Self Regenerative Markov Chain Monte Carlo with Adaptation

Sujit K. Sahu
Faculty of Mathematical Studies,
University of Southampton,
Highfield, Southampton, UK.

Anatoly A. Zhigljavsky
School of Mathematics,
Cardiff University
Cardiff, UK.

April 8, 2004

Summary

This article proposes a new method of construction of Markov chains with a given stationary distribution. The method is based on constructing an auxiliary chain with some other stationary distribution and picking elements of this auxiliary chain a suitable number of times. The proposed method is easy to implement and analyze; it could be more efficient than some related MCMC techniques. The main attractive feature of the associated Markov chain is that it regenerates whenever it accepts a new proposed point. This makes the algorithm easy to adapt and tune for practical problems. Theoretical study and numerical comparisons with some other available MCMC techniques are made.

Key Words: Adaptive method; Bayesian inference; Independence Sampler; Metropolis–Hastings algorithm; Regeneration.

1 Introduction

Markov Chain Monte Carlo (MCMC) is a key technique for calculating analytically intractable integrals in high dimensions. The propagation and advancement of model based Bayesian statistical inference and prediction have had a symbiotic relationship with the development, analysis and discovery of new computing

algorithms. The spectacular success of the current methodologies is mostly due to their easy programmability and universal applicability.

The techniques of constructing a Markov chain with a given stationary distribution, π, (sometimes called the target distribution) are based on the general methodology commonly known as the Metropolis-Hastings algorithm (Metropolis *et al.*, 1953 and Hastings, 1970). At each iteration of the algorithm, a candidate is generated from a proposal distribution and is accepted with a probability which depends on both the current and the candidate point. This ensures that π is the equilibrium distribution of the Markov chain so constructed. There is a huge literature in this area, see e.g., Gelfand and Smith (1990); Gilks *et al.* (1996); Robert and Casella (1999); Smith and Roberts (1993); Tierney (1994); Chib and Greenberg (1995) and references therein.

In this article we propose and study a new MCMC algorithm which we call self-regenerative (SR). This algorithm belongs to the general family of MCMC algorithms proposed by Hastings (1970). In the algorithm, given a draw from a proposal density, we compute how many times we want to keep the proposed point in the sample. The latter is a draw from the geometric distribution with an appropriate success probability. Once this has been performed, we go on to simulate another independent candidate point from the proposal distribution and iterate.

A regeneration point of a Markov chain is a time where its future becomes independent of the past, see e.g., Ripley (1987), Mykland *et al.* (1995), Robert (1995) and Gilks *et al.* (1998). In the sequel it is shown that the times when SR accepts new candidate points are its regeneration times. Thus, the regeneration times of the SR are identified without any extra work, such as Markov chain splitting (Nummelin, 1984), unlike for the MCMC algorithms based on the regular Metropolis-Hastings version. This regenerative property gives the name to the algorithm and permits a simple analysis of it.

The regenerative property provides a framework for Markov chain adaptation. Gilks *et al.* (1998) propose adaptations at regeneration points of the Markov chain. They obtain theoretical results that such adaptation does not disturb the stationary distribution of the Markov chain and maintains consistency of the sample path averages. Their setup can be used for adapting the SR algorithm. Although there is huge flexibility in devising adaptive methods, a simple (to implement) adaptation scheme is proposed in this paper. The proposal distribution is adapted every time the SR comes across a *trouble* point in the target space where the target and the current proposal distributions have large disparities between them. Theoretical results proved in Section 4.2 show that this adaptation scheme never makes the algorithm infinitely worse. Moreover, substantial improvements are seen in the examples we have looked at.

In the subsequent sections the SR and its adaptive version are compared with other MCMC schemes

including the Gibbs sampler and the slice sampler. The slice sampler has been shown to be uniformly better than the Metropolis-Hastings independence sampler by Mira and Tierney (2001). This article shows that the SR can be more efficient than the slice sampler. The proposed methods are also shown to be better than the Gibbs sampler as implemented in `Winbugs`, a general purpose software for making Bayesian inference (Spiegelhalter *et al.*, 1996).

The plan of the remainder of this article is as follows. In Section 2 the SR algorithm is introduced. Section 3 contains the main theoretical results of this paper. In Section 4 an adaptive methodology for the SR is proposed and some theoretical results are proved. Section 5 illustrates our methods with four examples. A few summary remarks are made in Section 6. Proofs of the main results are given in the Appendix.

2 The SR Algorithm

2.1 Description

Let $(\mathbb{E}, \mathcal{E})$ be a measurable space and $\pi(dx)$ and $\psi(dx)$ be probability measures on it. Let π and ψ have respective densities with respect to a σ-finite measure μ :

$$\pi(dx) = \pi(x)\mu(dx), \quad \psi(dx) = \psi(x)\mu(dx).$$

We assume that $\pi(x)$ is the target density and $\psi(x)$ is the proposal density. To avoid trivialities we assume throughout in this paper that

$$\psi(x) = 0 \text{ for some } x \text{ implies that } \pi(x) = 0. \tag{1}$$

Condition (1) on support of the distributions is typical in MCMC simulation.

In typical problems where MCMC is used, the normalizing constant of the target distribution $\pi(x)$ is not known. We also assume that the normalizing constant of $\psi(x)$ is unknown. Let $\tilde{\pi}(x)$ and $\tilde{\psi}(x)$ denote non-normalized versions of the densities. Let $w(x) = \pi(x)/\psi(x)$ and $\tilde{w}(x) = \tilde{\pi}(x)/\tilde{\psi}(x)$. If the normalizing constant of $\psi(x)$ is known, then we simply set $\tilde{\psi}(x) = \psi(x)$.

Let $c > 0$ be a constant such that $c\,\tilde{w}(x) = w(x)$ and κ be an arbitrary positive constant. Define the function

$$\alpha(x) = \frac{1}{1 + \kappa\, w(x)} = \frac{1}{1 + \kappa\, c\, \tilde{w}(x)}, \tag{2}$$

which we shall call the rejection function.

Let N be the total number of samples to be drawn from the proposal distribution $\psi(x)$. The main algorithm is as follows. Assume that $n = 1$ and $m = 0$.

Algorithm 1: (Self Regenerative MCMC.)

STEP I: Generate $Z_n \sim \psi$.

STEP II: Generate $U \sim$ Uniform(0, 1).

STEP III: If $U \leq 1 - \alpha(Z_n)$, set $X_{m+1} = Z_n$ and return to Step II with $m = m + 1$.

STEP IV: If $n = N$ then stop, else set $n = n + 1$ and return to Step I.

In Section 3 we analyze this algorithm theoretically; we show, in particular, that the associated Markov chain has $\pi(x)$ as its stationary density.

Observe that for fixed n and Z_n, Steps II and III of the above algorithm simulate Bernoulli trials with success probability $\alpha(Z_n)$ until the first success. Hence the two steps can be coalesced. Let $\xi \sim G(p)$ denote a geometric random variable with success probability p and probability mass function $Pr(\xi = i) = (1-p)^i p$ for $i = 0, 1, \ldots$ Now Algorithm 1 can be expressed in the following simpler but probabilistically equivalent form.

Let $n = 1$ and $m = 0$.

Algorithm 2:

STEP I: Generate $Z_n \sim \psi$.

STEP II: Generate $\xi_n \sim G(\alpha(Z_n))$. If $\xi_n > 0$, set $X_{m+j} = Z_n$ for $j = 1, \ldots, \xi_n$ and $m = m + \xi_n$.

STEP III: If $n = N$ then stop, else set $n = n + 1$ and return to Step I.

Let M denote the final value of m, that is,

$$M = \sum_{n=1}^{N} \xi_n. \tag{3}$$

Thus, M is the total number of MCMC samples drawn from the target distribution after simulating N samples from the proposal distribution. Clearly, M is a random variable. The stopping criteria in the above algorithms can be modified if it is desired to keep M, the total number of MCMC samples, fixed. In this case N becomes a random variable. We shall investigate the asymptotic relationship between M and N in Section 3.3.

2.2 Choice of κ

To implement the SR we need to specify the product of the positive constants κ and c in (2). There are accurate methods available to estimate c, see for example Chib (1995), Chib and Jeliazkov (2001) and the references therein. In our case, however, **we do not need to know the constant c exactly** since the implementation of SR requires only the product κc and X_m is shown to be a Markov chain with π as its unique stationary distribution for any $\kappa > 0$. A rough estimate of c can be obtained by using the following guideline.

We have

$$E_\psi \frac{\tilde{\pi}(Z)}{\tilde{\psi}(Z)} = \frac{1}{c}.$$

With B draws from ψ, a Monte Carlo estimate $\hat{c}$ of c is given by

$$\frac{1}{\hat{c}} = \frac{1}{B}\sum_{i=1}^{B} \frac{\tilde{\pi}(Z_i)}{\tilde{\psi}(Z_i)}. \tag{4}$$

Hence we can set c^{-1} to be this estimate. Also this can be automated, i.e. this can be performed before running the algorithm. Although the $\hat{c}$ defined in (4) only provides a rough estimate of c, it suffices to run the algorithm with this estimate since the algorithm is shown to work for any positive value of κ. An inaccurate estimate of c will only imply a different value of κ in (2).

In Section 3.5 we shall see that the efficiency of the SR is an increasing function of κ. If κ is chosen to be very large, then the rejection probability $\alpha(x)$ in equation (2) will be very small. As a result, the accepted proposals will be repeated many times, see step III of `Algorithm 1`. This will increase the auto-correlation of the SR chain. If κ is chosen to be very small, then $\alpha(x)$ will be large and many proposed moves will be rejected. In practice we set a value of κ which does not lead to either too many rejections or too many repetitions of the same candidate point. This is the only tuning needed to implement the algorithm, although we remark that κ can be adapted during the course of simulation, see Section 4.

3 Analysis

3.1 Basic Properties

The sequence X_m in Algorithm 1 forms a Markov chain by definition. To discuss convergence properties of X_m we introduce the following notations. Let $w^* = \sup_x w(x)$ and $\delta_x(y)$ denote the point mass at x. We also define the density

$$\phi(y) = \frac{\{1-\alpha(y)\}\psi(y)}{\int\{1-\alpha(z)\}\psi(z)\mu(dz)}. \tag{5}$$

Finally recall the definition of the total variation norm of a bounded sign measure λ on $(\mathbb{E}, \mathcal{E})$:

$$||\lambda|| = \sup_{A \in \mathcal{E}} \lambda(A) - \inf_{A \in \mathcal{E}} \lambda(A).$$

Theorem 1 *Assume that the support condition (1) holds. Then:*

(i) The sequence $X_1, X_2, \ldots$ form a Markov chain with the transition kernel

$$K(x, dy) = \Big[\alpha(x)\phi(y) + \{1 - \alpha(x)\}\delta_x(y)\Big]\mu(dy) \tag{6}$$

with π being an invariant distribution of K; that is,

$$\int \pi(dx)K(x, dy) = \pi(dy). \tag{7}$$

(ii) The transition kernel $K(x, dy)$ is reversible, π-irreducible and aperiodic. It is also ergodic, i.e. for every $x \in \mathbb{E}$

$$||K^m(x, \cdot) - \pi(\cdot)|| \to 0 \text{ as } m \to \infty. \tag{8}$$

Corollary 1 *Under the conditions of Theorem 1, the pair $\{\alpha(x), \phi(dy)\}$ provides a Nummelin splitting of the transition kernel $K(x, dy)$.*

Using Corollary 1 we see that every time a new candidate y is accepted is a regeneration time. See e.g. Nummelin (1984) and Gilks *et al.* (1998) for more on splitting and regeneration.

With a further assumption on π and ψ the SR achieves a geometric rate of convergence in (8).

Theorem 2 *If $w^* = \sup_x w(x) < \infty$, then Theorem 1 holds and $K(x, dy)$ is uniformly ergodic.*

Hence the SR can be used in MCMC simulation. Moreover, we have the following results.

Corollary 2 *Under the conditions of Theorem 2, $\beta = \frac{1}{1+\kappa w^*}$ is an upper bound on the rate of convergence of the SR.*

3.2 SR and the Metropolis-Hastings Algorithm

The Hastings (1970) algorithm generates a candidate point y from the proposal distribution $q(y|x)$ (where x is the current point) and accepts it with probability

$$\alpha^H(x, y) = \frac{s(x, y)}{1 + t^H(x, y)}$$

where $s(x,y)$ is a symmetric function of x and y such that $0 \leq \alpha^H(x,y) \leq 1$ and

$$t^H(x,y) = \frac{\pi(x)\, q(y|x)}{\pi(y)\, q(x|y)}. \tag{9}$$

The algorithm is also valid without the symmetry requirement on $s(x,y)$ if the resulting transtion kernel satisfies an appropriate invariant condition like (7). When the proposal distribution is taken to be independent of the past we assume $q(y|x) = q(y)$ and $t^H(x,y)$ in (9) simplifies to

$$t(x,y) = \frac{\pi(x)\, q(y)}{\pi(y)\, q(x)}. \tag{10}$$

The support condition (1) guarantees that $\phi(y)$ is a proper probability distribution. Now we choose the proposal distribution $q(y)$ of the Hastings algorithm to be $\phi(y)$ and

$$s(x,y) \equiv s^{SR}(x,y) = \alpha(x)\left\{1 + t(x,y)\right\}, \tag{11}$$

where $\alpha(x)$ is given in equation (2). Note that in this case $s(x,y)$ is not symmetric. Now we have

$$\alpha^H(x,y) = \alpha(x).$$

The resulting transition kernel, $K(x,dy)$ given in (6), is now seen to be the transition kernel of the Hastings type algorithm which generates independent proposals from $\phi(y)$ and accepts with probability $\alpha(x)$. Thus we have proved the following result.

Lemma 1 *Under the support condition (1), $K(x,dy)$ has the form of the transtion kernel of the Hastings algorithm with $\phi(y)$ as the proposal density.*

The Metropolis-Hastings sampler with $\phi(y)$ as the independent proposal distribution chooses

$$s(x,y) \equiv s^{MH}(x,y) = 1 + \min\{t(x,y),\, t(y,x)\}$$

see for example, Hastings (1970) and Peskun (1973). Clearly, $s^{MH}(x,y) \neq s^{SR}(x,y)$ for all x and y which proves that SR is not Metropolis-Hastings corresponding to the proposal density $\phi(y)$. Similarly, it is straightforward to show that the SR is not Metropolis-Hastings corresponding to the proposal density $\psi(y)$ as well as any other proposal density. Thus we have established that *the SR algorithm is not a special case of the Metropolis-Hastings algorithm.*

3.3 Sample Size

If N independent draws from ψ are performed, then we obtain M MCMC samples from π, where M is a random variable. M is the sum of N independent geometric random variables, ξ_n, with random success

probabilities $\alpha(Z_n)$, $Z_n \sim \psi$, see equation (3). The following result is an easy consequence of this hierarchical setup.

Lemma 2 *Let $\xi \sim G(\alpha(Z))$ where $Z \sim \psi$. Then:*

$$E(\xi) = \kappa \quad \text{and} \quad Var(\xi) \equiv \tau_\xi^2 = \kappa + \kappa^2 \left\{2E_\pi(w(X)) - 1\right\}. \tag{12}$$

Note that ξ_n in (3) are i.i.d. with the above mean and variance. Hence we have the following result.

Theorem 3 *Assume that τ_ξ^2 is finite. Then*

$$\frac{1}{\sqrt{N}}(M - N\kappa) \Rightarrow N(0, \tau_\xi^2) \quad as \quad N \to \infty,$$

where "$\Rightarrow$" denotes convergence in distribution and $N(a, \sigma^2)$ denotes the normal distribution with mean a and variance σ^2.

In order to study the M-step transitions $K^M(x, dy)$ and compare the rate of convergence in (8) with other MCMC techniques we modify the stopping criterion of Algorithm 1 and 2 to have M fixed. Then N becomes a random regeneration moment where

$$N = \inf\{n : \sum_{i=1}^{n} \xi_i \geq M\}. \tag{13}$$

We thus have an analogue of Theorem 3.

Theorem 4 *Assume that τ_ξ^2 is finite and let the random variable N be as defined in (13). Then*

$$\frac{1}{\sqrt{M}}(M - N\kappa) \Rightarrow N\left(0, \frac{\tau_\xi^2}{\kappa}\right) \quad as \quad M \to \infty.$$

Theorems 3 and 4 provide qualitative descriptions of the asymptotic relation $M \simeq N\kappa$ which we use later on in this paper.

3.4 Rate of Convergence

The rate of convergence of a Markov chain together with the starting value determines the number of iterations needed for convergence to the stationarity. Let $Q(x, dy)$ denote the transition kernel with unique equilibrium distribution $\pi(x)$ and T be a given integer. The rate of convergence, denoted by ρ, of such an algorithm is the minimum number $\eta \in [0, 1]$ such that

$$||Q^t(x, \cdot) - \pi(\cdot)|| \leq V(x)\eta^t$$

for a suitable function V and for all $t > T$. The requirement that $||\cdot||^T < \epsilon$ for some small ϵ dictates that the algorithm should be run a suitable multiple of $(\log \rho)^{-1}$ number of iterations in order to achieve the given accuracy ϵ, where the multiplier depends on ϵ and the function $V(x)$, which in turn depends on the starting point x.

Suppose that we look at the M step transition kernel K^M of the SR. Assume that $w(x)$ is bounded, i.e., $w(x) \leq w^* < \infty$ for all x. Then Theorem 2 holds and we have

$$||K^M(x,\cdot) - \pi(\cdot)|| \leq V(x)[1-\beta]^M$$

where $1 - \beta = \frac{\kappa w^*}{1+\kappa w^*}$. Since to perform M iterations of K we need N evaluations of π, it is natural to express the rate of convergence as a function of N rather than M. Using the asymptotic relation $M \simeq N\kappa$ we have

$$||K^M(x,\cdot) - \pi(\cdot)|| \leq V(x)[1-\beta]^M \simeq V(x)[1-\beta]^{N\kappa} = V(x)\left[(1-\beta)^\kappa\right]^N.$$

Hence

$$\rho_{\mathrm{SR}} \leq (1-\beta)^\kappa = \left(\frac{\kappa w^*}{1+\kappa w^*}\right)^\kappa \leq 1 - \frac{1}{w^*} + \frac{1+\kappa}{2\kappa}\frac{1}{(w^*)^2}, \tag{14}$$

where ρ_{SR} denotes the rate of convergence of the SR.

Note that the right hand side of (14) is a decreasing function of κ. However, (14) also suggests that it is more important to control w^* in the SR for faster convergence. Section 4 proposes an adaptive version of the SR to address this issue.

Lastly, the SR being a reversible Markov chain defines a positive definite operator on $L^2(\pi)$, the space of square integrable functions with respect to $\pi(x)$, as do the slice sampler and other Metropolis-Hastings schemes. Thus, for all these algorithms the characteristics related to the speed of convergence to stationarity and to efficiency are equivalent. Next we discuss the results concerning the efficiency in estimating the integrals.

3.5 Estimation of Integrals and Efficiency

One of the primary tasks of MCMC simulation is to estimate arbitrary (but finite) integrals with respect to the density π by forming ergodic averages. Let f be a real valued function and

$$I_f = \int f(x)\pi(x)\mu(dx) \quad \text{and} \quad \sigma_f^2 = \int f(x)^2\pi(x)\mu(dx) - I_f^2.$$

To estimate I_f we form the sample average

$$\bar{f}_N = \frac{1}{M}\sum_{j=1}^{M} f(X_j) = \frac{1}{M}\sum_{i=1}^{N}\xi_i f(Z_i) = \frac{\sum_{i=1}^N \xi_i f(Z_i)}{\sum_{i=1}^N \xi_i}. \tag{15}$$

After N steps of Algorithm 2, π is evaluated N times and the estimator $\bar{f}_N$ is based on N i.i.d. draws from the proposal distribution ψ. Hence we discuss the limiting behavior of $\bar{f}_N$ as $N \to \infty$.

Let $L^p(\pi)$ be the set of real valued functions f such that $\int |f(x)|^p \pi(x)\mu(dx) < \infty$ and $\tilde{f}(x) = f(x) - I_f$.

Theorem 5 *If (1) holds, $f \in L^1(\pi)$ and $N \to \infty$, then:*

(i) $\bar{f}_N - I_f \xrightarrow{P} 0$, *where "$\xrightarrow{P}$" denotes convergence in probability.*

(ii) *Suppose also that $f \in L^2(\pi)$ and*

$$\sigma^2 = \sigma^2(\kappa, \pi, \psi, f) = \sigma_f^2/\kappa + 2\int \tilde{f}(x)^2 w(x)\pi(x)\mu(dx) \tag{16}$$

is finite, then

$$\sqrt{N}(\bar{f}_N - I_f) \Rightarrow N(0, \sigma^2).$$

If $w^* < \infty$, then σ^2, the asymptotic variance of $\sqrt{N}\bar{f}_N$, is finite. Moreover, σ^2 can be finite even when $w^* = \infty$; two simple examples are: $w \in L^1(\pi)$ and f is bounded; $w \in L^2(\pi)$ and $f \in L^4(\pi)$.

The expression for σ^2 in (16) is the sum of two positive components. The first component σ_f^2/κ is N times the variance of the standard Monte Carlo estimator of I_f based on $M(\approx N\,\kappa)$ i.i.d. samples from $\pi(x)$. Therefore, the second component in (16) can be thought of as a penalty for not being able to obtain i.i.d. samples from $\pi(x)$.

Another way to understand the meaning of the second component in (16) is to consider the importance sampling estimator of I_f,

$$\hat{f}_N = \frac{1}{N}\sum_{i=1}^{N} f(Z_i)\frac{\pi(Z_i)}{\psi(Z_i)}, \quad Z_i \sim \psi(z) \text{ i.i.d.}$$

It is straightforward that

$$N\mathrm{Var}(\hat{f}_N) = \int \tilde{f}(x)^2 w(x)\pi(x)\mu(dx),$$

which is exactly the second term in (16) without the factor 2. Of course, it is more efficient to use the importance sampling alone but there is no Markov chain associated with this scheme. However, being in the MCMC framework we have serious advantages over the importance sampling, e.g. regeneration and adaptation. Moreover, the SR can be used to update components in Gibbs sampling which is often used to simulate from high dimensional distributions where the importance sampling fails.

Observe that the asymptotic variance σ^2 is a decreasing function of κ. Hence SR will be more efficient for larger values of κ. See Section 2.2 for more insights on tuning κ.

We remark that we have the above closed form expression (16) for the asymptotic variance of the estimator $\bar{f}_N$. In general, for the MCMC samplers such simple expressions are not available. The variance of the estimator is usually written as a suitable multiple of the integrated auto-correlation time of the process $\{f(X_m)\}$, see e.g., Geyer (1992) and Green and Han (1992). The integrated autocorrelation time is defined to be

$$\zeta = 1 + 2\sum_{k=1}^{\infty} \rho_k \tag{17}$$

where ρ_k is the lag-k autocorrelation of $\{f(X_m)\}$.

In order to compare SR with other MCMC algorithms we recall the definition of *asymptotic efficiency*: (see e.g., Tierney, 1994)

$$\text{Efficiency}(f) = \sigma_f^2 / \lim_{N\to\infty} N\text{Var}(\bar{f}_N), \tag{18}$$

for any given function $f(x) \in L^2(\pi)$. When the state space is discrete with d elements, say, then the asymptotic efficiency can be expressed as (see e.g., Peskun, 1973)

$$\text{Efficiency}(f) = \sigma_f^2 / \mathbf{f}^T \big(2BR - B - BA\big)\mathbf{f} \tag{19}$$

where $\mathbf{f}^T = (f_1, \ldots, f_d)$ is the function, f, vector evaluated at the states; $A = \mathbf{1}\boldsymbol{\pi}^T$ where $\boldsymbol{\pi}$ is the vector of probability mass points and $\mathbf{1}$ is the unit vector; B is the diagonal matrix with diagonal elements $\boldsymbol{\pi}$ and $R = \{I - (P - A)\}^{-1}$ if this inverse exists, where P is the Markov transition matrix.

If the proposal distribution $\psi(x)$ is the same as the target distribution $\pi(x)$, then the Metropolis-Hastings independence sampler (MHIS) is uniformly better than the SR. In this case MHIS generates independent samples from $\pi(x)$ with efficiency 1; the efficiency of the SR calculated using (16) and (18) is $1/(2 + \kappa^{-1})$.

On the other hand, the SR with $\kappa \geq 1$ can outperform many Metropolis-Hastings schemes when there is large disagreement between the proposal and the target distributions; the situations, where MCMC is needed. We give several examples here and in Section 5 to illustrate this.

Let $\pi(x) \propto x^{a-1}(1-x)^{b-1}$ for $0 < x < 1$, $a > 0$ and $b > 0$. Let $\psi(x) = 1$, $0 < x < 1$. Assume that $a = b = 3/4$, $f(x) = x$ and $\kappa = 1$. Here it is easy to see that $w^* = \infty$. The asymptotic variance of the estimator $\sqrt{N}\bar{f}_N$ for the SR has, however, a finite value (equal to 0.5635). On the other hand, the asymptotic variance of the estimator $\sqrt{N}\bar{f}_N$ for the MHIS is infinite. (This follows from the fact that a lower bound on the variance of the MHIS is a suitable multiple of w^*, see e.g., Liu, 1996.) Hence, in this example the asymptotic efficiency of the MHIS is zero while it is finite for the SR.

For complex target distributions it is often hard to use (18) for calculating the efficiency. In such situations it is suggested, see for example Kass *et al.* (1998), that we use the effective sample size (ESS) to compare the algorithms. The ESS is defined to be the number of MCMC samples drawn divided by the integrated

autocorrelation time, ζ. The estimation of ESS using the batch-means method is discussed for example by Roberts (1996). We use the ESS to compare the SR with other algorithms in Section 5.

4 Regeneration and Adaptation

4.1 The Adaptive SR Algorithm

The performance of the SR algorithm depends on the quantity $w^* = \sup_x w(x)$. The SR will exhibit slow mixing when w^* is very high or equivalently the $\alpha^* = \inf_x \alpha(x)$ is too small. For very small values of $\alpha(x)$ the SR will keep accepting the same proposal x in Step III of `Algorithm 1` for a long time. The adaptations are aimed at addressing these problems. We propose adaptations to occur when $\alpha(x)$ is less than a threshold value, $\bar{\alpha}$ say, to be specified by the user. As noted by one of the referees, it can also be worthwhile to adapt the SR when $\sup_x \alpha(x)$ is too close to 1 (a natural advice in this case is to increase κ). The adaptations make the chain non-Markovian, as a result the ergodic theorems are no longer guaranteed to hold. In our regenerative setup, however, the ergodic averages do converge appropriately, see Section 4.2 where we address these issues using theoretical results from Gilks *et al.* (1998).

We introduce the following notations to describe the adaptations. Let ψ_0 be the initial proposal density and ψ_k be the current proposal density after $k-1$ adaptations. Let

$$w_k(x) = \frac{\pi(x)}{\psi_k(x)}, \quad \text{and} \quad \alpha_k(x) = \frac{1}{1 + \kappa_k \, c_k \, \tilde{w}_k(x)}$$

where $\kappa_k > 0$ and c_k is such that $c_k \tilde{w}_k(x) = w_k(x)$. Let $\epsilon_k \in (0,1)$, $k \geq 1$ be a sequence of numbers such that $\bar{\epsilon} = \sum_k \epsilon_k < \infty$. Furthermore, let $\chi_k(x|y)$ for $k \geq 1$ be a density centered around the point y.

Our adaptation scheme is as follows. After the first $k-1$ adaptations, we iterate with the current proposal density ψ_{k-1} until we come across a candidate point z, labelled $x_{(k)}$, such that

$$\alpha_{k-1}(z) \leq \bar{\alpha}.$$

Then we update the proposal density ψ_{k-1} as follows:

$$\psi_k(x) = (1 - \epsilon_k) \, \psi_{k-1}(x) + \epsilon_k \, \chi_k(x|x_{(k)}). \tag{20}$$

Henceforth we call this the adaptive self regenerative (ASR) algorithm. Its pseudo-code is as below.

Let $n = 1$, $m = 0$ and $k = 1$.

`STEP I:` `Generate` $Z_n \sim \psi_{k-1}$.

STEP II: Calculate $\alpha_{k-1}(Z_n)$. If $\alpha_{k-1}(Z_n) < \bar{\alpha}$ then do Step III else go to Step IV.

STEP III: Set $x_{(k)} = Z_n$. Construct $\chi_k(x|x_{(k)})$. Obtain $\psi_k(x)$ using equation (20). Set $k = k+1$ and go to Step V.

STEP IV: Generate $\xi_n \sim G(\alpha_{k-1}(Z_n))$. If $\xi_n > 0$ set $X_{m+j} = Z_n$, for $j = 1, \ldots, \xi_n$ and $m = m + \xi_n$.

STEP V: If $n = N$ then stop, else set $n = n+1$ and return to Step I.

There is much flexibility in choosing χ_k and, as usual, the better choices are problem dependent. One heuristic is to choose χ_k such that it matches the target density in a neighborhood of $x_{(k)}$, for example by setting $\chi(x|x_{(k)})$ a normal density with mean $x_{(k)}$ and covariance matrix Σ_k, where Σ_k is the estimated covariance matrix from the samples obtained so far.

There are many possible alternatives for the sequence $\{\epsilon_k\}$. For example, a particular choice can be

$$\epsilon_k = a\,\frac{1}{k^2} \quad \text{with } a = \frac{6}{\pi^2}, \tag{21}$$

so that $\sum_{k=1}^{\infty} \epsilon_k = 1$. In Section 5 we adopt these particular values and take $\bar{\alpha} = 0.01$ always. However, these can be varied in different implementations if desired.

If the densities π and ψ are non-normalized, then the ratio of the normalizing constants c_k may be adapted as well. However, in the practical examples in Section 5 we take $\kappa_k = \kappa$ and $c_k = \hat{c}$ for all k, primarily because these worked reasonably well.

Implementation of the ASR is straightforward. The following further details can be used in the computation. For any $k \geq 1$ we have,

$$\psi_k(x) = \left[\prod_{j=1}^{k}(1-\epsilon_j)\right]\psi_0(x) + \left[\prod_{j=2}^{k}(1-\epsilon_j)\right]\epsilon_1\,\chi_1(x|x_{(1)}) + \cdots + \epsilon_k\,\chi_k(x|x_{(k)}).$$

Observe that the coefficients in the right hand side of the above add to 1. For the ϵ_k's given in equation (21) the coefficients can be obtained using

$$\prod_{j=m+1}^{k}\left\{1-\frac{a^2}{j^2}\right\} = \frac{\Gamma(k+1-a)\,\Gamma(k+1+a)\,\Gamma^2(m+1)}{\Gamma(m+1-a)\,\Gamma(m+1+a)\,\Gamma^2(k+1)},\ m = 0, 1, \ldots, k-1,$$

where $\Gamma(\cdot)$ is the standard gamma function. Hence, in order to sample from ψ_k we can draw a candidate point from the mixture density.

Suppose that the total number of adaptations is K. In these K adaptations, for notational convenience, we also include the initial regime where the proposals were generated using ψ_0. Let N_1, $N_2, \ldots, N_K$ be the

number of realizations from the corresponding proposal densities. Let $N = \sum_{k=1}^{K} N_k$. Also let $\bar{f}_{N_k}$ be the estimator (15) for I_f in the kth adaptive regime. Let σ_k^2/N_k be the asymptotic variance of $\bar{f}_{N_k}$ (following equation (16)) where

$$\sigma_k^2 = \sigma^2(\kappa_k, \pi, \psi_k, f) = \sigma_f^2/\kappa_k + 2\int \tilde{f}(x)^2 w_k(x)\pi(x)dx. \tag{22}$$

The final estimator of I_f from the ASR is

$$\bar{f}_N = \frac{\sum_{k=1}^{K} N_k \bar{f}_{N_k}}{N}. \tag{23}$$

Using the fact that the ASR is adapted at the regeneration points, we obtain that the asymptotic variance of $\bar{f}_N$ is

$$\mathrm{Var}(\bar{f}_N) = \frac{1}{N^2}\sum_{k=1}^{K} N_k \sigma_k^2.$$

We can decide whether to discard observations (for estimating I_f) from the past tours using the above expression for the variance. We illustrate this when $K = 2$. (The reader will see immediate extension to the general case.) We have $\bar{f}_N = \frac{N_1\bar{f}_{N_1} + N_2\bar{f}_{N_2}}{N_1+N_2}$. The question we ask is whether we shall use $\bar{f}_N$ or $\bar{f}_{N_2}$ as our estimator of I_f. We should use $\bar{f}_N$ instead of $\bar{f}_{N_2}$ if and only if

$$\mathrm{Var}(\bar{f}_N) < \mathrm{Var}(\bar{f}_{N_2}) \Leftrightarrow \ \frac{N_1}{N^2}\sigma_1^2 + \frac{N_2}{N^2}\sigma_2^2 < \frac{\sigma_2^2}{N_2} \Leftrightarrow \ \sigma_1^2 < \left(2 + \frac{N_1}{N_2}\right)\sigma_2^2.$$

In practice, we can estimate the two variances σ_k^2 from the output and decide which estimator to use. Intuitively, if adaptation fails to provide a great improvement over the starting proposal density then we shall include observations generated using the starting proposal distribution, otherwise we shall not.

4.2 Theoretical Results

It is clear that the stochastic process induced by the ASR is no longer Markov. However, the estimator $\bar{f}_N$ in (23) can still be used to estimate I_f. This follows from the central limit theorem for $\bar{f}_N$ proved in Gilks *et al.* (1998). This paper shows that if adaptations are performed at the regeneration points, then $\bar{f}_N$ formed by using the output of the adaptive process, converges to I_f under some regularity conditions. The main condition to check for the theorem in Gilks *et al.* (1998) to hold is the convergence of σ_k^2 to a limiting value. This follows from (20) and the theorem on convergence of supermartingales. Indeed, assume that $\epsilon_k < \frac{1}{2}$ and $\{\kappa_k\}$ is a non-decreasing sequence (that is, $\kappa_{k+1} \geq \kappa_k$ for all k). Then (20) and (22) give

$$E\left(\sigma_{k+1}^2 | \sigma_0^2, \ldots, \sigma_k^2\right) = \frac{\sigma_f^2}{\kappa_{k+1}} + 2\int \tilde{f}(x)^2 w_{k+1}(x)\pi(x)dx \ \leq \frac{\sigma_f^2}{\kappa_k} + \frac{2}{1-\epsilon_k}\int \tilde{f}(x)^2 w_k(x)\pi(x)dx \ \leq (1+2\epsilon_k)\sigma_k^2.$$

Since $\sum_k \epsilon_k < \infty$ the convergence of $\{\sigma_k^2\}$ follows. If c_k is adapted as well, the additional requirement for these arguments to hold is to adapt κ_k so that $c_k\kappa_k$ form a non-decreasing sequence.

Adaptations, however, can make MCMC algorithms converge slower than their non-adaptive versions. The following result guarantees that adaptation does not make the SR algorithm infinitely worse.

Proposition 1 *If* $w^* = \sup_x w_0(x) \leq \infty$ *then*

$$\sup_x w_k(x) \leq \bar{w}\ w^*,\ \textit{for any}\ k = 1, 2, \ldots$$

where $\bar{w} = \exp(\sum_j \epsilon_j)$ and $\bar{\alpha}$ is chosen such that $\bar{\alpha} \leq 1/(1 + \kappa\,\bar{w})$.

5 Examples

5.1 The witch's hat example

We consider the simplified rectangular version of "witch's hat" distribution (Mira and Tierney, 2001):

$$\pi(x) = \begin{cases} h & a < |x| < a + b \\ t + h & |x| \leq a \\ 0 & \text{otherwise} \end{cases}$$

where a, b, h and t are non-negative parameters satisfying the constraint

$$2(a + b)h + 2at = 1.$$

Mira and Tierney (2001) show that the uniform slice sampler (SS) is uniformly better than the MHIS. Further, by considering the partition of the state space $\mathbb{E} = A \cup A^C$, where $A = \{x : |x| < a\}$, they show that the rate of convergence of the SS Markov chain is given by

$$\lambda^{SS} = 1 - \frac{s}{p}$$

where $s = \frac{a}{a+b}$ and $p = \frac{a(h+t)}{a(h+t)+bh}$. Here s is the relative base of the spike and p is its probability content. The parameters s and p satisfy the constraints $0 \leq s \leq p \leq 1$.

For the SR algorithm we suppose that $\kappa = 1$, for simplicity in presentation. We let $\psi(x) = \frac{1}{2(a+b)}$, when $|x| < a + b$, as in the slice sampler case. The transition matrix of the two state (recall the partition of the state space) SR Markov chain is given by

$$K = \frac{1}{p(1-p) + s(1-s)} \begin{bmatrix} p\{1 - p + s(1-s)\} & (1-p)s(1-s) \\ ps(1-s) & (1-p)\{p + s(1-s)\} \end{bmatrix}.$$

The stationary distribution for this transition matrix is $\boldsymbol{\pi} = (p, 1-p)$, as expected. The eigenvalues of K are $\lambda_0 = 1$ and

$$\lambda^{SR} = \frac{p(1-p)}{p(1-p) + s(1-s)}.$$

The SR performs better, i.e. $\lambda^{SR} < \lambda^{SS}$ if the parameters p and s satisfy the inequality

$$p^2 - ps - s(1-s) > 0.$$

The above inequality is satisfied by the wide range of values of s and p, see Figure 1.

For a given s, SR is better than SS if

$$p > \frac{1}{2}\left(s + \sqrt{s^2 + 4s(1-s)}\right). \tag{24}$$

Otherwise SS outperforms SR. However, such cases are not interesting since in these cases both algorithms have rate of convergence much smaller than 1 (both algorithms converge very fast). For example, if $p = s$ then $\lambda^{SS} = 0$, but $\lambda^{SR} = \frac{1}{2}$. Thus the convergence of SR does not run into problems, though the performance of SR is worse than that of SS.

Consider the case when $s \to 0$. Since the right hand side of (24) approaches 0 when $s \to 0$, for all values of $p \geq s$ we have $\lambda^{SR} \leq \lambda^{SS}$, and thus the performance of SR cannot be worse than the performance of SS. Further, if $p \to 1$ then $\lambda^{SR} \to 0$ while $\lambda^{SS} \to 1$. Thus, *SR performs spectacularly better in the situation which is the most difficult one for SS.* In this limiting case, the total variation distance between the n-step transition density and the target distribution will stay close to its maximal value of 2 for the SS for arbitrarily large values of n, see Mira and Tierney (2001).

We further discuss the limiting case corresponding to $p \to 1$ when s is kept fixed at some intermediate value between 0 and 1. If p approaches 1 then λ^{SR} approaches 0. That is, if the probability content of the spike becomes very large then SR draws independent samples from $\pi(x)$. Also, the transition probability K_{21} approaches 1 when $p \to 1$ for fixed $0 < s < 1$. Thus, SR does not get stuck in the low probability region. This is in contrast to the behavior of SS which has the transition matrix:

$$T = \begin{bmatrix} s + (p-s)/p & s(1-p)/p \\ s & 1-s \end{bmatrix}.$$

The transition probability T_{21} depends on the relative base of the spike. As Mira and Tierney (2001) point out, SS gets stuck when $p \to 1$ and $s \to 0$ if the chain is started outside the spike. This does not happen for SR since K_{21} approaches 1 when $p \to 1$, as mentioned earlier. However, it is possible that $p \to 1$ but s does not approach zero in which case λ^{SR} will not approach 1.

5.2 Binomial Example

Suppose that the target distribution $\pi(x)$ is binomial with parameters r and θ and $\psi(x) = 1/(r+1)$ for all x. Let $w_i = \frac{\pi_i}{\psi_i}$ denote the importance weight for the state $i = 1, \ldots, d = r+1$. We compare the performance of SR, MHIS and the SS. We first obtain the transition matrices for the above three MCMC schemes.

For the SR we have

$$\alpha_i = \frac{1}{1+\kappa w_i} \quad \text{and} \quad \phi_i = \frac{\{1-\alpha_i\}\psi_i}{\sum_{j=1}^{d}\{1-\alpha_j\}\psi_j}.$$

Let $\boldsymbol{\alpha}^T = (\alpha_1, \ldots, \alpha_d)$; $\boldsymbol{\phi}^T = (\phi_1, \ldots, \phi_d)$ and $Diag(\boldsymbol{\alpha})$ be the $d \times d$ diagonal matrix with diagonal elements α_i. The transition matrix of the SR, following equation (6), is

$$K^{SR} = I - Diag(\boldsymbol{\alpha}) + \boldsymbol{\alpha}\boldsymbol{\phi}^T$$

where I is the $d \times d$ identity matrix.

The transition matrix, K^{MHIS} of the MHIS is obtained by Liu (1996) and Smith and Tierney (1996). Let the states be ordered (without loss of generality) such that $w_1 \geq w_2 \geq \cdots \geq w_d$ and $\lambda_i = \sum_{k>i}\left(\psi_k - \frac{\pi_k}{w_i}\right)$, $\quad 1 \leq i \leq d-1$. Now the transition matrix has elements

$$K_{ij}^{MHIS} = \begin{cases} \psi_j, & \text{if } j < i, \\ \psi_i + \lambda_i, & \text{if } j = i, \\ \frac{\pi_j}{w_i}, & \text{if } j > i. \end{cases}$$

To obtain the transition matrix of the SS, K^{SS}, we order the states in ascending order of the weights; that is, assume $v_0 < v_1 \leq v_2 \leq \cdots \leq v_d$, where $v_0 = 0$ and v_i are the weights π_i/ψ_i $(i = 1, \ldots, d)$. Let k be such that $v_k = \min(v_i, v_j)$, then

$$K_{ij}^{SS} = \frac{1}{v_i}\sum_{n=1}^{k}\frac{v_n - v_{n-1}}{r-n+2}\,.$$

The asymptotic efficiencies for the above algorithms are calculated using (19). For known value of r the efficiency of the SR is available in analytic form as a function of θ. The efficiencies for the other two algorithms are not available in closed form. That is why in the following discussion we calculate the efficiencies numerically.

Figure 2 plots the efficiencies of the three algorithms for different values of θ when r is fixed at 4. The plotted efficiencies correspond to $f(x) = x$. We plot the efficiency of the SR for $\kappa = 1$ and $\kappa = 2$. The SR is seen to be more efficient than the other two samplers when θ is either near 0 or 1. We know that for $\theta = \frac{1}{2}$ the binomial distribution is symmetric and for θ near 0 or 1 it is heavily skewed. Thus the SR performs

better than both the MHIS and SS when the distribution is skewed. With a uniform proposal distribution, intuitively, it is more difficult to simulate from a skewed distribution than a symmetric distribution. Hence the SR is more efficient in more difficult cases. The efficiency is higher for larger values of κ, as mentioned in Section 3.5. For $\kappa > 1$, the SR performs better than i.i.d. sampling for values of θ near 0 and 1. The variance formula (16) explains why this happens. For the extreme values of θ, the second term in (16) (corresponding to the importance sampling) is small compared to the first term which has the divisor κ.

A further interesting fact to observe is that the efficiency of the SR has a strictly positive lower bound. On the contrary, the lower bound for the MHIS is 0. Lastly, the efficiency curve for either the MHIS or the SS is very complex with a few local maxima and minima.

The above results agree with those from the witch's hat example, since any two point discrete distribution can essentially be treated as a Bernoulli distribution ($r = 1$). For large r, the SR with small values of κ ceases to be the most efficient sampler.

The above conclusions are only valid for the uniform proposal distribution described here and they may change if some other proposal distribution is used instead. Our choice of uniform proposal distribution is guided by the desire to compare the convergence properties of the algorithms where there is large discrepancy between the (uniform) proposal and (heavily skewed) target distributions.

5.3 Dugongs Example

We consider a real data example on age and length measurements on $n = 27$ dugongs (sea cows). Carlin and Gelfand (1991) provide a Bayesian analysis of the data set originally found in Ratkowsky (1983). The length y_i given the age x_i for the ith individual is assumed to follow a non-linear growth curve model:

$$y_i \sim N(\alpha - \beta\gamma^{x_i}, \sigma^2);\ 0 < \gamma < 1;\ i = 1, \ldots, n.$$

Following the implementation of this problem in the `Winbugs` software (Spiegelhalter *et al.*, 1996) we assume that $\tau = \sigma^{-2}$ has a Gamma prior with density proportional to $\tau^{a-1}e^{-a\tau}$, $a = 10^{-3}$. Flat priors are assumed for the remaining parameters. The joint posterior density of α, β, γ and τ is

$$\pi(\alpha, \beta, \gamma, \tau | y_1, y_2, \ldots y_n) \propto \tau^{n/2+a-1} \exp\left\{-a\tau - \tau/2 \sum_{i=1}^{n} (y_i - \alpha + \beta\gamma^{x_i})^2\right\}.$$

We integrate out τ to obtain the marginal posterior density of regression parameters, α, β and γ which is the target distribution for this example. The range of γ is between 0 and 1, the `Winbugs` software use an adaptive slice sampler for sampling from the full conditional distribution of γ given the other parameters. We compare this adaptive slice sampler method with the SR in this example.

For the SR we consider a normal proposal distribution with mean at the maximum likelihood estimate (mle) of α, β, γ and covariance matrix, $\Sigma = \sigma^2 \times I_3$ where $\sigma^2 = 0.042$. The parameter c is set at $\log \hat{c} = -13.5$, according to the estimate (4). We implement three versions of SR corresponding to $\log(\kappa\, c) = -13.5, -13$ and -12.5 to see the sensitivity.

Table 1 provides the effective sample size (ESS) defined in Section 3.5 and the lag-1 autocorrelations for this example. For each sampler we generate $N = 15,000$ samples from the proposal distribution and discard the first 5,000 MCMC samples. The number of MCMC samples, M, for the SR is random, as discussed previously. For the three implementations of the SR with different values $\log(\kappa\, c)$ the values of M were 19,233, 32,543 and 45,502, respectively. Using the asymptotic relationship $M \approx N\kappa$ the implied values of κ are 1.28, 2.17 and 3.03, respectively.

It is seen that all three versions of the SR are better than the `Winbugs` implementation. The lag-1 auto-correlation increases as κ increases since the chain accepts a particular proposal many more times corresponding to a larger value of κ. However, the ESS also increases as κ increases since the efficiency is an increasing function of κ, as mentioned in Section 3.5. Also M, the number of MCMC samples, increases as κ increases. We do not increase κ further since the ESS does not improve substantially. Further, the ESS per second (not shown) goes down since the running times for the three values of κ are 1, 2 and 3 seconds respectively. The running time for the `Winbugs` is 9 seconds which is longer than any of the three SR versions.

This example illustrates that the proposed regenerative schemes can outperform the Gibbs sampler which implements an adaptive uniform slice sampler to sample from its full conditional distribution.

5.4 A Bates-Watts example

We consider a data set given in Bates and Watts (1988, p.307), modeled by Newton and Raftery (1994) as follows. Response y_i $(i = 1, \ldots, n = 16)$ is modeled as

$$y_i = \beta_1 + \frac{\beta_2}{1 + \exp\{-\beta_4(x_i - \beta_3)\}} + \epsilon_i,$$

where ϵ_i is assumed to be i.i.d. normally distributed with mean 0 and variance σ^2. The prior for $\boldsymbol{\beta}$ and σ^2 is taken as $p(\boldsymbol{\beta}, \sigma^2) \propto \sigma^{-2}$.

For this model we first integrate out σ^2 analytically. Now the target distribution, $\pi(\boldsymbol{\beta}|y)$ is four dimensional. We take a 4-dimensional normal proposal distribution, ψ. We take the maximum likelihood estimate (mle) to be the mean. The proposal covariance matrix, Σ say, is taken as 0.1 times the diagonal matrix having the variances of the mle's along its diagonal.

Our objective in this example is to compare the ASR with other tuned MCMC methods. The following MCMC samplers are implemented. First, we implement a Metropolis-Hastings sampler with the proposal distribution (as suggested by a referee)

$$q(\boldsymbol{\beta}|\boldsymbol{\phi}) = \epsilon\, \psi(\boldsymbol{\beta}) + (1-\epsilon)\, \chi(\boldsymbol{\beta}|\boldsymbol{\phi}), \tag{25}$$

where $\epsilon > 0$ and χ is the multivariate normal distribution with mean $\boldsymbol{\phi}$ and covariance matrix Σ. The proposal distribution is a mixture of the MHIS proposal distribution ψ and a random walk proposal distribution $\chi(\boldsymbol{\beta}|\boldsymbol{\phi})$ where $\boldsymbol{\phi}$ is the current point of the chain. Thus the proposal distribution $q(\boldsymbol{\beta}|\boldsymbol{\phi})$ of the Metropolis-Hastings algorithm tries to combine the best properties of the independence sampler and a random walk Metropolis sampler. We optimize the value of ϵ by trial and error so that the average ESS under this scheme is the best.

Second, we consider the adaptive schemes proposed by Gilks *et al.* (1998) as implemented in `Winbugs`. This adaptive scheme adapts the symmetric (around the current point) normal proposal distribution whose standard deviation is tuned over the first 4000 iterations in order to get the acceptance rate between 20% and 40%.

Lastly, for implementing the ASR we take $\log \hat{c} = 114.5$ according to (4). The starting proposal for the ASR is assumed to be ψ, as described above. The densities χ_k's are chosen to be the densities of the multivariate normal distribution with the current proposal point $x_{(k)}$ as the mean and the covariance matrix Σ. Although we have experimented with an adaptive estimate of the covariance matrix, we report the results with the above fixed covariance matrix Σ for the χ_k's.

We generate $N = 15,000$ samples from the proposal distribution in each case. We discard the first 5,000 iterations and use the remaining samples for making the following comparisons. In each case we calculate the ESS. We also calculate the average of the ESS over different parameters. As in the previous example the running times for all the implementations are negligible.

The number of MCMC samples, M, for the Metropolis-Hastings and the `Winbugs` implementations is 15,000. For the ASR, M is a random variable. For the three values of $\log(\kappa c)$ reported in Table 2, the values of M were 15,821, 25,147 and 36,848, respectively. Using the approximate result $M \approx N\kappa$, the implied values of κ are 1.05, 1.68 and 2.46, respectively.

Observe that the last two versions of the ASR outperform the `Winbugs` and the Metropolis-Hastings scheme using the mixture proposal distribution (25). The first version with a small value of κ performs poorly, pointing the need to tune κ. It is possible to increase κ further. However, as mentioned in Section 2.2, larger values of κ lead to smaller rejection probability α. As a result the auto-correlation increases and further gain in efficiency is not substantial.

Figure 3 plots the time series and the corresponding autocorrelation plots for β_1 for the three samplers. The top row is for the output from the Metropolis-Hastings algorithm, the middle is for the output from `Winbugs` and the last row is for the ASR sampler with $\log(\kappa c) = 114.5$, The first two samplers show signs of slow mixing. This is confirmed by looking at the ACF plots. The ASR is seen to be the best mixing sampler. The plots for the remaining parameters looked similar and are not reported here.

6 Discussion

In this article we have introduced an MCMC sampler which can be superior to the Metropolis-Hastings samplers (including the slice sampler) in terms of efficiency and convergence properties. This sampler is easy to implement and adapt. The examples reported in the paper show that its convergence characteristics are considerably better than that of the Metropolis-Hastings schemes when there is large disagreement between the target and the proposal distributions.

The strength of the proposed algorithm lies in the fact that regeneration times are easily identified and this allows on-line adaptation of the proposal distribution. As we prove in the paper the proposed form of adaptation does not alter the ergodic behavior of the averages formed along the chain.

Acknowledgement

We are very grateful to Professor Peter Green and Antonietta Mira for their valuable discussion on an earlier draft of this paper. We are also grateful to the referees for many helpful comments.

	Winbugs		SR		SR		SR	
			$\log(\kappa\, c) = -13.5$		$\log(\kappa\, c) = -13$		$\log(\kappa\, c) = -12.5$	
	ESS	ρ_1	ESS	ρ_1	ESS	ρ_1	ESS	ρ_1
α	3465.94	0.94	5175.55	0.88	9641.95	0.93	13999.94	0.95
β	4372.78	0.64	5205.80	0.87	9711.19	0.92	14050.49	0.94
γ	3533.84	0.91	5278.49	0.85	9771.10	0.91	14138.00	0.93
Average	3790.85	0.83	5219.95	0.86	9708.08	0.92	14062.81	0.94

Table 1: Effective sample size and lag-1 autocorrelations for the dugongs example

	Metropolis-Hastings		Winbugs		ASR		ASR		ASR	
					$\log(\kappa\, c) = 114.5$		$\log(\kappa\, c) = 115$		$\log(\kappa\, c) = 115.5$	
	ESS	ρ_1	ESS	ρ_1	ESS	ρ_1	ESS	ρ_1	ESS	ρ_1
β_1	4150.90	0.70	3860.55	0.80	2882.23	0.51	6953.56	0.59	11157.05	0.70
β_2	4643.01	0.58	4033.42	0.74	2757.77	0.56	6851.10	0.61	11074.89	0.71
β_3	5281.82	0.45	3464.67	0.94	2672.36	0.59	6525.77	0.66	10794.10	0.74
β_4	5890.32	0.35	5654.00	0.38	2748.22	0.56	6698.30	0.63	10936.29	0.73
Average	4991.51	0.52	4253.16	0.72	2765.15	0.55	6757.18	0.62	10990.58	0.72

Table 2: Effective sample size and lag-1 autocorrelations for the Bates and Watts example.

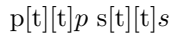

Figure 1: The SR is better than the slice sampler in the shaded region in the Witch's hat example.

Figure 2: Efficiency for the binomial example with $r = 4$.

Figure 3: Time series and acf plots of β_1 from the three samplers for the Bates and Watts example.

Appendix: Theorem Proofs

Proof of Theorem 1 (i). Let the pair m and n be as given in Algorithm 1. Let $\{U_j, j = 0, 1, \ldots\}$ be an i.i.d. sequence of Uniform(0,1) random variables. For any $A \in \mathcal{E}$ we have:

$$
\begin{aligned}
&Pr(X_{m+1} \in A | X_m = x) = Pr(U_0 \geq \alpha(x), x \in A) + \\
&\sum_{j=1}^{\infty} Pr\Big(U_0 < \alpha(x), U_1 < \alpha(Z_{n+1}), \cdots, U_{j-1} < \alpha(Z_{n+j-1}), U_j \geq \alpha(Z_{n+j}), Z_{n+j} \in A\Big) \\
&= \{1 - \alpha(x)\} 1_A(x) + \alpha(x) \sum_{j=1}^{\infty} \left(\int \alpha(z)\psi(z)\mu(dz) \right)^{j-1} \int_A \{1 - \alpha(z)\}\psi(z)\mu(dz) \\
&= \{1 - \alpha(x)\} 1_A(x) + \alpha(x) \frac{\int_A \{1 - \alpha(z)\}\psi(z)\mu(dz)}{1 - \int \alpha(z)\psi(z)\mu(dz)} \\
&= \{1 - \alpha(x)\} 1_A(x) + \alpha(x) \int_A \phi(z)\mu(dz).
\end{aligned}
$$

This proves (6). It is straightforward to verify that $\int K(x, dy) = 1$. Let us define the transition density

$$K(x, y) = \alpha(x)\phi(y) + \{1 - \alpha(x)\}\delta_x(y).$$

Hence $K(x, dy) = K(x, y)\mu(dy)$. Now we prove equation (7):

$$
\begin{aligned}
\int \pi(x)K(x, y)\mu(dx) &= \int \pi(x) \Big[\alpha(x)\phi(y) + \{1 - \alpha(x)\}\delta_x(y) \Big] \mu(dx) \\
&= \phi(y) \int \pi(x)\alpha(x)\mu(dx) + \{1 - \alpha(y)\}\pi(y) \\
&= \tfrac{\{1-\alpha(y)\}\psi(y)}{\int \{1-\alpha(z)\}\psi(z)\mu(dz)} \int \pi(x)\alpha(x)\mu(dx) + \{1 - \alpha(y)\}\pi(y) \\
&= \{1 - \alpha(y)\} \left[\psi(y) \left(\int \tfrac{\kappa\pi(z)\psi(z)}{\kappa\pi(z)+\psi(z)} \mu(dz) \right)^{-1} \int \tfrac{\pi(x)\psi(x)}{\kappa\pi(x)+\psi(x)} \mu(dx) + \pi(y) \right] \\
&= \{1 - \alpha(y)\} \left[\tfrac{\psi(y)}{\kappa} + \pi(y) \right] \\
&= \tfrac{\kappa\pi(y)}{\kappa\pi(y)+\psi(y)} \tfrac{\psi(y)+\kappa\pi(y)}{\kappa} \\
&= \pi(y).
\end{aligned}
$$

□

Lemma A.1 *If (1) holds then every bounded harmonic function is a constant under K.*

Proof of Lemma A.1. Let h be a harmonic function i.e.,

$$h(x) = \int K(x, dy)h(y) \quad \text{for all } x \in \mathbb{E}. \tag{A.1}$$

Now

$$
\begin{aligned}
\int K(x, dy)h(y) &= \int \Big[\alpha(x)\phi(y) + \{1 - \alpha(x)\}\delta_x(y) \Big] h(y)\mu(dy) \\
&= \alpha(x) \int \phi(y)h(y)\mu(dy) + \{1 - \alpha(x)\}h(x).
\end{aligned}
$$

Therefore (A.1) is equivalent to:

$$\alpha(x)h(x) = \alpha(x)\int \phi(y)h(y)\mu(dy).$$

The support condition (1) guarantees that $\alpha(x) \neq 0$ for all $x \in \mathbb{E}$. Hence we have,

$$h(x) = \int \phi(y)h(y)\mu(dy) = \text{ constant.}$$

□

Proof of Theorem 1 (ii). The condition (1) implies that $0 < \alpha(x) < 1$ for all $x \in \mathbb{E}$. Hence irreducibility and aperiodicity follow.

Using Theorem 1 of Tierney (1994) we see that K is positive recurrent and π is the unique invariant distribution of K. Now K is Harris recurrent by using Lemma A.1 and Theorem 2 of Tierney (1994). Hence by Theorem 1 of Tierney (1994), based on the results of Nummelin (1984), the proof is complete. □

Proof of Theorem 2. Note that $w(x) \leq w^* < \infty$ for all x implies the support condition (1). Hence Theorem 1 holds. Now we show that the state space $\mathbb{E}$ is small i.e.,

$$K(x, \cdot) \geq \beta\nu(\cdot) \quad \text{for all } x \in \mathbb{E} \tag{A.2}$$

for a suitable $\beta > 0$ and a probability measure $\nu(\cdot)$ on $\mathcal{E}$. Here we have the transition kernel

$$K(x, dy) = \Big[\alpha(x)\phi(y) + \{1 - \alpha(x)\}\delta_x(y)\Big]\mu(dy).$$

The choice of $\beta = \frac{1}{1+\kappa w^*}$ and $\nu(\cdot) = \phi(\cdot)$ suffice for the minorization condition (A.2) to hold. Hence using Proposition 2 of Tierney (1994) we have the result. □

We shall use the following standard result, see e.g., Rao (1973, page 122).

Lemma A.2 *Let X_n, X, Y_n be random variables such that $X_n \Rightarrow X$, and $Y_n \xrightarrow{P} y$. Then*

$$X_n + Y_n \Rightarrow X + y. \quad \textit{Also} \quad \frac{X_n}{Y_n} \Rightarrow \frac{X}{y} \quad \textit{if} \quad y \neq 0.$$

Proof of Lemma 2.

$$\begin{aligned} E(\xi) &= E_\psi\big\{E(\xi|Z)\big\} \\ &= E_\psi\big\{\tfrac{1-\alpha(Z)}{\alpha(Z)}\big\} \\ &= \int \kappa w(z)\psi(z)\mu(dz) \\ &= \kappa \int \pi(z)\mu(dz) = \kappa. \end{aligned}$$

$$
\begin{aligned}
\mathrm{Var}(\xi) &= E_\psi\big\{\mathrm{Var}(\xi|Z)\big\} + \mathrm{Var}_\psi\big\{E(\xi|Z)\big\} \\
&= E_\psi\big\{\tfrac{1-\alpha(Z)}{\alpha(Z)^2}\big\} + \mathrm{Var}_\psi\big\{\tfrac{1-\alpha(Z)}{\alpha(Z)}\big\} \\
&= E_\psi\big\{\kappa w(Z)[1+\kappa w(Z)]\big\} + \mathrm{Var}_\psi\big\{\kappa w(Z)\big\} \\
&= \kappa + \kappa^2 E_\psi\big\{w(Z)^2\big\} + \kappa^2 \mathrm{Var}_\psi\big\{w(Z)\big\} \\
&= \kappa + \kappa^2\left\{2E_\pi(w(X)) - 1\right\}.
\end{aligned}
$$

□

Proof of Theorem 4. First, it is straightforward that if $M \to \infty$ then $N \xrightarrow{P} \infty$; the proof follows from Lemma A.2 and the Central Limit Theorem. Let $V_i = \xi_i - \kappa$ and $\bar{V}_N = \sum V_i/N$. We have:

$$
\begin{array}{lccccc}
 & \sum_{i=1}^N \xi_i & \leq & M & \leq & \sum_{i=1}^{N+1} \xi_i \\
\text{iff} & \sum_{i=1}^N (\xi_i - \kappa) & \leq & M - N\kappa & \leq & \sum_{i=1}^N (\xi_i - \kappa) + \xi_{N+1} \\
\text{iff} & \sqrt{N}\bar{V}_N & \leq & \frac{M-N\kappa}{\sqrt{N}} & \leq & \sqrt{N}\bar{V}_N + \frac{\xi_{N+1}}{\sqrt{N}}.
\end{array}
$$

Note that $\frac{\xi_{N+1}}{\sqrt{N}} \xrightarrow{P} 0$ and the Central Limit Theorem implies that $\sqrt{N}\bar{V}_N \Rightarrow N(0, \tau_\xi^2)$. Also, we can replace N by $\frac{M}{\kappa}$ using Lemma A.2. Hence the result follows. □

Proof of Theorem 5. To prove the first part we obtain:

$$
E\{\xi f(Z)\} = E_\psi\left\{f(Z)E(\xi|Z)\right\} = \kappa I_f. \tag{A.3}
$$

(See the proof of Lemma 2 for more details.) Also we have:

$$
\begin{aligned}
\bar{f}_N - I_f &= \tfrac{\sum_{i=1}^N \xi_i f(Z_i)}{\sum_{i=1}^N \xi_i} - I_f \\
&= \Big(\sum_{i=1}^N \xi_i f(Z_i) - I_f \sum_{i=1}^N \xi_i\Big) / \sum_{i=1}^N \xi_i \\
&= \Big(\sum_{i=1}^N \xi_i \tilde{f}(Z_i)\Big) / \sum_{i=1}^N \xi_i = \bar{V}_N/\bar{\xi}_N,
\end{aligned}
$$

where $\bar{V}_N = \frac{1}{N}\sum_{i=1}^N V_i$, $V_i = \xi_i \tilde{f}(Z_i)$ and $\bar{\xi}_N = \frac{1}{N}\sum_{i=1}^N \xi_i$. Using equation (A.3) and Lemma 2 we have that $E(V_i) = 0$. Note that $\{V_i\}$ is a sequence of i.i.d. random variables. Now using the Khinchine's weak law of large numbers, see e.g., Rao (1973, page 112) we have that $\bar{V}_N \xrightarrow{P} 0$ and $\bar{\xi}_N \xrightarrow{P} \kappa$ where $\kappa > 0$. Hence the first part follows. □

To prove the second part we evaluate the variance of V_i:

$$
\begin{aligned}
\mathrm{Var}(V_i) &= E\big\{\xi^2 \tilde{f}(Z)^2\big\} \\
&= E_\psi\big\{\tilde{f}(Z)^2 E(\xi^2|Z)\big\} \\
&= E_\psi\big\{\tfrac{1-\alpha(Z)}{\alpha(Z)}\tfrac{2-\alpha(Z)}{\alpha(Z)}\tilde{f}(Z)^2\big\} \\
&= E_\psi\big\{\kappa w(Z)[1+2\kappa w(Z)]\tilde{f}(Z)^2\big\} \\
&= \kappa \textstyle\int w(z)[1+2\kappa w(z)]\tilde{f}(z)^2\psi(z)\mu(dz) \\
&= \kappa\big\{\sigma_f^2 + 2\kappa \textstyle\int \tilde{f}(z)^2 w(z)\pi(z)\mu(dz)\big\} = \kappa^2\sigma^2,
\end{aligned}
$$

where σ^2 is given in (16). Hence $\{V_i\}$ is a sequence of i.i.d. random variables with mean zero and finite variance $\kappa^2\sigma^2$. By the Central Limit Theorem we have that $\sqrt{N}\bar{V}_N \Rightarrow N(0, \kappa^2\sigma^2)$. Using the first part of this theorem and Lemma A.2, the result follows. □

Proof of Proposition 1. Let $\omega_k(x) = \frac{\psi_k(x)}{\pi(x)}$ denote the inverse importance ratio for any given k. It is straightforward that $\omega_{k+1}(x) \geq (1 - \epsilon_{k+1})\ \omega_k(x)$. From this recursion relation we have:

$$\begin{array}{rcccl} \frac{\omega_{k+i}(x)}{\omega_k(x)} & \geq & \prod_{j=k+1}^{k+i}(1-\epsilon_j) & \geq & \prod_{j=1}^{\infty}(1-\epsilon_j) \quad = \quad \exp\{\sum \log(1-\epsilon_j)\} \\ & \geq & \exp\{-\sum \epsilon_j\} & \geq & \exp\{-\log \bar{w}\} \quad = \quad \bar{w}^{-1}. \end{array}$$

This proves that $w_{k+i}(x) \leq \bar{w}\ w_k(x)$, $i = 1, 2, \ldots$ which implies the required. □

References

Bates, D. M. and Watts, D. G. (1988) *Nonlinear Regression Analysis & its Applications*. New York: John Wiley and Sons.

Carlin, B. P and Gelfand, A. E. (1991) An iterative Monte Carlo method for nonconjugate Bayesian analysis. *Statistics and Computing*, **1**, 119–128.

Chib, S. (1995) Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association*, **90**, 1313–1321.

Chib, S and Greenberg, E. (1995) Understanding the Metropolis-Hastings Algorithm. *American Statistician*, **49**, 327–335.

Chib, S. and Jeliazkov, I (2001) Marginal likelihood from the Metropolis-Hastings output. *Journal of the American Statistical Association*, **96**, 270–281.

Gelfand, A. E. and Smith, A. F. M. (1990) Sampling Based Approaches to Calculating Marginal Densities. *Journal of the American Statistical Association*, **85**, 398–409.

Geyer, C. J. (1992) Practical Markov chain Monte Carlo. *Statistical Science*, **7**, 473–483.

Gilks, W. R., Richardson, S. and Spiegelhalter, D. G. (1996) *Markov Chain Monte Carlo In Practice*, London: Chapman and Hall.

Gilks, W. R., Roberts, G. O. and Sahu, S. K. (1998) Adaptive Markov Chain Monte Carlo Through Regeneration. to appear *Journal of the American Statistical Association*, **93**, 1045–1054.

Green, P. J. and Han, X-L. (1992) Metropolis methods, Gaussian proposals, and antithetic variables. In *Stochastic models, Statistical Methods and Algorithms in Image Analysis. Lect. Notes Statist.,* **74**, 142–164, Springer-Verlag, Berlin.

Hastings, W. K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97–109.

Kass, R. E., Carlin, B. P., Gelman, A. and Neal, R. (1998) Markov chain Monte Carlo in practice: A roundtable discussion. *American Statistician*, **52**, 93–100.

Liu, J. S. (1996) Metropolized Independent Sampling with Comparisons to Rejection Sampling and Importance Sampling. *Statistics and Computing*, **6**, 113–119.

Mira, A. and Tierney, L. (2001) Efficiency and Convergence Properties of Slice Samplers. *Scandinavian Journal of Statistics*, to appear.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953) Equations of state calculations by fast computing machine. *J. Chem. Phys.*, **21**, 1087–1091.

Mykland, P. Tierney, L. and Yu, B. (1995) Regeneration in Markov chain samplers. *Journal of the American Statistical Association*, **90**, 233–241.

Newton, M. A. and Raftery, A. E. (1994) Approximate Bayesian inference with the weighted likelihood bootstrap (with discussion). *Journal of the Royal Statistical Society* B, **56**, 3–48.

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

Peskun, P. H. (1973) Optimum Monte-Carlo Sampling Using Markov Chains. *Biometrika*, **60**, 607–612.

Rao, C. R. (1973) *Linear Statistical Inference and its Applications.* New York: John Wiley and Sons.

Ratkowsky, D. (1983) *Nonlinear regression modelling.* Mercel Dekker: New York.

Ripley, B. D. (1987) *Stochastic Simulation.* New York: John Wiley and Sons.

Robert, C. P. (1995) Convergence Control Methods for Markov Chain Monte Carlo Algorithms. *Statistical Science*, **10**, 231–253.

Robert, C. P. and Casella, G. (1999) *Monte Carlo Statistical Methods.* New York: Springer.

Roberts, G. O. (1996) Markov chain concepts related to sampling algorithms. In *Markov Chain Monte Carlo in Practice.* (Eds. W. R. Gilks, S. Richardson and D. J. Spiegelhalter). London: Chapman and Hall, pp 45–57.

Smith, A. F. M. and Roberts, G. O. (1993) Bayesian Computation via the Gibbs Sampler and Related Markov Chain Monte Carlo Methods. *Journal of the Royal Statistical Society*, B, **55**, 3–23.

Smith, R. L. and Tierney, L. (1996) Exact Transition Probabilities for the Independence Metropolis Sampler. Preprint.

Spiegelhalter, D. J., Thomas, A. and Best, N. G. (1996) Computation on Bayesian graphical models. In *Bayesian Statistics 5*, (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith). Oxford: Oxford University Press, 407–426.

Tierney, L. (1994) Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics*, **22**, 1701–1762.

Tierney, L. (1998) A note on Metropolis-Hastings Kernels for General State Spaces. *Annals of Applied Probability*, **8**, 1–9.