Identification of Regeneration Times in MCMC Simulation, with Application to Adaptive Schemes

Anthony Brockwell
Carnegie Mellon University, abrock@stat.cmu.edu

Joseph B. Kadane
Carnegie Mellon University, kadane@stat.cmu.edu

Follow this and additional works at: http://repository.cmu.edu/statistics
Part of the Statistics and Probability Commons
Identification of Regeneration Times in MCMC Simulation, with Application to Adaptive Schemes

Anthony E. Brockwell and Joseph B. Kadane

December 13, 2002

Abstract

Regeneration is a useful tool in Markov chain Monte Carlo simulation, since it can be used to side-step the burn-in problem and to construct better estimates of the variance of parameter estimates themselves. It also provides a simple way to introduce adaptive behaviour into a Markov chain, and to use parallel processors to build a single chain. Regeneration is often difficult to take advantage of, since for most chains, no recurrent proper atom exists, and it is not always easy to use Nummelin’s splitting method to identify regeneration times. This paper describes a constructive method for generating a Markov chain with a specified target distribution and identifying regeneration times. As a special case of the method, an algorithm which can be “wrapped” around an existing Markov transition kernel is given. In addition, a specific rule for adapting the transition kernel at regeneration times is introduced, which gradually replaces the original transition kernel with an independence-sampling Metropolis-Hastings kernel using a mixture normal approximation to the target density as its proposal density. Computational gains for the regenerative adaptive algorithm are demonstrated in examples.

Keywords: regenerative simulation, adaptive Markov chain Monte Carlo, splitting, atoms, burn-in, mixture approximations, parallel processing

1 Introduction

Markov chain Monte Carlo (MCMC) methods have become popular in the last decade as a tool for exploring properties of distributions which are known only up to a constant of proportionality. The basic idea is to construct an ergodic Markov chain whose limiting distribution is the same as the distribution of interest $\pi$ (called the target distribution). Further details, as well as information about various aspects of implementation, are given in a variety of sources, including Tierney (1994), Gilks et al. (1996), and Robert and Casella (1999). Development of the underlying probabilistic theory of Markov chains taking values in a general state-space can be found in Revuz (1975), Nummelin (1984), and Meyn and Tweedie (1993).

There are several important problems associated with standard methods of constructing Markov chains for Monte Carlo simulation. The first is a convergence problem, sometimes known as the “burn-in problem”. One does not generally know how long it takes before the chain is (by some measure) sufficiently close
to its limiting distribution. A standard approach to dealing with this is simply to discard some initial portion of the chain, labelling it as a “burn-in” component of the chain. However, without the ability to start the chain with a “perfect sample” from the target distribution, the problem, although reduced, still remains. A second problem is the inherent correlation between successive elements of the chain, which makes it difficult to estimate the variance of the Monte Carlo estimates. One widely-adopted method for estimating this variance is the batch-means method, which relies on division of the Markov chain into equal-length segments, but as pointed out by Glynn and Iglehart (1990), batch-means estimates are not consistent for the true normalized variance. A related problem is that the chain might “mix” poorly. Even if the chain starts in exactly the limiting distribution, it may take a long time for the chain to explore all interesting parts of the state-space, particularly when the correlation between successive elements of the chain is high. A fourth problem is the simple computational burden required to carry out MCMC simulation. Errors in integrals approximated by MCMC simulation are approximately proportional to the inverse of the square root of the length of the chain. These are an order of magnitude larger than errors in other standard methods for numerical integration, and hence Markov chains need to be relatively long to give accurate estimates of integrals.

Regenerative simulation provides a means of dealing with all of these problems. As discussed in Crane and Iglehart (1975a), Crane and Iglehart (1975b), Crane and Lemoine (1977), Ripley (1987), and more recently in Mykland et al. (1995), Jones and Hobert (2001), and Robert and Casella (1999), regenerative simulation can be used to avoid the burn-in problem and to get consistent estimates of the variance of estimators themselves. Gilks et al. (1998) show how regeneration can be used to introduce so-called “adaptive” schemes into MCMC simulation. These schemes allow parameters of the transition kernel to be modified as the chain is being generated, to improve mixing properties. Mykland et al. (1995) hint at the possibility of using regeneration to “parallelize” generation of a Markov chain. Constructing a Markov chain on parallel processors can give dramatic improvements in speed. (See, e.g. Geyer, 1992, for discussion of the relative merits of constructing a single long chain rather than multiple short chains.)

Loosely speaking, a regenerative process “starts again” probabilistically at each of a set of random stopping times, called regeneration times. If regeneration times can be identified in an ergodic Markov chain, then tours of the chain between these times are (on an appropriate probability space) independent and identically distributed (i.i.d.) entities. Thus if a fixed number of tours is generated for the purpose of MCMC simulation (this is typically done by starting the chain in its regenerative state and stopping it at a regeneration time), it makes no sense to discard any initial tours, or indeed, any part of the first tour, since properties of the “clipped” tour would be different from properties of the other un-clipped tours. In this sense, the problem of burn-in is avoided. In addition, since the tours are i.i.d. when estimating the expectation of an arbitrary function of a parameter, the variance of the estimator itself can easily be estimated, and furthermore, there is no need for the tours to be generated on a single processor - they can be generated separately on different processors and then “patched together” in any order which doesn’t depend on the contents of the tours themselves. Once a mechanism for generating tours is in place, results of Gilks et al. (1998) establish that parameters (which could be, for instance, variances of proposal distributions used in a Metropolis-Hastings chain) can be modified at each regeneration time, without disrupting consistency of MCMC estimators. This procedure, carried out carefully, can significantly improve mixing of the chain.

Although every ergodic Markov chain is regenerative (this follows from Theorem 5.2.3 of Meyn and Tweedie, 1993), identification of regeneration times in a Markov chain is generally a difficult problem (unless the chain takes values in a finite state-space). Mykland et al. (1995) give an ingenious method,
based on the splitting technique of Nummelin (1978) for identifying regeneration times in a large class of Markov chains (including chains generated by the standard algorithms for MCMC simulation). Their method relies on establishing a so-called “minorization condition”, which is essentially a decomposition of the transition kernel of the chain into two components, one of which does not depend on the current state of the chain.

In this paper, we present an alternative way of identifying regeneration times, which relies on constructing a Markov chain on an enlarged state-space, and does not require checking of minorization conditions or analysis of the transition kernel of the chain. The procedure is to modify the initial target distribution $\pi$ by mixing it with a point mass concentrated on an “artificial atom” $\alpha$ which is outside the state-space $E$. It is then a straightforward matter (for instance, using the Metropolis-Hastings Algorithm) to construct a Markov chain with the new target distribution. For this chain, the state $\alpha$ is Harris-recurrent (i.e. with probability one, it occurs infinitely many times). By the Markov property, the times at which the new chain hits $\alpha$ are regeneration times. On the face of it, this might not seem particularly useful, since the new chain doesn’t have the desired target distribution. However, to recover an ergodic chain with limiting distribution $\pi$, it is sufficient simply to delete every occurrence of the state $\alpha$ from the new chain. The points immediately after the (deleted) occurrences of the state $\alpha$ are then regeneration times in a Markov chain with limiting distribution $\pi$. This makes identification of regeneration times trivial. Möller and Nicholls (2001) use a generalized form of this mixed target distribution for purposes of perfect-sampling. They do not, however address applications in regenerative simulation or parallel processing.

In addition to presenting this “constructive” method for identification of regeneration times, we describe how an existing transition kernel can be embedded into a new kernel for which identification of regeneration times is trivial. This is described in terms of our state-space augmentation approach, but the embedding can also be thought of as effectively adding an independence-sampler in between each application of the original kernel (Mykland et al., 1995, also advocate this idea at the end of their Section 4), and provides a simple way to make the transition from standard to regenerative MCMC simulation. We also describe an easily-implemented adaptive algorithm which progressively replaces an original kernel, with an independence sampler for which the proposal distribution is constructed as a crude mixture approximation to the target distribution. In terms of mixing, the adaptive algorithm typically outperforms non-adaptive algorithms, and also represents a refinement of some of the adaptive techniques described in Gilks et al. (1998). This idea is also quite similar to that used by Chauveau and Vandekerkhove (2002), but avoids problems associated with construction of a histogram to represent a current approximation to the target distribution.

Section 2 introduces our method for constructing a regenerative chain. Section 3 gives more explicit details on how it can be used for purposes of regenerative and adaptive simulation. A potentially adaptive algorithm is given for estimating the expectation of a function of an unknown parameter as well as the variance of the estimator itself, and a specific adaptation mechanism is described. In Section 4, we present examples of application of the method. Theoretical results are given in Appendix A.
2 The Method

We propose the following method for constructing a Markov chain and identifying regeneration times.

The main idea is to enlarge the state-space from \( E \) to

\[
E^* = E \cup \alpha,
\]

where \( \alpha \) is a new state called the *artificial atom*. Then it is possible (as described in Subsection 3.2) to construct a Markov chain \( \{Y_t, t = 0, 1, 2, \ldots\} \) with \( Y_0 = \alpha \) and with limiting distribution

\[
\pi_p^*(A) = (1 - p)\pi(A \setminus \alpha) + p I_A(\alpha),
\]

(1)

defined on appropriate subsets \( A \) of \( E^* \), where \( p \) is some constant in the interval \((0, 1)\). The new limiting distribution \( \pi_p^* \) is a mixture distribution which assigns mass \( p \) to the new state \( \alpha \) and mass \((1 - p)\) to the original distribution \( \pi \).

Next let \( \tau_Y(j), j = 1, 2, \ldots \), denote the \( j \)th time after time zero that the chain \( \{Y_t\} \) hits the state \( \alpha \), with \( \tau_Y(0) = 0 \), so that

\[
\tau_Y(j) = \min\{k > \tau_Y(j - 1) : Y_k = \alpha\}, \quad j = 1, 2, 3, \ldots,
\]

(2)

and define the tours \( Y^1, Y^2, \ldots \) to be the segments of the chain between the hitting times, that is

\[
Y^j = \{Y_t, \tau_Y(j - 1) < t \leq \tau_Y(j)\}, \quad j = 1, 2, \ldots.
\]

(3)

Once the Markov chain \( \{Y_t\} \) has been generated, the next step is to recover a regenerative chain with limiting distribution \( \pi \). This can be done quite simply. Define the chain \( \{Z_t, t = 0, 1, 2, \ldots\} \) to be exactly the chain \( \{Y_t\} \), with every occurrence of the state \( \alpha \) removed. Noting that the state \( \alpha \) occurs exactly once at the end of each tour \( Y^j \), \( \{Z_t\} \) can be constructed by stringing together the tours \( Y^j \) whose length is larger than one, after removing the last element (which is equal to \( \alpha \)) from each one. Tours \( Z^j \) of \( \{Z_t\} \) are then defined by their correspondence to the truncated tours of \( \{Y_t\} \). We denote by \( T^j \) the time at which the \((j + 1)\)th tour of \( \{Z_t\} \) begins, for \( j = 0, 1, 2, \ldots \) (with the convention that \( T_0 \) is equal to zero), so we can then write the tour lengths as \( N_j = T^j - T^{j-1}, j = 1, 2, \ldots \).

The following result, proved in Appendix A, establishes the validity of this procedure.

**Theorem 2.1** Suppose that \( \{Y_t, t = 0, 1, 2, \ldots\} \) is an ergodic Markov chain taking values in \( E^* \) with \( Y_0 = \alpha \) and limiting distribution \( \pi_p^* \) given by (1). Let the process \( \{Z_t\} \) be constructed from \( \{Y_t\} \) in the manner described above. Then \( \{Z_t\} \) is an ergodic Markov chain taking values in \( E \) with limiting distribution \( \pi \). Furthermore, the times \( T_j, j = 0, 1, 2, \ldots \) are regeneration times for the chain. In other words, the Markov chains \( \{Z_{t-T_i}^j, t = T_i, T_i + 1, \ldots\}, i \geq 0 \) are identically distributed, and given any \( T_i \) and non-negative integers \( s \) and \( t \), \( Z_s \) and \( Z_t \) are independent if \( s < T_i < t \).

3 Simulation Algorithms

In this section, we briefly review the technique of estimation using regenerative simulation, as discussed in Crane and Lemoine (1977) and, more recently, by others. We also explicitly state regenerative and
3.1 Estimating Functionals of the Target Distribution

One is often interested primarily in estimating

$$\pi_h = \int_E h(x) d\pi(x)$$

for some integrable function $h : E \rightarrow \mathbb{R}^d$. Let tours $Z_j$, tour lengths $N_j$, and cumulative tour lengths $T_j$ of the Markov chain $\{Z_i\}$ with limiting distribution $\pi$ be as defined in the previous section. Also let

$$H_j = \sum_{t=T_{j-1}}^{T_j-1} h(Z_t),$$

and assume that $H_j$ and $N_j$ have finite variances. Then by the strong law of large numbers, the ratio estimator

$$\hat{H}_n = \frac{\sum_{j=1}^n H_j}{\sum_{j=1}^n N_j} = \frac{\sum_{j=1}^n H_j}{T_j}$$

converges almost surely to $\pi_h$, and it follows from the central limit theorem that $\sqrt{n}(\hat{H}_n - \pi_h)$ converges to a $N(0, \sigma^2)$ distribution.

One way of estimating $\sigma^2$ is as follows. Let $V_j = H_j - \pi_h N_j$, so that $\{V_j\}$ is an i.i.d. sequence of random variables, and define $\bar{V}_n = n^{-1} \sum_{j=1}^n V_j$, $\bar{N}_n = n^{-1} \sum_{j=1}^n N_j$, and $\mu_N = \mathbb{E} N_1$. Then

$$\mathbb{E}(\sqrt{n}(\hat{H}_n - \pi_h)^2) = n \mathbb{E}(\bar{V}_n / \bar{N}_n)^2 \approx n \mathbb{E}(\bar{V}_n / \mu_N)^2 = \frac{1}{\mu_N^2} \text{Var}(\bar{V}_n) \simeq \hat{\sigma}_n^2,$$

where

$$\hat{\sigma}_n^2 = \frac{n^{-1} \sum_{j=1}^n (H_j - \bar{H}_n N_j)^2}{\bar{N}_n^2}.$$  \hspace{1cm} (7)

When $h(\cdot)$ is a vector-valued function, we estimate the covariance matrix of $\sqrt{n}(\hat{H}_n - \pi_h)$ by replacing the term $(H_j - \bar{H}_n N_j)^2$ in (7) with $(H_j - \bar{H}_n N_j)(H_j - \bar{H}_n N_j)^T$.

In the preceding argument, in order to avoid dealing with the ratio of random variables in (6), $\bar{N}$ is effectively treated as the constant $\mu_N$. Hence to limit the error in the approximation, it is desirable for the coefficient of variation $c_n$ of $\bar{N}$ to be small. Mykland et al. (1995) suggest that the estimator should not be used if the coefficient is larger than 0.01. The coefficient of variation $c_n$ depends on the distribution of the tour lengths $N_j$, which is usually not known. However, it may be estimated by

$$\hat{c}_n = \sum_{j=1}^n (N_j / T_n - 1/n)^2.$$  \hspace{1cm} (8)

Furthermore, noting that $c_n$ (apart from some random variation) is proportional to $n^{-1}$, if we have $\hat{c}_{n_1} > \epsilon$, then approximately $n_2 = n_1 (\hat{c}_{n_1}^{-1} - 1)$ additional tours are required to ensure that $c_{n_1 + n_2}$ is less than $\epsilon$. 

5
3.2 Constructing a Chain with Limiting Distribution $\pi^*_p$

In this section we give two methods which can be used to construct a chain with limiting distribution $\pi^*_p$. (recall the definition (1)).

3.2.1 Metropolis-Hastings

One of the most obvious ways to construct $\{Y_t\}$ is simply to use the standard Metropolis-Hastings Algorithm, making appropriate modifications to densities so that they are defined on $E^*$ instead of only on $E$.

Assume that the measure $\pi$ has a density (not necessarily with respect to Lebesgue measure), which, for the sake of simplicity, we will also denote by $\pi$. In order to allow for the usual case where the density is known only to within a constant of proportionality, we will assume that the density integrates to an unknown constant $\beta$.

Define the density

$$\pi^*(y) = \begin{cases} 
\pi(y), & y \in E, \\
k, & y = \alpha 
\end{cases}$$

on $E^*$, where $k$ is some positive constant, and let $g_\theta(Y_t, \cdot)$ denote a family of proposal distributions on $E^*$, satisfying the property that for each $\theta$, $g_\theta(y, \alpha) > 0$ for all $y \in E$ and $g_\theta(\alpha, A) > 0$ for all sets $A$ such that $\pi(A) > 0$. Also define the acceptance probabilities

$$a_1(Y_t, Z) = \min \left(1, \frac{\pi^*(Z) g_\theta(Z, Y_t)}{\pi^*(Y_t) g_\theta(Y_t, Z)} \right), \quad Y, Z \in E^*.$$  

We introduce the parameter $\theta$ so that we can use adaptive methods for modifying the proposal distribution as the chain runs; this is described in the next subsection. Then the standard Metropolis-Hastings Algorithm using proposal density $g_\theta(\cdot, \cdot)$, acceptance probabilities $a_1(\cdot, \cdot)$, and target density $\pi^*$, generates a Markov chain with limiting distribution $\pi^*_p, p = k(\beta + k)^{-1}$, on the augmented space $E$. (Note that $p$ is always strictly between 0 and 1, as required, in spite of the fact that $\beta$ is not known.)

3.2.2 A Hybrid Kernel

An alternative method for constructing $\{Y_t\}$ is to build a hybrid kernel around an existing transition kernel.

Suppose that a kernel $P_\theta(\cdot, \cdot)$ is given for an ergodic Markov chain with limiting distribution $\pi$, for instance, a Metropolis-Hastings or Gibbs sampling chain. The hybrid kernel consists of two components. The first component leaves the state at $\alpha$ if it is already $\alpha$, otherwise it updates the state using the $\pi$-invariant kernel $P_\theta(\cdot, \cdot)$. (Again, $\theta$ is a parameter which will enable us to use adaptive methods as described in the next section.) The second component is a Metropolis-Hastings step, in which the proposal is $\alpha$ if the current state is in $E$, and is drawn from a “re-entry proposal distribution” $\phi$ on $E$ if the current state is $\alpha$. It is not difficult to verify that each of these components has invariant distribution $\pi^*_p$ (for some $p \in (0, 1)$), and that the hybrid kernel inherits properties of irreducibility, aperiodicity, etc. from $P_\theta(\cdot, \cdot)$.
Let
\[
\alpha_2(V, W) = \begin{cases} 
\min(1, k\phi(V)), & W = \alpha \\
\min(1, \frac{\pi(W)}{k\phi(W)}), & W \in E.
\end{cases}
\]
where \(k\) is some positive constant. The following algorithm uses this hybrid kernel approach, building around an existing \(\pi\)-invariant kernel \(P_\theta(\cdot, \cdot)\) to obtain a Markov chain with limiting distribution \(\pi^*_p\), where (as in the previous subsection) \(p = k(\beta + k)^{-1}\)

**Algorithm 3.1: Hybrid Metropolis-Hastings**

1. Initialize: Set \(t = 0, Y_0 = \alpha\).
2. Is \(Y_t = \alpha\)?
   - No: Draw \(V\) from \(P_\theta(Y_t, \cdot)\). Set \(W = \alpha\).
   - Yes: Set \(V = \alpha\). Draw \(W\) from \(\phi\).
3. With probability \(\alpha_2(V, W)\), set \(Y_{t+1} = W\). Otherwise set \(Y_{t+1} = V\).
4. Replace \(t\) by \(t + 1\).

In Appendix A, it is proved that, as long as the re-entry proposal distribution \(\phi\) is equivalent to \(\pi\) (that is, \(\phi(A) > 0 \iff \pi(A) > 0\)), the chain \(\{Y_t, t = 0, 1, 2, \ldots\}\) generated by Algorithm 3.1 is ergodic with limiting distribution \(\pi^*_p\). As pointed out to the authors by Luke Tierney, it is also possible to establish the minorization condition directly for the chain obtained using Algorithm 3.1, with the \(\alpha\) states removed. Then the technique of Mykland et al. (1995) identifies the same regeneration times identified using our approach.

As a general rule, it is desirable to choose a re-entry proposal distribution \(\phi\) which is reasonably close to the normalized version of the target distribution \(\pi\), and to choose \(k\) to be roughly of the order of magnitude of \(\phi(x)/\pi(x)\), where \(x\) is some point in a high-density region for \(\pi\). (Recall that we do not assume that the density \(\pi\) is normalized.) Empirical observations suggest that as \(\phi\) becomes less similar to the normalized version of \(\pi\), the coefficient of variation of the distribution of tour lengths \(N_\ell\) tends to increase. Increasing the parameter \(k\) reduces the expected tour length, while decreasing it increases the expected length. In fact, if \(k\phi(x) \geq \pi(x)\) for all \(x \in E\), then the algorithm becomes exactly a rejection-sampling algorithm,
yielding a chain which alternates between $\alpha$ and $E$, in which the non-$\alpha$ values are i.i.d. draws from the target distribution.

In practice, a small segment of a chain generated using the original kernel $P_\theta(\cdot, \cdot)$ can be used to determine $\phi$ and $k$. In many cases, $\phi$ can be taken to be a mixture normal approximation to the distribution of the elements of the initial segment, obtained, for instance, using the method described later in this section. The constant $k$ can then be selected to be approximately the average value of $\pi$, evaluated over the elements of the initial segment (recall that $\pi$ is not normalized), divided by the average value of the density $\phi$, evaluated over a number of draws from $\phi$ itself.

Algorithm 3.1 is particularly useful since it provides a simple means of wrapping up an existing MCMC algorithm for sampling from $\pi$ to yield a Markov chain with limiting distribution $\pi_\theta^\#$. However, it is important to note that Algorithm 3.1 does not necessarily improve mixing of the underlying kernel $P_\theta(\cdot, \cdot)$. If $P_\theta(\cdot, \cdot)$ mixes badly, then the resulting chain $\{Y_t\}$ is also likely to mix badly.

### 3.3 Adaptive Modification of the Kernel

Implementation of MCMC sampling schemes typically requires a significant amount of effort to be devoted to design of a transition kernel in order to ensure that the chain is well-mixing. This design procedure can include the selection of an appropriate parameterization of the quantities of interest, as well as potential parameters of proposal distributions in a Metropolis-Hastings chain. It is tempting to construct algorithms which modify these design parameters automatically as the chain runs. However, even if for each fixed set of parameter values the kernel has the correct limiting distribution, introduction of self-tuning can alter the limiting distribution of the process (which will not necessarily be Markovian any more). Gelfand and Sahu (1994) give an interesting example where this problem occurs. Fortunately, however, there is a simple way around this problem. Gilks et al. (1998) have shown that if the Markov chain is regenerative, then one can modify design parameters at each regeneration time, and estimators are still well-behaved. In particular, Theorems 1 and 2 of Gilks et al. (1998) establish that under this self-tuning scheme, if some technical conditions are satisfied, then (recall the definitions (5) and (7)) $\hat{H}_n$ is MSE-consistent for $\pi_h$, and the quantity

$$n \sum_{j=1}^{n} (H_j - \pi_h N_j)^2 / T_n^2$$

converges in probability to $\sigma^2$. It is relatively straightforward to show that the difference between $\sigma_n^2$ and the expression in (9) converges to zero, and hence that $\sigma_n^2$ is consistent for $\sigma^2$.

Thus in Algorithm 3.1, the kernel parameter $\theta$ can be changed every time the chain hits the state $\alpha$, based on an analysis of the past behaviour of the chain. Even though the process itself is no longer (necessarily) Markovian, it consists of a sequence of tours which, in isolation, are each Markovian, and the estimates $\hat{H}_n$ and $\hat{\sigma}_n^2$ still have the desired consistency properties.

We now state an algorithm, based on the ideas presented in the previous sections, which estimates the expectation (with respect to the probability measure $\pi$) of a “function of interest” $h(\cdot)$, and allows for adaptive modification of a tuning parameter $\theta$. Given a desired number of tours $M > 0$, the following algorithm uses our method to construct a regenerative chain, an estimate $\hat{H}_M$ of $\int h(x) d\pi(x)$, and an estimate of the variance of $\hat{H}_M$ itself.
Intuitively, the algorithm works as follows. It repeatedly generates tours of a Markov chain \( \{Y_t\} \) with limiting distribution \( \pi_\star^p \). Tours of length one (that is, tours which consist only of the state \( \alpha \)) are thrown out. Remaining tours are truncated just before they hit \( \alpha \) so as to become tours of a chain \( \{Z_t\} \) with limiting distribution \( \pi \). The sum over each tour of the function of interest, evaluated at the chain’s state, is recorded, as is the length of the tour. At the end of the \( m \)th tour, the tuning parameter \( \theta_{m+1} \) can be chosen based on analysis of past history of the chain. Once a total of \( M \) tours have been generated, the algorithm terminates, and equations (5) and (7) can be used to obtain the desired estimates.

**Algorithm 3.2:**

1. Initialize: Set \( Y_0 = \alpha, m = 1, t = 0, T_0 = 0 \).
2. Choose some tuning parameter \( \theta_1 \).
3. Draw \( Y_{t+1} \) from \( Q_{\theta_m}(Y_t, d) \).
4. Replace \( t \) by \( t+1 \).
5. Is \( Y_t = \alpha \)?
   - Yes: Set \( n = 1, x = h(Y_t) \).
   - No: Draw \( Y_{t+1} \) from \( Q_{\theta_m}(Y_t, d) \).
7. Replace \( t \) by \( t+1 \).
8. Is \( Y_t = \alpha \)?
   - Yes: Replace \( x \) by \( x + h(Y_t) \).
   - No: Choose \( \theta_m \), based on analysis of \( Y_1, \ldots, Y_t \).
9. Is \( m \leq M \)?
   - Yes: Terminate and compute desired estimates.
   - No: Replace \( m \) by \( m+1 \). Set \( h_m = x, T_m = T_{m-1} + n, N_m = n \). Replace \( m \) by \( m+1 \).

Most of Algorithm 3.2 is easily implemented. The only complicated part, obtaining draws from a kernel \( Q_{\theta_m}(\cdot, \cdot) \) with invariant distribution \( \pi_\star^p \), can be carried out using the basic Metropolis-Hastings Algorithm appropriately modified as described in Section 3.2.1, or Algorithm 3.1. Some consideration should be given also to the method by which the kernel parameter \( \theta_m \) is updated, although in the simplest case, one can construct a non-adaptive chain by fixing \( \theta_m = \theta^* \) for all \( m \).

It might be tempting to use an alternative approach, stopping the procedure at a fixed time and discarding the last incomplete tour. However, this would not be a valid approach. It would introduce bias since the
discarded tour is more likely to be a long tour than a short tour, and different length tours typically have different probabilistic properties.

On a final note, before using the variance estimate computed in Step 9 of Algorithm 3.2, the estimate \( \hat{c}_M \) (recall its definition (8)) of the coefficient of variation should be checked. The variance estimate should not be considered reliable if \( \hat{c}_M \) is too large. Furthermore, it is important to keep in mind that even though Algorithm 3.2 side-steps the burn-in problem in the sense that there is no point in removing any initial portion of the chain, it is still possible that after stopping generation of the chain, not all of the important areas of the state-space have been explored.

### 3.4 An Adaptation Rule

Algorithm 3.2 allows for adaptation through selection (at the end of the \( m \)th tour) of the parameter \( \theta_{m+1} \), which determines the form of the kernel \( Q_{\theta_{m+1}}(\cdot, \cdot) \) used in the \((m+1)\)th tour. In this section we describe one method of doing this when the state space \( \mathbb{E} \) is \( \mathbb{R}^c \), based on the concept that an independence sampler closely matching the target distribution is near-optimal in terms of mixing. (There are, of course, many other possible ways of adapting parameters of a Markov chain; some of these are discussed briefly in Gilks et al., 1998)

Our approach is to start with a kernel \( Q_0(\cdot, \cdot) \), and as time goes by, to transform it progressively into an independence-sampling kernel which uses a normal mixture approximation to the target distribution as its proposal distribution. The mixture approximation itself is updated at the end of each tour, based on the contents of the tour itself.

We use a simple recursive update procedure, described by Titterington (1984), to compute our mixture approximations. The update rule is as follows. Given a multivariate normal mixture

\[
\xi_m \sim \sum_{i=1}^{d} \alpha_i N(\mu_i, \Sigma_i),
\]

for which the parameters \( \alpha_i, \mu_i, \Sigma_i, \) \( i = 1, 2, \ldots, d \) are unknown, and an infinite sequence of draws \( Y_1, Y_2, \ldots \) from \( \xi_m \), parameter estimates \( \hat{\alpha}_i^{(k)}, \hat{\mu}_i^{(k)}, \hat{\Sigma}_i^{(k)} \) can be updated sequentially, once after each draw, by the formulae

\[
\hat{\mu}_i^{(k+1)} \leftarrow \hat{\mu}_i^{(k)} + \frac{w_i^{(k)}}{k \hat{\alpha}_i^{(k)}} (y_{k+1} - \hat{\mu}_i^{(k)})
\]

\[
\hat{\Sigma}_i^{(k+1)} \leftarrow \hat{\Sigma}_i^{(k)} + \frac{w_i^{(k)}}{k \hat{\alpha}_i^{(k)}} [(y_{k+1} - \hat{\mu}_i^{(k)})^T (y_{k+1} - \hat{\mu}_i^{(k)})] - \hat{\Sigma}_i^{(k)}
\]

\[
\hat{\alpha}_i^{(k+1)} \leftarrow \hat{\alpha}_i^{(k)} + k^{-1} (w_i^{(k)} - \hat{\alpha}_i^{(k)})
\]

where \( w_i^{(k)} \propto \hat{\alpha}_i^{(k)} \phi(y_{k+1}; \hat{\mu}_i^{(k)}, \hat{\Sigma}_i^{(k)}) \) with \( \sum_{i=1}^{d} w_i = 1 \), \( \phi(\cdot; \cdot, \cdot) \) denoting the multivariate normal density. As Titterington (1984) points out, there is no guarantee of consistency of these sequentially updated estimators, but for our purposes, since the mixture approximation is to be used to generate proposals, we only require a crude approximation to the target distribution, and consistency is not necessary for our adaptive method. Better methods for constructing mixture approximations have been widely studied.
(see, e.g. McLachlan and Peel, 2000), but the vast majority of these methods cannot be implemented sequentially, which prevents them from being useful in the procedure we propose.

We now state the full adaptive procedure to be used in Algorithm 3.2. Let the adaptive parameters be defined by $\theta_m = (\eta_m, \xi_m)$, where, for each $m$, $\eta_m$ is a constant in the interval $(0, 1)$ and $\xi_m$ is a set of weights, mean vectors, and covariance matrices, defining a normal mixture distribution. We denote the density of the mixture distribution by $\xi_m(\cdot)$.

Let

$$Q_\theta(x, A) = (1 - \eta)Q_0(x, A) + \eta R_\xi(x, A),$$

where $R_\xi(x, A)$ is an independence-sampling Metropolis-Hastings kernel, in which the proposals have density $\xi(\cdot)$, that is,

$$R_\xi(x, A) = I_A(x) \int \xi(y)[1 - a_3(x, y)]dy + \int_A \xi(y)a_3(x, y)dy,$$

with $a_3(x, y) = \min(1, \pi(y)\xi(x)(\pi(x)\xi(y))^{-1})$. Thus the kernel $Q_\theta(\cdot, \cdot)$ is a mixture of the original kernel $Q_0(\cdot, \cdot)$ and the independence-sampling Metropolis-Hastings kernel.

To initialize the adaptive parameters, let $\eta_1 = 0$, and let $\xi_1$ be some initial normal mixture approximation to $\pi$.

At the end of the $m$th tour ($m = 1, 2, \ldots$), set

$$\eta_{m+1} = \max(1.0 - (1.0 - \eta_m) * \kappa, \zeta),$$

where the constants $\kappa$ and $\zeta$ are in the interval $[0, 1]$. In addition, calculate the parameters of $\xi_{m+1}$ by starting with the mixture distribution $\xi_m$ and updating it sequentially using the rules given above, once for each element of the $m$th tour.

The first of these two updates increases the relative contribution of the independence sampler in $Q_{\theta_{m+1}}(\cdot, \cdot)$, to a “maximal” proportional contribution of $\zeta$. Choosing $\zeta = 1$ allows the kernel to be (as time increases to infinity) completely replaced by the independence sampler. Choosing $\zeta < 1$ ensures that some of the original kernel $Q_0(\cdot, \cdot)$ is always retained, and guards against the possibility of building an independence sampler based on the potentially false belief that all important parts of the space have already been explored. The second update simply refines the normal mixture approximation to the target distribution.

Under ideal circumstances, this procedure for adapting $\theta$ will gradually replace the original kernel with an independence sampling Metropolis-Hastings kernel, whose proposal density $\xi_m$ becomes close to the true target density as $m$ increases.

In some cases (particularly when the state-space $E$ is high-dimensional), it can be impractical to construct approximations to the target distribution on the entire state-space. However, if the original kernel $Q_0(\cdot, \cdot)$ consists of Metropolis-Hastings block-updates for subspaces of $E$, it often makes sense to use the same basic idea, but to restrict approximations to the appropriate conditional target densities on the subspaces, and progressively replace the original block-update rules with Metropolis-Hastings independence sampling steps whose proposals are the respective conditional approximations. If conditional target densities are difficult to approximate, a slightly less optimal approach is to use the respective marginal target densities.
Algorithm 3.2 generates $M$ independent tours, possibly updating the kernel parameter $\theta$ after each tour is completed. Hence it lends itself readily to parallel processing implementations.

In the non-adaptive case (i.e. where $\theta_m = \theta_0$, $m = 1, 2, \ldots$), it is clear that tours can be generated on separate processors, and combined to obtain the desired estimates. There is a potential trap, however. Stopping at a fixed time and discarded incomplete tours leads to bias in favor of shorter tours. All currently-running tours must be allowed to run to completion before the algorithm is terminated. In the adaptive case, there is a further complication. Each processor can certainly carry out some kind of adaptive scheme based on only its own generated tours. (It is not hard to see that this is like a single large adaptive scheme in which adaptation at each time point depends only on a subset of previously-observed tours.) However, it is not clear that processors should be allowed to adapt their kernels based on results coming in from other processors, since in this case, a processor would be likely to observe short tours arriving before long tours.

4 Examples

To examine the performance of Algorithm 3.2 we apply to it two problems. As a measure of the mixing quality of Markov chains obtained, we compute the sample precision per iteration (SPPI) of our estimates, which we define to be $(\hat{\sigma}_n^2)^{-1}$, where $n$ is the length of the chain generated and $\hat{\sigma}_n^2$ is the estimate of the variance of the estimator given by (7). Assuming that computational cost to obtain an iteration is roughly invariant, precision per iteration is a direct measure of the computational efficiency of an MCMC simulation. Since this quantity is a random variable itself, we generate multiple realizations of Markov chains, and show box-plots of the resulting SPPI values across the different realizations.

4.1 Dugongs

First we consider a data set used in Ratkowsky (1983), which has also been considered in Carlin and Gelfand (1991). Length ($Y$) and age ($X$) measurements were made of 27 specimens of a particular species of sea cows (dugongs), captured near Townsville, Queensland. The data are shown in Table 1.

<table>
<thead>
<tr>
<th>$X$</th>
<th>1.0</th>
<th>1.5</th>
<th>1.5</th>
<th>1.5</th>
<th>2.5</th>
<th>4.0</th>
<th>5.0</th>
<th>5.0</th>
<th>7.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y$</td>
<td>1.80</td>
<td>1.85</td>
<td>1.87</td>
<td>1.77</td>
<td>2.02</td>
<td>2.27</td>
<td>2.15</td>
<td>2.26</td>
<td>2.35</td>
</tr>
<tr>
<td>$X$</td>
<td>8.0</td>
<td>8.5</td>
<td>9.0</td>
<td>9.5</td>
<td>9.5</td>
<td>10.0</td>
<td>12.0</td>
<td>12.0</td>
<td>13.0</td>
</tr>
<tr>
<td>$Y$</td>
<td>2.47</td>
<td>2.19</td>
<td>2.26</td>
<td>2.40</td>
<td>2.39</td>
<td>2.41</td>
<td>2.50</td>
<td>2.32</td>
<td>2.43</td>
</tr>
<tr>
<td>$X$</td>
<td>13.0</td>
<td>14.5</td>
<td>15.5</td>
<td>15.5</td>
<td>16.5</td>
<td>17.0</td>
<td>22.5</td>
<td>29.0</td>
<td>31.5</td>
</tr>
<tr>
<td>$Y$</td>
<td>2.47</td>
<td>2.56</td>
<td>2.65</td>
<td>2.47</td>
<td>2.64</td>
<td>2.56</td>
<td>2.70</td>
<td>2.72</td>
<td>2.57</td>
</tr>
</tbody>
</table>

Table 1: The dugong data set used in Section 4.1.
As discussed in Ratkowsky (1983), a frequently-used model for such a data set is

\[ Y_i \sim N(\mu_i, \tau^{-1}) \]  
\[ \mu_i = \alpha - \beta \gamma^{X_i} \]

for the data, where \( X_i \) and \( Y_i \) are the age and length of the \( i \)th dugong, respectively, and \( \alpha > 0, \beta > 0, \gamma \in (0, 1) \), and \( \tau > 0 \) are unknown model parameters. We assign the (relatively uninformative) priors

\[ \alpha \sim N(0, 10000) \]
\[ \beta \sim N(0, 10000) \]
\[ \gamma \sim U(0, 1) \]
\[ \tau \sim Ga(0.001, 0.001). \]

Our object is to determine the posterior mean of \( \log(\alpha), \log(\beta), \log(\gamma/(1 - \gamma)), \) and \( \tau^{-1} \).

A Markov chain with the posterior as its limiting distribution can be constructed by computing full-conditionals and updating each of the four parameters in turn. The parameters \( \alpha, \beta \) and \( \tau \) have conjugate priors for the likelihood (11). Hence they can be updated by sampling directly from their respective full-conditionals. The parameter \( \gamma \) does not have a conjugate prior. However, it can be updated by using a single Metropolis-Hastings step whose target density is proportional to the full-conditionals density of \( \gamma \). As our proposal distribution for the \( \gamma \)-update, we choose a uniform distribution on the interval \([0, 1]\).

We simulate using two methods. In the first we simply embed this kernel into Algorithm 3.1 to obtain a regenerative Markov chain, and we use the tours of the resulting chain to compute estimates of the posterior means, along with their (estimated) standard errors. The re-entry proposal distribution \( \phi \) is constructed by running the original chain for 1000 iterations, and using these iterations to construct a normal approximation to the target density, using the update rules described in Section 3.4. The constant \( k \) is chosen so that \( \log(k) \) is equal to the average log-target density over the 1000 iterations, minus the average log-density of \( \phi(\cdot) \) over 1000 draws, minus a constant which is chosen (experimentally) to adjust the distribution of the tour-lengths. After the initial 1000 iterations, the transition kernel is fixed.

Next we use Algorithm 3.2, with the adaptation rule described in Section 3.4. Sampling from the full-conditionals is retained for \( \alpha, \beta \) and \( \tau \), but the uniform proposals for the \( \gamma \)-update are gradually replaced with independent proposals coming from the conditional distribution of \( \gamma \) in a two-component mixture normal approximation to the marginal target density of the full parameter vector. The re-entry proposal distribution and the constant \( k \) are the same as chosen for the non-adaptive method.

Table 2 gives posterior means obtained using these two methods, along with their standard errors, estimated using both the batch-means method and the regenerative method. Figure 1 contains box plots of the SPPI, over 1000 separate regenerative chains, each one consisting of 1000 tours, in both the non-adaptive and adaptive case.

The difference between standard errors estimated using the batch-means method and using the regenerative approach is noticeable, but not very large. In this case, the batch-means method appears to consistently under-estimate the variance.

The improvement in SPPI due to use of the adaptive algorithm is clearly quite substantial. Roughly
Table 2: Posterior means, along with batch-means and regenerative standard error estimates, from (1) a non-adaptive chain (1000 tours, total length 261139) and (2) an adaptive chain (1000 tours, total length 263143).

speaking, it appears that one iteration of the adaptive chain is worth two to five iterations of the non-adaptive chain.

4.2 Monkeys and Free-Knot Splines

We next consider a more complicated problem. Ventura et al. (2002) describe experiments in which a macaque monkey watches images appear on a screen. The monkey is trained to move its eyes in response to certain visual cues, and an electrode measures numbers of “neuron-spikes” occurring in a neuron in the monkey’s supplementary eye field. (The supplementary eye field is thought to be involved in generating eye movements in response to stimuli.)

Figure 2 shows the number of spikes $y_k$ observed in time intervals $[0.03k, 0.03(k+1))$, $k = 0, 1, 2, \ldots, 99$, for one such task.

We assume that the observations $\{y_0, \ldots, y_{99}\}$ are Poisson with a time-varying rate, and can be modelled by

$$Y_k \mid r(\cdot) \sim \text{Pn}(\exp(r(k/99)), \sigma^2), \quad (12)$$

where $r(\cdot)$ is a cubic spline function with four knots, that is,

$$r(x) = \beta_0 + \beta_1x + \beta_2x^2 + \beta_3(x - \xi_1)_+^3 + \beta_4(x - \xi_2)_+^3 + \beta_5(x - \xi_3)_+^3 + \beta_6(x - \xi_4)_+^3, \quad x \in [0, 1],$$

where $(x)_+ = \max(x, 0)$. The knot positions $\xi_1, \ldots, \xi_4 \in [0, 1]$ and the coefficients $\beta_0, \ldots, \beta_6$ are not known. Our object is to determine the posterior distributions of $r(k/99)$, $k = 0, \ldots, 99$, as well as the posterior distribution of $\arg \max_x \ r(x)$, which corresponds to the time at which the firing rate reaches its maximum.

We adopt a Bayesian approach, assigning a Dirichlet$(3, 3, 3, 3, 3)$ prior distribution to the gaps between the knots, that is, to the vector $(\xi_1, \xi_2 - \xi_1, \xi_3 - \xi_2, \xi_4 - \xi_3, 1 - \xi_4)^T$. The coefficients $\beta_j$ are assigned (the relatively uninformative) independent normal priors with mean zero and variance $10^{12}$. The likelihood for the model is easily computed from (12).
Figure 1: Sample precision per iteration (SPPI) for each of the four parameters in the Dugong problem, for both non-adaptive and adaptive chains generated using Algorithm 3.2. SPPIs are evaluated for 1000 non-adaptive and 1000 adaptive chains, each consisting of exactly 1000 tours, with length approximately equal to 250000.
In order to apply Algorithm 3.2, we first construct a transition kernel for a chain whose limiting distribution is the posterior distribution of the parameter vector \((\xi_1, \ldots, \xi_4, \beta_0, \ldots, \beta_6)\). We build a hybrid kernel, which consists of Metropolis-Hastings kernels \(P_1, P_2,\) and \(P_3,\) with proposals which can be summarized as follows. Let \(\hat{\beta}(\xi_1, \ldots, \xi_4)\) denote the maximum likelihood estimate of the coefficient vector \((\beta_0, \ldots, \beta_6)\) given the knot positions \(\xi_1, \ldots, \xi_4,\) and let \(\Sigma_\beta(\xi_1, \ldots, \xi_4)\) denote the estimated covariance matrix of \(\hat{\beta}(\xi_1, \ldots, \xi_4),\) plus a small constant times the identity matrix. (These are straightforward to obtain, since conditioned on knot positions, the model (12) is simply a generalized linear model.) The proposal for \(P_1(\cdot, \cdot)\) is generated by moving one of the knots, chosen at random, a small (Gaussian) distance to the right or left. Conditioned on the proposed knot position \(\xi^* = (\xi^*_1, \ldots, \xi^*_4),\) the proposed coefficients are drawn from a multivariate normal distribution with mean \(\hat{\beta}(\xi^*)\) and covariance matrix \(\Sigma_\beta(\xi^*).\) For \(P_2,\) the proposal is generated by leaving the knots alone and choosing an entirely new set of coefficients, in the same manner as for \(P_1(\cdot, \cdot)\). For \(P_3,\) the (random walk) proposal is generated by leaving the knot positions alone and moving each of the coefficients a random distance proportional to the square root of the corresponding element of the covariance matrix \(\Sigma_\beta(\xi^*).\)

This hybrid kernel consisting of successive application of kernels \(P_1, P_2,\) and \(P_3\) generates a Markov chain on \(\mathbb{R}^{11}\) whose limiting distribution is the posterior distribution of the parameter vector.

As in the previous examples, in order to identify regeneration points for our chain, we embed this hybrid kernel into Algorithm 3.1, and then into Algorithm 3.2.

For this problem, the re-entry proposal distribution \(\phi\) needs to be chosen with care, since the conditional posterior distribution of the coefficients is highly sensitive to the knot positions. Hence for our re-entry proposal distribution \(\phi,\) we draw knot positions from a multivariate mixture normal (conditioning on the event \(0 \leq \xi_1 \leq \xi_2 \leq \xi_3 \leq \xi_4 \leq 1),\) and, conditioned on these knot positions, we draw a coefficient vector.
from a multivariate normal with mean $\hat{\beta}(\xi_1, \ldots, \xi_4)$ and covariance matrix $\Sigma_\beta(\xi_1, \ldots, \xi_4)$ (as done within the kernels $P_1$, $P_2$, and $P_3$). The multivariate mixture normal for knot positions is determined by running the original kernel for 1000 iterations, and constructing a mixture normal approximation to the observed knot positions using the method of Section 3.4. Our mixture approximation has two components, thereby allowing for the possibility that there may be more than one good set of knot positions. Once $\phi$ is determined, the constant $k$ is determined using the same procedure as in the previous example.

To construct an adaptive version of this algorithm, we use the approach described in Section 3.4, constructing a mixture normal approximation to the set of knot positions and gradually replacing the knot-shifting proposals in $P_1$ with independence proposals drawn from the mixture approximation.

Our “function of interest” consists of the 100 fitted values $r(k/99)$, $k = 0, \ldots, 99$, as well as the squares of these values, $\arg\max r(\cdot)$, and $[\arg\max r(\cdot)]^2$. Keeping track of the squares enables us to compute posterior standard deviations for $r(k/99)$, $k = 0, \ldots, 99$ and $\arg\max r(\cdot)$. Figure 3 shows posterior fitted values, and posterior fitted values plus and minus 1.96 times their estimated standard deviations, obtained using an adaptive chain of 1000 tours, with a total length of 30004 iterations. Vertical lines are shown at the estimate of $\arg\max r(\cdot)$ plus and minus 1.96 times its estimated standard deviation.

To measure performance of the adaptive chain relative to the non-adaptive chain, we generate 100 independent chains using each method, each consisting of 1000 tours, with average chain length approximately 30000. The SPPI for $\arg\max r(\cdot)$ is computed for each chain, and box-plots of the results are given in Figure 4.

For this example, the gains in SPPI are again noticeable. In addition, because of the need to repeatedly fit generalized linear models, iterations of the Markov chain are computationally quite intensive, and hence the ability to generate tours in parallel becomes very useful.

5 Discussion

Regeneration in Markov chains is useful since it can be used to avoid the burn-in problem, to obtain good estimates of the variance of MCMC estimators, to introduce indefinite adaptation into a chain, and to use parallel processors to construct a single long chain with a specified target distribution.

We have described a new way to think about identification of regeneration times in a Markov chain and demonstrated potential computational gains, measured by the sample precision per iteration, obtained by using regeneration to introduce adaptive behaviour into MCMC simulation. Of course, improved SPPI through the use of adaptive schemes is not the only reason to use the methods described here. As discussed in the introduction and pointed out by many authors, two major benefits, which come about simply as results of the use of regenerative simulation, are that the issue of burn-in can be ignored and that honest estimates of the variances of estimators can be obtained.

One issue which we have not considered in detail in this paper is the manner in which the chain makes transitions from the state $\alpha$ to the remainder of the state-space, or, in the context of Algorithm 3.2, selection of the re-entry distribution $\phi$ and the constant $k$. In theory, the methods work as long as the chain can jump from $\alpha$ to any set of positive $\pi$-measure with positive probability, but choice of the transition kernel from $\alpha$ affects the tour-length distribution, and the practical consequences of this choice
Figure 3: Posterior mean of rates $\exp(r(t))$, along with lines at fitted values plus/minus 1.96 times the square root of posterior variance of the fitted values. Vertical lines show the estimated 95% posterior interval for $\arg \max r(x)$, rescaled to the $[0, 3]$ time scale of the original data.
Figure 4: Sample precision per iteration (SPPI) for the posterior mean of \( \text{arg max} r(\cdot) \) in the free-knot spline problem, for both non-adaptive and adaptive chains generated using Algorithm 3.2. SPPIs are evaluated for 100 non-adaptive and 100 adaptive chains, each consisting of exactly 1000 tours, with length approximately equal to 30000.

are potentially worth further study.

6 Acknowledgements

The authors are grateful to Brad Carlin, Chris Genovese, Steve Knox, Antonietta Mira, Peter Müller, Geoff Nicholls, Mark Schervish, Mike Steele, Luke Tierney, and Valerie Ventura for their comments and suggestions. This research was supported in part by the National Science Foundation under grants DMS-9819950, DMS-9801401, and IIS-0083148.

A Appendix

In this appendix, we present a formal proof of Theorem 2.1. We also prove that Algorithm 3.1 constructs a Markov chain with limiting distribution \( \pi^*_p \).

We begin by introducing some notation and definitions. \( I_A(\cdot) \) will denote the indicator function, equal to 1 if \( x \in A \), and 0 if \( x \notin A \). We consider discrete-time Markov chains \( \{X_t\} \) with the time index \( t \) assumed to be in the set \( \{0, 1, 2, \ldots\} \). The chains take values either in \( E \) or in an augmented state-space \( E^* = E \cup \{\alpha\} \), where \( d \) is some positive integer and \( \alpha \) is the “artificial atom”. We sometimes abuse notation and refer to the set \( \{\alpha\} \) simply as \( \alpha \). Let \( \mathcal{B}(E) \) denote the family of Borel sets in \( E \), and let \( \mathcal{B}(E^*) \) denote the \( \sigma \)-field generated by all rectangles in \( E \) as well as the point \( \alpha \) (thus \( \mathcal{B}(E^*) \) is an extension of \( \mathcal{B}(E) \) to the space \( E^* \)). When the particular choice of the space (\( E \) or \( E^* \)) and corresponding \( \sigma \)-field (\( \mathcal{B}(E) \) or \( \mathcal{B}(E^*) \)) is context-dependent, we will refer to them simply as \( X \) and \( \mathcal{B} \), respectively. (In principle, the results in this paper also apply when \( E \) is replaced by a more general space. However, for
the sake of simplicity, and since most chains of interest take values in a space which can be mapped to $E$, we do not present explicit versions of our results which apply for more general spaces.)

The transition probability kernel of a Markov chain \( \{X_t, t = 0, 1, 2, \ldots \} \) taking values in $X$ is a function $P(x, A) = \Pr[X_{t+1} \in A | X_t = x]$, $x \in X$, $A \subseteq \mathcal{B}$. The $k$-step transition probability kernels are defined by $P^k(x, A) = \Pr[X_{t+k} \in A | X_t = x]$. The notation $P_x(A)$ is used to denote the conditional probability of the event $A$, given the initial condition $X_0 = x$ for the chain $\{X_t\}$ with transition probability kernel $P(\cdot, \cdot)$.

The chain $\{X_t\}$ is said to be irreducible (or $\phi$-irreducible) if there exists a measure $\phi$ on $\mathcal{B}$ with $\phi(X) > 0$ such that

$$\phi(A) > 0 \Rightarrow \sum_{k=1}^{\infty} P(x, A) > 0 \text{ for all } x \in X. \tag{13}$$

A $\phi$-irreducible chain $\{X_t\}$ is said to have a $v$-cycle if there exist sets $D_1, \ldots, D_v \in \mathcal{B}$ such that $P(x, D_{i+1}) = 1 \forall x \in D_i$, $i = 1, 2, \ldots, v-1$, $P(x, D_1) = 1$ for $x \in D_v$, and $\phi((\cup_{i=1}^{v} D_i)^c) = 0$. If the largest $v$ for which a $v$-cycle occurs is equal to one, then the chain is said to be aperiodic. The first return time for a set $A \in \mathcal{B}$ is

$$\tau_A = \min\{t \geq 1 : X_t \in A\} \tag{14}$$

and the occupation time (after time zero) is defined to be $\eta_A = \sum_{t=1}^{\infty} I_A(X_t)$. A set $A \in \mathcal{B}$ is said to be Harris recurrent if $P_x(\eta_A = \infty) = 1$ for all $x \in A$, and the Markov chain $\{X_t\}$ is said to be Harris recurrent if it is $\phi$-irreducible and every set $A$ with $\phi(A) > 0$ is Harris recurrent. A set $B \in \mathcal{B}$ is said to be uniformly accessible from $A \in \mathcal{B}$ if there exists $\delta > 0$ such that $\inf_{x \in A} P_x(\tau_B < \infty) \geq \delta$.

The chain $\{X_t\}$ is said to be ergodic if there exists a probability measure $\pi$ on $\mathcal{B}$ such that

$$\lim_{n \to \infty} \| P^n(x, \cdot) - \pi(\cdot) \| = 0 \text{ for all } x \in X, \tag{15}$$

where $\| \cdot \|$ is the total variation norm defined for signed measures on $\mathcal{B}$ by $\| \mu \| = \sup_{A \in \mathcal{B}} \mu(A) - \inf_{A \in \mathcal{B}} \mu(A)$. If this is the case, then the probability measure $\pi$ is called the limiting distribution of the chain.

### A.1 Incorporating the Artificial Atom

Recall that the space $E^*$ includes the artificial atom $\alpha$ as well as every element of $E$. As introduced in (1), $\pi^*_p$ denotes the probability measure

$$\pi^*_p(A) = (1 - p)\pi(A \setminus \alpha) + pI_A(\alpha), \quad A \in \mathcal{B}(E^*), \tag{16}$$

where $p$ is some constant in the interval $(0, 1)$. Let $\{Y_t, t = 0, 1, 2, \ldots\}$ be an ergodic Markov chain taking values in $E^*$ with initial value $Y_0 = \alpha$, whose limiting distribution is $\pi^*_p$. Let the hitting times $\tau_Y(j)$ and the tours $Y^j$ be as defined by (2) and (3), respectively. The tour lengths are given by

$$M_j = \tau_Y(j) - \tau_Y(j - 1), \quad j = 1, 2, \ldots, \tag{17}$$

and the elements of a tour $Y^j$ are referred to as $Y^j_k = Y_{\tau_Y(j-1)+k}$, $k = 1, 2, \ldots, M_j$.

The following result shows that $\alpha$ is a proper Harris recurrent atom for the chain. Hence the chain $\{Y_t\}$ is regenerative with regeneration times $\{\tau_Y(j), j = 0, 1, 2, \ldots\}$.
**Theorem A.1** Let \( \{Y_t, \ t = 0, 1, 2, \ldots\} \) be an ergodic Markov chain taking values in \( E^* \) with initial condition \( Y_0 = \alpha \) and limiting distribution \( \pi^*_p \) given by (16). Let the tours and their lengths be as defined in (3) and (17). Then

1. \( \alpha \) is a Harris recurrent state for the chain, and with probability one, the tour lengths \( \{M_j, \ j = 1, 2, \ldots\} \) are all finite, and
2. \( \mathbb{E}[M_j] = p^{-1} \).

**Proof of Theorem A.1:** To prove the first part, note (c.f. (14)) that
\[
P_x(\tau_\alpha < \infty) \geq P^n(x, \alpha) \quad \text{for every } n.
\]
Also, \( P^n(x, \alpha) \to \pi^*_p(\alpha) = p \) as \( n \to \infty \), regardless of \( x \). Thus \( \{\alpha\} \) is uniformly accessible from the state-space \( E^* \). It then follows from Theorem 9.1.3(ii) of Meyn and Tweedie (1993) that
\[
P_x(\eta_\alpha = \infty) = 1
\]
for all \( x \in E^* \), and hence that \( \alpha \) is Harris recurrent and with probability one, the tour lengths are all finite. The second part of the result is a direct consequence of Kac’s Theorem (see Kac, 1947; Meyn and Tweedie, 1993, Theorem 10.2.2).

**A.2 Removing the Artificial Atom**

Next let \( \{Z_t\} \) be the sequence derived from \( \{Y_t\} \) as described in Section 2. Formally, its construction can be described as follows. Let \( s_1 = \min\{i \geq 1 : M_i > 1\} \), and define
\[
s_j = \min\{j > s_{j-1} : M_j > 1\}.
\]
(Thus \( \{s_j\} \) is the set of indices of tours of \( \{Y_t\} \) whose length is strictly greater than one.) Then the tours \( Z^j \) can be written as
\[
Z^j = \{Y^s_{k^j}, \ k = 1, \ldots, M_{s_j} - 1\}, \tag{18}
\]
and their corresponding lengths are
\[
N_j = M_{s_j} - 1, \tag{19}
\]
for \( j = 1, 2, \ldots \). The cumulative tour lengths are then \( T_j = \sum_{i=1}^{j} N_j, \ j = 1, 2, \ldots \), with \( T_0 = 0 \). Finally, \( \{Z_t\} \) is the sequence obtained by concatenating the tours \( Z^1, Z^2, \ldots \) together in sequence. (It is not difficult to verify that the sequence \( \{Z_t\} \) is exactly the sequence \( \{Y_t\} \) with every occurrence of the state \( \alpha \) removed.)

We are now in a position to prove Theorem 2.1.

**Proof of Theorem 2.1:** The proof consists of three main parts. First it is shown that \( \{Z_t\} \) is a Markov chain, and its transition probability kernel is derived. Next it is shown that it has invariant distribution \( \pi \), and finally it is shown that the chain is ergodic, and that the times \( T_j \) are regeneration times.

To see that \( \{Z_t\} \) is a Markov chain, note that \( Z_t = Y_{\tau} \), where \( \tau \) is the stopping time defined to be the \( t \)-th time that \( \{Y_k\} \) is not equal to \( \alpha \). Also define the function \( \gamma_A(\cdot, \cdot, \ldots) \) to be equal to one if its second non-\( \alpha \) argument is in the set \( A \), and zero otherwise. Then
\[
\gamma_A(Y_{\tau}, Y_{\tau+1}, Y_{\tau+2}, \ldots) = I_A(Z_{t+1}), \quad A \in \mathcal{B}.
\]
By the strong Markov property (which every discrete-time Markov chain satisfies),
\[
E[\gamma_A(Y_t, Y_{t+1}, \ldots)|Y_t, Y_{t-1}, \ldots, Y_0] = E_{Y_t}[\gamma_A(Y_0, Y_1, \ldots)].
\] (20)
The term on the right of (20) only depends on \(Y_t = Z_t\). The term on the left is equal to \(\Pr[Z_{t+1} \in A|Z_t, Y_{t-1}, \ldots, Y_0]\). Thus \(Z_{t+1}|Z_t\) is independent of \(Y_{t-1}, Y_{t-2}, \ldots, Y_0\), and since
\[
\{Z_{t-1}, Z_{t-2}, \ldots, Z_0\} \subseteq \{Y_{t-1}, Y_{t-2}, \ldots, Y_0\},
\]

it follows that
\[
\Pr[Z_{t+1} \in A|Z_t, Z_{t-1}, \ldots, Z_0] = \Pr[Z_{t+1} \in A|Z_t],
\]
which establishes that \(\{Z_t\}\) is a Markov chain.

Now let \(P(x, A)\) denote the transition probability kernel of \(\{Y_t\}\) and define
\[
\pi'(A) = \pi(A \setminus \alpha), \quad A \in \mathcal{B}(E^\ast).
\] (21)

Let \(A\) be some set in \(\mathcal{B}(E^\ast)\), and let \(A' = A \setminus \{\alpha\}\). Consider the probability \(Q(x, A)\) that \(Z_{t+1} \in A\) given that \(Z_t = x\). By construction, \(Z_{t+1}\) can never be equal to \(\alpha\), so \(Q(x, \alpha) = 0\) and \(Q(x, A) = Q(x, A')\). Also, \(Z_t\) is equal to \(Y_s\) for some \(s \geq t\). So \(Z_{t+1}\) will be in \(A'\) if and only if \(Y_{s+1} \in A'\), or \(Y_{s+1} = \alpha\) and \(Y_{s+2} \in A'\), or \(Y_{s+1} = Y_{s+2} = \alpha\) and \(Y_{s+3} \in A'\), \ldots. Since these events are mutually exclusive, the transition probability kernel for \(\{Z_t\}\) is
\[
Q(x, A) = Q(x, A') = \Pr[Z_{t+1} \in A'|Z_t = x]
\]
\[
= P(x, A') + P(x, \alpha) \left( \sum_{j=0}^{\infty} P(\alpha, \alpha)^j \right) P(\alpha, A')
\]
\[
= P(x, A') + \frac{P(x, \alpha)P(\alpha, A')}{1 - P(\alpha, \alpha)}.
\] (22)

Now (c.f. (16) and (21)) \(d\pi'(x) = (1 - p)^{-1}[d\pi^\ast_p(x) - p I_\alpha(x)]\). Hence, using (22),
\[
\int_{E^\ast} Q(x, A)d\pi'(x) = \int Q(x, A')d\pi'(x) = \int P(x, A')d\pi'(x) + \int \frac{P(x, \alpha)P(\alpha, A')}{1 - P(\alpha, \alpha)}d\pi'(x)
\]
\[
= (1 - p)^{-1} \int P(x, A')d\pi^\ast_p(x) - p (1 - p)^{-1}P(\alpha, A')
\]
\[
+ \frac{P(\alpha, A')}{1 - P(\alpha, \alpha)} \left[ (1 - p)^{-1} \int P(x, \alpha)d\pi^\ast_p(x) - p (1 - p)^{-1}P(\alpha, \alpha) \right].\] (23)

Since \(\pi^\ast_p\) is the invariant distribution of the chain with kernel \(P(\cdot, \cdot)\), we know that
\[
\int P(x, A)d\pi^\ast_p(x) = \pi^\ast_p(A)
\]
for \(A \in \mathcal{B}(E^\ast)\). Making use of this result, along with the fact that \(\pi^\ast_p(\alpha) = p\), equation (23) becomes
\[
\int_{E^\ast} Q(x, A)d\pi'(x) = (1 - p)^{-1}\pi^\ast_p(A') - p (1 - p)^{-1}P(\alpha, A')
\]
\[
+ \frac{P(\alpha, A')}{1 - P(\alpha, \alpha)}[(1 - p)^{-1}\pi^\ast_p(\alpha) - p (1 - p)^{-1}P(\alpha, \alpha)]
\]
\[
= (1 - p)^{-1}\pi^\ast_p(A') = (1 - p)^{-1}(1 - p)\pi'(A) = \pi'(A),
\]
which applies for all $A \in \mathcal{B}(E^*)$. This means that $\pi'$ is the invariant distribution of the chain $\{Z_t\}$. Since $\{Z_t\}$ can never hit the point $\alpha$, it can also be regarded as a chain taking values in $E$ with invariant distribution $\pi$.

Next it is necessary to show that $\{Z_t\}$ is ergodic. First, note that from (22),

$$Q(x, A) \geq P(x, A) \text{ for any } A \in \mathcal{B}(E) \tag{24}$$

(because $A \in \mathcal{B}(E)$ implies that $A = A'$). Since $\pi(A) > 0 \Rightarrow \pi_p(A) > 0$, and $\pi_p(A) > 0 \Rightarrow \sum_{k=1}^{\infty} P(x, A) > 0$, it follows that $\pi(A) > 0 \Rightarrow \sum_{k=1}^{\infty} Q(x, A) > 0$ for all $A \in \mathcal{B}(E)$ and all $x \in E$. Thus $\{Z_t\}$ is $\pi$-irreducible. Aperiodicity of $\{Z_t\}$ also follows from (24), as does the property that every Harris recurrent set $A$ for $\{Y_t\}$ with $\alpha \notin A$ must be a Harris recurrent set for $\{Z_t\}$. Hence the chain $\{Z_t\}$ is Harris recurrent. Then since $\{Z_t\}$ is irreducible, aperiodic and positive Harris, it follows from the Aperiodic Ergodic Theorem (see, e.g., Meyn and Tweedie, 1993, Theorem 13.0.1) that it is ergodic, and hence that the invariant distribution $\pi$ is also the limiting distribution.

Finally, it follows directly from the strong Markov property that the times $\{T_i, i = 0, 1, 2, \ldots\}$ are regeneration times for the chain $\{Z_t\}$. \hfill $\Box$

### A.3 Analysis of Algorithm 3.1

Next we establish that Algorithm 3.1 does indeed generate a Markov chain with limiting distribution $\pi_p^*$.

**Theorem A.2** Suppose that $\pi$ has density $\beta f(\cdot)$ with respect to some measure $\nu$, where $\beta$ is an unknown constant, and let $\phi$ denote a re-entry density with respect to $\nu$, satisfying the property that

$$\phi(x) > 0 \iff f(x) > 0, \quad x \in E.$$  

Let $P(\cdot, \cdot)$ be the kernel of an ergodic Markov chain with limiting distribution $\pi$, and let $p^*$ be some arbitrary positive constant. Then Algorithm 3.1 generates an ergodic Markov chain with limiting distribution $\pi_p^*$, where $p = \beta p^*(1 + \beta p^*)^{-1}$.

**Proof:** Let $P_1(\cdot, \cdot)$ represent the extension of $P(\cdot, \cdot)$ given by

$$P_1(x, A) = \begin{cases} P(x, A \setminus \alpha), & x \in E \\ I_A(\alpha), & x = \alpha, \end{cases}$$

for $x \in E^*$ and $A \in \mathcal{B}(E^*)$. Let $P_2(\cdot, \cdot)$ be the kernel

$$P_2(x, A) = \begin{cases} I_A(\alpha) \int_E [1 - a(\alpha, y)] \phi(dy) + \int_A a(x, y) \phi(dy), & x = \alpha, \\ I_A(x) [1 - a(x, \alpha)] + I_A(\alpha) a(x, \alpha), & x \in E, \end{cases}$$

where $a(\cdot, \cdot)$ is the function defined in Step 4 of the algorithm. Thus $P_1$ represents the operation carried out in Step 2 of the algorithm; $P_2$ represents the operation carried out in Steps 3 and 4 of the algorithm, and the transition probability kernel for the chain $\{Y_t\}$ generated by Algorithm 3.1 can then be written as

$$Q(x, A) = (P_2 \circ P_1)(x, A) = \int P_1(x, dy) P_2(y, A).$$
It is not too difficult to show that for every \( q \in [0,1] \), \( \pi_p^* \) is an invariant distribution for \( P_1(\cdot, \cdot) \). Also, \( \pi_p^* \) is an invariant distribution for \( P_2(\cdot, \cdot) \). It follows directly that \( \pi_p^* \) is also an invariant distribution for \( Q(\cdot, \cdot) \), and hence that the Markov chain \( \{Y_t\} \) is positive recurrent.

Furthermore, since \( Q(x, \alpha) = \int_{E^*} P_1(x, dy)P_2(y, \alpha) \cdot P_2(y, \alpha) > 0 \) for all \( y \in E^* \), and \( P_1(x, E^*) = 1 \), by basic properties of integrals,

\[
Q(x, \alpha) > 0 \quad \forall x \in E^*.
\]  
(25)

Also, by the Chapman-Kolmogorov equations,

\[
Q^j(x, A) = \int_{E^*} Q^{j-1}(x, dy)Q(y, A) \geq Q^{j-1}(x, \alpha)Q(\alpha, A)
\]  
(26)

for \( j = 2, 3, \ldots \), and since \( P_1(\alpha, A) = I_A(\alpha) \),

\[
Q(\alpha, A) = P_2(\alpha, A).
\]

Since for any \( B \in B(E) \), \( \phi(B) > 0 \Rightarrow P_2(\alpha, B) > 0 \), and \( \phi \) is equivalent to \( \pi \), it follows that \( \pi(B) > 0 \Rightarrow Q(\alpha, B) > 0 \). In conjunction with the property (25) and the definition (1), it follows that

\[
\pi_p^*(A) > 0 \Rightarrow Q(\alpha, A) > 0 \quad \forall A \in B(E^*). 
\]  
(27)

Substituting this result in turn into the inequality (26), with \( j = 2 \), and using (25), we have

\[
\pi_p^*(A) > 0 \Rightarrow Q^2(x, A) > 0 \quad \forall x \in E^*, A \in B(E^*).
\]

An inductive argument using inequality (26) with \( j = 3, 4, \ldots \) shows that, more generally,

\[
\pi_p^*(A) > 0 \Rightarrow Q^j(x, A) > 0 \quad \forall x \in E^*, A \in B(E^*), \quad j = 2, 3, \ldots
\]  
(28)

This establishes that the Markov chain \( \{Y_t\} \) is \( \pi_p^* \)-irreducible and aperiodic. Since it also has invariant distribution \( \pi_p^* \), it is positive, and thus (see, e.g. Meyn and Tweedie, 1993, Theorem 9.0.1) the space \( E^* \) can be decomposed as \( E^* = H \cup N \), where \( \pi_p^*(N) = 0 \), \( N \) is transient, and the chain restricted to \( H \) is Harris recurrent. To show that \( \{Y_t\} \) is Harris recurrent, it suffices to show that \( Q_x(\tau_H < \infty) = 1 \) for all \( x \in N \). Let \( q_n = Q_x(\tau_H > n) \) for \( n = 1, 2, \ldots \), so that

\[
q_1 = \int_H Q(x, dy),\]

\[
q_2 = \int_H \int_H Q(x, dy)Q(y, dz),
\]

and so on. Similarly, let \( p_n = P_x(\tau_H > n) \) for \( n = 1, 2, \ldots \). Then since \( Q(x, A) \leq P(x, A) \) when \( A \subset H \), it follows that \( q_n \leq p_n \) for \( n = 1, 2, \ldots \). Next, since \( P(\cdot, \cdot) \) is assumed to be ergodic with limiting distribution \( \pi \), it is (by definition) Harris recurrent. Furthermore, since \( \pi_p^*(H) = 1 \), \( \pi(H) \) must be positive, so \( \lim_{n \to \infty} p_n = P_x(\tau_H = \infty) = 0 \). Hence \( Q_x(\tau_H = \infty) = \lim_{n \to \infty} q_n \leq \lim_{n \to \infty} p_n = 0 \), so \( \{Y_t\} \) is Harris recurrent.

Finally, since \( \{Y_t\} \) is irreducible, aperiodic, and Harris recurrent with invariant distribution \( \pi_p^* \), it is ergodic with \( \pi_p^* \) as the limiting distribution. \( \square \)
References


