 437.

Louis, T. A. (1982) Finding the observed information matrix when using the EM algorithm. *J. R. Statist. Soc. B*, **44**, 226 – 233.

Sinha, D., Tanner, M. A. and Hall, W. J. (1994) Maximization of the marginal likelihood of grouped survival data. *Biometrika*, **81**, 53 – 60.

Smith, D. M. (1983) Maximum likelihood estimation of the parameters of the beta binomial distribution. *Applied Statist.*, **32**, 196 – 204.

Wei, G. C. G. and Tanner, M. A. (1990) A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *J. Amer. Statist. Assoc.*, **85**, 699 – 704.

Weil, C. S. (1970) Selection of the valid number of sampling units and consideration of their combination in toxicological studies involving reproduction, teratogenesis or carcinogenesis. *Food and Cosmetic Toxicology*, **8**, 177 – 182.

Williams, D. A. (1975) The analysis of binary responses from toxicological experiments involving reproduction and teratogenicity. *Biometrics*, **31**, 949 – 952.

Zeger, S. L. (1988) A regression model for time series of counts. *Biometrika*, **75**, 621 – 629.