Wiley StatsRef:
Statistics Reference Online

Adaptive Monte Carlo Integration

By *Mark Huber*

Keywords: *Monte Carlo, integration, randomized algorithms, estimation, sequential methods*

Abstract: Adaptive Monte Carlo algorithms use randomness to approximate integrals and sums. They adapt in two ways. Some use a number of samples that automatically adjust themselves based on the problem under consideration. Others change the domain from which the sample is taken based on previous samples. Both kinds of methods typically have robust guarantees on the relative error of the resulting estimates.

Adaptive Monte Carlo refers to algorithms that use a random number of samples, or samples from changing distributions, to estimate an integral or sum. Modern work in the area builds upon the work of Wald[1] in designing sequential tests. The central problem is to find

$$Z = \int_{x \in \mathbb{R}^n} f(x)\, \mathrm{d}\mathbb{R}^n \text{ or } Z = \sum_{x \in \Omega} w(x)$$

where $f(x)$ and $w(x)$ are nonnegative functions.

We can also view these problems as finding the measures of various sets. If μ is the Lebesgue measure, then $\int_{x \in \mathbb{R}^n} f(x)\, \mathrm{d}\mathbb{R}^n = \mu(A)$, where $A = \{(x, y) : x \in \mathbb{R}^n, 0 \le y \le f(x)\}$. That is, the integral is the hypervolume under the curve given by f. Similarly, $\sum_{x \in \Omega} w(x) = \nu(\Omega)$, where $\nu(A) = \sum_{x \in A} w(x)$ where ν is a measure on discrete sets.

Typically, Z grows exponentially in the problem dimension n, and so the goal is to use methods that deliver an accurate approximation that uses something like $\ln(Z)$ samples. In this article, we assume that the user has the ability to draw samples from various distributions. In practice, this is accomplished using Markov chain Monte Carlo (*see* **Markov Chain Monte Carlo Algorithms**; **Markov Chain Monte Carlo (MCMC)**) or perfect simulation algorithms (*see* **Perfect Sampling**).

To be precise, let $\mathcal{I}$ be the input that describes the problem, and $Z(\mathcal{I})$ the true answer. An (ϵ, δ)-randomized approximation scheme ((ϵ, δ)-ras for short) is the output $\hat{Z}(\mathcal{I})$ of a randomized algorithm that has relative error greater than ϵ with probability at most δ. That is,

$$\mathbb{P}\left(\left|\frac{\hat{Z}(\mathcal{I})}{Z(\mathcal{I})} - 1\right| > \epsilon\right) \le \delta \tag{1}$$

For convenience, we will restrict ourselves to the continuous problem of finding $\mu(A)$ in what follows, but these algorithms apply equally well to the discrete case.

Claremont McKenna College, CA, USA

DOI: 10.1002/9781118445112.stat08070

1 Acceptance/Rejection

The simplest Monte Carlo integration algorithm is acceptance/rejection (AR), often referred to as rejection sampling (*see* **Collective Risk Models**). The goal is to find the measure of a region A. Suppose that we have a region B whose finite measure is known and $A \subset B$.

Let X be drawn randomly from the measure μ over B (so for all $C \in B$, $\mathbb{P}(X \in C) = \mu(C)/\mu(B)$.) The probability that $X \in A$ is $p = \mu(A)/\mu(B)$. If this number is estimated by $\widehat{p}$, then $\widehat{\mu}(A) = \widehat{p}\mu(B)$ is an estimate for $\mu(A)$. (It is because $\mu(B)$ is usually very large that it is important to have $\widehat{p}$ close to p in a relative error sense.)

For example, let $Z = \int_{\mathbb{R}} \exp(-|x|^{2.5}/2)\,\mathrm{d}x$. Then, $A = \{(x, y) : x \in \mathbb{R}, 0 \le y \le \exp(-|x|^{2.5}/2)\}$. Let $B = \{(x, y) : x \in \mathbb{R}, 0 \le y \le 1.5\exp(-x^2/2)\}$. Then, to draw (X, Y) uniformly from B (write $(X, Y) \sim \mathsf{Unif}(B)$), first draw X normal with mean 0 and variance 1 (write $X \sim \mathsf{N}(0, 1)$), and then draw Y given X uniformly over the interval from 0 to $1.5\exp(-X^2/2)$ (so $[Y|X] \sim \mathsf{Unif}([0, 1.5\exp(-X^2/2)])$. Then, $(X, Y) \in A$ if and only if $Y \le \exp(-|X|^{2.5}/2)$. The probability that this occurs is $p = Z/[1.5\sqrt{2\pi}]$ so $Z = (1.5\sqrt{2\pi})p$.

For a sample (X, Y), let W be 1 if $X_i \in A$ and 0 otherwise. Then, W has a Bernoulli distribution with mean p (write $W \sim \mathsf{Bern}(p)$). Given a stream of independent, identically distributed (IID) draws from (X, Y), we obtain a stream of IID draws from $\mathsf{Bern}(p)$. AR reduces the problem of integration to the problem of finding the mean of a stream of Bernoulli random variables.

The current fastest algorithm for this problem appeared in Ref. 2 and is called the Gamma Bernoulli Approximation Scheme (GBAS). This algorithm has the optimal constant in the first-order term of the running time. Say that $T \sim \mathsf{Exp}(1)$ if for all $a \ge 0$, $\mathbb{P}(T > a) = \exp(-a)$, and that a random variable has the gamma distribution with parameters k and $k - 1$ if it has density $f(s) = (k-1)^k s^{k-1}\exp(-(k-1)s)/\Gamma(k)$. All draws in algorithms are done independently.

Gamma Bernoulli Approximation Scheme *Input:* ϵ, δ

1. Let k be the smaller integer such that a gamma distributed random variable with parameters k and $k - 1$ falls outside $[(1+\epsilon)^{-1}, (1-\epsilon)^{-1}]$ with probability at most δ. Set $R \leftarrow 0$, $S \leftarrow 0$.
2. While $S < k$, draw $B \leftarrow \mathsf{Bern}(p)$, draw $T \leftarrow \mathsf{Exp}(1)$, set $S \leftarrow S + B$, set $R \leftarrow R + T$
3. Output $\widehat{p} \leftarrow (k-1)/R$

If the inverse gamma distribution is not easily available in Step 1 of GBAS,then

$$k = \lceil 2\epsilon^{-2}(1 - (4/3)\epsilon)^{-1}\ln(2\delta^{-1})\rceil$$

can be used to guarantee that the output is an (ϵ, δ)-ras. No matter what k is used, the expected number of draws used by the algorithm will be k/p, and any extra computation will be linear in the number of samples used.

2 Importance Sampling

AR uses $\{0, 1\}$ random variables to estimate the integral, but in many cases, it is possible to construct $[0, 1]$ random variables with the same mean but smaller variance.

Continuing the example from earlier, for $X \sim \mathsf{N}([0, 1])$, the Bernoulli was 1 if $[Y|X] \sim \mathsf{Unif}([0, 1.5\exp(-X^2/2)]$ was at most $\exp(-|X|^{2.5}/2)$. Suppose, instead we set W to be the probability that the Bernoulli is 1, that is, $W = \exp(-|X|^{2.5}/2)/[1.5\exp(-X^2/2)]$. As before, $\mathbb{E}[W] = \mathbb{P}((X, Y) \in A)$, but the new W will have lower variance than the Bernoulli with the same mean.

Wiley StatsRef: Statistics Reference Online, © 2014–2018 John Wiley & Sons, Ltd.

DOI: 10.1002/9781118445112.stat08070

This idea, called importance sampling (*see* **Importance Sampling**), leads to the necessity of building an (ϵ, δ)-ras for μ that is the mean of a $[0, 1]$ random variable X. Such an algorithm for $[0, 1]$ random variables was given by Dagum et al.[3], we will refer to their algorithm here as DKLR. Let $\mu = \mathbb{E}[W]$. Then, DKLR is adaptive because of the first step, which adjusts the number of samples used to get a rough estimate of μ before refining the estimate, and the third step, where the number of samples depends on the outcome of the first two steps. Updating DKLR using the more recent GBAS for the first step yields the following algorithm.

DKLR *Input:* ϵ, δ

1. Use GBAS to build $\widehat{\mu}_1$ that is a $(\min\{1/2, \epsilon^{1/2}\}, \delta/3)$-ras for μ.
2. Set $\Psi_2 \leftarrow 8(e-2)(1+\epsilon^{1/2})(1+2\epsilon^{1/2})\ln(3\delta^{-1})\epsilon^{-2}$, $N \leftarrow \Psi_2 \epsilon/\widehat{\mu}_1$, and $S \leftarrow 0$. For i from 1 to N do: draw X_1 and X_2 independently from X and set $S \leftarrow S + (X_1 - X_2)^2/2$.
3. Set $\widehat{\rho} \leftarrow \max\{S/N, \epsilon\widehat{\mu}_1\}$. Set $N = \Psi_2\widehat{\rho}/\widehat{\mu}_1^2$ and $S \leftarrow 0$. For i from 1 to N do: draw X' from X and set $S \leftarrow S + X'$. When complete, output $\widehat{\mu} \leftarrow S/N$.

3 Tootsie Pop Algorithm

The abovementioned AR and importance sampling algorithms work well for estimating $\mu(A)/\mu(B)$ when A and B are comparable in size, but require a number of samples (on average) proportional to $\mu(B)/\mu(A)$. In many applications, the size of this ratio grows exponentially in the dimension of the problem, and so methods that run in time polynomial in $\ln(\mu(B)/\mu(A))$ are needed. One such method is the Tootsie Pop Algorithm (TPA) introduced in Refs 4, 5.

In order to use this method, it is necessary to have a collection of sets $\{A(\beta)\}_\beta$ indexed by parameter β that smoothly interpolates between A and B. That is, there must be a β_A and β_B, so that $A(\beta_A) = A$ and $A(\beta_B) = B$. Also, $\mu(A(\beta))$ should be a continuous, increasing function of β. With such sets, TPA works as follows.

TPA *Input:* β_A, β_B

1. Start with $N \leftarrow 0$ and $\beta \leftarrow \beta_B$
2. Draw a random draw X from μ conditioned to lie in $A(\beta)$
3. Let $\beta \leftarrow \inf\{b : X \in A(b)\}$, and $N \leftarrow N + 1$
4. If $\beta \leq \beta_A$, stop and output $N - 1$, otherwise go back to step 2.

The remarkable fact about the output of TPA is that $N - 1$ has a Poisson distribution with mean $\ln(\mu(B)/\mu(A))$. We will write $N - 1 \sim \mathsf{Pois}(\ln(\mu(B)/\mu(A)))$. Each run of TPA will require $\ln(\mu(B)/\mu(A)) + 1$ samples on average. As Poisson random variables have the same mean and variance, on the order of $\ln(\mu(B)/\mu(A))$, draws from a Poisson distribution are necessary to obtain an estimate of the mean within a fixed additive error. Exponentiating then gives an estimate of the mean within a fixed relative error.

To see how this might be used for integration, consider again the example of estimating $\mu(B) = \int_{x\in\mathbb{R}} f(x)\,dx$, where $f(x) = \exp(-|x|^{2.5}/2)$ and $B = \{(x, y) : x \in \mathbb{R}, y \in [0, f(x)]\}$. Let $A(\beta) = \{(x, y) : x \in [-\beta, \beta], y \in [0, f(x)]\}$. Set $\alpha > 0$. Then, TPA operates by first setting β to be ∞, then drawing X from unnormalized density f conditioned to lie in $[-\beta, \beta]$ using perfect slice sampling[6]. The next value of β is then just $|X|$. Continue in this manner until $X \leq \alpha$. The number of samples needed for this to occur is $\mathsf{Pois}(\ln(\mu(B)/\mu(A(\alpha))))$. Then, use the stream of Poisson random variables to obtain an estimate $\widehat{r}_1$ for $\mu(B)/\mu(A(\alpha))$.

Next, consider the set $C = \{(x, y) : x \in [-\alpha, \alpha], y \in [0, 1]\}$. It is easy to draw uniformly from C and $\mu(C) = 2\alpha$. For small α, $\mu(A(\alpha))/\mu(C)$ will be close to 1, so GBAS can be used to quickly draw an estimate $\widehat{r}_2$ for $\mu(A(\alpha)/[2\alpha]$ that is highly accurate. Then, $\widehat{r}_1\widehat{r}_2(2\alpha)$ is an accurate estimate for $\mu(B)$.

DOI: 10.1002/9781118445112.stat08070

Note that the only requirement here was that C was built around a local mode. So, the same approach works equally well in n dimensions as in one.

4 Nested Sampling

Skilling[7] introduced an Adaptive Monte Carlo technique aimed directly at integration of Bayesian posterior distributions. Here, the integrand f is a product of a prior density π over a parameter vector and a likelihood of the parameter vector given the data. That is, the integral has the form

$$Z = \int_{\theta \in S} L(\theta)\pi(\theta)\,\mathrm{d}\theta$$

where S is the space of all allowable parameters. Note that $Z = \mathbb{E}[L(\Theta)]$, where Θ is a random draw from the prior density π. So, the basic Monte Carlo approach is to draw $\Theta_1, \ldots, \Theta_N$ IID from the prior, and then use the sample average $(L(\Theta_1) + \cdots + L(\Theta_N))N^{-1}$ to estimate Z.

The problem with this approach is that this sample average could have very high variance, especially if the densities L and π do not have much overlap. So, Skilling introduced the idea of *nested sampling*. Unlike the earlier methods presented, Skilling's approach does not yield an (ϵ, δ)-ras, however, it is very fast in practice on many problems.

The idea is as follows. Suppose that we break the problem of estimating $\mathbb{E}[L(\theta)]$ into pieces. For any positive random variable X and $\lambda_1 > 0$,

$$\mathbb{E}[L(\Theta)] = \mathbb{E}[L(\Theta)|L(\Theta) \in (0, \lambda_1]]\mathbb{P}(L(\Theta) \in (0, \lambda_1]) + \mathbb{E}[L(\Theta|L(\Theta) > \lambda_1)\mathbb{P}(L(\Theta) > \lambda_1)$$

Now, suppose that we draw N IID samples of Θ from the prior, and let $\lambda_1 = \min_i\{L(\Theta_i)\}$. Then, $\mathbb{P}(L(\Theta) \in (0, \lambda_1]) \approx 1/(N+1)$, and $\mathbb{E}[L(\Theta)|L(\Theta) \in (0, \lambda_1]] \approx \lambda_1$. Removing the value Θ_ℓ (assuming no ties in likelihood) where $L(\Theta_\ell) = \lambda_1$ leaves $N-1$ values conditioned to have $L(\Theta) \in (\lambda_1, \infty)$. So, by drawing one more value Θ from the prior conditioned to have $L(\Theta) > \lambda_1$, we can replenish our sample back up to N values, and we can then estimate $\mathbb{E}[L(\Theta)|L(\Theta) > \lambda_1]$ in the same way as the first step.

Continue in this manner for j steps. At this point, all that is leftis to build an estimate for $\mathbb{E}[L(\Theta)|L(\Theta) > \lambda_j)]\mathbb{P}(L(\Theta) > \lambda_j)$. By choosing j large, our estimate of $\mathbb{P}(L(\Theta) > \lambda_j) = (1 - 1/(N+1))^j$ is quite small, and so the basic Monte Carlo estimate for $\mathbb{E}[L(\Theta)|L(\Theta) > \lambda_j)]$ of $(L(\Theta_1) + \cdots + L(\Theta_2))N^{-1}$ is sufficient. Using $1 - 1/(1+N) \approx \exp(-1/N)$ then yields the presentation of nested sampling in Ref. 7.

Nested Sampling *Input: j*

1. Draw $\Theta_1, \ldots, \Theta_N$ IID from the prior density π. Set $Z = 0$.
2. For i from 1 to j do the following:
   a. Let $\lambda_i = \min_k\{L(\Theta_k)\}$.
   b. Set $Z \leftarrow Z + \lambda_i \exp(-(i-1)/N)(1 - \exp(-1/N))$.
   c. For $\ell = \arg\ \min_k\{L(\Theta_k)\}$, replace Θ_ℓ with a new point drawn from the prior distribution conditioned to have $L(\Theta_\ell) > \lambda_i$.
3. Set $Z \leftarrow Z + (L(\Theta_1) + \cdots + L(\Theta_n))N^{-1}\exp(-j/N)$.

Refinements of this basic algorithm exist, for instance see Ref. 8 for a description of the MultiNest algorithm that expands upon nested sampling.

Wiley StatsRef: Statistics Reference Online, © 2014–2018 John Wiley & Sons, Ltd.

DOI: 10.1002/9781118445112.stat08070

Related Articles

Markov Chain Monte Carlo Algorithms; **Collective Risk Models**; **Perfect Sampling**; **Importance Sampling**; **Markov Chain Monte Carlo (MCMC)**.

References

[1] Wald, A. (1947) *Sequential Analysis*, Dover Publications, Inc., New York.
[2] Huber, M. (2016) A Bernoulli mean estimate with known relative error distribution. Random Structures Algorithms, arXiv:1309.5413. To appear.
[3] Dagum, P., Karp, R., Luby, M., and Ross, S. (2000) An optimal algorithm for Monte Carlo estimation. *SIAM J. Comput.* **29** (5), 1484–1496.
[4] Huber, M.L. and Schott, S. (2010) Using TPA for Bayesian inference, in *Bayesian Statistics 9*, Oxford University Press, Oxford, pp. 257–282.
[5] Huber, M.L. and Schott, S. (2014) Random construction of interpolating sets for high dimensional integration. *J. Appl. Probab.* **51** (1), 92–105. arXiv:1112.3692.
[6] Mira, A., Møller, J., and Roberts, G.O. (2001) Perfect slice samplers. *J. R. Stat. Soc., Ser. B Stat. Methodol.* **63**, 593–606.
[7] Skilling, J. (2006) Nested sampling for general Bayesian computation. *Bayesian Anal.* **1** (4), 833–860.
[8] Feroz, F., Hobson, M.P., and Bridges, M. (2009) MultiNest: an efficient and robust Bayesian inference tool for cosmology and particle physics. *Mon. Not. R. Astron. Soc.* **398**, 1601–1614.

DOI: 10.1002/9781118445112.stat08070