

The Number of Iterations, Convergence Diagnostics and Generic Metropolis Algorithms

Adrian E. Raftery
University of Washington

Steven M. Lewis
University of Washington

1 Introduction

In order to use Markov chain Monte Carlo, MCMC, it is necessary to determine how long the simulation needs to be run. It is also a good idea to discard a number of initial “burn-in” simulations, since from an arbitrary starting point it would be unlikely that the initial simulations came from the stationary distribution intended for the Markov chain. Also, consecutive simulations from Markov chains are dependent, sometimes highly so. Since saving all simulations can require a large amount of storage, researchers using MCMC sometimes prefer saving only every third, fifth, tenth, etc. simulation, especially if the chain is highly dependent. This is sometimes referred to as *thinning* the chain. While neither burn-in nor thinning are mandatory practices, they both reduce the amount of data saved from a MCMC run.

In this chapter, we outline a way of determining in advance the number of iterations needed for a given level of precision in a MCMC algorithm. This is introduced in Section 2, and in Section 3 we describe the `gibbsit` software which implements it and is available free of charge from StatLib. In Section 4 we show how the output from this method can also be used to diagnose lack of convergence or slow convergence due to bad starting values, high posterior correlations, or “stickiness” of the chain. In Section 5 we describe how the methods can be combined with ideas of Müller (1991) and Gelman (1993) to yield an automatic generic Metropolis algorithm.

For simplicity, the discussion is in the context of a single long chain. However, the same basic ideas can also be used to determine the number of iterations and diagnose slow convergence when multiple sequences are used, as advocated by Gelman and Rubin (1992b); see Section 6.

2 Determining The Number of Iterations

In any practical application of MCMC, there are a number of important decisions to be made. These include the number of iterations, the spacing between iterations retained for the final analysis, and the number of initial burn-in iterations discarded. A simple way of

making these decisions was proposed by Raftery and Banfield (1991) and Raftery and Lewis (1992a).

To use MCMC for Bayesian inference, the sample generated during a run of the algorithm should “adequately” represent the posterior distribution of interest. Often interest focuses on posterior quantiles of functions of the parameters, such as Bayesian confidence intervals and posterior medians, and then the main requirement is that MCMC estimates of such quantities be approximately correct.

We thus consider probability statements regarding quantiles of the posterior distribution of a function U of the parameter θ . A quantile is the same thing as a percentile except that it is expressed in fractions rather than in percents. Suppose we want to estimate $P[U \leq u \mid \text{Data}]$ to within $\pm r$ with probability s , where U is a function of θ . We will find the approximate number of iterations required to do this when the actual quantile of interest is q . For example, if $q = 0.025$, $r = 0.0125$ and $s = 0.95$, this corresponds to requiring that the cumulative distribution function of the 0.025 quantile be estimated to within ± 0.0125 with probability 0.95. This might be a reasonable requirement if, roughly speaking, we wanted reported 95% intervals to have actual posterior probability between 0.925 and 0.975. We run the MCMC algorithm for an initial M iterations that we discard, and then for a further N iterations of which we store every k^{th} . Our problem is to determine M , N , and k .

We first calculate U_t for each iteration t where U_t is the value of U for the t^{th} iteration, and then form

$$Z_t = \begin{cases} 1 & \text{if } U_t \leq u \\ 0 & \text{otherwise} \end{cases} .$$

$\{Z_t\}$ is a binary 0-1 process that is derived from a Markov chain, but is not itself a Markov chain. Nevertheless, it seems reasonable to suppose that the dependence in $\{Z_t\}$ falls off fairly rapidly with lag, and hence that if we form the new process $\{Z_t^{(k)}\}$, where $Z_t^{(k)} = Z_{1+(t-1)k}$, consisting of every k^{th} iteration from the original chain, then $\{Z_t^{(k)}\}$ will be approximately a Markov chain for k sufficiently large.

To determine k , we form the series $\{Z_t^{(k)}\}$ for $k = 1, 2, \dots$. For each k , we compare the first-order Markov chain model with the second-order Markov chain model, and choose the smallest value of k for which the first-order model is preferred. We compare the models by calculating G^2 , the likelihood ratio test statistic between the second-order Markov model and the first-order Markov model, and then use the BIC criterion, $G^2 - 2 \log n$, where n is the number of iterations in a pilot sample, to see which of the two models better fits the pilot sample. BIC was introduced by Schwarz (1978) in another context and generalized to log-linear models by Raftery (1986); it provides an approximation to twice the logarithm of the Bayes factor for the second-order model.

Next, assuming that $\{Z_t^{(k)}\}$ is indeed a Markov chain, we now determine $M = mk$, the number of burn-in iterations to be discarded. In what follows we use standard results for two-state Markov chains; see, e.g. Cox and Miller (1965). Let

$$\mathbf{P} = \begin{pmatrix} 1 - \alpha & \alpha \\ \beta & 1 - \beta \end{pmatrix}$$

be the transition matrix for $\{Z_t^{(k)}\}$, where α is the probability of changing from the first state to the second state and β is the probability of changing from the second state to the first state. The equilibrium distribution is then $\boldsymbol{\pi} = (\pi_0, \pi_1) = (\alpha + \beta)^{-1}(\beta, \alpha)$, where $\pi_0 = P[U \leq u \mid \text{Data}]$ and $\pi_1 = 1 - \pi_0$, and the ℓ -step transition matrix is

$$\mathbf{P}^\ell = \begin{pmatrix} \pi_0 & \pi_1 \\ \pi_0 & \pi_1 \end{pmatrix} + \frac{\lambda^\ell}{\alpha + \beta} \begin{pmatrix} \alpha & -\alpha \\ -\beta & \beta \end{pmatrix},$$

where $\lambda = (1 - \alpha - \beta)$. Suppose that we require that $P[Z_m^{(k)} = i \mid Z_0^{(k)} = j]$ be within ε of π_i for $i, j = 0, 1$. If $e_0 = (1, 0)$ and $e_1 = (0, 1)$, then $P[Z_m^{(k)} = i \mid Z_0^{(k)} = j] = e_j \mathbf{P}^m e_i^T$, and so the requirement becomes

$$\lambda^m \leq \frac{(\alpha + \beta) \varepsilon}{\max(\alpha, \beta)},$$

which holds when

$$m = m^* = \frac{\log \left\{ \frac{(\alpha + \beta) \varepsilon}{\max(\alpha, \beta)} \right\}}{\log \lambda},$$

assuming that $\lambda > 0$, which is usually the case in practice. Thus

$$M = m^* k.$$

To determine N , we note that the estimate of $P[U \leq u \mid \text{Data}]$ is $\bar{Z}_n^{(k)} = \frac{1}{n} \sum_{t=1}^n Z_t^{(k)}$. For n large, $\bar{Z}_n^{(k)}$ is approximately normally distributed with mean q and variance

$$\frac{1}{n} \frac{(2 - \alpha - \beta) \alpha \beta}{(\alpha + \beta)^3}.$$

Thus the requirement that $P[q - r \leq \bar{Z}_n^{(k)} \leq q + r] = s$ will be satisfied if

$$n = n^* = \frac{(2 - \alpha - \beta) \alpha \beta}{(\alpha + \beta)^3} \left\{ \frac{\Phi^{-1} \left(\frac{1}{2} (s + 1) \right)}{r} \right\}^2,$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function. Thus we have

$$N = n^* k.$$

Use of a thinned chain with only N simulations to perform the final inference results in less precision than if no thinning had been done. Thus if our value of N is used without thinning (i.e. if all the iterations are used), then the probability of satisfying our accuracy criterion is greater than required, and so our criterion is conservative in this case.

Table 1: Maximum percent error in the estimated 0.025 quantile.

r	N_{\min} ($s=0.95$)	Percent error		
		N(0,1)	t_4	Cauchy
0.0025	14982	2	4	11
0.0050	3748	5	8	25
0.0075	1665	8	13	43
0.0100	936	11	19	67
0.0125	600	14	26	101
0.0150	416	19	37	150
0.0200	234	31	65	402

The method requires that the MCMC algorithm be run for an initial number of iterations in order to get a pilot sample of parameter values. As a rough guide to the size of this pilot sample, we note that the required N will be minimized if successive values of $\{Z_i\}$ are independent, implying that $\alpha = 1 - \beta = \pi_1 = 1 - q$, in which case $M = 0$, $k = 1$ and

$$N = N_{\min} = \Phi^{-1} \left(\frac{1}{2}(s+1) \right)^2 q(1-q)/r^2.$$

The user needs to give only the required precision, as specified by the four quantities q , r , s and ε . Of these, the result is by far the most sensitive to r , since $N \propto r^{-2}$. For example, when $q = 0.025$, $r = 0.0125$ and $s = 0.95$, we have $N_{\min} = 600$. N_{\min} is in general a reasonable choice for the size of the pilot sample.

Instead of specifying the required precision in terms of the error in the cumulative distribution function at the quantile, which is what r refers to, it may be more natural to specify the required precision in terms of the error in an estimate of the quantile itself. In order to see how r relates to accuracy on the latter scale, we have shown in Table 1 the approximate maximum percentage error in the estimated quantile, for $q = 0.025$, corresponding to selected values of r . This is defined as

$$100 \max \left\{ \frac{F^{-1}(q \pm r)}{F^{-1}(q)} - 1 \right\},$$

and is shown for three distributions: normal (light-tailed), t_4 (moderate tails), and Cauchy (heavy-tailed).

Suppose we regard a 14% relative error on the scale of a standardized variable, i.e. z -score, as acceptable, corresponding to an estimated 0.025 quantile between -2.24 and -1.68 in the normal distribution, compared with the true value of -1.96 . Then, if we knew $P[U | \text{Data}]$ to have light, normal-like, tails, Table 1 suggests that $r = 0.0125$ would be sufficiently small. However, with the heavier-tailed t_4 distribution, $r = 0.0075$ is required to achieve the same accuracy, while for the very heavy-tailed Cauchy, $r = 0.003$ is required, corresponding to $N_{\min} \approx 10,000$.

3 Software and Implementation

The method is implemented in the Fortran program `gibbsit`, which can be obtained free of charge by sending the e-mail message “send gibbsit from general” to `statlib@stat.cmu.edu`. Despite the name, `gibbsit` can be used for any MCMC, not just the Gibbs sampler. The program takes as input the pilot MCMC sample for the quantity of interest and the values of q , r and s , and it returns as output the estimated values of M , N and k .

We recommend that `gibbsit` be run a second time after the $(M + N)$ iterations have been produced, to check that the N iterations recommended on the basis of the pilot sample were indeed adequate. If not, i.e. if the value of $(M + N)$ from the second call to `gibbsit` is appreciably more than that from the first call, then the MCMC algorithm should be continued until the total number of iterations is adequate.

In any practical setting, several quantities will be of interest, and perhaps several quantiles of each of these. We recommend that `gibbsit` be called for each quantile of primary interest, and that the maximum values of M and N be used. Typically, tail quantiles are harder to estimate than central quantiles such as medians, so one reasonable routine practice would be to apply `gibbsit` to each quantity of primary interest twice, with $q = 0.025$ and $q = 0.975$.

4 Convergence Diagnostics

In this section we assume that the `gibbsit` program has been run. As output from this program we will have values for M , N and k . It would be possible to just take these parameters for a final run of the MCMC, but this would not be recommended. Performing a little output analysis on these values may indicate ways in which the MCMC algorithm could be improved. Graphical examination of the initial N_{\min} iterations can also suggest ways to improve the algorithm.

The `gibbsit` program outputs can be used for diagnostic purposes. These outputs can be combined to calculate

$$I = \frac{M + N}{N_{\min}}$$

(Raftery and Lewis, 1992b). This statistic measures the increase in the number of iterations due to dependence in the sequence. Values of I much greater than 1 indicate a high level of dependence. We have found that values of I greater than about 5 often indicate problems that can be alleviated by changing the implementation. Such dependence can be due to a bad starting value (in which case other starting values should be tried), to high posterior correlations (which can be remedied by crude correlation-removing transformations), or to “stickiness” in the Markov chain (sometimes removable by changing the MCMC algorithm). It may seem surprising that a bad starting value can lead to high values of N as well as M . This happens because progress away from a bad starting value tends to be slow and gradual, leading to a highly autocorrelated sequence and high values of N , since the entire pilot sequence (including the initial values) is used to estimate N , as well as M and k .

It is important to examine iteration-sequenced plots of the initial MCMC-generated posterior sample for at least the key parameters of the model. For example, in hierarchical models, it is important to look at a plot of the random effects variance or equivalent parameter. If the starting value for this parameter is too close to zero, componentwise MCMC (such as the Gibbs sampler) can get stuck for a long time close to the starting value. A plot of the simulated values of this parameter will show whether or not the algorithm remained stuck near the starting value. This problem arises in the example which follows.

Example We illustrate these ideas with an example from the analysis of longitudinal World Fertility Survey data (Raftery, Lewis, Aghajanian and Kahn, 1993; Lewis, 1993). The data are complete birth histories for about 5,000 Iranian women, and here we focus on the estimation of unobserved heterogeneity. Let π_{it} be the probability that woman i had a child in calendar year t . Then a simplified version of the model used is

$$\begin{aligned} \log(\pi_{it}/(1-\pi_{it})) &= \eta + \delta_i \\ \delta_i &\stackrel{\text{iid}}{\sim} N(0, \Sigma). \end{aligned} \tag{1}$$

The prior on η is Gaussian and the prior on Σ is inverted gamma with a shape parameter of 0.5 and a scale parameter of 0.02 (Lewis, 1994). The δ_i 's are random effects representing unobserved sources of heterogeneity in fertility such as fecundability and coital frequency. There are also measured covariates in the model, but these are omitted here for ease of exposition.

Figure 1 shows a run of a MCMC algorithm starting with a value of Σ close to zero, namely $\Sigma = 10^{-4}$, and with values of the δ_i 's randomly generated from $N(0, 10^{-4})$. (In Figures 1 and 2, the starting value has been omitted for reasons of scaling.) The Σ series seems highly autocorrelated and a run of the `gibbsit` program confirms this. With $q = 0.025$, $r = 0.0125$ and $s = 0.95$, we obtain $N = 4,258$, $k = 2$ and $M = 42$. Here $N_{\min} = 600$, so that $I = 7.2$. The high value of I and the trend-like appearance of Figure 1(a) suggest that there is a starting value problem. By contrast, the values of δ_9 *in the same run* are much less correlated (Figure 1(b)) with $I = 2.8$, so that diagnostics based on that series alone would mislead.

Figure 2 shows three other series of Σ from different starting values, illustrating a simple trial-and-error approach to the choice of an adequate starting value. Figure 2(a) starts with $\Sigma = 0.1$, and the method of Raftery and Lewis (1992b) yields $I = 3.3$. Figure 2(b) starts with $\Sigma = 1.0$ and has $I = 2.8$. Figure 2(c) starts with $\Sigma = 0.25$ and has $I = 2.4$. The results of these trajectories all seem satisfactory, suggesting that the results are relatively insensitive to the starting value when our diagnostics do not indicate there to be a problem.

This example bears out our main points. It is important to monitor the MCMC run for all the key parameters, and to start again with different starting values when the diagnostics suggest doing this.

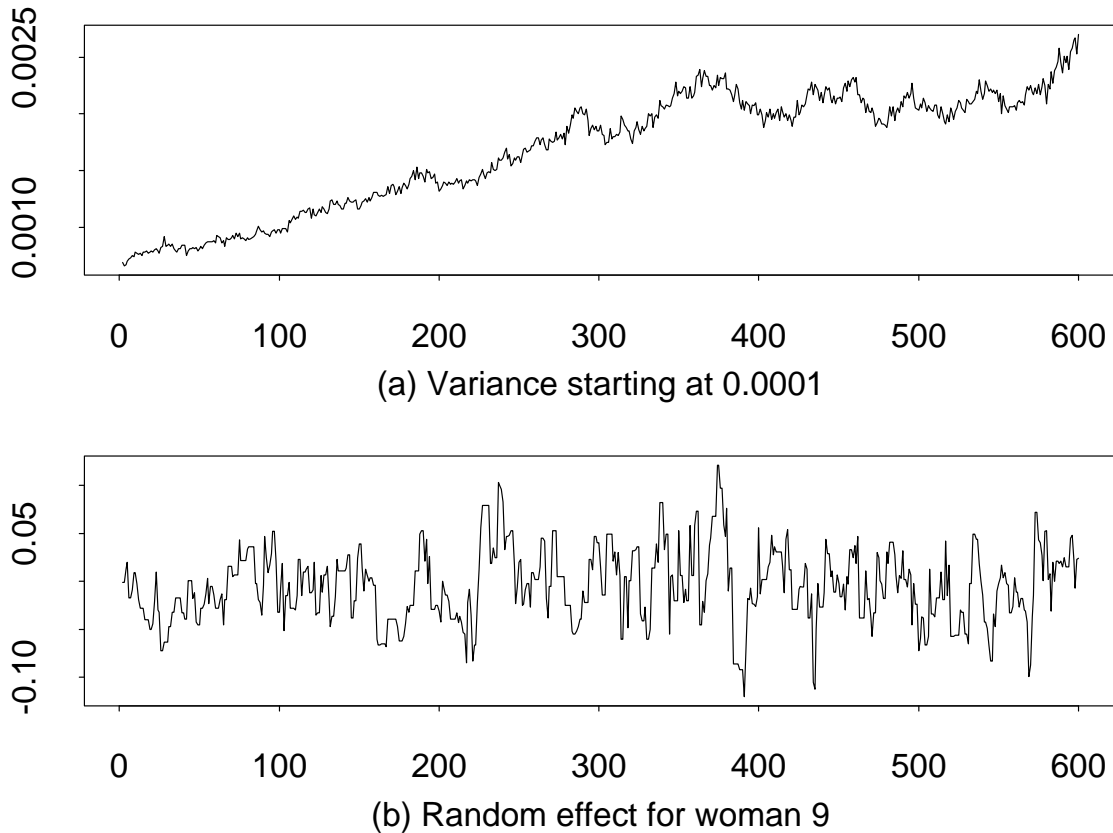


Figure 1: MCMC output for the model in equation (1) for the Iranian World Fertility Survey data starting with $\Sigma = 10^{-4}$: (a) series of Σ values; (b) series of values of δ_9 , the random effect for woman 9 in the survey.

5 Generic Metropolis Algorithms

In this section we develop a generic Metropolis (1953) algorithm which could be fully automated, although we have not yet done so. This combines the methods of Sections 2 and 4 with ideas of Müller (1991) and Gelman (1993). In the basic algorithm, one parameter is updated at a time, and the proposal distribution is normal centered at the current value. The user only has to specify the variance of the proposal distribution for each parameter, and here we outline a strategy for doing this. This version of the Metropolis algorithm is sometimes referred to as “Metropolis within Gibbs” because one parameter at a time is being updated. We prefer not using this name since this form of Metropolis algorithm is not a Gibbs sampler. The latter name is used when sampling from the full conditional distributions, which is not required for this generic algorithm.

The strategy consists of three applications of MCMC. We use θ to denote the vector of parameters and denote the variance of the proposal function for the j^{th} component of θ by σ_j^2 ; these latter values are set by the user. The performance of a Metropolis algorithm

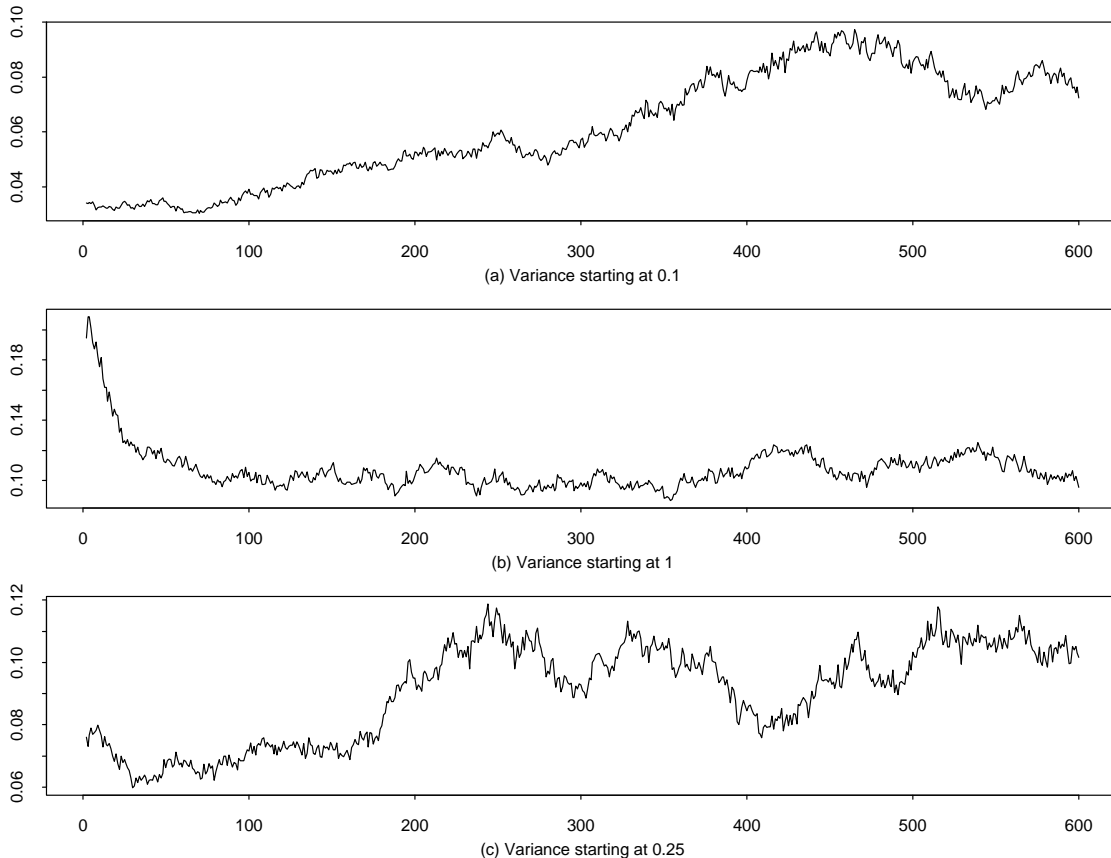


Figure 2: Values of Σ for three runs of the same MCMC algorithm as in Figure 1, with different starting values: (a) $\Sigma = 0.1$; (b) $\Sigma = 1$; (c) $\Sigma = 0.25$.

can be very sensitive to the (user-specified) σ_j^2 : if they are too large, the chain will almost never move from the current state, while if they are too small, the chain will move frequently but slowly. In either situation it will take a long time to get an adequate sample from the posterior distribution.

What values should be used for σ_j ? A number of values have been suggested in the literature. In a more general setting, Tierney (1991) suggested setting the standard deviation of the proposal distribution at a fixed multiple, such as $1/2$ or 1 , times the current estimated variance matrix. Others have suggested using a variable schedule, based in one way or another on how well the chain is behaving, to determine what the value of σ_j should be (Müller, 1991; Clifford, 1994).

Gelman (1993) has studied what the “optimal” variance should be for the case where one is simulating a univariate standard normal distribution using a Metropolis algorithm with a normal distribution centered at the last value as the proposal distribution. He considered two different optimality criteria, the asymptotic efficiency of the Markov chain and the geometric convergence rate of the Markov chain. In either case he found that for the unit

normal distribution the optimal proposal standard deviation is about 2.3. We would then expect that for an arbitrary normal distribution the optimal proposal standard deviation should be roughly 2.3 times the standard deviation of the normal.

Gelman's result is readily incorporated in the generic three-simulation strategy outlined here. The generic three-simulation strategy consists of the following steps:

Simulation 1

- Assign a large value to σ_j , such as 2.3 times the approximate marginal standard deviation of θ_j .
- Run MCMC for N_{\min} scans to obtain a pilot sample.

Simulation 2

- Use the pilot sample to calculate conditional standard deviations of each component of $\boldsymbol{\theta}$ given the sample estimates for the other components. This is done using linear regression.
- Assign $\sigma_j = 2.3 \text{ Sd}(\theta_j | \boldsymbol{\theta}_{-j})$, where $\boldsymbol{\theta}_{-j}$ denotes all of $\boldsymbol{\theta}$ except the j^{th} component.
- Run another MCMC for N_{\min} scans to obtain a second sample. The second sample is necessary to obtain reasonable estimates of the conditional standard deviations.

Simulation 3

- Calculate conditional standard deviations of the parameters from the second sample.
- Reassign $\sigma_j = 2.3 \text{ Sd}(\theta_j | \boldsymbol{\theta}_{-j})$.
- Run the `gibbsit` program to get k , M and N .
- Run MCMC for $(M + N)$ scans.
- Rerun the `gibbsit` program to check that N is big enough.
- If N is big enough, i.e. if the `gibbsit` program says that the number of iterations required is no greater than the number of iterations actually run, make inference using the final sample.
- If not, return to the beginning of Simulation 3.

Example To illustrate the three-simulation strategy, we use it to simulate from the following trivariate normal distribution for $\boldsymbol{\eta} = (\eta_0, \eta_1, \eta_2)^T$:

$$\boldsymbol{\eta} \sim N \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{V} = \begin{bmatrix} 99 & -7 & -7 \\ -7 & 1 & 0 \\ -7 & 0 & 1 \end{bmatrix} \right).$$

The conditional variances are

$$\begin{aligned} V(\eta_0 \mid \eta_1, \eta_2) &= 1 \\ V(\eta_1 \mid \eta_0, \eta_2) &= 1/50 \\ V(\eta_2 \mid \eta_0, \eta_1) &= 1/50. \end{aligned}$$

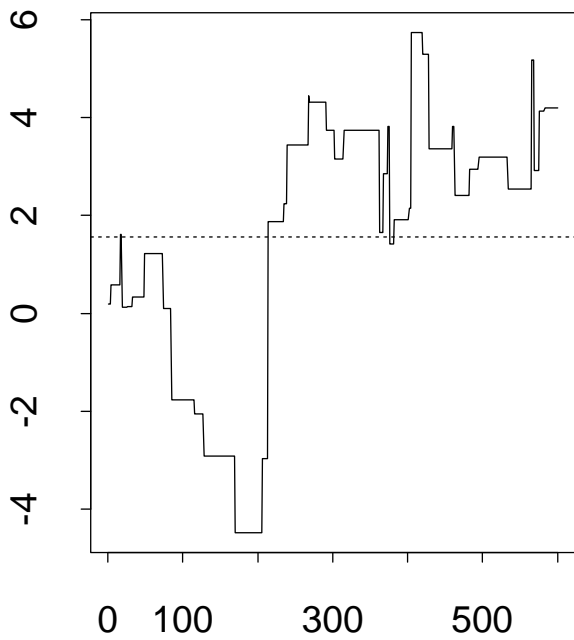
This kind of posterior covariance matrix is common for regression parameters, where η_0 is the intercept and the independent variables corresponding to η_1 and η_2 have non-zero means. It is a challenging example because the conditional variances are many times smaller than the marginal variances (even though there are no very high pairwise correlations).

We start by generating a small pilot sample of N_{\min} draws. In order to determine the size of this pilot sample we must first select values for q , r and s . We take $q = 0.025$, $r = 0.0125$ and $s = 0.95$, which corresponds to requiring that the estimated cumulative distribution function of the 0.025 quantile of η_0 fall within the interval $(0.0125, 0.0375)$ with probability 0.95. This leads to $N_{\min} = 600$. The proposal standard deviations for getting the pilot sample are set to $\sigma_0 = 2.3\sqrt{99} = 22.9$ and $\sigma_1 = \sigma_2 = 2.3\sqrt{1} = 2.3$. Since we know the actual distribution of η_0 is $N(0, 99)$, we can find the true 0.025 quantile for η_0 , $F^{-1}(0.025)$, which is -19.5 .

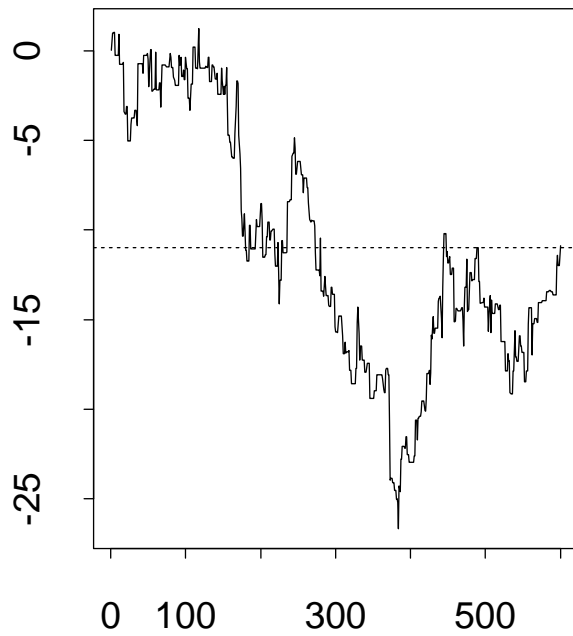
Figure 3 contains iteration-sequenced plots of η_0 . Figure 3(a) shows the 600-iteration pilot sample; this is clearly too few to obtain a good sample from the posterior. Next we calculated the sample covariance matrix for the pilot sample and from this the conditional variance of each parameter given the other two. These are reported in Table 2, which summarizes the results. It took four simulations before the number of iterations indicated by the `gibbsit` program was less than the actual number of iterations run during the current simulation.

The sample covariance matrix for the pilot sample is far from the truth. When we input the pilot sample into the `gibbsit` program it says that M should equal 229 and N should be set to 46,992. Hence $I = (M + N)/N_{\min} = 78.7$, so that the pilot sample is clearly not large enough. We recalculated the proposal standard deviations as $\sigma_0 = 2.3\sqrt{0.92} = 2.21$, $\sigma_1 = 2.3\sqrt{0.020} = 0.33$ and $\sigma_2 = 2.3\sqrt{0.018} = 0.31$. These were used as inputs to the second simulation. We then obtained another sample of $N_{\min} = 600$ draws from the posterior. The results are shown in the second simulation column of Table 2. The second sample covariance matrix was much closer to the true covariance matrix than was the pilot sample covariance matrix. It was not correct, but we were headed in the right direction.

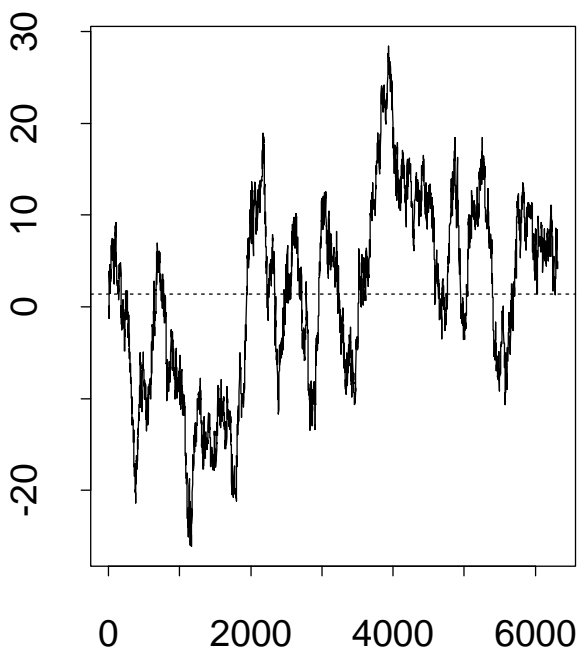
The conditional variances from the second sample were then used to recalculate proposal standard deviations one more time. These are shown as inputs to the third simulation. Also, using the second Metropolis sample as input to the `gibbsit` program produced values for k ,



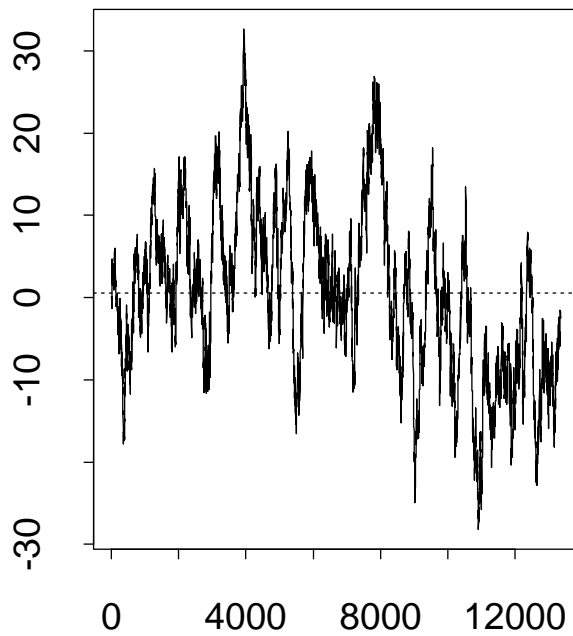
(a) First Simulation



(b) Second Simulation



(c) Third Simulation



(d) Fourth Simulation

Figure 3: Sequence plots of intercept estimate for trivariate normal example.

Table 2: Three-simulation strategy results for the trivariate normal example. The input control parameters for each simulation are reported above the double line. The resulting estimates from each simulation are shown below the line.

Variable	True Value	Simulation			
		First	Second	Third	Fourth
$M + N$		600	600	6,313	13,353
σ_0		22.9	2.2	2.2	2.3
σ_1		2.3	0.33	0.32	0.33
σ_2		2.3	0.31	0.31	0.33
$E(\eta_0)$	0	1.6	-11.0	1.4	0.5
$E(\eta_1)$	0	-0.8	0.9	-0.2	-0.1
$E(\eta_2)$	0	0.6	0.7	0	0
V_{11}	99	7.4	48	106	106
V_{12}	-7	-0.6	-4.7	-6.4	-7.4
V_{13}	-7	-0.4	-2.0	-8.6	-7.6
V_{22}	1	0.2	1.0	1.0	1.1
V_{23}	0	-0.1	-0.4	-0.1	-0.1
V_{33}	1	0.1	0.6	1.3	1.2
$V(\eta_0 \eta_1, \eta_2)$	1	0.9	0.9	1.0	1.0
$V(\eta_1 \eta_0, \eta_2)$	0.02	0.02	0.02	0.02	0.02
$V(\eta_2 \eta_0, \eta_1)$	0.02	0.02	0.02	0.02	0.02
k		1	1	1	1
M		229	57	148	138
N		46,992	6,256	13,205	12,276
$\hat{F}(F^{-1}(0.025))$	0.025		0.082	0.021	0.020

N and M to use for a third run of the Metropolis algorithm. In our case `gibbsit` indicated that k was 1, M should be set to 57 and N should be set at 6,256.

We then used the recalculated proposal standard deviations to perform a third Metropolis run for a total of $(M + N) = 6,313$ iterations. The marginal and conditional sample variances for this third sample were reasonably close to the correct values. However, when we ran the `gibbsit` program it said that $k = 1$, M should be 148 and N should be 13,205. The new N was considerably greater than the number of iterations used for the third sample, so we repeated the third simulation.

Accordingly, we obtained a fourth sample containing $(M + N) = 13,353$ iterations. The marginal and conditional sample variances for this fourth sample are even closer to the correct values, and the sequence looks reasonably stationary (Figure 3(d)). As an additional check we calculated the sample estimate of the cumulative distribution function at the true 0.025 quantile by finding the proportion of simulations of η_0 less than or equal to -19.5 to

be

$$\hat{F}\left(F^{-1}(0.025)\right) = 0.020$$

using the fourth sample. This estimate is between 0.0125 and 0.0375, satisfying our original criterion. Finally, when we use `gibbsit` on the fourth sample it says that N should be 12,276, which is less than the number of iterations used in the fourth sample. We conclude that the fourth sample is good enough to perform inference with.

6 Discussion

We have described a way of determining the number of MCMC iterations needed, together with the `gibbsit` software that implements it. We have shown how this can also be used to diagnose slow convergence, and to help design a fully automatic generic Metropolis algorithm. The method is designed for the common situation where interest focuses on posterior quantiles of parameters of interest, but it can also be applied to the estimation of probabilities rather than parameters, which arises in image processing, the analysis of pedigrees and expert systems, for example (Raftery and Lewis, 1992a).

For simplicity, we have described our methods in the context of a single long run of MCMC iterations. However, Gelman and Rubin (1992b) have argued that one should use several independent sequences, with starting points sampled from an overdispersed distribution. Our method can be used in this situation also, with `gibbsit` applied to pilot sequences from each starting point, and the N iterations distributed among the different sequences. Various implementations are possible, and more research is needed on this topic.

Is it really necessary to use multiple sequences? Gelman and Rubin (1992b) argue that, in their absence, MCMC algorithms can give misleading answers, and this is certainly true. However, the creation of an overdispersed starting distribution can be a major chore and can add substantially to the complexity of an MCMC algorithm (Gelman and Rubin, 1992b: Section 2.1). There is clearly a trade-off between the extra work and cost required to produce and use an overdispersed starting distribution and whatever penalty or cost might be experienced on those rare occasions where MCMC without the overdispersed starting distribution arrives at misleading parameter estimates; the answer ultimately depends on the application.

In our experience, MCMC algorithms often do converge rapidly even from poorly chosen starting values, and when they do not, simple diagnostics such as those of Section 4 usually reveal the fact. Then simple trial and error with new starting values often leads rapidly to a satisfactory starting value, as illustrated by the Example in Section 4. However, there are cases where diagnostics will not reveal a lack of convergence (e.g. Gelman and Rubin, 1992a), and so multiple sequences should certainly be used for final inferences when much is at stake.

Diagnostics such as those of Section 4 should be used even with multiple sequences. This is because a bad starting value close to a local mode (which is what the Gelman-Rubin

multiple sequence methods are designed to protect against) is only one of the possible causes of slow convergence in MCMC. Others include high posterior correlations (which can be removed by approximate orthogonalization; see Hills and Smith, 1992), and “stickiness” of the chain. The latter often arises in hierarchical models when the algorithm enters parts of the parameter space where the random effects variance is small, and can require redesigning the MCMC algorithm itself (e.g. Besag and Green, 1993).

Acknowledgment

This research was supported by ONR contract no. N00014-91-J-1074 and by NIH grant 5R01HD26330. We are grateful to Andrew Gelman for stimulating discussions and to Wally Gilks for helpful comments.

References

- Besag, J.E. and Green, P.J. (1993). “Spatial statistics and Bayesian computation.” *Journal of the Royal Statistical Society B*, **55**, 25–37.
- Clifford, P. (1994). Contribution to the discussion of “Approximate Bayesian inference with the weighted likelihood bootstrap”. *Journal of the Royal Statistical Society B*, **56**, 34–35.
- Cox, D.R. and Miller, H.D. (1965). *The Theory of Stochastic Processes*. London: Chapman and Hall.
- Gelman, A. (1993). “A note on efficient Metropolis jumping rules.” Unpublished Manuscript, Department of Statistics, University of California, Berkeley.
- Gelman, A. and Rubin, D.B. (1992a). “A single series from the Gibbs sampler provides a false sense of security.” In *Bayesian Statistics 4* (J.M. Bernardo *et al.*, eds.), Oxford University Press, pp. 625–632.
- Gelman, A. and Rubin, D.B. (1992b). “Inference from iterative simulation using multiple sequences (with Discussion).” *Statistical Science*, **7**, 457–511.
- Hills, S.E. and Smith, A.F.M. (1992). “Parametrization issues in Bayesian inference (with Discussion).” In *Bayesian Statistics 4* (J.M. Bernardo *et al.*, eds.), Oxford University Press, pp. 227–246.
- Lewis, S.M. (1993). “Contribution to the discussion of three papers on Gibbs sampling and related Markov chain Monte Carlo methods.” *Journal of the Royal Statistical Society, series B*, **55**, 79–81.
- Lewis, S.M. (1994). *Multilevel Modeling of Discrete Event History Data Using Markov Chain Monte Carlo Methods*. Unpublished doctoral dissertation, Department of Statistics, University of Washington, Seattle, Wa.
- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. (1953). “Equations of state calculations by fast computing machines.” *Journal of Chemical Physics*, **21**, 1087–1092.

- Müller, P. (1991). “A generic approach to posterior integration and Gibbs sampling.” Technical Report no. 91-09, Department of Statistics, Purdue University, West Lafayette, In.
- Raftery, A.E. (1986). “A note on Bayes factors for log-linear contingency table models with vague prior information.” *Journal of the Royal Statistical Society, series B* **48**, 249–250.
- Raftery, A.E. and Banfield, J.D. (1991). “Stopping the Gibbs sampler, the use of morphology, and other issues in spatial statistics.” *Annals of the Institute of Statistical Mathematics*, **43**, 32–43.
- Raftery, A.E. and Lewis, S.M. (1992a). “How many iterations in the Gibbs sampler?” In *Bayesian Statistics 4*, (J.M. Bernardo *et al.*, eds.), Oxford: University Press, pp. 765–776.
- Raftery, A.E. and Lewis, S.M. (1992b). “One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo.” *Statistical Science*, **7**, 493–497.
- Raftery, A.E., Lewis, S.M., Aghajanian, A. and Kahn, M.J. (1993). “Event history modeling of World Fertility Survey data.” Working Paper No. 93-1, Center for Studies in Demography and Ecology, University of Washington.
- Schwarz, G. (1978). “Estimating the dimension of a model.” *Annals of Statistics*, **6**, 461–464.
- Tierney, L. (1991). “Exploring posterior distributions using Markov chains.” *Proceedings of the 23rd Interface Between Computing and Statistics*, 563–570.