CHAPTER III

MARKOV CHAIN
MONTE CARLO ALGORITHMS
CHAPTER III

MARKOV CHAIN MONTE CARLO ALGORITHMS

3.0 INTRODUCTION

The goal of this chapter is to provide important concepts of some of the sampling algorithms. Sampling methods, which rely on Markov chain theory, are iterative. The principle is to build a succession of states and once convergence is reached, the consecutive states are assumed to be drawn from the target probability distribution. With these methods, it is possible to sample from general probability distributions, where as direct sampling algorithms only apply to specific probability distributions such as the Gaussian distribution. Especially, the probability distribution can be posterior distribution in a Bayesian context, which makes MCMC methods very attractive in Bayesian computation. In next section different sampling algorithms are discussed and compared.

3.1 STANDARD MCMC UPDATING SCHEME

The common approach to Markov chain theory is to start with some transition distribution modeling some process of interest, to determine conditions under which there is an invariant or stationary
distribution and then to identify the form of that limiting distribution. MCMC methods involve the solution of the inverse of this problem whereby the stationary distribution is known, and it is the transition distribution that needs to be identified, though in practice there may be infinitely many distributions to choose from. We shall denote the Markov chain transition distribution (or transition kernel) by $K$, so that, if the chain is at present in state $x$, then the conditional distribution of the next state of the chain $y$, given the present state, is denoted by $K(x,y)$.

The main theorem underpinning the MCMC method is that any chain, which is irreducible and aperiodic, will have a unique stationary distribution and that the $t$-step transition kernel will converge to that stationary distribution as $t \to \infty$ (Meyn and Tweedie, 1993). Thus to generate a chain with stationary distribution $\pi$, we need only to find transition kernels $K$ that satisfy these conditions and for which $\pi K = \pi$ i.e., $K$ is such that given an observation $x \sim \pi(x)$, if $y \sim K(x,y)$, then $y \sim \pi(y)$.

A Markov chain with stationary distribution $\pi$ is called time reversible if its transition kernel $K$ is such that it exhibits detailed balance i.e.,

$$\pi(x)K(x,y) = \pi(y)K(y,x)$$

(3.1.1)
This essentially means that the chain would run it either forward in time or backward. The behavior of reversible chains is well understood, and it is a desirable property for any MCMC transition kernel to have, since any transition kernel for which equation (3.1.1) holds will have stationary distribution \( \pi \).

Traditionally the MCMC literature has talked only in terms of MCMC samplers and algorithms. However, this unnecessarily restricts attention to Markov chains based on only a single form of transition kernel. In practice it is often more sensible to combine a number of different transition kernels to construct a Markov chain which performs well. Thus it is perhaps more appropriate to discuss not MCMC algorithms but MCMC updates or transitions and this is the approach that we adapt here. In practice the manner in which different transitions are combined is by splitting the state vector of the Markov chain into a number of distinct components and by using different transition kernels to update each component.

3.2 BURNING IN THE SAMPLER

An important issue in the successful implementation of Metropolis-Hastings or any other MCMC sampler is the number of runs (steps) until the chain approaches stationarity (the length of the burn-in period). The first 1000 to 5000 elements are thrown out, and
then one of the various convergence tests is used to assess whether
stationarity has been reached.

A poor choice of starting values and proposal distribution can
greatly increase the required burn-in time, and an area of much
current research is whether an optimal starting point and proposal
distribution can be found. Now we give some basic rules. One
suggestion for the starting value is to start the chain as close to the
center of the distribution as possible, for example taking a value close
to the distribution's mode (such as using an approximate MLE as the
starting value).

A chain is said to be poorly mixing if it stays in small regions of
the parameter space for long periods of time, as opposed to a well
mixing chain that seems to happily explore the space. A poorly mixing
chain can arise because the target distribution is multimodal and our
choice of starting values traps us near one of the modes (such
multimodal posteriors can arise if we have a strong prior in conflict
with the observed data). Two approaches have been suggested for
situations where the target distribution may have multiple peaks. The
most straightforward is to use multiple highly dispersed initial values
to start several different chains (Gelman and Rubin 1992). A less
obvious approach is to use simulated annealing on a single-chain.
3.3 JUMPING DISTRIBUTION

Since the Metropolis sampler works with any symmetric distribution, while Hasting-Metropolis is even more general, what are our best options for proposal distributions? There are two general approaches – random walks and independent chain sampling. Under a sampler using proposal distribution based on a random walk chain, the new value $y$ equals the current value $x$ plus a random variable $z$,

$$y = x + z$$  \hspace{1cm} (3.3.1)

In this case,

$$q(x, y) = g(y - x) = g(z)$$  \hspace{1cm} (3.3.2)

the density associated with the random variable $z$. If

$$g(z) = g(-z)$$  \hspace{1cm} (3.3.3)

i.e, the density for the random variable $z$ is symmetric as occurs with a normal or multivariate normal with mean zero, or a uniform centered around zero, then we can use Metropolis sampling as

$$q(x, y)/q(y, x) = g(z)/g(-z) = 1$$  \hspace{1cm} (3.3.4)

The variance of the proposal distribution can be thought of as a tuning parameter that can be adjusted to get better mixing.
Under a proposal distribution using an independent chain, the probability of jumping to point \( y \) is independent of the current position \( x \) of the chain, i.e.,

\[
q(x, y) = g(y). \tag{3.3.5}
\]

Thus the candidate value is simply drawn from a distribution of interest, independent of the current value. Again, any number of standard distributions can be used for \( g(y) \). Here we note that the proposal distribution is generally not symmetric, as \( g(x) \) is generally not equal to \( g(y) \), and Metropolis-Hasting sampling must be used.

The proposal distribution can be tuned to adjust the mixing, and in particular the acceptance probability, of the chain. This is generally done by adjusting the standard deviation (SD), of the proposal distribution. For example, by adjusting the variance (or the eigen values of the covariance matrix) for a normal (or multivariate normal), increasing or decreasing the range \((-a, a)\), if a uniform is used, or changing the degrees of freedom (df) if a \( \chi^2 \) is used (variance increasing with the \( df \)). To increase the acceptance probability, one decreases the proposal distribution SD (Draper 2000). Draper also noted a tradeoff in that if the SD is too large, moves are large (which is good), but are not accepted often. This leads to high autocorrelation and very poor mixing, requiring much longer chains. If the proposal SD is too small, moves are generally accepted (high acceptance
probability), but they are also small, again generating high autocorrelations and poor mixing.

3.4 CONVERGENCE DIAGNOSTICS

Note that we have still not answered the question of how to determine whether the sampler has reached its stationary distribution. Further, given that members in a Metropolis-Hasting sample are very likely correlated, how does this affect use of the sequence for estimating parameters of interest from the distribution? These issues are partly addressed in the subsequent sections.

3.4.1 Auto Correlation and Sample Size Inflation

The adjacent members from a Metropolis-Hastings sequence are expected to be positively correlated, and the nature of correlation can be quantified by using an autocorrelation function. Consider a sequence \((\theta_1, \ldots, \theta_n)\) of length \(n\). Correlations can occur between adjacent members \(\rho(\theta_i, \theta_{i+1}) \neq 0\), and more generally between more distant members \(\rho(\theta_i, \theta_{i+k}) \neq 0\). The \(k\)th order autocorrelation \(\rho_k\) can be estimated by

\[
\rho_k = \frac{\text{Cov}(\theta_i, \theta_{i+k})}{\text{Var}(\theta_i)} = \frac{\sum_{i=1}^{n-k} (\theta_i - \bar{\theta})(\theta_{i+k} - \bar{\theta})}{\sum_{i=1}^{n-k} (\theta_i - \bar{\theta})^2},
\]

with

\[
\bar{\theta} = \frac{1}{n} \sum_{i=1}^{n} \theta_i
\]
An important result from the theory of time series analysis is that if the $\theta_t$ are from a stationary (and correlated) process, correlated draws still provide an unbiased picture of the distribution provided the sample size is sufficiently large.

Some indication of the required sample size comes from the theory of a first order autoregressive process (or AR1), where

$$\theta_t = \mu + \alpha(\theta_{t-1} - \mu) + \epsilon$$  \hspace{1cm} (3.4.3)

where $\epsilon$ is the noise, that is $\epsilon \sim N(0, \sigma^2)$. Here $\rho_1 = \alpha$ and the $k$th order autocorrelation is given by $\rho_k = \rho_1^k$. Under this process, $E(\tilde{\theta}) = \mu$ with standard error

$$SE(\tilde{\theta}) = \frac{\sigma}{\sqrt{n}} \sqrt{\frac{1 + \rho}{1 - \rho}}$$  \hspace{1cm} (3.4.4)

The first ratio is the standard error for the noise, while the second ratio $\sqrt{(1 + \rho)/(1 - \rho)}$ is the sample size inflation factor, which shows how the autocorrelation inflates the sampling variance.

### 3.4.2 Test For Convergence

One should always look at the time series trace, the plot of the random variables being generated versus the number of iterations. In addition to showing evidence for poor mixing, such traces can also suggest a minimum burn-in period for some starting value. For example, suppose the trace moves very slowly away from the initial
value to a rather different value (say after 5000 iterations) around which it appears to settle down. Clearly, the burn-in period is at least 5000 in this case. It must be cautioned that the actual time may be far longer than suggested by the trace. Nevertheless, the trace often indicates that the burn-in is still not complete.

Two other graphs that are very useful in accessing a MCMC sampler look at the serial autocorrelations as a function of the time lag. A plot of $a_k$ vs. $k$ (the $k$th order autocorrelation vs. the lag) should show geometric decay is the sampler series closely follows an AR1 model. A plot of the partial autocorrelations as a function of lag is also useful. The $k$th partial autocorrelation is the excess correlation not accounted for by a $(k-1)^{th}$ order autoregressive model (AR$_{k-1}$). Hence, if the first order model fits, the second order partial autocorrelation is zero, as the lagged autocorrelations are completely accounted for the AR1 model ($\rho_k = \rho_1^k$). Both of these autocorrelation plots may indicate underlying correlation structure in the series not obvious from the time series trace.

There are formal tests available to test for stationarity of the sampler after a given point. Two important tests are considered below. Additional diagnostic checks for stationarity are discussed by Geyer (1992), Gelman and Rubin (1992), Raftery and Lewis (1992b), and Robert (1995).
The Geweke test (Geweke 1992) splits sample after removing a burn-in period into two parts: say the first 10% and last 50%. If the chain is at stationarity, the means of the two samples should be equal. A modified z-test can be used to compare the two subsamples, and the resulting test statistic is often referred to as a Geweke z-score. A value larger than 2 indicates that the mean of the series is still drifting, and a longer burn-in is required before monitoring the chain to begin.

A more informative approach is the Raftery-Lewis test (Raftery and Lewis 1992). Here, one specifies a particular quantile \( q \) of the distribution of interest (typically 2.5% and 97.5%, to give a 95% confidence interval), an accuracy \( \varepsilon \) of the quantile, and a power \((1-\beta)\) for achieving this accuracy on the specified quantile. With these three parameters set, the Raftery-Lewis test breaks the chain into a \((1,0)\) sequence, 1 if \( \theta_i \leq q \), zero otherwise. This generates a two-state Markov chain, and the Raftery-Lewis test uses the sequence to estimate the transition probabilities. With these probabilities in hand, one can then estimate the number of addition burn-ins (if any) required to approach stationarity, the thinning ratio (how many points should be discarded for each sampled point) and the total chain length required to achieve the preset level of accuracy.
3.4.3 One Long Chain or Many Smaller Chains

One can either use a single long chain (Geyer 1992, Raftery and Lewis 1992) or multiple chains each starting from different initial values (Gelman and Rubin 1992). Note that with parallel processing machines, using multiple chains may be computationally more efficient than a single long chain. Geyer, however, argues that using a single longer chain is the best approach. If long burn-in periods are required, or if the chains have very high autocorrelations, using a number of smaller chains may result in each not being long enough to be of any value. Applying the diagnostic tests discussed above can resolve some of these issues for any particular sampler.

3.5 THE GIBBS SAMPLER

The Gibbs sampler (introduced in the context of image processing Geman and Geman 1984), is a special case of Metropolis-Hastings sampling wherein the random value is always accepted (i.e., \( \alpha = 1 \)). The task remains to specify how to construct a Markov chain whose values converge to the target distribution. The key to the Gibbs sampler is that one only considers univariate conditional distributions - the distribution when all of the random variables but one are assigned fixed values. Such conditional distributions are far easier to simulate than complex joint distributions and usually have simple forms (often being normals, inverse \( \chi^2 \), or other common prior
distributions). Thus, one simulates $n$ random variables sequentially from the $n$ univariate conditionals rather than generating a single $n$-dimensional vector in a single pass using the full joint distribution.

To introduce the Gibbs sampler, consider a bivariate random variable $(x,y)$, and suppose we wish to compute one or both marginals, $p(x)$ and $p(y)$. The idea behind the sampler is that it is far easier to consider a sequence of conditional distributions $p(x/y)$ and $p(y/x)$, than it is to obtain the marginal by integration of the joint density $p(x,y)$, e.g., $p(x) = \int p(x,y) dy$. The sampler starts with initial value $y_0$ for $y$ and obtains $x_0$ by generating a random variable from the conditional distribution $p(x/y = y_0)$. The sampler then uses $x_0$ to generate a new value of $y_1$, drawing from the conditional distribution based on the value $x_0$, $p(y/x = x_0)$. The sampler proceeds as follows

\begin{align}
    x_i &\sim p(x/y = y_{i-1}) \\
    y_i &\sim p(y/x = x_i)
\end{align} 

(3.5.1) \hspace{1cm} (3.5.2)

Repeating this process $k$ times, generates a Gibbs sequence of length $k$, where a subset of points $(x_j,y_j)$ for $1 \leq j \leq m < k$ are taken as our simulated draws from the full joint distribution. One iteration of all the univariate distributions is often called a scan of the sampler. To obtain the desired total of $m$ sample points (here each point on the sampler is a vector of the two parameters), one samples the chain after a sufficient burn-in to removal the effects of the initial sampling
values and at set time points (say every $n$ samples) following the burn-in. The Gibbs sequence converges to a stationary distribution that is independent of the starting values, and by construction this stationary distribution is the target distribution we are trying to simulate (Tierney 1994).

When more than two variables are involved, the sampler is extended in the obvious fashion. In particular, the value of the $k$th variable is drawn from the distribution $p(\theta^{(k)}/\Theta^{(-k)})$ where $\Theta^{(-k)}$ denotes a vector containing all the variables but $k$. Thus, during the $i$th iteration of the sample, to obtain the value of $\theta_i^{(k)}$ we draw from the distribution

$$\theta_i^{(k)} \sim p(\theta^{(k)} | \theta^{(1)} = \theta_i^{(1)}, \ldots, \theta^{(k-1)} = \theta_i^{(k-1)}, \theta^{(k+1)} = \theta_{i-1}^{(k+1)}, \ldots, \theta^n = \theta_{i-1}^{(n)})$$

For example, if there are four variables, $(w, x, y, z)$, the sampler becomes

$$w_i \sim p(w | x = x_{i-1}, y = y_{i-1}, z = z_{i-1})$$
$$x_i \sim p(x | w = w_i, y = y_{i-1}, z = z_{i-1})$$
$$y_i \sim p(y | w = w_i, x = x_i, z = z_{i-1})$$
$$z_i \sim p(z | w = w_i, x = x_i, y = y_i)$$

Gelfand and Smith (1990) illustrated the power of the Gibbs sampler to address a wide variety of statistical issues, while Smith and Roberts (1993) showed the importance of the Gibbs sampler in
Bayesian statistics (obtaining posterior distributions). A nice introduction to the sampler is given by Casella and George (1992), while further details can be found in Tanner (1996), Besag et al. (1995), and Lee (1997). Finally, note that the Gibbs sampler can be thought of as a stochastic analog to the Expectation-Maximization (EM) approaches used to obtain likelihood functions when missing data are present. In the sampler, random sampling replaces the expectation and maximization steps.

3.5.1 Gibbs Sampler and Marginal Distributions

Any feature of interest for the marginals can be computed from the m realizations of the Gibbs sequence. For example, the expectation of any function f of the random variable x is approximated by

$$E[f(x)]_m = \frac{1}{m} \sum_{i=1}^{m} f(x_i)$$  \hspace{1cm} (3.5.3)

This is the Monte-Carlo (MC) estimate of f(x), as $E[f(x)]_m \to E[f(x)]$ as $m \to \infty$. Likewise, the MC estimate for any function of n variables $(\theta^{(1)}, ..., \theta^{(n)})$ is given by

$$E[f(\theta^{(1)}, ..., \theta^{(n)})]_m = \frac{1}{m} \sum_{i=1}^{m} f(\theta^{(1)}_i, ..., \theta^{(n)}_i)$$  \hspace{1cm} (3.5.4)

While computing the MC estimate of any moment using the sampler is straightforward, computing the actual shape of the marginal density is
slightly more involved. While one might use the Gibbs sequence of (say) $x_i$ values to give a rough approximation of the marginal distribution of $x$, this turns out to be inefficient, especially for obtaining the tails of the distribution. A better approach is to use the average of the conditional densities $p(x/y = y_i)$, as the functional form of the conditional density contains more information about the shape of the entire distribution than the sequence of individual realizations $x_i$ (Gelfand and Smith 1990, Liu et al. 1991). Since

$$p(x) = \int p(x/y) \, p(y) \, dy = E_y [p(x/y)]$$  \hfill (3.5.5)

One can approximate the marginal density using

$$\hat{p}_m(x) = \frac{1}{m} \sum_{i=1}^{m} p(x/y = y_i)$$  \hfill (3.5.6)

### 3.5.2 Monte Carlo Variance of Gibbs-Sampler Estimate

Suppose we are interested in using an appropriately thinned and burned-in Gibbs sequence $\theta_1, ..., \theta_n$ to estimate some function $h(\theta)$ of the target distribution, such as a mean, variance, or specific quantile (Cumulative probability value). Since we are drawing random variables, there is some sampling variance associated with the Monte Carlo estimate

$$\hat{h} = \frac{1}{n} \sum_{i=1}^{n} h(\theta_i)$$  \hfill (3.5.7)
By increasing the length of the chain (increasing \( n \)), we can decrease the sampling variance of \( \hat{h} \), but it would be nice to have some estimate of the size of this variance. One direct approach is to run several chains and use the between-chain variance in \( \hat{h} \). Specifically, if \( \hat{h}_j \) denotes the estimate for chain \( j \) \((1 \leq j \leq m)\) where each of the \( m \) chains has the same length, then the estimated variance of the Monte Carlo estimate is

\[
Var(\hat{h}) = \frac{1}{n-1} \sum_{j=1}^{m} (\hat{h}_j - \hat{h}^*)^2
\]

where

\[
\hat{h}^* = \frac{1}{n} \sum_{j=1}^{m} \hat{h}_j
\]

Using only a single chain, an alternative approach is to use results from the theory of time series. Estimate the lag – \( k \) auto covariance associated with \( h \) by

\[
\hat{\gamma}(k) = \frac{1}{n} \sum_{i=1}^{n-k+1} \left[ h(\theta_i) - \hat{h} \right] \left[ h(\theta_{i+k}) - \hat{h} \right]
\]

This is natural generalization of the \( k \)th order autocorrelation to the random variable generated by \( h(\theta) \). The resulting estimate of the Monte Carlo variance is

\[
Var(\hat{h}) = \frac{1}{n} \left( \hat{\gamma}(0) + 2 \sum_{i=1}^{k} \hat{\gamma}(i) \right)
\]
Here $\delta$ is the smallest positive integer satisfying $\hat{\gamma}(2\delta) + \hat{\gamma}(2\delta+1) > 0$ (i.e., the higher order (lag) auto covariances are zero).

One measure of the effects of autocorrelation between elements in the simpler is the effective chain size,

$$\hat{n} = \frac{\hat{\gamma}(0)}{\text{Var}(\hat{\gamma})} \quad (3.5.12)$$

In the absence of autocorrelation between members, $\hat{n} = n$.

### 3.5.3 Convergence Diagnostics: The Gibbs Stopper

The basic idea behind the Gibbs stopper is to assign the weight $w$ to the vector $\theta = (\theta_1, ..., \theta_d)$ which has been drawn from the current approximate to the point density $g_i$ via

$$w(\theta) = \frac{q(\theta_1, ..., \theta_d)}{g_i(\theta_1, ..., \theta_d)} \quad (3.5.13)$$

where $q(\theta_1, ..., \theta_d/Y)$ is proportional to the posterior density $p(\theta_1, ..., \theta_d/Y)$. As $g_i$ converges towards $p(\theta_1, ..., \theta_d/Y)$, the distribution of the weights should converge towards a spike distribution. This observation has been found useful in assessing convergence of the Gibbs sampler, as well as in transforming a sample from $g_i$ into a sample from the exact distribution (Ritter and Tanner, 1992). Our discussion of the various diagnostics for Metropolis-Hastings (MH)
also applies to Gibbs sampler, as Gibbs is a special case of MH. As with MH sampling, we can reduce the autocorrelation between monitored points in the sampler sequence by increasing the thinning ratio (increasing the number of points discarded between each sampled point). Draper (2000) noted that the Gibbs sampler usually produces chains with smaller autocorrelations than other MCMC samplers.

Tanner (1996) discussed an approach for monitoring convergence based on the Gibbs stopper, in which weights based on comparing the Gibbs sampler and the target distribution are computed and plotted as a function of the sampler iteration number. As the sampler approaches stationary, the distribution of the weights is expected to spike. More details can be seen from Tanner (1996).

3.6 THE METROPOLIS-HASTINGS ALGORITHM

One problem with applying Monte Carlo integration is in observing samples from some complex probability distribution $p(x)$. Attempts to solve this problem are the roots of MCMC methods. In particular, they trace to attempts by mathematical physicists to integrate very complex functions by random sampling (Metropolis and Ulam 1949, Metropolis et al., 1953, Hastings 1970) and the resulting Metropolis-Hastings algorithm. A detailed review of this method is given by Chib and Greenberg (1995)
Suppose our goal is to draw samples from some distribution \( P(\theta) \) where \( P(\theta) = f(\theta)/K \), where the normalizing constant \( K \) may not be known and very difficult to compute. The Metropolis algorithm generates a sequence of draws from this distribution as follows;

1. Start with any initial value \( \theta_0 \) satisfying \( f(\theta_0) > 0 \)

2. Using current \( \theta \) value, sample a candidate point \( \theta' \) from some jumping distribution \( q(\theta_1, \theta_2) \) which is the probability of returning a value \( \theta_2 \) given a previous value of \( \theta_1 \). This distribution is also referred to as the proposal or candidate-generating distribution. The only restriction on the jump density in the Metropolis algorithm is that it is symmetric, i.e., \( q(\theta_1, \theta_2) = q(\theta_2, \theta_1) \).

3. Given the candidate point \( \theta' \), calculate the ratio of the density at the candidate (\( \theta' \)) and current (\( \theta_{t-1} \)) points

\[
\alpha = \frac{p(\theta')}{p(\theta_{t-1})} = \frac{f(\theta')}{f(\theta_{t-1})}
\]

Because we are considering the ratio of \( p(x) \) under two different values, the normalizing constant \( K \) cancels out

4. If the jump increases the density (\( \alpha > 1 \)), accept the candidate point (set \( \theta_t = \theta' \)) and return to step 2. If the jump decreases the
density \((a < 1)\) then with probability \(a\) accept the candidate point, else reject it and return to step 2. We can summarize the Metropolis sampling as first computing

\[
\alpha = \min \left( \frac{f(\theta^*)}{f(\theta_{t-1})} \right)
\]  

(3.6.2)

and then accepting a candidate point with probability \(\alpha\) (the probability of a move). This generates a Markov chain \((\theta_0, \theta_1, \ldots, \theta_k, \ldots)\), as the transition probabilities from \(\theta_t\) to \(\theta_{t+1}\) depends only on \(\theta_t\) and not on \((\theta_0, \ldots, \theta_{t-1})\). Following a sufficient burn-in period (of, say, \(k\) steps), the chain approaches its stationary distribution and samples from the vector \((\theta_{k+1}, \ldots, \theta_{k+n})\) are samples from \(p(x)\).

Hastings (1970) generalized the Metropolis algorithm by using an arbitrary transition probability function \(q(\theta_1, \theta_2) = P(\theta_1 \rightarrow \theta_2)\), and setting the acceptance probability for a candidate point as

\[
\alpha = \min \left( \frac{f(\theta^*)q(\theta^*, \theta_{t-1})}{f(\theta_{t-1})q(\theta_{t-1}, \theta^*)} \right)
\]  

(3.6.3)

Assuming that the proposal distribution is symmetric, i.e., \(q(x, y) = q(y, x)\), recovers the original Metropolis algorithm.

### 3.6.1 Metropolis-Hasting Sampling as a Markov Chain

To demonstrate that the Metropolis-Hasting sampling generates a Markov chain whose equilibrium density is that candidate density
$p(x)$, it is sufficient to show that the Metropolis-Hasting transition kernel satisfy the detailed balanced equation \[ P(j \rightarrow k)\pi_j^* = P(k \rightarrow j)\pi_k^* \]
with \( p(x) \).

Under the Metropolis-Hasting algorithm, draw sample from \( q(x,y) = P(x \rightarrow y/q) \) and then accept the move with probability \( \alpha(x,y) \), so that the transition probability kernel is given by

\[
P(x \rightarrow y/q) = q(x,y)\alpha(x,y) = q(x,y) \cdot \min \left[ \frac{p(y)q(y,x)}{p(x)q(x,y)}, 1 \right] \tag{3.6.4}
\]

Thus if the Metropolis-Hasting kernel satisfies

\[
P(x \rightarrow y)p(x) = P(y \rightarrow x)p(y), \tag{3.6.5}
\]

or \( q(x,y)\alpha(x,y)p(x) = q(y,x)\alpha(y,x)p(y), \) for all \( x,y \) \tag{3.6.6}

then that stationary distribution from this kernel corresponds to draws from the target distribution. We show that the balance equation is indeed satisfied with this kernel by considering the three possible cases for any particular \((x, y)\) pair

1. \( q(x,y)p(x) = q(y,x)p(y) \). Here \( \alpha(x,y) = \alpha(y,x) = 1 \) implying

\[
P(x,y)p(x) = q(x,y)p(x) \\
P(y,x)p(y) = q(y,x)p(y) \\
P(x,y)p(x) = P(y,x)p(y) \tag{3.6.7}
\]

showing that the detailed balance equation holds
2. $q(x, y)p(x) > q(y, x)p(y)$, in which case

$$\alpha(x, y) = \frac{p(y)q(y, x)}{p(x)q(x, y)}$$ and $\alpha(y, x) = 1$

Hence

$$P(x, y)p(x) = q(x, y)\alpha(x, y)p(x)$$

$$= q(y, x)p(y) = q(y, x)\alpha(y, x)p(y)$$

$$= P(y, x)p(y) \quad (3.6.8)$$

3. $q(x, y)p(x) < q(y, x)p(y)$. Here

$$\alpha(x, y) = 1 \quad \text{and} \quad \alpha(y, x) = \frac{q(x, y)p(x)}{q(y, x)p(y)}$$

Hence

$$P(y, x)p(y) = q(y, x)\alpha(y, x)p(y)$$

$$= q(y, x)\left(\frac{q(x, y)p(x)}{q(y, x)p(y)}\right)p(y)$$

$$= q(x, y)p(x) = q(x, y)\alpha(x, y)p(x)$$

$$= P(x, y)p(x). \quad (3.6.9)$$
3.7 THE METROPOLIS-HASTINGS-GREEN ALGORITHM

All of the schemes described so far used to have separate theory. All are special cases of the Metropolis-Hastings-Green (MHG) algorithm (Green, 1995), which is essentially Metropolis-Hastings with measure functions rather than densities. Green was not the first to present an algorithm of this type. Multigrid methods in statistical physics are special cases, as are the methods for point processes presented by Geyer and Moller (1994), but Green gave the first general formulation. The MHG algorithm is best understood by comparison with the Metropolis-Hastings algorithm. The unnormalized density \( h \) is replaced by an unnormalized measure \( \pi \) on the state space \( \Omega \). The proposal density \( q(x, y) \) is now replaced by a proposal kernel \( Q(x, A) \).

We also need a symmetric measure \( \epsilon \) on \( \Omega \times \Omega \) to play the role played by \( \mu \times \mu \) in the ordinary Metropolis-Hastings algorithm. The measure must \( \epsilon \) dominate \( \pi(dx)Q(x,dy) \) so that there is a Radon-Nikodym derivative

\[
 f(x,y) = \frac{\pi(dx)Q(x,dy)}{\epsilon(dx,dy)} \tag{3.7.1}
\]

which replaces \( h(x)q(x,y) \) in the ordinary Metropolis-Hastings algorithm. Then the Hastings ratio becomes 'Green's ratio'

\[
 R = \frac{f(y,x)}{f(x,y)} \tag{3.7.2}
\]
The algorithm is

1. Simulate $y \sim Q(x, \cdot)$.

2. Evaluate Green's ratio (3.7.2)

3. Accept $y$ with probability $\min(1, R)$.

Green (1995) also proposed a more general form of the algorithm using state-dependent mixing. There are a finite or infinite set of proposal kernels $Q_i(x, A)$, $i \in I$, which are permitted to be substochastic. The requirements on the proposal kernels are $Q_i(x, \Omega)$ is known for all $i$.

$$
\sum_{i \in I} Q_i(x, \Omega) \leq 1, \quad \forall x \in \Omega
$$

For all $i$

$$
f_i(x, y) = \frac{\pi(dx)Q_i(x, dy)}{e_i(dx, dy)} \quad (3.7.3)
$$

measure $e_i$ may be used for each $i$.

For each $x$ and $i$, it is possible to simulate realizations from the distribution having is known, and it is possible to evaluate $f_i(x, y)$ for all $x$ and $y$. A different symmetric the normalized proposal distribution

$$
P_i(x, \cdot) = \frac{Q_i(x, \cdot)}{Q_i(x, \Omega)} \quad (3.7.4)
$$
1. Choose a proposal kernel $Q_i$ with probability $p_i(x) = Q_i(x, \Omega)$. With probability $1 - \sum p_i(x)$, skip the remaining steps and stay at $x$.

2. Simulate $y \sim P_i(x, \cdot)$ defined by (3.7.4).

3. Evaluate Green's ratio

$$R = \frac{f_i(y, x)}{f_i(x, y)}$$  \hfill (3.7.5)

where $f_i$ is defined by (3.7.3).

4. Accept $y$ with probability $\min(1, R)$.

All of this is the MHG algorithm described in preceding section combined with the idea of state-dependent mixing.

3.8 GLM AND BAYESIAN MODEL – AN EMPIRICAL COMPARISON

In this section we empirically evaluate and compare the performance of generalized linear model (GLM) and a hierarchical Bayesian model by analyzing the multiple indicator survey (MICS) data. The data is pertaining to 25 states and 7 Union territories of India (DWCD/UNCF, 2001). The data is used only for illustration. The MCMC is used to simulate the Hierarchical Bayesian model.
Several plausible transformations of the original data led to a GLM, which fits the data well. After resampling new data a Hierarchical Bayesian model is applied and the simulation results through MCMC is found to reconcile very well with the above model. MCMC model seems to provide better fit for the data than the other results.

3.8.1 Data Description

The MICS-2000 is a household survey designed to provide valuable information on indicators related to health, nutrition, education, water and sanitation. It was developed in collaboration with WHO, The UN Statistical Division, London school of hygiene and Tropical Medicine and CDC Atlanta, USA. A total of 53 surveys at the state level were conducted involving all the states and union territories. In all nearly 1,20,000 households were canvassed in 3260 clusters in rural and urban areas. So MICS-2000 compliments existing data from other sources including the Census of India 2001 and National Family Health Survey II. Other details of the survey can be found in MICS-2000 India summary report (2001).

For our comparison we have taken the orphanage related children and health related variables such as sanitation, health, drinking water, vaccines, immunization, fertility, literacy for all the states and union territories. We have considered the orphan rate
children, which reasonably assume a Poisson process. The variables considered for illustration of the methods are given below:

\[ X_1 \text{ (orprate)} \quad - \quad \text{Rate of orphan (Number of children below age 15 years with both parents dead/1000)} \]

\[ X_2 \text{ (matrag)} \quad - \quad \text{mean age at marriage} \]

\[ X_3 \text{ (tfr)} \quad - \quad \text{Total fertility rate} \]

\[ X_4 \text{ (pc5ill)} \quad - \quad \text{Percentage children < 5 years suffered from illness} \]

\[ X_5 \text{ (pcmanti)} \quad - \quad \text{Percentage mothers aged 15-49 years Antenatal} \]

\[ X_6 \text{ (pcinsidel)} \quad - \quad \text{Percentage institutional deliveries} \]

\[ X_7 \text{ (pccontrac)} \quad - \quad \text{Percentage currently married women aged 15-49 years using any contraceptive} \]

\[ X_8 \text{ (pchivknow)} \quad - \quad \text{Percentage women aged 15-49 years who have heard of HIV/AIDS} \]

\[ X_9 \text{ (pclmmuzi)} \quad - \quad \text{Percentage children aged 12-23 months fully Immunized} \]

\[ X_{10} \text{ (pchousewat)} \quad - \quad \text{Percentage households using improved source of drinking water} \]

\[ X_{11} \text{ (pcimprosan)} \quad - \quad \text{Percentage households having improved sanitation facility} \]

\[ X_{12} \text{ (pciodised)} \quad - \quad \text{Percentage households using iodized salt for Cooking} \]

\[ X_{13} \text{ (pcvaccine)} \quad - \quad \text{Percentage children aged 12-23 months who received various vaccines} \]

\[ X_{14} \text{ (pclitt7)} \quad - \quad \text{Percentage literate among population aged \geq 7 Years} \]

\[ X_{15} \text{ (pclitt15)} \quad - \quad \text{Percentage literate among population aged \geq 15 Years} \]
The paired picture given in Fig.3.1 shows the relationship among the variables considered. We have used the orphanage rate $R$ directly as well as $X_N$ which stands for absolute number of orphan children below 15 years. Empirically a Poisson distribution can describe $X_N$. We know that $\sqrt{X_N}$ has normal errors. We fitted the full model using $Y = \sqrt{X_N}$ and found that $R^2 = 0.622$ which is greater than $R^2 = 0.583$ where $X$ as the dependent variable.
We also considered the principal component analysis (PCA) and factor analysis method to get the overall impression about the importance of each variable. Although PCA and factor analysis may give a rough understanding of the data structure, the components thus obtained are not easy to interpret. Hence we used criteria $C_p$, AIC, BIC to select the subset.

The model is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_{15} X_{15} + \varepsilon \quad (3.8.1)$$

The various PCA results are presented in table 3.1 - 3.4.

**Table 3.1: PCA-Component Matrix**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>orprise</td>
<td>-.331</td>
</tr>
<tr>
<td>matrage</td>
<td>.745</td>
</tr>
<tr>
<td>tfr</td>
<td>-.804</td>
</tr>
<tr>
<td>pc5ill</td>
<td>-.431</td>
</tr>
<tr>
<td>pcmanti</td>
<td>.822</td>
</tr>
<tr>
<td>pcinsdel</td>
<td>.876</td>
</tr>
<tr>
<td>pccontrac</td>
<td>.473</td>
</tr>
<tr>
<td>pchivknow</td>
<td>.877</td>
</tr>
<tr>
<td>pclmmuzi</td>
<td>.797</td>
</tr>
<tr>
<td>pchousewat</td>
<td>.106</td>
</tr>
<tr>
<td>pcimprosan</td>
<td>.665</td>
</tr>
<tr>
<td>pciodised</td>
<td>.169</td>
</tr>
<tr>
<td>pcvaccine</td>
<td>.797</td>
</tr>
<tr>
<td>pclitt7</td>
<td>.891</td>
</tr>
<tr>
<td>pclitt15</td>
<td>.882</td>
</tr>
</tbody>
</table>
Table 3.2: Rotated Component Matrix – Varimax

<table>
<thead>
<tr>
<th>Component</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>orprate</td>
<td>.080</td>
<td>-.506</td>
<td>-.176</td>
</tr>
<tr>
<td>matrage</td>
<td>.856</td>
<td>.284</td>
<td>-.201</td>
</tr>
<tr>
<td>tfr</td>
<td>-.556</td>
<td>-.430</td>
<td>-.572</td>
</tr>
<tr>
<td>pc5ill</td>
<td>-.525</td>
<td>-.025</td>
<td>-.233</td>
</tr>
<tr>
<td>pcmanti</td>
<td>.278</td>
<td>.851</td>
<td>.203</td>
</tr>
<tr>
<td>pcinsdel</td>
<td>.365</td>
<td>.845</td>
<td>.193</td>
</tr>
<tr>
<td>pccontrac</td>
<td>.184</td>
<td>.241</td>
<td>.836</td>
</tr>
<tr>
<td>pchivknow</td>
<td>.669</td>
<td>.595</td>
<td>.018</td>
</tr>
<tr>
<td>pclmmuzi</td>
<td>.220</td>
<td>.893</td>
<td>.140</td>
</tr>
<tr>
<td>pchousewat</td>
<td>-.281</td>
<td>.186</td>
<td>.802</td>
</tr>
<tr>
<td>pcimprosan</td>
<td>.887</td>
<td>.121</td>
<td>-.153</td>
</tr>
<tr>
<td>pciodised</td>
<td>.684</td>
<td>-.472</td>
<td>.094</td>
</tr>
<tr>
<td>pcvaccine</td>
<td>.219</td>
<td>.894</td>
<td>.138</td>
</tr>
<tr>
<td>pclitt7</td>
<td>.833</td>
<td>.426</td>
<td>.095</td>
</tr>
<tr>
<td>pclitt15</td>
<td>.852</td>
<td>.409</td>
<td>.046</td>
</tr>
</tbody>
</table>

Table 3.3: Component Transformation Matrix

<table>
<thead>
<tr>
<th>Component</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>.693</td>
<td>.688</td>
<td>.215</td>
</tr>
<tr>
<td>2</td>
<td>.693</td>
<td>-.552</td>
<td>-.464</td>
</tr>
<tr>
<td>3</td>
<td>-.201</td>
<td>.471</td>
<td>-.859</td>
</tr>
</tbody>
</table>
Table 3.4: Communalities

<table>
<thead>
<tr>
<th></th>
<th>Initial</th>
<th>Extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>orprate</td>
<td>1.000</td>
<td>.294</td>
</tr>
<tr>
<td>matrage</td>
<td>1.000</td>
<td>.854</td>
</tr>
<tr>
<td>tfr</td>
<td>1.000</td>
<td>.821</td>
</tr>
<tr>
<td>pc5ill</td>
<td>1.000</td>
<td>.330</td>
</tr>
<tr>
<td>pcmanti</td>
<td>1.000</td>
<td>.843</td>
</tr>
<tr>
<td>pcinsdel</td>
<td>1.000</td>
<td>.885</td>
</tr>
<tr>
<td>pccontrac</td>
<td>1.000</td>
<td>.790</td>
</tr>
<tr>
<td>pchivknow</td>
<td>1.000</td>
<td>.802</td>
</tr>
<tr>
<td>pclmmuzi</td>
<td>1.000</td>
<td>.866</td>
</tr>
<tr>
<td>pchouselwat</td>
<td>1.000</td>
<td>.756</td>
</tr>
<tr>
<td>pcimprosan</td>
<td>1.000</td>
<td>.825</td>
</tr>
<tr>
<td>pciodised</td>
<td>1.000</td>
<td>.700</td>
</tr>
<tr>
<td>pcvaccine</td>
<td>1.000</td>
<td>.866</td>
</tr>
<tr>
<td>pclitt7</td>
<td>1.000</td>
<td>.886</td>
</tr>
<tr>
<td>pclitt15</td>
<td>1.000</td>
<td>.896</td>
</tr>
</tbody>
</table>

3.8.2 \( C_p \), AIC and BIC

The \( C_p \) statistic initially described by Mallows is an estimate of the standardized total MSE of estimation for the current set of data. It is computed as

\[
 C_p = \frac{ss(Res)_p}{s^2} + 2p' - n \tag{3.8.2}
\]

where \( ss(Res)_p \) is the Rss from the \( p \)-variable subset model being considered and \( s^2 \) is an estimation of \( \sigma^2 \), either from independent information or more commonly, from the model containing all
independent variables. The Akaike Information Criterion (AIC) (Akaike, 1973) is computed as

$$AIC(p') = n \ln(ss(Res)p) + 2p' - n \ln(n) \quad (3.8.3)$$

and Schwarz Bayesian Criterion SBC (or BIC) (Raftery, 1996) is computed as

$$BIC (p') = n \ln(ss(Res)_p) + [\ln(n)]p' - n \ln(n) \quad (3.8.4)$$

AIC value is significantly higher than BIC. The GLM was applied to the data and the Cp, AIC and BIC were calculated. The descriptive statistics, regression coefficients and the contrasts using GLM are presented in tables 3.5 - 3.6.
Table 3.5: Descriptive Statistics

<table>
<thead>
<tr>
<th></th>
<th>Range</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Mean Statistic</th>
<th>Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>32</td>
<td>0</td>
<td>7.0</td>
<td>2.34</td>
<td>.279</td>
</tr>
<tr>
<td>$X_2$</td>
<td>32</td>
<td>14.6</td>
<td>21.7</td>
<td>18.147</td>
<td>.2902</td>
</tr>
<tr>
<td>$X_3$</td>
<td>32</td>
<td>1.69</td>
<td>4.75</td>
<td>2.8963</td>
<td>.14825</td>
</tr>
<tr>
<td>$X_4$</td>
<td>32</td>
<td>27.0</td>
<td>60.6</td>
<td>43.131</td>
<td>1.5922</td>
</tr>
<tr>
<td>$X_5$</td>
<td>32</td>
<td>36.8</td>
<td>99.5</td>
<td>74.172</td>
<td>3.4785</td>
</tr>
<tr>
<td>$X_6$</td>
<td>32</td>
<td>12</td>
<td>96</td>
<td>45.86</td>
<td>4.258</td>
</tr>
<tr>
<td>$X_7$</td>
<td>32</td>
<td>14</td>
<td>71</td>
<td>47.83</td>
<td>2.387</td>
</tr>
<tr>
<td>$X_8$</td>
<td>32</td>
<td>12.7</td>
<td>92.8</td>
<td>54.059</td>
<td>4.0335</td>
</tr>
<tr>
<td>$X_9$</td>
<td>32</td>
<td>12.6</td>
<td>87.0</td>
<td>51.519</td>
<td>3.8908</td>
</tr>
<tr>
<td>$X_{10}$</td>
<td>32</td>
<td>38.7</td>
<td>99.2</td>
<td>78.413</td>
<td>2.8943</td>
</tr>
<tr>
<td>$X_{11}$</td>
<td>32</td>
<td>16.3</td>
<td>94.7</td>
<td>52.247</td>
<td>4.2744</td>
</tr>
<tr>
<td>$X_{12}$</td>
<td>32</td>
<td>23.5</td>
<td>97.5</td>
<td>63.241</td>
<td>3.6992</td>
</tr>
<tr>
<td>$X_{13}$</td>
<td>32</td>
<td>12.6</td>
<td>87.8</td>
<td>51.544</td>
<td>3.8982</td>
</tr>
<tr>
<td>$X_{14}$</td>
<td>32</td>
<td>52</td>
<td>92</td>
<td>70.58</td>
<td>2.017</td>
</tr>
<tr>
<td>$X_{15}$</td>
<td>32</td>
<td>45.5</td>
<td>92.2</td>
<td>66.556</td>
<td>2.2203</td>
</tr>
</tbody>
</table>
Table 3.6: GLM-Parameter Estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>B</th>
<th>Std Error</th>
<th>T</th>
<th>Sig.</th>
<th>(95% Confidence interval)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>29.227</td>
<td>11.961</td>
<td>2.444</td>
<td>.026</td>
<td>(3.992, 54.463)</td>
</tr>
<tr>
<td>X2</td>
<td>-.444</td>
<td>.425</td>
<td>-.045</td>
<td>.311</td>
<td>(-1.342, .453)</td>
</tr>
<tr>
<td>X3</td>
<td>-2.278</td>
<td>.874</td>
<td>-2.67</td>
<td>.018</td>
<td>(-4.122, -.434)</td>
</tr>
<tr>
<td>X4</td>
<td>-.061</td>
<td>.035</td>
<td>-.749</td>
<td>.098</td>
<td>(-.134, .013)</td>
</tr>
<tr>
<td>X5</td>
<td>.001</td>
<td>.039</td>
<td>.020</td>
<td>.984</td>
<td>(-.082, .084)</td>
</tr>
<tr>
<td>X6</td>
<td>-.090</td>
<td>.029</td>
<td>-.140</td>
<td>.006</td>
<td>(-.150, -.029)</td>
</tr>
<tr>
<td>X7</td>
<td>-.079</td>
<td>.041</td>
<td>-.921</td>
<td>.072</td>
<td>(-.166, .008)</td>
</tr>
<tr>
<td>X8</td>
<td>-.055</td>
<td>.025</td>
<td>-.188</td>
<td>.043</td>
<td>(-.108, -.002)</td>
</tr>
<tr>
<td>X9</td>
<td>4.593</td>
<td>2.989</td>
<td>1.537</td>
<td>.143</td>
<td>(-1.713, 10.899)</td>
</tr>
<tr>
<td>X10</td>
<td>-.040</td>
<td>.024</td>
<td>-.668</td>
<td>.114</td>
<td>(-.090, .011)</td>
</tr>
<tr>
<td>X11</td>
<td>.028</td>
<td>.023</td>
<td>1.232</td>
<td>.235</td>
<td>(-.020, .077)</td>
</tr>
<tr>
<td>X12</td>
<td>-.037</td>
<td>.021</td>
<td>-.796</td>
<td>.090</td>
<td>(-.081, .006)</td>
</tr>
<tr>
<td>X13</td>
<td>-4.537</td>
<td>2.975</td>
<td>-.525</td>
<td>.146</td>
<td>(-10.815, 1.741)</td>
</tr>
<tr>
<td>X14</td>
<td>-.144</td>
<td>.270</td>
<td>-.532</td>
<td>.601</td>
<td>(-.714, .426)</td>
</tr>
<tr>
<td>X15</td>
<td>.189</td>
<td>.251</td>
<td>.751</td>
<td>.463</td>
<td>(-.341, .718)</td>
</tr>
</tbody>
</table>
3.8.3 Hierarchical Bayesian Model

We carried out the Bayesian simulation based on the model with 6 variables.

\[ \sqrt{XN} = \beta_0 + \beta_3 X_3 + \beta_4 X_4 + \beta_6 X_6 + \beta_7 X_7 + \beta_8 X_8 + \beta_{12} X_{12} + \epsilon \]  (3.8.5)

The Bayesian model can be represented as

\[ Y^2 = XN, \frac{Y^2}{\lambda} = P_0(\lambda) \]

\[ \log(\lambda) = \beta_0 + \beta_3 X_3 + \beta_4 X_4 + \beta_6 X_6 + \beta_7 X_7 + \beta_8 X_8 + \beta_{12} X_{12} + \epsilon \] (3.8.6)

Although we employ lognormal distribution for \( \lambda \) is just empirical, one could as well adopt other distribution that have the similar shape e.g gamma distribution. Using BUGS package, (Spiegelhalter et al., 1999 and Thomas 1994), MCMC algorithm, we obtain the following results. The data is further explored in the chapter V.

**Table 3.8 : MCMC Measures**

<table>
<thead>
<tr>
<th>Measure</th>
<th>( \lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Quartile</td>
<td>0.283</td>
</tr>
<tr>
<td>Median</td>
<td>0.634</td>
</tr>
<tr>
<td>3rd Quartile</td>
<td>0.946</td>
</tr>
<tr>
<td>Mean</td>
<td>0.587</td>
</tr>
</tbody>
</table>
In summary, we have applied GLM and a Hierarchical Bayesian model to the orphan rate data. The results of GLM are compared with MCMC (Ehrlich, 1973 and Vandaele, 1978). The MCMC methods provide the best fit and better explanation of the data for making optimum decisions. In the subsequent chapters we provide a detailed comparison of MCMC algorithm and their application with special reference to Bayesian inference.