Wiley StatsRef:
Statistics Reference Online

Adaptive Monte Carlo Integration

By *Mark Huber*

Keywords: *Monte Carlo, integration, randomized algorithms, estimation, adaptation*

Abstract: Markov chain Monte Carlo uses a Markov chain to generate samples approximately drawn from a target distribution. There are several well-known techniques such as Metropolis–Hastings for building a Markov chain whose limiting distribution equals the target distribution. In Adaptive Markov chain Monte Carlo, the Markov chain used changes slightly from step to step, learning from the states of the chain already visited in order to make the process converge more rapidly to its limiting distribution. While the resulting process will not be strictly Markovian, as long as the adaptation obeys certain rules, the process will still converge to the target distribution. With a well-chosen adaptation, this convergence can be much faster than a basic Markov chain approach.

Monte Carlo methods use random samples from a target distribution in order to estimate quantities of interest. Often these are used to approximate high-dimensional integrals. That is, given the target integral value I, a Monte Carlo method must find a distribution π and function f such that for $X \sim \pi$,

$$I = \int_{\mathbb{R}^n} g(x)\, d\mathbb{R}^n = \mathbb{E}[f(X)]$$

For $X_1, X_2, X_3, \ldots$ independent and identically distributed (iid) with distribution X, the sample average provides an estimate that converges to I as n goes to infinity. That is,

$$\mathbb{E}[f(X)] = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} f(X_i) \tag{1}$$

with probability 1. This is the Strong Law of Large Numbers. The rate of convergence can be controlled by bounding the variance of X.

Often the distribution π is very difficult to sample from. In many applications, it is determined by an unnormalized density. So the target X has density of the form $f_X(x)/Z$. The value of $f_X(x)$ is easy to compute for any x but $Z = \int_{\mathbb{R}^n} f_X(x) d\mathbb{R}^n$ is unknown. Typically Z is #P-complete to compute exactly[1], and so it is unlikely that any efficient method for generating exactly from these types of distributions exists.

Claremont McKenna College, Claremont, CA, USA

DOI: 10.1002/9781118445112.stat07851

1 Markov Chain Monte Carlo

To deal with this difficulty, Markov chains are often used to approximately draw samples from π which are not independent. In this setting, the approximation method is known as Markov chain Monte Carlo, or MCMC for short. For simulation purposes, a Markov chain is defined through the use of a *parameterized update function*. Such a function ϕ takes as input the current state of the chain X_t, a random choice R_t and a parameter θ_t, and returns the next state of the chain. That is,

$$X_{t+1} = \phi(X_t, R_t, \theta_t)$$

Say that a distribution π is *stationary* for parameter θ_t if

$$X_t \sim \pi \Rightarrow \phi(X_t, R_t, \theta_t) \sim \pi$$

As long as the θ_t values are chosen in a deterministic fashion independently of the X_t values, then the $\{X_t\}$ process is a Markov chain.

The ergodic theorem for Markov chains states that under mild conditions (roughly speaking, the chain must be able to get an arbitrary part of the space eventually with positive probability), Equation (1) will still hold even though $X_0, X_1, X_2, \ldots$ are no longer independent, but instead form a Markov chain.

Unlike the case of iid samples, in the Markov chain case, there is no generally applicable method for determining how quickly the sample average converges to the true result. There are, however, heuristics that can give a guide to designing more effective Markov chains.

For example, suppose that the state space is $\mathbb{R}$, $R_0, R_1, \ldots$ are iid **standard normal** random variables, $\theta_t = \sigma > 0$ for all t, and $\phi(x, r, \theta) = x + r\theta$. This is equivalent to saying that the chain takes the current state and adds a normally distributed random variable with **standard deviation** σ to obtain the next state of the chain.

In this chain, if σ is large, each step of the chain will be fairly large. On the other hand, if σ is small, the chain will take small steps in the state space.

The size of the step becomes important when using a commonly used technique for building a Markov chain whose **stationary distribution** is π. This technique is known as **Metropolis–Hastings**[2, 3] (MH). In the MH approach, one Markov chain that does not have stationary distribution π is converted into another Markov chain that does have stationary distribution π by using an accept/reject step. Given X_t, R_t and θ_t, use the update function ν to propose a new possible state Y_{t+1}. Then X_{t+1} is either X_t or Y_{t+1}. The chance that it moves to the proposed state is a ratio based on the density of the proposals.

Given random variable R from the random choices distribution, let $q(x, \cdot, \theta)$ be the density of the random variable $\nu(x, R, \theta)$. Then set

$$\alpha(x, y, \theta) = \frac{f_X(y)q(y, x, \theta)}{f_X(x)q(x, y, \theta)}$$

MH then works as follows.

1. From current state X_t, randomly propose moving to $Y_{t+1} = \nu(X_t, R_t, \theta_t)$.
2. Draw U_t uniform over $[0, 1]$. If $U_t \leq \alpha(X_t, Y_{t+1}, \theta_t)$, then set $X_{t+1} = Y_{t+1}$. Otherwise, $X_{t+1} = X_t$

Now let us continue our earlier example with $R_t \sim \mathsf{N}(0, 1)$, $\theta_t = \sigma$, and the update is $\nu(x, r, \sigma) = x + r\sigma$. Suppose that we wish to create a Markov chain with stationary distribution π that is uniform over Ω. Then $f_X(s) = \mathbb{1}(s \in \Omega)$. Here $\mathbb{1}$ is the indicator function that is 1 if its argument is true and 0 otherwise. This makes $\alpha(x, y) = 1$ if $y \in \Omega$ and $\alpha(x, y) = 0$ if $y \notin \Omega$. This means that from X_t the state Y_{t+1} is proposed. If Y_{t+1} falls in the state space Ω, then the state is accepted, otherwise the state is rejected and the process stays at its current value.

Wiley StatsRef: Statistics Reference Online, © 2014–2019 John Wiley & Sons, Ltd.

DOI: 10.1002/9781118445112.stat07851

So for state space Ω, the update becomes

$$X_{t+1} = X_t + R_t\sigma\mathbb{1}(X_t + R_t\sigma \in \Omega)$$

If σ is too small, then it will take a large number of steps for the Markov chain to spread out over Ω. If σ is too big, then most of the steps will be rejected and the state will spend far too much time staying at the same value. So there is some *goldilocks* value for σ which balances the need to spread out with the need to have steps that are accepted. By analyzing certain limiting **random walks**, Roberts and Rosenthal suggest[4] that an acceptance rate of 23% should be near the goldilocks value.

One approach to finding the value of σ that gives this goldilocks acceptance rate is to run the chain for many different values and keep track of how often the chain is accepting. Then restart the entire process, fixing the parameters θ for the chain. Because we are restarting, all the theory we have for Markov chain applies. The result is a Markov chain where Equation (1) holds. Extra properties such as a law of large numbers and **central limit theorem** might also hold for this type of offline adaptation[5].

The disadvantage is that we have wasted all the steps we took in determining θ. In addition, it may be difficult to decide how many steps to use for adaptation before restarting the chain.

2 Adapting on the Fly

Adaptive Markov chain Monte Carlo (AMCMC) takes a different approach. In an AMCMC algorithm, the value of θ_t is not constant, rather, the choice of how far to try to move is made based on the past history of the chain.

In other words, there exists a choice function f_θ such that

$$\theta_t = f_\theta(X_0, X_1, X_2, \ldots, X_t)$$

and then $X_{t+1} = \phi(X_t, R_t, \theta_t)$ as before.

Because X_{t+1} depends on θ_t which in turn depends on the history $X_0, X_1, \ldots, X_t$, typically the $\{X_t\}$ process is no longer a Markov chain. Each step of the chain is typically still stationary with respect to the target distribution π, but we are no longer guaranteed that (1) holds.

In Ref. 6, Haario *et al.* proposed the following adaptive proposal for a chain operating in d dimensional space. They called their method *adaptive Metropolis* (AM) The parameter θ_t in this case is a **covariance matrix**, defined as

$$\theta_t = \begin{cases} C_0 & t \le t_0 \\ \frac{(2.4)^2}{d}\left[\mathrm{Cov}(X_0, \ldots, X_{t-1}) + \epsilon I_d\right] & t > t_0 \end{cases}$$

where I_d is the d by d identity matrix, $\epsilon > 0$ is a small constant chosen by the user and the sample covariance matrix for points $X_0, \ldots, X_n \in \mathbb{R}^n$ is defined as

$$\mathrm{Cov}(X_0, \ldots, X_k) = \frac{1}{k}\left(\sum_{i=0}^{k} X_i X_i^T - (k+1)\overline{X}_k\overline{X}_k^T\right)$$

where $\overline{X}_k$ is the sample average given by

$$\overline{X}_k = \frac{1}{k+1}\sum_{i=0}^{k} X_i$$

The proposal distribution is a Gaussian with mean equal to the current state and covariance equal to θ_t. That is, $[Y_{t+1}|X_t, \theta_t] \sim \mathrm{N}(X_t, \theta_t)$.

DOI: 10.1002/9781118445112.stat07851

In Ref. 6, the authors give conditions satisfied by their adaptation scheme so that Equation (1) is guaranteed to hold. For general processes, if the adaption is too large or happens too quickly, then the resulting algorithm might fail to have the sample averages converge to the desired result.

The AM algorithm updates all the components of the state simultaneously. Another algorithm developed by Haario *et al.*[7] is the single-component adaptive Metropolis (SCAM).

As the name indicates, in this algorithm, only one dimension of the state is updated as any one time. The remaining format is similar to AM. Let $X_t(i)$ denote the ith component of the current state.

For a given $i \in \{1, \ldots, d\}$, the proposal is as follows. Set $Y_{t+1}(j) = X_t(j)$ for j not equal to i, and then choose $Y_{t+1}(i) \sim \mathsf{N}(X_t(i), v_t(i))$, where

$$v_t(i) = \begin{cases} v_0(i) & t \le t_0 \\ (2.4)[\mathrm{Var}(X_0, \ldots, X_{t-1}) + \epsilon] & t > t_0 \end{cases}$$

and Var denotes the population **variance** estimator

$$\mathrm{Var}(X_0, \ldots, X_{t-1}) = (t-1)^{-1} \sum_{k=0}^{t-1} (X_k(i) - \overline{X}(i)), \quad \overline{X}(i) = (t+1)^{-1} \sum_{k=0}^{t-1} X_k(i)$$

As with the d dimensional update from earlier, the value of v_i will converge to the true value of the variance for that component as t goes to infinity.

3 How to Guarantee Ergodicity

We are interested in convergence, and so we need a means to assess how close the distribution of the current state is to the stationary distribution. To measure the distance between two probability distributions π and ν, we will use the total variation distance:

$$\mathrm{dist}_{\mathrm{TV}}(\pi, \nu) = \sup_A |\pi(A) - \nu(A)|$$

In Ref. 8, it was shown by Roberts and Rosenthal that for Equation (1) to hold it suffices for the adaptation to have two properties: *simultaneous uniform ergodicity* and *diminishing adaptation*.

1. *Simultaneous Uniform Ergodicity.* For all $\epsilon > 0$, there exists N a positive integer such that for all states x and parameters θ, the process started at state x without adaption will have total variation distance at most ϵ from π after N steps.
2. *Diminishing Adaptation.* Let

   $$D_n = \sup_{\text{states } x} \mathrm{dist}_{\mathrm{TV}}([X_{t+1}|X_t = x, \theta_t], [X_t|X_{t-1} = x, \theta_{t-1}])$$

   measure the amount of adaptation at step n of the process. Then the process has diminishing adaption if D_n converges to 0 in probability as $n \to \infty$.

Theorem 1. *If an adaptive MCMC algorithm over state space $\mathcal{X}$ is for any θ_t stationary with respect to π, then if it has the simultaneous uniform ergodicity and diminishing adaptation properties, the process is ergodic and satisfies (1).*

The second condition is under the control of the user, since the choice of adaptation procedure is completely user controlled. The first condition can be trickier to verify, although there are some easy to verify conditions that imply it holds.

Wiley StatsRef: Statistics Reference Online, © 2014–2019 John Wiley & Sons, Ltd.

DOI: 10.1002/9781118445112.stat07851

For instance, suppose that the number of states and the number of parameters are both finite, and there is positive probability from some states of staying at the current state. Then for each parameter value θ, the chain is ergodic. Then N can be set to be the maximum over θ of N_θ, where N_θ is the number of steps needed for the state indexed by parameter θ to get close to the stationary distribution.

Note that because MH usually has a positive chance of staying at the current state, the random variable $X_{t+1}|X_t, \theta_t$ does not typically have a density with respect to some fixed reference measure.

However, the proposed state $Y_{t+1}|X_t, \theta_t$ often does have a density with respect to a fixed reference measure. In Ref. 8, it was shown that as long as the density of $Y_{t+1}|X_t, \theta_t$ exists, and is continuous with respect to some topology where the Cartesian product of the states of the chain and the parameter values is compact, then simultaneous uniform ergodicity holds.

In particular for the case of Haario *et al.*[6], it was assumed that the states of the chain formed a compact set with respect to Lebesgue measure. In that example, the proposed states were **multivariate normal**, which has continuous density with respect to Lebesgue measure. Therefore, the newer result applies to say that the resulting algorithm satisfies (1).

4 Practical AMCMC

One simple way to ensure that simultaneous uniform ergodicity and diminishing adaptation holds is simply to stop adapting the parameter after a fixed number of time steps. This is a compromise between doing a total restart of the chain and continuously adapting throughout the run of the chain. This is the method used in the **R** package `atmcmc`[9].

Related Articles

Markov Chain Monte Carlo Algorithms; **Ergodic Theorems**; **Bayesian Analysis and Markov Chain Monte Carlo Simulation**; **Monte Carlo Methods**; **Markov Chain Monte Carlo, Introduction**; **Markov Chain Monte Carlo, Convergence and Mixing in**; **Markov Chain Monte Carlo Methods**; **Markov Chains**; **Markov Chain Monte Carlo (MCMC)**

References

[1] Valiant, L.G. (1979) The complexity of computing the permanent. *Theoret. Comput. Sci.*, **8**, 189–201.
[2] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., et al. (1953) Equation of state calculation by fast computing machines. *J. Chem. Phys.*, **21**, 1087–1092.
[3] Hastings, W.K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97–109.
[4] Roberts, G.O. and Rosenthal, J.S. (2001) Optimal scaling for various Metropolis-Hastings algorithms. *Stat. Sci.*, **16**, 351–367.
[5] Jones, G.L. (2004) On the Markov chain central limit theorem. *Probab. Surveys*, **1**, 299–320.
[6] Haario, H., Saksman, E., and Tamminen, J. (2001) An adaptive Metropolis algorithm. *Bernoulli*, **7**, 223–242.
[7] Haario, H., Saksman, E., and Tamminen, J. (2005) Componentwise adaptation for high dimensional MCMC. *Comput. Stat.*, **20**, 265–274.
[8] Roberts, G.O. and Rosenthal, J.S. (2007) Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *J. Appl. Probab.*, **44** (2), 458–475.
[9] Yang, J. and Rosenthal, J.S. (2017) Automatically tuned general-purpose MCMC via new adaptive diagnostics. *J. of Comp. Stat.*, **32**, 315–348.

DOI: 10.1002/9781118445112.stat07851