Understanding Confidence Intervals
in
Adaptive Markov Chain Monte Carlo*

Submitted by:

Arkajyoti Bhattacharjee †
Department of Mathematics & Statistics
Indian Institute of Technology, Kanpur

Supervised by:

Dr. Dootika Vats
Department of Mathematics & Statistics
Indian Institute of Technology, Kanpur

Submitted on:

18^{th} November, 2021

Abstract

In this report, we attempt to understand the problems in asymptotic variance estimation for Adaptive Markov Chain Monte Carlo (AMCMC) and the role of confidence intervals in providing consistent estimation procedures for the asymptotic variance. The report is primarily based on Atchadé (2012).

Contents

*This report has been prepared towards the fulfillment of the requirements of the Departmental Elective course MTH598A - M.Sc. Project - I

†Roll Number: 201277, Program: M.Sc. (2-yr) Statistics

1 Introduction

Over the course of the 21^{st} century, the use of Markov Chain Monte Carlo (MCMC) algorithms have grown exponentially, with manifold applications in astronomy (Thrane and Talbot (2019), Sharma (2017)), health science (Sorensen et al. (2002), Vajargah et al. (2021)), cognitive science (Kim et al. (2003)), image compression, optimization (Mahendran et al. (2012), Ma et al. (2015)) and machine learning (Andrieu et al. (2003), Hensman et al. (2015)). It is a sampling methodology widely used in estimating expected values under complicated and high-dimensional distributions, which are known up to a normalizing constant (see Brooks et al. (2011), Liu and Liu (2001), Gilks et al. (1995)).

MCMC consists of two parts - *Markov chain* and *Monte Carlo*. The *Monte Carlo* methods are a broad class of computational algorithms used to compute closed form analytical solutions of complicated numerical integrals. For example, we may be interested in obtaining an analytical solution of

$$\int_{\pi}^{2\pi} e^{sin(log(x))cos(e^x)} dx.$$

Clearly, finding a closed form solution of this integral is difficult as no standard anti-derivative exists of the integrand. A Monte Carlo approach to this problem is to sample a large number of $\mathscr{U}(\pi, 2\pi)$ variables and compute the above integral as an expectation under the uniform distribution. Mathematically,

$$\int_{\pi}^{2\pi} e^{sin(log(x)cos(e^x))} dx = \pi \mathbb{E}_{\mathscr{U}}(e^{sin(log(X))cos(e^X)}) \hat{=} \frac{1}{N} \sum_{i=1}^{N} e^{sin(log(X_i))cos(e^{X_i})},$$

where $X_1, ..., X_N$ is a random sample drawn from $\mathscr{U}(\pi, 2\pi)$, $\mathbb{E}_{\mathscr{U}}$ is expectation under $\mathscr{U}(\pi, 2\pi)$ and "$\hat{=}$" means 'is estimated by'. The right hand side of the above equation holds due to the weak law of large numbers, assuming N is large enough. The Monte Carlo approach easily provides 3.16 an 'estimated' solution of the otherwise intractable integral. Although other numerical integration techniques may be able to provide an 'approximate' solution, difficulty increases as the dimensionality increases, and MCMC is often the better alternative.

The *Markov chain* part of MCMC uses the Markovian property, which means that the proposed random value depends on the current value and not on the previous values of the sequential process (hence, 'chain').

MCMC is extensively used in Bayesian inference as it involves generating samples from complicated posterior distributions where the exact form of the likelihood is unknown or difficult to derive analytically. One of most popular class of MCMC algorithms is the class of *Metropolis-Hastings* (MH) algorithms (Metropolis et al. (1953), Hastings (1970), Chib and Greenberg (1995), Robert and Casella (1999)). Let $f(.)$ be the target density function. The aim of the MH algorithm is to propose values from a proposal distribution $Q(x,.)$, x being the current state of the Markov chain, and store them sequentially in a Markov chain if they are accepted with acceptance probability $\alpha(x,y) = min\{1, \frac{f(y)q(y,x)}{f(x)q(x,y)}\}$. $Q(x,.)$ is chosen such that it is the invariant distribution of f. The MH algorithm is given by in Algorithm 1.

Algorithm 1 Metropolis-Hastings algorithm

Let $X_n = x$. To obtain X_{n+1}:

1. $Y \sim Q(x,.)$ and independently $U \sim \mathscr{U}(0,1)$.
2. If $U < \alpha(x,y) = min\{1, \frac{f(y)q(y,x)}{f(x)q(x,y)}\}$,
   set $X_{n+1} = y$.
3. Else
   set $X_{n+1} = x$.

It is to be noted that $q(x,.)$ in Algorithm 1 is the proposal density corresponding to $Q(x,.)$.

Now, given a family of proposal distributions for a given target, determining the optimal proposal distribution is essential and maybe difficult. One naive method is to use 'trial and error' to tune the associated parameters in the proposal variance to achieve an optimal acceptance probability (see Besag and Green (1993), Besag et al. (1995), Gelman et al. (1997)). For example in Figure 1, we can see that having small variance may lead to high acceptances, but it limits the state space exploration of the Markov chain and leads to highly correlated samples; having high variance leads to more rejections, but allows more state space exploration and provides less correlated samples; an optimal variance allows efficient mixing of the Markov chain and provides lesser correlated samples.

This becomes increasingly difficult and practically impossible, both in terms

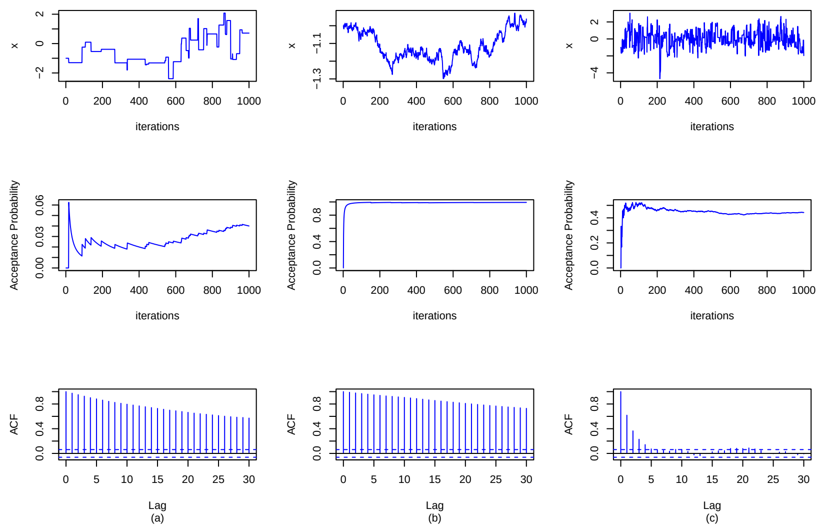

Figure 1: Plots showing how proposal variance effects the samples obtained via Random-walk MH algorithm, wherein the proposal variance σ^2 is (a) high, (b) low, and (c) optimal for $\mathcal{N}(\text{current state}, \sigma^2)$ proposal with target as $\mathcal{N}(0,1)$.

of implementation time as well as the amount of human intervention involved, as the dimension of the target distribution increases. Further, this in not a viable solution to find more complicated improvements like making the associated proposal variance matrix Σ_n approximately proportional to the target covariance matrix Σ (see Roberts and Rosenthal (2001)). One solution is to use *Adaptive Markov Chain Monte Carlo* (AMCMC). AMCMC algorithms automatically 'learn' better parameter values 'on the fly' i.e. while the algorithm runs (see Haario et al. (2001), Roberts and Rosenthal (2009), Chimisov et al. (2018)). A popular class of AMCMC algorithms is the *Adaptive Metropolis-Hastings* algorithm, given in Algorithm 2.

The difference between ordinary MCMC and AMCMC lies in their proposal kernels. In ordinary MCMC, the proposal distribution is kept fixed; in a corresponding AMCMC algorithm, at each iteration n, the proposal distribution $Q_n(x,.)$ changes, as is evident in Algorithm 2.

Another advantage of using AMCMC is observed in multi-modal distributions.

Algorithm 2 Adaptive Metropolis-Hastings algorithm

Let $X_n = x$. To obtain X_{n+1}:

1. $Y \sim Q_n(x,.)$ and independently $U \sim \mathscr{U}(0,1)$.
2. If $U < \alpha(x,y) = min\{1, \frac{f(y)q_n(y,x)}{f(x)q_n(x,y)}\}$,
   set $X_{n+1} = y$.
3. Else
   set $X_{n+1} = x$.

Craiu et al. (2009) show that AMCMC algorithms outperforms usual MCMC algorithms in case of multi-modal targets or when the targets have local properties. Furthermore, AMCMC algorithms can perform significantly better in cases of high dimensional targets with highly correlated structure. In such situations, a fixed kernel in ordinary MCMC algorithms will result in larger rejections. On the contrary, AMCMC algorithms will try to learn the structure of the target and adapt the proposal kernel to make intelligent moves (Roberts and Rosenthal (2009), Mallik and Jones (2017)).

After obtaining samples via MCMC algorithms, we are usually interested in knowing how good the samples are. Suppose θ is the parameter of interest and we estimate it, using samples $X_1,...X_n$, obtained via an MCMC algorithm, through $\hat{\theta}(X_1,...X_n)$. To assess the 'goodness' of this estimate, we look into the Monte carlo error or equivalently, the effective sample size. Alternatively, confidence intervals can be formed and their width can be used as a stopping rule. However, in AMCMC, problems arise with the ergodicity of the algorithms and furthermore, the process remains no longer Markovian.

In Section 2, we discuss about ergodicity in AMCMC and the sufficient conditions under which ergodicity holds. In Section 3, we discuss how adaptation leads to problems in asymptotic variance estimation in AMCMC. In Section 4, we present the main results of Atchadé (2012) which theoretically justifies a Central Limit Theorem in AMCMC, under certain conditions. We verify, using simulation, the theorems of Section 4 using two examples in Section 5.

2 Ergodicity

Let π be the probability measure of interest on some measure space $(\mathscr{X}, \mathscr{B})$. $\{P_\theta\}_{\theta\in\Theta}$ is a family of Markov transition kernels on $(\mathscr{X}, \mathscr{B})$, for some measurable space $(\Theta, \mathscr{A})$, where the map $(x,\theta) \mapsto P_\theta(x,.)$ is $(\mathscr{B}\times\mathscr{A})$-measurable. We assume that for each P_θ, π is the unique invariant distribution. We let

$$\mathscr{F}_n \stackrel{def}{=} \sigma(X_0,...,X_n,\theta_0,...,\theta_n)$$

be the filtration generated by the random process $\{(X_n,\theta_n)\}$ with values in $\mathscr{X}\times\Theta$. Then,

$$P_\theta(x,A) \stackrel{def}{=} \Pr(X_{n+1}\in A|X_n = x, \theta_n = \theta, \mathscr{F}_{n-1}), \quad A\in\mathscr{B},\ x\in\mathscr{X},\ \theta\in\Theta \qquad (1)$$

for $n = 0,1,2,....$ The conditional distribution of θ_{n+1} given $\mathscr{F}_n$ and X_{n+1} depends on the adaptive algorithm chosen. The marginal sequence $\{X_n\}_{n\geq 0}$ is called the *adaptive Markov chain*. We note that if $\theta_n = \theta$, then there is no adaptation and $\{X_n\}_{n\geq 0}$ becomes an ordinary Markov chain.

If θ_n is independent of X_n for all n, then the adaptive Markov chain algorithm is called an *independent adaptation*. For independent adaptations, π-stationarity is guaranteed (Proposition 1, Roberts and Rosenthal (2007)). However, irreducibility may be destroyed (Example 1, Roberts and Rosenthal (2007)).

Again, if the adaptation in an adaptive Markov chain algorithm is stopped after a finite time and usual MCMC is run, then the adaptation is called *finite adaptation* (see Pasarica and Gelman (2010)). In this case, if each individual Markov kernel P_θ is ergodic, then the finite adaptation MCMC algorithm is also ergodic (Proposition 2, Roberts and Rosenthal (2007)).

Greater interest, thus, is in the case of *infinite, dependent adaptation*. In such cases, the pair sequence $\{(X_n,\theta_n)\}_{n\geq 0}$ is usually Markovian and the corresponding algorithm is called *Markovian adaptation*. Here, although each individual Markov kernel $\{P_\gamma\}$ is π-invariant, the adaptive algorithm may not be converge to π. For a counterexample, we refer the reader to Andrieu and Thoms (2008), Atchadé and Rosenthal (2005), Roberts and Rosenthal (2009), Roberts and Rosenthal (2007).

So, we require conditions to guarantee convergence in distribution of $\{X_n\}_{n\geq 0}$

to π. Ergodicity properties of adaptive MCMC under various assumptions have been proved (see Andrieu and Moulines (2006), Roberts and Rosenthal (2007)).

2.1 Sufficient conditions for ergodicity in AMCMC

Roberts and Rosenthal (2007) proved that ergodicity of the adaptive algorithm i.e.

$$\lim_{n\to\infty} \sup_{A\in\mathscr{B}} ||P_{\theta}^{n}(x,A)-\pi(A)|| = 0 \ \ \forall\, x \in \mathscr{X},$$

and also, the Weak Law of Large Numbers, i.e.

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} g(X_i) = \pi(g) \stackrel{def}{=} \int g(x)\pi(dx)$$

for all bounded $g : \mathscr{X} \to \mathscr{R}$ holds, assuming only *Diminishing (a.k.a. Vanishing) Adaptation* condition

$$\lim_{n\to\infty} \sup_{x\in\mathscr{X}} ||P_{\theta_{n+1}}(x,.)-P_{\theta_n}(x,.)|| = 0 \ \text{ in probability,} \tag{2}$$

and also the *Containment (a.k.a Bounded convergence)* condition

$$\{M_{\varepsilon}(X_n,\theta_n)\}_{n=0}^{\infty} \ \text{ is bounded in probability } \forall\, \varepsilon > 0, \tag{3}$$

where

$$M_{\varepsilon}(x,\theta) \stackrel{def}{=} \inf_{n\geq 1} ||P_{\theta}^{n}(x,.)-\pi(.)| \leq \varepsilon$$

is the convergence time of the kernel P_{θ}, when starting from $x \in \mathscr{X}$.

Unlike the adaptive Metropolis algorithm of Haario et al. (2001), the adaptive parameters of an AMCMC algorithm may not converge to a deterministic limit, but rather to a random limit. We show this using the following example.

Example 1: We consider the the target density $\pi(x) = \frac{1}{2}\mathbf{1}_D(x)$, where $D = [\frac{-7}{4}, \frac{-3}{4}] \cup [\frac{3}{4}, \frac{7}{4}]$. We use an adaptive Random Walk Metropolis algorithm with uniform proposal $\mathscr{U}(x-\theta_n, x+\theta_n)$, where x is the current step at the n^{th} iteration and $\theta_n > \frac{3}{2}$ for each iteration n. The algorithm is described in Algorithm 3. We have taken the positive constant $c_0 = 10$ and the constants a and A satisfy $0 < a < A < 1$

and the min and max parts of the update ensures that the adaptive parameter remains within the compact set $[a,A]$. We tune θ_n to achieve approximately 30% acceptance probability. Figure 2 shows that this probability is achieved by two values of θ i.e. $\theta_n \to \theta_*$ where θ_* takes two values.

Algorithm 3 Adaptive Random-walk Metropolis-Hastings algorithm with Uniform proposals

---

Let $X_n = x$. To obtain X_{n+1}:

1. $Y \sim \mathscr{U}(x-\theta_n, x+\theta_n)$ and independently $U \sim \mathscr{U}(0,1)$.
2. If $U < \alpha(x,y) = min\{1, \frac{\pi(y)}{\pi(x)}\}$,

   set

   $X_{n+1} = y$,
   $\theta_{n+1} = \max(a, \min(\theta_n + \frac{c_0}{n}(1-0.3), A))$.
3. Else

   set

   $X_{n+1} = x$,
   $\theta_{n+1} = \max(a, \min(\theta_n + \frac{c_0}{n}(0-0.3), A))$.

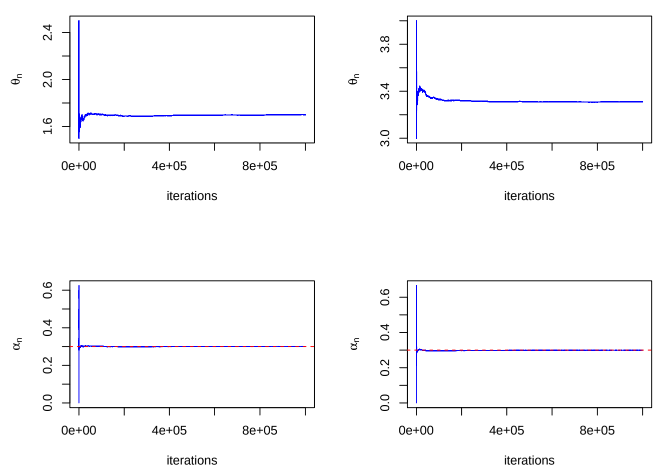

Figure 2: The top row plots show the sample path of θ_n through iteration n. The bottom row plots show the acceptance probability at iteration n. The red line indicates 0.30 acceptance probability.

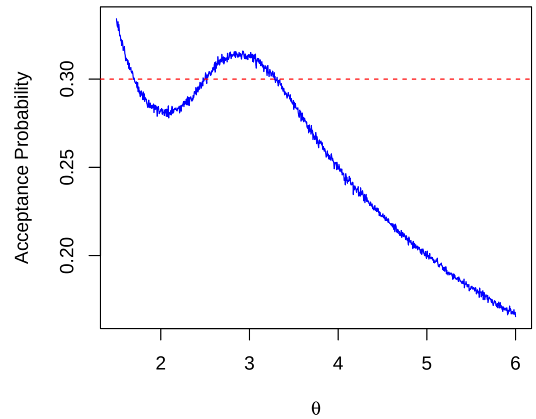

Figure 3: Plot of acceptance probability against θ.

In fact, θ_n can converge to multiple values of θ, depending on the starting value, for a fixed acceptance probability close to 30%, as is evident in Figure 3. Problems in asymptotic variance estimation arises because of the convergence of the adaptive parameter to a random limit, which we discuss it in the next section.

3 Asymptotic variance estimation

Suppose we are interested in estimating $\pi(h) \overset{def}{=} \int_{\mathscr{X}} h(x)\pi(dx)$. The Monte Carlo estimator of $\pi(h)$ is $\hat{\pi}_n(h) \overset{def}{=} \frac{1}{n}\sum_{i=1}^{n} h(X_i)$, where $h : \mathscr{X} \to \mathscr{R}$. Now, to assess the goodness of this estimator, we look into the Mean Square Error (MSE) of $\hat{\pi}_n(h)$. Since MSE = bias^2 + variance, we look into the bias and variance of $\hat{\pi}_n(h)$ in estimating $\pi(h)$. The bias of $\hat{\pi}_n(h)$ satisfies $\mathbb{E}(\hat{\pi}_n(h) - \pi(h)) = o(n^{-1/2})$ in most practical situations i.e. $\hat{\pi}_n(h)$ is consistently unbiased for $\pi(h)$ (Atchadé (2012)). The variance of $\hat{\pi}_n(h)$ is such that $nVar(\hat{\pi}_n(h))$ converges to a limit called the *asymptotic variance* of h. For a Markov chain with transition kernel P, the asymptotic variance is given by

$$\sigma_P^2(h) \overset{def}{=} \sum_{l=-\infty}^{+\infty} \gamma_l(P,h), \tag{4}$$

where for $l \geq 0$,

$$\gamma_l(P,h) \stackrel{def}{=} \int (h(x) - \pi(h)) P^l h(x) \pi(dx), \text{ and } \gamma_{-l}(P,h) = \gamma_l(P,h).$$

For estimating $\sigma_P^2(h)$, we consider lag-window estimators of the form

$$\Gamma_n^2(h) \stackrel{def}{=} \sum_{k=-n+1}^{n-1} w(kb_n)\gamma_{n,k}, \tag{5}$$

which is a weighted average of the k^{th}-order sample auto-covariances $\gamma_{n,k}$ of $\{h(X_n)\}_{n\geq 0}$. More precisely, for $0 \leq l \leq n-1$,

$$\gamma_{n,l} \stackrel{def}{=} \frac{1}{n}\sum_{j=1}^{n-l}(h(X_j) - \hat{\pi}_n(h))(h(X_{j+l}) - \hat{\pi}_n(h)), \text{ and } \gamma_{n,-l} = \gamma_{n,l}.$$

$\{b_n\}_{n\geq 1}$, called the *bandwidth*, is a non-increasing sequence of integers such that $b_n \downarrow 0$, and $w : \mathscr{R} \to \mathscr{R}$, with support $[-1,1]$, is an even weight function (i.e. $w(x) = w(-x)$).

Once we can estimate $\sigma_P^2(h)$ by $\Gamma_n^2(h)$, we can then estimate the standard error of $\hat{\pi}_n(h)$. As an alternative, we can form a confidence interval for $\pi(h)$ using $\hat{\pi}_n(h) \pm z_\alpha \sqrt{\Gamma_n^2(h)/n}$, where z_α is the appropriate quantile of the standard normal distribution and see if

$$P\left(\pi(h) \in \left(\hat{\pi}_n(h) \pm z_\alpha \sqrt{\Gamma_n^2(h)/n}\right)\right) = 1 - \alpha.$$

All this is common practice in MCMC backed by the fact that for $c_n(:= 1/b_n) = o(n)$, and under some regularity conditions (e.g. geometric ergodicity and existence of (2+ε)-moment for h under π), $\Gamma_n^2(h)$ converges in probability to $\sigma_P^2(h)$ (Damerdji (1995); Flegal and Jones (2010);Atchadé (2011)).

For AMCMC, the behaviour of the asymptotic variance is not necessarily similar. With $\{(X_n, \theta_n)\}_{n\geq 0}$ as defined above, if θ_n converges to a (possibly random) limit θ_*, say, the asymptotic variance for h is typically

$$\lim_{n\to\infty} nVar(\hat{\pi}_n(h)) = \mathbb{E}(\Gamma^2(h)), \tag{6}$$

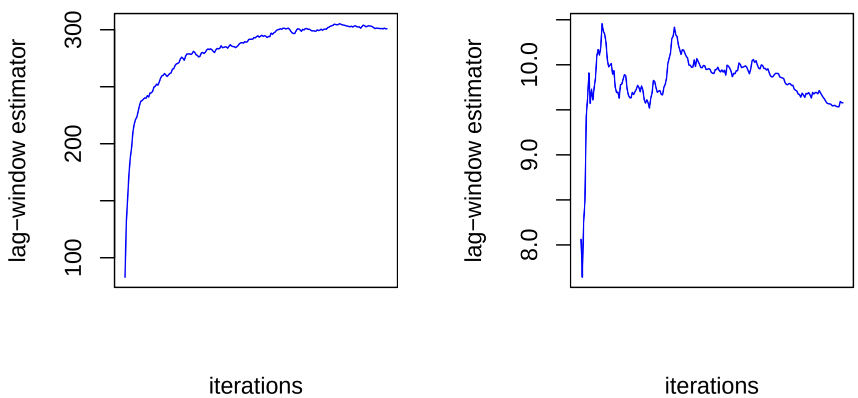

Figure 4: Sample paths of lag-window estimators converging to different limits.

where $\Gamma^2(h)$ is a non-negative, finite random variable called the *asymptotic average squared variation* of h and $\sigma^2(h) = \mathbb{E}(\Gamma^2(h))$.

Although we can still compute the same lag-window estimate $\Gamma_n^2(h)$ given in (5) from the adaptive chain$\{X_n\}_{n\geq 0}$, it turns out that if θ_* is random, $\Gamma_n^2(h)$ is inconsistent, in general, in estimating the right-hand-side of (6) (Atchadé (2011)). In fact, $\Gamma_n^2(h)$ converges to the random limit $\Gamma^2(h)$, instead of $\sigma^2(h)$. We show this for Example 1 in Section 2.1 in Figure 4.

However, Atchadé (2012) establishes that the lag-window estimators $\Gamma_n^2(h)$ can be used to derive asymptotically valid confidence interval for $\pi(h)$ in AMCMC simulation. By Proposition 3.1 of Atchadé (2011),

$$\sqrt{n}(\hat{\pi}_n(h) - \pi(h)) \xrightarrow{w} \sqrt{\Gamma^2(h)}Z, \tag{7}$$

where $Z \sim \mathcal{N}(0,1)$ is a standard Gaussian random variable that is independent of $\Gamma^2(h)$. Thus, $\sqrt{n}(\hat{\pi}_n(h) - \pi(h))$ converges to a mixture of Gaussian distribution of the form $\sqrt{\Gamma^2(h)}Z$. The confidence interval is obtained by deriving the limiting distribution of the random variable

$$T_n = \frac{\sqrt{n}(\hat{\pi}_n(h) - \pi(h))}{\sqrt{\Gamma_n^2(h)}} = \frac{\frac{1}{\sqrt{n}}\sum_{j=1}^{n} \bar{h}(X_j)}{\sqrt{\Gamma_n^2(h)}}, \tag{8}$$

where $\bar{h} = h - \pi(h)$.

On the other hand, if θ_n converges to a deterministic limit θ, then $\Gamma^2(h) \equiv \sigma^2(h)$ and $\Gamma_n^2(h)$ converges to the asymptotic variance $\sigma^2(h)$. By (7),

$$\sqrt{n}(\hat{\pi}_n(h) - \pi(h)) \xrightarrow{w} \mathscr{N}(0, \sigma^2(h)). \tag{9}$$

Again, in some cases, one can prove that $\theta \to \theta_*$, where θ_* is a discrete random variable with support $\{\tau_1, ..., \tau_N\} \subset \Theta$. The asymptotic distribution of $\sqrt{n}(\hat{\pi}_n(h) - \pi(h))$, in this case, is a mixture

$$\sum_{k\geq 1} p_k \mathscr{N}(0, \sigma_k),$$

where $p_k := \Pr(\theta_* = \tau_k)$ and $\sigma_k^2(h) = \pi(h^2) + 2\sum_{j\geq 1} \pi(hP_{\tau_k}^j h)$. A valid confidence interval for $\pi(h)$, thus, will require the knowledge of the mixing distribution p_k and the asymptotic variances $\sigma_k^2(h)$, which is more than one can obtain from $\Gamma_n^2(h)$.

However, Theorem 2.1 of Atchadé (2012) shows that when $c_n = o(n)$, T_n has a standard Gaussian limit and when $c_n = n$, Theorem 2.2 shows that T_n converges in distribution to a standard Gaussian random variable scaled by an infinite sum of chi-squared. The case $c_n = n$ corresponds to the so-called *fixed-b asymptotics* well-known in Econometrics (Kiefer and Vogelsang (2005)). These two results, presented in Section 4, allow us to derive asymptotically valid confidence intervals for $\pi(h)$ in MCMC and AMCMC simulation, which we verify in Section 5.

4 Main Results

Here, we present the main results of Atchadé (2012) (Section 2).

4.1 Setup and notations

Let $h : \mathscr{X} \to \mathscr{R}$ be a fixed measurable function. For each $\theta \in \Theta$, we assume well-defined the functions g_θ and $P_\theta g_\theta$, where

$$g_\theta(x) \stackrel{def}{=} \sum_{j\geq 0} P_\theta^j \bar{h}(x), \quad and, \quad P_\theta g_\theta \stackrel{def}{=} \int P_\theta(x,dz) g_\theta(z), \;\; x \in \mathscr{X}.$$

For each $\theta \in \Theta$, the function g_θ satisfies the Poisson's equation

$$g_\theta(x) - P_\theta g_\theta(x) = \bar{h}(x).$$

For integer $n \geq 1$, set $D_n \stackrel{def}{=} g_{\theta_{n-1}}(X_n) - P_{\theta_{n-1}} g_{\theta_{n-1}}(X_{n-1})$. For $p > 1$ and integers $n \geq k \geq 1$, let

$$a_n \stackrel{def}{=} \mathbb{E}^{\frac{1}{2p}}(|P_{\theta_n} g_{\theta_n}(X_n)|^{2p}), \qquad b_n \stackrel{def}{=} \mathbb{E}^{\frac{1}{2p}}(|P_{\theta_n} g_{\theta_n}(X_n) - P_{\theta_n} g_{\theta_{n-1}}(X_{n-1})|^{2p}),$$

$$\kappa_n \stackrel{def}{=} \mathbb{E}^{\frac{1}{2p}}(|D_n|^{2p}), \qquad \delta_{n,k}^{(1)} \stackrel{def}{=} a_{k-1} + \sum_{j=1\vee(k-c_n+1)}^{k} b_j + \frac{1}{c_n} \sum_{j=1\vee(k-c_n+1)}^{k} a_{j-1},$$

$$\text{and} \quad \delta_{n,k}^{(2)} \stackrel{def}{=} \sqrt{\sum_{j=1\vee(k-c_n+1)}^{k} \kappa_j^2}.$$

Further,

$$r_n = \Big(\frac{1}{n^{p\wedge 2}} \sum_{k=1}^{n} \kappa_k (1 + \delta_{n,k}^{(1)})^{p\wedge 2}\Big)^{\frac{1}{p\wedge 2}} + \frac{1}{nc_n} \sum_{k=1}^{n} a_k(a_k + \delta_{n,k}^{(1)} + \delta_{n,k}^{(2)}) + \frac{1}{n} \sum_{k=1}^{n} b_k(a_k + \delta_{n,k}^{(1)} + \delta_{n,k}^{(2)})$$

$$+ \frac{1}{n} a_n(a_{n-1} + \delta_{n,n}^{(1)} + \delta_{n,n}^{(2)}) + \frac{1}{n} a_0^2.$$

We define the kernel $\rho_* : [0,1] \times [0,1] \to \mathscr{R}$ by

$$\rho_*(s,t) = w(t-s) - g(t) - g(s) + \int_0^1 g(u) du,$$

where $g(t) = \int_0^1 w(t-u) du$. Clearly, ρ_* is symmetric: $\rho_*(s,t) = \rho_*(t,s)$. The kernel ρ_* induces a compact operator $\phi \mapsto (s \mapsto \int_0^1 \rho_*(s,t)\phi(t)dt)$ on $L^2[0,1]$. We abuse notation and denote it by ρ_* as well. The kernel ρ_* is said to be *positive defi-*

nite if: for all $n \geq 1$, all $a_1, \ldots, a_n \in \mathbb{R}$, and $t_1, \ldots, t_n \in [0,1]$, $\sum_{i=1}^n \sum_{j=1}^n a_i a_j \rho_*(ti, tj) \geq 0$. The positive definiteness assumption of the kernel ρ_* implies that the operator ρ_* has nonnegative eigenvalues. In which case, we will denote $\{\alpha_i, i \in I\}$ the non-empty (countable) set of positive eigenvalues of ρ_* (each repeated according to its multiplicity).

4.2 Assumptions

Assumption 1 (*A1*, Atchadé (2012))**.** *For each $\theta \in \Theta, g_\theta$ and $P_\theta g_\theta$ are well defined, and there exists $p > 1$ such that, as $n \to \infty$,*

$$a_n + \frac{1}{c_n} \sum_{k=1}^n a_k + \sum_{k=1}^n b_k + \sqrt{\sum_{k=1}^n \kappa_k^2} = O(\sqrt{n})$$

Assumption 2 (*A2*, Atchadé (2012))**.** *There exists a random variable σ_*^2, positive almost surely such that*

$$\sum_{k=1}^n \kappa_k^{2p} = o(n^p), \quad and \quad n^{-1} \sum_{k=1}^n D_k^2 \xrightarrow{a.s.} \sigma_*^2,$$

as $n \to \infty$, where p is the same as in A1.

Assumption 3 (*A3*, Atchadé (2012))**.** *The function $w : \mathbb{R} \to [0,1]$ has support $[-1,1]$, is even and satisfies:*

$$w(0) = 1, w(1) = 0$$

.

4.3 Theorems

Theorem 1 (*Theorem 2.1*, Atchadé (2012))**.** *Assume A1-A3 and $lim_n n^{-1} c_n = 0$. If $lim_n r_n = 0$, and $n^{-p\wedge 2} \sum_{k=1}^n \{\kappa_k \delta_{n,k}^{(2)}\}^{p\wedge 2} = 0$, then as $n \to \infty$, $\Gamma_n^2(h)$ converges in probability to $\sigma *^2$ and $T_n \xrightarrow{w} \mathcal{N}(0,1)$.*

Theorem 2 (*Theorem 2.1*, Atchadé (2012))**.** *Assume A1-A3 and suppose that ρ_* is positive definite. If $c_n = n$ and $lim_n r_n = 0$, then*

$$T_n \xrightarrow{w} \frac{Z_0}{\sqrt{\sum_{i\in I} \alpha_i Z_i^2}},$$

where $\{Z_0, Z_i, i \in I\}$ are i.i.d. $\mathcal{N}$(0,1) and $\{\alpha_i, i \in I\}$ is the set of positive eigenvalues of ρ_*.

As mentioned in Section 3, Theorem 1 states that under certain regularity conditions and $c_n = o(n)$, T_n in (8) converges in distribution to the standard normal. On the other hand, Theorem 2 states that under the same regularity conditions and $c_n = n$, T_n in (8) converges in distribution to a standard normal variable scaled by a sum of chi-squared random variables.

5 Examples[1]

5.1 Univariate Standard Normal

We consider the target to be the standard Normal distribution with pdf

$$\pi(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$$

We propose from a normal distribution with mean at the current step of the Markov chain and variance s^2, where s^2 is the parameter value so chosen that the acceptance probability is about 44% (which is the optimal acceptance probability for one-dimensional target distributions, Roberts and Rosenthal (2001)). Two types of Random-walk Metropolis algorithms are run - one adaptive and another non-adaptive, where s is replaced by s_n for the n^{th} iteration. They are described in Algorithm 4 and Algorithm 5. We verify Theorem 1 by checking the percentage of times the true mean 0 lies in the confidence interval $(\bar{X} - z_\alpha\sqrt{\Gamma_n^2(h))/n},\ \bar{X} + z_\alpha\sqrt{\Gamma_n^2(h))/n})$ via simulated iterations. Note that h here is the identity function

[1] To calculate $\Gamma_n^2(h)$, we have used the **mcmcse** (Flegal et al. (2021)) package in R. Further, we have used $c_n = \sqrt{n}$ and the Bartlett kernel $w(u) = (1 - |u|)\mathbf{1}_{(-1,1)(u)}$.

(i.e $h(x) = x$), $\pi(h) = 0$ and $\hat{\pi}(h) = \bar{X}$. We perform 500 such iterations for sample sizes 10^3, 10^4 and 10^5. The results are summarized in Table 1.

Algorithm 4 Simple Random-walk Metropolis-Hastings algorithm with Gaussian proposals

Let $X_n = x$. To obtain X_{n+1}:

1. $Y \sim \mathcal{N}(x, s^2)$ and independently $U \sim \mathcal{U}(0,1)$.
2. If $U < \alpha(x,y) = min\{1, \frac{\pi(y)}{\pi(x)}\}$,
   set $X_{n+1} = y$.
3. Else
   set $X_{n+1} = x$.

Algorithm 5 Adaptive Random-walk Metropolis-Hastings algorithm with Gaussian proposals

Let $X_n = x$. To obtain X_{n+1}:

1. $Y \sim \mathcal{N}(x, s_n^2)$ and independently $U \sim \mathcal{U}(0,1)$.
2. If $U < \alpha(x,y) = min\{1, \frac{\pi(y)}{\pi(x)}\}$,
   set $X_{n+1} = y$.
3. Else
   set $X_{n+1} = x$.
4. $\log(s_{n+1}) = \log(s_n) + \frac{1}{n}(\alpha(x,y) - 0.44)$.

	Sample Sizes		
Markov Chain	10^3	10^4	10^5
Adaptive	92.2%	94%	94%
Non-adaptive	91.8%	95%	95%

Table 1: Simulated Results for Univariate Gaussian Example

Clearly, Theorem 1 holds true in this example, as we can see for large sample sizes, the probability that 0 is contained in the confidence interval is close to 0.95. We observe that in this univariate case, the non-adaptive Markov chain is performing better than its adaptive counterpart.

5.2 Multivariate Logistic Regression

Let

$$y_i \sim \mathbb{B}(p(x_i'\beta))$$

independently for all $i = 1, ..N$, where

$$p(x_i'\beta) = \frac{1}{1+e^{-x_i'\beta}}.$$

We further assume a Gaussian prior on β i.e.

$$\beta \sim \mathcal{N}(0, s^2 I_p)$$

where $s = 100$. Therefore, the posterior distribution of β is given by -

$$\pi(\beta|X) \propto \Pi_{i=1}^{N} p(x_i'\beta)^{y_i}(1-p(x_i'\beta))^{1-y_i}\ e^{-\frac{\Sigma\beta_i^2}{2s^2}}$$

For the purpose of simulation, we consider the *logit* [2] dataset, where $N = 100$ and $d = 4$. We again run two types of Markov Chains - a non-adaptive Gaussian Random Walk with proposal variance h^2 chosen so that the acceptance probability is about 23.4% (Roberts and Rosenthal (2001)) and an adaptive Gaussian Random Walk using the algorithm of Roberts and Rosenthal (2009). They are described in Algorithm 6 and Algorithm 7. In Algorithm 6, we take $s = 0.5$ and in Algorithm 7, we have taken $\beta = 0.35$. We consider the above posterior distribution as our target distribution and take $h(x) = x$. Further, we run the adaptive Markov chain for 10^6 iterations and take the sample posterior mean of β as the true posterior mean. We verify Theorem 1 by checking the percentage of times the true posterior mean lies in the confidence interval of β via simulated iterations. We perform 500 such iterations for sample sizes 10^3, 10^4 and 10^5. The results are summarized in Table 2.

[2] We have used the *logit* dataset from the **mcmc** (Geyer and Johnson (2020)) package in R.

Algorithm 6 Simple Random-walk Metropolis-Hastings algorithm with multivariate Gaussian proposals

Let $X_n = x$. To obtain X_{n+1}:

1. $Y \sim \mathcal{N}_d(x, s^2 I_d)$ and independently $U \sim \mathscr{U}(0,1)$.
2. If $U < \alpha(x,y) = min\{1, \frac{\pi(y)}{\pi(x)}\}$,
   set $X_{n+1} = y$.
3. Else
   set $X_{n+1} = x$.

Algorithm 7 Adaptive Random-walk Metropolis-Hastings algorithm with multivariate Gaussian proposals

Let $X_n = x$. To obtain X_{n+1}:

1. $Y \sim Q_n(x,.)$ and independently $U \sim \mathscr{U}(0,1)$, where

   $Q_n(x,.) = \mathcal{N}_d(x, (0.1)^2 I_d/d)$ for $n \leq 2d$ and,
   $Q_n(x,.) = (1-\beta)\mathcal{N}_d(x, (2.38)^2 \Sigma_n/d) + \beta \mathcal{N}_d(x, (0.1)^2 I_d/d)$ for $n > 2d$.

2. If $U < \alpha(x,y) = min\{1, \frac{\pi(y)}{\pi(x)}\}$,
   set $X_{n+1} = y$.
3. Else
   set $X_{n+1} = x$.

Sample Sizes			
Markov Chain	10^3	10^4	10^5
Non-Adaptive	83.4%	92.2%	94.6%
Adaptive	73.4%	94.4%	95.2%

Table 2: Simulated Results for Multivariate Logistic Example

Clearly, Theorem 1 holds true in this example, as we can see for large sample sizes, the probability that the sample posterior mean of β (taken as true mean) is contained in the confidence interval of β is close to 0.95. Furthermore, the adaptive Markov chain seems to be performing much better in comparison with its non-adaptive counterpart, which is expected in a multi-dimensional case.

Another interesting application would be the *heart*[3] dataset which is given as an example in Atchadé (2012) having $N = 217$ and $d = 14$. Due to computational shortcomings, the dataset could not be analyzed and it can be explored in the future to further verify Theorem 1.

6 Conclusion

We have discussed the problems that come with adaptation in MCMC, particularly in estimating asymptotic variance of a Monte carlo estimator. We verified conditions under which a Central Limit Theorem holds in AMCMC using two examples - one univariate and one multivariate. We note that Theorem 1 and Theorem 2 both concern univariate random variables. An interesting future work may involve developing such confidence intervals in the multivariate setup.

7 Supplementary Material

The interested reader is directed to https://github.com/ArkaB-DS/MTH598A which contains all the figures present here in the directory `images` and the corresponding codes to generate them in the `codes` directory.

[3] https://archive.ics.uci.edu/ml/datasets/Heart+Disease

8 Acknowledgements

I would like to heartily thank my supervisor Dr. Dootika Vats for her valuable feedback and constant guidance on this project.

References

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine learning*, 50(1):5–43.

Andrieu, C. and Moulines, É. (2006). On the ergodicity properties of some adaptive MCMC algorithms. *The Annals of Applied Probability*, 16(3):1462–1505.

Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4):343–373.

Atchadé, Y. F. (2011). Kernel estimators of asymptotic variance for adaptive Markov chain Monte Carlo. *The Annals of Statistics*, 39(2):990–1011.

Atchadé, Y. F. (2012). Adaptive Markov chain Monte Carlo confidence intervals. *arXiv preprint arXiv:1209.0703*.

Atchadé, Y. F. and Rosenthal, J. S. (2005). On adaptive markov chain monte carlo algorithms. *Bernoulli*, 11(5):815–828.

Besag, J., Green, P., Higdon, D., and Mengersen, K. (1995). Bayesian computation and stochastic systems. *Statistical science*, pages 3–41.

Besag, J. and Green, P. J. (1993). Spatial statistics and Bayesian computation. *Journal of the Royal Statistical Society: Series B (Methodological)*, 55(1):25–37.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of markov chain monte carlo*. CRC press.

Chib, S. and Greenberg, E. (1995). Understanding the metropolis-hastings algorithm. *The american statistician*, 49(4):327–335.

Chimisov, C., Latuszynski, K., and Roberts, G. (2018). Adapting the Gibbs sampler. *arXiv preprint arXiv:1801.09299*.

Craiu, R. V., Rosenthal, J., and Yang, C. (2009). Learn from thy neighbor: Parallel-chain and regional adaptive MCMC. *Journal of the American Statistical Association*, 104(488):1454–1466.

Damerdji, H. (1995). Mean-square consistency of the variance estimator in steady-state simulation output analysis. *Operations Research*, 43(2):282–291.

Flegal, J. M., Hughes, J., Vats, D., Dai, N., Gupta, K., and Maji, U. (2021). *mcmcse: Monte Carlo Standard Errors for MCMC*. Riverside, CA, and Kanpur, India. R package version 1.5-0.

Flegal, J. M. and Jones, G. L. (2010). Batch means and spectral variance estimators in Markov chain Monte Carlo. *The Annals of Statistics*, 38(2):1034–1070.

Gelman, A., Gilks, W. R., and Roberts, G. O. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The annals of applied probability*, 7(1):110–120.

Geyer, C. J. and Johnson, L. T. (2020). *mcmc: Markov Chain Monte Carlo*. R package version 0.9-7.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. (1995). *Markov chain Monte Carlo in practice*. CRC press.

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, pages 223–242.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications.

Hensman, J., Matthews, A. G. d. G., Filippone, M., and Ghahramani, Z. (2015). Mcmc for variationally sparse Gaussian processes. *arXiv preprint arXiv:1506.04000*.

Kiefer, N. M. and Vogelsang, T. J. (2005). A new asymptotic theory for heteroskedasticity-autocorrelation robust tests. *Econometric Theory*, 21(6):1130–1164.

Kim, W., Navarro, D. J., Pitt, M. A., Myung, I. J., Thrun, S., and Saul, L. (2003). An MCMC-Based Method of Comparing Connectionist Models in Cognitive Science. In *NIPS*, pages 937–944. Citeseer.

Liu, J. S. and Liu, J. S. (2001). *Monte Carlo strategies in scientific computing*, volume 10. Springer.

Ma, Y.-A., Chen, T., and Fox, E. B. (2015). A complete recipe for stochastic gradient mcmc. *arXiv preprint arXiv:1506.04696*.

Mahendran, N., Wang, Z., Hamze, F., and De Freitas, N. (2012). Adaptive MCMC with Bayesian optimization. In *Artificial Intelligence and Statistics*, pages 751–760. PMLR.

Mallik, A. and Jones, G. L. (2017). Directional metropolis-hastings. *arXiv preprint arXiv:1710.09759*.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

Pasarica, C. and Gelman, A. (2010). Adaptively scaling the Metropolis algorithm using expected squared jumped distance. *Statistica Sinica*, pages 343–364.

Robert, C. P. and Casella, G. (1999). The metropolis—hastings algorithm. In *Monte Carlo Statistical Methods*, pages 231–283. Springer.

Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical science*, 16(4):351–367.

Roberts, G. O. and Rosenthal, J. S. (2007). Coupling and ergodicity of adaptive Markov chain Monte carlo algorithms. *Journal of applied probability*, 44(2):458–475.

Roberts, G. O. and Rosenthal, J. S. (2009). Examples of adaptive MCMC. *Journal of computational and graphical statistics*, 18(2):349–367.

Sharma, S. (2017). Markov chain Monte Carlo methods for Bayesian data analysis in astronomy. *Annual Review of Astronomy and Astrophysics*, 55:213–259.

Sorensen, D., Gianola, D., and Gianola, D. (2002). Likelihood, Bayesian and MCMC methods in quantitative genetics.

Thrane, E. and Talbot, C. (2019). An introduction to Bayesian inference in gravitational-wave astronomy: parameter estimation, model selection, and hierarchical models. *Publications of the Astronomical Society of Australia*, 36.

Vajargah, K. F., Benis, S. G., and Golshan, H. M. (2021). Detection of the quality of vital signals by the Monte Carlo Markov Chain (mcmc) method and noise deleting. *Health Information Science and Systems*, 9(1):1–10.