Importance Sampling Tutorial

Nathan Sanford

November 24, 2019

This short tutorial explains the basics of importance sampling by introducing the concept of Monte Carlo sampling, how importance sampling can be used to apply Monte Carlo sampling to rare events and where these methods are being used in current science. As part of this, I introduce discuss my own research on rare events in a mode-locked laser. This tutorial will hopefully give someone unfamiliar with importance sampling a sense of how powerful it can be, and where to start with applying it to problems involving random walks.

1 Monte Carlo Integration

Monte Carlo integration allows one to numerically approximate an integral by drawing a large number of random samples from a probability distribution. For our purposes, consider an integral of the form

$$I = \int_D f(\mathbf{x}) p(\mathbf{x})\, d\mathbf{x} \tag{1}$$

where the integral is over some region D in n-dimensional space and f and p are functions on that space. Additionally, we require that p is a probability density function (PDF), meaning essentially that it is non-negative and integrates to 1 over D. Monte Carlo integration works to approximate I by randomly sampling from p and then averaging f. For instance, after drawing N samples of n-dimensional real random variables from p, our approximation to I is

$$\hat{I}_N = \frac{1}{N} \sum_{l=1}^{N} f(\mathbf{X}_l) \tag{2}$$

where $\mathbf{X}_l$ is the l^{th} sample drawn from $p(\mathbf{x})$. This works as the integral I in Eq. (1) is also an *expected value integral*.

To make this concrete, let's consider the example of computing the one-dimensional integral

$$\mathcal{I} = \int_0^\infty e^{-x} \cos(x)\, dx. \tag{3}$$

If we take $p(x) = \exp(-x)$ to be the probability distribution (note that it is a valid distribution as it is nonnegative and its total integral on 0 to ∞ is 1), then the above integral is equivalent to the expectation $\mathcal{I} = E[\cos(X)]$ with X drawn from $p(x)$. Therefore, we can compute the value of $\mathcal{I}$ via Monte Carlo integration by repeatedly drawing samples from the exponential distribution, taking the cosine of each sample, and then taking the mean of all the results. After a sufficiently large amount of samples, we obtain an estimate near the true value of $\mathcal{I} = 1/2$. Code for performing this simple example is contained in the folder `Monte_Carlo_in_C` (written in C). There is code for performing this calculation in serial and **in parallel** using two platforms: Open MPI which utilizes multiple CPUs or CUDA which runs on NVIDIA brand GPUs.

Now, the law of large numbers says that $\hat{I}_N$ approaches the true value of I as $N \to \infty$. However, it will do so rather slowly. The standard deviation of $\hat{I}_N$ is $\mathcal{O}(N^{-1/2})$, meaning that if the number of samples is multiplied by 100, the expected error in the result will only be cut by a factor of 10. That's not a problem for simple examples, but can be especially pernicious in situations where the integration requires many samples to be drawn from the tail of a probability distribution. The relatively low probabilities of such samples exacerbates the scaling of the simulations and makes many applications infeasible.

As a simple example, let's consider the case of a one-dimensional random walk with normally distributed steps. This problem can be described by a sum of random variables that all are drawn from the Gaussian distribution:

$$Z_J = \sum_{j=1}^{J} X_j \quad \text{where the } X_j \text{ are i.i.d. normally distributed R.V.s} \tag{4}$$

$$\text{so } p_G(x_j; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(-\frac{(x_j - \mu)^2}{2\sigma^2} \Big) \quad \text{for all } j.$$

For simplicity let the mean μ of every step be 0 and the variance σ^2 of every step be 1. This dictates that the probability density function of the result after J steps Z_J is also Gaussian-distributed with mean 0 and variance J (see this Wikipedia page for more on the analytical derivation of this result). In other words, the PDF for Z_J is given by $p_G(z_J; 0, J) = (2\pi J)^{-1/2} \exp[-z_J^2/(2J)]$. We can simulate this process with Monte Carlo simulations by drawing samples of the random walk (by drawing J standard normals and summing them) and then binning them to create a histogram which can be compared with the analytical result for the PDF. Therefore, the integral we are computing using Monte Carlo is

$$I = \int f(z_J) p_J(z_J)\, dz$$

where $p_J(z_J)$ represents drawing samples of the random walk and $f(z_J)$ represents the binning of samples. The binning process can be thought of as using a series of indicator functions (functions that are 1 over the width of a bin and are 0 elsewhere) in place of f.

The results of Monte Carlo simulations are shown in figure 1 and reflect that the simulation method struggles to capture large deviations in the random walk. There are

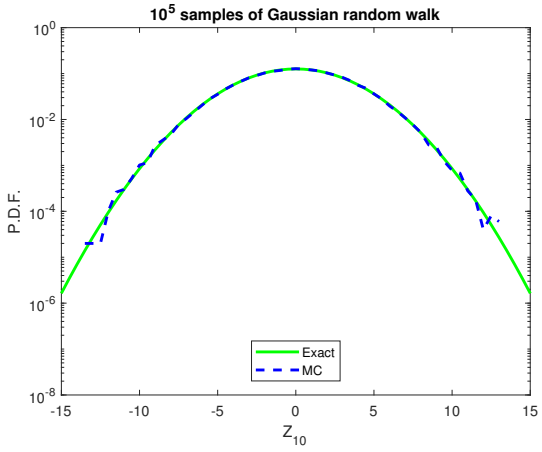

Figure 1: Exact and numerical approximations of the PDF of the Gaussian random walk from Eq. (4) with $N = 10$ steps and 10,000 samples taken for the numerical approximation.

relatively few samples that deviate far from the mean of 0, which is evidenced by the fact that the PDF estimate is jagged near its edges. There were 10^5 samples drawn for this simulation, so naturally we expect it to only capture probabilities to roughly the level of 10^{-5} in the PDF.

2 Importance Sampling Basics

The slow convergence in Monte Carlo simulations for small probabilities can be caused by a mismatch between the function f and the distribution p in Eq. (1). Obviously, the integral I will be dominated by regions where f is approximately maximal, and in many cases p is not maximal in the same region. When this happens, the "important" regions for f are undersampled and the estimate $\hat{I}$ in Eq. (2) converges very slowly. An augmentation to MC simulation called importance sampling (IS) corrects for this by introducing an artificial probability distribution, called a biasing distribution, into the simulation from which samples are drawn. The goal of this technique is to replace p with a distribution that heavily weights regions of probability space where events of interest occur most often.

The insight which makes IS possible is that Eq. (1) can be rewritten as

$$I = \int_D f(\mathbf{x}) \frac{p(\mathbf{x})}{p^*(\mathbf{x})} p^*(\mathbf{x})\, d\mathbf{x}. \tag{5}$$

The function p^* is the biasing distribution and will be treated as given for this discussion. The choice of biasing distribution is an application specific problem and investigating rationales for choosing biasing distributions is a rich field of inquiry in many contexts. Once a biasing distribution is chosen we perform MC simulations by drawing samples

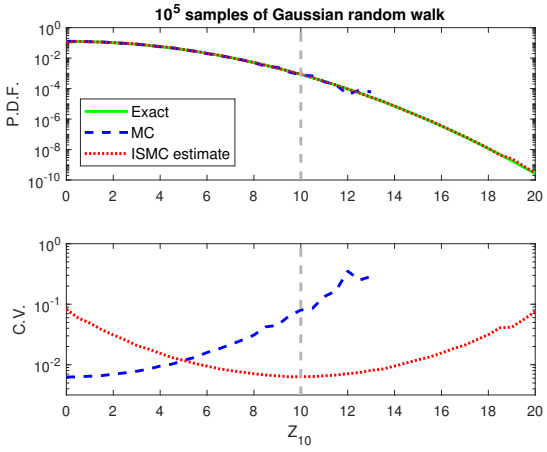

Figure 2: MC and ISMC simulations of the Gaussian random walk around $Z_{10} = 10$ with 10^5 samples in each. The top panel shows the PDF's from each simulation, as well as the exact solution for comparison. The CV's shown in the lower panel demonstrate that the ISMC simulation converges best around the target of $Z = 10$ while the MC simulation's convergence grows increasingly poor the further we get from $Z = 0$.

from p^* and then computing

$$\hat{I}_N = \frac{1}{N} \sum_{l=1}^{N} f(\mathbf{X}_l) \frac{p(\mathbf{X}_l)}{p^*(\mathbf{X}_l)}. \tag{6}$$

In an ISMC simulation, the sample results are now weighted by the likelihood ratio, or

$$L(\mathbf{X}_l) = \frac{p(\mathbf{X}_l)}{p^*(\mathbf{X}_l)}, \tag{7}$$

which corrects for the fact that samples were drawn by the biasing distribution and gives results equivalent to if the original distribution had been used.

The success of importance sampling is principally judged by two factors: that the resulting integral or probability computed is reasonable, and that the convergence is suitably fast. The second criterion can be addressed graphically by examining the variance of the IS integrand via the coefficient of variation (CV). The CV is a measure of the simulation's sample standard deviation divided by the probability value. In this way, it adjusts for the probability being simulated, and gives a readily comparable assessment of the simulation's variance.

To get a sense of how this works in practice, let's return to the Gaussian random walk. Suppose we were concerned with the probability that a walker moves far to the right, say to the area around 10. We can do this by performing importance sampled Monte Carlo (ISMC) simulations where the biasing distribution for each step is $p_G(x; 1, 1)$, or a Gaussian distribution with mean 1. This means that, on average, the walker will take a step of 1 to the right in each step and end up at roughly $Z = 10$ after 10 steps. Figure 2 shows the results of such simulations. The top panel shows the computed PDF's (MC and ISMC) after 10^5 samples in the vicinity of $Z_{10} = 10$ and the bottom panel shows the

CV's of each as well. The ISMC simulation matches the exact PDF closely down into the tales, at probability levels far lower than can be achieved by an equivalent amount of samples in MC. Drawing biased samples and then weighting them with the likelihood ratio is a trivial amount of work to be done for the ability to capture probabilities that would require many trillions of samples under normal conditions. Running the file `ISGaussian.m` in MATLAB demonstrates the efficiency of this approach: it can be run with or without performing ISMC (it always performs MC) and the code executes virtually as quickly either way.

A number of augmentations to importance sampling exist. One especially interesting technique is *multiple importance sampling*, where multiple biasing distributions are used simultaneously. Samples are then weighted using the balancing heuristic, a weighting scheme that allows for the combination of multiple likelihood ratios.[1] Funnily enough, this method was developed in the context of natural light rendering in computer graphics, and earned its creator, Eric Veach, a Scientific and Engineering Oscar due to its use in Pixar films. It can easily be used in our Gaussian walk example to capture the entire PDF by including multiple biasings that target various areas of the distribution. A multiply importance sampled Monte Carlo (MISMC) simulation with six targets between $Z = -15$ and $Z = 15$ is seen in figure 3. The contributions of each target are shown in black, and are combined in order to give an ensemble estimate that closely matches the entire distribution very closely down to probability density levels of 10^{-10}. The CV in the bottom panel confirms this impression, as the MISMC estimate's CV is at a low uniform level throughout $-15 < Z < 15$, indicating an even spread of samples throughout the region. This is preferable to the MC simulation, even though it has a lower CV near 0, because more samples are spread into the tails, as desired.

3 Research for Rare Events in Complicated Systems

The treatment of importance sampling up to this point has been very scant on the toughest aspect of IS: determining the biasing distribution to use in a particular simulation. The reason for this is that it is often the most difficult portion of a particular importance sampling project. Broadly speaking, however, the biasing distribution is usually arrived at via the solution to a constrained optimization or optimal control problem. These problems ask for the most likely way for noise to lead to a rare event of interest, usually by minimizing the zero-mean noise. For instance, in the Gaussian random walk example, the biasing distribution for arriving at $Z_{10} = 10$ is the solution to

$$\min_X \sum_{j=1}^{10} X_j^2 \quad \text{where} \quad \sum_{j=1}^{10} X_j = 10.$$

The solution is the same as described previously: namely that $X_j = 1$ for all j.

[1] For more, see Eric Veach, Robust Monte Carlo Methods for Light Transport Simulation, Ph.D. thesis, Stanford University (1997).

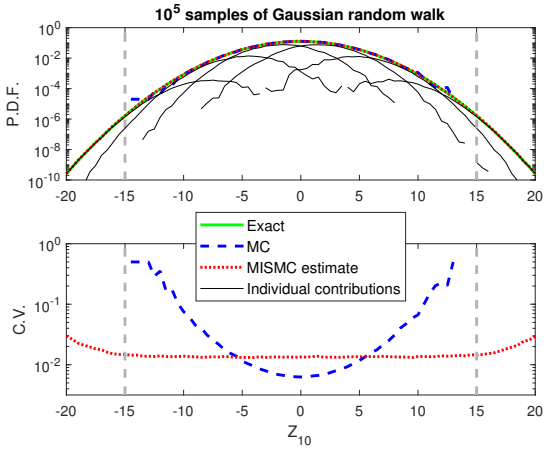

Figure 3: MC and MISMC simulations of the Gaussian random walk with 10^5 total samples in each. The MISMC simulations used 6 evenly spaced distribution targets between $-15 < Z_{10} < 15$. The top panel shows the PDF's from each simulation, as well as the exact solution for comparison. The thin black lines are contributions from the individual biasings, which gives a sense of how the balance heuristic works to create a composite estimate. The CV's shown in the lower panel demonstrate that the MISMC simulation converges evenly throughout the target region within $|Z| < 15$ while the MC simulation's convergence grows increasingly poor the further we get from $Z = 0$.

A vast theory of rare events has developed with this optimization framework. Many researchers have taken to applying the most advanced optimization algorithms to construct rare event paths in general systems.[2] The work in nonlinear optics that I've been a part of seeks to use previously developed mathematical theory, called soliton perturbation theory, to formulate rare event optimal path problems in approximate frameworks that nonetheless perform quite well in real systems. The benefits of using soliton perturbation theory are chiefly that it is computationally less expensive to compute error paths and that such paths are more interpretable since they are in terms of known pulse attributes. An example of this is a mode-locked soliton laser model which employs active mode-locking, a mechanism to stabilize the generation of ultra-short pulses. Such laser sources are potentially applicable to cutting-edge sciences and technologies such as optical clocks and optical storage.[3] In my work I found that a proposed mechanism for underdamped (or oscillatory) mode-locking in a fiber ring laser allowed for exit paths that accumulated oscillations as the propagation distance (or number of passes around the laser ring) increased, as seen in figure 4, which made applying importance sampling for longer distances increasingly complicated.[4]

[2] A good example of such work is Heymann, M. and Vanden-Eijnden, E. (2008), The geometric minimum action method: A least action principle on the space of curves. Comm. Pure Appl. Math., 61: 1052-1117.

[3] Papers on such work include this paper on optical clocks and this paper on optical storage.

[4] N. Sanford, G.M. Donovan, and W.L. Kath. *Slip Rates and Slip Modes in an Actively Mode-Locked Laser*, submitted 2019.

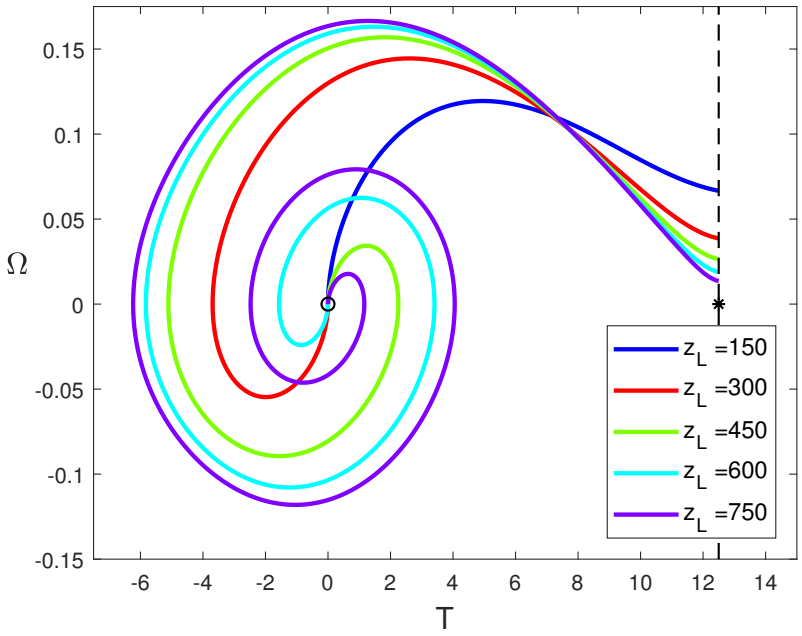

Figure 4: Error paths for the underdamped mode-locked laser. As the propagation distance z_L increases, the number of oscillations also increases in error paths for a pulse to slip relative to the timing signal of the laser. Such errors are pernicious in the decoding of bit-streams, as pulses do not arrive in their intended bit-slots, which can lead to transmission corruption.