18 July 2020  
Prof. Enrico Colosimo  
Editor-in-Chief  
Brazilian Journal of Probability and Statistics  

Dear Professor Colosimo,

I am writing to submit my revised manuscript entitled *A Monte Carlo Integration Approach to Estimating Drift and Minorization Coefficients for Metropolis-Hastings Samplers* for consideration for publication in *Brazilian Journal of Probability and Statistics*. I would like to express my appreciation to the reviewers for their insightful comments. I believe their comments and their having been addressed has made this a much stronger manuscript. In the next two pages, I respond to the reviewer comments.

Please feel free to contact me with any questions or concerns. I can be reached via email at spade@uwm.edu. Thank you for your consideration of my revised manuscript.

Sincerely,

David A. Spade  
Assistant Professor of Statistics  
University of Wisconsin—Milwaukee
Responses to First Reviewer’s Comments

1) “The author needs to make clear what is the contribution of this work when compared to the two
main references cited in the paper. If I understand correctly, these works consider only the
update of univariate blocks via random walk proposals, is that correct? Please make this clear in
the introduction. A brief description of the RSM and RWM algorithms would be helpful here.”

The author has added a clearer description of the contribution of this work in Section 1, as
well as a brief description of the RSM and RWM samplers.

2) “It would help to motivate the work if the author provided some discussion about why and when
a MH MCMC that is not the RSM or RWM should be considered.”

The author now addresses this in Section 1, outlining the need for a Metropolis-Hastings
algorithm that is not the RSM or RWM for complicated, highly asymmetric target densities.

3) “It would be interesting to have a discussion about how much is gained or lost in precision and
efficiency in comparison to the two previous works. In particular, the author could compare the
methodologies for the univariate examples with random walk proposals presented in Section 4.”

This comparison has been expanded upon at the end of Section 4.1.

4) “As pointed out by the author in the conclusions, the proposed method suffers from the ‘curse of
dimensionality’. I would expect a more detailed discussion about this issue as to give a clearer
idea of what to expect and how robust this is with w.r.t the dimension and behavior of the target
distribution. In particular, the author should highlight the quite reasonable cost in the example
with dimension 10 presented in the paper.”

This discussion is now included in the third paragraph of Section 5.

5) “A deeper discussion about the choice of d and V should be provided, including the impact on the
results. The examples from Section 4 do not help much regarding this issue. Also, is it
guaranteed that the V’s chosen in the examples are actual drift functions?”

This discussion is now included in the fourth paragraph of Section 5.

6) “Since the bound from Theorem 1 is known to be conservative, can the author say anything
about how likely it is that the (Monte Carlo) bounds provided by the proposed method are
genuine bounds? That is, the Monte Carlo error is mitigated by the conservative result to
guarantee the validity of the bounds?”

This discussion is now included in the fourth and fifth paragraphs of Section 5.
7) “In the first example, the exact posterior distribution is known (Beta). This could be used to further explore the results.”

Drift and minorization conditions remained difficult to verify analytically even for this example, but QQ plots of the quantiles of the post-burn-in output of each of the two chains versus the quantiles of the target distribution as well as a brief discussion of them are now included in Section 4.1.

8) “I think there is something wrong in Definition 5.”

This definition has been reformulated in the correct way.

Response to the Second Reviewer’s Comments

“The author describes a procedure for bounding the mixing time of an MCMC method. The problem, in my opinion, is that this procedure depends on the estimation of the marginal likelihood (used as a drift function. If this estimation is done using the same samples in the chain (i.e. generated by the MCMC), clearly this is an issue (we are using an estimator built using samples in a chain in order to state if the same chain has converged). Please comment with a clear remark on this point.”

A clarifying comment on the estimation of the marginal density has been added to make clear that the estimation of the marginal density is done independently of the chain under study.
A Monte Carlo Integration Approach to Estimating Drift and Minorization Coefficients for Metropolis-Hastings Samplers

David A. Spade

University of Wisconsin–Milwaukee

Abstract. Bayesian statistical methodology has become highly popular in a myriad of applications over the past several decades. In Bayesian statistics, it is often required to draw samples from intractable probability distributions. Markov chain Monte Carlo (MCMC) algorithms are common methods of obtaining samples from these distributions. When an MCMC algorithm is used, it is important to be able to obtain an answer to the question of how many iterations the chain must run before it is “close enough” to its target distribution to allow approximate sampling from this distribution. Several methods of approaching this question exist in the literature. Some rely on the output of the chain, and some are based on Markov chain theory. These techniques suffer from major practical limitations. This work provides a computational method of bounding the mixing time of a Metropolis-Hastings algorithm. This approach extends the work of Spade (2016) and Spade (2020) to general versions of the Metropolis-Hastings algorithm, while examining the convergence behavior of such samplers under symmetric and asymmetric proposal densities.

1 Introduction

In many settings, especially those in Bayesian statistical analysis, parameter estimation requires sampling from intractable probability distributions. The literature contains a myriad of ways of doing this, but one of the most common classes of techniques for obtaining such samples is the class of MCMC algorithms. When MCMC algorithms are used, a major question concerning them is how long the chain needs to run before subsequent states of the chain may be viewed as approximate samples from the target density. Rosenthal (1995) presents a theory-based method of bounding the mixing time of geometrically ergodic Markov chains using the values of drift and minorization coefficients. Meyn and Tweedie (2005) provide a set of conditions under which a Markov chain is geometrically ergodic. These results
are theoretically important, but they can be very difficult to use in practice. Fortunately, several investigators have provided conditions under which particular versions of a Markov chain are geometrically ergodic. Mengersen and Tweedie (1996) give conditions under which the underlying Markov chain for an independent Metropolis-Hastings algorithm is geometrically ergodic. Roberts and Rosenthal (1997) and Roberts and Rosenthal (1998) give a drift function for a geometrically ergodic hybrid sampler, and Roberts and Rosenthal (1999) describe situations in which the Markov chain associated with a slice sampler is geometrically ergodic. Rosenthal (2001) gives a comprehensive review of these results. Since the publication of this review, several other theoretical results have come about. Jarner and Hansen (2000) provide a set of conditions under which a random-walk Metropolis (RWM) sampler is geometrically ergodic, and they provide a suitable drift function for RWM samplers that satisfy the conditions. Fort et al. (2003) give a set of conditions under which a hybrid sampler known as the random-scan Metropolis (RSM) algorithm is geometrically ergodic, and they provide a drift function for RSM samplers that satisfy the conditions.

MCMC methods come up in a variety of settings. For example, MCMC algorithms have come up in evolutionary biology when Bayesian methods are used to make inferences about the evolutionary history among a group of genes or species based on present-day genetic data (Li et al., 2000; Yang and Rannala, 2010), in financial applications (Eraker, 2001), in digital communications (Chen, 2002), in making determinations about whether a particular gene demonstrates differential expression behavior (Erkkila et al., 2010), in economics to estimate value-at-risk (Koike and Minami, 2019), and in many other applications. The frequency with which MCMC algorithms are used means that the ability to gain insight into their convergence behavior is extremely important. While more recent theoretical results are somewhat easier to use in practice than are some of the earlier ones, circumstances often arise in which these results remain analytically intractable. In these situations, it is common to rely on convergence diagnostics that make use of the chain’s output. Heidelberger and Welch (1983) give a technique that is based on the theory of the Browninan bridge and spectral analysis. Gelman and Rubin (1992) assess convergence through comparison of variation between and within chains, where several independent chains are initialized from different regions of the state space. Raftery and Lewis (1992) diagnose convergence through a technique that has its roots in the theory of two-state Markov chains, and Geweke (1992) presents a method that relies on estimating the spectral density of the chain. Several other methods are given by Roberts (1992), Yu (1994), Yu and Mykland (1994), Zellner and Min (1995),
Cowles and Carlin (1996) give a comprehensive review of all these diagnostic methods. While useful for a particular run of an MCMC algorithm or in situations where each Markov chain is initialized from the same point, there exist several important limitations to the output-based convergence diagnostics. The major issue with them is that none of them give a clear picture of how long a general version of a particular Markov chain takes to approach its stationary distribution. They only give a sense of the burn-in time from a particular starting point. Due to its use of several independent chains started from various points in the state space, the Gelman and Rubin (1992) diagnostic is somewhat robust to this issue. Some diagnostics will suggest a lack of convergence for a given chain, while others will indicate that no evidence of a lack of convergence exists. Therefore, the diagnosis of convergence is somewhat dependent on the diagnostic that is chosen. Furthermore, if a diagnostic suggests that the chain has not burned in, the investigator will have to re-run the chain for a larger number of steps and then re-assess convergence. This can result in a similar issue to the one that arises with the use of the Rosenthal (1995) upper bound on the mixing time, which is well-known to be highly conservative. The investigator may run the chain much longer than is necessary to ensure approximate convergence. Lastly, when diagnosing convergence, the output-based methods do not give a sense of what a diagnosis of “no lack of convergence” means with regard to closeness to the target distribution. Having a theoretical upper bound on the mixing time prior to running the chain addresses all of these problems except for that of how conservative those bounds on the mixing time tend to be. The usefulness of the theory-based methods is hampered by their frequent lack of analytical tractability, which makes it helpful to have an intermediary approach between the output-based methods and the theory-based techniques of bounding the mixing time.

In the literature, one may find such intermediary methods for bounding the mixing time of a geometrically ergodic Markov chain. Cowles and Rosenthal (1998) present a method that uses auxiliary simulations to estimate drift and minorization coefficients. Spade (2016) presents a technique for estimating an upper bound on the mixing time of a geometrically ergodic RSM chain. This method relies on Monte Carlo integration to estimate drift and minorization coefficients. This method is equipped to handle higher dimensions than is the technique Cowles and Rosenthal (1998) provide, but it will still suffer from the “curse of dimensionality” if the dimension of the state space is sufficiently large. Spade (2020) extends this procedure to the full-updating RWM sampler, and the goal of this manuscript is to demon-
strate that the method provided by Spade (2020) extends well to general Metropolis-Hastings algorithms, provided one can find a suitable drift function.

The main contribution of this work is the construction of a computational approach to bounding the mixing time of the Metropolis-Hastings algorithm. This work generalizes the related techniques for the RSM and RWM algorithms that were mentioned earlier. The need for such an approach arises because in many situations in which MCMC algorithms are used, the target density is asymmetric. In such cases, the Metropolis-Hastings algorithm tends to perform better than the RSM and RWM algorithms in terms of mixing behavior because it allows asymmetric proposal densities. In general, the performance of Metropolis and Metropolis-Hastings algorithms is better when the proposal density has a similar shape to that of the target density. The RSM algorithm updates one variable at a time, and it does so by choosing a variable at random and then proposing an update to it from a symmetric proposal density. Therefore, it is not the best choice for settings when the target density is asymmetric. The RWM algorithm updates all variables at one time, but it also proposes updates from a symmetric proposal density. Consequently, the RWM algorithm is not as well-suited for settings in which the target density is asymmetric. Since settings in which an asymmetric target density is used come up quite frequently, it is important to have the ability to obtain a useful bound on the mixing time of the Metropolis-Hastings algorithm. That is precisely what this work aims to provide.

The remainder of this manuscript is organized as follows. Section 2 presents the background on Markov chains and the Metropolis-Hastings algorithm that is required for an understanding of the methods presented in later sections. Section 3 presents a technique for estimating drift and minorization coefficients en route to bounding the mixing time. Section 4 gives several illustrative examples of the proposed methods at work. In this section, bounds on the mixing time are obtained for each example, and stability analysis and computing times are provided. Section 5 provides a discussion of the work presented in earlier sections, as well as the advantages and limitations of the methods presented in this manuscript.

2 Preliminaries

This section provides the necessary background on Markov chains, as well as that on the Metropolis-Hastings algorithm, that is required for an understanding of the rationale behind the methods presented in Section 3. To
begin, we provide some concepts and results pertaining to general Markov chains. Let \((\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m), \mathbb{P})\) be a probability space, where \(\mathcal{B}(\mathbb{R}^m)\) is the \(\sigma\)-field of Borel subsets of \(\mathbb{R}^m\) and \(\mathbb{P}\) is a probability measure. Let \((X_t)_{t \geq 0}\) be an ergodic Markov chain on \(\mathbb{R}^m\) with transition kernel \(K(\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m)) \mapsto [0,1]\) and stationary measure \(\pi\) that admits a probability density \(p(\cdot)\). These terms are defined below.

**Definition 1.** Let \(K : \{K(x, A) : x \in \mathbb{R}^m, A \in \mathcal{B}(\mathbb{R}^m)\}\). If \(K\) is such that for all \(x \in \mathbb{R}^m\), \(K(x, \cdot)\) is a probability measure on \(\mathcal{B}(\mathbb{R}^m)\) and for all \(A \in \mathcal{B}(\mathbb{R}^m)\), \(K(\cdot, A)\) is a measurable function on \(\mathbb{R}^m\), \(K\) is a transition kernel. For the Markov chain \((X_t)_{t \geq 0}\), the transition kernel defines the probability that if \(X_t = x\), \(X_{t+1} \in A\). In other words,

\[
K(x, A) = P(X_{t+1} \in A | X_t = x).
\]

The \(k\)-step transition kernel \(K^k(\cdot, \cdot)\), where \(k\) is a positive integer, for \((X_t)_{t \geq 0}\) is a transition kernel with the property that

\[
K^k(x, A) = P(X_{t+k} \in A | X_t = x).
\]

**Definition 2.** A \(\sigma\)-finite probability measure \(\pi\) on \(\mathcal{B}(\mathbb{R}^m)\) with the property that for all \(A \in \mathcal{B}(\mathbb{R}^m)\),

\[
\pi(A) = \int_{\mathbb{R}^m} \pi(dx)K(x, A)
\]

is called a stationary measure for the Markov chain \((X_t)_{t \geq 0}\) with transition kernel \(K(\cdot, \cdot)\).

In order to examine the question of how “close” a Markov chain is to its stationary distribution after \(t\) steps, it common to look at the total variation distance between \(\pi(\cdot)\) and \(K^t(\cdot, \cdot)\).

**Definition 3.** The total variation distance between \(K^t(\cdot, \cdot)\) and \(\pi(\cdot)\) is given by

\[
\delta(K^t, \pi) = \sup_{x \in \mathbb{R}^m} \sup_{A \in \mathcal{B}(\mathbb{R}^m)} \|K^t(x, A) - \pi(A)\|.
\]

For a given threshold \(\varepsilon\), the Markov chain \((X_t)_{t \geq 0}\) is said to have achieved \(\varepsilon\)-mixing at time \(t\) if \(\delta(K^t, \pi) \leq \varepsilon\).
Definition 4. The mixing time of \((X_t)_{t \geq 0}\) for a given threshold \(\varepsilon\) is given by

\[
\tau_{\text{mix}}(\varepsilon) = \min \left\{ t > 0 : \delta(K^t, \pi) \leq \varepsilon \right\}.
\]

The most common theory-based approach to bounding the mixing time of \((X_t)_{t \geq 0}\) relies on verifying that \((X_t)_{t \geq 0}\) satisfies a minorization condition and an associated drift condition. We define these terms below.

Definition 5. A Markov chain \((X_t)_{t \geq 0}\) satisfies a minorization condition if there exist \(\varepsilon \in (0, 1)\) a set \(C \in \mathcal{B}(\mathbb{R}^m)\), a positive integer \(k\), and a probability measure \(\nu(\cdot)\) such that for all \(x \in C\) and for all \(A \in \mathcal{B}(\mathbb{R}^m)\),

\[
K^h(x, A) \geq \varepsilon \nu(A).
\] (1)

The set \(C\) is called a small set.

Definition 6. A Markov chain \((X_t)_{t \geq 0}\) satisfies a drift condition if there exist constants \(\lambda \in (0, 1)\) and \(b < 1\), a function \(V : \mathbb{R}^m \to [1, \infty)\), a positive integer \(h\), and a small set \(C \in \mathcal{B}(\mathbb{R}^m)\) such that for all \(x \in \mathbb{R}^m\),

\[
K^h V(x) \leq \lambda V(x) + b \|C(x)\), where
\]

\[
K^h V(x) = \mathbb{E}[V(X_{t+h}) | X_t = x] \quad \text{and the expectation is taken with respect to the h-step transition kernel.}
\]

Rosenthal (1995) uses the values of \(\varepsilon, \lambda, \) and \(b\) in Definitions 5 and 6 to derive an upper bound on the total variation distance between the \(n\)-step transition kernel for \((X_t)_{t \geq 0}\) and the stationary measure.

Theorem 1. (Rosenthal, 1995) Assume that for a function \(V : \mathbb{R}^m \to [1, \infty)\), a positive integer \(h\), and constants \(\lambda \in (0, 1)\) and \(b < \infty\), \((X_t)_{t \geq 0}\) satisfies

\[
K^h V(x) \leq \lambda V(x) + b \|C(x)
\]

for all \(x \in \mathbb{R}^m\), where \(C = \{ x : V(x) \leq d \}\) and \(d > \frac{2b}{1-\lambda} - 1\). Assume that for some \(\varepsilon > 0\), some probability measure \(\nu(\cdot)\) on \(\mathcal{B}(\mathbb{R}^m)\), and some positive integer \(k_0\),

\[
K^{hk_0}(x, B) \geq \varepsilon \nu(B)
\]

for all \(x \in C\) and for all \(B \in \mathcal{B}(\mathbb{R}^m)\). Then for any \(r \in (0, 1)\) with \((X_t)_{t \geq 0}\)
beginning in the initial distribution $\Psi$ and for any positive integer $n$,

$$
\delta(K^n, \pi) \leq (1 - \varepsilon)\left[\frac{r_n}{p_{X_0}}\right] + (\alpha A)^{-1} \left(\alpha^{-1} (1 - r_{k_0}) A^\lambda \right)^{\frac{1}{2}}
\times \left(1 + \frac{b}{1 - \lambda} + \mathbb{E}_{\Psi}[V(X_0)] \right), \text{ where}
$$

$$
\alpha^{-1} = \frac{1 + 2b + \lambda d}{1 + d},
A = 1 + 2(\lambda d + b), \text{ and}
$$

[·] denotes the greatest integer function.

At this point, it is required to give a full description of the Metropolis-Hastings sampler. Let $x_0$ be the initial state of a Markov chain $(X_t)_{t \geq 0}$, where $x_0$ lies within the support of the target density $p(\cdot)$. Given the current state $x_t$, the state $X_{t+1}$ is obtained in the following way. Draw a proposal $x^*$ from a density $q(\cdot|x_t)$ that may depend on the current state of the chain. This proposal is accepted as the new state with probability

$$
\alpha(x_t, x^*) = \min \left\{ 1, \frac{p(x^*)q(x_t|x^*)}{p(x_t)q(x^*|x_t)} \right\},
$$

while the proposal is rejected, and $X_{t+1} = x_t$ with probability $1 - \alpha(x_t, x^*)$.

The transition density for this chain is given by

$$
k(x_{t+1}|x_t) = \alpha(x_t, x^*)q(x^*|x_t)
+ \int_{\mathbb{R}^m} \left[1 - \alpha(x_t, x^*)\right] q(x^*|x_t) \, dx^* \delta_{x_t}(X_{t+1}),
$$

where $\delta_{x}(\cdot)$ is the Dirac mass measure concentrated at $x$. A Markov chain that makes transitions in this way is the underlying chain for a Metropolis-Hastings algorithm.

### 3 Bounding the Mixing Time

This section details the process by which we obtain an estimated upper bound on the mixing time of a geometrically ergodic Metropolis-Hastings chain $(X_t)_{t \geq 0}$. Section 3.1 describes the estimation of $\lambda$ and $b$, and Section 3.2 details the estimation of the minorization coefficient $\varepsilon$. To begin, we need to choose a drift function $V(x)$. This requires a bit more experimentation than does choosing a drift function for an RSM or an RWM algorithm, as
corresponding theoretical results that lead to the choice of a drift function for the general Metropolis-Hastings algorithm do not exist as they do for the RSM and RWM algorithm. Suppose that $V(x)$ is a candidate drift function. In many cases, $V(x)$ may depend on the marginal density $m(D)$ of the data. In these cases, it is often required to estimate $m(D)$. Since our small set and our drift function are often functions of the target density, the estimate of the marginal density must be obtained prior to estimating any of the drift or minorization coefficients. The simplest approach is to use a naïve Monte Carlo estimator. This is the approach to marginal density estimation that is used in all of the examples presented in Section 4. Wei and Tanner (1990) do this by presenting the marginal density as an expected likelihood with respect to the prior density. Formally, draw a sample of size $M$ from the prior density. For each sampled value $x_i$, $i = 1, 2, \ldots, M$, compute the likelihood $L(x_i|D)$ based on the data. Then an estimate $\hat{m}(D)$ is obtained by taking

$$\hat{m}(D) = \frac{1}{M} \sum_{i=1}^{M} L(x_i|D).$$

The main thing that needs to be avoided in estimating the marginal density of the data is doing so in a way that relies on the MCMC algorithm for which a bound on the mixing time is being estimated. The approach described above relies on direct sampling from the prior density without any dependence on the MCMC method under study. If the prior density cannot be sampled directly, then one of the many other techniques of marginal density estimation that exist in the literature, including importance sampling, adaptive importance sampling (Oh and Berger, 1989), reverse importance sampling (Gelfand and Dey, 1994), annealed importance sampling (Neal, 1998), or one of several other methods (Chib et al., 1998; Ishwaran et al., 2001; Liang, 2007) may be useful in solving this problem. An exhaustive review of marginal density estimation techniques is provided by Llorente et al. (2020). For the remainder of this discussion, assume that marginal density estimation is necessary, so that the process depends on an estimated drift function $\hat{V}(x)$ that makes use of the estimated marginal density. In all the examples described in Section 4, the prior density can be sampled directly when marginal estimation is required. We are now ready to describe the process of estimating drift and minorization coefficients, and these coefficients will be used in the formula presented in Theorem 1 to obtain a bound on the mixing time. To begin, let $C = \left\{ x : \hat{V}(x) \leq d \right\}$ for a suitably chosen value of $d$. 
3.1 Estimating $\lambda$ and $b$

Recall that the transition density for the underlying Markov chain $(X_t)_{t \geq 0}$ of a Metropolis-Hastings algorithm is given by

$$k(x_{t+1}|x_t) = \alpha(x_t, x^*)q(x^*|x_t)$$

$$+ \left( \int_{\mathbb{R}^m} [1 - \alpha(x_t, x^*)]q(x^*|x_t) \, dx^* \right) \delta_{x_t}(X_{t+1}).$$

Furthermore, for $x \notin C$, 

$$\frac{K\hat{V}(x)}{\hat{V}(x)} \leq \lambda,$$

where $K\hat{V}(x) = \mathbb{E}[\hat{V}(X_{t+1})|X_t = x]$. Consequently, the idea is to estimate $\lambda$ by estimating the integral

$$\frac{K\hat{V}(x)}{\hat{V}(x)} = \int_{\mathbb{R}^m} \hat{V}(x)^* \alpha(x, x^*)q(x^*|x) \, dx^*$$

$$+ \left( \int_{\mathbb{R}^m} [1 - \alpha(x, x^*)]q(x^*|x) \, dx^* \right) \delta_x(X_{t+1}). \quad (5)$$

This integral breaks $\lambda$ estimation into a segment where $x^*$ is possibly accepted as the next state of the chain, and a segment where $x^*$ is possibly rejected. It is possible to simplify estimation of the integral in (5) by considering situations in which $\alpha(x, x^*) \geq 1$. Let

$$\mathcal{A}(x) = \left\{ x^* : \frac{p(x^*)q(x|x^*)}{p(x)q(x^*|x)} \geq 1 \right\}, \text{ and}$$

$$\mathcal{R}(x) = \left\{ x^* : \frac{p(x^*)q(x|x^*)}{p(x)q(x^*|x)} < 1 \right\}.$$ 

Then

$$\frac{K\hat{V}(x)}{\hat{V}(x)} = \int_{\mathcal{A}(x)} \frac{\hat{V}(x^*)}{\hat{V}(x)} q(x^*|x) \, dx^*$$

$$+ \int_{\mathcal{R}(x)} \frac{\hat{V}(x^*)}{\hat{V}(x)} \frac{p(x^*)q(x|x^*)}{p(x)q(x^*|x)} q(x^*|x) \, dx^*$$

$$+ \int_{\mathcal{R}(x)} \left( 1 - \frac{p(x^*)q(x|x^*)}{p(x)q(x^*|x)} \right) q(x^*|x) \, dx^*. \quad (6)$$
The estimation of $\lambda$ proceeds through approximating each of these integrals and adding them. To begin, choose a number $d$ to define the small set $C = \{ x : \hat{V}(x) \leq d \}$. It is advisable to choose $d$ large enough to avoid issues with the requirements of Theorem 1. Next, choose $N_{C'}$ points outside of $C$ by taking random samples $x$ from a diffuse density that is supported on the state space, compute $\hat{V}(x)$, and add $x$ to a sampled set $\hat{C}'$ if $\hat{V}(x) > d$. We choose $N_{C'}$ so that estimates of $\lambda$ are fairly stable. For a given $x_i$, $i = 1, 2, \ldots, N_{C'}$, take $N_0$ samples from $q(\cdot|x_i)$. The value of $N_0$ is chosen so that intermediate estimates of $\lambda$ are fairly stable. For each of the $N_0$ values of $x^*$, compute the ratio $\frac{p(x^*)q(x_i|x^*)}{p(x_i)q(x^*|x_i)}$. For values of $x^*$ where this ratio is at least one, place $x^*$ into the sure acceptance set $\hat{A}(x_i)$. If $x^*$ is such that this ratio is less than one, place $x^*$ into the possible rejection set $\hat{R}(x_i)$. Once these sets have been constructed, compute the sum

$$\frac{\hat{K}\hat{V}(x_i)}{\hat{V}(x_i)} = \frac{1}{N_0} \sum_{x^* \in \hat{A}(x_i)} \hat{V}(x^*) \frac{\hat{V}(x^*)}{\hat{V}(x_i)} + \frac{1}{N_0} \sum_{x^* \in \hat{R}(x_i)} \hat{V}(x^*) \frac{\hat{V}(x^*)}{\hat{V}(x_i)} \frac{p(x^*)q(x_i|x^*)}{p(x_i)q(x^*|x_i)} + \frac{1}{N_0} \sum_{x^* \in \hat{R}(x_i)} \left(1 - \frac{p(x^*)q(x_i|x^*)}{p(x_i)q(x^*|x_i)}\right). \quad (7)$$

This gives an estimate $\hat{\lambda}_{x_i}$ given that $x_i$ is the initial state of the chain. The estimation of $\lambda$ is completed by computing

$$\hat{\lambda} = \max_{i=1,\ldots,N_{C'}} \hat{\lambda}_{x_i}.$$  

At this point, it is important to make some remarks on the selection of the drift function. The choice of the drift function $V$ is not an easy one, and one that may require some experimentation. Unlike for other MCMC algorithms, including the RSM and RWM algorithms, the literature does not contain much guidance on how to select a drift function for general Metropolis-Hastings algorithms. Thus, as was the case while Cowles and Rosenthal (1998) implemented their technique, if it is difficult after having tried a candidate drift function to get an estimate of $\lambda$ that is smaller than one, it is necessary to select a different function as a candidate function and try again to estimate $\lambda$. If a candidate drift function leads to an estimate of
λ that is smaller than one, this provides evidence of a useful drift function. This offers no mathematical guarantee that the drift function that is used is truly a suitable drift function, but it provides compelling evidence of such. The only clear guideline on the selection of a drift function is that the shape of the drift function, in a rough sense, be similar to a reflection of the target density across a horizontal line. This ensures that areas where the drift function is small correspond to regions of high posterior probability. In terms of choosing a value of $d$, it is best initially to select a large value, and then to decrease it if it is too computationally expensive to form a sampled small set for estimation of $b$ or to form a set of sampled values outside of the small set in the estimation of $\lambda$, but to maintain $d$ large enough that the value that $d$ exceeds $\frac{2b}{1-\lambda} - 1$.

The estimation of $b$ is carried out in a way that is very similar to the process used to estimate $\lambda$. The primary difference is in the fact that the main goal is to estimate $K\hat{V}(x)$ from points inside $C$. If $x \in C$, then

$$K\hat{V}(x) \leq \lambda \hat{V}(x) + b.$$ 

Here, we estimate the integral

$$K\hat{V}(x) = \int_{\mathcal{A}(x)} \hat{V}(x^*) q(x^*|x) \, dx^*$$

$$+ \int_{\mathcal{R}(x)} \hat{V}(x^*) \frac{p(x^*)q(x|x^*)}{p(x)q(x^*|x)} q(x^*|x) \, dx^*$$

$$+ \hat{V}(x) \int_{\mathcal{R}(x)} \left( 1 - \frac{p(x^*)q(x|x^*)}{p(x)q(x^*|x)} \right) q(x^*|x) \, dx^*. \quad (8)$$

In order to do this, choose $N_C$ points inside $C$ in a similar fashion to how the points in $\hat{C}$ were chosen in estimating $\lambda$. For each $x_i$ in the sampled set $\hat{C}$, draw $N_1$ samples $x^*$ from $q(\cdot|x_i)$, where $N_1$ is chosen to obtain stability of the estimates of the integral in (8). Construct $\hat{A}(x_i)$ and $\hat{R}(x_i)$ in the same way that they were constructed in the estimation of $\lambda$, and then compute
the sum

\[
\hat{K}\hat{V}(x_i) = \frac{1}{N_1} \sum_{x^* \in A(x_i)} \hat{V}(x^*) \\
+ \frac{1}{N_1} \sum_{x^* \in R(x_i)} \hat{V}(x^*) \frac{p(x^*)q(x_i|x^*)}{p(x_i)q(x^*|x_i)} \\
+ \frac{1}{N_1} \hat{V}(x_i) \sum_{x^* \in R(x_i)} \left(1 - \frac{p(x^*)q(x_i|x^*)}{p(x_i)q(x^*|x_i)}\right).
\]

(9)

This sum gives an estimate \(\hat{K}\hat{V}(x_i)\) of \(\hat{V}(x_i)\) given that the chain is initialized at \(x_i\). To obtain an intermediate estimate \(\hat{b}_{x_i}\) of \(b\) given that \(x_i\) is the initial state of the chain, compute

\[
\hat{b}_{x_i} = \hat{K}\hat{V}(x_i) - \hat{\lambda}\hat{V}(x_i),
\]

and then take

\[
\hat{b} = \max_{i=1,...,N_C} \hat{b}_{x_i}
\]

to obtain a final estimate of \(b\).

3.2 Estimating \(\varepsilon\)

In order to estimate \(\varepsilon\), we make use of a fact that Cowles and Rosenthal (1998) used. If \((X_t)_{t \geq 0}\) satisfies a minorization condition, then

\[
\varepsilon = \int \inf_{x_{t+1} \in C} k(x_{t+1}|x_t) \, dx_{t+1}.
\]

(10)

In order to estimate \(\varepsilon\) conservatively, we use the properties of infima to say that

\[
\inf_{x_{t+1} \in C} k(x_{t+1}|x_t) \geq \inf_{x_t \in C} \alpha(x_t, x^*)q(x^*|x_t) \\
+ \inf_{x_t \in C} \left(\int_{\mathbb{R}^m} [1 - \alpha(x_t, x^*)]q(x^*|x_t) \, dx^* \right) \delta_{x_t}(X_{t+1}) \\
\geq \inf_{x_t \in C} \left(\int_{\mathbb{R}^m} [1 - \alpha(x_t, x^*)]q(x^*|x_t) \, dx^* \right) \delta_{x_t}(X_{t+1}).
\]

(11)

Hence, \(\varepsilon\) is no smaller than the infimum of the rejection probabilities from the initial states inside \(C\). Consequently, we begin by choosing \(N_C\) points
inside $C$ in the same manner as detailed in the estimation of $b$ and so that estimates of $\varepsilon$ are stable. Next, we estimate the rejection probability from each of the $N^\tilde{C}$ points in our sampled set $\tilde{C}$. In order to do this, we run $N_2$ chains from each sampled initial state, where $N_2$ is chosen to ensure stability in estimation of the integral in (11), and we keep track of the proportion of proposals that are rejected. This proportion provides an estimate $\hat{\varepsilon}_{x_i}$ of $\varepsilon$ given that $x_i$ is the initial state of the chain. Repeat this process for each of the $N^\tilde{C}$ initial values. The estimate of $\varepsilon$ is then obtained by taking

$$\hat{\varepsilon} = \min_{i=1,2,\ldots,N^\tilde{C}} \hat{\varepsilon}_{x_i}.$$ 

4 Examples

In this section, we describe four examples of the methods of Section 3 at work. In the first one, we present a univariate example dealing with allele frequency where we use an asymmetric proposal density and then a symmetric proposal density to explore the posterior distribution of the allele frequency. In the second example, we estimate the coefficients of a logistic regression model designed to predict the incidence of coronary heart disease (CHD). The third example was presented by Kamary et al. (2014) and deals with a mixture of geometric and Poisson distributions, and the fourth example examines the performance of the proposed technique with respect to a higher-dimensional example dealing with low birth weight. In each example, we estimate drift and minorization coefficients and use these estimates to bound the mixing time. We provide trace plots for one of these examples to assess the usefulness of the proposed techniques, and we provide stability analysis for our estimates and the computing time for each example.

4.1 Allele Frequency

In a given population, the major allele “0” occurs with probability $p$, and the minor allele “1” occurs with probability $1-p$. In a set of 1,000 individual allele pairs, the goal is to estimate the probability of the major allele. Let $n_0$ denote the number of 00 allele pairs, let $n_H$ denote the number of 01 or 10 pairs, and let $n_1$ denote the number of 11 pairs. The likelihood function for $p$ given our genetic data $D$ is

$$L(p|D) \propto p^{n_H+2n_0}(1-p)^{n_H+2n_1}. $$

Given a lack of prior information about $p$, our prior density is the $U[0,1]$ density. Therefore,

$$\pi(p) = I_{[0,1]}(p).$$
where $\mathbb{I}_{[0,1]}(\cdot)$ is the indicator function. The posterior density is then given by

$$p_D(p) \propto p^{n_H+2n_O}(1-p)^{n_H+2n_1}\mathbb{I}_{[0,1]}(p).$$

We initialize a Metropolis-Hastings algorithm from a $U[0,1]$ distribution. We propose updates to $p$ from a beta density so that our proposed update $p^*$ given the current state $p_t$ is such that

$$p^*|p_t \sim \text{Beta}(150p_t, 28.6p_t).$$

This choice gives an acceptance rate of 24.02%, which is approximately optimal for mixing (Gelman et al., 1997). Typically, we want an acceptance rate between 20% and 40%. In estimating drift and minorization coefficients, we use the drift function

$$\hat{V}(p) = [\hat{p}(p|D)]^{-0.005},$$

where $\hat{p}(\cdot|\cdot)$ is the estimated target density that results from using an estimate of the marginal density of the observed data. We set $d$ at 2,000. In estimating $\lambda$, we choose 200 points outside of $C$, where $C$ is defined as before, and draw 500 samples from the proposal density $q_A(p^*|p_t)$ given each of the initial states. The resulting estimate of $\lambda$ is $\hat{\lambda} = 0.018$. To estimate $b$, we choose 200 points inside $C$ as initial states and draw 500 samples from $q_A(p^*|p_t)$ given each of the initial states. These choices lead to an estimate $\hat{b} = 36.384$ of $b$. To estimate $\varepsilon$, we select 500 points inside $C$ and run 1,000 one-step chains from each. This yields an estimate $\hat{\varepsilon} = 0.087$ of $\varepsilon$, and these values of $\hat{\lambda}$, $\hat{b}$, and $\hat{\varepsilon}$ lead to an upper bound of 3,391 steps on the mixing time of the chain. The trace plot in Figure 1a indicates that this is sufficient as a burn-in time. The estimation procedure took 6.09 seconds of CPU time to complete. To assess stability, we carried out this procedure 50 times. The histograms are in Figure 1b. The empirical standard deviation of $\hat{\lambda}$ is 0.0002, the empirical standard deviation of $\hat{b}$ is 0.0861, and the empirical standard deviation of $\hat{\varepsilon}$ is 0.0054, which gives an empirical standard deviation of 200.675 steps on the mixing time. The value of $\frac{2b}{1-\lambda} - 1$ is 73.12, so $d = 2,000$ is a sufficiently large choice of $d$. We investigated this scenario again using a symmetric proposal $p^*|p_t \sim N(p_t, 0.0016)$. This proposal density gave an acceptance rate of 30.04%. To estimate $\lambda$, $b$, and $\varepsilon$, we used the same drift function and the same choices of tuning parameters $N_{Cv}$, $N_{C}$, $N_{Cv}$, $N_0$, $N_1$, and $N_2$ as in the example using the beta proposal density, but $d$ was set at 5,000 here. The resulting estimates were $\hat{\lambda} = 0.821$, $\hat{b} = 0.176$, and $\hat{\varepsilon} = 0.455$. This gave an upper bound of 1,242 steps on the mixing time. The trace plot in Figure 2a indicates that this is a sufficient burn-in time. The value of $\frac{2b}{1-\lambda} - 1$ is 0.966, so $d = 5,000$ is...
sufficiently large. The process took 21.699 seconds of CPU time to complete.

For this example, despite the fact that we are only estimating one parameter, analytical verification of a drift and minorization condition remains a difficult task. However, since the target distribution is completely known, it is possible to examine quantile-quantile (QQ) plots of the output of each chain to determine whether, after the first number of steps corresponding to their respective recommended mixing times have been discarded as burn-in, the samples approximate the target distribution. Figure 3a shows the QQ plot from the chain with the asymmetric proposal density, and Figure 3b shows the QQ plot from the chain with the random walk proposal, where the theoretical quantiles are those of the Beta(1711, 291) distribution, or our target distribution. Both plots are nearly perfectly linear along the line $y = x$, so this provides compelling evidence that each chain approximately reaches the target distribution in the recommended number of steps, so the bounds on the mixing time provided by each method appear to be sufficient burn-in times for the respective chains. To assess stability of $\lambda$, $b$, and $\varepsilon$, we again carried out the estimation procedure 50 times. The histograms are given in Figure 2b. The empirical standard deviation of $\lambda$ is 0.0062, the empirical standard deviation of $b$ is 0.0061, and the empirical standard deviation of $\varepsilon$ is 0.0073. The empirical standard deviation of the upper bound on the mixing time is 137.922 steps. For the remaining examples, trace plots, as well as the histograms for the stability analysis, are omitted in order to
(a) Trace plot of Metropolis-Hastings chain with Normal proposal. (b) Histograms of $\lambda$, $\hat{b}$, and $\hat{c}$ for the Metropolis-Hastings chain with Normal proposal.

Figure 2: Plots for Allele Frequency Example with Symmetric Proposal Density

(a) QQ Plot of Quantiles of the Asymmetric Chain versus the Random Walk Chain versus the Quantiles of the Target Distribution (b) QQ Plot of Quantiles of the Target Distribution

Figure 3: QQ Plots of the Two Chains
For this example, the asymmetric Metropolis-Hastings chain was prescribed a longer burn-in than was the random walk Metropolis chain for sampling the allele frequencies. This is slightly surprising, but not particularly so, as this chain has a simple, unimodal target density. However, for this example, the process was much faster for estimating $\lambda$, $b$, and $\varepsilon$ than the method designed for RWM samplers. This is attributable to the selection of points outside of $C$ for estimating $\lambda$, as our choice of $d$ for the asymmetric chain is 2,000, versus 5,000 for the random walk chain. If these values were the same for both, the method for the RWM sampler would estimate these coefficients more quickly as there is no requirement to compute the value of the proposal density in the acceptance probability. Both methods perform similarly in terms of stability in estimation, so generally we will lose computing time by using the method proposed in this manuscript. However, in situations with complicated, asymmetric target densities, we should not expect this sort of result. In those cases, the RWM sampler is a poor choice, and mixing is unlikely to occur quickly. Consequently, the result of the method of estimating drift and minorization coefficients for that algorithm gives a much higher bound on the mixing time than the method proposed here. Since these estimates are computed based on proposing a value from the proposal density, it is unlikely that we would obtain stable estimates in these situations as quickly using a method designed for the RWM sampler compared to using a method designed for a more suitable Metropolis-Hastings algorithm.

The take-away is that there is little to be done about the increased computational cost of the proposed method. However, this cost does not seem to be increased enough to outweigh the added stability of the estimates from the asymmetric proposal density or the risk of running a poorly-behaved chain due to the use of an ill-selected proposal density.

We saw here that the symmetric proposal density gives a more stable estimate of $b$, but less stable estimates of $\lambda$ and $\varepsilon$. The symmetric proposal density example took much longer to complete, likely because since the value of $d$ was much larger, it took more time to find points outside of $C$. The symmetric proposal gives a smaller upper bound on the mixing time, but both bounds appear to be sufficient burn-in times for their respective chains.

### 4.2 CHD Example

In this example, we examine a logistic regression model that relates incidence of coronary heart disease to age. The data consist of information on 100 males aged 20 to 69 years. Let $X$ be a $100 \times 2$ matrix in which the first column is a column of ones and the second column is the age. Let $\beta$ be the
vector of regression coefficients $\beta_0$ and $\beta_1$, and let the presence $y_i$ of CHD in individual $i$ be a Bernoulli random variable with success probability $p_i$, where

$$p_i(\beta) = \frac{e^{X_i\beta}}{1 + e^{X_i\beta}},$$

and $X_i$ is the $i^{th}$ row of $X$. Assume a priori that $\beta$ follows a normal distribution with mean vector $\beta_0 = [-5\ 0]^T$ and covariance matrix $\Sigma = \text{diag}(0.25, 0.0625)$. The posterior density here, then, is

$$p(\beta|y) \propto \prod_{i=1}^{100} \left( p_i(\beta)^{y_i}(1 - p_i(\beta))^{1-y_i} \right) e^{-(\beta_0+5)^2 - 8\beta_1^2}.$$ 

The Metropolis-Hastings algorithm that explores this density proposed an update $\beta'$ given $\beta$, is the current state of the chain from a normal distribution with mean vector $\beta_t$ and covariance matrix $\text{diag}(0.04, 0.0036)$. This proposal density gave an acceptance rate of 20.02%. In order to estimate $\lambda$, $b$, and $\varepsilon$, we used the drift function $\hat{V}(\beta) = [\hat{p}(\beta|y)]^{-0.01}$. To specify our small set, $d$ was set at 40. In estimating $\lambda$, we chose 200 points outside $C$ and proposed 500 updates from each. This led to an estimate $\hat{\lambda} = 0.817$. To estimate $b$, we selected 200 points in $C$ and proposed 500 updates from each of them. This gave an estimate $\hat{b} = 0.609$ of $b$. In estimating $\varepsilon$, 500 points inside $C$ were chosen as initial states, and 1,000 one-step chains were run from each of them. This gave an estimate $\hat{\varepsilon} = 0.445$ for $\varepsilon$, which resulted in an upper bound of 422 steps on the mixing time. The process took 29.866 seconds to complete. The value of $\frac{2b}{1-\lambda} - 1$ is 5.64, so $d = 40$ is sufficiently large. Trace plot examination indicates that 422 is a sufficient burn-in time for this chain. Running this procedure 50 times gave empirical standard deviations of 0.0054 on $\hat{\lambda}$, 0.114 on $\hat{b}$, 0.0058 on $\hat{\varepsilon}$, and 20.984 steps on the upper bound on the mixing time.

### 4.3 Mixture of Poisson and Geometric Distributions

This example examines a chain studied by Kamary et al. (2014), where the goal is to estimate the parameters of a mixture of a Poisson and a geometric distribution with the same mean $\lambda$. Let $\alpha$ be the mixture parameter, so that for observations $x_1, \ldots, x_n$, where $n = 123$, the likelihood is given by

$$L(\alpha, \lambda|x) = \prod_{i=1}^{123} \left[ \alpha \frac{\lambda^{x_i} e^{-\lambda}}{x_i!} + (1 - \alpha) \lambda^{x_i} (1 + \lambda)^{-(x_i+1)} \right].$$
Assume that the prior distributions are specified so that

\[
\pi(\lambda) \propto \frac{1}{\lambda} \quad \text{and} \quad \pi(\alpha) \propto [\alpha(1 - \alpha)]^{-0.5}
\]

for \( \lambda > 0 \) and \( 0 < \alpha < 1 \). The posterior density is given by

\[
p(\lambda, \alpha|x) \propto \prod_{i=1}^{123} \left[ \frac{\lambda^{x_i} e^{-\lambda}}{x_i!} + (1 - \alpha)\lambda^{x_i}(1 + \lambda)^{-(x_i+1)} \right] e^{-(\beta_0+5)^2-8\beta_1^2} \times \frac{1}{\lambda\sqrt{\alpha(1 - \alpha)}}.
\]

The chain proposes updates \((\alpha^*, \lambda^*)\) given \((\alpha_t, \lambda_t)\) is the current state of the chain by drawing \(\alpha^*\) from a Beta density with parameters \(\alpha_t + 1\) and \(2 - \alpha_t\) and choosing \(\lambda^*\) from a lognormal density with mean parameter \(\log(\lambda_t)\) and variance parameter \(1+(\log(\lambda))^2\). This proposal gives an acceptance rate of 24.31\%. To estimate drift and minorization coefficients, we use the drift function \(\hat{V}(\alpha, \lambda) = [\hat{p}(\alpha, \lambda)]^{-0.01}\), and our \(d\) value for constructing \(C\) is 1,000. To estimate \(\lambda\), we chose 400 points outside of \(C\) and proposed 1,000 updates from each. This resulted in a value of \(\hat{\lambda}\) equal to 0.654. For estimation of \(b\), we took 1,500 points inside \(C\) and proposed 2,500 updates from each. The resulting estimate of \(b\) was \(\hat{b} = 2.841\). We ran 2,500 one-step chains from each of 1,500 points inside \(C\) to estimate \(\varepsilon\), which resulted in an estimate \(\hat{\varepsilon} = 0.2675\). The estimated upper bound on the mixing time was 883 steps, and trace plot examination indicates that this is sufficient. The value of \(\frac{2\hat{b}}{1-\hat{\lambda}} - 1\) is 15.409, so \(d = 1,000\) is sufficiently large. The process took 203.009 seconds of CPU time to complete. After repeating the process 50 times, we found empirical standard deviations of 0.0095 for \(\hat{\lambda}\), 0.6479 for \(\hat{b}\), and 0.0181 for \(\hat{\varepsilon}\), leading to an empirical standard deviation of 62.768 steps for the upper bound on the mixing time.

### 4.4 Low Birth Weight Example

The following example is designed to evaluate the performance of the proposed technique in higher-dimensional state spaces. In this example, we examine a logistic regression model designed to relate incidence of low birth weight to several predictors related to the mother, including age, whether they smoke, and seven others. Let \(x_i\) denote the \(i^{th}\) row of the design matrix \(X\), and let \(\beta\) be the vector of regression coefficients. The response vector \(y\) is a vector of 0s and 1s, where \(y_i = 0\) if newborn \(i\) weighs 2,500 grams or
more, and \( y_i = 1 \) otherwise. The random variable \( Y_i \) follows a Bernoulli distribution with probability \( p_i(\beta) \), where

\[
p_i(\beta) = \frac{e^{x_i \beta}}{1 + e^{x_i \beta}}.
\]

A priori, \( \beta = [\beta_0 \quad \beta_1 \quad \ldots \quad \beta_9]^T \) is assumed to follow a normal distribution with mean vector

\[
\beta_0 = [1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0.5 \quad 2 \quad 0.75 \quad 0]^T
\]

and covariance matrix

\[
\Sigma = \text{diag}(0.25, 0.0625, 0.0625, \ldots, 0.0625)
\]

The target density, then, is given by

\[
p(\beta | y) \propto \left( \prod_{i=1}^{n} (p_i(\beta))^{y_i}(1 - p_i(\beta))^{1-y_i} \right) e^{-\frac{1}{2}(\beta - \beta_0)^T \Sigma^{-1}(\beta - \beta_0)}.
\]

If \( \beta_t \) is the current state of the chain, an update \( \beta^* \) is proposed from a normal density with mean vector \( \beta_t \) and covariance matrix

\[
\Sigma^* = \text{diag}(0.0004, 0.0001, 0.0001, \ldots, 0.0001).
\]

This proposal density gave an acceptance rate of 25.12%.

To estimate drift and minorization coefficients, we use the drift function \( \hat{V}(\beta) = [\hat{p}(\beta | y)]^{-0.005} \) and specify our small set by selecting \( d = 1,000 \). To estimate \( \lambda \), we chose 500 points outside of \( C \) and proposed 200 updates from each of them. This gave an estimate of \( \lambda \) equal to \( \hat{\lambda} = 0.8309 \). In estimating \( b \), 500 points in \( C \) were chosen and 500 updates were proposed from each of them. The resulting estimate of \( b \) was \( \hat{b} = 0.389 \). For estimation of \( \epsilon \), 500 points were chosen from inside \( C \) and 1,000 one-step chains were run from each. This yielded an estimate \( \hat{\epsilon} = 0.457 \) of \( \epsilon \), and this combination of estimates gave an approximate upper bound of 4,072 steps on the mixing time. Trace plot examination indicates that this is a sufficient burn-in time for the chain. The value of \( \frac{2b}{1-\lambda} - 1 \) is 2.854, so \( d = 1,000 \) is sufficiently large.

The process took 100.176 seconds to complete. Repeating this process 50 times gave empirical standard deviations of 0.0071 for \( \hat{\lambda} \), 0.035 for \( \hat{b} \), and 0.005 for \( \hat{\epsilon} \), resulting in an empirical standard deviation for the upper bound on the mixing time of 67.317 steps.
5 Concluding Remarks

In this manuscript, we extended the methods of Spade (2020) to construct a technique for bounding the mixing time of geometrically ergodic Metropolis-Hastings algorithms. As is seen in Section 3, the method presented by Spade (2020) for RWM samplers extends in a straightforward way to general Metropolis-Hastings algorithms. The additional difficulty here is the lack of theoretical results pertaining to Metropolis-Hastings algorithms in general that aid in the selection of a useful drift function. That is a difficulty whose resolution is beyond the scope of this paper, but if one were to choose a drift function that results in an estimate of $\lambda$ that is at least one, a new choice of drift function needs to be made, and the procedure needs to be run again with the new drift function.

Prior to this work, assessment of convergence for Metropolis-Hastings algorithms was limited to finding drift and minorization coefficients analytically, relying on the output of the chain, or using the Cowles and Rosenthal (1998) method. The Cowles and Rosenthal (1998) auxiliary simulation approach, in principle, is a useful intermediary between analytical and ad hoc convergence assessment methods. The major issue with this approach, however, is that it does not perform efficiently for moderate-to-high dimensional state spaces. The reason for this is that their method of estimating $\varepsilon$ requires dividing the state space into bins that are small enough to ensure that the transition density is roughly constant over each bin. Thus, in the dimension of the state space, the number of bins increases exponentially. This means that the number of chains that need to be run from each initial state will also increase exponentially in order to provide adequate coverage of the bins to ensure stability in estimation of $\varepsilon$. This issue makes the Cowles and Rosenthal (1998) method extremely computationally expensive in moderate dimensions, and prohibitively so in higher dimensions. The method described in Section 3 suffers much less from this difficulty as it exploits the structure of the Metropolis-Hastings transition kernel to obtain fairly stable estimates of $\varepsilon$ in a much more computationally efficient way. In terms of estimating $\lambda$ and $b$, there is little difference in computational efficiency between the two methods. The method described in Section 3 will, of course, run into computational issues at some point as the dimension of the state space increases, but it will not happen as quickly as it does with the Cowles and Rosenthal (1998) method.

The computational efficiency of the proposed method is not tied entirely to the dimension of the state space. There will be instances in which the estimation of drift and minorization coefficients can be carried out efficiently.
and so that estimates of these coefficients are stable in higher dimensions. This was seen in Section 4.4 with the logistic regression model for low birth weight, where the state space was ten-dimensional. The computational cost of obtaining estimates of these coefficients was 100.176 seconds. This is a fairly reasonable computational cost for a ten-dimensional state space. We see that from the one-dimensional allele frequency example of Section 4.1 to the two-dimensional CHD example of Section 4.2, the process took 29.866 seconds. These three examples show the expected increase in computing time as the dimension of the state space increases. However, the behavior of the target density also impacts the computational time. We see this in the mixture of Poisson and geometric random variables in Section 4.3. The other three target densities are unimodal, non-mixture densities. Mixture distributions take more sample points to give stable estimates of drift and minorization coefficients because we need to be sure that adequate samples are being drawn from all parts of the state space. We see in Section 4.3 that doing this incurs a substantial computational cost, as the estimation process took 203.009 seconds to complete, more than twice the time the estimation took in the ten-dimensional example. This time would likely increase exponentially if the number of parameters in the model were to increase. The takeaway is that while higher dimensionality of the state space naturally increases the time the process takes to complete, this seems to be a much greater concern with mixture and multimodal target densities than it is with unimodal, non-mixture distributions. One observation that arose from the work presented here was that, at least in the examples presented in Section 4, there was a tendency to require more initial states and/or a larger number of proposed updates or chains to be run from each one in order to ensure stability in estimates of the mixing time when asymmetric proposals are used than when symmetric ones are used. There is little reason to suspect that this might be the case in general, but it happened to be the case here. In Section 4.3, however, this was not surprising. The fact that we have the mixture means that the drift function will have multiple valleys, and constructing a small set will require obtaining adequate samples from each region of the drift function.

One point that bears mention here is that, in the estimation of the mixing time, the bound presented by Rosenthal (1995) is quite sensitive to the value of \( r \), and the set of initial states, naturally, is dependent on the choice of \( d \). However, if \( d \) is chosen large enough, then larger choices of \( d \) will have little impact on the estimates. Where larger choices of \( d \) will cause issues is in selecting a set of initial values outside \( C \) to estimate \( \lambda \), as the set \( C \) is larger. Thus, it will take more time to find initial states outside this set.
While the theory dictates that the mixing time can be bounded for any $d$ that satisfies the condition specified by Rosenthal (1995) and any value of $r$ between 0 and 1, different choices of these parameters may result in varying bounds on the mixing time. This is an issue that comes up in all computational approaches to bounding the mixing time by estimating drift and minorization coefficients, and to this point, there appears to be little that can be done to help it. In some of the examples, the bound on the mixing time was a much larger number of steps than what was indicated by trace plot examination to be an adequate burn-in. This is an issue inherent to using the result of Theorem 1, as the bound given there is known to be very conservative. This is a concern with all of the intermediary methods listed in Section 1 as well. Also inherent to both methods is the lack of a mathematical guarantee that the choice of $V$ gives a suitable drift function. Because of this, it is not possible to provide any mathematical guarantee that the upper bound on the mixing time that results from the use of this method is a genuine one. However, if the tuning parameters are chosen in such a way that the estimates of $\lambda$, $b$, and $\varepsilon$ are stable and so that the estimate of $\lambda$ is smaller than one, this provides very compelling evidence that $V$ is a suitable drift function and that the resulting upper bound on the mixing time is very likely a genuine bound. This allows sampling from the output of the chain after it has been run for the number of steps specified by the resulting bound on the mixing time with a high level of confidence that these samples are approximate samples from the desired distribution. The choice of $V$ can have a substantial impact on the estimates of the drift and minorization coefficients, but Rosenthal (1995) guarantees that any function $V$ that can be used as a drift function will yield a true upper bound on the mixing time.

While there is some error in estimation of the drift and minorization coefficients, the bound given in (4) is conservative enough that if the estimates of these coefficients are stable, the resulting error in the estimated upper bound is unlikely to be so great that it suggests an inadequate number of steps. Furthermore, the bound presented by Rosenthal (1995) is designed to be a bound on the mixing time regardless of where the chain is initialized. In practice, initialization of the chain is done using information about the problem at hand rather than simply selecting an initial state at random. Thus, it is highly unlikely that the chain would be initialized from one of the points that would cause the upper bound provided by Rosenthal (1995) to be as conservative as it is. This consideration also lends credence to the genuineness of the upper bound on the mixing time that results from the use of the method described in Section 3.
The proposed technique is not without its limitations. First, the proposed method of estimating $\varepsilon$ only works for a one-step minorization condition. The mathematical construction, and consequently, the computational construction, becomes very complicated as the number of steps for which we aim to verify a minorization condition increases. This is an issue that the Cowles and Rosenthal (1998) method is designed to handle well. The second major limitation of this method is that it does not extend in any obvious way to other MCMC methods. Specifically, this method depends on the rejection of some proposals in order to estimate the minorization coefficient. Since the Gibbs sampler is a special case of the Metropolis-Hastings algorithm in which all proposals are accepted, the proposed method will not work for Gibbs sampling algorithms. Until a technique for doing this more efficiently for Gibbs samplers is constructed, the Cowles and Rosenthal (1998) method remains the only method available for estimating drift and minorization coefficients. Despite these limitations, the proposed method does provide upper bounds on the mixing time of other Metropolis-Hastings algorithms in fairly computationally efficient way. As a result, it represents an important advance in bounding the mixing times of Markov chains effectively, efficiently, and without a need to rely on the output of only a single chain.

References


University of Wisconsin–Milwaukee
Department of Mathematical Sciences
3200 N. Cramer Street
EMS E403
Milwaukee, WI, USA 53211
E-mail: spade@uwm.edu