Markov chain Monte Carlo

Karl Oskar Ekvall* Galin L. Jones†
University of Minnesota

March 15, 2019

Abstract

Practically relevant statistical models often give rise to probability distributions that are analytically intractable. Fortunately, we now have a collection of algorithms, known as Markov chain Monte Carlo (MCMC), that has brought many of these models within our computational reach. MCMC is a simulation technique that allows one to make (approximate) draws from complex, high dimensional probability distributions. A staggering amount of research has been done on both the theoretical and applied aspects of MCMC. This article does not intend to be a complete overview of MCMC but only hopes to get the reader started in the right direction. To this end, this article begins with a general description of the types of problems that necessitate the use of MCMC. It then introduces the fundamental algorithms and addresses some general implementation issues.

1 Introduction

Many statistical methods at some stage require sampling from a probability distribution. In ideal cases, one can generate independent and identically distributed (i.i.d.) observations from the desired distribution. In many practically relevant models, however, this is either not possible or prohibitively time consuming. Fortunately, in a wide range of settings, Markov chain Monte Carlo (MCMC) can be used in place of i.i.d. sampling [cf. 1, 11]. Because of this, after the seminal paper by Gelfand and Smith [3], MCMC has become integral to Bayesian analysis where such complicated distributions often arise, but is also important in some frequentist settings [5].

*ekvall@umn.edu
†galin@umn.edu

We illustrate with one of the most common purposes of generating (pseudo-)random numbers. If h is some real-valued function and X is a random variable, then we want to calculate an expectation in the form

$$\mu_h := \mathbb{E}(h(X)) < \infty.$$

For different choices of h, many problems both in statistics and other disciplines can be written in this way. If μ_h is complicated to calculate analytically or by numerical integration, an alternative is to generate $X_1, \ldots, X_m$, independent with the same distribution as X, and approximate μ_h by

$$\hat{\mu}_h := \frac{1}{m} \sum_{i=1}^{m} h(X_i).$$

Here, X can be a random vector or something more general, but the main points of our discussion are equally well grasped thinking of X as a random number. Several useful properties of $\hat{\mu}_h$ are immediate from classical statistical results: (i) $\hat{\mu}_h$ is unbiased, (ii) the **law of large numbers** (LLN) says that $\hat{\mu}_h$ is consistent as m tends to infinity, and (iii) if $\mathbb{E}(h(X)^2) < \infty$, the central limit theorem (CLT) says that $\hat{\mu}_h$ is approximately normally distributed for large m. The LLN is important because it says, loosely speaking, that we can improve the estimate by simulating longer, and the the CLT is important because it lets us quantify the uncertainty in the estimate. In particular, $\mathrm{var}(\hat{\mu}_h)$ can be estimated by s_h^2/m, where $s_h^2 = m^{-1} \sum_{i=1}^{m} (h(X_i) - \hat{\mu}_h)^2$, and approximate **confidence intervals** for μ_h can be created by appealing to the CLT. The only difference between this and classical statistics is that the variables are generated in a computer. Methods that use the generation of random numbers are called **Monte Carlo** (MC) methods and $\hat{\mu}_n$ is called a MC estimator of μ_h. MC methods with i.i.d. variables are sometimes called ordinary MC or i.i.d. MC.

MCMC is similar to ordinary MC but with the key difference that the generated variables $X_1, \ldots, X_m$ need be neither independent, nor have the same distributions. Rather, as the name suggests, they are generated as a **Markov chain**. Since i.i.d. variables form a Markov chain, ordinary MC is a special case of MCMC. The power of MCMC, however, is that the useful properties of $\hat{\mu}_h$ discussed above (LLN and CLT) continue to hold for much more general chains. Such chains can be constructed in many cases where i.i.d. sampling is infeasible and, hence, MCMC is more widely applicable than ordinary MC. For an introduction to Markov chain theory see [10, 12].

We say that $X_1, X_2, \ldots$ is a Markov chain if the conditional distribution of X_i given $X_1, \ldots, X_{i-1}$, $i \geq 2$, depends only on X_{i-1}; this is known as the Markov property. It follows from the Markov property that a Markov chain is characterized by its initial

distribution (the distribution of X_1), and its transition kernel P defined by

$$P(x, A) = \mathsf{P}(X_i \in A \mid X_{i-1} = x),$$

for any subset A of the state space, the set in which the Markov chain takes its values. If the initial distribution and kernel is such that the distribution of X_2 is the same as that of X_1 we say that the initial distribution is invariant for the kernel. More generally, a distribution F is invariant for the transition kernel P if $X_i \sim F$ implies $X_{i+1} \sim F$, $i \geq 1$. Using this definition it can be shown that if the initial distribution is invariant for P, then in fact every X_i, $i \geq 1$ has the same distribution. Such Markov chains are called stationary, and they are indeed stationary in the usual sense for stochastic processes.

Let F_X denote the distribution of X and suppose we generate a Markov chain with initial distribution F_X and a kernel P for which F_X is invariant. Then by the preceding discussion $X_1, \ldots, X_m$ are possibly dependent but identically distributed random variables with the same distribution as X. Hence, $\hat{\mu}_h$ is an unbiased estimator of μ_h. Moreover, it can be shown that under additional conditions $\hat{\mu}_h$ is consistent and asymptotically normal with variance κ_h^2/m, where, with $\sigma_h^2 = \operatorname{var}(h(X))$,

$$\kappa_h^2 = \sigma_h^2 + 2\sum_{i=1}^{\infty} \operatorname{cov}(h(X_1), h(X_{1+i})).$$

Recall, however, that MCMC is often used precisely because sampling from F_X is infeasible. Hence, generating the stationary chain is also infeasible as it requires $X_1 \sim F_X$. Fortunately, it can be shown that if $\hat{\mu}_h$ is consistent and satisfies a CLT when the initial distribution is F_X, then the same is true for any other initial distribution. This tells us that if the simulation is long enough (m is large enough), then the starting value X_1, whether selected at random or set to some fixed number, is unimportant. In practice this argument is somewhat problematic because it is hard to know what long enough means, but let us ignore that for now—we will return to the issue of starting values later. Next we outline how to, given a target distribution, generate Markov chains for which that distribution is invariant. The focus is on two of the most common algorithms, the Metropolis–Hastings (MH) algorithm and the **Gibbs sampler**.

1.1 Metropolis–Hastings

Suppose that we want to estimate μ_h and know the target density only up to a normalizing constant, or up to scaling. That is, we know that X has a distribution F_X

with density f_X satisfying

$$f_X(x) = cp(x)$$

for some $c > 0$ and function p. Of course, the fact that densities must integrate to one tells us that $c = 1/\int p(x)\,\mathrm{d}x$, but if p is complicated to integrate, c is not known in any practical sense. Thus, even though p can be evaluated we cannot compute c or easily sample from F_X. Settings like this are exceedingly common in Bayesian statistics.

Given an unnormalized density like p, the MH algorithm (Algorithm 1) constructs a Markov chain with a transition kernel for which F_X is invariant. That is, the MH algorithm lets us sample approximately from F_X even though we only know the corresponding density f_X up to a normalizing constant. The algorithm transitions between states as follows: given that the chain is at state $X_i = x_i$, a move is proposed to a new state y drawn from some distribution with density $q(y \mid x_i)$. Then, the move is either accepted, which happens with a probability that depends on $p(y)$, $p(x_i)$, $q(y \mid x_i)$, and $q(x_i \mid y)$, or rejected. If the move is rejected, the chain stays in the same place for one iteration and then another move is proposed. Since the proposal distribution and the acceptance probability depend on the current state but no previous states the algorithm indeed generates a Markov chain.

To implement the MH algorithm one needs to select a proposal distribution (density) $q(\cdot \mid \cdot)$. Any proposal distribution having support containing that of p will lead to the chain having the right invariant distribution. However, the convergence properties of the chain are in general affected by the choice. A discussion of standard strategies for selecting the proposal distribution is provided by Robert and Casella [11].

The following example illustrates how the MH algorithm can be used in Bayesian statistics. The example is chosen to be simple enough that no MCMC is actually required, which makes the results easy to verify using numerical integration or i.i.d. sampling, but also complicated enough to convey some key ideas.

Example 1.1 (Bayesian estimation of normal mean and variance with conjugate priors)**.** Suppose we have 30 independent observations $y_1, \ldots, y_{30}$ drawn from a **normal distribution** with the unknown mean $\mu^\star = 1$ and variance $1/\tau^\star = 1$, where the stars are used to indicate unknown population values. We wish to incorporate prior information about the parameters and specify the **prior distribution** in two stages by letting $\mu \mid \tau \sim \mathcal{N}(a, \tau^{-1}b^{-1})$ and $\tau \sim \mathcal{G}(c, d)$, where $\mathcal{G}(c, d)$ denotes the gamma distribution with mean c/d. That is,

$$f(\mu \mid \tau) = (2\pi)^{-1/2}(b\tau)^{1/2}e^{-b\tau(\mu-a)^2/2} \quad \text{and} \quad f(\tau) = \frac{d^c}{\Gamma(c)}\tau^{c-1}e^{-\tau d}I(\tau > 0),$$

Algorithm 1 Metropolis–Hastings

1: *Input:* Starting value X_1 and length of chain m

2: **for** $i = 1, \ldots, m$ **do**

3: Given $X_i = x_i$, draw proposal y from a distribution with density $q(y \mid x_i)$.

4: Calculate the Hastings ratio

$$r(x_i, y) = \frac{p(y)q(x_i \mid y)}{p(x_i)q(y \mid x_i)}.$$

5: Randomly pick the next state X_{i+1} by accepting or rejecting proposal y:

$$X_{i+1} = \begin{cases} y & \text{w. prob. } \alpha(x_i, y) = \min[1, r(x_i, y)] \\ x_i & \text{w. prob. } 1 - \alpha(x_i, y) \end{cases}$$

6: **end for**

for hyperparameters $a \in \mathbb{R}$, $b > 0$, $c > 0$, and $d > 0$, where $\Gamma(\cdot)$ denotes the gamma function. In an application the hyperparameters would be chosen to reflect the prior beliefs about μ and τ. For concreteness, we here somewhat arbitrarily set them to $a = 0$, $b = c = d = 1$.

In Bayesian statistics the interest is in the **posterior density** $f(\mu, \tau \mid y)$. By standard rules for probability densities, the posterior density satisfies

$$\begin{aligned} f(\mu, \tau \mid y) &= \frac{f(y \mid \mu, \tau) f(\mu \mid \tau) f(\tau)}{f(y)} \\ &\propto f(y \mid \mu, \tau) f(\mu \mid \tau) f(\tau) \end{aligned}$$

where $\propto$ means equality holds up to scaling by a quantity not depending on μ or τ. Multiplying the prior densities by the likelihood

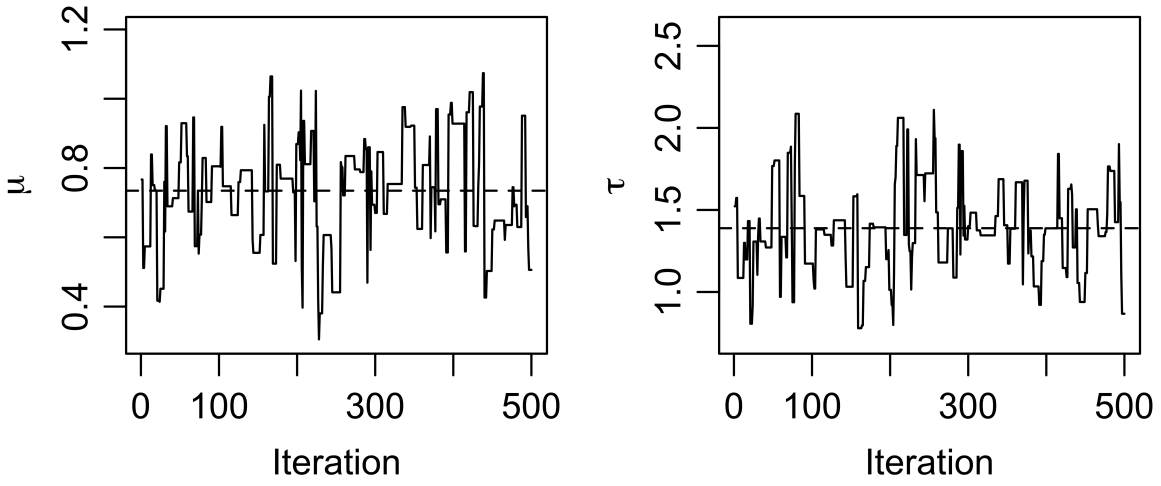
$$f(y \mid \mu, \tau) = (2\pi)^{-n/2} \tau^{n/2} \exp\left(-\frac{\tau}{2} \sum_{i=1}^{n} (y_i - \mu)^2 \right)$$

we find that the posterior satisfies

$$f(\mu, \tau \mid y) \propto \tau^{n/2} \exp\left(-\frac{\tau}{2} \sum_{i=1}^{n} (y_i - \mu)^2 - \tau\mu^2/2 - \tau \right). \tag{1}$$

Let us define $\theta = (\mu, \tau)$ and denote the expression in (1) by $p(\theta; y)$. The density from which we want to sample is $f(\theta \mid y) = p(\theta; y) / \int p(\theta; y) \, \mathrm{d}\theta$. In this example the state space of the Markov chain to be generated is the parameter set of the model for the data y—this is typical for Bayesian statistics. Accordingly, for the remainder of this

Figure 1: Metropolis–Hastings Example chain. Horizontal dashed lines indicate sample averages of the plotted observations.



example we let $\Theta = \mathbb{R} \times (0, \infty)$ denote the state space, $\theta_i = (\mu_i, \tau_i)$ the ith state of the Markov chain, and ξ the proposed value with conditional density $q(\xi \mid \theta_i)$ in the MH algorithm, so as to not confuse it with the observed data.

We are ready to define a MH algorithm for exploring the distribution with density $f(\theta \mid y)$ on Θ, which as mentioned amounts to selecting a proposal distribution. We take the proposal distribution to be multivariate normal and centered at the current state. More precisely, $\xi \mid \theta_i \sim \mathcal{N}(\theta_i, 0.25 I_2)$. The covariance matrix is selected with some experimentation to result in an acceptance rate (the proportion of accepted proposals) of roughly 0.25 (see Rosenthal [13] for a motivation of this number). The starting value is set to $\theta_0 = (\mu_0, \tau_0) = (\bar{y}, 1/s_y^2)$, where $\bar{y}$ is the sample average and s_y^2 the biased sample variance; these are the **maximum likelihood estimators** (MLEs) of the parameters μ and τ. The hope is that the MLEs are in a high-density region of the posterior.

Figure 1 shows the output from running the MH algorithm for 500 iterations. The short, flat segments where the chain stays in the same place for a few iterations correspond to rejected proposals. The sample paths can be used to get an MCMC estimate of $\int h(\theta) f(\theta) \, \mathrm{d}\theta$ for any function h for which the integral exists. If we are, for example, interested in the Bayes estimate of μ, $\mathbb{E}(\mu \mid y)$, then we can take $h(\theta) = h((\mu, \tau)) = \mu$ and estimate $\int h(\theta) f(\theta \mid y) \, \mathrm{d}\theta = \int \mu f(\mu \mid y) \, \mathrm{d}\mu$ by $\sum_{i=1}^{500} h(\theta_i)/500 = \sum_{i=1}^{500} \mu_i/500$. This sample average, which is indicated by a dashed line in 1, is 0.73. This can be compared to the MLE $\bar{y} = 0.77$, and the true mean $\mu^\star = 1$. We will return to this example below when implementing a Gibbs sampler.

1.2 Gibbs samplers

Suppose that the random variable X which distribution F_X we would like to sample from is multivariate, i.e. a random vector. We can then split X into sub-vectors, say $X = (X^{(1)}, \ldots, X^{(s)})$; each $X^{(i)}$ can be univariate or multivariate. We will call $X^{(1)}, \ldots, X^{(s)}$ the components of X. To implement a Gibbs sampler, we need to be able to sample from the conditional distribution of any one component given the rest, also known as the component's full conditional distribution. The Gibbs sampler proceeds by updating the states of the components iteratively, drawing new states from the full conditional distributions as detailed in Algorithm 2.

To implement a Gibbs sampler one has to select how to partition the vector X into components. In other words, one has to select which elements of the Markov chain to update together. As an example, if we have a three-dimensional target distribution, then $X_i = (X_{i,1}, X_{i,2}, X_{i,3})$ can be updated in four ways: (i) each element is updated separately, (ii) $X_{i,1}$ and $X_{i,2}$ are updated together (and $X_{i,3}$ is updated by itself), (iii) $X_{i,1}$ and $X_{i,3}$ are updated together, or (iv) $X_{i,2}$ and $X_{i,3}$ are updated together. Which elements are updated together can affect the convergence properties of the chain and hence it can be worthwhile to consider different configurations. There is no general rule to guide the selection, though there is some evidence that strongly correlated components should be updated together. We next illustrate how a Gibbs sampler can be implemented in practice.

Algorithm 2 Gibbs sampler

1: *Input:* Starting value $X_1 = x_1$ and length of chain m
2: **for** $i = 1, \ldots, m$ **do**
3: **for** $j = 1, \ldots, s$ **do**
4: Draw $x_{i+1}^{(j)}$ from the distribution of

$$X^{(j)} \mid \left(X^{(1)} = x_{i+1}^{(1)}, \ldots, X^{(j-1)} = x_{i+1}^{(j-1)}, X_i^{(j+1)} = x_i^{(j+1)}, \ldots, X^{(s)} = x_i^{(s)}\right)$$

5: **end for**
6: **end for**

Example 1.1 (continued)**.** Recall, the posterior distribution that we are considering has a density that is known up to scaling:

$$f(\mu, \tau \mid y) \propto \tau^{n/2} \exp\left(-\frac{\tau}{2}\sum_{i=1}^{n}(y_i - \mu)^2 - \tau\mu^2/2 - \tau\right).$$

Since the chain is bivariate, our only choice is to update μ and τ separately if we want to implement a Gibbs sampler—there is no other way to split the chain into components. To find the necessary full conditional distributions, notice that for a fixed τ the exponent of $f(\mu, \tau \mid y)$ is a quadratic function in μ, and for for a fixed μ the exponent is linear in τ. Using this, one can show that

$$\mu \mid \tau, y \sim \mathcal{N}\left((n+1)^{-1}\sum_{i=1}^{n} y_i, \tau^{-1}(n+1)^{-1}\right)$$

and

$$\tau \mid \mu, y \sim \mathcal{G}(n/2+1, \sum_{i=1}^{n}(y_i-\mu)^2/2 + \mu^2/2 + 1).$$

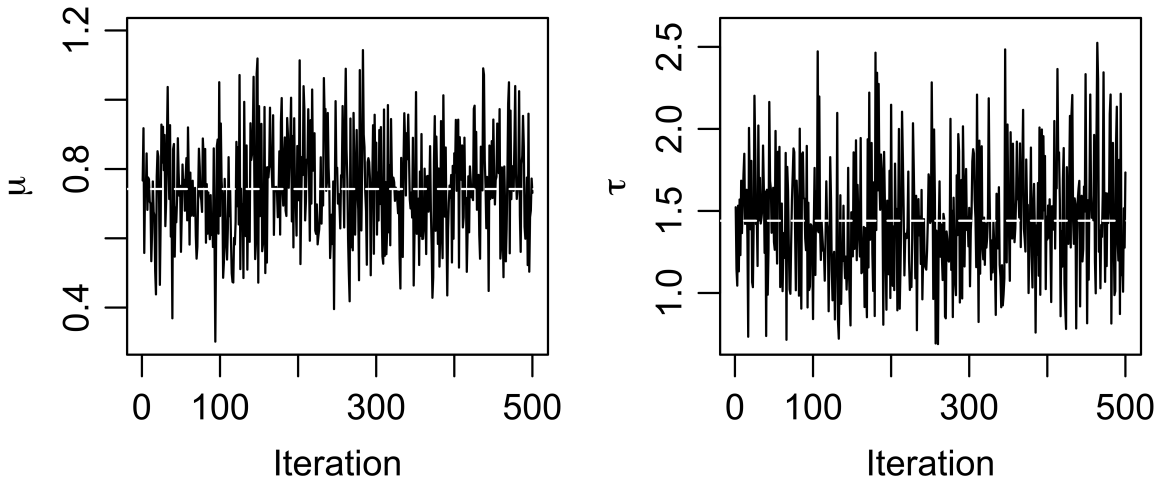
We implement a Gibbs sampler that first updates τ and then μ. There is no specific reason for this choice—updating μ first works equally well. As with the MH algorithm, the starting value is $\theta_0 = (\bar{y}, 1/s_y^2)$. Notice, however, that with the Gibbs sampler only the starting value for μ matters since τ is updated first, and the distribution from which τ_2 is drawn does not depend on τ_1. Figure 2 shows the output from running the Gibbs sampler for 500 iterations. The sample paths there depicted can be used in the same way as those of the MH algorithm. Of course, estimates based on the output in Figure 2 will be different from those based on the output in Figure 1. In this particular case, one might prefer estimates based on the Gibbs sampler since, for both μ and τ, the sample path depicted in Figure 2 looks more like that of uncorrelated variables than that in Figure 1. This indicates the variance of estimators based on the Gibbs chain may be lower than that of those based on the MH chain in this example. The Gibbs chain is also exploring a larger part of the state space than the MH chain in the first 500 iterations.

1.3 Variance estimation

We have said that MCMC often leads to a consistent and asymptotically normal $\hat{\mu}_h$, and we have shown how to construct Markov chains that have the desired invariant distribution. However, not all chains with the right invariant distribution give a consistent and asymptotically normal $\hat{\mu}_h$. Conditions that ensure these properties are fairly technical and in general have to be verified on a case-by-case basis [7]. For the remainder of this introduction we assume that $\hat{\mu}_h$ is both consistent and asymptotically normal. That is, we assume that approximately for large m,

$$\hat{\mu}_h \sim \mathcal{N}(\mu_h, \kappa_h^2/m),$$

Figure 2: Gibbs chain. Horizontal dashed lines indicate sample averages of the plotted observations.



where as before

$$\kappa_h^2 = \sigma_h^2 + 2\sum_{i=1}^{\infty} \mathrm{cov}(h(X_1), h(X_{1+i})). \tag{2}$$

Under i.i.d. sampling, the infinite sum of autocovariances in (2) vanishes and $\kappa_h^2 = \sigma_h^2$ can be estimated by the sample variance s_h^2. In contrast, for more general MCMC algorithms the infinite sum typically does not vanish and κ_h^2 is more challenging to estimate than σ_h^2. It is also important to notice that κ_h^2 and σ_h^2 quantify different things: σ_h^2 is a characteristic of the invariant distribution only but κ_h^2 depends on the joint distribution of all the variables in the chain. In particular, two stationary Markov chains with the same invariant distribution but different autocovariances will lead to the same σ_h^2 but different κ_h^2. Since κ_h^2 directly determines the uncertainty in the estimate $\hat{\mu}_h$, it is desirable to, all else equal, pick an algorithm that leads to small (or negative) autocovariances. When we return to our example below, we will see that two perfectly reasonable MCMC algorithms can, for the same problem, generate chains that have the same invariant distribution but substantially different autocovariances.

In most realistic settings κ_h^2 is unknown and must be estimated using the Markov chain if we hope to say something about the uncertainty in $\hat{\mu}_h$. There are several methods for estimating κ_h^2 that use the same chain used to estimate μ_h [2, 4, 15]. Here, we will give a short introduction to the method of batch means which gives an estimator that is easy and fast to compute. Suppose that b is a divisor of m and define,

for $k = 1, \ldots, m_b = m/b$, the batch mean

$$\hat{\mu}_{h,k} = b^{-1} \sum_{i=1}^{b} h(X_{b(k-1)+i}).$$

Thus, $\hat{\mu}_{h,1}$ is the MCMC estimator of μ_h based on only the first b variables in the chain, $\hat{\mu}_{h,2}$ is that based on only the next b variables, and so on. The batch means estimator of κ_h^2 is

$$\hat{\kappa}_h^2 = \frac{\sqrt{b}}{m_b} \sum_{i=1}^{m_b} (\hat{\mu}_{h,i} - \hat{\mu}_h)^2.$$

When deciding on the number of batches there is a trade-off between estimating the mean in each batch precisely, which requires a large b, and estimating the variability among batches precisely, which requires a large m_b. Geyer [6] suggests that 20 – 30 batches is enough for most applications. Another common choice is to pick the number of batches to be approximately $m_b = \sqrt{m}$. After computing $\hat{\kappa}_h^2$, confidence intervals for μ_h, and corresponding tests, can be computed using a t-distribution; for $\alpha \in (0, 1)$, a $100(1 - \alpha)\%$ confidence interval for μ_h is given by

$$\hat{\mu}_h \pm t_{b-1,1-\alpha/2} \sqrt{\hat{\kappa}_h^2 / m},$$

where $t_{b-1,1-\alpha/2}$ is the $(1 - \alpha/2)$th quantile of the t-distribution with $b - 1$ degrees of freedom.

In practice, we are rarely interested in estimating just one characteristic of the target distribution, so h is usually multivariate, or vector-valued. For such settings a multivariate analysis that takes into account the dependence between the components of the multivariate estimator $m^{-1} \sum_{i=1}^{m} h(X_i)$ is appropriate. Methods for multivariate output analysis are available [14].

Example 1.1 (continued)**.** We have previously in this example generated two Markov chains that have the desired invariant distribution and either could be used to estimate μ_h for some h of interest. Let us continue to focus on the choice $h(\theta) = h((\mu, \tau)) = \mu$, indicating the quantity of interest is the posterior mean $\mathbb{E}(\mu \mid y)$. For both the MH algorithm and the Gibbs sampler, we estimate the posterior mean by the sample mean of the generated μ-chain, i.e. the first component of the bivariate chain. The 500 iterations of the chains displayed in Figures 1 and 2 are not enough to get reliable estimates μ_h or κ_h^2. After running the chains for 50,000 iterations, the estimate of μ_h based on the MH chain is 0.743 and the estimate based on the Gibbs chain is 0.742. Moreover, if we calculate the sample variance of each chain, s_h^2, they are both approximately 0.024. However, the batch means estimate of κ_h^2 is 0.160 for the MH chain and 0.021 for the Gibbs chain. This illustrates what was mentioned before,

namely that μ_h and σ_h^2 are the same for both chains, but κ_h^2 is different since the autocovariances in the chains are different. The estimates of κ_h^2 indicate that in this particular example, the infinite sum in (2) is larger for the MH chain than the Gibbs sampler. Indeed, we noted earlier that the sample paths in Figure 2 look more like those of uncorrelated variables than those in Figure 1.

1.4 Starting and stopping

Whether one is using a MH algorithm, a Gibbs sampler, or something else, deciding where to start the algorithm and when to stop sampling is usually up to the user. It is generally a good idea to pick a starting point in a high-density region of the target distribution. If we could start the chain with the invariant distribution we would likely get a starting value in such a region, and hence, intuition suggests, those points are good to start at. Of course, in many problems we do not have a good guess for a high-density region and then this method is not applicable. Another common practice is to discard a number of the early observations in the chain, say $X_1, \ldots, X_B$ for some $B < m$, and instead estimate μ_h using only the last $m - B$ observations. This practice is known as burn-in and the idea is that the burn-in should bring the chain closer to its stationary distribution, approximating a situation where the initial value is drawn from the invariant distribution. This intuitive idea is not so easily motivated formally, however, and many authorities consider burn-in questionable [6].

Having selected where to start, one also needs to decide when to stop. In general, a longer chain is better and it is hence uncontroversial to say that m should be as large as possible. Even so, it is in many settings desirable to have an idea about when m is large enough, in some quantifiable sense. There are ways to measure the distance between the distribution of the chain and the target distribution that can in some cases be used to determine an appropriate m before starting the simulation. This subject is rather technical and we refer the reader to Jones and Hobert [9] for an introduction. One may also construct stopping rules that use, for example, an estimate of $\mathrm{var}(\hat{\mu}_h)$ to decide when to terminate a simulation in real time [8, 14]. For example, one may calculate the width of the confidence interval for μ_h, i.e. $t_{d-1,1-\alpha/2}\sqrt{\hat{\kappa}_h^2/m}$, and terminate when it falls below some pre-specified threshold. We illustrate this idea in the context of the running example.

Example 1.1 (continued)**.** We have considered $\sum_{i=1}^{m} \mu_i/m$ as an estimator for the posterior mean $\mathbb{E}(\mu \mid y)$. In Figures 1 and 2 we used $m = 500$. When discussing variance estimation we instead used $m = 50000$. To see how to employ a stopping rule to decide on an appropriate m based on the batch means estimate of $\mathrm{var}(\hat{\mu}_h)$, suppose

we are content with a width of 0.05 for a 95 % confidence interval for $\mathbb{E}(\mu \mid y)$. To avoid stopping the simulation too early due to poor estimates of $\text{var}(\hat{\mu}_h)$ for small m, let us implement a stopping rule as follows: for every $m = 10000, 11000, 12000, \ldots$ calculate a 95% confidence interval for $\mathbb{E}(\mu \mid y)$; if its width is less than 0.05, simulate for another 1000 iterations and try again, and otherwise stop the simulation. Implementing this in our example, we find that the MH algorithm stops after $m = 142000$ iterations while the Gibbs sampler stops after $m = 134000$ iterations.

Related Articles

See also **Bayesian Inference**; **Bayesian Analysis and Markov Chain Monte Carlo Simulation**; **Monte Carlo Methods, Univariate and Multivariate**; **Markov Chain Monte Carlo, Introduction**; **Markov Chain Monte Carlo, Convergence and Mixing in**; **Markov Chain Monte Carlo Methods**; **Markov Chain Monte Carlo Algorithms**

References

[1] S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, editors. *Handbook of Markov chain Monte Carlo*. Chapman & Hall / CRC, 2011.

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

[3] A. E. Gelfand and A. F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409, 1990.

[4] C. J. Geyer. Practical Markov chain Monte Carlo. *Statistical Science*, 7(4):473–483, nov 1992.

[5] C. J. Geyer. On the convergence of Monte Carlo maximum likelihood calculations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(1):261–274, 1994.

[6] C. J. Geyer. Introduction to Markov chain Monte Carlo. In S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, editors, *Handbook of Markov chain Monte Carlo*. Chapman & Hall / CRC, 2011.

[7] G. L. Jones. On the Markov chain central limit theorem. *Probability Surveys*, 1(0):299–320, 2004.

[8] G. L. Jones, M. Haran, B. S. Caffo, and R. Neath. Fixed-width output analysis for Markov chain Monte Carlo. *Journal of the American Statistical Association*, 101(476):1537–1547, 2006.

[9] G. L. Jones and J. P. Hobert. Honest exploration of intractable probability distributions via Markov chain Monte Carlo. *Statistical Science*, 16(4):312–334, nov 2001.

[10] S. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press, 2011.

[11] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer New York, 2013.

[12] G. O. Roberts and J. S. Rosenthal. General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1(0):20–71, 2004.

[13] J. S. Rosenthal. Optimal proposal distributions and adaptive MCMC. In S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, editors, *Handbook of Markov chain Monte Carlo*. Chapman & Hall / CRC, 2011.

[14] D. Vats, J. M. Flegal, and G. L. Jones. Multivariate output analysis for markov chain monte carlo. *Biometrika*, 2018+.

[15] D. Vats, J. M. Flegal, and G. L. Jones. Strong consistency of multivariate spectral variance estimators in Markov chain monte carlo. *Bernoulli*, 24(3):1860–1909, 2018.