Mini-Short Course on Monte Carlo Methods in Bayesian Computation

By
Ming-Hui Chen
Department of Statistics
University of Connecticut

Outline

1 Introduction

1.1 The Bayesian Paradigm

The Bayesian paradigm is based on specifying a probability model for the observed data D, given a vector of unknown parameters $\boldsymbol{\theta}$, leading to the likelihood function $L(\boldsymbol{\theta}|D)$.

Then we assume that $\boldsymbol{\theta}$ is random and has a *prior* distribution denoted by $\pi(\boldsymbol{\theta})$.

Inference concerning $\boldsymbol{\theta}$ is then based on the *posterior* distribution, which is obtained by Bayes' theorem. The posterior distribution of $\boldsymbol{\theta}$ is given by

$$\pi(\boldsymbol{\theta}|D) = \frac{L(\boldsymbol{\theta}|D)\pi(\boldsymbol{\theta})}{\int_\Omega L(\boldsymbol{\theta}|D)\pi(\boldsymbol{\theta})\ d\boldsymbol{\theta}},$$

where Ω denotes the parameter space of $\boldsymbol{\theta}$.

1.2 A Motivated Example: New Zealand Apple Data

Chen and Deely (1996) consider the problem of estimating apple production y in New Zealand. Consider the linear model

$$y = \sum_{j=1}^{10} \beta_j x_j + \epsilon,$$

where x_j is the number of trees of age j, $\epsilon \sim N(0, \sigma^2)$, and β_j equals the average number of cartons per tree at age j. Younger trees are known to produce fewer apples on average, so the model is subject to the constraints

$$0 \le \beta_1 \le \beta_2 \le \cdots \le \beta_{10}.$$

Using a noninformative prior for $\beta_1, \ldots, \beta_9$, and σ^2 as well as a proper prior for β_{10}, the full joint posterior density is given by

$$\pi(\boldsymbol{\beta}, \sigma^2 | D) \propto \frac{\exp\left\{-\frac{(\beta_{10}-\mu_{10})^2}{2\sigma_{10}^2}\right\}}{\sigma^{N+1}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{N}\left(y_i - \sum_{j=1}^{10}\beta_j x_{ij}\right)^2\right\},$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_{10})$ satisfies the above constraints, and $\sigma^2 > 0$.

Using the New Zealand apple data, $N = 207$, which can be found in

http://www.springer-ny.com/editorial/authors.html

by clicking on my name, or

http://www.stat.uconn.edu/∼mhchen/mcbook

we obtain $\mu_{10} = .998$, and $\sigma^2_{10} = .0891$, where μ_{10} and σ^2_{10} are specified using method-of-moments estimates from the growers' data for trees of age 10.

Using the Gibbs sampler, they compute

- **Posterior Estimate of β**

  For example, the posterior means and standard deviations are 0.0137 and 0.0081 for β_1, and 0.0250 and 0.0082 for β_2.

- **Posterior Prediction**

  Suppose the tree numbers from two growers are:

$$x_{\nu_1} = (0,\ 0,\ 0,\ 0,\ 0,\ 130,\ 340,\ 61,\ 224,\ 130)'$$

and

$$x_{\nu_2} = (0,\ 0,\ 558,\ 0,\ 1266,\ 0,\ 236,\ 0,\ 731,\ 86)'.$$

Then, the Bayesian predicted sums for the two growers were calculated as 2689. (Note that the actual total submission was 2665.)

- **Marginal density**

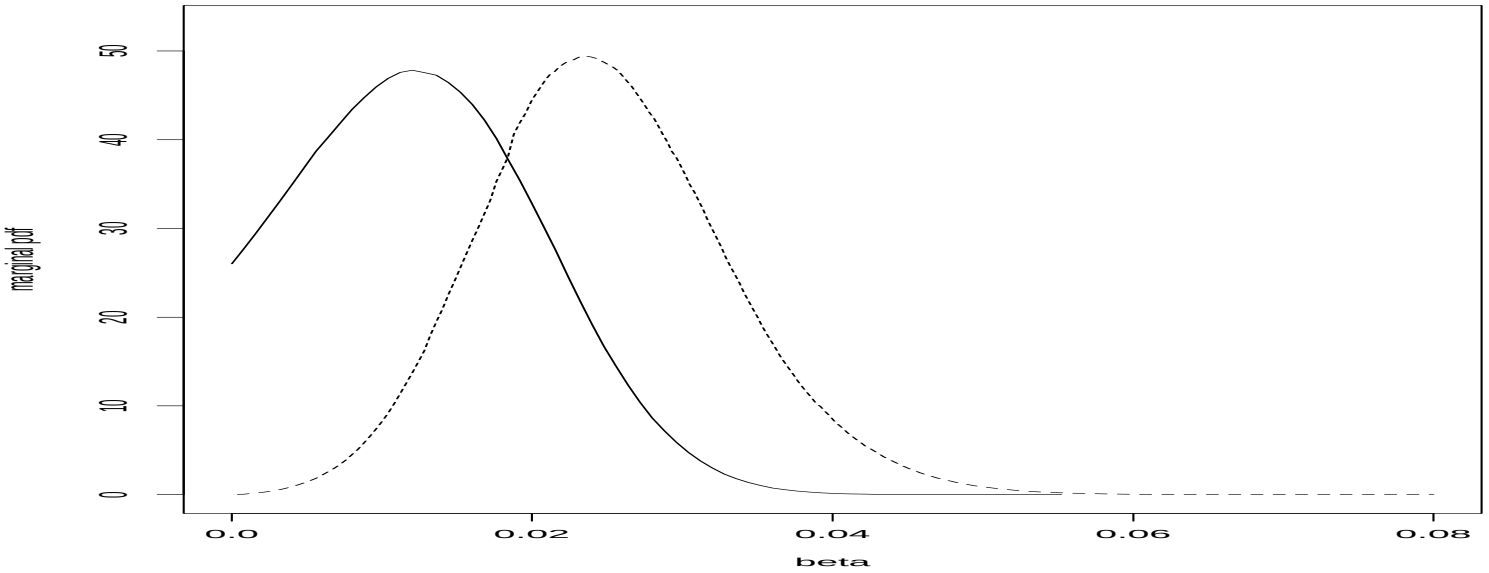


FIGURE 1.1. The marginal densities for β_1 and β_2.

The computational issues involved in this example are

(i) sampling from the posterior distribution (Gibbs sampler, due to the constrained parameters), and (ii) computing posterior quantities of interest (posterior means and standard deviations, prediction, and marginal density estimates).

1.3 Two Major Challenges in Advanced Bayesian Computation

(1) **How to sample from posterior distributions?**

One solution: Markov chain Monte Carlo (MCMC) sampling.

Books include Tanner (1996, Springer), Gilks, Richardson, and Spiegelhalter (1996, Chapman & Hall/CRC), Gamerman (1997, Chapman & Hall/CRC), Robert and Casella (1999, Springer, textbook), Robert (Ed.) (1998, Springer), and Chen, Shao, Ibrahim (2000, Chapter 2, Springer).

(2) **How to compute posterior quantities of interest using MCMC samples?**

This is also referred to as Bayesian computation after MCMC.

Chen, Shao, Ibrahim (2000, Springer) address this issue in detail.

1.4 Posterior Quantities

In Bayesian data analysis, many posterior quantities are of the form

$$E[h(\boldsymbol{\theta})|D] = \int_{R^p} h(\boldsymbol{\theta})\pi(\boldsymbol{\theta}|D)\ d\boldsymbol{\theta}, \tag{1}$$

where $h(\cdot)$ is a real-valued function of $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_p)'$. We call (1) an integral-type posterior quantity, or the posterior expectation of $h(\boldsymbol{\theta})$. In (1), we assume that

$$E(|h(\boldsymbol{\theta})| \mid D) = \int_{R^p} |h(\boldsymbol{\theta})|\pi(\boldsymbol{\theta}|D)\ d\boldsymbol{\theta} < \infty.$$

Integral-type posterior quantities include posterior means, posterior variances, covariances, higher-order moments, and probabilities of sets by taking appropriate functional forms of h.

For example, (1) reduces to:

(a) the posterior mean of $\boldsymbol{\theta}$ when $h(\boldsymbol{\theta}) = \boldsymbol{\theta}$;

(b) the posterior covariance of θ_j and θ_{j^*} if $h(\boldsymbol{\theta}) = (\theta_j - E(\theta_j|D))(\theta_{j^*} - E(\theta_{j^*}|D))'$;

(c) the posterior predictive density when $h(\boldsymbol{\theta}) = f(z|\boldsymbol{\theta})$, where $f(z|\boldsymbol{\theta})$ is the predictive density given the parameter $\boldsymbol{\theta}$; and

(d) the posterior probability of a set A if $h(\boldsymbol{\theta}) = 1\{\boldsymbol{\theta} \in A\}$, where $1\{\boldsymbol{\theta} \in A\}$ denotes the indicator function.

In (d), the posterior probability leads to a Bayesian p-value (see Meng 1994) by taking an appropriate form of A. In addition, the marginal posterior densities are also integral-type posterior quantities.

Some other posterior quantities such as *normalizing constants*, **Bayes factors**, and **posterior model probabilities**, may not simply be written in the form of (1). However, they are actually functions of integral-type posterior quantities.

2 Markov Chain Monte Carlo Sampling

2.1 Gibbs Sampler

The Gibbs sampler may be one of the best known MCMC sampling algorithms in the Bayesian computational literature.

- Original Idea: Grenander (1983)
- Formal Term: Geman and Geman (1984)
- Introduction to Bayesian Computation: Gelfand and Smith (1990)
- A Similar Idea: Tanner and Wong (1987)

Gibbs Sampling Algorithm

Step 0. Choose an arbitrary starting point $\boldsymbol{\theta}_0 = (\theta_{1,0}, \theta_{2,0}, \ldots, \theta_{p,0})'$, and set $i = 0$.

Step 1. Generate $\boldsymbol{\theta}_{i+1} = (\theta_{1,i+1}, \theta_{2,i+1}, \ldots, \theta_{p,i+1})'$ as follows:

- Generate $\theta_{1,i+1} \sim \pi(\theta_1 | \theta_{2,i}, \ldots, \theta_{p,i}, D)$;
- Generate $\theta_{2,i+1} \sim \pi(\theta_2 | \theta_{1,i+1}, \theta_{3,i}, \ldots, \theta_{p,i}, D)$;

  $\ldots \quad \ldots \quad \ldots$

- Generate $\theta_{p,i+1} \sim \pi(\theta_p | \theta_{1,i+1}, \theta_{2,i+1}, \ldots, \theta_{p-1,i+1}, D)$.

Step 2. Set $i = i + 1$, and go to Step 1.

Example 2.1 Bivariate normal model

Assume that the posterior distribution $\pi(\boldsymbol{\theta}|D)$ is a bivariate normal distribution $N_2(\boldsymbol{\mu}, \Sigma)$ with

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}.$$

Then the Gibbs sampler requires sampling from

$$\theta_1 \sim N\left(\mu_1 + \rho\frac{\sigma_1}{\sigma_2}(\theta_2 - \mu_2), \sigma_1^2(1-\rho^2)\right)$$

and

$$\theta_2 \sim N\left(\mu_2 + \rho\frac{\sigma_2}{\sigma_1}(\theta_1 - \mu_1), \sigma_2^2(1-\rho^2)\right).$$

Let $\{\boldsymbol{\theta}_i = (\theta_{1,i}, \theta_{2,i})',\ i \geq 0\}$ denote the Markov chain induced by the Gibbs sampler for the above bivariate normal distribution. If we start with $\boldsymbol{\theta}_0$, which is a fixed point or from some the stationary distribution, i.e., $\boldsymbol{\theta}_0 \sim N(\mu, \Sigma)$, then each of $\{\theta_{1,i},\ i \geq 0\}$ and $\{\theta_{2,i},\ i \geq 0\}$ is an AR(1) process

To see this, let $\{z_{1,i}, z_{2,i},\ i \geq 0\}$ be an i.i.d. $N(0,1)$ random variable sequence. Then the structure of the Gibbs sampler implies $\boldsymbol{\theta}_0 = (\theta_{1,0}, \theta_{2,0})$ for a fixed point or

$$\theta_{1,0} = \mu_1 + \sigma_1 z_{1,0},$$
$$\theta_{2,0} = \mu_2 + \rho\frac{\sigma_2}{\sigma_1}(\theta_{1,0} - \mu_1) + \sigma_2\sqrt{1-\rho^2}z_{2,0},$$

and

$$\theta_{1,i+1} = \mu_1 + \rho\frac{\sigma_1}{\sigma_2}(\theta_{2,i} - \mu_2) + \sigma_1\sqrt{1-\rho^2}z_{1,i+1},$$
$$\theta_{2,i+1} = \mu_2 + \rho\frac{\sigma_2}{\sigma_1}(\theta_{1,i+1} - \mu_1) + \sigma_2\sqrt{1-\rho^2}z_{2,i+1},$$

for $i \geq 0$.

Now, we consider the first component $\theta_{1,i+1}$. For $i \geq 0$,

$$\begin{aligned}\theta_{1,i+1} =& \mu_1 + \rho\frac{\sigma_1}{\sigma_2}\left[\rho\frac{\sigma_2}{\sigma_1}(\theta_{1,i} - \mu_1) + \sigma_2\sqrt{1-\rho^2}z_{2,i}\right] \\ &+ \sigma_1\sqrt{1-\rho^2}z_{1,i+1} \\ =& \mu_1 + \rho^2(\theta_{1,i} - \mu_1) + \rho\sigma_1\sqrt{1-\rho^2}z_{2,i} \\ &+ \sigma_1\sqrt{1-\rho^2}z_{1,i+1}.\end{aligned}$$

Let $\psi = \rho^2$ and $\sigma_1^{*2} = \sigma_1^2(1-\rho^4)$. Let $\{z_i^*,\ i \geq 0\}$ denote an i.i.d. $N(0,1)$ random variable sequence. Since $z_{1,i}$ and $z_{2,i+1}$ are independently and identically distributed as $N(0,1)$, then we can write $\boldsymbol{\theta}_0$ or

$$\begin{aligned}\theta_{1,0} &= \mu_1 + \sigma_1 z_0^*, \\ \theta_{1,i+1} &= \mu_1 + \psi(\theta_{1,i} - \mu_1) + \sigma_1^* z_{i+1}^* \quad \text{for } i \geq 0.\end{aligned}$$

Thus, $\{\theta_{1,i},\ i \geq 0\}$ is an AR(1) process with lag-one autocorrelation $\psi = \rho^2$.

2.2 Metropolis–Hastings Algorithm

The Metropolis–Hastings algorithm is developed by Metropolis et al. (1953) and subsequently generalized by Hastings (1970).

Let $q(\boldsymbol{\theta}, \boldsymbol{\vartheta})$ be a proposal density, which is also termed as a *candidate-generating density* by Chib and Greenberg (1995), such that

$$\int q(\boldsymbol{\theta}, \boldsymbol{\vartheta})\ d\boldsymbol{\vartheta} = 1.$$

Also let $U(0,1)$ denote the uniform distributionUniform distribution over $(0,1)$. Then, a general version of the Metropolis–Hastings algorithm for sampling from the posterior distribution $\pi(\boldsymbol{\theta}|D)$ can be described as follows:

Metropolis–Hastings Algorithm

Step 0. Choose an arbitrary starting point $\boldsymbol{\theta}_0$ and set $i = 0$.

Step 1. Generate a candidate point $\boldsymbol{\theta}^*$ from $q(\boldsymbol{\theta}_i, \cdot)$ and u from $U(0,1)$.

Step 2. Set $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}^*$ if $u \leq a(\boldsymbol{\theta}_i, \boldsymbol{\theta}^*)$ and $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i$ otherwise, where the acceptance probability is given by

$$a(\boldsymbol{\theta}, \boldsymbol{\vartheta}) = \min\left\{\frac{\pi(\boldsymbol{\vartheta}|D)q(\boldsymbol{\vartheta}, \boldsymbol{\theta})}{\pi(\boldsymbol{\theta}|D)q(\boldsymbol{\theta}, \boldsymbol{\vartheta})}, 1\right\}. \tag{2}$$

Step 3. Set $i = i + 1$, and go to Step 1.

A Few Special Cases:

- Independence Chain Metropolis (Tierney 1994)

$$q(\boldsymbol{\theta}, \boldsymbol{\vartheta}) = q(\boldsymbol{\vartheta})$$

- The Gibbs sampler

  This relationship is first pointed out by Gelman (1992) and further elaborated on by Chib and Greenberg (1995).

- Random Walk Chain (Müller 1991)

  Take $q(\boldsymbol{\theta}, \boldsymbol{\vartheta}) = q_1(\boldsymbol{\vartheta} - \boldsymbol{\theta})$, where $q_1(\cdot)$ is a multivariate density. The candidate $\boldsymbol{\theta}^*$ is thus drawn according to the process $\boldsymbol{\theta}^* = \boldsymbol{\theta} + \boldsymbol{\omega}$, where $\boldsymbol{\omega}$ is called the increment random variable and follows the distribution q_1.

- Hit-and-Run Algorithm

  See Smith (1980), Bélisle, Romeijn, and Smith (1993), Chen and Schmeiser (1996). This algorithm is very popular in the OR society.

2.3 Techniques for Improving Convergence of MCMC Sampling

- Centering Covariates
- Grouping, Collapsing, and Reparameterizations

  Liu (1994) and Liu, Wong, and Kong (1994): for the grouped and collapsed Gibbs techniques and

  Gelfand, Sahu, and Carlin (1995, 1996): the hierarchical centering method of Gelfand, Sahu, and Carlin (1995, 1996).
- Adaptive Direction Sampling (Gilks, Roberts, and George 1994; Roberts and Gilks 1994)
- Multiple-Try Metropolis (Liu, Liang, and Wong 2000)
- Auxiliary Variable Methods (Besag and Green 1993; Damien, Wakefield, and Walker 1999)
- Simulated Tempering (Marinari and Parisi 1992; Geyer and Thompson 1995)
- Grouped Move and Multigrid MC Sampling (Liu and Wu 1997; Liu

and Sabatti 1998 and 1999)

- Covariance-adjusted MCMC Sampling (Liu 1998)
- Many Others

2.4 Grouped and Collapsed Gibbs

To illustrate his idea, we consider a three-dimensional posterior distribution $\pi(\boldsymbol{\theta}|D)$, where $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)'$.

Liu (1994) considers the following three variations of the Gibbs sampler to sample from $\pi(\boldsymbol{\theta}|D)$:

Algorithm 1: *Standard* (*Original*) *Gibbs Sampler*

The standard Gibbs sampler requires drawing:

(i) $\theta_1 \sim \pi(\theta_1|\theta_2, \theta_3, D)$;

(ii) $\theta_2 \sim \pi(\theta_2|\theta_1, \theta_3, D)$;

(iii) $\theta_3 \sim \pi(\theta_3|\theta_1, \theta_2, D)$.

Algorithm 2: *Grouped Gibbs Sampler*

The grouped Gibbs sampler requires drawing:

(i) $(\theta_1, \theta_2) \sim \pi(\theta_1, \theta_2|\theta_3, D)$; $\Longleftarrow$ Grouping

(ii) $\theta_3 \sim \pi(\theta_3|\theta_1, \theta_2, D)$.

Algorithm 3: *Collapsed Gibbs Sampler*

The collapsed Gibbs sampler requires drawing:

(i) $(\theta_1, \theta_2) \sim \pi(\theta_1, \theta_2|D)$; $\Longleftarrow$ Collapsing

(ii) $\theta_3 \sim \pi(\theta_3|\theta_1, \theta_2, D)$.

Algorithm 3(a): *Modified Collapsed Gibbs Sampler*

The modified collapsed Gibbs sampler is similar to the original version by changing step (i) to:

(ia) $\theta_1 \sim \pi(\theta_1|\theta_2, D)$;

(ib) $\theta_2 \sim \pi(\theta_2|\theta_1, D)$.

A Note:

In general, auxiliary variable methods would increase autocorrelation in MCMC sampling path, which results in poor mixing.

The combination of the auxiliary variable method and the collapsing method is a very effective remedy.

2.5 Hierarchical Centering for Poisson Random Effects Models

Consider a Poisson mixed model:

$$y_i \sim \mathcal{P}(\mu_i),$$

where $\mu_i = \exp(x_i\beta + \epsilon_i)$ for $i = 1, 2, \ldots, n$. Here the random effects

$$\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)' \sim N(0, \Sigma),$$

where

$$\Sigma = \sigma^2 \begin{pmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{n-1} \\ \rho & 1 & \rho & \cdots & \rho^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{n-1} & \rho^{n-2} & \rho^{n-3} & \cdots & 1 \end{pmatrix}.$$

Proper Priors: $\sigma^2 \sim \mathcal{IG}(\delta_0, \lambda_0)$ ($\delta_0 > 0$, $\lambda_0 > 0$), and $\rho \sim U(-1, 1)$.

Posterior:

$$\begin{aligned} & \pi(\boldsymbol{\beta}, \rho, \sigma^2, \boldsymbol{\epsilon}|D) \\ \propto \quad & \exp\left\{ y'(X\boldsymbol{\beta} + \boldsymbol{\epsilon}) - J_n' Q(\boldsymbol{\beta}, \epsilon) - \frac{1}{2}\boldsymbol{\epsilon}'\Sigma^{-1}\boldsymbol{\epsilon} \right\} \\ & \times \frac{1}{\sigma^n (1-\rho^2)^{\frac{n-1}{2}}} \times (\sigma^2)^{-(\delta_0+1)} \exp(-\frac{\lambda_0}{\sigma^2}), \end{aligned}$$

where $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)'$, $J_n = (1, 1, \ldots, 1)'$, $Q(\boldsymbol{\beta}, \boldsymbol{\epsilon}) = (q_1, q_2, \ldots, q_n)'$, $q_i = \exp(\boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i) + \log(y_i!)$, X is the covariate matrix, and $D = (n, \boldsymbol{y}, X)$.

♠ **Gibbs sampling before hierarchical centering**

We need to sample from

(i) $[\boldsymbol{\beta}|\rho, \sigma^2, \boldsymbol{\epsilon}, D]$ (log-concave)

(ii) $[\boldsymbol{\epsilon}|\rho, \sigma^2, \boldsymbol{\beta}, D]$ (log-concave)

(iii) $[\rho|\sigma^2, \boldsymbol{\beta}, \boldsymbol{\epsilon}, D]$

(iv) $\sigma^2|\boldsymbol{\beta}, \rho, \boldsymbol{\epsilon}, D]$ (Inverse gamma)

For (iii), we use "**Localized Metropolis**" algorithm.

• **Localized Metropolis** Algorithm

First, we make the transformation:

$$\rho = \frac{-1 + e^{\xi}}{1 + e^{\xi}}, \ -\infty < \xi < \infty.$$

Then, we have

$$\pi(\xi|\sigma^2, \boldsymbol{\beta}, \boldsymbol{\epsilon}, D) = \pi(\rho|\sigma^2, \boldsymbol{\beta}, \boldsymbol{\epsilon}, D) \frac{2e^{\xi}}{(1 + e^{\xi})^2}.$$

Finally, we generate ξ by using a normal proposal $N(\hat{\xi}, \hat{\sigma}^2_{\hat{\xi}})$, where $\hat{\xi}$ is a maximizer of the logarithm of $\pi(\xi|\sigma^2, \boldsymbol{\beta}, \boldsymbol{\epsilon}, D)$, which can be obtained by, for example, Nelder-Mead algorithm or Newton-Raphson algorithm, and $\hat{\sigma}^2_{\hat{\xi}}$ is the minus of the inverse of the second derivative of $\log \pi(\xi|\sigma^2, \boldsymbol{\beta}, \boldsymbol{\epsilon}, D)$ evaluated at $\xi = \hat{\xi}$, that is,

$$\hat{\sigma}^{-2}_{\hat{\xi}} = - \left. \frac{d^2 \log \pi(\xi|\sigma^2, \boldsymbol{\beta}, \boldsymbol{\epsilon}, D)}{d\xi^2} \right|_{\xi = \hat{\xi}}.$$

The algorithm to generate ξ operates as follows:

(iiia) Let ξ be the current value.

(iiib) Generate a proposal value ξ^* from $N(\hat{\xi}, \hat{\sigma}^2_{\hat{\xi}})$.

(iiic) A move from ξ to ξ^* is made with probability

$$\min\left\{\frac{\pi(\xi^*|\sigma^2,\boldsymbol{\beta},\boldsymbol{\epsilon},D)\phi\left(\frac{\xi-\hat{\xi}}{\hat{\sigma}_{\hat{\xi}}}\right)}{\pi(\xi|\sigma^2,\boldsymbol{\beta},\boldsymbol{\epsilon},D)\phi\left(\frac{\xi^*-\hat{\xi}}{\hat{\sigma}_{\hat{\xi}}}\right)},1\right\},$$

where ϕ is the $N(0,1)$ pdf.

Note: The proposal $(\hat{\xi}, \hat{\sigma}^2_{\hat{\xi}})$ does not depend on the current value of ξ, which will typically produce a small autocorrelation among ξ's.

♠ **Hierarchical Centering**

Let $\boldsymbol{\eta} = \boldsymbol{\epsilon} + X\boldsymbol{\beta}$. Then we have

$$\pi(\boldsymbol{\beta}, \rho, \sigma^2, \boldsymbol{\eta}|D) \quad \propto \quad \exp\left\{\boldsymbol{y}'\boldsymbol{\eta} - J_n' Q(\boldsymbol{\eta}) - \frac{1}{2}(\boldsymbol{\eta} - X\boldsymbol{\beta})'\Sigma^{-1}(\boldsymbol{\eta} - X\boldsymbol{\beta})\right\}$$
$$\times \frac{1}{\sigma^n (1-\rho^2)^{\frac{n-1}{2}}} \times (\sigma^2)^{-(\delta_0+1)} \exp(-\frac{\lambda_0}{\sigma^2}).$$

The Gibbs sampler after hierarchical centering

- Sampling $\boldsymbol{\beta}$

$$\boldsymbol{\beta}|\rho, \sigma^2, \boldsymbol{\eta}, D \ \sim \ N(\hat{\boldsymbol{\beta}}, B^{-1}),$$

  where $B = X'\Sigma X$ and $\hat{\boldsymbol{\beta}} = B^{-1}X'\Sigma\boldsymbol{\eta}$.

  (It is very convenient to draw $\boldsymbol{\beta}$ after hierarchical centering.)

- Sampling σ^2

$$\sigma^2 \sim \mathcal{IG}\left(\delta_0 + n/2, \lambda_0 + \frac{1}{2}(\boldsymbol{\eta} - X\boldsymbol{\beta})'\Sigma(\boldsymbol{\eta} - X\boldsymbol{\beta})\right).$$

- Sampling $\boldsymbol{\eta}$

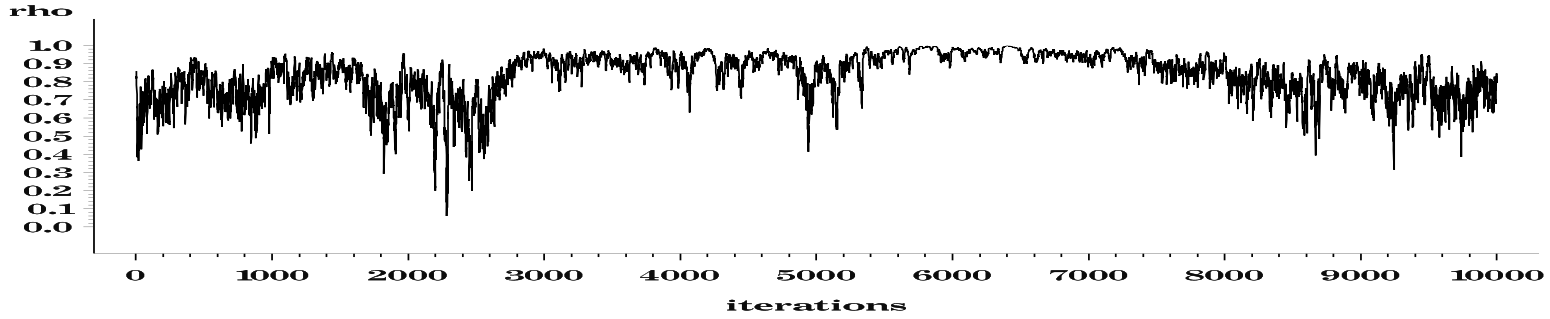
  Use the adaptive rejection algorithm of Gilks and Wild (1992).

- **Sampling correlation** ρ

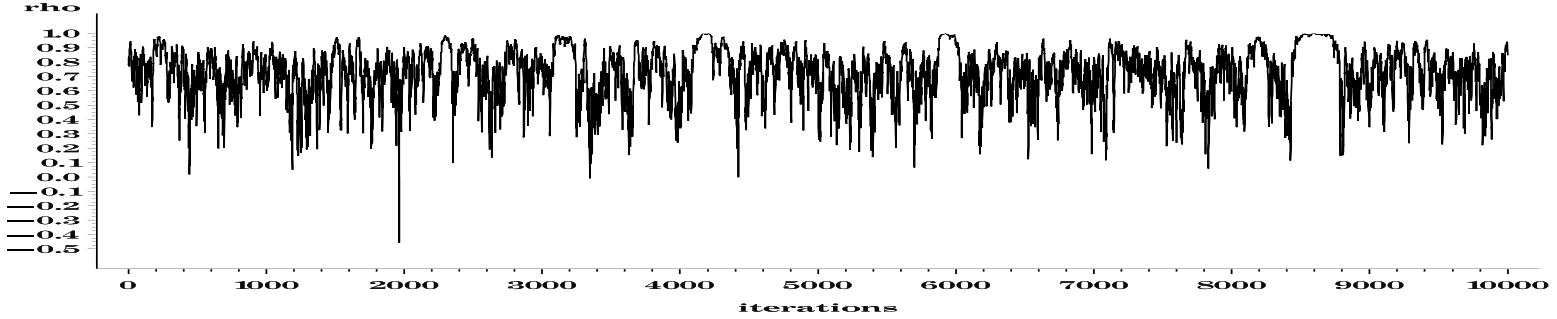
  Use the "Localized Metropolis".

We use the 1994 pollen count data to illustrate the effectiveness of hierarchical centering.

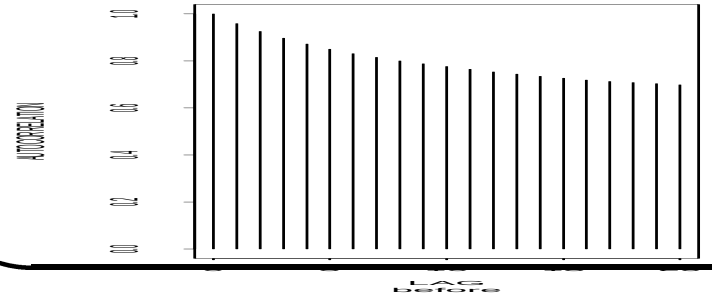
Trace plot for ρ before hierarchical centering

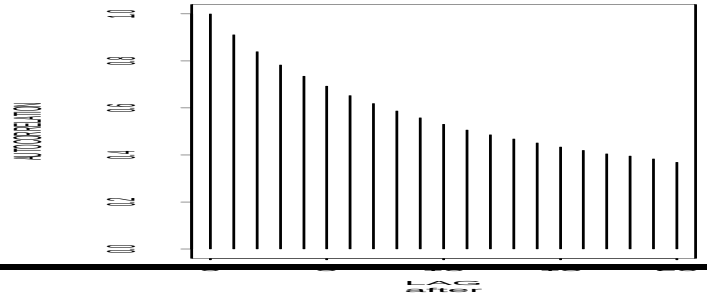


Trace plot for ρ after hierarchical centering



Autocorrelation Plots





2.6 Grouped Move and Multigrid Monte Carlo Sampling

Goodman and Sokal (1989) present a comparative review of the multigrid Monte Carlo (MGMC) method, which is a stochastic generalization of the multigrid (MG) method for solving finite-difference equations. Liu and Wu (1997) and Liu and Sabatti (1998 and 1999) generalize Goodman and Sokal's MGMC via groups of transformations with applications to MCMC sampling. They propose a Grouped Move Multigrid Monte Carlo (GM-MGMC) algorithm and a generalized version of the MGMC algorithm for sampling from a target posterior distribution.

GM-MGMC Algorithm

MCMC Step. Generate an iteration $\boldsymbol{\theta}_i$ from the parent MCMC.

GM Step. Draw the group element g from

$$g \sim \pi(g|\boldsymbol{\theta}_i)H(g) \propto \pi(g(\boldsymbol{\theta}_i)|D)J_g(\boldsymbol{\theta}_i)H(dg),$$

and *adjust*

$$\boldsymbol{\theta}_i \leftarrow g(\boldsymbol{\theta}_i).$$

In the GM step, $H(dg)$ is the right-invariant Haar measure on Ω and $J_g(\boldsymbol{\theta}_i)$ is the Jacobian of g evaluated at $\boldsymbol{\theta}_i$.

Notes:

(i) The adjusted MCMC, GM-MGMC, can converge faster than the parent MCMC.

(ii) From the implementational point of view, it requires only adding an additional subroutine to the parent MCMC main program. Thus, it is very convenient to do GM step.

2.7 Covariance-Adjusted MCMC Algorithm

Liu (1998) provides an alternative method for speeding up an MCMC algorithm using the idea of covariance adjustment. Let $\{\boldsymbol{\theta}_i,\ i = 0, 1, 2, \dots\}$ be generated by the parent MCMC algorithm, having the stationary distribution $\pi(\boldsymbol{\theta}|D)$. Also let $(\boldsymbol{\xi}, \boldsymbol{\delta}) = \mathcal{M}(\boldsymbol{\theta})$ be a one-to-one mapping from Ω on which the target distribution is defined onto the space $\Xi \times \Delta$. Then, the covariance-adjusted MCMC (CA-MCMC) algorithm at the i^{th} iteration consists of the following two steps:

CA-MCMC Algorithm

MCMC Step. Generate an iteration $\boldsymbol{\theta}_i$ from the parent MCMC and compute $(\boldsymbol{\xi}_i, \boldsymbol{\delta}_i) = \mathcal{M}(\boldsymbol{\theta}_i)$.

CA Step. Draw $\boldsymbol{\delta}_i^*$ from the conditional posterior distribution $\pi(\boldsymbol{\delta}|\boldsymbol{\xi}_i, D)$ and *adjust* $\boldsymbol{\theta}_i$ by

$$\boldsymbol{\theta}_i \ \leftarrow\ \boldsymbol{\theta}_i^* = \mathcal{M}^{-1}(\boldsymbol{\xi}_i, \boldsymbol{\delta}_i^*),$$

where $\mathcal{M}^{-1}(\boldsymbol{\xi}, \boldsymbol{\delta})$ is the inverse mapping of $(\boldsymbol{\xi}, \boldsymbol{\delta}) = \mathcal{M}(\boldsymbol{\theta})$.

Properties of CA-MCMC:

- If the Markov chain induced by an MCMC algorithm is irreducible, aperiodic, and stationary with the equilibrium distribution $\pi(\boldsymbol{\theta}|D)$, so is the covariance-adjusted Markov chain.
- The CA-MCMC algorithm converges at least as fast as its parent MCMC algorithm in the sense that the CA-MCMC algorithm results in a smaller reversed Kullback–Leibler information distance (e.g., Liu, Wong, and Kong 1995). This implies that the Markov sequence induced by the CA-MCMC algorithm has less dependence than that induced by the parent MCMC algorithm.

Example 2.2 One-way analysis of variance with random effects

Assume that the error variance σ_e^2 is known and that a single observation y_i for each population, i.e.,

$$y_i = \mu + \alpha_i + \epsilon_i, \;\; i = 1, 2, \ldots, m,$$

where $\epsilon_i \sim N(0, \sigma_e^2)$, $\alpha_i \sim N(0, \sigma_\alpha^2)$, and σ_α^2 is also known. We assume that $\pi(\mu) \propto 1$ and let $\bar{y} = (1/m) \sum_{i=1}^m y_i$ and $D = (y_1, y_2, \ldots, y_m)$.

For this one-way analysis of the variance model, the vector of model parameters is $\boldsymbol{\theta} = (\mu, \alpha_1, \alpha_2, \ldots, \alpha_m)'$. We use the Gibbs sampler as the parent MCMC algorithm. To apply the CA-MCMC algorithm, we need to construct $\boldsymbol{\xi}$ and $\boldsymbol{\delta}$.

- Potential slow convergence:

  μ and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)'$ may be highly correlated.

- The sufficient statistic for μ

$$\bar{\alpha} = \frac{1}{m} \sum_{i=1}^{m} \alpha_i.$$

- Defining the mapping

  Let $\xi_i = \alpha_i - \bar{\alpha}$. Define

$$\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_m)' \text{ and } \boldsymbol{\delta} = (\mu, \bar{\alpha})',$$

  and let $\Xi = \{\xi : \sum_{i=1}^{m} \xi_i = 0,\ \xi_i \in R \text{ for } i = 1, 2, \ldots, m\}$. Then, this transformation clearly defines a one-to-one mapping from R^{m+1} to $\Xi \times R^2$. The Jacobian of this transformation is $J_{(\mu,\alpha)\to(\mu,\bar{\alpha},\xi_1,\ldots,\xi_{m-1})} = 1$.

CA-MCMC for One-Way Analysis of Variance with Random Effects

Gibbs Step. Draw $(\mu|\boldsymbol{\alpha}, D) \sim N(\bar{y} - \bar{\alpha}, \sigma_e^2/m)$ and

$$(\alpha_i|\mu, D) \sim N\left(\frac{\sigma_\alpha^2}{\sigma_e^2 + \sigma_\alpha^2}(y_i - \mu), \frac{\sigma_e^2\sigma_\alpha^2}{\sigma_e^2 + \sigma_\alpha^2}\right).$$

CA Step. Draw $(\bar{\alpha}^*|\boldsymbol{\xi}, D) \sim N(0, \sigma_\alpha^2/m)$ and

$$(\mu^*|\bar{\alpha}^*, \boldsymbol{\xi}, D) \sim N\left(\bar{y} - \bar{\alpha}^*, \frac{\sigma_e^2}{m}\right),$$

then *adjust*

$$\mu \leftarrow \mu^* \text{ and } \alpha_i \leftarrow \xi_i + \bar{\alpha}^* \text{ for } i = 1, 2, \ldots, m.$$

An Interesting Note:

The draws of $(\mu^*, \bar{\alpha}^*, \xi_1 + \bar{\alpha}^*, \xi_2 + \bar{\alpha}^*, \ldots, \xi_m + \bar{\alpha}^*)$ are independent.

The reason is that μ^*, $\bar{\alpha}^*$, ξ_1, $\ldots$, ξ_m do not depend on the values of $(\mu, \boldsymbol{\alpha})$ from the previous iteration.

In fact, let the z_i's be i.i.d. from $N(0,1)$. Then, in the Gibbs step

$$\alpha_i = \frac{\sigma_\alpha^2}{\sigma_e^2 + \sigma_\alpha^2}(y_i - \mu) + z_i \left(\frac{\sigma_e^2 \sigma_\alpha^2}{\sigma_e^2 + \sigma_\alpha^2} \right)^{1/2},$$

and thus,

$$\xi_i = \alpha_i - \bar{\alpha} = \frac{\sigma_\alpha^2}{\sigma_e^2 + \sigma_\alpha^2}(y_i - \bar{y}) + (z_i - \bar{z}) \left(\frac{\sigma_e^2 \sigma_\alpha^2}{\sigma_e^2 + \sigma_\alpha^2} \right)^{1/2},$$

where $\bar{y}$ and $\bar{z}$ denote the sample means of the y_i's and z_i's, respectively. Therefore, the ξ_i's do not depend on the values of μ from the previous Gibbs iteration.

Some Facts:

- For this particular example, the rate of convergence of the CA-MCMC algorithm is 0 (prefect sampling).
- Roberts and Sahu (1997) show that the rate of convergence of the Markov chain using the Gibbs step only is $\sigma_\alpha^2/(\sigma_\alpha^2+\sigma_e^2)$ and that the rate of convergence of the Gibbs sampler based on the hierarchically centered transformation, namely, $\mu=\mu$ and $\eta_i=\mu+\alpha_i$ for $i=1,2,\ldots,m$, is $\sigma_e^2/(\sigma_\alpha^2+\sigma_e^2)$.
  (If $\sigma_e^2>\sigma_\alpha^2$, the Gibbs is better than the hierarchical centering; and If $\sigma_e^2<\sigma_\alpha^2$, the hierarchical centering is more preferable.)
- CA-MCMC sampling outperforms original Gibbs sampling as well as hierarchical centering.

2.8 A Comprehensive Illustration

Data

Gender	Rating	Gender	Rating
F	good	F	good
M	fair	M	poor
F	good	F	good
M	poor	M	good

Define

$$Y = \begin{cases} 1 & \text{poor} \\ 2 & \text{fair} \\ 3 & \text{good} \end{cases},$$

and X denote the covariate for *Gender* with $X = 1$ for female and $X = 0$ for male.

Consider a probit model. Albert and Chib (1993) introduced latent variables Z_i such that

$$Y_i = l \text{ iff } \gamma_{l-1} \le Z_i < \gamma_l,$$

for $l = 1, 2, \ldots, L = 3$, where the cutpoints $-\infty = \gamma_0 < \gamma_1 = 0 < \gamma_2 < \gamma_3 = \infty$. Let $Z = (Z_1, Z_2, \ldots, Z_n)'$ $(n = 8)$. Then, the complete-data likelihood for $\boldsymbol{\beta} = (\beta_0, \beta_1)$ and γ_2 is

$$L(\boldsymbol{\beta}, \gamma_2, Z|D) \propto \prod_{i=1}^{n} \left[\exp\{-\frac{1}{2}(Z_i - \boldsymbol{x}_i'\boldsymbol{\beta})^2\} 1_{\{\gamma_{y_i-1} \le Z_i < \gamma_{y_i}\}} \right].$$

Consider a prior distribution for (β, γ_2) taking the form

$$\pi(\boldsymbol{\beta}, \gamma_2) \propto \pi(\boldsymbol{\beta}) \propto \exp\Big\{ -\frac{\tau}{2}\boldsymbol{\beta}'\boldsymbol{\beta} \Big\},$$

where $\tau = 0.001$. Note that when $\pi(\boldsymbol{\beta}, \gamma_2) \propto 1$, then the posterior is improper; see Chen and Shao (1998, AISM). With the choice of $\tau = 0.001$, it is expected that the resulting posterior is essentially flat.

AC Algorithm (Albert and Chib, 1993, a regular Gibbs)

Step 1: Sample $\boldsymbol{\beta}$ from

$$\boldsymbol{\beta}|Z,\gamma \ \sim \ N(\hat{\boldsymbol{\beta}}, B^{-1}),$$

where $B = \tau I_2 + X'X$ and $\hat{\beta} = B^{-1}X'Z$.

Step 2: Sample Z_i from

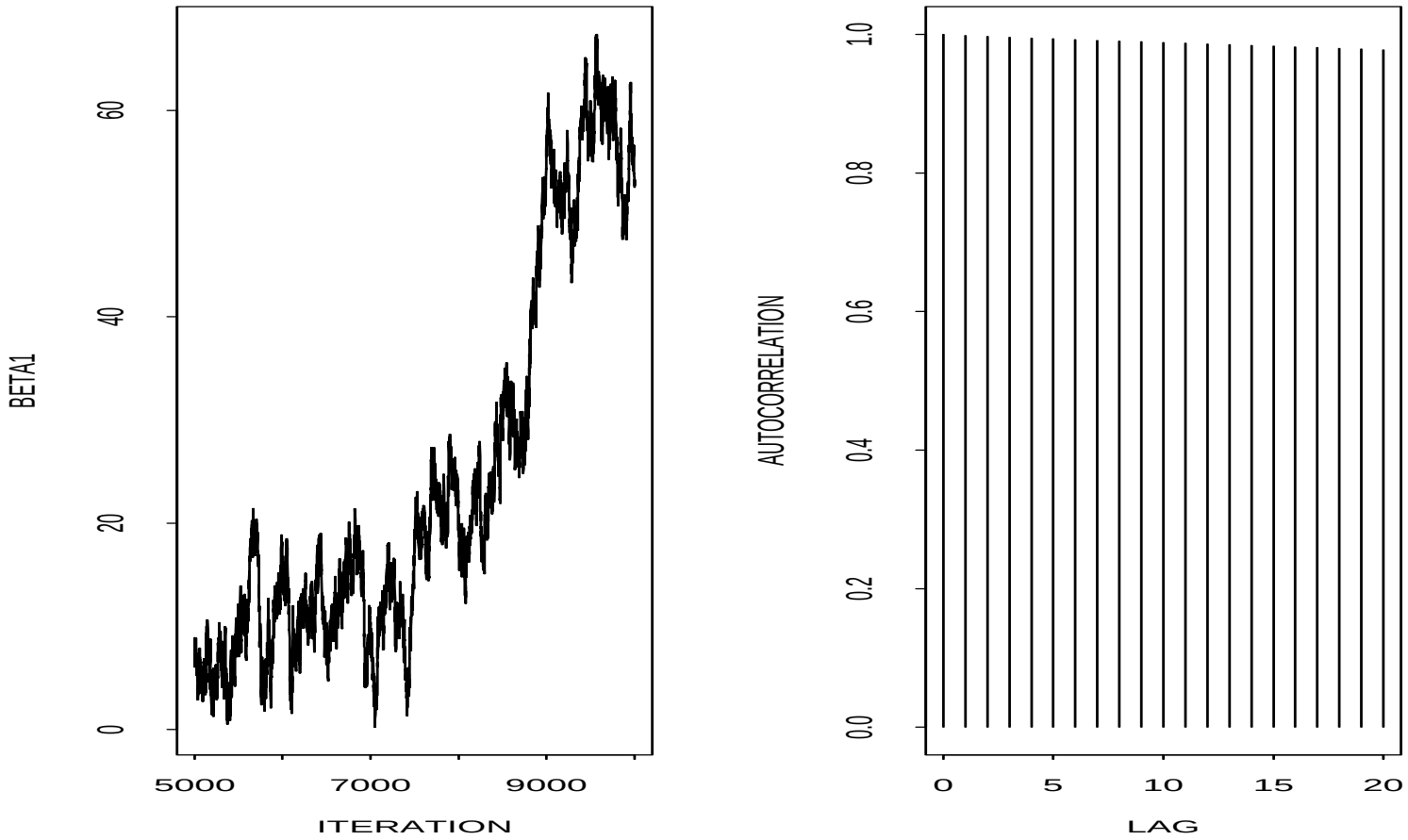
$$Z_i \ \sim \ N(x_i\boldsymbol{\beta}, 1) \ \ \gamma_{y_i-1} \le Z_i \le \gamma_{y_i}.$$

Step 3: Sample γ from

$$\gamma_l|\gamma_{(-l)}, \boldsymbol{\beta}, Z \ \sim \ U[a_l, b_l],$$

where $a_l = \max\left\{\gamma_{l-1}, \max\limits_{y_i=l} Z_i\right\}$, $b_l = \min\left\{\gamma_{l+1}, \min\limits_{y_i=l+1} Z_i\right\}$, and $\gamma_{(-l)}$ is γ with γ_l deleted.

AC Algorithm for β_1



Fact: The convergence is very slow.

Nandram-Chen's Algorithm (Transformation):

Nandram and Chen (1996) considered a reparameterization approach, which is based on:

$$\delta = 1/\gamma_{L-1}, \quad \gamma^* = \delta\gamma, \quad \boldsymbol{\beta}^* = \delta\boldsymbol{\beta} \text{ and } Z^* = \delta Z. \tag{3}$$

When $L = 3$, $-\infty = \gamma_0^* < \gamma_1^* = 0 < \gamma_2^* = 1 < \gamma_3^* = \infty$.

No unknown cutpoints.

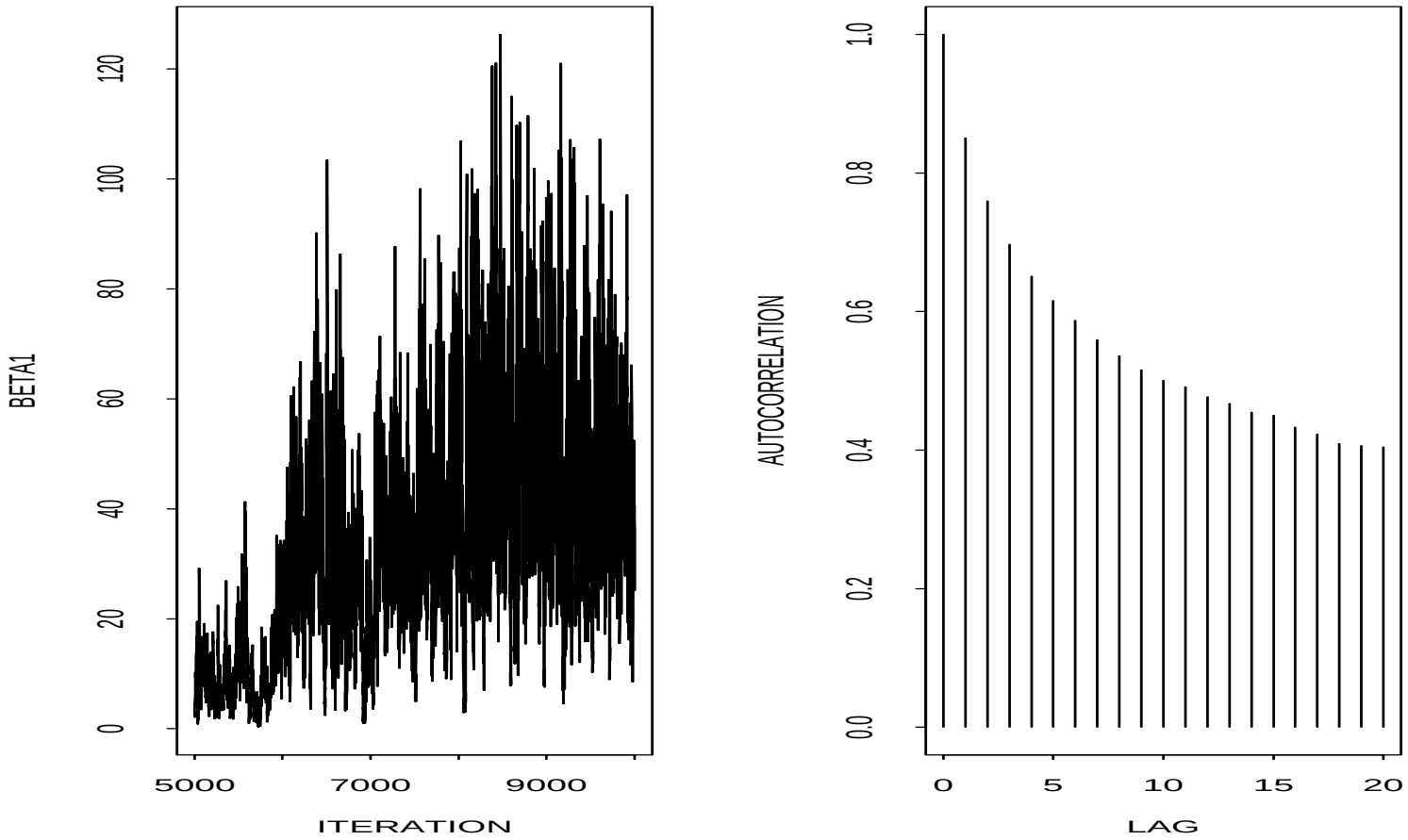
NC Algorithm:

Steps 1 and 2: Similar to AC's.

Step 3: Sample δ^2 from

$$\delta^2|\boldsymbol{\beta}^*, Z^* \sim \mathcal{IG}\left\{\frac{n+p+L-1}{2}, \frac{1}{2}\left[(Z^* - X\boldsymbol{\beta}^*)'(Z^* - X\boldsymbol{\beta}^*) + \tau\boldsymbol{\beta}^{*\prime}\boldsymbol{\beta}^*\right]\right\}.$$

NC Algorithm for β_1



Fact: NC converges faster than AC.

MGMC Algorithm

- **AC Steps.**
- **Adjusted Step:** Sample G^* from

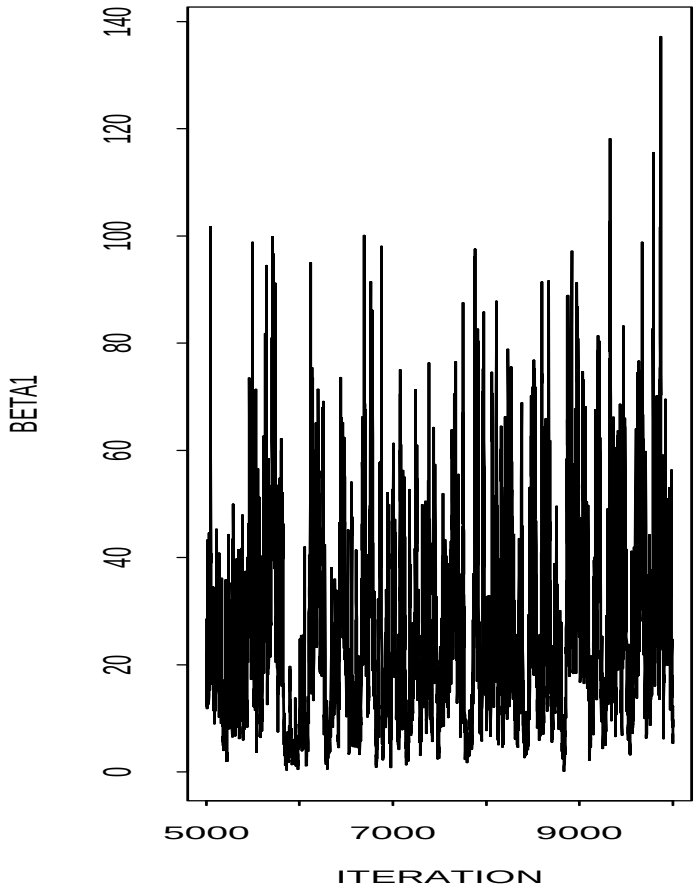
$$G^*|Z,\boldsymbol{\beta} \sim \text{Gamma}\left(\frac{n+p+L-1}{2}, \frac{(Z-X\boldsymbol{\beta})'(Z-X\boldsymbol{\beta})+\tau\boldsymbol{\beta}'W^{-1}\boldsymbol{\beta}}{2}\right).$$

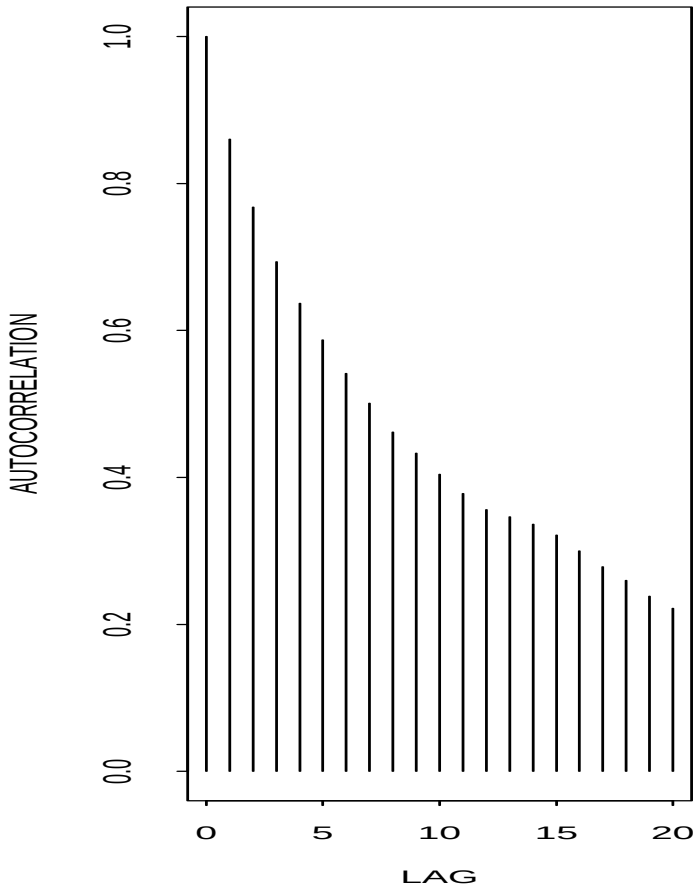
Calculate $g=\sqrt{G^*}$. Then, update $\boldsymbol{\beta}$, γ, and Z by

$$\boldsymbol{\beta} \leftarrow g\boldsymbol{\beta},\ \gamma \leftarrow g\gamma,\ \text{and}\ Z \leftarrow gZ.$$

Here $n=8$, $p=2$, and $L=3$.

MGMC Algorithm for β_1





Fact: MGMC is slightly better than NC.

Further Adjustment

Note that the hard convergence parameter is β_1 due to the lack of information to estimate it. We can use the CA-MC of C. Liu (1998) to present a further adjustment step.

- MGMC steps
- CA-MC adjustment step.

The Technical Detail for further Acceleration

To speed up the accelerated Gibbs further, we add another CA-step, which draws $\mu = \beta_0 + \beta_1$ jointly with its sufficient statistic

$$T = \frac{1}{4} \sum_{x_i=1} z_i,$$

conditioning on the current draws of $\{Z_i : x_i = 0\}$, γ_2, β_0, and $\{Z_i^* = Z_i - T : x_i = 1\}$. Denoting by

$$\mathrm{N}(\alpha\beta_0, \sigma_0^2)$$

the conditional distribution of μ given β_0 from the prior distribution for

β, we have the conditional distribution of (μ, T) given $\{Z_i : x_i = 0\}$, γ_2, β_0, and $\{Z_i^* = Z_i - T : x_i = 1, \text{ where } \sum_{x_i=1} Z_i^* = \sum_{x_i=1}(Z_i - T) = 0\}$:

$$\mathrm{N}_2\left(\begin{bmatrix} \alpha\beta_0 \\ \alpha\beta_0 \end{bmatrix}, \begin{bmatrix} \sigma_0^2 & \sigma_0^2 \\ \sigma_0^2 & \sigma_0^2 + \frac{1}{4} \end{bmatrix}\right) \qquad \left(\max_{x_i=1}(\gamma_2 - z_i^*) \le T < \infty\right).$$

Thus, the corresponding CA-step can be accomplished by

(i) taking a draw of T from

$$\mathrm{N}\left(\alpha\beta_0, \sigma_0^2 + \frac{1}{4}\right) \qquad \left(\max_{x_i=1}(\gamma_2 - Z_i^*) \le T < \infty\right)$$

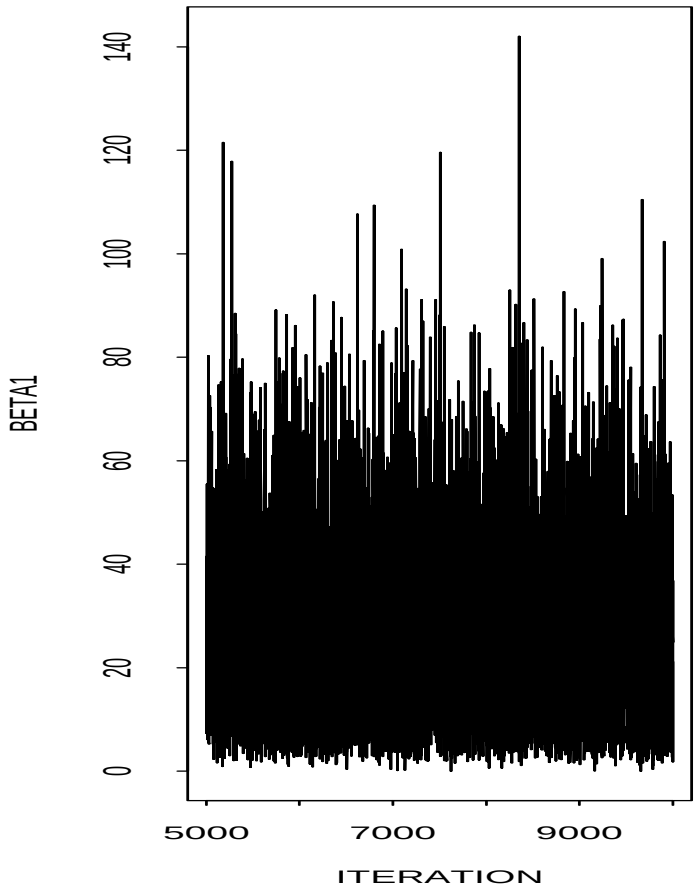
and then a draw of μ from

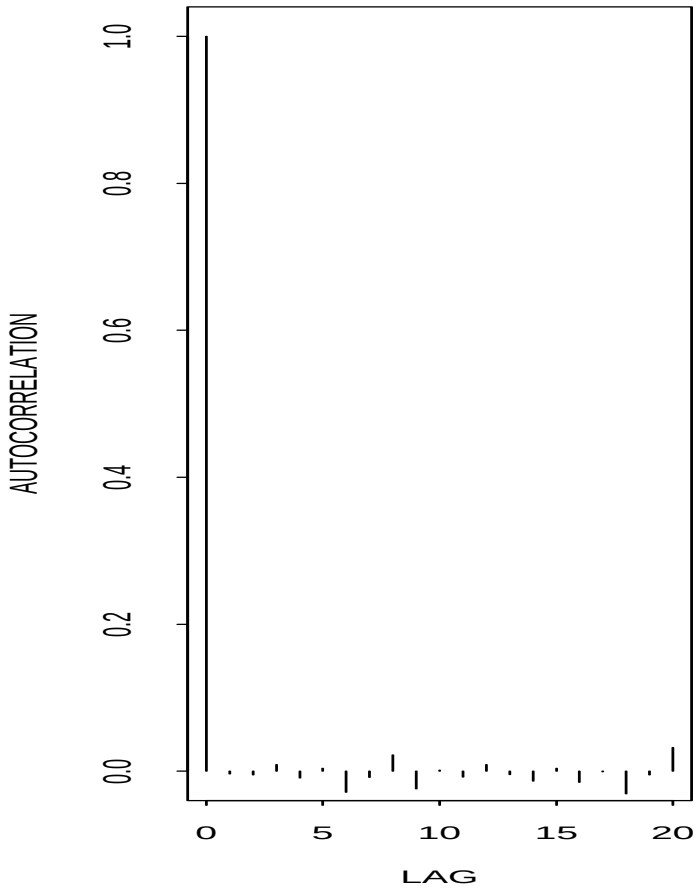
$$\mathrm{N}\left(\alpha\beta_0 + \frac{\sigma_0^2}{\sigma_0^2 + 1/4}(T - \alpha\beta_0), \sigma_0^2 - \frac{\sigma_0^4}{\sigma_0^2 + 1/4}\right);$$

(ii) updating β_1 and $\{Z_i : x_i = 1\}$ by

$$\beta_1 \leftarrow \mu - \beta_0 \text{ and } \{Z_i : x_i = 1\} \leftarrow \{Z_i^* + T : x_i = 1\}.$$

CA-MGMC Algorithm for β_1





Note that: the autocorrelations of β_1 from the CA GM-MGMC algorithm disappear even at lag 1.

3 Basic Monte Carlo Methods

3.1 Introduction

Objective: Estimate $E[h(\boldsymbol{\theta})|D] = \int_{R^p} h(\boldsymbol{\theta})\pi(\boldsymbol{\theta}|D)\, d\boldsymbol{\theta}$.

Computational Nature: Data are generated from computer, the distribution from which the data are generated is known, sample size is generally large, and some additional information may also be available.

Basic Data: An MCMC sample $\{\boldsymbol{\theta}_i, i = 1, 2, \ldots, n\}$ from $\pi(\boldsymbol{\theta}|D)$ or some additional information.

Basic Estimator: The usual sample mean of the $h(\boldsymbol{\theta}_i)$, given by

$$\hat{E}_{\text{avg}}(h) = \frac{1}{n}\sum_{i=1}^{n} h(\boldsymbol{\theta}_i).$$

However, when the distribution of $h(\boldsymbol{\theta})$ is skewed, the usual sample mean is unstable.

3.2 Weighted Monte Carlo (MC) Estimator

$$\hat{E}_w(h) = \frac{\sum_{i=1}^n w_i h(\theta_i)}{\sum_{l=1}^n w_l}$$

subject to certain constraints.

- **A Question?**

How to construct the weight $w = (w_1, w_2, \ldots, w_n)'$?

- **An Original Idea (Trotter and Tukey 1956)**

Instead of sampling θ alone, they suggested generating a pair (θ, w), where w is a real-valued weight, from some joint distribution $\pi(\theta, w)$ so that for all reasonable real-valued functions $h(\theta)$,

$$\int_{R^{d+1}} wh(\theta)\pi(\theta, w)\ d\theta\ dw = KE(h(\theta)|D),$$

where $K \neq 0$ is an h-independent constant.

- **Problem:** It is difficult to construct $\pi(\theta, w)$.

- **Possible Solutions**

(i) The Dynamic Weighting Method (Wong and Liang 1997; and Liu, Liang and Wong 1998)

Construct a Markov chain to sample (θ, w) jointly from $\pi(\theta, w)$ so that the resulting Markov chain Monte Carlo (MCMC) samples can be used for estimating $E(h(\theta)|D)$.

(ii) The Recycling Method (Casella and Robert 1996; and Casella and Robert 1998))

Recycle the random variables involved in the acceptance-rejection or Metropolis algorithm to construct the weights w_i's.

(iii) Partition-Weighted Monte Carlo Estimation (Chen and Shao 2001)

Borrow the idea from the stratified sampling method (Thompson 1992) to construct the weights w_i's by partitioning the sample space Ω into more homogeneous subspaces and then assign the same weight in each subspace.

(iv) George-Peng Method (1999)

They partition a Monte Carlo sample (not the support of posterior distribution) into several subsets, and then assign a fixed weight or a random weight to each subset.

♠ Optimal Fixed Weighted Estimator

Theorem *Let Σ denote the covariance matrix of $h(\boldsymbol{\theta}_1)$, $h(\boldsymbol{\theta}_2)$, $\ldots$, $h(\boldsymbol{\theta}_n)$. Then, the value of $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)'$ that minimizes the variance of $\hat{E}_w(h)$ is*

$$\boldsymbol{w}_{\text{opt}} = \frac{\Sigma^{-1}\mathbf{1}}{\mathbf{1}'\Sigma^{-1}\mathbf{1}},$$

where $\mathbf{1} = (1, 1, \ldots, 1)'$, and the optimal weighted MC estimator is given by

$$\hat{E}_{\text{opt}}(h) = (h(\boldsymbol{\theta}_1), h(\boldsymbol{\theta}_2), \ldots, h(\boldsymbol{\theta}_n))\boldsymbol{w}_{\text{opt}}$$

with variance

$$\text{Var}(\hat{E}_{\text{opt}}(h)) = \frac{\mathbf{1}}{\mathbf{1}'\Sigma^{-1}\mathbf{1}}.$$

A Note:

If $\Sigma = \sigma^2 I_n$, i.e., the $\boldsymbol{\theta}_i$'s are i.i.d. observations, then $\boldsymbol{w}_{\text{opt}} = (1/n)\mathbf{1}$, and the optimal weighted MC estimate $\hat{E}_{\text{opt}}(h)$ reduces to

$$\hat{E}_{\text{avg}}(h) = \frac{1}{n}\sum_{i=1}^{n} h(\boldsymbol{\theta}_i)$$

with variance σ^2/n. Thus, for i.i.d. samples, the usual sample mean of the $h(\boldsymbol{\theta}_i)$ is the best estimate of $E(h(\boldsymbol{\theta})|D)$.

Example 3.1 Independent sample versus dependent sample.

In MC estimation, it is usually believed that only negative correlations help improve the MC estimates. In this example, we will illustrate that with the optimal weights, positive correlations can also be helpful.

Let θ_1, θ_2, and θ_3 be i.i.d. observations with a common variance σ^2. Then the usual sample mean is

$$\hat{\theta} = \frac{1}{3}\sum_{i=1}^{3}\theta_i$$

with variance

$$\mathrm{Var}(\hat{\theta}) = \frac{\sigma^2}{3}.$$

Also let ξ_1, ξ_2, and ξ_3 be dependent observations with covariance matrix

$$\Sigma = \sigma^2 \begin{pmatrix} 1 & 0.7 & 0 \\ 0.7 & 1 & 0.7 \\ 0 & 0.7 & 1 \end{pmatrix}.$$

Then ξ_i has the same marginal variance as θ_i. The optimal weighted MC estimator is given by

$$\hat{\xi}_{\text{opt}} = 1.5\xi_1 - 2\xi_2 + 1.5\xi_3$$

with variance

$$\text{Var}(\hat{\xi}_{\text{opt}}) = \frac{\sigma^2}{10}.$$

Therefore,

$$\frac{\text{Var}(\hat{\theta})}{\text{Var}(\hat{\xi}_{\text{opt}})} = \frac{10}{3}.$$

Thus, the optimal weighted MC estimate with the dependent sample has a variance that is less one-third of the variance of the optimal estimate with the independent sample.

Example 3.2 A dependent sample from an AR(1) process.

Assume that $\{\theta_1, \theta_2, \ldots, \theta_n\}$ is a dependent sample from an AR(1) process with marginal variance σ^2 and lag-one autocorrelation ρ. Consider $h(\theta) = \theta$. Then, the variance of the usual sample mean of the $h(\theta_i)$ can be expressed as

$$\mathrm{Var}(\hat{E}_{\mathrm{avg}}(h)) = \frac{\sigma^2}{n}\left[\frac{1+\rho}{1-\rho} - \frac{2\rho(1-\rho^n)}{n(1-\rho)^2}\right].$$

It can be shown that the optimal fixed weighted estimator is given by

$$\hat{E}_{\mathrm{opt}}(h) = \frac{\theta_1 + (1-\rho)\sum_{i=2}^{n-1}\theta_i + \theta_n}{n-(n-2)\rho}$$

with variance

$$\mathrm{Var}(\hat{E}_{\mathrm{opt}}(h)) = \frac{\sigma^2(1+\rho)}{n-(n-2)\rho}.$$

Thus,

$$\lim_{n\to\infty}[\mathrm{Var}(\hat{E}_{\mathrm{opt}}(h))/\mathrm{Var}(\hat{E}_{\mathrm{avg}}(h))] = 1,$$

which implies that the usual sample mean is as efficient as the optimal weighted estimator asymptotically.

A Note:

From Example 3.2, it is clear that the weighted estimator cannot substantially improve the simulation efficiency over the usual sample mean if the weight w_i is fixed (not random). Thus, in order to obtain a better weighted estimator, the weight w_i must be random or depend on the sample $\boldsymbol{\theta}_i$ in a particular functional form.

♠ **Partition-Weighted MC Estimation** (Chen and Shao 2001)

Let $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_n$ denote n i.i.d. random variables from $\pi(\boldsymbol{\theta}|D)$ defined on a p-dimensional Euclidean space R^p, and let h be a real-valued function.

Assume that $\mu = E[h(\boldsymbol{\theta})] \neq 0$ and $\sigma^2 = \text{Var}(h(\boldsymbol{\theta})) < \infty$, where the expectation and variance are taken with respect to the posterior distribution $\pi(\boldsymbol{\theta}|D)$.

Let $\Omega \subset R^p$ denote the support of $\pi(\boldsymbol{\theta}|D)$, and let $A_1, A_2, \ldots, A_\kappa$ be a partition of Ω such thatPartition!definition: (i) $\cup_{l=1}^{\kappa} A_1 = \Omega$; (ii) $A_l \cap A_{l^*} = \emptyset$ for $l \neq l^*$; and (iii) $\int_{A_l} \pi(\boldsymbol{\theta}|D)\, d\boldsymbol{\theta} > 0$ for $l = 1, 2, \ldots, \kappa$. Also, let

$$\mu_l = E\left[h(\boldsymbol{\theta}) 1\{\boldsymbol{\theta} \in A_l\}\right] \quad \text{and} \quad b_l = E\left[h^2(\boldsymbol{\theta}) 1\{\boldsymbol{\theta} \in A_l\}\right].$$

Chen and Shao (2001) proposed the following Partition-weighted MC "estimator":

$$\hat{E}_a(h) = \frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{\kappa} a_l h(\boldsymbol{\theta}_i) 1\{\boldsymbol{\theta}_i \in A_l\},$$

where $a = (a_1, a_2, \ldots, a_\kappa)'$ is a vector of fixed weights subject to

$$\sum_{l=1}^{\kappa} a_l \mu_l = \mu.$$

The corresponding variance of $\hat{E}_a(h)$ is given by

$$\mathrm{Var}(\hat{E}_a(h)) = \frac{1}{n}\left(\sum_{l=1}^{\kappa} a_l^2 b_l - \mu^2\right).$$

Theorem *The value of* $a = (a_1, a_2, \ldots, a_\kappa)'$ *that minimizes the variance of* $\hat{E}_a(h)$ *is given by*

$$a_{\text{opt},l} = \frac{\mu_l}{b_l} \frac{\mu}{\sum_{j=1}^{\kappa} \mu_j^2 / b_j} \quad \textit{for } l = 1, 2, \ldots, \kappa.$$

Let $a_{\text{opt}} = (a_{\text{opt},1}, a_{\text{opt},2}, \ldots, a_{\text{opt},\kappa})'$. *Then, the optimal weighted estimator* $\hat{E}_{a_{\text{opt}}}(h)$ *has variance*

$$\text{Var}(\hat{E}_{a_{\text{opt}}}(h)) = \frac{1}{n} \Big(\frac{\mu^2}{\sum_{l=1}^{\kappa} \mu_l^2 / b_l} - \mu^2 \Big),$$

and

$$\text{Var}(\hat{E}_{a_{\text{opt}}}(h)) \le \text{Var}(\hat{E}_{\text{avg}}(h)) = \frac{\sigma^2}{n}.$$

Note: this theorem indicates that this new weighted estimator can always be better than the usual sample mean for an i.i.d. sample.

Remark:

Although a_l is a fixed weight, the estimate $\hat{E}_a(h)$ indeed uses the random weights. Let

$$w_i = \sum_{l=1}^{k} a_l 1\{\theta_i \in A_l\}.$$

Then, we can rewrite $\hat{E}_a(h)$ as

$$\hat{E}_a(h) = \frac{1}{n} \sum_{i=1}^{n} w_i h(\theta_i).$$

Therefore, w_i is random, and in fact, it is a function of θ_i. This property also distinguishes our weighted estimator from a usual stratified weighted estimator such as the Horvitz-Thompson estimator (see, Thompson 1992, p. 49), in which a fixed weight is assigned to each $h(\theta_i)$.

When $\theta_1, \theta_2, \ldots, \theta_n$ are not independent, a similar result can still be obtained. We give a brief explanation as follows. Let

$$\sigma_{l,v} = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathrm{Cov}(h(\theta_i)1\{\theta_i \in A_l\}, h(\theta_j)1\{\theta_j \in A_v\})$$

and put $\Sigma = (\sigma_{l,v}, 1 \le l, v \le k)$. It is easy to see that

$$\mathrm{Var}\left(\hat{E}_a(h)\right) = \frac{1}{n^2} a'\Sigma a.$$

It can be shown that the value of $a = (a_1, a_2, \ldots, a_k)'$ that minimizes the variance of $\hat{E}_a(h)$ is given by

$$a_{opt} = \frac{\mu\Sigma^{-1}d}{d'\Sigma^{-1}d},$$

where $d = (\mu_1, \mu_2, ..., \mu_k)'$. Moreover, the optimal weighted estimator $\hat{E}_{a_{opt}}(h)$ has variance

$$\mathrm{Var}\left(\hat{E}_{a_{opt}}(h)\right) = \frac{\mu^2}{n^2 d'\Sigma^{-1}d}.$$

3.3 Simulation Standard Error Estimation

Suppose $\hat{E}(h)$ is a Monte Carlo estimator of $E(h(\boldsymbol{\theta})|D)$ using the weighted sample $\{(\boldsymbol{\theta}_i, w_i),\ i = 1, 2, \ldots, n\}$. Let $\mathrm{Var}(\hat{E}(h))$ be the variance of $\hat{E}(h)$, and let $\widehat{\mathrm{Var}}(\hat{E}(h))$ be an estimate of $\mathrm{Var}(\hat{E}(h))$. Then, the simulation standard error of $\hat{E}(h)$ is defined as

$$\mathrm{se}(\hat{E}(h)) = [\widehat{\mathrm{Var}}(\hat{E}(h))]^{1/2},$$

which is the square root of the estimated variance of the MC estimator $\hat{E}(h)$. Computing the simulation standard error is important, since it provides the magnitude of the simulation accuracy of the estimator $\hat{E}(h)$.

Since the sample generated by an MCMC sampling algorithm is often dependent, a complication that arises from the autocorrelation is that $\mathrm{Var}(\hat{E}(h))$ is difficult to obtain.

There are several methods available for obtaining a dependent sample based estimate of $\mathrm{Var}(\hat{E}(h))$.

♠ **Time Series Approach** (Geyer 1992)

Suppose that our interest is in estimating the variance of the sample mean $\hat{E}(h) = \frac{1}{n}\sum_{i=1}^{n} h(\boldsymbol{\theta}_i)$. Let $\gamma_t = \gamma_{-t} = \mathrm{Cov}(h(\boldsymbol{\theta}_i), h(\boldsymbol{\theta}_{i+t}))$ denote the lag t autocovariance of the stationary time series $\{h(\boldsymbol{\theta}_i),\ i = 1, 2, \dots\}$. The natural estimator of the lagged autocovariance γ_t is the empirical autocovariance

$$\hat{\gamma}_{n,t} = \hat{\gamma}_{n,-t} = \frac{1}{n}\sum_{i=1}^{n-t}[h(\boldsymbol{\theta}_i) - \hat{E}(h)][h(\boldsymbol{\theta}_{i+t}) - \hat{E}(h)].$$

Then, the window estimator is given by

$$\mathrm{se}_{\mathrm{win}} = \frac{\hat{\sigma}_n}{\sqrt{n}},$$

where

$$\hat{\sigma}_n^2 = \sum_{-\infty}^{\infty} w_n(t)\hat{\gamma}_{n,t},$$

and w_n is some weight function, called a lag window, satisfying $0 \le w_n(t) \le 1$, with the choice of the window depending on n.

♠ **Overlapping Batch Statistics (obs)** (Schmeiser, Avramidis and Hashem 1990)

Suppose that $\{(\boldsymbol{\theta}_i, w_i),\ i = 1, 2, \ldots, n\}$ is a dependent sample, from which a point estimator $\hat{\xi}$ of the posterior quantity of interest is computed. The obs estimate of the variance of $\hat{\xi}$ is

$$\hat{V}(m) = \left[\frac{m}{n-m}\right] \frac{\sum_{j=1}^{n-m+1} (\hat{\xi}_j - \hat{\xi})^2}{(n-m+1)},$$

where $\hat{\xi}_j$ is defined analogously to $\hat{\xi}$, but is a function of only $(\boldsymbol{\theta}_j, w_j)$, $(\boldsymbol{\theta}_{j+1}, w_{j+1})$, $\ldots$, $(\boldsymbol{\theta}_{j+m-1}, w_{j+m-1})$. Then, the simulation standard error of $\hat{\xi}$ is

$$\mathrm{se}(\hat{\xi}) = \sqrt{\hat{V}(m)}.$$

Here, m is the batch size. For many situations, choosing m so that $10 \leq n/m \leq 20$ is reasonable.

3.4 Effect of "Burn In"

Consider a bivariate normal distribution $N_2(\boldsymbol{\mu}, \Sigma)$ with

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}.$$

From Example 2.1, the marginal Gibbs path is given by

$$\theta_{1,i+1} = \mu_1 + \psi(\theta_{1,i} - \mu_1) + \sigma_1^* z_{i+1}^*,$$

for $i \geq 0$, where $\psi = \rho^2$, $\sigma_1^{*2} = \sigma_1^2(1-\rho^4)$, and $\{z_i^*,\ i \geq 0\}$ denote an i.i.d. $N(0,1)$ random variable sequence.

Let

$$\hat{\mu}_1 = \frac{1}{n} \sum_{i=n_0+1}^{n+n_0} \theta_{1,i}.$$

Then

$$\begin{aligned} E[\hat{\mu}_1] &= \mu_1 + \frac{1}{n} \sum_{i=n_0+1}^{n+n_0} \psi E(\theta_{1,i-1} - \mu_1) \\ &= \mu_1 + \frac{1}{n} \sum_{i=n_0+1}^{n+n_0} \psi^i (\theta_{1,0} - \mu_1) \\ &= \mu_1 + \frac{1}{n} \times \psi^{n_0+1} \times \frac{1-\psi^n}{1-\psi} (\theta_{1,0} - \mu_1). \end{aligned}$$

Thus, the bias due to the fixed point initialization is

$$\text{bias} = \frac{1}{n} \times \psi^{n_0+1} \times \frac{1-\psi^n}{1-\psi} (\theta_{1,0} - \mu_1).$$

Consider $\theta_{1,0} - \mu_1 = 10$ (poor starting value). We have the following results

Biases when $n = 1000$

	ρ			
n_0	0.5	0.9	0.99	0.999
50	0.0	0.0	0.18	3.91
100	0.0	0.0	0.07	3.53
500	0.0	0.0	0.02	1.59
1000	0.0	0.0	0.00	0.58

Biases when $n = 10,000$

	ρ			
n_0	0.5	0.9	0.99	0.999
50	0.0	0.0	0.02	0.45
100	0.0	0.0	0.01	0.41
500	0.0	0.0	0.00	0.18
1000	0.0	0.0	0.00	0.07

So, when n is relatively large, the effect of "burn in" is very small. However, when n is small and ρ is large, the "burn in" effect is relatively strong.

3.5 Improving MC Estimation

There are several approaches, which can improve MC estimation. The following lists some of them:

- Recycling the wasted random variables used in MCMC sampling;
  Casella and Robert (1996, 1998)
- Variance-Reduction MCMC Sampling
  Chen and Schmeiser (1991) and Tierney (1994)
- Rao–Blackwellization
  Geyer (1995), Robert (Ed.) (1998), and Robert and Casella (1999)
  Geyer (1995) pointed out that conditioning could do worse when one uses MCMC samples. See Exercise 3.7.
- Partition-Weighted MC estimation
  Chen and Shao (1999) and Peng (1998)