A Primer on Markov Chain Monte Carlo

Jeff Gill, jgill@polisci.ufl.edu

1 Who is Markov and What is He Doing with Chains?

The use of Markov Chain Monte Carlo (MCMC) methods to evaluate integral quantities has exploded over the last fifteen years. Beginning roughly with the publication of Geman and Geman's (1984) introduction of the Gibbs sampler as a method for obtaining difficult posterior quantities in the process of image restoration and the subsequent key integrating article by Gelfand and Smith (1990), the rate of publication of MCMC works has grown exponentially. While this is a recent development, the genesis dates back to the 1953 essay by Metropolis, et al. The lack of recognition of the importance of this contribution is a testament either to the barriers that may exist between statistical physics and other fields, or a lack of recognition by Bayesians that statistical computation could answer some of their most difficult questions.

The primary distinction made in this chapter is between standard Monte Carlo simulation methods and the *Markov Chain* type of Monte Carlo methods which are characterized by a dependence structure between consecutive simulated values. Standard Monte Carlo methods produce a set of independent simulated values according to some desired probability distribution. MCMC methods produce chains in which each of the simulated values is mildly dependent on the preceding value. The basic principle is that once this chain has run sufficiently long enough it will find its way to the desired posterior distribution of interest and we can summarize this distribution by then letting the chain wander around, thus producing summary statistics from recorded values.

1.1 What is a Markov Chain?

The initial definition required is that of a primitive concept that underlies Markov chains. A *stochastic process* is a consecutive set of random quantities defined on some known state space, Θ, indexed such that the order is known: $\{\theta^{[t]}: t \in T\}$. Frequently, but not necessarily, T is the set of positive integers implying consecutive, even-spaced time intervals: $\{\theta^{[t=0]}, \theta^{[t=1]}, \theta^{[t=2]}, \ldots\}$. With MCMC we are concerned only with this restricted type of stochastic process. A stochastic process must also be defined with respect to a *state space*, Θ, which identifies the range of possible values of θ. This state space is either discrete or continuous depending on how the variable of interest is measured, but the implications for our purposes apply more to notation than to fundamental theory. Standard references on schochastic processes include: Doob (1953/1990), Hoel, Port and Stone (1987), Karlin and Taylor (1981, 1990), and Ross (1996)

A *Markov chain* is a stochastic process with the property that any specified state in the series, $\theta^{[t]}$, is dependent only the previous value of the chain, $\theta^{[t-1]}$, and is therefore conditionally independent of all other previous values: $\theta^{[0]}, \theta^{[1]}, \ldots, \theta^{[t-2]}$. This can be stated more formally:

$$P(\theta^{[t]} \in A | \theta^{[0]}, \theta^{[1]}, \ldots, \theta^{[t-2]}, \theta^{[t-1]}) = P(\theta^{[t]} \in A | \theta^{[t-1]}) \tag{1}$$

where A is any identified set (an event or range of events) on the complete state space. So a Markov chain wanders around the state space remembering only where it has been in the last period. This property turns out to be enormously useful in generating samples from desired limiting distributions of the chain because when the chain eventually finds the region of the state space with the highest density it will the produce a sample from this distribution that is only mildly non-independent. These are the sample values that we will then use to describe the posterior distribution of interest.

A fundamental concern is the transition process that defines the probabilities of moving to other points in the state space given the current location of the chain. The most convenient way to think about this structure is to define the *transition kernel*, K, as a general mechanism for describing the probability of moving to some other specified state based on the current chain status (Robert and Casella 1999, p.141). The advantage of this notation is that it subsumes both the continuous state space case as well as the discrete state space case. It is required that $K(\theta, A)$ be a defined probability measure for all θ points in the state space to the set $A \in \Theta$. Thus $K(\theta, A)$ maps potential transition events to their probability of occurrance.

When the state space is discrete then K is a matrix mapping, $k \times k$ for k discrete elements in A, where each cell defines the probability of a state transition from the first term to all possible states:

$$P_A \begin{bmatrix} p(\theta_1, \theta_1) & \dots & p(\theta_1, \theta_k) \\ : & & : \\ p(\theta_k, \theta_1) & \dots & p(\theta_k, \theta_k) \end{bmatrix}, \tag{2}$$

where the row indicates where the chain is at this period and the column indicates where the chain is going in the next period. The rows of P_A sum to one and define a conditional PMF since they are all specified for the same starting value and cover each possible destination in the state space: for row i: $\sum_{j=1}^{k} p(\theta_i, \theta_j)$. Each matrix element is a well-behaved probability, $p(\theta_i, \theta_j) \geq 0$, $\forall i, j \in A$. When the state space is continuous then K is a conditional PDF: $f(\theta|\theta_i)$, meaning a properaly defined probability statement for all $\theta \in A$, given some given current state θ_i.

An important feature of the transition kernel is that transition probabilities between two selected states for arbitrary numbers of steps m can be calculated multiplicatively. For instance the probability of transitioning from the state $\theta_i = x$ at time 0 to the state $\theta_j = y$ in exactly m steps is given by the multiplicative series:

$$p^m(\theta_i^{[0]} = x, |\theta_j^{[m]} = y) = \underbrace{\sum_{\theta_1} \sum_{\theta_2} \cdots \sum_{\theta_{m-1}}}_{\text{all possible paths}} \underbrace{p(\theta_i, \theta_1) p(\theta_1, \theta_2) \cdots p(\theta_{m-1}, \theta_j)}_{\text{transition products}}. \tag{3}$$

So $p^m(\theta_i^{[0]} = x, |\theta_j^{[m]} = y)$ is also a stochastic transition matrix, and this property holds for all discrete chains exactly as given, and for continuous Markov chains with only a slight modification involving integrals rather than summations. The basic idea of (3) is that the complete probability of transitioning from x to y is a product of all of the required intermediate steps where we sum over all possible paths that reach y from x.

1.1.1 A Simple Numerical Example of a Markov Chain

Start with a two dimensional state space, which can be thought of as discrete vote choice between two political parties, a commercial purchase decision between two brands, or some other choice. Suppose that voters/consumers that normally select θ_1 have an 80% chance of continuing to do so, and voters/consumers that normally select θ_2 have only a 40% chance of continuing to do so. Since there are only two choices, this leads the transition matrix P:

$$\begin{array}{cc} & \overbrace{\begin{array}{cc}\theta_1 & \theta_2\end{array}}^{\text{next period}} \\ \text{current period}\left\{\begin{array}{c}\theta_1\\ \theta_2\end{array}\right. & \left[\begin{array}{cc}0.8 & 0.2\\ 0.6 & 0.4\end{array}\right]\end{array}.$$

All Markov chains begin with a starting point assigned by the researcher. This two-dimensional initial state defines the proportion of individuals selecting θ_1 and θ_2 before beginning the chain. For the purposes of this example, assign the starting point:

$$S_0 = \left[\begin{array}{cc} 0.5 & 0.5 \end{array}\right].$$

That is, before running the Markov chain 50% of the observed population select each alternative. To get the to the first state we simply multiply then initial state by the transition matrix:

$$S_1 = \left[\begin{array}{cc} 0.5 & 0.5 \end{array}\right]\left[\begin{array}{cc}0.8 & 0.2\\ 0.6 & 0.4\end{array}\right] = \left[\begin{array}{cc} 0.7 & 0.3 \end{array}\right] = S_1.$$

So after the first iteration we have the new proportions: 70% select θ_1 and 30% select θ_2. This process continues multiplicatively as long as we like:

$$\begin{aligned}
\text{second state:}\quad S_2 &= \left[\begin{array}{cc} 0.7 & 0.3 \end{array}\right]\left[\begin{array}{cc}0.8 & 0.2\\ 0.6 & 0.4\end{array}\right] = \left[\begin{array}{cc} 0.74 & 0.26 \end{array}\right]\\
\text{third state:}\quad S_3 &= \left[\begin{array}{cc} 0.74 & 0.26 \end{array}\right]\left[\begin{array}{cc}0.8 & 0.2\\ 0.6 & 0.4\end{array}\right] = \left[\begin{array}{cc} 0.748 & 0.252 \end{array}\right]\\
\text{fourth state:}\quad S_4 &= \left[\begin{array}{cc} 0.748 & 0.252 \end{array}\right]\left[\begin{array}{cc}0.8 & 0.2\\ 0.6 & 0.4\end{array}\right] = \left[\begin{array}{cc} 0.7496 & 0.2504 \end{array}\right].
\end{aligned} \tag{4}$$

As one might guess, the choice proportions are converging to $[0.75, 0.25]$. This is because the transition matrix is pushing toward a steady state or more appropriately "stationary" distribution of the proportions. So when we reach this distribution all future states, S, are constant: $SP = S$.

Imagine that this stationary distribution was the articulation of some PMF or PDF that we could not analytically describe. If we could run this Markov chain sufficiently long we would eventually get the stationary distribution *for any point in the state space.* In fact, for this simple example we could solve directly for the steady state $S = [s_1, s_2]$ by stipulating:

$$\left[\begin{array}{cc} s_1 & s_1 \end{array}\right]\left[\begin{array}{cc}0.8 & 0.2\\ 0.6 & 0.4\end{array}\right] = \left[\begin{array}{cc} s_1 & s_2 \end{array}\right],$$

and solving the resulting two equations for the two unknowns. While this example is wildly over-simplified, it serves to show some basic characteristics of Markov chains. The operation of running a Markov chain until it reaches its stationary distribution is exactly the process employed in MCMC.

1.1.2 The Chapman-Kolmogorov Equations

The form of (3) also leads to a more general notion of how chain probabilities are strung together. The Chapman-Kolmogorov equations specify how successive events are bound together probabilistically. These are given here for both discrete and continuous state spaces where we abbreviate the left-hand side expression of (3):

$$\begin{aligned} p^{m_1+m_2}(x,y) &= \sum_{\text{all } z} p^{m_1}(x,z)p^{m_2}(z,y) \qquad \text{Discrete Case} \\ p^{m_1+m_2}(x,y) &= \int_{\text{range } z} p^{m_1}(x,z)p^{m_2}(z,y)dz \qquad \text{Continuous Case.} \end{aligned} \tag{5}$$

The Chapman-Kolmogorov equations are particularly elegant for the discrete case because (5) can be represented as a series of transition matrix multiplications:

$$p^{m_1+m_2} = p^{m_1}p^{m_2} = p^{m_1}p^{m_2-1}p = p^{m_1}p^{m_2-2}p^2 = \dots . \tag{6}$$

Thus iterative probabilities can be decomposed into segmented products in any way that we like, depending on the interim steps.

1.1.3 Marginal Distributions

The final basic notational characteristic of Markov chains that we will provide here is the *marginal* distribution at some step m from the transition kernel. For the discrete case the marginal distribution of the chain at the m step is obtained by inserting the current value of the chain, $\theta_i^{[m]}$, into the row of the transition kernel for the m^{th} step, p^m:

$$\pi^m(\theta) = [p^m(\theta_1), p^m(\theta_2), \dots, p^m(\theta_k)]. \tag{7}$$

So the marginal distribution at the first step of the Markov chain is given by:

$$\pi^1(\theta) = \pi^0(\theta)p^1 \tag{8}$$

where π^0 is the initial starting value assigned to the chain and $p^1 = p$ is the simple transition matrix given in (2). A really neat consequence of the defining characteristic of the transition matrix is the relationship between the marginal distribution at some (possibly distant) step and the starting value:

$$\pi^n = p\pi^{n-1} = p(p\pi^{n-2}) = p(p\pi^{n-3}) = \dots = p^n\pi^0.$$

Since it is clear here that successive products of probabilities quickly result in lower probability values, the property above shows how Markov chains eventually "forget" their starting points. The marginal distribution for the continuous case is only slightly more involved since we cannot just list as a vector the quantity:

$$\pi^m(\theta_j) = \int_\theta p(\theta,\theta_j)\pi^{m-1}(\theta)d\theta, \tag{9}$$

which is the marginal distribution of the chain given it is currently on point θ_j at step m.

2 General Properties of Markov Chains

There are several properties of Markov chains that are important to us, particularly when discussing convergence. These properties have intimidating names that are inherited from mathematical Markov chain theory, but in reality are fairly straightforward ideas. Generally, if we can describe the mathematical status of a particular chain, then we can often determine if it is producing useful sample from the distribution of interest. The properties are only summarized briefly here and those interested in a more technical and detailed treatment should read: Gamerman (1997) Chapter 4, Norris (1997), Nummelin (1984), Robert and Casella (1999) Chapter 4, or Tierney (1996).

2.1 Homogeneity

A Markov chain is said to be *homogeneous* at step m if the transition probabilities at this step do depend on the value of m. For example, at the starting point the chain cannot be homogeneous since the marginal distribution for the first step is clearly not independent of the initial values that are hand-picked. One reason that the Gibbs sampler and the Metropolis-Hastings algorithm, both given in detail in this chapter, dominate MCMC implementations is that the chains they define eventually obtain this property.

2.2 Recurrence

A Markov chain is called *recurrent* with regard to a given state, A, which is a single point or a defined collection of points (required for the continuous case), if the probability that the chain occupies A infinitely often over time is one. This can also be restated as saying that if the chain is currently in A, it will eventually return to A with probability one. The Markov chain is *positive recurrent* if the mean time to return to A is bounded, otherwise it is called *null recurrent*. Recurrence is a desirable property in Markov chains and there are also stricter forms such as *Harris recurrence* which stipulates the same condition for every possible starting value. See Robert and Casella (1999), Chapter 4. for details.

2.3 Irreducibility

There are also properties directly associated with states. A state is *absorbing* if once the chain enters this state it cannot leave: $p(A, A^c) = 0$. The obverse of absorbing is *transient*. A state is transient if given that a chain currently occupies state A, the probability of not returning to A is non-zero. A more general case of absorbing is the situation where a state, A, is *closed* with regard to some other state, B: $p(A, B) = 0$.

A Markov chain is *irreducible* if every point or collection of points (a subspace, required in the continuous case), A, can be reached from every other point or collection of points.[1] That is, $p(\theta_i, \theta_j) \neq 0$, $\forall i, j \in A$. Notice that irreducibility is a characteristic of both the chain and the subspace. So irreducibility implies the existence of a path between any two points in the subspace and therefore we expect there to be a relationship to the recurrence characteristic. This relationship is expressed as follows: *if a subspace is closed, finite, and irreducible, then all states within this subspace are recurrent.*

[1] A convenient way to remember the principle behind irreducibility is the notion that you could reduce the set if you wanted (this is obviously always possible except for the null set), but that *you do not want to* because then there will be points that cannot be reached from other points.

If we take a set of recurrent states (they must be non-empty, and bounded or countable), then their union creates a new state which is closed and irreducible (Meyn and Tweedie 1993). This means that the linkage between recurrence and irreducibility is important in defining a subspace that captures a Markov chain and at the same time assures that this Markov chain will explore all of the subspace. Whenever a chain wanders into a closed, irreducible set of recurrent states then stays there and visits every single state (eventually) with probability one.

2.4 Stationarity

Define $\pi(\theta)$ as the stationary distribution of the Markov chain for θ on the state space A. We denote $p(\theta_i, \theta_j)$ to indicate the probability that the chain will move from θ_i to θ_j at some arbitrary step t from the transition kernel, and $\pi^t(\theta)$ as the marginal distribution. This stationary distribution is then defined as satisfying:

$$\begin{aligned} \sum_{\theta_i} \pi^t(\theta_i) p(\theta_i, \theta_j) &= \pi^{t+1}(\theta_j) \qquad \text{Discrete Case} \\ \int \pi^t(\theta_i) p(\theta_i, \theta_j) d\theta_i &= \pi^{t+1}(\theta_j) \qquad \text{Continuous Case.} \end{aligned} \tag{10}$$

Therefore multiplication by the transition kernel and evaluating for the current point (the summation step for discrete sample spaces and the integration step for continuous sample spaces) produces the same marginal distribution: $\pi = \pi p$ in shorthand. This demonstrates that the marginal distribution remains fixed when the chain reaches the stationary distribution and we might as well drop the superscript designation for iteration number and just use $\pi(\theta)$.

Once the chain reaches its stationary distribution (also called its *invariant distribution*, *equilibrium distribution*, or *limiting distribution* if discussed in the asymptotic sense), it stays in this distribution and moves about, or "mixes", throughout the subspace according to marginal distribution, $\pi(\theta)$, forever. This is exactly what we want and expect from MCMC. If we can set up the Markov chain such that it reaches a stationary distribution that is the desired posterior distribution from our Bayesian model, then all we need to do is let it wander about this subspace for a while producing empirical samples to be summarized. The good news is that the two forms of MCMC kernels that we will use have the property that they are guaranteed to eventually reach a stationary distribution which is the desired posterior distribution.

2.5 Ergodicity

It is also possible to define the *period* of a Markov chain. This is simply the length of time to repeat an identical cycle of chain values. It is desirable to have an aperiodic chain, i.e. where the only length of time for which the chain repeats some cycle of values is the trivial case with cycle length equal to one. Why? It seems as though we would not necessarily care if there was some period to the chain values, particularly if the period was quite long, or perhaps in the discrete state if it included every value in the state space. The answer is that the recurrence property alone is not enough to assure that the chain reaches a state where the marginal distribution remains fixed and identical to the posterior of interest.

If a chain is irreducible, positive recurrent, and aperiodic, then we call it *ergodic*. Ergodic Markov chains have the property:

$$\underset{n\to\infty}{p^n} \lim(\theta_i, \theta_j) = \pi(\theta), \tag{11}$$

for all θ_i, and θ_j in the subspace (Nummelin 1984). Therefore, in the limit the marginal distribution at one step is identical to the marginal distribution at all other steps. Better yet, because of the recurrence requirement, this limiting distribution is now closed and irreducible meaning that the chain will never leave it and is guaranteed to visit every point in the subspace. Once a specified chain is determined to have reached its ergodic state, sample values behave as if they were produced by the posterior of interest from the model.

The *ergodic theorem* is the equivalent of the strong law of large numbers but for Markov chains. It states that any specified function of the posterior distribution can be estimated with samples from a Markov chain in its ergodic state because averages of sample values give strongly consistent parameter estimates. More formerly, suppose $\theta_{i+1}, \dots, \theta_{i+n}$ are n (not necessarily consecutive) values from a Markov chain that has reached its ergodic distribution, a statistic of interest, $h(\theta)$, can be calculated empirically:

$$\hat{h}(\theta_i) = \frac{1}{n} \sum_{i=i+1}^{i+n} h(\theta_i) \approx h(\theta), \tag{12}$$

and for finite quantities this converges almost surely: $p[\hat{h}(\theta_i) \to h(\theta), \text{ as } n \to \infty] = 1$ (Roberts and Smith 1994, p.210; Tierney 1994, p.1717).

The remarkable result from ergodicity is that even though Markov chain values, by their very definition, have serial dependence, the mean of the chain values provides a strongly consistent estimate of the true parameter. Furthermore, provided that the limiting variance of the empirical estimator $\hat{h}(\theta_i)$ is bounded, then subject to very general regularity conditions the central limit theorem also applies:

$$\sqrt{n}\frac{\hat{h}(\theta_i) - h(\theta)}{\sqrt{VAR(\hat{h}(\theta_i))}} \underset{n\to\infty}{\longrightarrow} \mathcal{N}(0,1). \tag{13}$$

There is also the notion of *geometric ergodicity*: the geometric rate of reduction in time for the total variation distance between some arbitrarily time point and convergence to the limiting distribution (see Mengerson and Tweedie (1996) for details). There are some additional nuances and complications and the seriously interested reader is referred to the references in this section for extended, and often highly technical, discussions.

3 The Gibbs Sampler

The Gibbs sampler, originating with Geman and Geman (1984), is by far the most widely used MCMC technique. This is a testament to its flexibility and reliability as method of producing useful chain values. The Gibbs sampler requires specific knowledge about the conditional nature of relationship between the variables of interest. The basic idea, which is not difficult to conceptualize, is that if it is possible to express each of the coefficients to be estimated as conditioned on all of the others, then by cycling through these conditional statements we can eventually reach the true joint distribution of interest.

3.1 Description

The Gibbs sampler is a transition kernel created by a series of full conditional distributions that is a Markovian updating scheme based on conditional probability statements. If the limiting distribution of interest is $\pi(\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is an k length vector of coefficients to estimate, then the objective is to produce a Markov chain that cycles through these conditional statements moving towards and then around this distribution. The set of full conditional distributions for $\boldsymbol{\theta}$ are denoted $\boldsymbol{\Theta}$ and defined by $\pi(\boldsymbol{\Theta}) = \pi(\theta_i)|\boldsymbol{\theta}_{-i}$ for $i = 1, \ldots, k$, where the notation $\boldsymbol{\theta}_{-i}$ indicates a specific parametric form from $\boldsymbol{\Theta}$ without an the θ_i coefficient.

It is essential that there be a definable conditional statement for each coefficient in the $\boldsymbol{\theta}$ vector and that these probability statements be completely articulated such that it is possible to draw samples from the described distribution. This requirement facilitates the iterative nature of the Gibbs sampling algorithm:

1. choose starting values: $\boldsymbol{\theta}^{[0]} = [\theta_1^{[0]}, \theta_2^{[0]}, \ldots, \theta_k^{[0]}]$
2. at the j^{th} starting at $j = 1$ complete the single cycle by drawing values from the k distributions given by:

$$\begin{aligned}
\theta_1^{[j]} &\sim \pi(\theta_1|\theta_2^{[j-1]}, \theta_3^{[j-1]}, \ldots, \theta_{k-1}^{[j-1]}, \theta_k^{[j-1]}) \\
\theta_2^{[j]} &\sim \pi(\theta_2|\theta_1^{[j]}, \theta_3^{[j-1]}, \ldots, \theta_{k-1}^{[j-1]}, \theta_k^{[j-1]}) \\
\theta_3^{[j]} &\sim \pi(\theta_3|\theta_1^{[j]}, \theta_2^{[j]}, \ldots, \theta_{k-1}^{[j-1]}, \theta_k^{[j-1]}) \\
&\vdots \\
&\vdots \\
\theta_{k-1}^{[j]} &\sim \pi(\theta_{k-1}|\theta_1^{[j]}, \theta_2^{[j]}, \theta_3^{[j]} \ldots, \theta_k^{[j-1]}) \\
\theta_k^{[j]} &\sim \pi(\theta_k|\theta_1^{(j)}, \theta_2^{[j]}, \theta_3^{[j]} \ldots, \theta_{k-1}^{[j]})
\end{aligned}$$

3. increment j and repeat until convergence.

Once convergence is reached all simulation values are from the target posterior distribution and a sufficient number should then drawn such that all areas of the posterior are explored. Notice the important feature that during the each iteration of the cycling through the $\boldsymbol{\theta}$ vector, conditioning occurs on $\boldsymbol{\theta}$ values that have already been sampled for that cycle, otherwise the $\boldsymbol{\theta}$ values are taken from the last cycle. So in the last step for a given j cycle, the sampled value for the k^{th} parameter gets to condition on *all* j-step values.

If the Gibbs sampler has run sufficiently long, forthcoming full cycles of the algorithm produce a complete sample of the coefficients in the $\boldsymbol{\theta}$ vector. That is, all future iterations produce samples from the desired limiting distribution and can therefore be described empirically. The most impressive aspect of the Gibbs sampler is that these conditional distributions contain sufficient information to eventually produce a sample from the full joint distribution of interest.

3.2 Example: Changepoint Analysis in a Poisson Model

This example is a simplified version of an analysis done by Carlin, Gelfand, and Smith (1992) that looks at a series of coal mine disasters over a 112 year history in Britain. The data are characterized by relatively high disaster counts in the early era and relative low disaster counts in the late era. Thus the question from a public policy perspective is when did improvements in technology and safety practices have an actual effect on the rate of serious accidents?

The data, covering the years 1851 to 1962, are given by:

4	5	4	1	0	4	3	4	0	6	3	3	4	0	2	6	3	3	5
4	5	3	1	4	4	1	5	5	3	4	2	5	2	2	3	4	2	1
3	2	2	1	1	1	1	3	0	0	1	0	1	1	0	0	3	1	0
3	2	2	0	1	1	1	0	1	0	1	0	0	0	2	1	0	0	0
1	1	0	2	3	3	1	1	2	1	1	1	1	2	4	2	0	0	0
1	4	0	0	0	1	0	0	0	0	0	1	0	0	1	0	1		

Statistically, the objective is to use this sequence to estimate the *changepoint* and also to obtain posterior estimates of the intensity parameters of the two separate Poisson processes (Poisson because these are counts). This process is addressed more generally for an unknown number of changepoints by Phillips and Smith (1996).

Specifically, $x_1, x_2, \dots, x_n$ are a series of count data where there exist the possibility of a changepoint at some period, k, along the series. Therefore there are two Poisson data generating processes:

$$\begin{aligned} x_i|\lambda &\sim \mathcal{P}(\lambda) & i &= 1, \dots, k \\ x_i|\phi &\sim \mathcal{P}(\phi) & i &= k+1, \dots, n \end{aligned}$$

where the determination of which to apply depends on the location of the changepoint k. So now there are three parameters to estimate: λ, ϕ, and k. This problem is distinguished by the added complexity that one of the parameters, k, operates in a different capacity on the others: determining a change in the serial data generation process, rather than as a conventional parametric input.

The three independent priors applied to this model are:

$$\begin{aligned} &\lambda \sim \mathcal{G}(\alpha, \beta) \\ &\phi \sim \mathcal{G}(\gamma, \delta) \\ &k \sim \text{discrete uniform on}[1, 2, \dots, n] \end{aligned}$$

where for purposes of this example the hyperparameters are assigned according to: $\alpha = 4$, $\beta = 1$, $\gamma = 1$, $\delta = 2$. Since the mean of a gamma distribution is the product of its parameters, this assignment of hyperparameters roughly resemble the mean of the first 50% of the data ($\alpha\beta$), and the second 50% of the data ($\gamma\delta$). This leads the joint posterior and its proportional simplification:

$$\begin{aligned} \pi(\lambda, \phi, k|\mathbf{y}) &\propto L(\lambda, \phi, k|\mathbf{y})\pi(\lambda|\alpha, \beta)\pi(\phi|\gamma\delta)\pi(k) \\ &= \left(\prod_{i=1}^{k} \frac{e^{-\lambda}\lambda^{y_i}}{y_i!}\right) \left(\prod_{i=1}^{k} \frac{e^{-\phi}\phi^{y_i}}{y_i!}\right) \left(\frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}\right) \left(\frac{\delta^\gamma}{\Gamma(\gamma)}\phi^{\gamma-1}e^{-\delta\phi}\right) \frac{1}{n} \\ &\propto \lambda^{\alpha-1+\sum_{i=1}^{k} y_i}\phi^{\gamma-1+\sum_{i=k+1}^{n} y_i}\exp[-k(\lambda+\phi) - \lambda\beta - \phi(\delta + n_{)}]. \end{aligned}$$

Without much agony over integration, this joint posterior produces the following conditional posterior distributions for the three quantities of interest:

$$\begin{aligned}
\lambda|\phi, k &\sim \mathcal{G}(\alpha + \sum_{i=1}^{k} y_i, \beta + k) \\
\phi|\lambda, k &\sim \mathcal{G}(\gamma + \sum_{i=k+1}^{n} y_i, \delta + n - k) \\
\pi(k|\lambda, \phi) &= \left(\lambda^{\alpha-1+\sum_{i=1}^{k} y_i} \phi^{\gamma-1+\sum_{i=k+1}^{n} y_i} \exp[-k(\lambda+\phi) - \lambda\beta - \phi(\delta + n_{)}\right) \\
&\quad \times \left(\sum j = 1n\lambda^{\alpha-1+\sum_{i=1}^{j} y_i} \phi^{\gamma-1+\sum_{i=j+1}^{n} y_i}\right. \\
&\qquad\qquad \left.\exp[-k(\lambda+\phi) - \lambda\beta - \phi(\delta + n_{)}]\right)^{-1}.
\end{aligned}$$

The posterior for k looks daunting until we notice that it can be reexpressed as the kernel of an exponential distribution and therefore becomes:

$$\pi(k|\lambda, \phi) \propto \exp[-(\lambda + \phi)k - \beta\lambda - \delta\phi - n\phi].$$

The constant terms are left in the exponent because they will help us in estimating the exponential centrality parameter and therefore save the trouble of calculating the normalizing factor. The discrete exponential PMF can be expressed as $f(x|\theta, a) = \theta e^{-(x-a)\theta}$ where the a term simply shifts the mean: $E[x|\theta, a] = \frac{1}{\theta} - a$. We can therefore use it to "tune" the constant value in order to obtain a rough estimate of the normalizing factor. This is done adjusting the only data-sourced value in the exponent (n).

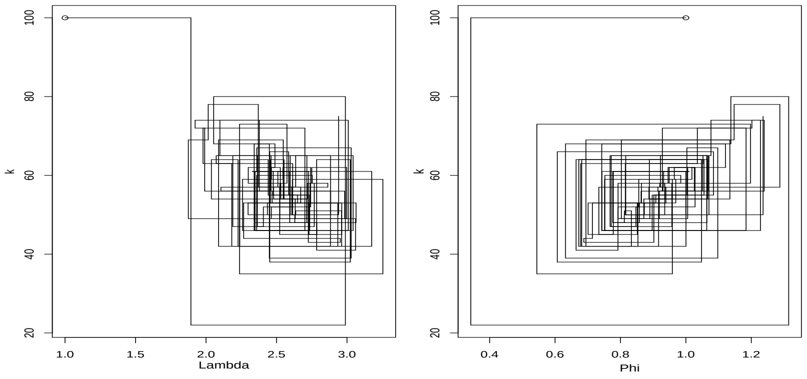
Table 1: MCMC Posterior Summary

Quantile	λ	ϕ	k
Minimum	1.000	0.5001	32.00
First Quartile	2.389	0.7987	49.00
Median	2.612	0.8768	54.00
Third Quartile	2.836	0.9774	60.00
Maximum	3.777	1.4080	100.00
Mean	2.622	0.8896	54.55

A run of 1000 iterations is performed and the posterior quantiles along with the mean are given in Table 1. All of the necessary `R` code for running this example are provided in this chapter's **Computational Addendum**.

Figure 1 shows the first 100 iterations of the sampler in a pair of two dimensional graphs where k is depicted on the y-axis in both cases plotted against λ and ϕ respectively. Notice that the Gibbs sampler converges rather quickly to the region of the reported posterior. Each movement is a straight line here since the intermedate steps of the Gibbs sampler hold all other parameter values constant while sampling for a single parameter conditioned on these other values. So looking at the first step in the left-hand side panel of Figure 1, we begin by conditioning on $k = 100$ and move from the starting value $\lambda = 1.0$ to a sampled value just under 2.0. Then the second half of the first iteration holds this value for λ constant and draws a value for k just over 20. After these two steps the first full iteration

Figure 1: Gibbs Sampling for Coal Mine Disasters



of the sampler is done. This process then continues for 999 more iterations.

3.3 Summary of Properties of the Gibbs Sampler

We finish this section with a summary of the properties of the Gibbs sampler that make it the most commonly used MCMC kernel specification.

- Since the Gibbs sampler on conditions on values from the last iteration of its chain values, it clearly constitutes a Markov chain.
- The Gibbs sampler has the true posterior distribution of parameter vector as its limiting distribution: $\boldsymbol{\theta}^{(i)} \xrightarrow[i=1\rightarrow\infty]{d} \boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$.
- The Gibbs sampler is a homogeneous Markov chain. That is, the consecutive probabilities are independent of n, the current length of the chain.
- The Gibbs sampler converges at a geometric rate: the total variation distance between an arbitrary time and the point of convergence decreases at a geometric rate in time (t).
- The Gibbs sampler is ergodic.

Casella and George (1992) provide a very clear basic introduction to the Gibbs sampler and its properties (the original title of the conference paper was “Gibbs for Kids”). A very useful discussion of the Gibbs sampler and its relation to other MCMC techniques is provided by Smith and Roberts (1993).

4 The Metropolis-Hastings Algorithm

The Gibbs sampler of Geman and Geman (1984) obviously does not work when the complete conditionals for the $\boldsymbol{\theta}$ parameters do not have an easily obtainable form. In these cases an update can be produced for these parameters using the Metropolis-Hastings algorithm from statistical physics (Chib and Greenberg 1995; Metropolis et al. 1953; Hastings 1970; Peskun 1973) which is often applied in fields completely unrelated to the original application (c.f. Cohen et al. 1998).

4.1 Background

The original work by Metropolis et al. postulated a two-dimensional enclosure with $n = 10$ molecular particles, and sought to estimate the state-dependent total energy of the system at equilibrium. The central problem that they confronted is that there is an incredibly large number of locations for the molecules in the system which must be accounted and this number grows exponentially with time. The key contribution of Metropolis et al. is to model the system by generating moves that are more likely than others based on positions that are calculated using uniform probability generated candidate jump points. The moves are accepted probabilistically and likely final states are determined after a set of periods where many such decisions are made. Therefore the simulation produces an estimated force based on a statistical, rather than deterministic, arrangement of particles. The critical assumptions are already familiar to us: any molecular state can be reached from another (ergodicity), and state changes are induced probabilistically with a instrumental distribution. The result, after convergence, is a distribution of particles from which energy calculations can be made.

Hastings (1970) as well as Peskun (1973) generalized the Metropolis, et al. algorithm by suggesting that other distributions such as the normal or Poisson could be used to provide potential jumping positions for each element of the chain. They both showed that the state space for the algorithm could include continuous forms, which is vast improvement in the statistical applicability. In addition, a limiting restriction of the original Metropolis algorithm, now known to be less restrictive, was that the candidate generating distribution used to suggest potential jumping points is had to be symmetrical in the original paper. That is, for two points in the state space, θ_1 and θ_2, $q(\theta_1|\theta_2) = q(\theta_2|\theta_1)$.

4.2 The Algorithm

The simplest Metropolis-Hastings algorithm for a single selected parameter works as follows. First transform θ_j, ($j \in J$ parameters in $\boldsymbol{\theta}$) such that it has the posterior distribution $\pi(\theta)$ with support over $\Re$.[2] At the t^{th} step in the chain, draw θ'_j from a distribution over the same support. One convenient possibility is a normal random variable centered at the current value of θ_j in the simulation and using the variance from past iterations: $s^2_{\theta_j}$ (specify $s^2_{\theta_j} = 1$ as a starting value). This distribution is called the instrumental, jumping, or proposal distribution and is denoted $q_t(\theta'_j|\theta_j)$. It must be possible to determine $q_t(\theta_j|\theta'_j)$, and under the original constraints of Metropolis, et al. the two conditionals need to be equal (symmetry), although we now know that this is not necessary.

The decision that produces the $t+1^{st}$ point in the chain is determined probabilistically according to:

$$\theta_j^{[t+1]} = \begin{cases} \theta'_j & \text{with probability} \quad P\left(\min\left(\frac{q_t(\theta_j|\theta'_j)}{q_t(\theta'_j|\theta_j)}\frac{\pi(\theta'_j)}{\pi(\theta_j^{[t]})}, 1\right)\right) \\ \theta_j^{[t]} & \text{with probability} \quad 1 - P\left(\min\left(\frac{q_t(\theta_j|\theta'_j)}{q_t(\theta'_j|\theta_j)}\frac{\pi(\theta'_j)}{\pi(\theta_j^{[t]})}, 1\right)\right) \end{cases} \tag{14}$$

So unlike the Gibbs sampler, the Metropolis-Hastings algorithm does not necessitate movement on every iteration. Notice that in the case of symmetry in the candidate generating density, $q_t(\theta_j|\theta'_j) = q_t(\theta'_j|\theta_j)$, the decision simplifies to a ratio of the posterior density values at the two points. An important characteristic of this algorithm is that on each iteration accepted values of from the candidate generating must have the characteristic that $\pi(\theta'_j)/q_t(\theta'_j|\theta_j) > \pi(\theta_j)/q_t(\theta_j|\theta'_j)$.

We can describe a symmetric (14) serially in the following steps:

[2] This is done purely as a computational convenience rather than for theoretical reasons.

1. Sample θ' from $q(\theta'|\theta)$.
2. Sample u from $u[0:1]$.
3. If $\pi(\theta')/\pi(\theta) > u$ then accept θ'.
4. Otherwise keep θ.

Obviously we want to choose the $q()$ distribution such that it is easy to sample from, but it is also important that $\pi(\theta')/q(\theta'|\theta)$ is fully known up to some arbitrary constant independent of θ. So three things can happen here: we can sample a value of higher density and move with probability one, we can sample a value of lower density but move anyway by drawing a small uniform random variable in Step 2. above, or we can draw a uniform random variable larger than the ratio of posteriors and therefore stay in the same location. One interesting feature of the Metropolis-Hastings algorithm, in contrast to EM, is that it is "okay" to move to lower density points, albeit probabilistically. That is, since this algorithm describes the full posterior density after convergence, it is necessary at times to move from a high density point to a low density point. The EM algorithm never makes this kind of decision and therefore is only a mode finder rather than a method of fully sampling from the target distribution.

It can be shown both that the Gibbs sampler is a generalization of Metropolis-Hastings where the probability of accepting the candidate value is always 1 (Tanner 1996, p.182), and that Metropolis-Hastings is a generalization of the Gibbs sampler where movement is not necessary, the full conditionals are not required, and the previous value of the component to be (potentially) updated is consulted (Gamerman 1997, p. 166, Besag et al. 1995).

5 Metropolis-Hastings Properties

5.1 The Chain

Consider a posterior of interest, π_θ with $\int \pi_\theta = 1$, for θ on some state space, S, which is defined on a d-dimensional Lebesgue measure: $S \subseteq \Re$. The motivation for seeking an iterative solution through MCMC techniques is that this π_θ distribution is analytically complicated or unwieldy (Gelman 1992), so we want a procedure that eventually arrives at this distribution through simulation. The Metropolis-Hastings algorithm provides this function with a two-part transition kernel that has the property that it is closed with respect to the limiting distribution of π_θ:

$$\pi(\theta)p(\theta,\theta') = \pi(\theta')p(\theta',\theta) \qquad \forall \theta, \theta' \in S, \tag{15}$$

where $p(a,b)$ defines a transition kernel from state a to state b. This is called the *reversibility* condition for $p()$. The values for π are simply the posterior distribution evaluated at the two points: θ and θ', and $p(\theta,\theta')$ is the appropriately sized transition kernel from θ to θ'. Robert and Casella (1999, p.235) provide a proof that under very general conditions, virtually any conditional distribution over the appropriate support will provide a candidate jumping distribution that provides a Metropolis-Hastings chain that will eventually converge to the limiting distribution of $\pi(\theta)$ provided that (15) holds. This is the key theoretical importance of the algorithm because it shows that the right thing *will* happen if we let the chain run long enough.

A less general, and therefore less useful condition, is *symmetry*: $p(\theta,\theta') = p(\theta',\theta)$. This was originally the specified requirement related to the distributional relationship between the two points, but the contribution of Hastings (1970) was to demonstrate that this is not strictly necessary. Obviously with this scheme, once the chain reaches its stationary distribution, all future values generated by the chain are representations of $\pi(\theta)$.

5.2 A Brief Derivation

There are actually two parts to the transition kernel, a jumping density $q(\theta, \theta')$, and a jumping probability $\alpha(\theta', \theta)$:

$$p(\theta, \theta') = q(\theta, \theta')\alpha(\theta, \theta') \tag{16}$$

which determine the distribution of new θ' values to move to and the probability of making such a move, respectively. The distribution used for $q(\theta, \theta')$ is arbitrary, but a specification is required and typical forms are the normal and t distributions. Thus we sample from $q(\theta, \theta')$ to get potential values of the chain to jump to. The form of this jumping distribution plays only a part in determining values to jump to and does not affect the *decision* to jump. By symmetric logic we can also define the reverse jump:

$$p(\theta', \theta) = q(\theta', \theta)\alpha(\theta', \theta). \tag{17}$$

The decision to jump or not represents a second level of randomization as determined by the probability $\alpha(\theta, \theta')$. The candidate jumping point is favored if its posterior conditional probability is large relative to the posterior conditional probability associated with remaining at the current point, where the condition is respect to the current values in the other dimensions at that stage of the chain.

A decision rule can be derived by starting with (16) solved for $\alpha(\theta, \theta')$

$$\alpha(\theta, \theta') = \frac{p(\theta', \theta)}{q(\theta', \theta)},$$

and then inserting (15) solved for $p(\theta, \theta')$

$$\alpha(\theta, \theta') = \frac{\pi(\theta')p(\theta', \theta)}{\pi(\theta)q(\theta, \theta')},$$

and (17) solved for $p(\theta, \theta')$ to produce

$$\alpha(\theta, \theta') = \frac{\pi(\theta')q(\theta', \theta)\alpha(\theta', \theta)}{\pi(\theta)q(\theta, \theta')}.$$

This can be arranged as:

$$\frac{\alpha(\theta, \theta')}{\alpha(\theta', \theta)} = \frac{\pi(\theta')q(\theta', \theta)}{\pi(\theta)q(\theta, \theta')}. \tag{18}$$

An acceptance rule (Hastings 1970) that meets this criteria and accommodates the situation where we require for a jump $\alpha(\theta', \theta) > \alpha(\theta, \theta')$ is:

$$D(\theta', \theta) = \min\left[\frac{\pi(\theta')q(\theta', \theta)}{\pi(\theta)q(\theta, \theta')}, 1\right]. \tag{19}$$

This is exactly (14), restated where $q(\theta', \theta) = q_t(\theta_j|\theta'_j)$ and $q(\theta, \theta') = q_t(\theta'_j|\theta_j)$. If we were willing to substitute symmetry for reversibility, we would get the original simplified rule from Metropolis et al., which is similar, but much more intuitive:

$$D(\theta', \theta) = \min\left[\frac{\pi(\theta')}{\pi(\theta)}, 1\right], \tag{20}$$

because of cancellation. This states that the chain will move with probability one in a direction of higher posterior probability if offered by the jumping distribution, and will move with probability $r = \pi(\theta')/\pi(\theta)$ to the new point otherwise. Therefore low values of r will often result in staying at the same chain value for that dimension. Due to the two levels of randomization, three things can now happen during each chain iteration: move to the new point with probability one, generate a uniform random variable (bounded by zero and one) that is less than r thus moving to the new point, or generate a uniform random variable greater than r and then stay at the current point.

5.3 The Transition Kernel

With only a little more trouble we can rigorously define the properties of the Metropolis-Hastings transition kernel $p(\delta', \delta)$. Start by defining an indicator function for the event that the δ' point is accepted: $I(\delta') = 1$ if δ' accepted, 0 otherwise. Thus the probability of transitioning from $\theta = \theta^{[k]}$ to the proposed jumping value $\theta' = \theta^{[k+1]}$ at the k^{th} step is:

$$\begin{aligned} p(\theta', \theta) &= p(\theta^{[k+1]} = \theta', I(\theta') | \theta^{[k]} = \theta) \\ &= p(\theta^{[k+1]} = \theta' | \theta^{[k]} = \theta) p(I(\theta')) \\ &= q(\theta', \theta) \min\left[1, \frac{\pi(\theta')}{\pi(\theta)}\right] \qquad \theta \neq \theta'. \end{aligned} \tag{21}$$

The probability calculation for transitioning from $\theta = \theta^{[k]}$ to the current value (that is, not moving at all) is only slightly more complicated because it can occur two ways: a successful transition to the current state and a failed transition to a different state. The first event has probability zero in continuous state space, but is worth covering for to discrete applications. This probability is:

$$\begin{aligned} p(\theta, \theta) &= \underbrace{p(\theta^{[k+1]} = \theta, I(\theta) | \theta^{[k]} = \theta)}_{\text{moving back to same point}} + \underbrace{p(\theta^{[k+1]} \neq \theta, \neg I(\theta) | \theta^{[k]} = \theta)}_{\text{not moving}} \\ &= q(\theta, \theta) + \sum_{\theta' \neq \theta} q(\theta', \theta)(1 - \min\left[1, \frac{\pi(\theta')}{\pi(\theta)}\right]. \end{aligned} \tag{22}$$

In both of these calculations the more simple situation of symmetry is assumed, but moving to the reversibility assumption is just a matter of substituting $(q(\theta', \theta)\pi(\theta'))/(q(\theta, \theta')\pi(\theta))$ for $\pi(\theta')/\pi(\theta)$.

A critical component of the choice for the jumping distribution is the specified variance. If this variance is too large then the jumping distribution will be too wide relative to the target distribution and each successive step will move too far in some direction causing us to move awkwardly through the sample space in exagerated steps. It is also possible to stipulate a jumping distribution variance that is too small causing overly cautious small steps through the sample space. In this case we will converge slowly and mix poorly through the limiting distribution once we have converged.

6 Data Augmentation

Tanner and Wong (1987) introduced data augmentation (sometimes called substitution sampling) as a method for dealing with missing data or unknown parameter values by augmenting known information with candidate values much in the same way that EM does and iteravely improving the quality of these augmented quantities. In fact, data augmentation can be used instead of EM to estimate models with missing data when more than the mode of the likelihood function is required. Data augmentation is an MCMC technique because successively substitutes improved estimates conditioned on the previous state and therefore forms a Markov chain.

Much like the situation in which we applied the EM algorithm, suppose that we are interested in estimating single dimension (for now) parameter θ. We observe some relevant data but lack the complete set: $\mathbf{X} = [\mathbf{X}_{obs}, \mathbf{X}_{mis}]$, where all of the data (observed or not) is conditional on $\boldsymbol{\theta}$. Data augmentation requires that we know the parametric form of the posterior $p(\theta|\mathbf{X})$ corresponding to the complete data specification, and the predictive form for the missing data according to $p(\mathbf{X}_{mis}|\mathbf{X}_{obs})$. The algorithm proceeds by augmenting the observed data with simulated values of the missing data, obtained by cycling through these conditions according to the algorithm now described.

Start by defining the *posterior identity*, which is the desired quantity stated as if we could integrate out the missing data:

$$p(\theta|\mathbf{X}_{obs}) = \int_{\mathbf{X}_{mis}} p(\theta|\mathbf{X}_{obs}, \mathbf{X}_{mis}) p(\mathbf{X}_{mis}|\mathbf{X}_{obs}) d\mathbf{X}_{mis}. \tag{23}$$

We can also define the *predictive identity* by asserting that there is some unknown parameter, ϕ on the sample space of θ, critical to generating the unobserved data but integrated out:

$$p(\mathbf{X}_{mis}|\mathbf{X}_{obs}) = \int_{\Phi} p(\mathbf{X}_{mis}|\phi, \mathbf{X}_{obs}) p(\phi|\mathbf{X}_{obs}) d\phi. \tag{24}$$

Now insert (24) into (23) for the last term and interchange the order of integration:

$$\begin{aligned} p(\theta|\mathbf{X}_{obs}) &= \int_{\mathbf{X}_{mis}} p(\theta|\mathbf{X}_{obs}, \mathbf{X}_{mis}) \left[\int_{\Theta} p(\mathbf{X}_{mis}|\phi, \mathbf{X}_{obs}) p(\phi|\mathbf{X}_{obs}) d\phi \right] d\mathbf{X}_{mis} \\ &= \int_{\Theta} \underbrace{\int_{\mathbf{X}_{mis}} p(\theta|\mathbf{X}_{obs}, \mathbf{X}_{mis}) p(\mathbf{X}_{mis}|\phi, \mathbf{X}_{obs}) d\mathbf{X}_{mis}}_{K(\theta,\phi)} p(\phi|\mathbf{X}_{obs}) d\phi. \end{aligned} \tag{25}$$

Here Tanner and Wong use the shorthand $K(\theta, \phi)$ to make the notation cleaner, not to imply that this is a joint probability (it is really a *transition kernel*!).

The form of (25) implies that an iterative algorithm could be constructed that generates values of θ given an approximation for $\mathbf{X}_{mis}$, and then generates new values of $\mathbf{X}_{mis}$ given this θ. Specifically, data augmentation at the i^{th} iteration is (beginning with candidate values for $\mathbf{X}_{mis}$):

- **[Imputation-Step:]**
  - generate $\theta^{[i]}$ from $p^{[i-1]}(\theta|\mathbf{X}_{obs})$,
  - generate m values of of $\mathbf{X}_{mis}$ from $p(\mathbf{X}_{mis}|\theta^{[i]}, \mathbf{X}_{obs})$.
- **[Posterior-Step:]**

– update the parametric approximation using $\mathbf{X}_{mis,1}, \dots, \mathbf{X}_{mis,m}$:

$$p^{[i]}(\theta|\mathbf{X}_{obs}) = \frac{1}{m}\sum_{j=1}^{m} p(\theta|\mathbf{X}_{obs}, \mathbf{X}_{mis,i}). \tag{26}$$

In very similar fashion to importance sampling, there are two interrelated researcher-generated specifications here: the tolerance value for determining convergence $(p^{[i]}(\theta|\mathbf{X}_{obs}) - p^{[i+1]}(\theta|\mathbf{X}_{obs}))$, and the number of simulation values at each step (m). The second decision is perhaps more crucial, and is a (now) familiar balance between speed and accuracy. In their original article, Tanner and Wong provide m values 1,600 and 6,400 for relatively simple model specifications. It is therefore recommended that a similar value can be used as a starting point. With computers becoming faster every day, it is unlikely that this will place a heavy computation burden on the average system. Since larger values of m give better intermediate approximations, there is necessarily a tradeoff between longer runs and longer calculations at each run. In fact, if $m = 1$, then data augmentation is actually Gibbs sampling, and obviously the emphasis is then purely on the length of runs.

Convergence of the data augmentation algorithm is demonstrated in Tanner and Wong. Rosenthal (1993) showed that this convergence is on the order of the log of the number of missing cases in the data. This is particularly encouraging because it means that even for the very largest data sets that we see in the social and behavioral sciences, data augmentation will likely be a reasonable computation process. Also, including latent data from the data augmentation procedure also does not preclude model comparison or the generation of the standard model tests. For instance, Raftery (1996, p.182) gives the details of Bayes Factor tests when one or both of the model include such latent data.

There are enough similarities between data augmentation and EM that one might wonder when one is more applicable than the other. Both algorithms rely upon using the likelihood function under the most simple circumstances possible, and making these circumstances simple by completing the data with successively improved estimates of the missingess. EM makes more sense when the objective is simply to get the mode of the posterior because it is faster (less within-step calculations to perform). However, if the goal is to describe the complete posterior distribution, then data augmentation is more appropriate. Since data augmentation is a special case of Gibbs sampling and unlike Gibbs sampling it is not directly implemented in `BUGS` (see Congdon (2001, p.114) for a convenient work-around, however), then one might have a natural preference for Gibbs sampling in practice unless one wanted to treat the missingness more explicitly.

7 Practical Considerations and Admonitions

This section covers practical issues with regard to implementing a MCMC. There are several critical design questions that must be answered before actually setting up the chain and running it. These include decisions about: determination of the "burn-in" period for the chain, whether to "thin" the chain values, determination of where to start the chain(s), and various software implementation problems.

7.1 Thinning the Chain

It is sometimes the case that with very long simulation runs the storage of the values on the computer becomes an issue. Large storage files can result from: high autocorrelation of the iterations, slow convergence, many specified simultaneous chains, and high dimensionality of the problem. The idea of thinning the chain is to run the chain normally but record only every k^{th} value of the chain, thus reducing the storage demands while still preserving the integrity of the Markov process. It is important to note that thinning does not in any way improve the quality of the estimate (suggested presumably but erroneously as a way to increase the independence of the final evaluated values (Geyer 1992)), speed up the chain, or help in convergence. Instead, it is purely a device for dealing with possibly limited computer resources. Of course, given the current pace of improvement on computer hardware pricing and capacity, this becomes less of an issue over time.

The obvious question that remains is what value should one pick for k. Clearly there are tradeoffs here between storage and accuracy as well diagnostic ability. The greater the amount of thinning, the more potentially important information is lost. Conversely, prior to assumed convergence thinning is irrelevant. This decision is intertwined with other decisions such as the length of the burn-in period.

7.2 The Burn-In Period

First, one must decide the length of the burn-in period, the beginning set of runs that are discarded under the assumption that they represent pre-convergence values and are therefore not representative of the desired limiting distribution. The slower the chain is to converge, the more careful one should be about the burn-in period. Unfortunately, even starting the chain right in the area of the mode of the highest density area does not guarantee that burn-in period is unimportant as it will still take the Markov chain some time to forget its starting position.

There is no systematic, universal, guaranteed way to calculate the length of the burn-in period and considerable work on convergence diagnostics has been done to make specific recommendations and identify helpful tests. Raftery and Lewis (1992, 1996) suggest a running diagnostic for M, the length of the burn-in period, that starts with the analysis of an initial run. The idea is to solve for the number of iterations required to estimate some quantile of interest within an acceptable range of accuracy, at a specified probability level. The procedure is based on conventional normal distribution theory and implemented in their `gibbsit` software.

7.3 Starting Points or Startling Points

Starting points are an under-studied issue, except perhaps in the case of one particular convergence diagnostic (Raftery and Lewis 1992). Generally it is best to try several starting points in the state space and observe whether they lead to noticeably different descriptions of the posteriors. This is surely a sign of non-convergence of the Markov chain. Unfortunately the inverse is not true: it is not the case that if one starts several Markov chains in different places in the state space and they congregate for a time in the same region that this is the region that describes the stationary distribution. It could be that all of the chains are seduced by the same local maxima and will for a time mix around in its local region.

Overdispersing the starting points relative to the expected modal point is most likely to provide a useful assessment (Gelman and Rubin 1992a, 1992b). We will now look at strategies for determining such overdispersed points. If one can determine the mode with reasonable certainty, either through the EM algorithm, a grid search, or some other technique (perhaps analytical), then it is relatively simple to spread starting points around it at some distance. If this is not possible, or perhaps excessively complicated in high dimensions, then perhaps spreading the starting points widely throughout the sample space in general. Obviously this leads to wildly incorrect starting points on occasion and therefore the chain will require longer runs.

Sometimes starting points of *theoretical interest* are appropriate. It might be the case that a starting point can be assigned to values associated with other studies, subject-matter expertise, or previous work with the same data. It is still wise, however, to also incorporate overdispersed starting points as well.

8 Historical Comments

The background of the development of modern MCMC methods is interesting unto itself. The first notable event was the publication of a 1953 paper by Nicholas Metropolis, et al., where the "et al." is Arianna Rosenbluth, Marshall Rosenbluth, Augusta Teller, and another Teller whose first name is Edward and who apparently made contributions to nuclear physics related to rather large explosions. Because the paper was published in the *Journal of Chemical Physics* and because it was applied exclusively to the problem of particles moving around a square, interest was restricted primarily to physics.

Metropolis, et al. were interested in obtaining the positions and therefore the potential between all molecules in an enclosure and noted that even very modest sized setups lead to integrals of very high dimensions. Specifically, if $\hbar_{ij}$ represents the shortest distance between particles i and j, and $V(\hbar_{ij})$ is the associated potential. Then the total potential energy of the whole system is given by: $E = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} V(\hbar_{ij})$. Due to some simplifications this leads to an integral of "only" 200 dimensions for the force of the system. So rather than try to track and calculate future positions, they arrived at the idea of setting a molecule movement criterion in a formal model sense and then simulating a series of potential positions. In other words it was sufficient to know where the molecules were probabilistically at some future point in time as opposed to exactly.

The Metropolis, et al. paper was slow to permeate other disciplines including statistics partly because the authors appear not to have been aware of the widespread applicability of their technique. This is a presumed explanation for why the authors chose not to generalize it beyond the single application given. The key to the dissemination of the algorithm was the refinement and generalization done by Hastings (1970), some time later. He showed that reversibility can be substituted for symmetry in the approximation distribution, applicability to continuous state spaces, and he makes the ideas accessible to statisticians. In addition, Peskun (1973) should be credited with further introducing the Metropolis algorithm to the statistics community and proving a number of important properties including principles of reversibility.

The Geman and Geman (1986) paper introduces a new use for the Gibbs distribution in simulation and applies the tool to restoration of degraded images. This paper is very difficult to work through and most people do not persevere. Instead, the landmark Gibbs sampling paper as far as widespread effects are concerned is that of Gelfand and Smith (1990). They demonstrate how universally useful Gibbs sampling is in terms of setting up Markov chains to estimate posterior distributions. While nothing is entirely new in this paper, the synthesis and integration of Gibbs sampling into Markov chain Monte Carlo theory for the first time is an invaluable contribution.

One key reason for the explosion of academic attention to MCMC that occurred in the 1990s is the substantial improvement in computing power on the average desktop. This cause cannot be overstated. By definition these techniques are computer-intensive and it is hard to imagine earlier researchers being pleased with either the speed of their microcomputer or the convenience of their campus mainframe. Compared to a PC purchased only three years ago, a typical machine calculates three times faster, contains ten times as much storage, accesses the Internet a thousand times faster, and contains four times as much material in active use (Gill and Conklin 2002).

8.1 Full Circle?

An inquiring mind may have realized that many of the properties used to analyze and exploit MCMC techniques are *frequentist* in nature. That is, the central limit theorem, the law of large numbers, general asymptotic analysis, and transition invariance, are all basic principles from traditional non-Bayesian statistics. Specifically, the tool that revolutionized Bayesian statistics is in fact a frequentist construction. Efron (1998) notes that Fisher's work directly implied several modern statistical computing techniques that Fisher couldn not have employed for purely mechanistic reasons. These include bootstrapping (the bootrap plug-in principle is anticipated by the calculation of Fisher information), empirical calculation of confidence intervals, empirical Bayes (developing a prior using the data), and Bayes Factors.

9 References

Besag, J., P. J. Green, D. Higdon, and K. Mengerson. 1995. "Bayesian Computation and Stochastic Systems (with discussion)." *Statistical Science* 10, 3-66.

Carlin, Bradley, Alan E. Gelfand, and A. F. M. Smith. 1992. "Hierarchical Bayesian Analysis of Changepoint Problems." Bradley P. Carlin, Alan E. Gelfand, Adrian F. M. Smith Applied Statistics, Vol. 41, No. 2. (1992), pp. 389-405.

Casella, George, and Edward George. 1992. "Explaining the Gibbs Sampler." *The American Statistician* 46, 167-74.

Chib, Siddhartha, and Edward Greenberg. 1995. "Understanding the Metropolis-Hastings Algorithm." *The American Statistician.* 49, 327-35.

Cohen, Jacqueline, Daniel Nagin, Garrick Wallstrom, and Larry Wasserman. 1998. "Hierarchical Bayesian Analysis of Arrest Rates." *Journal of the American Statistical Association* 93, 1260-70.

Congdon, Peter. 2001. *Bayesian Statistical Modeling.* New York: Wiley & Sons.

Doob, J. L. 1990. *Stochastic Processes.* New York: Wiley & Sons.

Efron, Bradley. 1998. "R. A. Fisher in the 21st Century." *Statistical Science* 13, 95-122.

Gamerman, Dani. 1997. *Markov Chain Monte Carlo.* New York: Chapman & Hall.

Gelfand, A. E., and A. F. M. Smith. 1990. "Sampling-based Approaches to Calculating Marginal Densities." *Journal of the American Statistical Association* 85: 389-409.

Gelman, Andrew. 1992. "Iterative and Non-iterative Simulation Algorithms." In *Proceedings of the 24th Symposium on the Interface*, H. Joseph Newton (ed.). College Station, TX: Interface Foundation of North America.

Geman, S., and D. Geman. 1984. "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6,721-41.

Geyer, C. J. 1992. "Practical Markov Chain Monte Carlo." *Statistical Science* 7, 473-511.

Gill, Jeff, and David Conklin. 2002. "Congressional Support for K-12 Internet Access: the E-Rate Case." In *Congress and the Internet.* James A. Thurber (ed). New York: Houghton Mifflin.

Hastings, W. K. 1970. "Monte Carlo Sampling Methods Using Markov Chains and Their Applications." *Biometrika* 57, 97-109.

Karlin, Samual, and Howard M. Taylor. 1990. *A First Course in Stochastic Processes.* New York: Academic Press.

Karlin, Samuel, and Howard M. Taylor. 1981. *A Second Course in Stochastic Processes.* New York: Academic Press.

Koppel, Jonathan G. S. 1999. "The Challenge of Administration by Regulation: Preliminary Findings Regarding the U.S. Government's Venture Capital Funds." *Journal of Public Administration Research and Theory* 9, 641-66.

Hoel, Paul G., Sidney C. Port, and Charles J. Stone. 1987. *An Introduction to Stochastic Processes.* Prospect Heights, IL: Waveland Press.

Mengersen, K. L., and R. L. Tweedie. 1996. "Rates of Convergence of the Hastings and Metropolis Algorithms." *The Annals of Statistics* 24, 101-21.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. 1953. "Equation of State Calculations by Fast Computing Machine." *Journal of Chemical Physics* 21, 1087-91.

Meyn, Sean P., and Richard L. Tweedie. 1996. *Markov Chains and Stochastic Stability.* New York: Springer-Verlag.

Norris, J. R. 1997. *Markov Chains.* Cambridge: Cambridge University Press.

Nummelin, E. 1984. *General Irreducible Markov Chains and Non-negative Operators.* Cambridge: Cambridge University Press.

Peskun, P. H. 1973. “Optimum Monte Carlo Sampling Using Markov Chains.” *Biometrika* 60, 607-12.

Phillips, David B., and Adrian F. M. Smith. 1996. “Bayesian Model Comparison Via Jump Diffusions.” In *Markov Chain Monte Carlo in Practice.* W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (eds.). New York: Chapman and Hall.

Raftery, Adrian E., and Steven M. Lewis. 1996. “Implementing MCMC.” In *Markov Chain Monte Carlo in Practice.* W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (eds.). New York: Chapman and Hall.

Raftery, Adrian E., and Steven M. Lewis. 1992. “How Many Iterations in the Gibbs Sampler?” In *Bayesian Statistics 4.* J. O. Berger, J. M. Bernardo, A. P. Dawid, and A. F. M. Smith (eds.). Oxford: Oxford University Press.

Robert, Christian P., and George Casella. 1999. *Monte Carlo Statistical Methods.* New York: Springer-Verlag.

Rosenthal, Jeffrey. 1993. “Rates of Convergence for Data Augmentation on Finite Sample Spaces.” *Annals of Statistics* 3, 819-839.

Ross, Sheldon. 1996. *Stochastic Processes.* New York: Wiley & Sons.

Smith, A. F. M., and G. O. Roberts. 1993. “Bayesian Computation via the Gibbs Sampler and Related Markov Chain Monte Carlo Methods.” *Journal of the Royal Statistical Society, Series B* 55, 3-24.

Tanner, Martin A. 1996. *Tools for Statistic Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions.* New York: Springer.

Tanner, Martin A., and Wing Hung Wong. 1987. “The Calculation of Posterior Distributions by Data Augmentation.” *Journal of the American Statistical Society* 82, 528-50.

Tierney, Luke. 1996. “Introduction to General State-Space Markov Chain Theory.” In *Markov Chain Monte Carlo in Practice.* W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (eds.). New York: Chapman and Hall.

Tierney, Luke. 1994. “Markov Chains for Exploring Posterior Distributions.” *Annals of Statistics* 22, 1701-1728.

10 Computational Addendum: R and BUGS for MCMC

The following section of R code implements the example from Carlin, Gelfand, and Smith (1992), where they develop a Bayesian changepoint model. The point is to show the workings of the Gibbs sampler, not to provide software for research purposes. The BUGS package is recommended in general for implementing MCMC estimation models.

```
coal.mining.disasters <- c(4,5,4,0,1,4,3,4,0,6,3,3,4,0,2,6,
                           3,3,5,4,5,3,1,4,4,1,5,5,3,4,2,5,
                           2,2,3,4,2,1,3,2,2,1,1,1,1,3,0,0,
                           1,0,1,1,0,0,3,1,0,3,2,2,0,1,1,1,
                           0,1,0,1,0,0,0,2,1,0,0,0,1,1,0,2,
                           3,3,1,1,2,1,1,1,1,2,4,2,0,0,1,4,
                           0,0,0,1,0,0,0,0,0,1,0,0,1,0,1)

gibbs.poisson.gamma <- function(theta.matrix,y,reps)  {
    alpha <- 4; beta <- 1; gamma <- 1; delta <- 2
    for (i in 2:(reps+1))  {
        lambda <- rgamma(1,alpha
                  +sum(y[1:theta.matrix[(i-1),3]]),
                  1/(beta+theta.matrix[(i-1),3]))
        phi    <- rgamma(1,gamma
                  +sum(y[theta.matrix[(i-1),3]:length(y)]),
                  1/(delta+length(y)-theta.matrix[(i-1),3]))
        m      <- round(rexp(1,lambda+phi)
                  +beta*lambda+phi*delta+length(y)*phi/alpha)
        theta.matrix <- rbind(theta.matrix,c(lambda,phi,m))
    }
    theta.matrix
}

start <- matrix(c(1,1,20),1,3)
gibbs.poisson.gamma(start,coal.mining.disasters,1000)
```

Figure 1 was produced using the following function for graphing the path of a Gibbs sampler above in two chosen dimensions. You must give it a value for `sim.rm` which removes the third variable of choice. For larger dimensional joint posteriors, obvious modifications to this function are easy to perform.

```
plot.walk.multi <- function(walk.mat,sim.rm)  {
    walk.mat <- walk.mat[,-sim.rm]
    points(walk.mat[1,1],walk.mat[1,2])
     for(i in 1:(nrow(walk.mat)-1))  {
        segments(walk.mat[i,1],walk.mat[i,2],
                 walk.mat[(i+1),1],walk.mat[i,2])
        segments(walk.mat[(i+1),1],walk.mat[i,2],
                 walk.mat[(i+1),1],walk.mat[(i+1),2])

    }
}
```