Chapter 3

Basic Algorithms

This chapter describes the two basic “algorithms” for Markov chain Monte Carlo. The word “algorithms” is in quotation marks because what will actually be described are elementary update steps, bits of algorithm that change the state variable of the Markov chain in such a way so as to preserve a specified invariant distribution. These updates can be combined as described in Section 1.7.1 to make more complicated Markov transition mechanisms preserving the same invariant distribution. Repeating an update mechanism, basic or combined, again and again simulates a Markov chain. The two types of basic update step are the Gibbs update described in Section 1.7.2, the basic component of the “Gibbs sampler,” and the Metropolis-Hastings-Green update, the basic component of the so-called “Metropolis-Hastings-Green algorithm.”

3.1 Combining Update Mechanisms

3.1.1 Simple Composition and Mixing

We already met “composition” and “mixing” of elementary update mechanisms in Section 1.7.1 (commonly called “fixed scan” and “random scan” in the MCMC literature). Then in Chapter 2 we learned that composition corresponded to operator multiplication and mixing to a convex combination of operators.

The composition of update mechanisms that correspond to Markov transition kernels \( P_1, \ldots, P_d \) is the kernel \( P_1 \cdots P_d \). The proof that if \( P_1, \ldots, P_d \) each preserves a distribution \( \pi \), then so does the composition \( P_1 \cdots P_d \) is trivial, just the fact that kernel multiplication is associative (2.18), so

\[
\pi P_1 P_2 \cdots P_d = \pi P_2 \cdots P_d = \cdots = \pi P_d = \pi.
\]

The mixture of update mechanisms that correspond to Markov transition kernels \( P_1, \ldots, P_d \) and uses the mixing distribution with probabilities \( a_1, \ldots, a_d \) is

\[
a_d = \sum_i a_i P_i.
\]

The proof that if \( P_1, \ldots, P_d \) each preserves a distribution \( \pi \), then
so does the composition $\sum_{i=1}^{d} a_i P_i$ is just as trivial
\[ \pi \left( \sum_{i=1}^{d} a_i P_i \right) = \sum_{i=1}^{d} a_i \pi P_i = \left( \sum_{i=1}^{d} a_i \right) \pi = \pi \]

No good theoretical reasons are known for choosing any particular mixing distribution, but the most common choice is the discrete uniform distribution $a_i = 1/d$, perhaps because of lack of imagination and spirit of adventure in MCMC practitioners.

### 3.1.2 Non-Finite Mixtures

Mixtures can use any mixing distribution, discrete or continuous. We need an argument that this is o.k. when the mixture is not finite.

**Theorem 3.1.** Suppose $\mu$ is a probability distribution and for each $z$ in the domain of $\mu$ there is a Markov kernel $P_z$ satisfying $\pi = \pi P_z$, and suppose that the map $(z, x) \mapsto P_z(x, A)$ is jointly measurable for each $A$. Then
\[ Q(x, A) = \int \mu(dz) P_z(x, A) \]
defines a kernel $Q$ that is Markov and satisfies $\pi = \pi Q$.

**Proof.** First we need to show that $Q$ is a kernel. The double integral
\[ \iint (\pi \times \mu)(dx, dz) P_z(x, A) \]
exists because $(x, z) \mapsto P_z(x, A)$ is jointly measurable and bounded. Hence $x \mapsto Q(x, A)$ is measurable (one of the conclusions of the Fubini theorem). To check that $A \mapsto Q(x, A)$ is a measure, we need only check countable additivity. If $A_n \uparrow A$, then
\[ \lim_{n \to \infty} Q(x, A_n) = Q(x, A) \]
by the monotone convergence theorem.

The Markovness of $Q$ is obvious. That $\pi$ is invariant for $Q$ is just the Fubini theorem
\[
\begin{align*}
\int \pi(dx)Q(x, A) &= \int \pi(dx) \int \mu(dz) P_z(x, A) \\
&= \int \mu(dz) \int \pi(dx) P_z(x, A) \\
&= \int \mu(dz) \pi(A) \\
&= \pi(A)
\end{align*}
\]
3.1.3 The Hit-and-Run Algorithm

An example of a non-finite mixing distribution is the so-called “hit-and-run” algorithm (Bélisle, Romeijn, and Smith 1993; Chen and Schmeiser 1993). In its simplest form this algorithm is just a mixture of Gibbs updates that condition on a direction in the state space.

Consider a bounded set $A$ in $\mathbb{R}^d$. A conventional Gibbs sampler uses $d$ updates, one for each coordinate. The $i$-th update updates the $i$-coordinate, giving it a new value simulated from its conditional distribution given the rest of the coordinates, which is uniform on some set.

If the region $A$ is a rectangle parallel to the coordinate axes, the sampler produces i. i. d. samples. Starting at the point $(x_1, y_1)$ in the figure, it simulates a new $x$ value uniformly distributed over its possible range thereby moving to a position uniformly distributed along the horizontal dashed line, say to $(x_2, y_1)$. Then it simulates a new $y$ value uniformly distributed over its possible range thereby moving to a position uniformly distributed along the vertical dashed line, say to $(x_2, y_2)$. This clearly produces a point uniformly distributed in the rectangle and uncorrelated with the previous point.

If the region $A$ is not a rectangle parallel to the coordinate axes, then the Gibbs sampler has autocorrelation.
The update moves are still parallel to the coordinate axes. The possible range of values for each update is the intersection of a horizontal or vertical line, as the case may be, with $A$. Clearly, starting from the point $(x_1, y_1)$ shown in the figure, it would take several moves to get into the upper half of the rectangle. Conclusion: the Gibbs sampler for the second rectangle is less efficient.

This example is an important toy problem. What it lacks in realism, it makes up for in simplicity. It is very easy to visualize this Gibbs sampler. Moreover, it does share some of the characteristics of realistic problems.

**Example 3.2. Hit-and-Run Sampler for a Uniform Distribution.**

The hit-and-run sampler is almost the same as the Gibbs sampler, except that it moves in an arbitrary direction. A hit-and-run step simulates a random angle $\theta$ uniformly distributed between 0 and $2\pi$. Then it simulates a new point uniformly distributed along the intersection of $A$ and the line through the current point making angle $\theta$. 
CHAPTER 3. BASIC ALGORITHMS

It is obvious from the figure that some hit-and-run update steps move farther than Gibbs update steps. Some hit-and-run steps, not many, only those in a fairly small range of angles, can go from one end of the rectangle to the other. No Gibbs update step can do that.

Tentative conclusion: the hit-and-run sampler is more efficient than the Gibbs sampler. Is that right? When we think about the comparison a bit more deeply we see that it is not at all obvious that hit-and-run is better. If we really want to know, we will have to do some simulation experiments and see.

3.1.4 Random Sequence Scans

Composition and mixing are the only ways to combine kernels, since multiplication and convex combination are the only operations that combine kernels to make other kernels, but we can mix a set of kernels that are themselves products of other kernels.

The best known example of combining composition and mixing is the so-called “random sequence scan.” If there are \( d \) elementary update mechanisms having kernels \( P_1, \ldots, P_d \), a random sequence scan chooses a random permutation \((k_1, k_2, \ldots, k_d)\) of the integers \( 1, 2, \ldots, d \) and then applies the updates in that order. We may use any distribution for the mixing distribution. If we let \( \mathcal{P} \) denote the set of all \( d! \) permutations, then a mixing distribution is given by real numbers \( a_k, k \in \mathcal{P} \) that are nonnegative and sum to one. The random sequence scan update can then be described as follows.

1. Choose a random permutation \( k = (k_1, \ldots, k_d) \in \mathcal{P} \), choosing \( k \) with
2. Update the state using the composite update mechanism with kernel $P_{k_1} \cdots P_{k_d}$.

The composite update mechanism referred to in step 2 first does the update with kernel $P_{k_1}$, next the update with kernel $P_{k_2}$, and so forth. The whole random sequence scan update has kernel

$$P = \sum_{(k_1, \ldots, k_d) \in P} a_k P_{k_1} \cdots P_{k_d}. \quad (3.1)$$

This is clearly a mixture, the mixing distribution being the uniform distribution on $P$, and the kernels being mixed having the form $P_{k_1} \cdots P_{k_d}$.

When $a_k = 1/d!$ for all $k$, we say we are using a uniform random sequence scan, but the “uniform” is often dropped. As with the simple random scan, the uniform mixing distribution seems to be the default. An efficient procedure for producing uniform random permutations is given by Knuth (1998, p. 145). It uses computer memory and time proportional to $d$ to generate the random permutation. Since it also takes time proportional to $d$ to execute the scan, this is a minor issue, but there is some reason to consider random sequence scans that don’t require additional memory proportional to $d$.

For example, we could choose uniformly at random from among the $2(d-1)$ permutations that cycle through the integers in normal or reversed order. With four variables these permutations are

$$1234 \quad 2341 \quad 3412 \quad 4123$$
$$4321 \quad 3214 \quad 2143 \quad 1432$$

This random sequence scan uses only two random variates per iteration, one to decide whether to cycle forward or backward and one to decide which update to start with. The uniform random sequence scan needs $d-1$ random variates to generate a random permutation.

### 3.1.5 Auxiliary Variable Random Sequence Scans

Random scan and random sequence scan have an odd property when used with Gibbs updates. Gibbs updates are idempotent, that is, they satisfy $P^2 = P$ (Exercise 3.1). Thus whenever a scan starts with the same update that ended the preceding scan, no progress is made, but we cannot just omit the useless update, because then we would not have a Markov chain. For example if there are two updates with kernels $P_1$ and $P_2$ and we are using simple random scan and the first 10 updates are $P_1P_2P_1P_2P_1P_2P_1P_2P_1P_1$, then the distribution of $X_{10}$ given $X_0$ is

$$P_1P_2P_1P_2P_1P_2P_1P_2P_1P_1 = P_1P_2P_1P_2P_1P_2P_1$$

But we cannot use the kernel on the right hand side, because we must do 10 elementary updates and output the state $X_n$ after each one.
Of course, this problem only occurs in $1/d$ scans on average, so is not serious when $d$ is large. Even when $d$ is small, it does not affect correctness only efficiency. Still there is some reason to see whether we can find a random sequence scan that never repeats an update consecutively.

To accomplish this we need a new idea: let the random sequence we choose depend on the preceding one. If this is not to destroy the Markov property, we must enlarge the state space to include the scan sequence and verify that we still have a Markov chain with the desired invariant distribution. This trick of enlarging the state space is widely used in MCMC under the name “auxiliary variable methods.” We will see it again and again.

Suppose we try choosing a scan sequence uniformly at random from all possible scans that do not begin with same elementary update that was the end of the preceding scan, so there are no repeats of elementary updates. Then the scan chosen depends on the index of the last elementary update of the preceding scan. In order to continue using Markov chain theory, we must add that index to the state space.

You can do anything in MCMC, but everything the update depends on must be part of the state.

If the original state space was $S$, then the enlarged state space is $D \times S$, where $D = \{1, \ldots, d\}$ is the index set of the updates. The Markov chain we simulate will have the form $(I_n, X_n)$, $i = 1, 2, \ldots$, where $I_n \in D$ and $X_n \in S$. The “auxiliary variable random sequence scan” update can now be described as follows.

- Choose a scan sequence $k_1, \ldots, k_d$ uniformly from the permutations of $(1, \ldots, d)$ not beginning with $I$. Set $I = k_d$.
- Update $X$ using the update mechanism with kernel $P_{k_1} \ldots P_{k_d}$.

In a Markov chain problem, the “given” is a probability distribution $\pi$ on $S$ that we want to study. If we had not enlarged the state space, $\pi$ would have been the invariant distribution of our Markov chain. Now, however, the invariant distribution of the chain (assuming it has one) will be a distribution on $D \times S$, since that is now the state space. In order for the new Markov chain to be of any use in learning about $\pi$, we need the $X_n$ to still have marginal distribution $\pi$. Thus the marginal for $X$ of the invariant distribution should be $\pi$. Since the update mechanism for $X$ preserves $\pi$ regardless of the index $I$ and all of the index values are treated the same, it stands to reason that the invariant distribution if $\mu \times \pi$ where $\mu(i) = 1/d$.

We must, of course, check that this guess is correct. The kernel of the update on the enlarged state space can be written

$$P((i, x), \{j\} \times A) = \frac{1}{(d - 1) \cdot (d - 1)!} \sum_{(k_1, \ldots, k_d) \in P \atop k_1 \neq i \atop k_d = j} P_{k_1} \ldots P_{k_d}(x, A),$$
CHAPTER 3. BASIC ALGORITHMS

where $P$ is the set of all permutations of indices as in Section 3.1.4, the factor $(d - 1) \cdot (d - 1)! = d! - (d - 1)!$ being the number of permutations that do not start with $i$. If we left multiply by $\mu \times \pi$, we get

$$\frac{1}{d} \sum_{i=1}^{d} \int \pi(dx)P((i, x), \{j\} \times A)$$

$$= \frac{1}{(d - 1) \cdot d!} \sum_{i=1}^{d} \sum_{(k_1, \ldots, k_d) \in P} \sum_{\substack{k_1, \ldots, k_d \in P \\text{if} \ k_1 \neq i \text{ and} \ k_d = j}} \int \pi(dx)P_{k_1} \ldots P_{k_d}(x, A)$$

$$= \pi(A) \frac{1}{d}$$

because each $P_k$ preserves $\pi$ so the integral is $\pi(A)$ and the result must integrate to one with respect to $\mu \times \pi$. Thus we have proved that this update does preserve $\mu \times \pi$.

There is something strange about the case $d = 2$. There is no longer any randomness in the scan orders. If we start with $I = 2$, then we must use the scan $P_1 P_2$ and have $I = 2$ at the end of the scan. So every scan uses the same order and $I_n = 2$ for all $n$. Similarly, if we start with $I = 1$. Thus the method is essentially fixed scan. We choose one scan order at the beginning and use it ever after.

3.1.6 Subsampling a Markov Chain

Powers are a special case of kernel multiplication (composition). If $P$ is a Markov kernel, so is $P^n$, and if $P$ preserves $\pi$, so does $P^n$. Doing one $P^n$ update is the same as doing the $P$ update $n$ times. Hence the algorithm that does $n$ update steps between each “sample” that is used in subsequent calculations has kernel $P^n$. In effect we run the chain with kernel $P$, but only use $X_n, X_{2n}, X_{3n}, \ldots$. This is called subsampling the chain.

If we take a mixture of powers, we get a randomly subsampled chain. Consider a sampling distribution on the nonnegative integers giving probability $a_n$ to $n$. Then the kernel of the mixture is

$$P_a = \sum_{n=0}^{\infty} a_n P^n$$

(recall that $P^0 = I$). We are assured by our theorem about mixtures that this kernel preserves $\pi$.

What simulation has $P_a$ as its kernel? Just follow the instructions for a random mixture.

• Generate a random nonnegative integer $N_i$ with distribution $a_i$, i.e., $P(N_i = n) = a_n$. 

• Run the chain having kernel $P$ for $N_i$ steps. (Running for zero steps means doing nothing.)
• Output the current state as $Y_i$. (This means $Y_i = Y_{i-1}$ if $N_i = 0$.)
• Set $i = i + 1$ and repeat.

If $X_1, X_2, \ldots$ are a Markov chain with transition probability kernel $P$, then $Y_1, Y_2, \ldots$, where

$$Y_k = X_{N_1 + \cdots + N_k}$$

is a Markov chain with transition probability kernel $P_a$.

Curiously the notion of subsampling a chain at a fixed interval, using the kernel $P^n$, is very widely used, probably overused, in MCMC. But random subsampling, using the kernel $P_a$, is almost never used. This is surprising because random subsampling, using the kernel $P_a$ is a major tool of Markov chain theory, used again and again in (Meyn and Tweedie 1993, Section 5.5). They call the notion “sampled chains” rather than our “subsampled,” but the concept is the same.

### 3.1.7 Preserving Reversibility

Reversibility of a Markov chain is not necessary for MCMC and much of the literature ignores reversibility. However, reversibility does have some theoretical and practical consequences (Besag and Clifford 1989; Geyer 1992), and most elementary update mechanisms that have been proposed for MCMC are reversible, because the easiest way to show that an update mechanism preserves a specified distribution is to show that it is reversible with respect to that distribution. Hence the only way anyone makes a Markov chain for Monte Carlo that is non-reversible is to combine reversible elementary update steps in a nonreversible way. This is all right if one doesn’t care whether the sampler is reversible, but one should know how to obtain a reversible sampler.

Suppose that we have $d$ elementary update mechanisms with kernels $P_i$ that are reversible with respect to the same distribution $\pi$. Let us see whether composition and mixing preserve reversibility.

#### Composition

When we combine by composition, we immediately see that reversibility is not, in general, preserved. Since $P_1$ and $P_2$ are self-adjoint operators on $L^2(\pi)$,

$$(f, P_1 P_2 g) = (P_1 f, P_2 g) = (P_2 P_1 f, g), \quad f, g \in L^2(\pi),$$

and this says the adjoint of $P_1 P_2$ is $P_2 P_1$. Thus the composition is self-adjoint if and only if $P_1 P_2 = P_2 P_1$, that is, if $P_1$ and $P_2$ are commuting operators on $L^2(\pi)$. In general the elementary update operators do not commute and hence the composition is not self-adjoint and reversibility is not preserved. Similarly, for $d$ operators, the adjoint of $P_1 \cdots P_d$ is $P_d \cdots P_1$, and reversibility is not preserved.
Some special forms of composition do, however, preserve reversibility. Consider the “scan” \( P_1 P_2 P_2 P_1 \). Its adjoint has the operators multiplied together in reversed order, but that gives us the same thing again. Hence it is self-adjoint.

Let us say a composition of operators is \textit{palindromic} if it reads the same forwards and backwards.\(^1\) Then it is obvious that any palindromic composition of self-adjoint operators is self-adjoint and preserves reversibility.

\section*{Mixing}

What happens when we combine by mixing? Now it is obvious that reversibility is preserved. Since \( P_1 \) and \( P_2 \) are self-adjoint operators on \( L^2(\pi) \),
\[
(f, [aP_1 + bP_2]g) = a(f, P_1 g) + b(f, P_2 g)
\]
\[
= a(P_1 f, g) + b(P_2 f, g)
\]
\[
= ([aP_1 + bP_2]f, g), \quad f, g \in L^2(\pi),
\]
and this says \( aP_1 + bP_2 \) is self-adjoint for any real scalars \( a \) and \( b \). This obviously extends to arbitrary linear combinations, even to arbitrary non-finite mixtures (Exercise 3.2).

\section*{Random Sequence Scans}

The the kernel (3.1) is self-adjoint if
\[
\sum_{(k_1, \ldots, k_d) \in \mathcal{P}} a_k P_{k_1} \cdots P_{k_d} = \sum_{(k_1, \ldots, k_d) \in \mathcal{P}} a_k P_{k_d} \cdots P_{k_1}. \tag{3.4}
\]
If we define an operator \( r \) (for reverse) on \( \mathcal{P} \) by \( r((k_1, \ldots, k_d)) = (k_d, \ldots, k_1) \), then (3.4) holds if \( a_r(k) = a_k \) for all \( k \). In words, a random sequence scan is reversible if each scan sequence has the same probability as its reverse sequence. Both of the specific methods discussed in Section 3.1.4 have this property.

\subsection*{3.1.8 State-Dependent Mixing}

Green (1995) proposed an algorithm that involves \textit{state-dependent mixing} having mixing probabilities that depend on the current state. Even in the case of finite mixtures, the theory developed so far does not work. Consider a mixing distribution with probabilities \( a_i(x) \) that depend on the current state \( x \). That is, we propose to use the kernel
\[
P(x, A) = \sum_{i=1}^d a_i(x) P_i(x, A)
\]

\(^1\)A \textit{palindrome} is a phrase that reads the same forwards and backwards, such as “Able was I ere I saw Elba.”
Now $\pi P = \pi$ is
\[
\sum_{i=1}^{d} \int \pi(dx) a_i(x) P_i(x, A) = \pi(A),
\]
and this equation is no longer easy to verify. It is not implied by $\pi P_i = \pi$ for each $i$. The problem is that multiplication of a kernel by $a_i(x)$ is not multiplication of the operator $P_i$ by a scalar. In fact, this operation is another kernel multiplication. Define the kernel
\[
M_i(x, B) = a_i(x) I(x, B)
\]
and check that
\[
(M, P_i)(x, B) = \int a_i(x) I(x, dy) P_i(y, B) = a_i(x) P_i(x, B).
\]
Now we see that in operator notation
\[
P = \sum_{i=1}^{d} M_i P_i
\]
There is no reason why $P$ should preserve $\pi$ whenever all the $P_i$ do, because $M_i$ does not preserve $\pi$.

Green’s ingenious notion was to use reversibility directly. Define $K_i = M_i P_i$, written out in full
\[
K_i(x, A) = a_i(x) P_i(x, A).
\]
Suppose each $K_i$ is reversible with respect to $\pi$, that is, satisfies (2.24) with $P$ replaced by $K_i$. Then clearly $P$ is also reversible with respect to $\pi$. If $P$ is Markov, then it does everything we want.

Thus we are lead to treating the $K_i$ rather than the $P_i$ as the primary objects. Let us see what the relation between the two is. Since $a_i(x)$ is a probability, it is between zero and one. Hence
\[
K_i(x, B) \geq 0, \quad B \in \mathcal{B} \tag{3.6a}
\]
\[
K_i(x, S) \leq 1, \tag{3.6b}
\]
where $(S, B)$ is, as usual, the state space. A kernel having these properties is called substochastic. Using
\[
a_i(x) = K_i(x, S) \tag{3.7}
\]
we see that
\[
P_i(x, A) = \frac{K_i(x, A)}{K_i(x, S)} \tag{3.8}
\]
So (3.5) and the pair of equations (3.7) and (3.8) can be used to go back and forth between $K$’s and $P$’s, and we may consider that we have been given the $K_i$ to specify the algorithm.

As in Theorem 3.1 we can consider arbitrary mixtures. For those we change the index from $i$ to $z$. 
CHAPTER 3. BASIC ALGORITHMS

Theorem 3.2. Suppose \( \mu \) is a \( \sigma \)-finite positive measure and for each \( z \) in the domain of \( \mu \) there is a substochastic kernel \( K_z \) that is reversible with respect to \( \pi \), suppose that the map \( (z, x) \mapsto K_z(x, A) \) is jointly measurable for each \( A \), and suppose
\[
\int \mu(dz)K_z(x, A) \leq 1, \quad x \in S, \ A \in B.
\]
Then
\[
Q(x, A) = \int \mu(dz)K_z(x, A)
\]
defines a substochastic kernel \( Q \) that is reversible with respect to \( \pi \).

Proof. The proof that \( Q \) is a kernel is exactly like the proof in Theorem 3.1. That \( Q \) is substochastic is again obvious. By the Fubini theorem
\[
\int_A \pi(dx)Q(x, B) = \int_A \pi(dx) \int \mu(dz)K_z(x, B) = \int \mu(dz) \int_A \pi(dx)K_z(x, B).
\]
Reversibility of \( Q \) with respect to \( \pi \) is the property that the left hand side of (3.11) is unchanged by swapping \( A \) and \( B \), which is true because swapping \( A \) and \( B \) in the right hand side leaves it unchanged by the reversibility of each \( K_z \). \( \square \)

This theorem is often used in the case where \( \mu \) is counting measure, so for ease of reference we state that as a corollary.

Corollary 3.3. Suppose \( \{ K_i : i \in I \} \) is a family of substochastic kernels, each reversible with respect to \( \pi \), and suppose
\[
\sum_{i \in I} K_i(x, A) \leq 1, \quad x \in S, \ A \in B.
\]
Then
\[
Q(x, A) = \sum_{i \in I} K_i(x, A)
\]
defines a substochastic kernel \( Q \) that is reversible with respect to \( \pi \).

Remark. If the index set \( I \) is finite or countable, the meaning of the sums is obvious. If \( I \) is uncountable, the sum means integration with respect to counting measure on \( I \), that is,
\[
\sup_{F \text{finite}} \sum_{i \in F} K_i(x, A)
\]
The kernel \( Q \) defined in the corollary will be stochastic (Markov) if and only if the mixing probabilities \( a_i(x) = K_i(x, S) \) sum to one for each \( x \). Sometimes this is hard to verify (more precisely, it is hard to invent \( K_i \) having this property). Then a simple trick allows us to use the corollary anyway. Define the defect
\[
d(x) = 1 - \sum_{i \in I} K_i(x, S), \quad x \in S
\]
CHAPTER 3. BASIC ALGORITHMS

and a new kernel
\[ \tilde{K}(x, A) = d(x)I(x, A). \]  
(3.13)

Then \( \tilde{K} \) is reversible with respect to any distribution \( \pi \) since
\[ \iint f(x)g(y)\pi(dx)\tilde{K}(x, dy) = \iint f(x)g(y)d(x)\pi(dx) \]
is trivially symmetric under the interchange of \( f \) and \( g \). If we add \( \tilde{K} \) to our set of kernels, then the sum is stochastic.

Thus we have the following formulation of state-dependent mixing. Suppose we are given a family of substochastic kernels as described in the corollary. Then the combined update described as follows

1. Choose a random index \( i \in I \), choosing index \( i \) with probability \( p_i(x) \) defined by (3.7). With probability (3.12) skip step 2 and stay at the current position.

2. Simulate a new value of \( x \) from the probability distribution \( P_i(x, \cdot) \) defined by (3.8).

has the stochastic transition kernel \( \tilde{K} + \sum_i K_i \) and is reversible with respect to \( \pi \) if each of the \( K_i \) is reversible with respect to \( \pi \).

In the general case described by the theorem, the algorithm is a bit more complicated to describe, partly because the notation is a bit confusing. Now the probability of using the kernel \( K_z \) is denoted \( a_z(x) = K_z(x, S) \), and we need to think of this as a subprobability density with respect to \( \mu \), but in that role \( z \) is the variable, \( x \) being fixed. So let us write \( f_z(z) = a_z(x) \). Then
\[ \int f_z(z)\mu(dz) \leq 1 \]
by (3.9) so \( f_z \) is indeed a subprobability density. The defect of \( f_z \) is
\[ d(x) = 1 - \int f_z(z)\mu(dz), \]  
(3.14)
and we define \( \tilde{K} \) by (3.13) exactly as before except that the defect is defined by (3.14) rather than (3.12).

In order to carry out the combined update described by the theorem, we need to be able to simulate a random variate \( Z \) having this density with respect to \( \mu \). The update is described as follows.

1. Simulate a random variate \( z \) having probability density function \( f_z \) with respect to \( \mu \). With probability (3.14) skip step 2 and stay at the current position.

2. Simulate a new value of \( x \) from the probability distribution \( P_z(x, \cdot) \) defined by (3.8) with \( i \) replaced by \( z \).
3.2 The Metropolis-Hastings Algorithm

In one form (Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller 1953), this is the oldest MCMC algorithm, dating to the dawn of the computer age when the only place something like this could have been done was Los Alamos. In its modern form (Green 1995), it is the newest MCMC algorithm, which solves many problems MCMC researchers have stumbled over in the past. In between, a key improvement was made by Hastings (1970), which in a curious episode in the sociology of science was not really understood for 20 years. The paper was published in a prestigious journal (Biometrika) and was cited by some MCMC authors (Ripley 1987), but many problems that now seem trivial (“just use Metropolis-Hastings”) were stumbled over because the importance of Hastings’ improvement was not understood.

### 3.2.1 Unnormalized Probability Densities

The section heading refers to a concept that is familiar, being a standard problem in introductory probability courses, but usually is not given a name. Here we do give it a name so we can use it better. A function $h$ is an unnormalized probability density with respect to a positive measure $\mu$ if $h$ is nonnegative and has a finite, nonzero integral. Then the integral $c = \int h(x)\mu(dx)$ is called the normalizing constant for $h$, and the function $f$ defined by $f(x) = h(x)/c$ is called the normalized density corresponding to $h$.

As we said, this concept is very familiar from introductory probability problems like: What constant $k$ makes $kx^2$ a probability density for $0 < x < 1$? But lack of a name for this concept keeps people from noticing that it plays a key role in several areas of statistics.

It is part of the definition, but it needs to be emphasized that calling $h$ and unnormalized density asserts

- it is nonnegative,
- it does not integrate to zero (i.e., is strictly positive on some set having positive $\mu$-measure), and
- it does not integrate to infinity.

Checking the first two items is trivial. Checking the third is nontrivial, but it must be done. Arguments about “unnormalized densities” that integrate to infinity are mathematical nonsense.

### Bayesian Inference

The computational problems that make Bayesian inference difficult all involve unnormalized densities, the reason being

$\text{likelihood} \times \text{prior} = \text{unnormalized posterior}$
CHAPTER 3. BASIC ALGORITHMS

If a Bayesian has a data model \( f(x|\theta) \) and a prior \( g(\theta) \), the problem is to calculate properties of the posterior

\[
h(\theta|x) = \frac{f(x|\theta)g(\theta)}{\int f(x|\theta)g(\theta) \, d\theta}
\]  

(3.15)

Because \( f \) and \( g \) appear in both the numerator and the denominator, both may be unnormalized, considered as functions of \( \theta \). Unnormalized versions of \( f(x|\theta) \) are a concept with a name. A function \( L_x(\theta) \) is a likelihood for the problem if

\[
L_x(\theta) = a(x)f(x|\theta)
\]

for an arbitrary strictly positive function \( a(x) \). If we plug this into (3.15) we get

\[
h(\theta|x) = \frac{L_x(\theta)g(\theta)}{\int L_x(\theta)g(\theta) \, d\theta}
\]  

(3.16)

(the \( a(x) \) terms in the numerator and denominator cancel). It is also clear that we could plug in \( cg(\theta) \) for \( g(\theta) \) for an arbitrary positive constant \( c \) and the \( c' \)'s would cancel, leaving the result unchanged.

Equation (3.16) even makes sense when \( g \) is not an unnormalized density. It can be any nonnegative function on the parameter space, so long as the numerator \( L_x(\theta)g(\theta) \) is an unnormalized density. When \( g(\theta) \) does not integrate, we say it is an improper prior.

When the prior is proper, there is no need to show that the likelihood times the prior is integrable. It is automatically integrable by the laws of probability. The integral of the numerator in (3.15) is the marginal density for \( x \), which is finite. When the prior is improper, a proof that the likelihood times the prior is integrable is a required part of the problem. Omitting the proof risks committing nonsense.\(^2\)

**Conditioning and Unnormalized Densities**

Not surprising, Bayes rule being just a rearrangement of the definition of conditional probability, the relationship between unnormalized densities and conditioning we saw in Bayesian inference is a general phenomenon

\[
A \text{ joint density is an unnormalized conditional density. The marginal is its normalizing constant.}
\]

\(^2\)It happened once to your humble author (Geyer 1992, see the “Note added in proof”). Don’t let it happen to you. There is some MCMC literature on what happens when you try to simulate an “improper posterior” (you omitted the proof of integrability, and there isn’t a proof, and you are in the realm of mathematical nonsense), but a short digest of that literature is that there is nothing to be said, no one has a clue about what will happen. Moreover, the whole notion of “improper posterior” seems to have no theoretical foundation. Even if you could simulate it in some sense, no Bayesian theoretician I’ve talked to thinks it has any meaning.
CHAPTER 3. BASIC ALGORITHMS

What this means is the following. Say \( f(x, y) \) is a joint density considered as a function of two variables \( x \) and \( y \). Considered as a function of one variable, say \( x \), it is an unnormalized density defined by

\[
h_y(x) = f(x, y).
\]

The normalizing constant for \( h_y \) is the marginal of \( y \)

\[
p(y) = \int h_y(x) \, dx = \int f(x, y) \, dx.
\]

Really we should call \( p(y) \) a “normalizing function” rather than “normalizing constant” because it is a function of \( y \). Dividing by the normalizing function gives the conditional density

\[
f(x|y) = \frac{f(x, y)}{p(x)}.
\]

The same phenomenon holds when the joint distribution is unnormalized, but we have to be a bit careful with our terminology. Suppose we now have the unnormalized density \( h(x, y) = cf(x, y) \), where \( c \) is an unknown constant. Again, we write

\[
h_y(x) = h(x, y),
\]

but now the normalizing function is not the marginal, though it is proportional to the marginal

\[
c(y) = \int h_y(x) \, dx = \int cf(x, y) \, dx = cp(y).
\]

But still, normalizing \( h_y \) gives the conditional density

\[
\frac{h_y(x)}{c(y)} = \frac{h(x, y)}{c(y)} = \frac{f(x, y)}{p(x)} = f(x|y)
\]

Models Specified by Unnormalized Densities

If for each \( \theta \) in a parameter space \( \Theta \) we have a function \( h_\theta \) that is an unnormalized probability density with respect to \( \mu \), we say that the family of unnormalized densities \( \{ h_\theta : \theta \in \Theta \} \) is a family of unnormalized densities. Again the normalizing constants

\[
c(\theta) = \int h_\theta(x) \mu(dx), \quad \theta \in \Theta
\]

define a function \( c : \Theta \to (0, \infty) \) called the normalizing function of the family. As always, the use of the term “unnormalized densities” implies that \( 0 < c(\theta) < \infty \) for all \( \theta \). The normalized densities of the family are defined by

\[
f_\theta(x) = \frac{1}{c(\theta) h_\theta(x)}, \quad x \in S
\]

(3.17)
CHAPTER 3. BASIC ALGORITHMS

(Where \( S \) is, as usual, the sample space).

This notion may seem unfamiliar, but it is a widely used technique for specifying models for complicated phenomena. It may be very difficult to specify a model for which the normalizing constant is known for complicated data. As we will see, it is not necessary to have a closed-form expression for the normalizing constant in order to use the family as a statistical model. We will always be able to simulate data from the model by MCMC, and

\[
\text{when we can simulate, we can do inference.}
\]

This assertion may be a bit hard to swallow until some examples have been seen, but we will see them in due course.

3.2.2 The Metropolis-Hastings Update

The Metropolis-Hastings update preserves any distribution \( \pi \) specified by an unnormalized density \( h \) with respect to a measure \( \mu \). There is no restriction on \( h(x) \) other than that it actually be an unnormalized density (its normalizing constant is nonzero and finite) and that it can be evaluated, that is, for each \( x \) we can calculate \( h(x) \). There is no requirement that we be able to do any integrals or know the value of the normalizing constant. In particular, unlike the Gibbs sampler, we do not need to know anything about any conditional distributions of \( \pi \).

The Metropolis-Hastings update uses an auxiliary transition probability specified by a density \( q(x, y) \) called the proposal distribution. For every point \( x \) in the state space, \( q(x, \cdot) \) is a (normalized) probability density with respect to \( \mu \) having two properties: for each \( x \) we can simulate a random variate \( y \) having the density \( q(x, \cdot) \) and for each \( x \) and \( y \) we can evaluate the \( q(x, y) \). To summarize, this is what we need:

1. For each \( x \) we can evaluate \( h(x) \).
2. For each \( x \) and \( y \) we can evaluate \( q(x, y) \).
3. For each \( x \) we can simulate a random variate with density \( q(x, \cdot) \) with respect to \( \mu \).

There is no necessary connection between the auxiliary density \( q(x, y) \) and the density \( h(x) \) of the stationary distribution. We can choose any density that we know how to simulate. For example, if the state space is \( d \)-dimensional Euclidean space \( \mathbb{R}^d \) we could use a multivariate normal proposal density with mean \( x \) and variance a constant times the identity. If \( \phi \) denotes a Normal\((0, \sigma^2 I)\) density, then we have \( q(x, y) = \phi(y - x) \). We can easily simulate multivariate normal variates and evaluate the density.

The Metropolis-Hastings update then works as follows. The current position is \( x \), and the update changes \( x \) to its value at the next iteration.

1. Simulate a random variate \( y \) having the density \( q(x, \cdot) \).
2. Calculate the “Hastings ratio”

\[ R = \frac{h(y)q(y, x)}{h(x)q(x, y)} \]  \hspace{1cm} (3.18)

3. Do “Metropolis rejection” with probability \( \min(1, R) \) set \( x = y \).

Later in this section we will prove that this update always preserves \( \pi \).

We often say we “accept” the “proposal” \( y \) if we set the value \( x = y \) in step 3. Otherwise we say we “reject” the proposal. When we reject, the value of the state of the Markov chain remains the same for two consecutive iterations.

**Warning:** Those familiar with so-called rejection sampling in ordinary Monte Carlo note that Metropolis rejection is completely different. In ordinary rejection sampling, proposals are made over and over until one is accepted. The first proposal accepted is the next sample. In Metropolis rejection only one proposal is made, if it is not accepted, then the Markov chain doesn’t move and \( X_{n+1} \) is equal to \( X_n \). If Metropolis rejection were done like ordinary rejection, the resulting Markov chain would not preserve \( \pi \).

Note also that the denominator of the Hastings ratio (3.18) can never be zero if the chain starts at a point where \( h(x) \) is nonzero. A proposal \( y \) such that \( q(x, y) = 0 \) occurs with probability zero, and a proposal \( y \) such that \( h(y) = 0 \) is accepted with probability zero. Thus there is probability zero that denominator of the Hastings ratio is ever zero during an entire run of the Markov chain so long as \( h(X_1) > 0 \). If we do not start in the support of the stationary distribution we have the problem of defining how the chain should behave when \( h(x) = h(y) = 0 \), that is, how the chain should move when both the current position and the proposal are outside the support of the stationary distribution. The Metropolis-Hastings algorithm says nothing about this. It is a problem that is best avoided by starting at a point where \( h(x) \) is positive.

Also note specifically that there is no problem if the proposal is outside the support of the stationary distribution. If \( h(y) = 0 \), then \( R = 0 \) and the proposal is always rejected, but this causes no difficulties.

### 3.2.3 The Metropolis Update

The special case when we use a proposal density satisfying \( q(x, y) = q(y, x) \) is called the Metropolis update. In this case the Hastings ratio (3.18) reduces to the odds ratio

\[ R = \frac{h(y)}{h(x)} \]

and there is no need to be able to evaluate \( q(x, y) \) only to be able to simulate it. Thus the requirements for Metropolis are a bit different from those for Metropolis-Hastings

1. For each \( x \) we can evaluate \( h(x) \).
2. \( q(x, y) = q(y, x) \) for each \( x \) and \( y \).

3. For each \( x \) we can simulate a random variate with density \( q(x, \cdot) \) with respect to \( \mu \).

(the first and third requirements are unchanged, only the second is different).

Metropolis proposals save the trouble of evaluating \( q(x, y) \) in calculating the Hastings ratio. Evaluating \( q(x, y) \) is usually not that much work, so avoiding it is not worth much additional trouble in making proposals.

\emph{Gibbs and Metropolis are all right when they are easy and effective. Otherwise they are part of the problem, not part of the solution.}

Always keep the general method in mind (for now “general” means Metropolis-Hastings, later it will mean Metropolis-Hastings-Green).

### 3.2.4 A Good Default MCMC Sampler

The objective of this section is to outline a good “default” MCMC sampler. One way to think of what we are looking for is a method that will give reasonably good answers with a minimum of trouble.

The normal proposal mentioned above is a Metropolis proposal. By the symmetry of the multivariate normal distribution, \( q(x, y) = \phi(y - x) \) is equal to \( q(y, x) = \phi(x - y) \), where \( \phi \) is any non-degenerate multivariate normal density, that is, the proposal is \( y \sim \text{Normal}(x, \Sigma) \), where \( \Sigma \) is any positive-definite matrix and \( x \) is the current position.

Although there are good reasons for using this method with general \( \Sigma \), a method that asks the user to specify an arbitrary covariance matrix having the dimension of the state space has to many parameters to be considered easy to use. So we will restrict \( \Sigma \) to be diagonal. If the coordinate variables of the state vector have approximately the same variance under the distribution \( \pi \) we want to simulate, then we can use an even simpler proposal with \( \Sigma = \sigma^2 I \). Now there is only one parameter (\( \sigma \)) that must be adjusted by the user. We can’t do any

\footnote{Another way to think of what we are looking for is a default setting for the worlds most obnoxious seminar question. A statistician who shall remain nameless often asks seminar questions of the following form: “The most simple minded approach to this problem I can think of is blah. Can you explain why your method works any better than that?” Here “blah” stands for any really simple method, preferably one that can be explained in one sentence and took about fifteen seconds to think up. The reason the question is so obnoxious is that many people do write papers and give talks about very complicated methods that can be proved to have various properties, but cannot be proved to be better than the “most simple minded approach” I can think of. If the speaker understands the question, he is left with nothing to say. If the speaker doesn’t get the point, and blathers on without addressing the issue of whether is method is good for anything, he seems a fool. In MCMC the method of this section is a good “most simple minded approach.” I can’t tell you how many MCMC talks I’ve heard or papers I’ve read that gave no reason to believe the methods proposed were better than this default. }
better than that. If $\sigma$ is chosen ridiculously small, say $10^{-10}$, the chain can’t get anywhere in any reasonable number of iterations. If $\sigma$ is chosen ridiculously large, say $10^{10}$, all of the proposals will be so far out in the tail that none will be accepted in any reasonable number of iterations. In either case, the chain will not produce a representative sample from its invariant distribution in the amount of time anyone is willing to wait. So we have a “Goldilocks problem.” We don’t want the porridge too cold or too hot. Of course we could choose $\sigma = 1$ and hope that will be about right for most problems, but that seems a too much to hope for.

How do we choose $\sigma$? Gelman, Roberts, and Gilks (1996) considering the performance of this algorithm in simulating multivariate normal distributions showed that adjusting $\sigma$ so that about 20% of proposals are accepted gives the best performance (if you are simulating a multivariate normal). This came as a shock to many MCMC practitioners whose naive intuition told them that high acceptance rates like 90% would be right. So even though the recommendation was not exactly right for any non-toy problem it had a huge effect on practice, because what everyone was doing was grossly wrong. Geyer and Thompson (1995) came to a similar conclusion, that a 20% acceptance rate is about right, in a very different situation. They also warned that a 20% acceptance rate could be very wrong and produced an example where a 20% acceptance rate was impossible and attempting to reduce the acceptance rate below 70% would keep the sampler from ever visiting part of the state space. So the 20% magic number must be considered like other rules of thumb we toss around in statistics: $n > 30$ means the $z$-test is o. k. and more than 5 expected in each cell of a contingency table means the chi-square test is o. k. We know these rules of thumb can fail. There are many examples in the literature where they do fail. We keep repeating them because we want something simple to tell beginners, and they are all right for many problems.

*The rule of thumb says 20% but your mileage may vary.*

From the *Jargon File* (Raymond 1996)

**Your mileage may vary** (YMMV) /caveat/ [from the standard disclaimer attached to EPA mileage ratings by American car manufacturers] 1. A ritual warning often found in Unix freeware distributions. Translates roughly as “Hey, I tried to write this portably, but who knows what’ll happen on your system?” 2. More generally, a qualifier attached to advice. “I find that sending flowers works well, but your mileage may vary.”

**Example 3.3. Bayesian Logistic Regression.**

Here we do Bayesian logistic regression with a flat prior on the kyphosis data that comes with S-PLUS (Chambers and Hastie 1993, pp. 200 ff.). The problem has three predictor variables plus an intercept, so the log likelihood is

$$L(\beta) = \prod_{i=1}^{n} \rho(\beta_i)^{y_i} q(\beta_i)^{1-y_i}$$
where
\[ p(\theta) = \frac{e^\theta}{e^\theta + 1} \quad q(\theta) = 1 - p(\theta) = \frac{1}{e^\theta + 1} \]
and
\[ \theta_i = \beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + x_{i3}\beta_3. \]

The responses \( y_i \) are all zero or one. The covariates \( x_{ij} \) are arbitrary real numbers. Here we use a flat prior \( g(\beta) \equiv 1 \).

A few short runs, the first four lines of the following table, establish that \( \sigma = 0.2 \) is about right.

<table>
<thead>
<tr>
<th>sample size</th>
<th>subsample size</th>
<th>spacing</th>
<th>acceptance rate (%)</th>
<th>computer time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>1</td>
<td>1.00</td>
<td>0.0</td>
<td>2.3</td>
</tr>
<tr>
<td>10000</td>
<td>1</td>
<td>0.10</td>
<td>2.1</td>
<td>2.1</td>
</tr>
<tr>
<td>10000</td>
<td>1</td>
<td>0.01</td>
<td>35.5</td>
<td>2.0</td>
</tr>
<tr>
<td>10000</td>
<td>1</td>
<td>0.03</td>
<td>11.9</td>
<td>2.1</td>
</tr>
<tr>
<td>10000</td>
<td>1</td>
<td>0.02</td>
<td>18.1</td>
<td>2.1</td>
</tr>
<tr>
<td>10000</td>
<td>10</td>
<td>0.02</td>
<td>17.9</td>
<td>18.7</td>
</tr>
<tr>
<td>10000</td>
<td>100</td>
<td>0.02</td>
<td>17.9</td>
<td>187.3</td>
</tr>
</tbody>
</table>

Figure 3.1 shows a time series plot for \( \beta_0 \). Of the four parameters, this one has the worst plot. The series hardly looks stationary. We need a longer run, because we don’t want to fill up the disk, we use a wider spacing. The last line of the table shows a run of \( 10^6 \) iterations, subsampled at every 100 iterations, so we only write out \( 10^4 \) samples. We can’t plot more than that anyway. Figure 3.2 is better than Figure 3.1 but not by much. The chain appears more or less stationary, but has so much autocorrelation that any estimates based on it will have low precision. Since this run only took three minutes we could increase the spacing by a factor of 100 again if we were willing to wait several hours for the results, but we could also think a little bit.

A little though about regression (not about MCMC) comes to the idea that the problem may be ill conditioned because of correlation among the predictor variables (a. k. a. collinearity). This leads to high correlation among the regression coefficients. When we check for that, we see that \( \beta_0 \) and \( \beta_3 \) are fairly highly correlated (Figure 3.3). This leads to the further idea that if we used orthogonal predictors, we might get a better behaved sampler. In fact, since the constant predictor is one of the ones causing trouble, we might just orthogonalize the other predictors to it, i. e., subtract off their means. This is equivalent to a change of parameters. Call the new parameters \( \beta'_i \). Then we have

\[ \beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + x_{i3}\beta_3 = \beta'_0 + (x_{i1} - \bar{x}_1)\beta'_1 + (x_{i2} - \bar{x}_2)\beta'_2 + (x_{i3} - \bar{x}_3)\beta'_3 \]

from which we see

\[ \begin{align*}
\beta_i &= \beta'_i, \\
\beta_0 &= \beta'_0 - \bar{x}_1\beta'_1 - \bar{x}_2\beta'_2 - \bar{x}_3\beta'_3
\end{align*} \]

(3.19)
Figure 3.1: Time series plot of Metropolis sampler output for $\beta_0$ in the four-
parameter logistic regression for the kyphosis data (Chambers and Hastie 1993). The sampler is the “default” Metropolis with $\sigma = 0.02$. 
Figure 3.2: Time series plot of Metropolis sampler output for $\beta_0$ in the same model as in Figure 3.1. The only difference is the chain is subsampled with spacing 100 and runs 100 times as long.
Thus we can easily convert back to the original parameters if we so desire. We may not bother. The coefficient for the constant predictor has to be in the model, but we are not interested in its actual value.

With this change of parameters, things go much better.

<table>
<thead>
<tr>
<th>sample size</th>
<th>subsample spacing</th>
<th>acceptance rate (%)</th>
<th>computer time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>1</td>
<td>0.020</td>
<td>33.4</td>
</tr>
<tr>
<td>10000</td>
<td>1</td>
<td>0.030</td>
<td>22.6</td>
</tr>
<tr>
<td>10000</td>
<td>1</td>
<td>0.040</td>
<td>16.2</td>
</tr>
<tr>
<td>10000</td>
<td>1</td>
<td>0.035</td>
<td>19.0</td>
</tr>
<tr>
<td>10000</td>
<td>10</td>
<td>0.035</td>
<td>18.9</td>
</tr>
<tr>
<td>10000</td>
<td>100</td>
<td>0.035</td>
<td>19.4</td>
</tr>
</tbody>
</table>

One indication we are doing better is that we get higher acceptance rates for the same $\sigma$ or, what is the same thing, can take bigger steps with the same acceptance rate. Figure 3.4 is the analog of Figure 3.1 for the run in line four of this table. Figure 3.4 looks much better than Figure 3.1. We continue making longer runs (the last two lines of the table) and then look at the analog of Figure 3.3. In order to do this we have to transform back to the original parameterization using (3.19).

It is clear that the transformation has turned a moderately hard problem...
Figure 3.4: Time series plot of Metropolis sampler output for $\beta_0$ the same logistic regression data as in 3.1 but using the parameterization (3.19). The length of run and spacing of samples is the same as in 3.1.
Figure 3.5: Scatter plot of $\beta_0$ versus $\beta_3$ for the Metropolis sampler output for the same data, same Monte Carlo sample size and same spacing of subsamples as in Figure 3.3. The only difference is that the parameters $\beta_i'$ were used and then translated back to the original parameterization.
into an easy one. We do not continue with the example, because we have already seen what was to be learned. That we needed a simple trick should not be surprising, nothing in statistics works “right out of the box.” Why would MCMC be an exception?

There are no idiot-proof MCMC methods, not even the “default.”

Example 3.4. The Dumbbell Distribution.
This is a toy problem that shows the 20% rule failing.

A few quick runs show us that $\sigma = 1.3$ is about right according to the 20% rule. But what $\sigma$ is really optimal?

<table>
<thead>
<tr>
<th>sample size</th>
<th>subsample spacing</th>
<th>acceptance rate (%)</th>
<th>computer time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>1</td>
<td>1.0</td>
<td>30.4</td>
</tr>
<tr>
<td>10000</td>
<td>1</td>
<td>2.0</td>
<td>11.1</td>
</tr>
<tr>
<td>10000</td>
<td>1</td>
<td>1.3</td>
<td>21.6</td>
</tr>
</tbody>
</table>

Suppose we are trying to estimate the mean of $x$ (the horizontal coordinate). Of course, we know this is the center of symmetry in this toy problem, but you have to imagine we don’t know the mean and must estimate it. What $\sigma$ gives the most accuracy in estimating the mean?

We look at some more runs, this time also estimating the variance in the central limit theorem $\sigma_{clt}^2$ (1.10) by the method of batch means (Section 1.6.3) with 100 batches.

<table>
<thead>
<tr>
<th>sample size</th>
<th>subsample spacing</th>
<th>acceptance rate (%)</th>
<th>computer time (sec)</th>
<th>$\sigma_{clt}^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^5$</td>
<td>100</td>
<td>1.3</td>
<td>21.3</td>
<td>130.1</td>
</tr>
<tr>
<td>$10^5$</td>
<td>100</td>
<td>2.0</td>
<td>11.4</td>
<td>131.7</td>
</tr>
<tr>
<td>$10^5$</td>
<td>100</td>
<td>3.0</td>
<td>6.2</td>
<td>132.8</td>
</tr>
<tr>
<td>$10^5$</td>
<td>100</td>
<td>4.0</td>
<td>4.0</td>
<td>132.7</td>
</tr>
<tr>
<td>$10^5$</td>
<td>100</td>
<td>5.0</td>
<td>2.9</td>
<td>136.5</td>
</tr>
<tr>
<td>$10^5$</td>
<td>100</td>
<td>6.0</td>
<td>2.3</td>
<td>133.2</td>
</tr>
<tr>
<td>$10^5$</td>
<td>100</td>
<td>7.0</td>
<td>1.8</td>
<td>133.1</td>
</tr>
<tr>
<td>$10^5$</td>
<td>100</td>
<td>8.0</td>
<td>1.5</td>
<td>133.0</td>
</tr>
</tbody>
</table>

It is clear that $\sigma = 1.3$ in not optimal and in fact $\sigma = 7$ is more like it and the optimal acceptance rate is more like 2% than 20%.

I imagine some reader will now protest that most problems are not like the “dumbell distribution” so what is the point? I reply by saying that asking the question like that misses the point. Unlike criminal defend ents, math is guilty
until proven innocent. You are not entitled to assume that “most problems” are not “like” the dumbell distribution until you have a precise definition of the class of problems you are talking about and a proof that 20% acceptance rate (or whatever) is optimal for all problems in the class. As it stands now, we have a counterexample that disproves the conjecture that 20% is optimal for all problems. Until someone comes up with a better conjecture, that’s the end of the story.

I imagine that some readers are still not satisfied. They would be happy to leave math and rely on practical experience. To them I would say that practical experience with complicated problems shows they do have bottlenecks like this toy problem. It is easy for the sampler to move around some parts of the state space, but hard for the sampler to get from one part of the state space to another (through a “bottleneck”). Real problems with bottlenecks tend to be so hard that the kind of experimentation we did here would take a very long time. But there is every reason to suspect that real problems do exhibit phenomena similar to the dumbell distribution.

### 3.2.5 Reversibility of Metropolis-Hastings

We can now write down the transition probability kernel for the Metropolis-Hastings update. The transition probability has two terms. For accepted proposals, we propose \( y \) and then accept it, which happens with probability density

\[
p(x, y) = q(x, y)a(x, y),
\]

where \( a(x, y) = \min(R, 1) \) is the acceptance probability. Hence for any set \( A \)

\[
\int_A q(x, y)a(x, y)\mu(dy)
\]

is the part of \( P(x, A) \) that results from accepted proposals. If the integral on the right hand side is taken over the whole state space, it gives the total probability that the proposal will be accepted. Thus the probability that the proposal is rejected is

\[
r(x) = 1 - \int q(x, y)a(x, y)\mu(dy).
\]

If the proposal is rejected we stay at \( x \). Hence

\[
P(x, A) = r(x)I(x, A) + \int_A q(x, y)a(x, y)\mu(dy),
\]

(3.20)

Where \( I(x, A) \) is the identity kernel, which we now recognize as the Markov kernel that corresponds to “doing nothing.”

We now want to verify that the Metropolis-Hastings update is reversible with respect to \( \pi \).
CHAPTER 3. BASIC ALGORITHMS

Lemma 3.4. Suppose the transition probability kernel of a Markov chain has the following form

\[ P(x, A) = r(x)I(x, A) + \int_A p(x, y)\mu(dy), \]  

where \( p(x, \cdot) \) is a subprobability density for each \( x \) and

\[ r(x) = 1 - \int p(x, y)\mu(dy). \]

Suppose \( h(x) \) is an unnormalized density with respect to \( \mu \) and

\[ h(x)p(x, y) = h(y)p(y, x), \quad \text{for all } x \text{ and } y. \]  

Then this Markov chain is reversible with respect to the distribution \( \pi \) having unnormalized density \( h \) with respect to \( \mu \).

Proof. What is to be shown is that

\[
\iint f(x)g(y)\pi(dx)P(x, dy) \\
= \int f(x)g(x)r(x)\pi(dx) + \iint f(x)g(y)\pi(dx)p(x, y)\mu(dy).
\]

is unchanged when we interchange \( f \) and \( g \) (2.25).

The first term is obviously unchanged by interchanging \( f \) and \( g \). So we work on the second term, which multiplied by the normalizing constant for \( h(x) \) is

\[
\iint f(x)g(y)h(x)p(x, y)\mu(dx)\mu(dy) = \iint f(x)g(y)h(y)p(y, x)\mu(dx)\mu(dy) = \iint f(y)g(x)h(x)p(x, y)\mu(dy)\mu(dx)
\]

where (3.22) gives the first equality, and interchanging the dummy variables \( x \) and \( y \) gives the second. Now, except for the order of integration, the second line is just the left hand side of the first with \( f \) and \( g \) interchanged. Reversal of the order of integration is justified by the Fubini theorem.

Corollary 3.5. The Metropolis-Hastings update is reversible with respect to the distribution \( \pi \) having unnormalized density \( h \) with respect to \( \mu \).

Proof. The Metropolis-Hastings kernel (3.20) has the form (3.21) with \( p(x, y) = q(x, y)a(x, y) \). Thus we need only verify (3.22).

The probability that a proposal is accepted is

\[ a(x, y) = \min(1, R) = \min \left( 1, \frac{h(y)q(y, x)}{h(x)q(x, y)} \right). \]
Note that if \( R \leq 1 \) then
\[
a(x, y) = \frac{h(y)q(y, x)}{h(x)q(x, y)} \quad \text{and} \quad a(y, x) = 1
\]
and if \( R \geq 1 \) then
\[
a(x, y) = 1 \quad \text{and} \quad a(y, x) = \frac{h(x)q(x, y)}{h(y)q(y, x)}
\]
In either case
\[
a(x, y)h(x)q(x, y) = a(y, x)h(y)q(y, x),
\]
which is (3.22).

3.2.6 One-Variable-at-a-Time Metropolis-Hastings

When the state \( X \) is a vector \( X = (X_1, \ldots, X_d) \), the Metropolis-Hastings update can be done one variable at a time, just like the Gibbs update. The algorithm is essentially the same as before, although some changes in notation are required because the proposal only changes a single variable and hence the proposal density \( q(x, y) \) is not a density with respect to the measure \( \mu \) on the whole space. (Warning: for the rest of the section, subscripts indicate components of the state vector, not the time index of a Markov chain.)

Suppose \( \mu \) is a product measure \( \mu_1 \times \cdots \times \mu_d \). For a Metropolis-Hastings update of the \( i \)-th variable, we need a proposal density \( q_i(x, \cdot) \) with respect to \( \mu_i \). The update then works as follows. The current position is \( x \), and the update changes \( x \) to its value at the next iteration.

1. Simulate a random variate \( y \) having the density \( q_i(x, \cdot) \). Note that \( y \) has the dimension of \( x_i \) not \( x \). Let \( x_y \) denote the state with \( x_i \) replaced by \( y \)
\[
x_y = (x_1, \ldots, x_{i-1}, y, x_{i+1} \ldots x_d).
\]
2. Evaluate the Hastings ratio
\[
R = \frac{h(x_y)q_i(x_y, x_i)}{h(x)q_i(x, y)}.
\]
3. Do Metropolis rejection: with probability \( \min(1, R) \) set \( x = x_y \).

Note that, as with the original Metropolis-Hastings update, this update also stays in feasible states if started in a feasible state.

It is easy enough to go through the statements and proofs of Lemma 3.4 and Corollary 3.5 making the necessary notational changes to obtain the analogous results for one-variable-at-a-time Metropolis-Hastings. But we won’t bother, since variable-at-a-time Metropolis is a special case of the Metropolis-Hastings-Green algorithm, and we will give proofs for that.
3.2.7 Why Gibbs is a Special Case of Metropolis-Hastings

Gibbs updates a variable $x_i$ from its conditional distribution given the rest. The unnormalized joint density of all the variables is $h(x) = h(x_1, \ldots, x_d)$. We know from our slogan about conditioning and unnormalized densities that this is also an unnormalized conditional density of $x_i$ given $x_{-i}$.

A Gibbs update is a Metropolis-Hastings update in which the proposal density is $x_i \mapsto h(x_1, \ldots, x_d)$. Thus

$$q_i(x, y) = h(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_d)/c$$

where $c$ is the unknown normalizing constant that makes $h$ a proper conditional probability density. Then using the notation of the preceding section, the Hastings ratio is

$$\frac{h(x_{-i})q_i(x_{-i}, x_i)}{h(x)q_i(x, y)} = \frac{h(x_{-i})h(x)}{h(x_{-i})h(x_{-i}, y, x_{i+1}, \ldots, x_d)} = 1.$$ 

Thus this Metropolis-Hastings simulates a new value of $x_i$ from its conditional given the rest and always accepts the proposal. Hence it does exactly the same thing as a Gibbs update.

3.3 The Metropolis-Hastings-Green Algorithm

Metropolis-Hastings-Green is just like Metropolis-Hastings except that measures replace densities. Why would we want something like that? One reason is one-variable-at-a-time Metropolis-Hastings in which the whole state space is $\mathbb{R}^d$, but the proposal lies in a one-dimensional subset

$$A_{i,x} = \{ (x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_d) : y \in \mathbb{R} \}.$$ 

Since the support $A_{i,x}$ of the proposal depends on the current position $x$, the proposal distribution cannot have a density with respect to one single measure, that is, it cannot have a density $q_i(x, \cdot)$ with respect to $\mu$ we used in the general Metropolis-Hastings algorithm. That’s why we were forced to use different notation for one-variable-at-a-time Metropolis-Hastings (and would have needed a different proof of reversibility had we attempted one).

But, as we shall see, there are many other situations in which we want to make proposals in subsets of the state space that depend on the current position. In order to describe all of these using the same theory, we need a more general theory.

3.3.1 Metropolis-Hastings-Green, the Dominated Case

The Metropolis-Hastings-Green (MHG) update (Green 1995) is best described as Metropolis-Hastings with measures replacing densities.

- The unnormalized density $h$ is replaced by an unnormalized measure $\eta$. 
• The proposal density \( q(x, y) \) is replaced by a proposal kernel \( Q(x, A) \).

• The Hastings ratio (3.18) is replaced by “Green’s ratio”

\[
R(x, y) = \frac{\eta(dy)Q(y, dx)}{\eta(dx)Q(x, dy)}
\]  

(3.23)

Before we can make sense of this we have to clarify what each of these means.

By an “unnormalized measure” we mean a positive real measure. Here we want an unnormalized measure \( \eta \) that is proportional to the desired invariant distribution \( \pi \), that is, \( \eta = c\pi \) or, written out in more detail, \( \eta(B) = c\pi(B) \) for all measurable sets \( B \). Since \( \pi \) is a probability measure, \( c = \eta(S) \), where \( S \) is the state space. Allowing the measure to be unnormalized doesn’t affect the characterization of reversibility. We say the kernel \( P \) is reversible with respect to the positive measure \( \eta \) if (2.24) holds when \( \pi \) is replaced by \( \eta \). Clearly, a kernel is reversible with respect to both \( \eta \) and \( \pi \) or neither.

The proposal kernel \( Q \) needs almost no explanation. When \( x \) is the current position, \( Q(x, \cdot) \) is a probability measure used to make the proposal.

Strictly speaking (3.23) is meaningless nonsense. It is shorthand for a Radon-Nikodym derivative. We will later give precise definitions, for now we adopt the temporary definition\(^5\) that (3.23) means

\[
\iint g(x, y)R(x, y)\eta(dx)Q(x, dy) = \iint g(x, y)\eta(dy)Q(y, dx)
\]  

(3.24)

holds for every function \( g \) for which the integrals are defined, in particular for every indicator function.

There is ambiguity in defining \( R \) by (3.24), since \( R \) can be arbitrarily redefined on a set of measure zero without affecting the values of the integrals. In many interesting examples the point \( (x, y) \) will have measure zero. If we are allowed to redefine \( R \) before each use, the value \( R(x, y) \) will be arbitrary whenever we use it. That’s won’t do at all! In order to have an algorithm we need to settle on one version of \( R \), that is, one function that satisfies (3.24), and use that same function always. It doesn’t matter which version we choose, so long as we stick with our choice ever after.

Now the obvious changes of notation transform Metropolis-Hastings into the more general MHG update. The current position is \( x \), and the update changes \( x \) to its value at the next iteration.

1. Simulate a random variate \( y \) having the probability distribution \( Q(x, \cdot) \).
2. Calculate “Green’s ratio” \( R(x, y) \).
3. Do “Metropolis rejection” with probability \( \min[1, R(x, y)] \) set \( x = y \).

We see that the conditions we need are

1. For each \( x \) we can simulate a random variate with distribution \( Q(x, \cdot) \).
2. For each \( x \) and \( y \) we can evaluate \( R(x, y) \).

\(^5\)The meaning of (3.23) will later be generalized to cases in which (3.24) does not hold.
CHAPTER 3. BASIC ALGORITHMS

Green’s Algorithm

The MHG update really gains power when combined with state-dependent mixing. The algorithm proposed in Green (1995) used both ideas. 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

- $Q_i(x, S)$ is known for all $i$.

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

- For all $i \in I$

$$R_i(x, y) = \frac{\pi(dy)Q_i(y, dx)}{\pi(dx)Q_i(x, dy)}$$ (3.25)

is known$^6$ and it is possible to evaluate $R_i(x, y)$ for all $x$ and $y$.

- for each $x$ and $i$, it is possible to simulate realizations from the distribution having the normalized proposal distribution

$$P_i(x, \cdot) = \frac{Q_i(x, \cdot)}{Q_i(x, S)}$$ (3.26)

Then one step of Green’s algorithm, starting from current position $x$ goes as follows.

1. Simulate a random index $i$, choosing $i \in I$ with probability $Q_i(x, S)$. With probability $1 - \sum_{i \in I} Q_i(x, S)$, skip the remaining steps and stay at $x$.

2. Simulate $y \sim P_i(x, \cdot)$ defined by (3.26).

3. Calculate Green’s ratio $R_i(x, y)$.

4. Accept $y$ with probability $\min[1, R_i(x, y)]$.

All of this is just the MHG update described in preceding section combined with the idea of state-dependent mixing (Section 3.1.8).

3.3.2 Spatial Point Processes

Poisson Processes

A spatial point process is a random process having values that are point patterns in a region of $\mathbb{R}^d$. Both the number of points and their positions within the region are random. A point process is simple if the locations of points never

$^6$We take the Radon-Nikodym derivative here to have the same meaning here as in the preceding section, i.e., (3.24) holds with $Q$ and $R$ replaced by $Q_i$ and $R_i$. Also we must fix one version of $R_i$ to be used throughout. As promised for the simple MHG update, we will later generalize to cases in which (3.24) does not hold.
coincide, that is, with probability one the location of every point is different. A point process is called **finite** if the number of points is finite with probability one. We will only be interested in finite simple point processes.

The process illustrated in Figure 3.6 is the simplest of all spatial point processes, the **homogeneous Poisson process**, which is simulated as follows.

- Simulate a Poisson random variate $N$.
- Simulate $N$ i. i. d. points uniformly distributed in the region.

For the patterns in Figure 3.6, the expected number of points was 8.75 (the actual numbers are 8, 11, and 6). Any nonnegative number of points is possible, including zero (the empty pattern) though this may be very rare (probability $1.6 \times 10^{-4}$ in this example). The notch in the side of the region is only to avoid being square. The region can be any shape.

For any point process on a region $A$ and any measurable subset $B$ of $A$, let $N_B$ denote the number of points in $B$. This is a random variable, because it is a function of the random point pattern. Define $\lambda(B) = E(N_B)$. Then $\lambda$ is a positive measure on $A$, called the parameter measure of the process. When the process is simple, the only case of interest to us, $\lambda$ is also called the intensity measure of the process.

Any finite, nonatomic measure $\lambda$ on a region $A$ determines an inhomogeneous Poisson process with intensity measure $\lambda$, which is simulated as follows.

- Simulate a Poisson random variate $N$ with expectation $\lambda(A)$.
- Simulate $N$ i. i. d. points with distribution $\nu$ defined by

$$\nu(B) = \frac{\lambda(B)}{\lambda(A)} \tag{3.27}$$

It is a remarkable fact about the Poisson process that it has two characterizations that have no obvious connection with each other.

**Theorem 3.6.** In order that a simple, finite point process be Poisson, it is necessary and sufficient that there be a finite nonatomic measure $\lambda$ such that $E(N_B) = \lambda(B)$ for each measurable set $B$.

---

7 A measure is nonatomic if every one-point set has measure zero. A positive measure $\lambda$ is finite if $\lambda(A) < \infty$. 

---

Figure 3.6: Three realizations of the same spatial point process.
CHAPTER 3. BASIC ALGORITHMS

This combines Theorems 2.4.II and 2.4.III in Daley and Vere-Jones (1988).

**Theorem 3.7.** In order that a simple, finite point process be Poisson, it is necessary and sufficient that for any measurable partition $B_1, B_2, \ldots, B_k$ of the domain, the random variables $N_{B_1}, N_{B_2}, \ldots, N_{B_k}$ are independent.

This is Theorem 2.4.VII in Daley and Vere-Jones (1988). That the simulation method described above satisfies the characterizations in the theorems is left as an exercise (Exercise 3.5).

**Non-Poisson Processes**

So far we have gotten away with not precisely specifying the probability measure for the Poisson process, or even the sample space. This turns out to be slightly tricky, the issue being whether we consider the points of the pattern to be ordered or not. Notationally, the easiest to work with is to consider ordered patterns of points. Then conditional on $N_A = n$, the $n$ points of the pattern are an element of $A^n$. This is not the Right Thing because we really want to consider the points as unordered, in which case the ordered view overcounts by distinguishing the $n!$ permutations of $n$ points. However, the Wrong Thing can be made to work as long as we choose probability models that are symmetric under permutations of the points in a pattern. Then both views will produce the same answers to all questions that do not explicitly mention the ordering. For more on this issue, see Daley and Vere-Jones (1988, Section 5.3).

In the “ordered view,” the state space of a finite simple point process in a region $A$ can be taken to be

$$S = \bigcup_{n=0}^{\infty} A^n.$$  

When there are $n$ points, the state is a vector of a points in $A$, hence an element of $A^n$. $A^0$ is the singleton set $\{\emptyset\}$. This agrees with the definition of $A^0$ in abstract set theory, where 0 is defined to be the empty set, so $A^0 = A^\emptyset$, which is the set of all functions from the empty set to $A$ and there is one such function, the empty function. This notation is felicitous, the empty set being an appropriate notation to represent the empty point pattern having zero points. If $A$ is the $\sigma$-field for $A$, then the product $\sigma$-field for $A^k$ is denoted $\mathcal{A}^k$, and the natural $\sigma$-field for $S$, call it $\mathcal{B}$, is the family of sets $B \subset S$ such that $B \cap A^k$ is an element of $\mathcal{A}^k$.

Now we can write down the probability measure of the Poisson process with intensity measure $\lambda$. It is a measure $P$ on $(S, \mathcal{B})$ defined by

$$P(B) = \sum_{n=0}^{\infty} \frac{\lambda^n (B \cap A^n)}{n!} e^{-\lambda(A)}, \quad B \in \mathcal{B}.$$  

We see that this is the right formula because

$$\Pr(N_A = n) = P(A^n) = \frac{\lambda^n (A^n)}{n!} e^{-\lambda(A)} = \frac{\lambda(A)^n}{n!} e^{-\lambda(A)}.$$
which is the right formula for $N_A$ to be Poisson with mean $\lambda(A)$, and

$$\Pr(X \in B|N_A = n) = \frac{P(B \cap A^n)}{P(A^n)} = \frac{\lambda^n(B \cap A^n)}{\lambda^n(A^n)}$$

is just $\lambda^n$ renormalized to be a probability measure, which is also the right thing (the $n$ points are i. i. d. because $\lambda^n$ is product measure). It saves a little bit of ink in formulas if we also define the unnormalized measure $\mu$ for the Poisson process that throws away the constant $e^{-\lambda(A)}$, giving

$$\mu(B) = \sum_{n=0}^{\infty} \frac{\lambda^n(B \cap A^n)}{n!}, \quad B \in \mathcal{B}. \quad (3.28)$$

We now want to consider families of probability distributions for point processes defined by families of unnormalized densities $\{h_\theta : \theta \in \Theta\}$ with respect to $\mu$. The Poisson process is symmetric under permutation of the points in the point patterns. We want the same property for our new models. Write $x \equiv y$ if $x, y \in S$ are patterns having the same number of points and the same locations of the points only a different ordering. Then we need to require that our unnormalized densities satisfy the symmetry requirement

$$h_\theta(x) = h_\theta(y), \quad \text{whenever } x \equiv y. \quad (3.29)$$

Recall that $h_\theta$ on $S$ is an unnormalized density if it is nonnegative, not almost everywhere zero, and integrable. The first two are easy to check. The last is not trivial. The normalizing function for the family is given by

$$c(\theta) = \int h_\theta(x)\mu(dx) = \sum_{n=0}^{\infty} \frac{1}{n!} \int_{A^n} h_\theta(x)\lambda^n(dx)$$

if the integral is finite (that’s what we have to check). The normalized density $f_\theta$ corresponding to $h_\theta$ is given, as usual, by (3.17), from which we see that the probability of a measurable set $B$ in $S$ is

$$\frac{1}{c(\theta)} \int_B h_\theta(x)\mu(dx) = \frac{1}{c(\theta)} \sum_{n=0}^{\infty} \frac{1}{n!} \int_{B \cap A^n} h_\theta(x)\lambda^n(dx) \quad (3.30)$$

It turns out that for a variety of reasons we will only be interested in processes that satisfy the following stability condition

**Condition 3.8.** A process with unnormalized density $h$ with respect to $\mu$ is stable if there exists a real number $M$ such that

$$h(x \cup \xi) \leq M h(x), \quad \text{for all } x \in S \text{ and } \xi \in S. \quad (3.31)$$

This condition will have other uses later on. For now, it implies that the normalizing function is finite on $\Theta$. First we see that if $x$ has $n$ points, then by using (3.31) $n$ times, we obtain

$$h_\theta(x) \leq M^n h(\emptyset)$$
and this implies
\[ c(\theta) \leq h(\emptyset) \sum_{n=0}^{\infty} \frac{M^n}{n!} \int_{A^n} \lambda^n(dx) = h(\emptyset) \sum_{n=0}^{\infty} \frac{M^n \lambda(A)^n}{n!} = h(\emptyset) e^{M \lambda(A)} \]

which is finite. For more on this subject see the chapter by Geyer and the chapter by Baddeley in (Kendall, Barndorff-Nielsen, and van Lieshout 1998).

Simulating Spatial Point Processes

This section actually discusses a “prequel” of the Metropolis-Hastings-Green algorithm, a method for simulating spatial point processes due to Geyer and Møller (1994) that, although a special case of Metropolis-Hastings-Green, was invented prior to it. This is typical of the way theories develop, special cases first, general theories later.

It is a truism that textbooks and research papers make for bad history, bad psychology, and bad sociology of science. Textbooks and papers never tell it like it was and hence are useless for learning about science was done or should be done. Authors start with a half-baked idea, often a wrong idea. They work it over, modify it to make proofs easier (or possible!) or interpretations simpler. Sometimes they make the treatment more abstract and mathematically sophisticated. By the time an article appears in print, there may be no trace of the train of thought that lead the authors to their discovery. Result: you can’t learn about how to do science by reading science (or math). Textbooks are worse. The start with the distortions of the original authors and add more of their own. One of the best services the author of a textbook can perform is to really clean up a subject, eliminating all the blind alleys and presenting a clear path through the material. But that really distorts the history. It requires presenting material out of historical sequence and selecting material to present on the basis of importance to the textbook author’s take on the subject rather than historical importance. This book is no different, but for once, I’ll present a subject as it really developed.

One way to think of the state of a point process is as a random integer \( N \) and a random \( N \)-vector \( X = (X_1, \ldots, X_N) \). Before Green (1995) there was no general method for simulating such a thing, no way to “jump dimensions”. But if we could put every state on a space of the same dimension, we could use ordinary Metropolis-Hastings. No finite dimensional space will do, so let’s pad out the space to \( \mathbb{R}^\infty \). Now the state of the point process is a random nonnegative integer \( N \) and a random sequence \( X = (X_1, X_2, \ldots) \in \mathbb{R}^\infty \). The observable state of the point process is \( (X_1, \ldots, X_N) \). The rest of the variables are junk added to help us apply the Metropolis algorithm. They can be defined any way we like. A simple definition that turns out to be useful is to define them to be i. i. d. on the region containing the process.

Starting with a model having unnormalized density \( h_\theta \) with respect to the measure \( \mu \) defined by (3.28), which is proportional to the probability measure for a Poisson process with intensity measure \( \lambda \), we want to define a new model...
as one having unnormalized density \( \tilde{h}_\theta(x, n) \) with respect to some measure \( \tilde{\mu} \) on \( \mathbb{R}^\infty \times \mathbb{N} \). We take \( \tilde{\mu} \) to be the measure on \( \mathbb{R}^\infty \times \mathbb{N} \) that is \( \nu^\infty \) times counting measure on \( \mathbb{N} \), where \( \nu \) is the measure defined by (3.27), that is, \( \lambda \) normalized to be a probability measure. Then we define \( \tilde{h}_\theta \) by

\[
\tilde{h}_\theta(x, n) = \frac{h_\theta((x_1, \ldots, x_n)) \lambda(A)^n}{n!} \quad (3.32)
\]

Since (3.32) does not involve \( x_{n+1}, x_{n+2}, \ldots \), it says that conditional on \( N = n \) the variable \( X_{n+1} \) is independent of all other \( X_k \) and has the distribution \( \nu \), which was one property we wanted. It is also clear that for any measurable set \( B \) in \( A^n \) that

\[
\Pr((x_1, \ldots, x_n) \in B \& N = n) = \frac{1}{n!} \int_B h_\theta(x) \lambda^n(dx)
\]

Comparing with (3.30) we see that this model does capture the same probability structure as the other.

Now consider a Metropolis-Hastings update of \( N \). The simplest is to propose to increase \( N \) by one with probability \( \frac{1}{2} \) and decrease it by one with probability \( \frac{1}{2} \) (unless \( N = 0 \) already, in which case increase \( N \) by one with probability \( \frac{1}{2} \) and do nothing with probability \( \frac{1}{2} \)). This is a Metropolis proposal: between each two numbers \( n \) and \( n + 1 \) there is the same probability of a proposal going up and a proposal going down (i.e., \( \frac{1}{2} \)). The odds ratio for a move from \( n \) to \( n + 1 \) is

\[
R = \frac{h_\theta((x_1, \ldots, x_{n+1}))}{h_\theta((x_1, \ldots, x_n))} \frac{\lambda(A)}{n+1} \quad (3.33)
\]

and the odds ratio for a move the other way, from \( n + 1 \) to \( n \) is the reciprocal of (3.33), but we usually think of a move from \( n \) to \( n - 1 \) (the current position being \( n \)). That gives

\[
R = \frac{h_\theta((x_1, \ldots, x_{n-1}))}{h_\theta((x_1, \ldots, x_n))} \frac{n}{\lambda(A)} \quad (3.34)
\]

One problem with this description of the algorithm is that it seems to require an infinite state. We can’t allow that! But since the infinite tail past \( N \) is independent of the part of the state we are interested in, we can ignore it and simulate as needed. When we move from \( n \) to \( n + 1 \) we get a new \( X_{n+1} \), but it is independent of the other \( X_i \) and has distribution \( \nu \). We can simulated it when needed in the proposal part of the update.

One update step, starting from current position \( (x_1, \ldots, x_n) \) goes as follows.

1. Flip a coin. On heads try to move from \( n \) to \( n + 1 \). On tails, try to move from \( n \) to \( n - 1 \), unless \( n = 0 \), in which case skip the remaining steps (doing nothing).

2. If going up simulate \( x_{n+1} \) independent of the current state and having distribution \( \nu \) given by (3.27).
3. Evaluate the odds ratio, (3.33) if going up or (3.34) if going down.

4. Accept the move with probability \( \min(1, R) \).

There’s no question algorithm has the correct invariant distribution. It’s just Metropolis. There’s nothing fancy about it except for the somewhat mysterious and ghostly infinite sequence of random variables that are only used in wooing about the algorithm, playing no role in the simulation. It seems likely that most examples of the Metropolis-Hastings-Green algorithm could be treated something like this, thereby eliminating any need to know what Radon-Nikodym derivatives are, but then the algorithm would lose its generality and every doable example would require a special story with its own special ghostly variables. Better to suffer the measure theory.

So let’s translate our algorithm into Metropolis-Hastings-Green terminology. We know what the proposal is, going down, we will delete \( x_n \), and going up we will add a new \( x_{n+1} \), which will have distribution \( \nu \) given by (3.27). The way Green’s algorithm works is that one kernel, call it \( Q_n \) describes both a move and its “reverse move”. If \( Q_n \) describes a move up from \( A^n \) to \( A^{n+1} \), it should also describe the reverse move down from \( A^{n+1} \) to \( A^n \). To keep things simple, we should leave it at that. Then there will be a different \( Q_n \) for every \( n \geq 0 \).

The next task is to figure out what \( \pi(dx)Q_n(x, dy) \) is in each case, going up or down. Going up the current state \( x \) can be any element of \( A^n \), but the proposal \( y \) must agree with \( x \) in the first \( n \) coordinates, so the pair \( (x, y) \) is concentrated on the set

\[
D_n = \{ (x, y) \in S^2 : x \in A^n, y \in A^{n+1}, x_i = y_i, i = 1, \ldots, n \}.
\]

The unnormalized joint distribution of \( (x, y) \) is

\[
\eta(dx)Q_n(x, dy) = h_\theta(x)\mu(dx)I(x, A^n) \frac{\lambda(dy_{n+1})}{\lambda(A)}
\]

\[= h_\theta(x) \frac{\lambda^n(dx) \lambda(dy_{n+1})}{n! \lambda(A)}
\]

\[= h_\theta(x) \frac{\lambda^{n+1}(dy)}{n! \lambda(A)}. \tag{3.35}\]

Going down the current state \( x \) can be any element of \( A^{n+1} \) and the proposal \( y \) is deterministic, being the element of \( A^n \) that agrees with \( x \) in the first \( n \) coordinates, so the pair \( (x, y) \) is concentrated on the set \( \varphi(D_n) \) where \( \varphi \) is the function that swaps coordinates in \( S^2 \), that is, \( \varphi : (x, y) \mapsto (y, x) \). The unnormalized joint distribution of \( (x, y) \) is

\[
\eta(dx)Q_n(x, dy) = h_\theta(x)\mu(dx)I(x, A^{n+1}) \]

\[= h_\theta(x) \frac{\lambda^{n+1}(dx)}{(n+1)!}. \tag{3.36}\]
Thus going up Green’s ratio is (3.36) with \(x\) and \(y\) interchanged divided by (3.35)
\[
R(x, y) = \frac{h_\theta(y)}{h_\theta(x)} \cdot \frac{\lambda(A)}{n + 1}
\]
which is just the expression we had before, (3.33) in slightly different notation. Similarly going down Green’s ratio is (3.35) with \(x\) and \(y\) interchanged divided by (3.36)
\[
R(x, y) = \frac{h_\theta(y)}{h_\theta(x)} \cdot \frac{n + 1}{\lambda(A)}
\]
which agrees with (3.34) when we recall that in (3.34) we had changed \(n\) to \(n - 1\).

In calculating Green’s ratio we just “cancelled” the \(\lambda^{n+1}(dy)\) terms in the numerator and denominator. To be very careful, we should have checked that (3.24) holds, but it obviously does.

A minor blemish on this algorithm is the way it treats the points in the pattern asymmetrically. Recall that we really consider the points unordered. We insist that the model have a symmetric density, so that the probability of a pattern does not depend on the ordering of the points. But the MHG algorithm described above doesn’t treat the points symmetrically. It always adds or deletes the last point in the ordering. We can cure this blemish by composing our MHG update with another basic update, which simply reorders the \(n\) points of the pattern, choosing among the \(n!\) orders with equal probability. This clearly preserves the distribution with unnormalized density \(h_\theta\) because we have required \(h_\theta\) to be symmetric. We do not even have to actually permute the points. The only effect this random permutation has on the MHG updates is that in steps down a random point rather than the \(n\)-th is deleted. This gives us an algorithm that reflects the symmetry of the model.

As usual, we describe one basic update step starting at a pattern \(x\) with \(n\) points
1. Flip a coin. On heads try to add a point. On tails, try to delete one (or if \(n = 0\) so there are no points to delete, do nothing, skip the remaining steps).
2. If going up simulate \(\xi\) independent of the current state and having distribution \(\nu\) given by (3.27).
3. Evaluate the odds ratio, (3.33) if going up or (3.34) if going down.
4. Accept the move with probability \(\min(1, R)\).

### 3.3.3 Bayesian Model Selection

The Bayesian competitor to frequentist model selection procedures (like all-subsets regression) involves computing Bayes factors for the various models under consideration. For a concrete example, consider again Bayesian logistic
regression (Example 3.3). In that model there were three predictors. There are \(2^3 = 8\) different models that can be formed by including or excluding any of these predictors. One, the \textit{full model}, which has all three predictors and four regression coefficients including the intercept, is the one we already analyzed in Example 3.3. Another, the \textit{null model} has no predictors and just one regression coefficient, the intercept, and just fits a Bernoulli model to the data (i.e., the data \(Y_i\) are i.i.d. Bernoulli(\(p\)) with \(p\) the single unknown parameter). Between these are three models with one predictor and another three with two predictors. The \textit{model selection problem} is to select the single model that best fits the observed data. The \textit{model comparison problem} is a bit more vague. It only asks for comparison of the models, leaving a decision to the user. The Bayesian solution to either involves Bayes factors.

The parameter spaces for different submodels typically have different dimensions. For our logistic regression example, the parameter spaces have dimensions between one (for the null model) and four (for the full model). The parameter spaces for the models have the form \(\mathbb{R}^I\), where \(I\) is a subset of \(\{0, 1, 2, 3\}\) that contains 0, and are shown in the diagram below. The parameter spaces of the logistic regression model selection problem are partially ordered by embedding, the arrows in the diagram denoting the natural embeddings, which set certain coordinates to zero, for example, the arrow going from \(\mathbb{R}^\{0,1,2,3\}\) to \(\mathbb{R}^\{0,2,3\}\) represents the embedding \((\beta_0, 0, \beta_2) \mapsto (\beta_0, \beta_2)\).

\footnote{Recall that \(\mathbb{R}^S\) means the set of all functions from \(S\) to \(\mathbb{R}\), hence an element \(\beta \in \mathbb{R}^{\{0,1,2,3\}}\) is a function from \(\{0, 1, 2, 3\}\) to \(\mathbb{R}\), which can be specified by giving its values \(\beta(0)\), \(\beta(1)\) and \(\beta(3)\) at the points of the domain. If we write \(\beta_i\) instead of \(\beta(i)\) we get the more familiar notation for vectors. An element \(\beta \in \mathbb{R}^{\{0,1,2,3\}}\) represents a 3-vector \((\beta_0, \beta_1, \beta_2)\). Notice the value of the notation. The parameter spaces \(\mathbb{R}^{\{0,1,2,3\}}\) and \(\mathbb{R}^{\{0,2,3\}}\) are different. They index different models. If we denoted both of them by \(\mathbb{R}^3\), we would not be able to distinguish them.}
We now need an abstract framework that describes any model selection problem. Let \( \mathcal{M} \) be an index set for the models. Corresponding to a model \( M \in \mathcal{M} \), there is a parameter space \( \Theta_M \). In the logistic regression problem the \( \Theta_M \) are the spaces \( \mathbb{R}^I \) in the diagram. Assume the \( \Theta_M \) are disjoint. Then the parameter space for the entire problem is the union\(^9\)

\[
\Theta = \bigcup_{M \in \mathcal{M}} \Theta_M.
\]

For each \( \theta \in \Theta \) there is a data model \( f(x|\theta) \), and there is also a prior, which is a probability measure \( \gamma \) on \( \Theta \). In model comparison, proper priors are de rigueur. See Bernardo and Smith (1994, pp. 421–424) for the reasons why, and read all of Chapter 6 in Bernardo and Smith (1994) if you really want to understand Bayesian model comparison.

The object of Bayesian analysis is, as always, to calculate the posterior. In the model comparison problem, we are not interested in the posterior distribution of the parameter values \( \theta \), but only in the posterior probabilities of the models

\[
p(M|x) = \frac{\int_{\Theta_M} f(x|\theta)\gamma(d\theta)}{\int_{\Theta} f(x|\theta)\gamma(d\theta)}
\]

We do not need the denominator, since we are only interested in the relative probabilities of the models

\[
p(M|x) \propto \int_{\Theta_M} f(x|\theta)\gamma(d\theta)
\]

\(^9\)If the \( \Theta_M \) were not disjoint, then we would have to use the notion of disjoint union (Jänich 1984, p. 10), which treats the sets as if they were disjoint.
CHAPTER 3. BASIC ALGORITHMS

and not even in them, exactly. The prior $\gamma$ can be divided into two parts: the marginal for the models $\gamma(\Theta_M)$ and the conditional distribution for $\theta$ given $M$

$$\gamma(A|M) = \frac{\gamma(A \cap \Theta_M)}{\gamma(\Theta_M)}$$

If you and I agree about the conditional of $\theta$ given $M$, but disagree about the marginals, then our posterior probabilities will be proportional to our prior probabilities

$$p(M|x) \propto \gamma(\Theta_M) \int_{\Theta_M} f(x|\theta) \gamma(d\theta|M)$$

One way to take out part of the subjectivity involved in this inference is to divide by the prior odds $\gamma(\Theta_M)$. This gives the Bayes factor, which is the ratio of posterior to prior odds

$$B(M) = \frac{p(M|x)}{\gamma(\Theta_M)} \propto \int_{\Theta_M} f(x|\theta) \gamma(d\theta|M).$$

The integral defines the Bayes factors up to an overall constant of proportionality. Call it the unnormalized Bayes factors

$$B_u(M) = \int_{\Theta_M} f(x|\theta) \gamma(d\theta|M).$$

To use the Bayes factors to compare models, you multiply $B_u(M)$ by your (or your client’s) personal prior probabilities $\gamma(\Theta_M)$ to obtain your own posterior model probabilities $p(M|x)$ up to a constant of proportionality. The constant usually does not matter. For example, the solution to the model selection problem is to select the model with the highest $p(M|x)$ and this is the same as the model with the highest $\gamma(\Theta_M)B_u(M)$ because multiplying by a constant does not change which model is highest. If you need actual probabilities, simply normalize the unnormalized Bayes factors by dividing by their sum

$$p(M|x) = \frac{\gamma(\Theta_M)B_u(M)}{\sum_{M \in \mathcal{M}} \gamma(\Theta_M)B_u(M)}$$

To return to our logistic regression model, the data model is the same as before (Example 3.3). The only difference is that for the submodels we set some of the regression coefficients $\beta_i$ to zero. So far we haven’t specified the set $\mathcal{M}$ except to say that it indexes the models. To be specific now, let $\mathcal{M}$ be the set of exponents in the diagram, the subsets of $\{0, 1, 2, 3\}$ that contain 0. Then $\Theta_M = \mathbb{R}^M$. The prior must be a probability measure on $\Theta = \bigcup_{M \in \mathcal{M}} \Theta_M$. Only measure theory gives us a simple notation for something like that. We might, for example, choose a normal distribution for the restriction of $\gamma$ to the parameter space $\mathbb{R}^{\{0, 1, 2, 3\}}$ of the full model and obtain all the restrictions of $\gamma$ to the parameter spaces of the submodels by conditioning the normal distribution for the full model to lie in the the parameter spaces of the submodels.\(^\text{10}\)

\(^{10}\)To be continued. The code for an MHG sampler for this model is yet to be written.
3.3.4 Metropolis-Hastings-Green, the General Case

The description of the MHG update given in the preceding section is usable for many problems, but in some respects it is a step backward. It doesn’t include some ordinary Metropolis updates, such as the one for the dumbbell distribution.

Radon-Nikodym Derivatives and Lebesgue Decomposition

This section briefly sketches three important measure-theoretic notions: absolute continuity, Lebesgue decomposition, and Radon-Nikodym derivatives.

If \( \mu \) and \( \nu \) are two positive measures on the same measurable space \((S, \mathcal{B})\), we say \( \mu \) is absolutely continuous with respect to \( \mu \) if \( \nu(B) = 0 \) implies \( \mu(B) = 0 \). An alternative terminology is that \( \nu \) dominates \( \mu \). A notation indicating this condition is \( \mu \ll \nu \).

If \( \mu \ll \nu \) and \( \nu \ll \mu \), we say that \( \mu \) is equivalent to \( \nu \) and write \( \mu \sim \nu \). Note that this says only that \( \mu \) and \( \nu \) have the same null sets. It is easy to see that this is an equivalence relation on the class of all positive real measures.

A function \( f \) on \( S \) is said to be a density of \( \mu \) with respect to \( \nu \) if
\[
\mu(B) = \int_B f(x)\nu(dx), \quad B \in \mathcal{B},
\] (3.37)
which implies
\[
\int g(x)\mu(dx) = \int g(x)f(x)\nu(dx)
\]
for any integrable function \( g \). This is a generalization of the usual notion of a probability density function. When \( \nu \) is Lebesgue measure \( dx \) and \( \mu \) is a probability measure, \( f \) is just the familiar p. d. f. of \( \mu \).

The Radon-Nikodym theorem (Rudin 1987, Theorem 6.10) says that \( \mu \ll \nu \) implies that \( \mu \) has a density with respect to \( \mu \). The converse assertion is also true: if (3.37) holds, then \( \mu \ll \nu \).

The Radon-Nikodym theorem implies the existence of a density, but is it unique? Since integrals over sets of measure zero are zero, a density can be redefined arbitrarily on a set of measure zero and still be a density. But an elementary theorem of measure theory (Rudin 1987, Theorem 1.39(b)) says that is the only arbitrariness allowed: two densities of \( \mu \) with respect to \( \nu \) must be equal except on a set of \( \nu \) measure zero. Another way to say this is that if \( f \) is a density of \( \mu \) with respect to \( \nu \), then \( f \) is unique considered as an element of \( L^1(\nu) \).

Because a density \( f \) of \( \mu \) with respect to \( \nu \) is unique (in the \( L^1 \) sense), it makes sense to give it a name and notation as something determined by \( \mu \) and \( \nu \). When (3.37) holds, we say that \( f \) is the Radon-Nikodym derivative of \( \mu \) with respect to \( \nu \) and write
\[
f = \frac{d\mu}{d\nu}
\]
This is just another terminology for (3.37). We are not defining a new operation.
So now we see where (3.24) comes from. If the measure in the numerator of (3.23) is absolutely continuous with respect to the measure in the denominator, then the condition that \( R(x,y) \) be a density of one with respect to the other is (3.24). We now want to generalize to the situation when absolute continuity is not present.

Measures \( \mu \) and \( \nu \) are mutually singular if there exists a measurable set \( B \) such that \( \mu(B) = 0 \) and \( \nu(B^c) = 0 \) (hence \( \mu \) is concentrated on \( B^c \) and \( \nu \) is concentrated on \( B \)). A notation indicating this condition is \( \mu \perp \nu \). In a sense mutual singularity is the opposite of absolute continuity.

The Lebesgue decomposition theorem (Rudin 1987, Theorem 6.10) says that if \( \mu \) and \( \nu \) are arbitrary positive real measures on the same state space, then \( \mu \) can be decomposed as the sum \( \mu = \mu_a + \mu_s \), where \( \mu_a \ll \nu \) and \( \mu_s \perp \nu \). The pair \((\mu_a, \mu_s)\) is called the Lebesgue decomposition of \( \mu \) relative to \( \nu \).

Now we can give the most general notion of a Radon-Nikodym derivative. If \( \mu \) and \( \nu \) are arbitrary positive real measures on the same state space, and \( \mu = \mu_a + \mu_s \) is the Lebesgue decomposition of \( \mu \) relative to \( \nu \), then we often say that \( f = d\mu_a/d\nu \) is the Radon-Nikodym derivative of \( \mu \) with respect to \( \nu \). Of course, \( f \) is now the density of \( \mu_a \) (not \( \mu \)) with respect to \( \nu \), but that is the best we can do. The mutually singular part \( \mu_s \) has no relation to \( \nu \) whatsoever.

With these preliminaries out of the way, let us return to considering what (3.23) means. We said it was a Radon-Nikodym derivative, but of what measures? It is obvious that the intention is that \( \eta(dx)Q(x,dy) \) indicate the unnormalized joint distribution of the current state \( x \) and the proposal \( y \). To be mathematically precise we must define this as a measure \( \mu \) on \((S^2, B^2)\) by

\[
\mu(B) = \iint 1_B(x,y)\eta(dx)Q(x,dy), \quad B \in B^2.
\]

The numerator in (3.23) is the denominator with \( x \) and \( y \) reversed, but \( \mu \) is a function of one argument (the set \( B \)) rather than two, so we can’t obtain the measure in the numerator by swapping arguments. Instead we have to proceed a bit differently, first defining the function \( \varphi : (x,y) \mapsto (y,x) \) that switches coordinates in \( S^2 \). Then the measure in the numerator is \( \mu \circ \varphi \), defined by

\[
(\mu \circ \varphi)(B) = \mu[\varphi(B)].
\]

So we finally have a rigorous general definition of Green’s ratio

\[
R = \frac{d(\mu \circ \varphi)}{d\mu}(\mu \circ \varphi) \quad (3.40)
\]

where \( \mu \) is defined by (3.38).

The following lemmas give some useful properties of Radon-Nikodym derivatives that are helpful in calculations.

**Lemma 3.9 (Chain Rule).** If \( \lambda \ll \mu \ll \nu \), then

\[
\frac{d\lambda}{d\nu} = \frac{d\lambda}{d\mu} \cdot \frac{d\mu}{d\nu} \quad (3.41)
\]

holds \( \nu \) almost everywhere.
Corollary 3.10 (Reciprocal Rule). If \( \mu \sim \nu \), then
\[
\frac{d\mu}{d\nu} = \left( \frac{d\nu}{d\mu} \right)^{-1}
\] (3.41)
holds \( \mu \) almost everywhere.

Remark. “\( \mu \) almost everywhere” here is the same as “\( \nu \) almost everywhere” because \( \mu \) and \( \nu \) have the same null sets. The set on which the right hand side is undefined because \( \frac{d\nu}{d\mu} = 0 \) is a set of \( \nu \) measure zero. Hence we may define the right hand side arbitrarily for such points so long as we produce a measurable function (for example, we could set it equal to an arbitrary constant).

Lemma 3.9 and Corollary 3.10 are Problems 32 and 33 of Chapter 8 in Fristedt and Gray (1997).

Corollary 3.11 (Ratio Rule). If \( \mu \ll \xi \) and \( \nu \ll \xi \), then
\[
\frac{d\mu}{d\nu} = \frac{d\mu}{d\xi} \frac{d\xi}{d\nu}
\] (3.42)
holds \( \nu \) almost everywhere.

Remark. The set on which the right hand side is undefined because \( \frac{d\nu}{d\xi} = 0 \) is a set of \( \nu \) measure zero. Hence we may define the right hand side arbitrarily for such points so long as we produce a measurable function.

Proof. Let \( (\mu_a, \mu_s) \) be the Lebesgue decomposition of \( \mu \) with respect to \( \nu \). Then \( \mu_a \ll \nu \ll \xi \), so by the chain rule
\[
\frac{d\mu_a}{d\xi} = \frac{d\mu_a}{d\nu} \frac{d\nu}{d\xi}.
\]
Also
\[
\frac{d\mu_s}{d\xi} \frac{d\nu}{d\xi} = 0, \quad \xi \text{ almost everywhere},
\]
because otherwise we would have \( \mu_s(B) > 0 \) and \( \nu(B) > 0 \) for some set \( B \), which contradicts \( \mu_s \perp \xi \). By the remark, we need only prove (3.42) when \( \frac{d\nu}{d\xi} > 0 \), which implies \( \frac{d\mu_s}{d\xi} = 0 \) and
\[
\frac{d\mu}{d\nu} = \frac{d\mu_a}{d\nu} = \frac{d\mu_a}{d\xi} \frac{d\xi}{d\nu} = \frac{d\mu}{d\xi} \frac{d\xi}{d\nu}
\]
and we are done. \( \square \)

Lemma 3.12. If \( (\mu_a, \mu_s) \) is the Lebesgue decomposition of \( \mu \) relative to \( \nu \) and \( (\nu_a, \nu_s) \) is the Lebesgue decomposition of \( \nu \) relative to \( \mu \), then \( \mu_a \perp \nu_s \), \( \mu_a \sim \nu_a \), and
\[
\frac{d\mu}{d\nu} = \frac{d\mu_a}{d\nu_a}
\]
Proof. First, \( \mu \perp \nu \) implies \( \mu_a \perp \nu_s \). Together with \( \mu_a \ll \nu \), this implies \( \mu_a \ll \nu_a \). Suppose \( D \) is a set such that \( \nu_s(D) = 0 \) and \( \nu_a(D^c) = 0 \), the existence of such a set being guaranteed by the Lebesgue decomposition theorem. Then if \( f = d\mu/d\nu = d\mu_a/d\nu \)

\[
\mu_a(B) = \int_{B \cap D} f(x)\nu_a(dx) + \int_{B \cap D^c} f(x)\nu_s(dx) \tag{3.43}
\]

Taking \( B = D^c \), we get \( \mu_a(D^c) \leq \nu_a(D^c) = 0 \), so we must have \( f(x) = 0 \), \( x \in D^c \). Thus the second term on the right hand side of \( (3.43) \) is always zero and \( f \) is also a density of \( \mu_a \) with respect to \( \nu_a \).

Lemma 3.13. If \( \varphi \) is a function on the domain of \( \mu \) satisfying \( \varphi = \varphi^{-1} \). If \( \nu = \mu \circ \varphi \), and if \( \mu_a, \mu_s, \nu_a, \nu_s \) are as in Lemma 3.12, then

\[
\nu_a = \mu_a \circ \varphi \quad \text{and} \quad \nu_s = \mu_s \circ \varphi.
\]

Proof. First we note that

\[
\mu_a \circ \varphi + \mu_s \circ \varphi = (\mu_a + \mu_s) \circ \varphi = \nu
\]

is a decomposition of \( \nu \), so what we need to show is

\[
\mu_a \circ \varphi \ll \mu \tag{3.44a}
\]

\[
\mu_s \circ \varphi \perp \mu \tag{3.44b}
\]

What we are given to work with is

\[
\mu_a \ll \mu \circ \varphi \tag{3.44c}
\]

\[
\mu_s \perp \mu \circ \varphi \tag{3.44d}
\]

(3.44a) is shown by

\[
\mu(B) = 0 \iff (\mu \circ \varphi)(\varphi[B]) = 0 \iff \mu_a(\varphi[B]) = 0 \iff (\mu_a \circ \varphi)(B) = 0,
\]

the middle implication being (3.44c) and the other implications being \( \varphi = \varphi^{-1} \) and the definition of functional composition.

Now (3.44d) implies the existence of a set \( B \) such that \( \mu_s(B) = (\mu \circ \varphi)(B^c) = 0 \). Hence

\[
(\mu_s \circ \varphi)(\varphi[B]) = \mu_s(B) = 0
\]

and

\[
\mu(\varphi[B^c]) = \mu(\varphi[B^c]) = (\mu \circ \varphi)(B^c) = 0
\]

and this proves (3.44b).

Corollary 3.14. Suppose \( \mu, \mu_a \) and \( \varphi \) are as in the lemma, and \( \xi \) satisfies \( \xi \circ \varphi = \xi \) and \( \mu \ll \xi \). Then

\[
d(\mu \circ \varphi) = \frac{d\mu}{d\xi} \circ \varphi = \frac{d\mu_a}{d\xi} = \frac{d\mu_s}{d\xi}
\]

What does all this tell about MHG calculations? Taking (3.40) as our official definition of Green’s ratio,
Metropolis-Hastings-Green is Reversible

We can now write down the transition probability kernel for the Metropolis-Hastings-Green update. As we saw with Metropolis-Hastings, the transition probability has two terms. For accepted proposals, we propose \( y \) and then accept it, which happens with probability density \( a(x, \cdot) \) with respect to \( Q(x, \cdot) \) where \( a(x, y) \) is again the acceptance probability

\[
a(x, y) = \min[1, R(x, y)].
\]

Hence for any set \( A \)

\[
\int_A Q(x, dy) a(x, y)
\]

is the part of \( P(x, A) \) that results from accepted proposals. If the integral on the right hand side is taken over the whole state space, it gives the total probability that the proposal will be accepted. Thus the probability that the proposal is rejected is

\[
r(x) = 1 - \int Q(x, dy) a(x, y).
\]

If the proposal is rejected we stay at \( x \). Hence

\[
P(x, A) = r(x) I(x, A) + \int_A Q(x, dy) a(x, y).
\]

We now want to verify reversibility of the MHG update, but first we collect some simple facts about Radon-Nikodym derivatives.

**Lemma 3.15.** If \( \mu \) and \( \nu \) are positive real measures, \((\mu_a, \mu_s)\) is the Lebesgue decomposition of \( \mu \) relative to \( \nu \), \((\nu_a, \nu_s)\) is the Lebesgue decomposition of \( \nu \) relative to \( \mu \), then \( \mu_a \ll \nu_a \) and \( \nu_a \ll \mu_a \),

\[
\frac{d\mu}{d\nu} = \frac{d\mu_a}{d\nu_a} \quad \text{and} \quad \frac{d\nu}{d\mu} = \frac{d\nu_a}{d\mu_a}.
\]

Moreover,

\[
\frac{d\mu_a}{d\nu_a} = \left( \frac{d\nu_a}{d\mu_a} \right)^{-1}
\]

\( \mu_a \) (or \( \nu_a \)) almost everywhere.

**Proof.** Since \( \mu_a \ll \nu \) and \( \mu \perp \nu_s \), we must have \( \mu_a \ll \nu_a \), and similarly with \( \mu \) and \( \nu \) reversed. If \( \nu_a \) is concentrated on \( B \) and \( \nu_s \) on \( B^c \), then \( \mu(B^c) = 0 \), and if \( f = d\mu/d\nu \), then

\[
\mu(B^c) = \int_{B^c} f(x) \nu_s(dx) = 0
\]

Hence \( f = 0 \), almost everywhere (\( \nu_a \)), and

\[
\mu(A) = \int_A f(x) \nu(dx) = \int_A f(x) \nu_a(dx)
\]
which shows that \( f = d\mu/d\nu = d\mu_a/d\nu_a \).

Finally, if \( f = d\mu_a/d\nu_a \) and \( g = d\nu_a/d\mu_a \), then

\[
\mu_a(B) = \int_B f(x)\nu_a(dx) = \int_B f(x)g(x)\mu_a(dx)
\]

holds for all measurable \( B \), which implies \( fg = 1 \) almost everywhere \((\mu_a)\) \cite{Rudin1987, Theorem 1.39(b)). This is the same as almost everywhere \((\nu_a)\) because \( \mu_a \) and \( \nu_a \) have the same sets of measure zero.

**Corollary 3.16.** If \( R \) is defined by (3.40), then \( R(x,y) = 1/R(y,x) \) almost everywhere \( \mu \).

**Proof.** Let \( \mu_a \) denote the part of \( \mu \) that is absolutely continuous with respect to \( \mu \circ \varphi^{-1} \), and apply the lemma, yielding the conclusion that

\[
R = \frac{d(\mu_a \circ \varphi^{-1})}{d\mu_a} \quad \text{and} \quad S = \frac{d\mu_a}{d(\mu_a \circ \varphi^{-1})}
\]

\( RS = 1 \) almost everywhere \( \mu_a \), hence almost everywhere \( \mu \). Also

\[
\int_B R d\mu_a = (\mu_a \circ \varphi^{-1})(B) = \int_{\varphi^{-1}(B)} d\mu_a = \int_{\varphi^{-1}(B)} S d(\mu_a \circ \varphi^{-1}) = \int_B (S \circ \varphi) d\mu_a,
\]

the first equality being the definition of \( R \), the second the definition of \( \mu_a \circ \varphi^{-1} \), the third the definition of \( S \), and the fourth the change of variable theorem for abstract integration \cite[Theorem 16.12]{Billingsley1979}. Since this holds for all \( B \), we conclude \( R = S \circ \varphi \). \( \Box \)

**Theorem 3.17.** The Metropolis-Hastings-Green update is reversible with respect to \( \eta \).

**Proof.** What is to be shown is that

\[
\iint f(x)g(y)\eta(dx)P(x,dy) = \iint f(x)g(x)r(x)\eta(dx) + \iint f(x)g(y)\eta(dx)Q(x,dy)a(x,y)
\]

is unchanged when we interchange \( f \) and \( g \), as in the proof of Lemma 3.4. Again, the first term is obviously unchanged by interchanging \( f \) and \( g \). So we work on the second term.

\[
\iint f(x)g(y)a(x,y)\eta(dx)Q(x,dy) = \iint f(y)g(x)a(y,x)\eta(dy)Q(y,dx)
\]

is unchanged when we interchange \( f \) and \( g \), as in the proof of Lemma 3.4. Again, the first term is obviously unchanged by interchanging \( f \) and \( g \). So we work on the second term.

\[
\iint f(x)g(y)a(x,y)\eta(dx)Q(x,dy) = \iint f(y)g(x)a(y,x)\eta(dy)Q(y,dx)
\]

is unchanged when we interchange \( f \) and \( g \), as in the proof of Lemma 3.4. Again, the first term is obviously unchanged by interchanging \( f \) and \( g \). So we work on the second term.
the first equality from interchanging the dummy variables $x$ and $y$ and the second being (3.23). In order to finish the proof we only need to show that

$$a(x, y) = a(y, x)R(x, y), \quad x, y \in S,$$

which is the “detailed balance for densities” condition analogous to (3.22) that we need here.

The proof is just like the proof of Corollary 3.5. In the case $R(x, y) \geq 1$ we have

$$a(x, y) = 1 \quad \text{and} \quad a(y, x) = R(y, x) \quad (3.47)$$

which implies (3.46), and in the case (3.23) less than or equal to one we have (3.47) with $x$ and $y$ interchanged, which also implies (3.46). Now

$$\int g(x)h(y)a(x, y)f(x, y)\xi(dx, dy) = \int g(x)h(y)a(y, x)f(y, x)\xi(dx, dy)$$

$$= \int g(y)h(x)a(x, y)f(x, y)\xi(dx, dy) \quad (3.48)$$

$$= \int g(y)h(x)a(y, x)f(y, x)\xi(dx, dy)$$

where (3.46) gives the first equality, interchanging the dummy variables $x$ and $y$ gives the second, and the symmetry of $\xi$ gives the third. We do not need Fubini here, because there are no iterated integrals.\(^\dagger\)

\(^\dagger\)We do need something, because, strictly speaking, the notation $\xi(dx, dy)$ is meaningless, $\xi$ being a measure on $S^2$. What we need is the general change of variable formula for integration, for any function $w$, any measure $\xi$, and any measurable transformation $\varphi$

$$\int (w \circ \varphi) d\xi = \int w d(\xi \circ \varphi^{-1})$$

(Billingsley 1979, Theorem 16.12).

A formally correct argument now goes as follows. Let

$$w(x, y) = g(y)h(x)a(y, x)f(y, x)$$

[the last integrand in (3.48)]. Then we can rewrite the second and third equalities in (3.48) as

$$\int (w \circ \varphi) d\xi = \int w d(\xi \circ \varphi^{-1}) = \int w d\xi$$

the first equality being the change-of-variable formula and the second being the symmetry of $\xi$.

**Exercises**

3.1. Prove that Gibbs updates are idempotent (satisfy $P^2 = P$).

3.2. Prove that if each kernel $P_z$ in Theorem 3.1 is reversible with respect to $\pi$, then so is the kernel $Q$.

3.3. Verify directly that lines 2 and 3 of (3.2) are equal, that is, count the number of terms in the double sum, divide by $d! \cdot (d - 1)$ and get $d$. 

\[^{\dagger}\]We do need something, because, strictly speaking, the notation $\xi(dx, dy)$ is meaningless, $\xi$ being a measure on $S^2$. What we need is the general change of variable formula for integration, for any function $w$, any measure $\xi$, and any measurable transformation $\varphi$
3.4. Explain why $\mu$ was not required to be a $\sigma$-finite measure in the definition of “unnormalized probability density” at the beginning of Section 3.2.1. Show that if $h$ is an unnormalized density with respect to $\mu$ and $h$ is strictly positive, then $\mu$ is automatically $\sigma$-finite, it need not be part of the definition. Then show that even if $h$ is not strictly positive, the restriction of $\mu$ to the support of $h$ (i.e., the set $\{ x : h(x) > 0 \}$ is $\sigma$-finite.

3.5. Show that the simulation method described for the Poisson process does indeed satisfy the characterizations in Theorems 3.6 and 3.7.

3.6. Redo the logistic regression example using the kyphosis data set that comes with S-PLUS. Calculate posterior means and variances with Monte Carlo standard errors. The info on the computing info web page may help.

If you are feeling adventurous, do probit instead of logit regression (the C library functions $\text{erf}$ and $\text{erfc}$ may help with the probit calculation).

3.7. Show that one-variable-at-a-time Metropolis-Hastings is a special case of Metropolis-Hastings-Green.

3.8. Formulate the Metropolis-Hastings analog of the hit and run algorithm of Section 3.1.3. Show that your algorithm is a special case of Metropolis-Hastings-Green with general state-dependent mixing and hence is valid with no further proofs. This is not new, see Chen and Schmeiser (1993), but don’t look up the reference. Reinvent the wheel.
Trans-dimensional Markov chain Monte Carlo

by Peter Green
(University of Bristol,
P.J.Green@bristol.ac.uk
http://www.stats.bris.ac.uk/~peter).

(Thanks to all my collaborators on trans-dimensional MCMC problems:
Carmen Fernández, Paolo Giudici, Miles Harkness, David Hastie, Matthew Hodgson, Antonietta Mira, Agostino Nobile, Marco Pievatolo, Sylvia Richardson, Luisa Scaccia and Claudia Tarantola.)

© University of Bristol, 2002

Trans-dimensional Markov chain Monte Carlo

What if ‘the number of things you don’t know is one of the things you don’t know’?

Ubiquitous in statistical modelling, both

- in traditional modelling situations such as variable selection in regression, and
- in more novel methodologies such as object recognition, signal processing, and Bayesian nonparametrics.

Formulate generically as joint inference about a model indicator $k$ and a parameter vector $\theta_k$, where the model indicator determines the dimension $n_k$ of the parameter, but this dimension varies from model to model.

Hierarchical model

Suppose given

- a prior $p(k)$ over models $k$ in a countable set $\mathcal{K}$, and
- for each $k$
  - a prior distribution $p(\theta_k|k)$, and
  - a likelihood $p(Y|k, \theta_k)$ for the data $Y$.

For definiteness and simplicity, suppose that $p(\theta_k|k)$ is a density with respect to $n_k$-dimensional Lebesgue measure, and that there are no other parameters, so that where there are parameters common to all models these are subsumed into each $\theta_k \in \mathcal{R}^{n_k}$.

Additional parameters, perhaps in additional layers of a hierarchy, are easily dealt with. Note that all probability distributions are proper.
The joint posterior

\[ p(k, \theta_k | Y) = \frac{p(k)p(\theta_k | k)p(Y | k, \theta_k)}{\sum_{k' \in K} \int p(k')p(\theta_k' | k')p(Y | k', \theta_k') \, dk'} \]

can always be factorised as

\[ p(k, \theta_k | Y) = p(k|Y)p(\theta_k | k, Y) \]

– the product of posterior model probabilities and model-specific parameter posteriors.
– very often the basis for reporting the inference, and in some of the methods mentioned below is also the basis for computation.

Note the generality of this basic formulation: it embraces both

- genuine model-choice situations, where the variable \( k \) indexes the collection of discrete models under consideration, but also
- settings where there is really a single model, but one with a variable dimension parameter, for example a functional representation such as a series whose number of terms is not fixed (in which case, \( k \) is unlikely to be of direct inferential interest).

Compatibility across models

Some would argue that responsible adoption of this Bayesian hierarchical model presupposes that, e.g., \( p(\theta_k | k) \) should be compatible in that inference about functions of parameters that are meaningful in several models should be approximately invariant to \( k \).

Such compatibility could in principle be exploited in the construction of MCMC methods (how?).

But it is philosophically tenable that no such compatibility is present, and we shall not assume it.

Non-Bayesian uses

Trans-dimensional MCMC has many applications other than to Bayesian statistics. Much of what follows will apply equally to them all; however, for simplicity, I shall use the Bayesian motivation and terminology throughout.

Across- and within-model simulation

Two main approaches:

- *across*: one MCMC simulation with states of the form \((k, \theta_k)\)
- *within*: separate simulations of \( \theta_k \) for each \( k \).
Across-model simulation

Reversible jump MCMC

The state space for an across-model simulation is \( \{(k, \theta_k)\} = \bigcup_{k \in \mathcal{K}} \{(k) \times \mathbb{R}^{n_k}\} \).

Mathematically, this is not a particularly awkward object. But at least a little non-standard!

We use Metropolis-Hastings to build a suitable reversible chain.

On the face of it, this requires measure-theoretic notation, which may be unwelcome! The point of the ‘reversible jump’ framework is to render the measure theory invisible, by means of a construction using only ordinary densities. Even the fact that we are jumping dimensions becomes essentially invisible!

Metropolis-Hastings on a general state space

We wish to construct a Markov chain on a state space \( \mathcal{X} \) with invariant distribution \( \pi \).

As usual in MCMC we will consider only reversible chains, so the transition kernel \( P \) satisfies the detailed balance condition

\[
\int_{(x, x')} \pi(dx) P(x, dx') = \int_{x'} \pi(dx') P(x', dx)
\]

(both integrals over \( (x, x') \in A \times B \)), for all Borel sets \( A, B \subset \mathcal{X} \).

Compare this with

\[
\pi(x) P(x, x') = \pi(x') P(x', x)
\]

Now \( \pi(dx) q(x, dx') \) is dominated by a symmetric measure \( \mu \) on \( \mathcal{X} \times \mathcal{X} \); let its density (Radon-Nikodym derivative) with respect to this \( \mu \) be \( f \). Then DB requires

\[
\int_{(x, x')} \alpha(x, x') f(x, x') |\mu(dx, dx')| = \int_{(x, x')} \alpha(x', x) f(x', x) |\mu(dx', dx)|
\]

Using the symmetry of \( \mu \), this is clearly satisfied for all Borel \( A, B \) if

\[
\alpha(x, x') = \min \left\{ 1, \frac{f(x', x)}{f(x, x')} \right\}.
\]

This might be written more informally in the apparently familiar form

\[
\alpha(x, x') = \min \left\{ 1, \frac{\pi(dx') q(x', dx) \alpha(x', x)}{\pi(dx) q(x, dx') \alpha(x, x')} \right\}.
\]
A constructive representation in terms of random numbers

Now let’s get rid of this abstraction!

Consider how the transition will be implemented; we find the dominating measure and Radon-Nikodym derivatives can be generated implicitly.

Assume $\mathcal{X} \subset \mathcal{R}^d$, and that $\pi$ has a density (also denoted $\pi$) with respect to $d$–dimensional Lebesgue measure.

At the current state $x$, we generate, say, $r$ random numbers $u$ from a known joint density $g$, and then form the proposed new state as a deterministic function of the current state and the random numbers: $x' = h(x, u)$, say.

The reverse transition from $x'$ to $x$ would be made with the aid of random numbers $u' \sim g'$ giving $x = h(x', u')$.

Detailed balance says the two integrals are equal: it holds if

$$\pi(x) g(u) \alpha(x, x') = \pi(x') g'(u') \alpha(x', x) \frac{\partial(x', u')}{\partial(x, u)}$$

where the last factor is the Jacobian of the diffeomorphism from $(x, u)$ to $(x', u')$.

Thus, a valid choice for $\alpha$ is

$$\alpha(x, x') = \min \left\{ 1, \frac{\pi(x') g'(u')}{\pi(x) g(u)} \left| \frac{\partial(x', u')}{\partial(x, u)} \right| \right\},$$

involving only ordinary joint densities.

The equilibrium probability of jumping from $A$ to $B$ is then an integral with respect to $(x, u)$:

$$\int_{(x, x' = h(x, u)) \in A \times B} \pi(x) g(u) \alpha(x, x') dx \, du.$$

The equilibrium probability of jumping from $B$ to $A$ is an integral with respect to $(x', u')$:

$$\int_{(x = h'(x', u'), x') \in A \times B} \pi(x') g'(u') \alpha(x', x) dx' \, du.$$

If the transformation from $(x, u)$ to $(x', u')$ is a diffeomorphism (the transformation and its inverse are differentiable), then we can apply the standard change-of-variable formula, to write this as an integral with respect to $(x, u)$.

What’s the point?

Perhaps a little indirect! – but a flexible framework for constructing quite complex moves using only elementary calculus.

The possibility that $r < d$ covers the typical case that given $x \in \mathcal{X}$, only a lower-dimensional subset of $\mathcal{X}$ is reachable in one step.

(The Gibbs sampler is the best-known example of this, since in that case only some of the components of the state vector are changed at a time, although the formulation here is more general as it allows the subset not to be parallel to the coordinate axes.)
Deliberate redundancy

Separating the generation of the random innovation \( u \) and the calculation of the proposal value through the deterministic function \( x' = h(x, u) \) is deliberate; it allows the proposal distribution \( q(x; B) = \int h(x, u) \, g(u) \, du \) to be expressed in many different ways, for the convenience of the user.

Dimension matching

Suppose the dimensions of \( x, x', u \) and \( u' \) are \( d, d', r \) and \( r' \) respectively, then we have functions
\[
  h : \mathbb{R}^d \times \mathbb{R}^r \to \mathbb{R}^d \quad \text{and} \quad h' : \mathbb{R}^{d'} \times \mathbb{R}^{r'} \to \mathbb{R}^{d'},
\]
used respectively in \( x' = h(x, u) \) and \( x = h'(x', u') \).

For the transformation from \( (x, u) \) to \( (x', u') \) to be a diffeomorphism requires that \( d + r = d' + r' \), so-called ‘dimension-matching’; if this equality failed, the mapping and its inverse could not both be differentiable.

Dimension matching is necessary but not sufficient.

The trans-dimensional case

But the main benefit of this formalism is that
\[
  \alpha(x, x') = \min \left\{ 1, \frac{\pi(x') g'(u')}{\pi(x) g(u)} \left| \frac{\partial(x', u')}{\partial(x, u)} \right| \right\},
\]
applies, without change, in a variable dimension context.

(Use the same symbol \( \pi(x) \) for the target density whatever the dimension of \( x \) in different parts of \( \mathcal{X} \).)

Provided that the transformation from \( (x, u) \) to \( (x', u') \) remains a diffeomorphism, the individual dimensions of \( x \) and \( x' \) can be different. The dimension-jumping is ‘invisible’.

Details of application to model-choice

We wish to use these reversible jump moves to sample the space \( \mathcal{X} = \bigcup_{k \in \mathcal{K}} \{ k \} \times \mathbb{R}^{n_k} \) with invariant distribution \( \pi \), which here is \( p(k, \theta_k | Y) \).

Just as in ordinary MCMC, we typically need multiple types of moves to traverse the whole space \( \mathcal{X} \). Each move is a transition kernel reversible with respect to \( \pi \), but only in combination do we obtain an ergodic chain.

The moves will be indexed by \( m \) in a countable set \( \mathcal{M} \), and a particular move \( m \) proposes to take \( x = (k, \theta_k) \) to \( x' = (k', \theta'_{k'}) \) or vice-versa for a specific pair \( (k, k') \); we denote \( \{ k, k' \} \) by \( \mathcal{K}_m \).
The detailed balance equation becomes

\[
\int_{(x,x') \in A \times B} \pi(dx) q_m(x, dx') \alpha_m(x, x') \] 

\[
= \int_{(x,x') \in A \times B} \pi(dx') q_m(x', dx) \alpha_m(x', x)
\]

for each \( m \), where now \( q_m(x, dx') \) is the joint distribution of move type \( m \) and destination \( x' \).

The complete transition kernel is obtained by summing over \( m \), so that for \( x \not\in B \),
\[
P(x, B) = \sum_M \int_B q_m(x, dx') \alpha_m(x, x').
\]

Toy example

..... of no statistical use at all!

Suppose \( x \) lies in \( \mathcal{R} \cup \mathcal{R}^2 \): \( \pi(x) \) is a mixture:
with probability \( p_1 \), \( x \) is \( U(0, 1) \),
with probability \( p_2 \), it is Uniform on the triangle \( 0 < x_2 < x_1 < 1 \).

I will use three moves:

1. within \( \mathcal{R} \): \( x \to U(x - \epsilon, x + \epsilon) \), suppressing moves outside \((0, 1)\).
2. within \( \mathcal{R}^2 \): \((x_1, x_2) \to (1 - x_2, 1 - x_1)\).
3. between \( \mathcal{R} \) and \( \mathcal{R}^2 \)

In \( \mathcal{R} \), choose (1) or (3) with probabilities \( 1 - r_1, r_1 \).
In \( \mathcal{R}^2 \), choose (2) or (3) with probabilities \( 1 - r_2, r_2 \).
Thus \( j_1(x) = r_1 \) for all \( x \in \mathcal{R} \) and \( j_2(x') = r_2 \) for all \( x' \in \mathcal{R}^2 \).

The acceptance probability derivation is modified correspondingly, and yields
\[
\alpha_m(x, x') = \min \left\{ 1, \frac{\pi(x') j_m(x') g_m(u')}{\pi(x) j_m(x) g_m(u)} \left| \frac{\partial(x', u')}{\partial(x, u)} \right| \right\}.
\]

Here \( j_m(x) \) is the probability of choosing move type \( m \) when at \( x \), the variables \( x, x', u, u' \) are of dimensions \( d_m, d'_m, r_m, r'_m \) respectively, with \( d_m + r_m = d'_m + r'_m \), we have \( x' = h_m(x, u) \) and \( x = h'_m(x', u') \), and the Jacobian has a form correspondingly depending on \( m \).

Of course, when at \( x = (k, \theta_k) \), only a limited number of moves \( m \) will typically be available, namely those for which \( k \in \mathcal{K}_m \). With probability \( 1 - \sum_{m: k \in \mathcal{K}_m} j_m(x) \) no move is attempted.

Dimension-changing with move (3)

Proposal:
To go from \( x \in \mathcal{R} \) to \((x_1, x_2) \in \mathcal{R}^2 \), draw \( u \) from \( U(0, 1) \) [so \( g_0(u) = 1 \) if \( 0 < u < 1 \)] and propose \((x_1, x_2) = (x, u) \) For reverse move, no \( u' \) required [write \( g_0(u') \equiv 1 \) and set \( x = x_1 \)]. This certainly gives a bijection: \( (x, u) \leftrightarrow (x_1, x_2) \), with Jacobian \( = 1 \).

Acceptance decision:
\[
\alpha = \min \left\{ 1, \frac{\pi(x') j_3(x') g_3(u')}{\pi(x) j_3(x) g_3(u)} \left| \frac{\partial(x', u')}{\partial(x, u)} \right| \right\}
\]

\[
= \min \left\{ 1, \frac{p_2 I[x < x_1]}{p_1} \frac{r_2}{r_1} \left| I[u < x] \right| \right\}
\]

For reverse move, \( \alpha = \min \{ 1, (p_1 r_1) / (2 p_2 r_2) \} \).
First 30 steps

\[ p_1 = 0.4, \, p_2 = 0.6, \, r_1 = 0.7, \, r_2 = 0.4, \, \epsilon = 0.3. \]

5000 steps

Some remarks and ramifications

- key role of joint state-proposal equilibrium distributions \[ \pi(dx | q(x, x')) \]

- insights into Metropolis-Hastings applying quite generally
  - state-dependent mixing permissible if move probabilities enter into the acceptance probability calculation
  - contrast between this randomised proposal mechanism, and related mixture proposals
  - (contrary to some accounts that connect it with the jump in dimension) the Jacobian comes into the acceptance probability only because the proposal destination \[ x' = h(x, u) \] is specified indirectly

- nested models: RJ \( \equiv \) proposals with atoms and usual M-H formula

- there are alternative derivations and descriptions, e.g. Waagepetersen and Sorensen (2001) and Besag (1997, 2000) (giving a novel formulation in which variable dimension notation is circumvented by augmenting \( x \) by \( u \))

- RJ is only Metropolis-Hastings (so if it doesn’t seem to work....)
Relations to other across-model approaches

Several alternative formalisms for across-model simulation are more or less closely related to reversible jump.

Jump diffusion


Had they corrected for the time discretisation by a M-H accept/reject decision (Metropolis-adjusted Langevin algorithm), this would have been an example of reversible jump.

Phillips and Smith (1996) applied jump-diffusion to a variety of Bayesian statistical tasks, including mixture analysis, object recognition and variable selection.

Point processes, with and without marks

Point processes: natural example of a variable-dimension distribution, since the number of points in view is random; in the basic case, a point has only a location, but more generally has a mark, a random variable in a general space.

A continuous-time Markov chain approach to simulating certain spatial point processes using birth-and-death processes was investigated by Preston (1977) and Ripley (1977).

– Geyer and Møller (1994) proposed a M-H sampler, as an alternative; their construction is a special case of reversible jump.

Product-space formulations

Several relatives of RJ work in a product space framework, in which the simulation keeps track of all \( \theta_k \), not only the ‘current’ one.

The state space is \( \mathcal{K} \times \otimes_{k \in \mathcal{K}} \mathcal{R}^{n_k} \) instead of \( \bigcup_{k \in \mathcal{K}} (\{k\} \times \mathcal{R}^{n_k}) \).

Advantage: circumvents the trans-dimensional character of the problem

Cost: requires that the target distribution be augmented to model all \( \theta_k \) simultaneously (for some variants of this approach, this is just a formal device, for others it leads to significantly extra work).
Carlin and Chib (1995)

Let $\theta_{-k}$ denote all $\theta_l, l \neq k$ catenated together. Then the joint distribution of $(k, (\theta_l : l \in K), Y)$ can be expressed as

$$p(k)p(\theta_k | k)p(\theta_{-k} | k, \theta_k)p(Y | k, \theta_k),$$

making the natural assumption that

$$p(Y | k, (\theta_l : l \in K)) = p(Y | k, \theta_k).$$

The third factor $p(\theta_{-k} | k, \theta_k)$ has no effect on the joint posterior $p(k, \theta_k | Y)$; the choice of these ‘pseudo-priors’ is entirely a matter of convenience, but may influence sampler efficiency.

Carlin and Chib used conditionally independent pseudo-priors: $p(\theta_{-k} | k, \theta_k) = \prod_{l \neq k} p(\theta_l | k)$, and assumed $p(\theta_l | k)$ does not depend on $k$ for $k \neq l$.

They used a Gibbs sampler, updating $k$ and all $\theta_l$ in turn: involves sampling from the pseudo-priors, so they design these pseudo-priors to ensure reasonable efficiency, by approximate matching to the posteriors: $p(\theta_l | k) \approx p(\theta_l | l, Y)$.

Variants on Carlin and Chib

Green and O’Hagan (1998) pointed out both that M-H moves could be made in this setting: also there is no need to update $\{\theta_l, l \neq k\}$ for irreducibility. In this form the pseudo-priors are only used in computing the update of $k$.

Dellaportas et al. (2002) proposed ‘Metropolised Carlin and Chib’ approach, in which joint model indicator/parameter updates were made: only necessary to resample the parameter vectors for the current and proposed models.

Composite model space framework

Godsill (2001) provides a general framework that embraces all of these methods, including reversible jump, facilitating comparisons between them. He takes a fixed pool of parameters $\{\theta_1, \theta_2, \ldots, \theta_N\}$, of which model $k$ needs only $\theta_{I(k)}$, parameter vectors that can overlap.

Then

$$p(k)p(\theta_{I(k)} | k)p(\theta_{-I(k)} | k, \theta_{I(k)})p(Y | k, \theta_{I(k)}).$$

The pseudo-prior is now $p(\theta_{-I(k)} | k, \theta_{I(k)})$.

This framework

- helps to reveal that a product-space sampler may or may not entail possibly cumbersome additional simulation, updating parameters that are not part of the ‘current’ model
- provides useful insight into some of the important factors governing the performance of reversible jump
Godsill’s formulation deserves further attention, as it provides a useful language for comparing approaches, and in particular examining one of the central unanswered questions in trans-dimensional MCMC:

Suppose the simulation leaves model $k$ and later returns to it. With reversible jump, the values of $\theta_k$ are lost as soon as we leave $k$, while with some versions of the product-space approach, the values are retained until $k$ is next visited. Intuitively either strategy has advantages and disadvantages for sampler performance, so which is to be preferred?

Alternatives to joint model-parameter sampling

The direct approach of an across-model simulation is in many ways the most appealing, but alternative indirect methods that treat the unknowns $k$ and $\theta_k$ differently should not be neglected.

Integrating out the parameters  If in each model $k$, the prior is conjugate for the likelihood, then $p(\theta_k | k, Y)$ may be explicitly available, and thence can be calculated the marginal likelihoods

$$p(Y | k) = \frac{p(\theta_k | k) p(Y | k, \theta_k)}{p(\theta_k | k, Y)}$$

and finally the posterior probabilities $p(k | Y) \propto p(k) p(Y | k)$.

Within-model simulation

If samplers for the within-model posteriors $p(\theta_k | Y, k)$ are available for each $k$, joint posterior inference for $(k, \theta_k)$ can be constructed by combining separate simulations conducted within each model (see Carlin and Louis (1996, §6.3.1) for more detail).

The posterior $p(\theta_k | Y, k)$ for the parameters $\theta_k$ is the target for an ordinary Bayesian MCMC calculation for model $k$.

For the posterior model probabilities, since

$$\frac{p(k_1 | Y)}{p(k_0 | Y)} = \frac{p(k_1) p(Y | k_1)}{p(k_0) p(Y | k_0)}$$

(the second factor is Bayes factor for model $k_1$ vs. $k_0$), to find $p(k | Y)$ for all $k$ it is sufficient to estimate the marginal likelihoods

$$p(Y | k) = \int p(\theta_k, Y | k) d\theta_k$$

separately for each $k$, using individual MCMC runs.
Estimating marginal likelihoods

\[
p(Y|k) = \left\{ \int \left[ p(\theta_k|Y) / p(Y|k, \theta_k) \right] d\theta_k \right\}^{-1}
\]

\[
= \int p(Y|k, \theta_k) p(\theta_k|k) d\theta_k
\]

so we have the estimates

\[
\hat{p}_1(Y|k) = N \sum_{t=1}^{N} \left\{ p(Y|k, \theta_k^{(t)}) \right\}^{-1}
\]

and

\[
\hat{p}_2(Y|k) = N^{-1} \sum_{t=1}^{N} p(Y|k, \theta_k^{(t)})
\]

based on MCMC samples \( \theta_k^{(1)}, \theta_k^{(2)}, \ldots \) from the posterior \( p(\theta_k|Y, k) \) and the prior \( p(\theta_k|k) \), respectively.

Both of these are simulation-consistent, but have high variance, with possibly few terms contributing substantially to the sums in each case. Composite estimates, based like \( \hat{p}_1 \) and \( \hat{p}_2 \) on the importance sampling identity \( E_p(f) = E_q(f p/q) \), perform better, including those of Newton and Raftery (1994) and Gelfand and Dey (1994).

For example, Newton and Raftery propose to simulate from a mixture \( \bar{p}(\theta_k; Y, k) \) of the prior and posterior, and use

\[
\bar{p}_3(Y|k) = \frac{\sum_{t=1}^{N} p(Y|k, \theta_k^{(t)}) w(\theta_k^{(t)})}{\sum_{t=1}^{N} w(\theta_k^{(t)})}
\]

where \( w(\theta_k) = p(\theta_k|k) / \bar{p}(\theta_k; Y, k) \).

Chib (1995): new, indirect, estimates of the marginal likelihood based on the identity

\[
p(Y|k) = p(Y|k, \theta_k^*) p(\theta_k^*) / p(\theta_k^*|k, Y)
\]

for any fixed parameter point \( \theta_k^* \).

The factors in the numerator are available, and when the parameter can be decomposed into blocks with explicit full conditionals, the denominator can be estimated using simulation calculations that use the same Gibbs sampling steps as the posterior simulation.

(Note, however, that Neal (1999) has demonstrated that Chib’s application of this idea to mixture models is incorrect.)

Chib and Jeliazkov (2001) extend the idea to cases where Metropolis-Hastings is needed.

Some issues in choosing a sampler

- Is \( k \) a model indicator really, or a parameter?
- Do we want results across \( k \), within each \( k \), or for one \( k \) of interest?
- Jumping between models as an aid to mixing (c.f. simulated tempering: mixing may be better in the ‘other’ model)
- Are samplers for individual models already written and tested?
- Are standard strategies like split/merge likely to work?
- Trade-off between remembering and forgetting \( \theta_k \) when leaving model \( k \)
Methodological extensions

A simple automatic generic RJ sampler

For each model $k$, fix a $n_k$–vector $\mu_k$ and a $n_k \times n_k$–matrix $B_k$.

Suppose we are at $(k, \theta_k)$ and have proposed a move to model $k'$, drawn from some transition matrix $(r_{k,k'})$.

We set:

\[ \theta'_{k'} = \begin{cases} 
\mu_{k'} + B_{k'}[RB_k^{-1}(\theta_k - \mu_k)]_u & \text{if } n_{k'} < n_k \\
\mu_{k'} + B_{k'}RB_k^{-1}(\theta_k - \mu_k) & \text{if } n_{k'} = n_k \\
\mu_{k'} + B_{k'} R \left( B_k^{-1}(\theta_k - \mu_k) \right) & \text{if } n_{k'} > n_k 
\end{cases} \]

Here $[\cdots]^m$ denotes the first $m$ components of a vector, $R$ is a fixed orthogonal matrix of order $\max\{n_k, n_{k'}\}$, and $u$ is a $(n_{k'} - n_k)$–vector of random numbers with density $g(u)$.

Note that if $n_{k'} \leq n_k$, the proposal is deterministic (apart from the choice of $k'$).

Since everything is linear, the Jacobian is trivial: if $n_{k'} > n_k$, we have

\[ \frac{\partial(\theta_{k'})}{\partial(\theta_k, u)} = \frac{B_{k'}}{B_k} \]

Thus the acceptance probability is $\min\{1, A\}$ where

\[ A = \frac{p(k', \theta'_{k'} | y) r_{k', k} |B_k|}{p(k, \theta_k | y) r_{k,k'} |B_k|} \times \begin{cases} 
g(u) & \text{if } n_{k'} < n_k \\
1 & \text{if } n_{k'} = n_k \\
g(u)^{-1} & \text{if } n_{k'} > n_k 
\end{cases} \]

Since it is orthogonal, the matrix $R$ doesn’t appear.

If the targets $p(\theta_k | k, y)$ were normal distributions, $N(\mu_k, B_kB_k^T)$, if the innovation variables $u$ were $N(0, I)$, and if we could choose

\[ r_{k,k'} / r_{k',k} = p(k' | Y) / p(k | Y), \]

these proposals would already be in detailed balance, with no need to compute the M-H accept/reject decision. This is the motivation.

The idea might work adequately, if $p(\theta_k | k, y)$ are reasonably unimodal, with mean and variance approximately equal to $\mu_k$ and $B_kB_k^T$. Simple modifications:

- use $t$-distributions in place of the normals for $u$
- randomise over the orthogonal matrix $R$ – or, to simplify implementation, take $R$ to be a random permutation matrix
- use skewness transformations (David Hastie)
- use mixtures (Christophe Andrieu)

In practice, determine $\mu_k$ and $B_k$ by short pilot runs within each $k$ – only practical for a small finite set of models

Some experiments

These use a Fortran program, which calls a function written by the user to compute:

- $\log p(k, \theta_k, y)$
- the number of models
- their dimensions, and
- rough settings for the centre and spread of each variable, used for initial values and spread parameters for the RWM moves

The code is set up to alternate between model-jumping moves as described above, and within-model moves by RWM.
(a) Variable selection in a small logistic regression problem

Dellaportas et al. (2002) illustrate their algorithm comparisons on a $2 \times 2$ factorial experiment with a binomially distributed response. All 5 interpretable models are entertained, with numbers of parameters $(n_k)$ equal to 1, 2, 2, 3 and 4 respectively. We use the same prior settings, etc.

One million sweeps of the automatic sampler - many more than is needed for reliable results - takes about 18 seconds on a 800MHz PC. The acceptance rate for the model-jumping moves was 29.4%, and the integrated autocorrelation time for estimating $E(k|y)$ was estimated to be 2.90. The posterior model probabilities were computed to be $(0.0051, 0.4929, 0.0113, 0.4388, 0.0519)$, consistent with the results of Dellaportas et al.

(b) Change point analysis for a point process

We revisit the change point analysis of the coal mine disaster data. In this illustration, we condition on $1 \leq k \leq 6$. The prior settings, etc., are as in Green (1995). There are $2k + 1$ parameters in model $k$.

For this problem, 1 million sweeps takes about 28 seconds on a 800MHz PC.

On this problem, the automatic sampler mixes much less well (presumably due to the extremely multi-modal parameter posteriors): the acceptance rate for model-jumping is 5.9%, while the integrated autocorrelation time rises to 118.

The sampler described in Green (1995) takes 14 seconds for 1 000 000 sweeps on this computer, with an acceptance rate of 21% and estimated autocorrelation time of 67.8. The relative efficiency of the automatic sampler is only $(14 \times 67.8)/(28 \times 118) \approx 29\%$, but of course the implementation time was far less.

Delayed rejection

An interesting modification to Metropolis-Hastings is the splitting rejection idea of Tierney and Mira (1999), which has recently been extended to the RJ setting by Green and Mira (2001), who call it delayed rejection.

If a proposal is rejected, instead of ‘giving up’, staying in the current state, and advancing time to the next transition, we instead attempt a second proposal, usually from a different distribution, and possibly dependent on the value of the rejected proposal.

It is possible to set the acceptance probability for this second-stage proposal so that detailed balance is obtained, individually within each stage. The idea can be extended to further stages.

By the results of Peskun (1973) and Tierney (1998), this always reduces asymptotic variances of ergodic averages, on a sweep-by-sweep basis, since the probability of moving increases by stage.

Whether it is actually worth doing will depend on whether the reduction in Monte Carlo variance compensates for the additional computing time for the extra stages; the experiments in Green and Mira (2001) suggest that this can be the case.
The second-stage acceptance probability is calculated similarly as in deriving RJ above. We use two vectors of random numbers \( u_1 \) and \( u_2 \), drawn from \( g_1 \) and \( g_2 \), and two deterministic functions mapping these and the current state into the proposed new states, \( y = h_1(x, u_1) \) and \( z = h_2(x, u_1, u_2) \).

Both \( u_1 \) and \( u_2 \) appear in \( z \) to allow this second-stage proposal to be dependent on the rejected first-stage candidate \( y \); for example, \( z \) may be a move in a different ‘direction’ in some sense.

![Diagram](image)

The first-stage proposal is accepted with probability \( \alpha_1(x, y) \) calculated as usual:

\[
\alpha_1(x, y) = \min \left\{ \frac{\pi(y) g_1(u_1')}{\pi(x) g_1(u_1)} \left| \frac{\partial(y, u_1')}{\partial(x, u_1)} \right| \right\},
\]

where \( u_1' \) is such that \( x = h_1(y, u_1') \).

Consider the case where the move to \( y \) is rejected. We need to find \( \alpha_2(x, z) \) for detailed balance at the second-stage. As for one stage, we set up a diffeomorphism between \( (x, u_1, u_2) \) and \( (z, u_1, u_2) \), where \( u_1 \) and \( u_2 \) would be the random numbers used in the first- and second-stage attempts from \( z \). Then \( x = h_2(z, u_1, u_2) \) and the first-stage move, if accepted, would have taken us to \( y^* = h_1(z, u_1) \).

Equating integrands after making the change of variable, we find that a valid acceptance probability is

\[
\alpha_2(x, z) = \min \left\{ \frac{\pi(z) g_1(u_1) g_2(u_2) [1 - \alpha_1(z, y^*)]}{\pi(x) g_1(u_1) g_2(u_2) [1 - \alpha_1(x, y)]} \times \left| \frac{\partial(z, u_1, u_2)}{\partial(x, u_1, u_2)} \right| \right\}.
\]

In a model-jumping problem, we would commonly take \( y \) and \( z \) to lie in the same model, and \( y^* \) to be in the same model as \( x \).

![Diagram](image)

Efficient proposal choice for reversible jump MCMC

The most substantial recent methodological contribution to reversible jump MCMC generally is work by Brooks, Giudici and Roberts (RSS ordinary meeting, Banff, July 2002, [RSS(B), 2002]) on the efficient construction of proposal distributions.

This is focussed mainly on the quantitative question of selecting the proposal density \( g(u) \) well, having already fixed the transformation \( x' = h(x, u) \) into the new space. The qualitative choice of such a transformation \( h \) is perhaps more elusive and challenging.
Brooks et al. propose several new methods, falling into two main classes.

1. using analysis of the acceptance rate as a function of $u$ for small $u$ (having chosen an appropriate scale of measurement for it), having assumed that uniformly high acceptance rate is desirable.

2. methods that work in a product-space formulation, including some novel formulations with autoregressively constructed auxiliary variables.

Their methods are implemented and compared on examples including choice of autoregressive models, graphical gaussian models, and mixture models.

References and preprints available from

http://www.stats.bris.ac.uk/~peter

.../papers/hssschapter.ps

P.J.Green@bristol.ac.uk

6. Random walk Metropolis algorithm

The key point to RWM is that the proposal distribution is ‘centred’ upon the current state of the Markov chain. That is,

\[ y^* = x + W, \]

where the random variable \( W \) is such that \( E[W] = 0 \) and is symmetric. Thus

\[ \alpha(x, y^*) = \min \left\{ \frac{\pi(y^*)}{\pi(x)}, 1 \right\}. \]
RWM is possibly the most used MCMC algorithm but why? There are a number of reasons.

1. It generally works pretty well.

2. It is very straightforward to implement even in high dimensions.

3. Correlation/dependence between different parameters can be overcome.
Thus far the RWM algorithm seems ideal but it does have some drawbacks. These include:

1. It is too general.

2. There can still be problems with convergence.

3. There is still the problem of dependence between samples and the need for appropriate choice of variance for the proposal distribution.

4. Multi-modal distributions. There can be problems with the chain moving between different modes.
6.2 Proposal variance

The choice of proposal variance is a very important issue for RWM. We shall illustrate why with a simple example.

Let $X \sim N(0, 1)$, and therefore

$$\pi(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2).$$

For $\sigma > 0$, let the proposal distribution $W \sim N(0, \sigma^2)$, thus $y^* \sim N(x, \sigma^2)$. 
The algorithm was run with the following choices of $\sigma$; $\sigma = 0.1, 5, 50$.

The algorithm starts in stationarity so there is no need for a burn-in period.

The results of 500 iterations are plotted below, in the order $\sigma = 0.1, 5, 50$. 
All three choices of $\sigma$ result in the same stationary (posterior) distribution for the Markov chain.

The (acceptance probability) results were:

\[
\begin{array}{c|ccc}
\sigma & 0.1 & 5 & 50 \\
\hline
\text{Acceptance Probability} & 0.923 & 0.243 & 0.018 \\
\end{array}
\]
For each of the three choices of $\sigma$, the batch mean estimate of the MCSE was calculated based upon 100 batches of size 1000, (ie. a larger sample consisting of 100000 iterations were obtained for each $\sigma$.) The results were:

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>0.1</th>
<th>5</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCSE</td>
<td>365.0</td>
<td>8.6</td>
<td>61.9</td>
</tr>
</tbody>
</table>

**Rule of Thumb:** Tune the proposal variance such that approximately 1 in 4 proposed moves are accepted.
6.3 Multi-modal distribution

The main problem being that the algorithm can get ‘stuck’ at one of the modes. That is, the MCMC chain moves around but remains close to one modal value and takes a very long time to explore the other modal values.

Let $X$ have posterior density,

$$
\pi(x) = \frac{1}{2\sqrt{2\pi}} \left( \exp(-x^2/2) + \exp(-(x - 10)^2/2) \right).
$$
At a first glance, both runs appear to have converged, judging by a visual inspection of the output.

However, closer inspection of the $y$-axis shows that for $\sigma = 1$, the MCMC chain has spent all 1000 iterations centred about the mode at 10.

Therefore the sample obtained is not representative of the posterior distribution even though the algorithm is correct.
On the other hand, in the case $\sigma = 10$ the algorithm moves freely between the two modal values of 0 and 10. Thus this sample is far more representative of the distribution $X$.

The moral of the story is that MCMC can appeared to have converged even when they have not.

It is important that the MCMC chain explores the whole of the posterior distribution and does not get ‘stuck’ at one mode.
For multi-modal distributions hybrid RWM algorithms can be very effective. That is, RWM algorithms which has proposed moves with both low and high proposal variances.

**Low variance:** For moving within modes.

**High variance:** For moving between modes.
6.4.1 Multiplicative RWM

Let $X$ have pdf $\pi(x)$, where $\pi(x) = 0$ for $x < 0$.

Multiplicative RWM essentially transforms the problem onto the log-scale.

Propose

$$\log Y \sim N(\log x, \sigma^2) \quad \text{as} \quad n \to \infty.$$ 

The acceptance probability is

$$\min \left\{ 1, \frac{Y \pi(Y)}{x \pi(x)} \right\}.$$
We demonstrate the benefits of multiplicative RWM with

\[ \pi(x) = \begin{cases} 
\frac{3}{(1+x)^4} & (x > 0) \\
0 & (x \leq 0) 
\end{cases} \]

Left: RWM with \( \sigma = 2 \); Right: Multiplicative RWM with \( \sigma = 1 \).
RWM for heavy tailed distributions

Random walk Metropolis performs badly for heavily tailed distributions.

For $\pi(x) = \frac{3}{(1+x)^4}$, both algorithms perform well for $0 \leq x \leq 10$.

In 100000 iterations, on average 75 observations greater than 10 and 11 observations greater than 20.

For the RWM algorithm there was only 23 observations greater than 10 and none greater than 20 (the maximum value was 13.5).

For multiplicative RWM there were 67 observations greater than 10 and 17 observations greater than 20.

Maximum value for multiplicative RWM: 41
6.4.2 MALA

Standard RWM algorithm proposes a new value centered on the current value.

Make use of knowledge of $\pi$.

Take into account the derivatives of $\pi$. 
The MALA algorithm (Metropolis adjusted Langevin algorithm) is as follows.

1. Propose a new value $Y$, by setting

$$Y = x + \frac{\sigma^2}{2} \frac{d}{dx} \log \pi(x) + N(0, \sigma^2),$$

where $X_t = x$.

2. Accept the proposed value with probability

$$\min \left\{ 1, \frac{\pi(Y) \exp \left( -\frac{1}{2\sigma^2} \left\{ x - Y - \frac{\sigma^2}{2} \frac{d}{dx} \log \pi(Y) \right\}^2 \right)}{\pi(x) \exp \left( -\frac{1}{2\sigma^2} \left\{ Y - x - \frac{\sigma^2}{2} \frac{d}{dx} \log \pi(x) \right\}^2 \right)} \right\}. $$

3. If the proposed value is accepted set $X_{t+1} = Y$, otherwise set $X_{t+1} = x$. 

Note that the proposal is not symmetric, \( q(x, Y) \neq q(Y, x) \).

The class of models to which MALA can be applied is smaller than RWM and MALA is more involved in the need to find the derivative of \( \pi \).

Where MALA can be used it is preferable to RWM.

For high-dimensional problems MALA is seen to have some very nice properties. In particular, if \( d \), the number of dimensions, is large, the optimal choice of \( \sigma^2 \) for MALA is \( O(d^{-1/3}) \) compared with \( O(d^{-1}) \) for RWM.
6.5 Example: Logit regression

35 patients.

Data: Angina (1–has angina, 0–does not have angina).

Covariate: cholesterol level.
Proposed model

\[ Y_j \sim \text{Bernoulli} \left( \frac{\exp(\alpha + \beta c_j)}{1 + \exp(\alpha + \beta c_j)} \right), \]

\( Y_j = 1 \) if the \( j^{th} \) patient has angina (\( Y_j = 0 \) otherwise);

\( c_j \) is the cholesterol level of patient \( j \);

\[ P(Y_j = 1) = \frac{\exp(\alpha + \beta c_j)}{1 + \exp(\alpha + \beta c_j)}. \]

Parameters \( \alpha \) and \( \beta \) are unknown and are to be estimated using MCMC.
Likelihood

For patient $j$,

$$P(Y_j = y_j | \alpha, \beta, c_j) = \left( \frac{\exp(\alpha + \beta c_j)}{1 + \exp(\alpha + \beta c_j)} \right)^{y_j} \times \left( \frac{1}{1 + \exp(\alpha + \beta c_j)} \right)^{1-y_j}.$$ 

Therefore since the patients are independent,

$$L(\alpha, \beta | y, c) = \prod_{j=1}^{35} P(Y_j = y_j | \alpha, \beta, c_j)$$

$$= \prod_{j=1}^{35} \left( \frac{\exp(\alpha + \beta c_j)}{1 + \exp(\alpha + \beta c_j)} \right)^{y_j} \times \left( \frac{1}{1 + \exp(\alpha + \beta c_j)} \right)^{1-y_j}. $$
Priors

$\alpha$ and $\beta$ can take any values positive or negative.

Suggest normally distributed priors.

$\pi(\alpha) \sim N(0, \sigma^2)$ and $\pi(\beta) \sim N(0, \sigma^2)$.

Choosing $\sigma$ to be large will give uninformative vague priors.
The mean (standard deviations) of $\alpha$ and $\beta$ are $-1.813(1.039)$ and $0.267(0.203)$, respectively. Posterior density plots for $\alpha$ and $\beta$ are given in the plot below.