Bayesian Computational Methods: Monte Carlo Methods

©These notes are copyrighted by the authors. Unauthorized use is not permitted.
Monte Carlo Methods

Noniterative Monte Carlo Methods

1. Direct Sampling

Approximations to integrals can be carried out by Monte Carlo integration. The definition of Monte Carlo integration is as follows. Suppose a random variable $\theta$ has a density generically denoted by $p(\theta)$, and we wish to compute $E[f(\theta)]$. This is given by

$$\gamma = E[f(\theta)] = \int f(\theta)p(\theta)d\theta.$$ 

Then if $\theta_1, \ldots, \theta_N$ are i.i.d. samples from $p(\theta)$, we have

$$\hat{\gamma} = \frac{1}{N} \sum_{j=1}^{N} f(\theta_j),$$
Monte Carlo Methods

which converges to $E[f(\theta)]$ with probability 1 as $N \to \infty$, by the strong law of large numbers.

In Bayesian inference, $p(\theta)$ typically corresponds to the posterior distribution of $\theta$, $p(\theta|x)$, and thus $E[f(\theta)]$ is the posterior mean of $f(\theta)$.

Hence the computation of posterior expectations requires only a sample size $N$ from the posterior distribution.

Thus, we must be able to directly sample from the posterior distribution in order to use the Monte Carlo approximation. Notice that as $N$ increases, the quality of the approximation increases.
Monte Carlo Methods

The Monte Carlo size, $N$ is within our control, where as the sample size $n$ is not. Another constraint with asymptotic methods is that it allows us to evaluate its accuracy for any fixed $N$. Since $\hat{\gamma}$ is itself a sample mean of independent observations, we have

$$V ar(\hat{\gamma}) = \frac{1}{N} V ar (f(\theta)) .$$

But $V ar((f(\theta))$ can be estimated by the sample variance of the $f(\theta_j)$ values, so that a standard error estimate of $\hat{\gamma}$ is given by

$$se(\hat{\gamma}) = \sqrt{\frac{1}{N(N - 1)} \sum_{j=1}^{N} (f(\theta_j) - \hat{\gamma})^2}.$$
Monte Carlo Methods

Finally, the CLT implies that \( \hat{\gamma} \pm 2se(\hat{\gamma}) \) provides an approximate 95% interval for the true value of the posterior mean \( \gamma \).

Remark

While it may seem strange for us to recommend use of a frequentist interval estimator here, Monte Carlo simulations provide one (and perhaps only one) example where they are clearly appropriate!

In addition note that we can also estimate quantities such as

\[
p = P(a < f(\theta) < b | x)
\]

using Monte Carlo methods.
Monte Carlo Methods

A estimate of $p$ is simply

$$\hat{p} = \frac{\sum_{i=1}^{N} I(\{ f(\theta_i) \}'s \in (a, b))}{N}$$

where $N$ is the number of Monte Carlo samples.

The associated simulation standard error is

$$\text{se}(\hat{p}) = \sqrt{\frac{\hat{p}(1 - \hat{p})}{N(N - 1)}}.$$ 

This suggests that a histogram of the sampled $\theta_j$’s would estimate the posterior itself, since the probability in each histogram bin converges to the true bin probability.
Monte Carlo Methods

Alternatively, we could use a kernel density estimate to “smooth” the histogram.

\[ \hat{p}(\theta | x) = \frac{1}{N k_N} K \left( \frac{\theta - \theta_j}{k_N} \right). \]

where the \( K \) is a “kernel” density (typically a normal or rectangular distribution) and \( k_N \) is a window width satisfying \( k_N \to 0 \) and \( N k_N \to \infty \) as \( N \to \infty \).

Details of density estimation are given in a book by Silvermann (1986, Chapman Hall).
Monte Carlo Methods

Example

Suppose $x_1, \ldots, x_n$ are i.i.d. $N(\mu, \sigma^2)$, $\tau = \frac{1}{\sigma^2}$, and $\pi(\mu, \sigma^2) \propto \tau^{-1}$. Recall that

$$\mu|\tau, x \sim N\left(\bar{x}, \frac{1}{n\tau}\right),$$

$$\tau|x \sim \text{Gamma}\left(\frac{n - 1}{2}, \frac{(n - 1)s^2}{2}\right),$$

where $s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$. 

Monte Carlo Methods

We can generate samples from the posterior of \((\mu, \sigma^2)\) as follows.

1. Sample \(\tau_j \sim \text{Gamma}\left(\bar{x}, \frac{n-1}{2}, \frac{(n-1)s^2}{2}\right)\)

2. Then sample \(\mu_j \sim \text{N}\left(\bar{x}, \frac{1}{n\tau_j}\right)\)

\(j = 1, \ldots, N\).

This process creates the set \(\{(\mu_j, \sigma^2_j), j = 1, \ldots, N\}\) from \(p(\mu, \sigma^2|x)\). To estimate, for example, the posterior mean of \(\mu\), we would use

\[
\hat{E}(\mu|x) = \frac{1}{n} \sum_{j=1}^{N} \mu_j,
\]

or, to obtain a 95% HPD interval for \(\mu\), we simply use the empirical 0.025 and 0.975 quantiles of the sample of \(\mu_j\) values.
Monte Carlo Methods

Estimates of functions of parameters are also easily obtained. For example, suppose we want to estimate the posterior distribution of $\gamma = \sigma/\mu = \tau^{-1/2}\mu^{-1}$, which is known as the coefficient of variation.

We simply define the transformed Monte Carlo samples $\gamma_j = \tau_j^{-1/2}\mu_j^{-1}$, $j = 1, \ldots, N$, and create a histogram or kernel density based on these values.

As a final illustration, suppose we wanted to estimate

$$P(z > c|x) = \int_{\Theta} \left[ \int_c^{\infty} p(z|\theta)dz \right] p(\theta|x)d\theta$$

where $z$ is a future observation.
Monte Carlo Methods

Note that the inner integral can be written as

\[ P(z > c | \theta) = 1 - \Phi \left( \frac{c - \mu}{\sigma} \right) = 1 - \Phi \left( \frac{c - \mu}{\tau^{-1/2}} \right) \]

where \( \Phi(\cdot) \) is the standard normal c.d.f.

Thus

\[
P(z > c | x) = E_{\theta|x} \left[ 1 - \Phi \left( \frac{c - \mu}{\tau^{-1/2}} \right) \right] \approx \frac{1}{N} \sum_{j=1}^{N} \left[ 1 - \Phi \left( \frac{c - \mu_j}{\tau_j^{-1/2}} \right) \right].
\]
Monte Carlo Methods

2. Indirect Methods
If we cannot **directly sample** from the joint posterior, as in the previous example, then we cannot use **direct Monte Carlo**. We need to search for ways to sample from the joint posterior **indirectly**.
Several methods have been proposed in the literature for **indirect sampling**. We will discuss three of them. These are

i) importance sampling

ii) rejection sampling

iii) weighted bootstrap
Monte Carlo Methods

Importance Sampling

This approach was outlined carefully by Hammersly and Handscombe (1964, Monte Carlo Methods, Chapman and Hall) and championed for Bayesian analysis by Geweke (1989, Econometrica).

Suppose we wish to approximate a posterior expectation, say

\[
E [f(\theta)|x] = \frac{\int f(\theta)L(\theta)\pi(\theta)d\theta}{\int L(\theta)\pi(\theta)d\theta}
\]

where \(L(\theta) \propto p(x|\theta)\) is the likelihood function of \(\theta\).
Monte Carlo Methods

Suppose we can roughly approximate the normalized likelihood times the prior, \( c^{-1}L(\theta)\pi(\theta) \), by some density \( g(\theta) \), from which we can easily sample, say a multivariate \( t \) density, or perhaps some variation of a multivariate \( t \). Then by defining weight function

\[
\omega(\theta) = \frac{L(\theta)\pi(\theta)}{g(\theta)},
\]

We have

\[
E[f(\theta)|x] = \frac{\int f(\theta)\omega(\theta)g(\theta)d\theta}{\int \omega(\theta)g(\theta)d\theta} \approx \frac{1}{N} \sum_{j=1}^{N} f(\theta_j)\omega(\theta_j),
\]
Monte Carlo Methods

where the $\theta_j$ are i.i.d. samples from $g(\theta)$. Here, $g(\theta)$ is called the importance function. How closely $g(\theta)$ resembles $c^{-1}L(\theta)\pi(\theta)$ (i.e. the normalized posterior) controls how good the approximation is. To see this, note that if $g(\theta)$ is a good approximation, the weights will all be roughly equal, which in turn will minimize the variance of the numerator and denominator in the approximation. If $g(\theta)$ is a poor approximation, many of the weights will be close to zero, and thus few $\theta_j$’s will dominate the sums, producing an inaccurate approximation.
Monte Carlo Methods

Rejection Sampling

Rejection sampling is an extremely general and quite common method of random variate generation. Excellent summaries are given in the books by Ripley (1987, John Wiley and Sons) and Devroye (1986, Springer-Verlag).

In this method, instead of trying to approximate the normalized posterior

\[
p(\theta|x) = \frac{L(\theta)\pi(\theta)}{\int L(\theta)\pi(\theta)d\theta} = \frac{L(\theta)\pi(\theta)}{c}
\]
Monte Carlo Methods

We try to “blanket” it. That is, suppose there exists an identifiable constant $M > 0$ and a smooth density $g(\theta)$, called an envelope function, such that

$$L(\theta)\pi(\theta) < M g(\theta)$$

for all $\theta$.

The rejection method proceeds as follows:

i) Generate $\theta_j \sim g(\theta)$

ii) Generate $U \sim U(0, 1)$

iii) if $M U g(\theta_j) < L(\theta_j)\pi(\theta_j)$, accept $\theta_j$, otherwise, reject.

iv) Return to step i) and repeat until the desired number of samples are $N$ obtained.
Monte Carlo Methods

The members of \( \{ \theta_j, j = 1, \ldots, N \} \) will then be variables from \( p(\theta|x) \).

Intuition suggests that \( M \) should be chosen as small as possible, so as not to waste samples unnecessarily. This is easy to confirm, since if \( k \) denotes the number of iterations required to get one acceptable candidate \( \theta_j \), then \( k \) is a geometric random variable. That is,

\[
P(k = i) = (1 - p)^{i-1} p,
\]

where \( p \) is the probability of acceptance.
Monte Carlo Methods

So $P(k = i)$ decreases monotonically and at a n exponential rate. It can be shown that $p = c/M$. where $c = \int L(\theta)\pi(\theta)d\theta$. Since $E(k) = p^{-1} = M/c$, we do need to minimize $M$.

Note that if $p(\theta|x)$ were available as the $g$ function, we would choose the minimum acceptable value $M = c$, obtaining an acceptance probability of 1.

The envelope density $g$ should be similar to the posterior in general appearance, but with heavier tails and sharper infinite peaks, in order to assure that there are sufficiently many rejection candidates available across its entire domain.
Monte Carlo Methods

Weighted Bootstrap

This method was presented by Smith and Gelfand (1992, American Statistician) and is very similar to the sampling-importance sampling resampling algorithm of Rubin (1988). Suppose an $M$ appropriate for the rejection is not readily available, but we do have a sample $\theta_1, \ldots, \theta_N$ from some approximating density $g(\theta)$. Define

$$\omega_i = \frac{L(\theta_i) \pi(\theta_i)}{g(\theta_i)},$$

$$q_i = \frac{\omega_i}{\sum_{i=1}^{N} \omega_i}.$$
Monte Carlo Methods

Now draw $\theta^*$ from the **discrete** distribution over $\{\theta_1, \ldots, \theta_N\}$, which places mass $q_i$ at $\theta_i$. Then the sample is from $\pi(\theta|x)$ with the approximation improving as $N \to \infty$.

This is a **weighted bootstrap**, since instead of resampling from the set with equally likely probabilities of selection, we are resampling some points more than others due to the unequal weighting.

To see why this method works, note that for the standard bootstrap,
Monte Carlo Methods

\[ P(\theta^* \leq a) = \sum_{i=1}^{n} \frac{I(-\infty < \theta_i \leq a)}{N} \]

\[ \rightarrow \int_{-\infty}^{a} g(\theta) d\theta \]

as \( N \to \infty \), so that \( \theta^* \) is approximately distributed as \( g(\theta) \). For the weighted bootstrap

\[ P(\theta^* \leq a) = \sum_{i=1}^{n} q_i I(-\infty < \theta_i \leq a) \]

\[ = \frac{1}{N} \sum_{i=1}^{n} \omega_i I(-\infty < \theta_i \leq a) \]

\[ = \frac{1}{N} \sum_{i=1}^{n} \omega_i \]

Bayesian Computational Methods – p.22/??
Monte Carlo Methods

\[
E_g \left[ \frac{L(\theta_i)\pi(\theta_i)}{g(\theta)} I(-\infty < \theta_i \leq a) \right] \rightarrow \frac{E_g \left[ \frac{L(\theta_i)\pi(\theta_i)}{g(\theta)} \right]}{\int_{-\infty}^{\infty} \frac{L(\theta_i)\pi(\theta_i)}{g(\theta)} g(\theta) d\theta}
\]

as \( N \rightarrow \infty \), and

\[
\begin{align*}
\int_{-\infty}^{a} & \left[ \frac{L(\theta_i)\pi(\theta_i)}{g(\theta)} g(\theta) \right] d\theta \\
= & \int_{-\infty}^{\infty} \left[ \frac{L(\theta_i)\pi(\theta_i)}{g(\theta)} g(\theta) \right] d\theta \\
= & \int_{-\infty}^{a} [L(\theta_i)\pi(\theta_i)] d\theta \\
= & \int_{-\infty}^{\infty} [L(\theta_i)\pi(\theta_i)] d\theta \\
= & \int_{-\infty}^{a} p(\theta|x) d\theta
\end{align*}
\]
Monte Carlo Methods

Similar to importance sampling, we need to choose
\( g(\theta) \approx p(\theta|x) \), or else a very large \( N \) will be required to obtain acceptable accuracy.
Monte Carlo Methods

Markov Chain Monte Carlo Methods

Importance sampling, rejection sampling, and the weighted boot-strap are all noniterative methods. They draw a sample of size $N$, and stop. Hence there is no notation of the algorithm “converging” - we simply require sufficiently large $N$. But for many problems, especially high-dimensional ones, it may be difficult of even impossible to find an importance sampling density (or envelope function) which is an acceptable accurate approximation to the log posterior, but still easy to sample from. This facilitates a need for new algorithms.
Monte Carlo Methods

Gibbs Sampling

Suppose we have a collection of $k$ random variables denoted by $U = U_1, \ldots, U_k$. We assume the full conditional distributions

$$\{p(u_i|u_j, j \neq i), i = 1, \ldots, k\}$$

are available for sampling. Here, “available” means that the samples may be generated by some method. Thus, we do not require the one dimensional conditional distributions $[U_i|U_j, j \neq i], i = 1, \ldots, k$ to have closed form, but only that we need to be able to write them up to a normalizing constant.
Monte Carlo Methods

Gibbs Sampling Algorithm

1. Suppose we have a set of arbitrary starting values \( \{U_1^{(0)}, \ldots, U_k^{(0)}\} \).

2. Draw \( U_1^{(1)} \) from \( [U_1|U_2^{(0)}, \ldots, U_k^{(0)}] \)

3. Draw \( U_2^{(1)} \) from \( [U_2|U_1^{(1)}, U_3^{(0)}, \ldots, U_k^{(0)}] \)

   \[ \vdots \]

4. Draw \( U_k^{(1)} \) from \( [U_k|U_1^{(1)}, \ldots, U_{k-1}^{(1)}] \)

This completes one iteration of the Gibbs sampler. Thus after 1 iteration, we have \( \{U_1^{(1)}, \ldots, U_k^{(1)}\} \), and after \( t \) iterations we obtain \( \{U_1^{(t)}, \ldots, U_k^{(t)}\} \).
Monte Carlo Methods

Remark:
For notational convenience, we let \([X|Y]\) denote the conditional distribution of \(X\) given \(Y\). We are led to the following theorem.

**Theorem**

For the Gibbs sampling algorithm outlined above,

a) \(\{U_1^{(t)}, \ldots, U_k^{(t)}\} \rightarrow [U_1, \ldots, U_k]\) as \(t \rightarrow \infty\).

b) The convergence in a) is **exponential** in \(t\).

Monte Carlo Methods

In the context of Bayesian analysis, we are interested in sampling from the joint posterior distribution \([U_1, \ldots, U_k | x]\). The Gibbs sampler then requires draws from each of the univariate conditional distributions \(p(u_i | u_j, x, i \neq j)\).

A marginal density estimate of \(U_i\) is given by

\[
\hat{p}(u_i | x) = \frac{1}{m} \sum_{j=1}^{m} p(u_i | u_{1,j}^{(t)}, u_{i-1,j}^{(t)}, \ldots, u_{i+1,j}^{(t)}, \ldots, u_{k,j}^{(t)}, x).
\]

This type of estimate has less variability than an estimate obtained by a kernel density.
Monte Carlo Methods

Example
Suppose $x_1, \ldots, x_n$ are i.i.d. $N(\mu, \sigma^2)$, $\tau = 1/\sigma^2$, and
$\pi(\mu, \tau) \propto \tau^{-1}$. We want to devise the Gibbs sampler to sample
from the joint posterior density of $(\mu, \tau)$. We have

$$
\mu | \tau, x \sim N \left( \bar{x}, \frac{1}{n\tau} \right),
$$

$$
\tau | \mu, x \sim \text{Gamma} \left( \frac{n}{2}, \frac{\sum(x_i - \mu)^2}{2} \right),
$$
Monte Carlo Methods

To obtain the joint posterior samples from $[\mu, \tau | x]$ we

1. Start with arbitrary $(\mu^{(0)}, \tau^{(0)})$.
2. Sample $\mu^{(1)}$ from $[\mu | \tau^{(0)}, x]$.
3. Sample $\tau^{(1)}$ from $[\tau | \mu^{(1)}, x]$.

Repeat until we have $t$ iterations yielding

$(\mu^{(1)}, \tau^{(1)}), \ldots, (\mu^{(t)}, \tau^{(t)})$. 

Monte Carlo Methods

Example: Logistic Regression

Suppose $y_1, \ldots, y_n$ are independent Binomial$(1, p_i)$, where

$$p_i = \frac{\exp \{\beta_0 + \beta_1 x_i\}}{1 + \exp \{\beta_0 + \beta_1 x_i\}},$$

$i = 1, \ldots, n$. We have

$$L(\beta) = \exp \left[ \sum_{i=1}^{n} \left\{ y_i (\beta_0 + \beta_1 x_i) - \log(1 + \exp\{\beta_0 + \beta_1 x_i\}) \right\} \right].$$

How do we set up the Gibbs sampler for sampling from the joint posterior of $[\beta_0, \beta_1 | x, y]$? The univariate conditionals do not have closed form.
Monte Carlo Methods

\[ p(\beta_o | \beta_1, x, y) \propto L(\beta), \]

where \( \beta_1 \) is fixed. Now

\[ p(\beta_1 | \beta_o, x, y) \propto L(\beta), \]

where \( \beta_o \) is treated as fixed. Note that the univariate conditionals for \( \beta_o \) and \( \beta_1 \) have the exact same functional form, with different variables being treated as fixed. In general, the univariate conditional distribution is always proportional to the joint distribution.
Monte Carlo Methods

That is, $\theta = (\theta_1, ..., \theta_k)$, then

$$p(\theta_i|\theta_j, x, i \neq j) \propto L(\theta)\pi(\theta),$$

$i = 1, ..., k$. For the logistic regression problem, since the univariate conditionals do not have closed form. We need another algorithm to sample from these univariate conditionals. In many situations, we need to use rejection sampling within Gibbs sampling, see Gelfand and Smith (1990), Gilks et al. (1993, JRSS-B), Gelfand et al. (1990, JASA), Wakefield et al. (1994, Appl. Stat.).
Monte Carlo Methods

Metropolis-Hastings Algorithm

Like the Gibbs sampling algorithm, the Metropolis-Hastings algorithm is a MCMC (Markov Chain Monte Carlo) method. The originator of the algorithm is Metropolis et al. (1953, *Chem. Phys.*). Hastings (1970, *Biometrika*) introduced this algorithm for statistical problems.

Suppose we wish to sample from the joint distribution $[U] = [U_1, ..., U_k]$. Denote the density of $U$ by $p(U)$. 
Monte Carlo Methods

Let $q(u, v)$ be a density in $u$ such that $q(u, v) = q(v, u)$. The function $q$ is called a candidate or proposal density. We generate values as follows

**Metropolis Algorithm**

1. Draw $v \sim q(\cdot, u)$, where $u = U^{t-1}$, the current state of the Metropolis algorithm.
2. Compute the odds ratio

$$r = \frac{p(v)}{p(u)} = \frac{L(v)\pi(v)}{L(u)\pi(u)}$$

3. Set $U^t = v$ with probability $\min(r, 1)$ and $U^t = u$ with probability $1-\min(r, 1)$. 
Monte Carlo Methods

Theorem
For the Metropolis algorithm, under mild conditions, \([U^t] \to [U]\) as \(t \to \infty\).

Remark: The Gibbs sampling algorithm and the Metropolis-Hastings algorithm are **Markov Chain Monte Carlo** algorithms since the samples that are generated by these algorithms are samples from a certain **Markov chain**.
Monte Carlo Methods

For continuous parameter settings, the most convenient choice for $q$ is a $N \left( \theta^{(t-1)}, \tilde{\Sigma} \right)$, where $\tilde{\Sigma}$ can be taken as

$$
\tilde{\Sigma} = - \left[ \frac{\partial^2 \log[p^*(\theta|x)]}{\partial \theta \partial \theta'} \right]^{-1} \bigg|_{\theta = \theta^{(t-1)}}
$$

where $p^*(\theta|x) = L(\theta)\pi(\theta)$. We note that this choice of $q$ is easily sampled and clearly symmetric in $\nu$ and $\theta^{(t-1)}$.

A simple but important generalization of the Metropolis algorithm was provided by Hastings (1970, *Biometrika*). Hastings drops the requirement that $q(u, \nu)$ be symmetric, and redefined the odds ratio as
Monte Carlo Methods

\[ r = \frac{p(v)q(u, v)}{p(u)q(v, u)} \]

With this modification, it can be shown that this algorithm converges to the required “target” distribution for any candidate density \( q \). Chib and Greenburg (1995, *American Statistician*) give an excellent discussion of the Metropolis-Hastings algorithms and discuss choices of \( q(u, v) \).
Monte Carlo Methods

Example

Consider for data by Bliss (1935, *Ann. of Appl. Bio*.). The data record the number of adult beetles killed after 5 hours of exposure to various levels of gaseous carbon disulphide (CS$_2$).

<table>
<thead>
<tr>
<th>Dosage $w_i$</th>
<th>no. killed $y_i$</th>
<th>no. exposed $n_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.6907</td>
<td>6</td>
<td>59</td>
</tr>
<tr>
<td>1.7242</td>
<td>13</td>
<td>60</td>
</tr>
<tr>
<td>1.7552</td>
<td>18</td>
<td>62</td>
</tr>
<tr>
<td>1.7842</td>
<td>28</td>
<td>56</td>
</tr>
<tr>
<td>1.8113</td>
<td>52</td>
<td>63</td>
</tr>
<tr>
<td>1.8369</td>
<td>53</td>
<td>59</td>
</tr>
<tr>
<td>1.8601</td>
<td>61</td>
<td>62</td>
</tr>
<tr>
<td>1.8839</td>
<td>60</td>
<td>60</td>
</tr>
</tbody>
</table>
Monte Carlo Methods

Consider the model

\[ P(\text{Death} | w_i) \equiv h(w_i) = \left[ \frac{\exp(x_i)}{1 + \exp(x_i)} \right]^{m_1} \]

where \( m_1 > 0 \), \( w_i \) is the covariate (dose), and \( x_i = \frac{w_i - \mu}{\sigma} \), for \( \mu \in \mathbb{R}^1 \) and \( \sigma > 0 \). For the prior distributions we take

\[ m_1 \sim \text{Gamma}(a_o, b_o) \]
\[ \mu \sim \text{N}(c_o, d_o) \]
\[ \sigma^2 \sim \text{IG}(e_o, f_o) \]

the Inverse Gamma distribution.
Monte Carlo Methods

Note that is $z \sim IG(a, b)$, then

$$p(z|a, b) \propto z^{-(a+1)}e^{-\frac{b}{2z}}$$

and $z^{-1} \sim \text{Gamma}(a, b)$. We take $m_1, \mu, \sigma^2$ to be independent \textit{a priori}. The joint posterior is given by

$$p(m_1, \mu, \sigma^2|y) \propto p(y|m_1, \mu, \sigma^2)\pi(m_1, \mu, \sigma^2)$$

$$\propto \left\{ \prod_{i=1}^{k}[h(w_i)]^{y_i}[1 - h(w_i)]^{n_i-y_i} \right\}$$

$$\times \frac{m_1^{a_o-1}}{(\sigma^2)^{e_o+1}}\exp\left\{ -\frac{1}{2} \left( \frac{\mu - c_o}{d_o} \right)^2 - \frac{m_1}{b_o} - \frac{1}{f_o\sigma^2} \right\}$$
Monte Carlo Methods

Let $\theta = (\theta_1, \theta_2, \theta_3) = (\mu, \frac{1}{2} log(\sigma^2), log(m_1))$. This transforms the parameter space to $R^3$. This will be nice if we want to work with a **Gaussian proposal density**. Upon making this transformation, we get

$$p(\theta|y) \propto \left\{ \prod_{i=1}^{k} [h(w_i)]^{y_i} [1 - h(w_i)]^{n_i-y_i} \right\} \times \exp\{a_o \theta_3 - 2e_o \theta_2\} \times \exp\left\{ -\frac{1}{2} \left( \frac{\theta_1 - c_o}{d_o} \right)^2 - \frac{\exp(\theta_3)}{b_o} - \frac{\exp(-2\theta_2)}{f_o} \right\}$$
Monte Carlo Methods

Numerical stability is improved by working on the log-scale for computing the Metropolis odds ratio as

\[ r = \exp[\log p^*(v|y) - \log p^*(\theta^{(t-1)}|y)]. \]

The hyperparameters are chosen to be \( a_o = 0.25, b_o = 4 \), so that \( m_1 \) has a prior mean equal to 1 (corresponding to the standard logit model), and prior standard deviation 2. Vague priors are specified for \( \mu \) and \( \sigma^2 \), by setting \( c_o = 2, d_o = 10, e_o = 2 \) and \( f_o = 1000 \). We use a \( N_3 \left( \theta^{(t-1)}, \tilde{\Sigma} \right) \) proposal density, with

\[ \tilde{\Sigma} = Diag(0.00012, 0.033, 0.10). \]
Monte Carlo Methods

The chains mix very slowly, as can be seen from the high lag 1 sample autocorrelations, estimated from the chain 2 output and printed above the monitoring plots. The reason for this slow convergence is the high posterior correlations amongst the parameters, estimated as $\hat{\rho}(\theta_1, \theta_2) = -0.78$, $\hat{\rho}(\theta_1, \theta_3) = -0.94$, and $\hat{\rho}(\theta_2, \theta_3) = -0.89$. As a result the proposal acceptance rate is low (13.5%), and convergence is slow. Convergence can be accelerated by using a nondiagonal proposal covariance matrix designed to better mimic the posterior surface.
Monte Carlo Methods

From the output of the initial chains, we obtain a better estimate of the posterior covariance matrix in the usual way as

$$\hat{\Sigma} = \frac{1}{N} \sum_{j=1}^{N} (\theta_j - \bar{\theta})(\theta_j - \bar{\theta})',$$

where $j$ indexes the Monte Carlo samples. Now consider $\tilde{\Sigma} = 2\hat{\Sigma}$. The results indicate improved convergence, with lower observed autocorrelations and higher Metropolis acceptance rate (27.3%).
Monte Carlo Methods

Convergence Monitoring and Acceleration

When we say that an MCMC algorithm has converged as iteration $T$, we mean that its output can be safely thought of as coming from the true stationarity distribution of the Markov chain or all $t > T$.

The most common source of MCMC convergence difficulty is due to model overparameterization. When a model becomes overparameterized, the parameters become nonidentifiable. As an example, consider the problem of finding the posterior distribution of $\theta_1$, where the likelihood of $x_i$ is defined by

$$x_i \sim N(\theta_1 + \theta_2, 1),$$
Monte Carlo Methods

for \( i = 1, \ldots, n \), and we take \( \pi(\theta_1, \theta_2) \propto 1 \). Here, only the sum of the two parameters is identified by the data, so without proper priors for \( \theta_1 \) and \( \theta_2 \), their marginal posterior distribution is improper as well. Unfortunately, a naive Gibbs sampler would not reveal this problem. In addition, models that are overparameterized typically lead to high posterior correlations amongst the parameters, or cross-correlations. Traditional remedy: reparameterize, Gelfand et al. (1995). Alternatively, convergence may be monitored by a diagnostic statistic.
Monte Carlo Methods

Characteristics of diagnostic statistic.

1. Theoretical - based on some expected criterion
2. Diagnostic Goal, e.g. reducing variance and bias
3. Output format, e.g. graphical
4. Replication, e.g. comparing replicate chains
5. Dimensionality, considering different aspects of the posterior sampling domain
6. Application depending on the algorithm
7. Ease of Use, i.e. interpretability and computational efficiency
Monte Carlo Methods

Convergence Diagnostic Statistics The single most popular approach is due to Gelman and Rubin (1992, Stat. Scien.). In this approach we

1. Run $m$ parallel chains from different starting values.
2. These chains must be initially overdispersed with respect to the true posterior.
3. Running the $m$ chains for $2N$ iterations each, we then compare the variation within each chain to the variation between each chain. Specifically we monitor convergence by the scaled reduction factor,
Monte Carlo Methods

\[ \sqrt{\hat{R}} = \sqrt{\frac{N - 1}{N}} + \frac{m + 1}{mN} \left( \frac{B}{W} \right) \left( \frac{df}{df - 2} \right), \]

where \( B/N \) is the variance between the means from the \( m \) parallel chains, \( W \) is the average of the \( m \) within-chain variances, and \( df \) is the degrees of freedom of an approximating \( t \) density to the posterior distribution.

Gelman and Rubin (1992) show that \( \hat{R} \to 1 \) as \( N \to 1 \), with a value near one suggesting good convergence.
Monte Carlo Methods

This approach has been criticized because

1. It’s a univariate diagnostic, and must be applied to each parameter separately.

2. The approach focuses solely on the bias component of convergence, with no information on the accuracy.

3. The method relies heavily on the user’s ability to find a starting distribution that is actually overdispersed with respect to the true posterior distribution, a condition we cannot really check without knowledge of the latter.