Chapter 5

Monte Carlo

Monte Carlo methods are a collection of techniques that use pseudo-random (computer simulated) values to approximate solutions to mathematical problems, typically for evaluating analytically intractable integrals or summations. In probability and statistics, Monte Carlo methods are used to approximate properties of distributions, most commonly evaluating analytically intractable expected values. They may be used for a wide range of statistical problems, from p-value calculations in frequentist hypothesis testing to approximating properties of posterior distributions in Bayesian inference and evaluating intractable likelihood functions for maximum likelihood estimation. The ability to use computers to simulate from complicated probability models has had an enormous impact on probability and the application of statistical methods to scientific problems.

5.1 Simple (Independent) Monte Carlo

We will begin by focusing on using Monte Carlo methods for computing expectations. Suppose we want to evaluate $\mu = E_f(g(x))$ where if f is a density, $\mu = \int g(x)f(x)dx$ and if f is a probability mass function, $\mu = \sum_x g(x)f(x)$. The simplest Monte Carlo method for evaluating this expectation is as follows:

1. Simulate independent draws $X_1, \ldots, X_n$ from f.
2. Let $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^n g(X_i)$.
3. If $\mu = E_f(|g(x)|) < \infty$, we have:
   $\hat{\mu}_n \to \mu$ a.s. (almost surely) by the Strong Law of Large Numbers (S.L.L.N.) That is,

$$P(\lim_{n\to\infty} \hat{\mu}_n = \mu) = 1. \tag{5.1}$$

Furthermore, if $\sigma^2 = \text{Var}_f(g(x)) < \infty$, we can establish a convergence rate for this estimate, that is we can establish how quickly $\hat{\mu}_n$ converges to μ from the **Central Limit Theorem**:

$$\sqrt{n}(\hat{\mu}_n - \mu) \to N(0, \sigma^2) \text{ in distribution} \tag{5.2}$$

Example 1: Suppose we want to calculate Pr$(-1 < X < 0)$ when X is a Normal(0,1) random variable. We could easily do this by Monte Carlo since a probability may be written as an expectation of an indicator function. That is, for a random event X, $P_f(X) = E_f(I(X))$ where the indicator $I(X) = 1$ if X and 0 otherwise.

- Generate $X_1, \ldots, X_n \sim N(0, 1)$.
- Compute the estimate
$$\hat{\mu}_n = \frac{\sum_{i=1}^n 1(-1 < X_i < 0)}{n},$$
which is simply the proportion of times $X_i \in (-1, 0)$ for sampled values $X_1, \ldots, X_n$.

For large enough n, $\hat{\mu}_n$ will be very close to Pr$(-1 < X < 0)$. Of course, Monte Carlo is not really needed for this toy problem since statistical software can easily calculate such probabilities.

Example 2: Suppose we want to conduct a simple hypothesis test to see if the correlation ρ between two random variables is significant. Assume that the two random variables X and Y come from a bivariate normal distribution, and we observe 30 data points $(X_1, Y_1), \ldots, (X_{30}, Y_{30})$, and want to conduct a hypothesis test based on these data. The sample correlation, $\hat{\rho}$, is the test statistic. For these data, $\hat{\rho} = 0.3$. To find the associated p-value, we need to find the probability $P(\hat{\rho} > 0.3)$ under the null hypothesis that there is no correlation ($\rho = 0$), and the alternative that $\rho > 0$. To calculate this probability, we would need to know the sampling distribution of $\hat{\rho}$ under the null hypothesis that $\rho = 0$. This null distribution is not easy to calculate, but it is given in Anderson (2003, pg.125). The sample corrrelation coefficient $\hat{\rho}$, for a sample of size N from a bivariate Normal with mean $\boldsymbol{\mu}$ and correlation ρ, depends only on ρ and N (not on $\boldsymbol{\mu}$ or the marginal variances), and is given by:

$$f(\gamma) = \frac{2^{N-3}(1-\rho^2)^{0.5(N-1)}(1-\gamma^2)^{-0.5(N-4)}}{\pi\Gamma(N-2)} \sum_{i=0}^{\infty} \Gamma\left(\frac{N+\alpha-1}{2}\right)^2 \frac{(2\rho\gamma)^\alpha}{\alpha!},$$

where $\gamma \in (-1, 1)$. Now, for $\rho = 0$, the sampling distribution simplifies to:

$$f(\gamma) = \frac{2^{N-3}(1-\gamma^2)^{-0.5(N-4)}}{\pi\Gamma(N-2)} \Gamma\left(\frac{N-1}{2}\right)^2.$$

Finding the above distribution is a non-trivial and time consuming problem, and even though we can assume that we did not have to work to find the distribution (since the theory has already been worked out), finding the p-value for this hypothesis test would still involve integrating the above density over the interval $(0.3, \infty)$. However, if we simply use the fact that the distribution of $\hat{\rho}$ only depends on ρ and the sample size N, a Monte Carlo solution to this problem is very simple:

- To draw a single sample of $\hat{\rho}$ from the null distribution, generate a sample $X_1, \ldots, X_N \sim N(0,1)$ and a sample $Y_1, \ldots, Y_N \sim N(0,1)$. In R you would use the command `xs=rnorm(30,0,1)`, for instance. Find the sample correlation r_1 based on the pairs $(X_1, Y_1), \ldots, (X_N, Y_N)$. Repeat this process m times to obtain m sample correlations $r_1, \ldots, r_m$, generated from the null distribution.

- Compute the estimate
$$\hat{\mu}_n = \frac{\sum_{i=1}^{m} 1(r_k > 0.3)}{m}.$$
For large enough m, this is an accurate estimate of the desired p-value.

Note that the only theory necessary was recognizing that the sampling distribution of $\hat{\rho}$ depends only on ρ and N, which is easy to prove by the invariance of the distribution of $\hat{\rho}$ to affine transforms of a bivariate normal random variable (see Anderson (2003) for details). It was not necessary to derive the complicated formula for the sampling distribution above, nor was it necessary to compute the integral; the p-value is easily estimated through a simple Monte Carlo procedure. Of course, since the exact distribution is available here, and the p-value only involves a 1-dimensional integral, it is possible to do this by using numerical integration procedures. In more complicated situations, Monte Carlo will often be the only solution.

5.2 Monte Carlo standard errors

Informally, (5.2) states that the distribution of $\hat{\mu}_n$ approaches a Normal distribution, N($\mu, \sigma^2/n$). Hence, to obtain confidence intervals and error estimates for the estimator $\hat{\mu}_n$, we need to estimate σ^2. Monte Carlo standard error is an assessment of the error of our Monte Carlo estimator. For the simple independent Monte Carlo scenario above, we can easily estimate σ^2 by the sample variance, $\hat{\sigma}^2$. Then, the estimate of Monte Carlo standard error is simply $\hat{\sigma}/\sqrt{n}$. Since $\hat{\sigma}^2$ is a consistent estimator of σ^2, we can use Slutsky's Theorem and (5.2) to obtain asymptotic 95% confidence intervals for Monte Carlo estimates in the usual way: $\hat{\mu}_n \pm 1.96\hat{\sigma}/\sqrt{n}$. In practice we would not know σ^2 and would instead have to use an estimate of it. As long as σ^2 is replaced by a consistent estimate of σ^2, (5.2) still holds.

We note the following:

1. The independence requirement for the samples is unnecessarily restrictive as we will see later on when we discuss Markov chain Monte Carlo methods. The S.L.L.N. will typically hold under similar conditions for the dependent case but the C.L.T. may not hold and estimating σ^2 is typically very difficult.

2. f may be multivariate so the random variable for which f is the probability distribution may be multidimensional.

3. The accuracy of Monte Carlo estimates is *independent of the dimensionality* of the space sampled. This is clear from (5.2) and is a very important reason why Monte Carlo methods are very powerful. However, high dimensionality can cause other serious problems for Monte Carlo methods, primarily because obtaining draws from high dimensional distributions can often be very difficult.

As is clear from the preceding discussion, Monte Carlo is essentially like basic statistics in that we are using sample information to estimate quantities of interest with respect to the distribution of the sample information. Monte Carlo is a computational technique where we treat the probability distributions as fixed and are only interested in learning about that specific distribution via computer simulations. Strictly speaking, Monte Carlo methods are entirely about probability calculations. Because probability calculations like p-values and expectations with respect to posterior distributions are central to statistical inference, Monte Carlo methods also happen to be very useful for statistical inference.

5.3 Random variate generation

5.3.1 Uniform random variates

Monte Carlo methods rely on our ability to simulate random draws from distributions. How do we simulate random samples from an arbitrary distribution f? The general approach for generating such samples may be outlined as follows:

1. Generate $U_1, U_2, \ldots$ independent Uniform(0,1) random variables.

2. Generate $X \sim f$ using $U_1, U_2, \ldots,$

The first step requires that we generate uniform random variates on a computer. This is achieved via sophisticated deterministic algorithms. Essentially these algorithms are used to produce a sequence of values in $[0,1]$ such that these values imitate the properties of a sequence of i.i.d. Uniform(0,1) random variates. A simple uniform random variate generator uses a deterministic algorithm of the following form:

1. Use an initial value, X_0, called 'the seed'. (in R, the seed is set using the function 'set.seed' and it can be obtained using 'save.seed').

2. Construct a sequence $X_1, X_2, \ldots$ by setting $X_{n+1} = (aX_n + c)$ modulo m where $a, c, m > 0$. So $X_n \in \{0, \ldots, m-1\}$ and $X_n/m \sim$Unif(0,1).

Examples of more sophisticated random number generators are Knuth's TAOCP (1997/2002), Marsaglia's Super-Duper (1970s), and the Mersenne-Twister (Matsumoto and Nishimura (1998)). We note the following properties of uniform random variates generated by such algorithms.

- They are repetitive. After enough draws, the sequence will start to repeat itself (as soon as $X_k = X_0$ for some k). The number of draws before repetition is known as **the period** of the generator. This is an important property to note, particularly when doing large scale simulations. However, this is often not a problem since modern generators have large periods. For instance, the default generator used in `R`, the Mersenne-Twister (Matsumoto and Nishimura (1998)), has period $2^{19937}-1$. In practice, the fact that we can control the seed is an advantage in that our computer experiments and simulations are reproducible (the same seed results in the same sequence every time).

- They are not independent. Clearly, each value depends on the last value of the sequence. This is not an issue as long as they give the appearance of being an independent sequence, that is, as long as statistical tests are unable to distinguish between this sequence and a sequence of truly independent uniform random variates.

- They are not truly from a continuous distribution. This is not a real issue at all since a computer is incapable of truly representing an uncountable set of values anyway.

5.3.2 Non-uniform random variates

Random variate generation for all other distributions use algorithms based on sequences of uniform(0,1) random variates. For instance, the **inverse-cdf method** is a simple method for generating non-uniform random variates. It works as follows:

1. Draw $U \sim$Uniform(0,1).
2. Suppose F is the cdf of the desired random variable. Set $X = F^{-1}(U)$ (where $F^{-1}(u) = \{x$ s.t. $F(x) = u\}$).

Theorem 1. *X is a draw from F.*

Proof. $P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x)$, where the last equality follows from the fact that U is a Unif(0,1) random variable. □

Unfortunately, this is a useful algorithm only if we are interested in univariate random variables and F has an explicit inverse. Hence, this works for densities like the cauchy, exponential, weibull or logistic, but not the normal, beta or many gamma densities.

Most classic Monte Carlo books spend a significant amount of time on a discussion of various methods for uniform random variate generation and for generating from standard univariate distributions using uniform random variate generation. We will bypass the discussion of methods for generating random variates from uniform and other standard distributions. The philosophy here is as follows: If excellent, reliable implementations are available in modern statistical computing

languages such as `R` (Ihaka and Gentleman, 1996), it is not worth studying the associated algorithms. The reader is instead refered to the books by Devroye (Devroye, 1986), now available online at: `http://cgm.cs.mcgill.ca/~luc/rnbookindex.html`) and Ripley (Ripley (1987)) for more on uniform random variate generation. In addition, both books have several algorithms for generating draws from basic statistical distributions such as the Normal, Poisson and Gamma. It is generally a bad idea to attempt to write your own random variate generator (say based on an algorithm from a book) when a well established and efficient one is already in existence. For example, when writing code in `C`, it is best to use well known random number generators such as those in the `R` libraries.

5.3.3 Unknown normalizing constants

In general, drawing a sample from an arbitrary (non-standard) distribution can be difficult, especially if the distribution is high-dimensional. An additional complication is the fact that in many important problems, we may often not even know the normalizing constant. In other words, we only know $h(x)$ where $h(x)/c = f(x)$ (we know $f(x)$, the distribution of interest, 'up to a constant of proportionality'). The unknown normalizing constant c=$\int h(x)dx$ if $h(x)$ is a pdf, and $\sum h(x)$ if $h(x)$ is a pmf, so it is clear that finding the normalizing constant may involve a complicated summation or integration and can be a difficult problem in its own right. Hence, algorithms that can draw samples from f by only using h are useful.

To see how distributions with intractable normalizing constants arise naturally, consider a binary Markov random field (or the Ising Model): This model or several versions of this model have been used in fields as diverse as image analysis and sociology. The model comes from statistical physics and a version of it can be described as follows: Consider an $N \times N$ lattice of pixels where each pixel has value $+1$ or -1. If 1 corresponds to black and -1 corresponds to white, the space of all black and white images with N^2 pixels would be $\Omega = \{(x_1, x_2, \ldots, x_{N^2},$ such that $x_m \in \{-1, 1\}\}$. We can describe a probability distribution on this space as follows $P(X = x) = \frac{1}{Z}\exp(-2JU(x))$ where $J > 0$ is fixed and $U(x) = \sum_{i \sim j} I(x_i \neq x_j)$ with $i \sim j$ when the ith and jth pixels are adjacent to one another (the summation is over all such pairs). Z is the unknown normalizing constant. This model favors smooth images, especially for large J, since it places higher probability on images where neighbors are alike. Clearly, $Z = \sum_{x \in \Omega} \exp(-2JU(x))$. The normalizing constant is intractable.

Normalizing constants are rarely available or easy to compute in Bayesian inference. In the Bayesian setup, suppose the likelihood of the data Y given parameters Θ is $L(Y|\Theta)$, and the prior distribution on the parameters is $\pi(\Theta)$, resulting in a posterior distribution $\pi(\Theta|Y) = \frac{L(Y|\Theta)\pi(\Theta)}{\int_\Theta L(Y|\Theta)\pi(\Theta)d\Theta}$. The denominator of this pdf is a normalizing constant which can be a high dimensional and intractable integral. In general, it is very common to see useful distributions where normalizing con-

stants are intractable both in frequentist and Bayesian settings. Most useful simulation algorithms therefore do not require that the normalizing constants be known.

5.4 Rejection Sampling

Rejection or Accept-Reject Sampling is a simple but very powerful method for generating random variates from a distribution. Suppose we know the desired distribution f up to a normalizing constant so we know $h(x) = f(x)/c$. Now if we have a distribution q so that we know how to sample from q and we know K such that

$$\sup_x \frac{h(x)}{q(x)} \leq K < \infty. \tag{5.3}$$

The accept-reject sampling algorithm is as follows:

1. Draw $X \sim q$ and draw $U \sim$Uniform(0,1).
2. If $U \leq \frac{h(X)}{Kq(X)}$ return X else do not return X.

Theorem 2. *The samples returned by the rejection sampler have distribution f (X is a draw from f).*

Proof. The distribution of any random variable X returned by this algorithm can be derived as follows:

$$\begin{aligned}
P(X \leq x | X \text{accepted}) =& P\left(X \leq x | U \leq \frac{h(X)}{Kq(X)}\right) \\
=& \frac{E_q\left(P(X \leq x, U \leq \frac{h(X)}{Kq(X)} | X\right)}{E_q\left(P(U \leq \frac{h(X)}{Kq(X)} | X\right)} \\
=& \frac{E_q\left(I(X \leq x)\frac{h(X)}{Kq(X)}\right)}{E_q\left(\frac{h(X)}{Kq(X)}\right)} \\
=& \frac{E_q\left(I(X \leq x)\frac{cf(X)}{Kq(X)}\right)}{E_q\left(\frac{cf(X)}{Kq(X)}\right)} \\
=& \frac{E_q\left(I(X \leq x)\frac{f(X)}{q(X)}\right)}{E_q\left(\frac{f(X)}{q(X)}\right)} \\
=& \frac{\int_{-\infty}^{x} f(x)dx}{1} = F(x).
\end{aligned}$$

Hence, $X \sim f(x)$. □

Although this proof assumes that f is a univariate density for simplicity, this result applies to multivariate and discrete settings. Also, it is easy to see that the proof would work even when q is unnormalized so the proposal is complicated enough that we either do not know the normalizing constant or we want to increase computational efficiency by avoiding the evaluation of the constant.

To summarize the above, simulating draws from f by rejection sampling requires:
(1) A proposal distribution q such that $\sup_x h(x)/q(x) < \infty$. That is, q has 'heavier tails' than h. q is sometimes referred to as an **envelope density**.
(2) Value of k such that $\sup_x h(x)/q(x) \leq K$. It is not necessary for K to be the smallest values that satisfies this condition. However, the smaller the value of K, the more efficient the algorithm. This is because the **acceptance probability of the rejection sampler** can be shown to be c/K where c is the normalizing constant for $h(x)$ (this also shows that we can only estimate the acceptance rate of a rejection sampler if we do not know the normalizing constant).

While rejection sampling is a very powerful algorithm in principle, the envelope condition (5.3) is very restrictive since finding an appropriate q is difficult and obtaining K may be even more difficult (although Caffo et al. (2002) describe a way to estimate K empirically and still use rejection sampling).

Another issue is that the rejection sampler does not scale well since it requires an envelope q for the entire joint distribution f. As the dimensions of the distribution gets large, K will get large, resulting in very low acceptance rates. To see this, we consider a simple illustrative example from (Jordan, 1999, pp 184–187). Suppose we want to sample from an M-dimensional Gaussian with mean zero, say $X \sim N_M(0, \sigma_f^2 I) = f(x)$ and we use a (very good) proposal $q = N_M(0, \sigma_q^2)$ with $\sigma_q = 1.01\sigma_f$ so the proposal is heavy-tailed with respect to the target density. The smallest bounding constant K satisfying (5.3) is obtained by maximizing f/q, which happens at the origin, resulting in $K = \frac{(2\pi\sigma_f^2)^{-M/2}}{(2\pi\sigma_q^2)^{-M/2}} = (\frac{\sigma_q}{\sigma_f})^M = \exp(M\log(\sigma_q/\sigma_f))$. If $M = 1000$, $K = \exp(10) \approx 20,000$ with acceptance rate=1/20,000. In general, K grows exponentially with M. Thus, even in this simple scenario, a well tuned proposal does not prevent the algorithm from becoming very inefficient with increase in dimensions.

5.5 Ratio of Uniforms Sampling

The ratio of uniforms algorithm is an alternative method for producing independent samples for a given density. An advantage of the ratio of uniforms over the rejection sampler is that there is no need to find a heavy-tailed proposal distribution. On the other hand, constructing the ratio of uniforms algorithm can be challenging due to the complicated regions from which it is necessary to sample uniformly; this problem is worse for higher dimensional distributions.

The univariate ratio of uniforms algorithm for univarite densities (Kinderman and Monahan,

1977) may be summarized as follows. Assume f is a possibly unnormalized density.

Theorem 3. *Suppose U, V are uniform on the region $\mathcal{A}_f = \{(u, v) : 0 < v \leq \sqrt{f(u/v)}\}$, then $Y = U/V$ has density proportional to f.*

The above result follows from a simple transformation of variables argument. An interesting facet of the ratio of uniforms algorithm is that it is involves sampling from a higher dimensional distribution, in this case a two-dimensional distribution for obtaining samples from a univariate density. Increasing the dimensions of the distribution can actually be beneficial in some situations. These are sometimes called "auxiliary variables" approaches.

- If $f(x) = f_0(x)f_1(x)$ and U, V has density proportional to $f_0(u, v)$ on $\mathcal{A}_f = \{(u, v) : 0 < v \leq \sqrt{f_1(u/v)}\}$ then $X = U/V$ has density proportional to f.
- If (U, V) are uniform on $\mathcal{A}_f = \{(u, v) : 0 < v \leq \sqrt{f(u/v + \mu)}\}$ then $X = U/V + \mu$ has density proportional to f. Here selecting μ to be the mode of f results in a more efficient algorithm.
- Higher dimensions (Wakefield, Gelfand and Smith, 1991; Stefanescu and Vaduva, 1987): If U, V are uniform on $\mathcal{A}_f = \{(u, v) : u \in R^d, 0 < v \leq \sqrt[d+1]{f(u/v + \mu)}\}$ then $X = U/V + \mu$ has density proportional to f.

The ratio of uniforms method is limited by the complexity of the region on which uniform sampling needs to be carried out, particularly in the multivariate generalization above. However, this issue can be addressed in some cases by using an approach like the Metropolis-Hastings algorithm (described in the next chapter).

5.6 Importance Sampling

Importance sampling, unlike rejection sampling, is not a method for generating samples from the target distribution f. Instead, it is a technique for estimating expectations with respect to f by reweighting samples from a different distribution, q, called an **importance function**. We therefore do *not* consider importance sampling to be an i.i.d. Monte Carlo method, even when it uses i.i.d. draws from q.

The basic idea is as follows. As before, suppose we want to estimate $\mu = E_f(g(x))$, an expectation of some function g with respect to the probability distribution $f(x)$. Let $q(x)$ be a distribution from which we know how to draw i.i.d. samples. In addition, we require that $q(x) > 0$ whenever $g(x)f(x) > 0$. We use the fact that $E_f(g(x)) = E_q(g(x)\frac{f(x)}{q(x)})$. This is easy to see since:

$$E_f(g(x)) = \int f(x)g(x)dx = \int \frac{g(x)f(x)}{q(x)}q(x)dx = E_q\left(g(x)\frac{f(x)}{q(x)}\right)$$

when f and q are pdfs (and similarly when they are pmfs), assuming in both cases that wherever $q(x) = 0$, $f(x)g(x) = 0$.The importance sampling estimator based on an i.i.d. sample $X_1, \ldots, X_n \sim q$ is therefore:

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{N}\frac{g(X_i)f(X_i)}{q(X_i)}. \tag{5.4}$$

By the strong law of large numbers, $\hat{\mu}_n \to \mu$ with probability 1. The requirement that the samples be i.i.d. is rather restrictive and unnecessary; this method works in much more general situations, such as when the samples are obtained from a Markov chain sampler (see Chapter 6). However, for now we will simply assume that the draws from q are i.i.d.

The simple estimator above assumes that normalizing constants are known, which makes it much less useful. Suppose we require that $q(x) > 0$ whenever $f(x) > 0$, a stronger condition than above. We then have $E_q(\frac{f(x)}{q(x)}) = 1$ and the corresponding estimator, $\sum_{i=1}^{N}\frac{f(X_i)}{q(X_i)}$ converges to 1 by the strong law of large numbers as before. We can thus define a more useful importance sampling estimator, the **ratio importance sampling estimator**:

$$\tilde{\mu}_n = \frac{\sum_{i=1}^{N} g(X_i)\frac{f(X_i)}{q(X_i)}}{\sum_{i=1}^{N}\frac{f(X_i)}{q(X_i)}} = \frac{\sum_{i=1}^{N} g(X_i)\frac{h(X_i)}{q(X_i)}}{\sum_{i=1}^{N}\frac{h(X_i)}{q(X_i)}} \tag{5.5}$$

where $h(x)/c = f(x)$ for some unknown normalizing constant c. It can be shown that this estimator converges to μ by using a version of Slutsky's theorem since the numerator converges to μ and the denominator converges to 1. This estimator is biased while (5.4) is unbiased; however, this bias tends to be small and the estimator actually has lower mean squared error (MSE) than (5.4) (cf. Liu (2001).) Note that **even the normalizing constants for the importance function q can be ignored.** This can be useful when reweighting samples obtained from a distribution for which the normalizing constant is missing, to estimate expectations with respect to another distribution. For example: when estimating posterior expectations under two prior distributions, one can avoid sampling twice by simply reweighting the samples from one posterior distribution (a result of using prior 1, say) to obtain expectations with respect to another posterior distribution (using prior 2.)

Alternatively, the estimator $\tilde{\mu}_n$ can also be derived as follows, again assuming $q(x) > 0$ whenever $f(x) > 0$.

$$\mu = E_q\left(g(x)\frac{f(x)}{q(x)}\right) = cE_q\left(g(x)\frac{h(x)}{q(x)}\right) = \frac{E_q\left(g(x)\frac{h(x)}{q(x)}\right)}{E_q\left(\frac{h(x)}{q(x)}\right)}, \tag{5.6}$$

since $c^{-1} = E_q\left(\frac{h(x)}{q(x)}\right)$. A natural estimator for μ is then $\tilde{\mu}_n$ (5.5). A potentially useful generalization can be obtained by using the fact that the numerator and denominator need not use the same

importance function. That is,

$$\mu = \frac{E_{q_1}\left(g(x)\frac{h(x)}{q_1(x)}\right)}{E_{q_2}\left(\frac{h(x)}{q_2(x)}\right)} \tag{5.7}$$

as long as $q_1(x) > 0$ whenever $g(x)f(x) > 0$ and $q_2(x) > 0$ whenever $f(x) > 0$. Then, different sets of samples, $X_1, \dots, X_n \overset{iid}{\sim} q_1$ and $Y_1, \dots, Y_m \overset{iid}{\sim} q_2$, would be drawn for estimating the numerator and denominator respectively. This flexibility can be useful in producing greater efficiency in situations where $g(x)h(x) > 0$ on a very different region than where $h(x) > 0$. This is common when estimating tail probabilities, as described in the next subsection.

5.6.1 Uses of importance sampling

Importance sampling is a very general, very powerful method for calculating expectations of a function with respect to some distribution (f, using our notation so far.) However, it is particularly useful in a few special situations which we describe in this section.

Importance sampling was originally conceived as a method for reducing the variance of the estimates (Monte Carlo standard error). Consider the importance sampling estimator (from 5.4). If we are able to select q such that $g(X_1)\frac{f(X_1)}{q(X_1)}$ has very low variability, $\hat{\mu}_n$ can have lower variance than that of an i.i.d. Monte Carlo estimator. As an extreme case, if $g(X_1)\frac{f(X_1)}{q(X_1)}$ is constant, the simple importance estimator has variance:

$$\text{Var}(\hat{\mu}_n) = \frac{1}{n}\text{Var}\left(g(X_1)\frac{f(X_1)}{q(X_1)}\right) = 0.$$

Of course, this represents a situation where we know so much about the problem that we probably did not need to use importance sampling in the first place. Even if we are able to reduce the variability in this manner, there is not enough reason to do importance sampling as long as an i.i.d. sampler is reasonably efficient. Gain in efficiency should always take into account time taken to set up and write the code for the algorithm and the time the new algorithm actually takes to produce the estimates. You will see more about comparing Monte Carlo methods later.

There are many important problems where importance sampling can be very useful. Here we describe two categories of problems. Note that importance sampling can be useful in these problems even when methods for simulating from the target f are available:

(1) **Rare event problems**: Consider the following toy example (from Robert and Casella (2005)): Suppose we want to estimate $P(Z > 4.5)$ where $Z \sim N(0,1)$. Naive i.i.d. Monte Carlo estimation would involve draws $Z_1, \dots, Z_n \sim N(0,1)$ and the estimate $\hat{\mu}_n = \sum_{i=1}^n I(Z_i > 4.5)/n$. For $n = 100,000$, this would typically produce an estimate of 0 since the probability of observing $Z > 4.5$ is small. However, we can get much better estimates by importance sampling with

the importance function $q =$TrExp$(4.5, 1)$, where TrExp$(4.5, 1)$ is a Truncated Exponential density where an exponential density with scale 1 is shifted by 4.5 units to the right of the origin on the x-axis. If we simulate independent draws $Y_1, \dots, Y_n \sim$TrExp$(4.5, 1)$ and use the simple importance estimate $\hat{\mu}_n$, we can get a much more accurate estimate. The true value (using R's pnorm function) is 3.3976×10^{-6}. With $n = 10,000$, I obtained an estimate of 3.35×10^{-6} and with $n = 100,000$ my estimate was 3.38×10^{-6}.

(2) **Expectations with respect to multiple distributions**: A context in which importance sampling can be very efficient is a situation where its variance may actually be higher than for an i.i.d. sampler. Our concept of efficiency includes both the variability (and hence accuracy) of the estimates while still accounting for simulation effort (computer time used to produce the samples.) The efficiency here arises from the fact that it is possible to use a single set of samples to estimate expectations with respect to several distributions.

Consider a problem where we are interested in calculating expectations with respect to several distributions, say a parametric family $\{f_\theta : \theta \in \Theta\}$. Obviously, we will not be able to evaluate the desired expectations with respect to all possible f_θ but we would at least like to be able to estimate the expectation for many different values of θ. It is easy to imagine situations where one is interested in estimating such expectations, for instance to study how the expected value changes as a function of θ or to maximize an expectation with respect to the parameter θ; we will later see more examples in the context of Monte Carlo Maximum Likelihood.

The usual i.i.d. Monte Carlo method would involve simulating independent $X_1^{(i)}, \dots, X_n^{(i)} \sim f_{\theta_i}$ for each θ_i at which we want an estimate. This is clearly highly inefficient as the number of samples required would quickly become prohibitively large. On the other hand, if we simulate a single random sample $Y_1, \dots, Y_n \sim q$, for an appropriately chosen q, would could easily use the estimator $\tilde{\mu}_n$ (5.5) to estimate the expectations for as many θ values as we like. Of course, we would require that for each θ, $q(x) > 0$ whenever $f_\theta(x) > 0$ for all x. Although this would always work in principle, finding a q that produces good estimates for all θ is not always an easy problem.

Estimating densities from importance sampling: Importance sampling is a method for estimating expectations, but what if we wanted to estimate the entire distribution of interest rather than just some expectations with respect to it? Importance sampling is not the most natural method for doing this, but it is certainly feasible to obtain estimates for this. For instance, one could estimate the cdf for each value since the cdf can be written as an expectation of an indicator function. For a density,

$$F(x) = \int I(X < x)f(x)dx = E_f(I(X < x)),$$

which can be estimated via importance sampling with respect to an importance function q as before. Hence, one can estimate the cdf at many values of x to indirectly estimate the density of interest.

5.6.2 Monte Carlo standard errors

Monte Carlo standard errors for the simple importance sampling estimator (5.4) are very easily obtained.

$$\text{Var}(\hat{\mu}_n) = \text{Var}\left(\frac{1}{n}\sum_{i=1}^{N}\frac{g(X_i)f(X_i)}{q(X_i)}\right) = \frac{1}{n}\text{Var}_q\left(\frac{g(X)f(X)}{q(X)}\right), \tag{5.8}$$

since $X_1, \ldots, X_n$ are iid. It is easy to estimate $\text{Var}_q g(X)\frac{f(X)}{q(X)}$ from the samples generated. We would simply compute the sample variance, $\hat{\sigma}^2$ based on $g(X_1)\frac{f(X_1)}{q(X_1)}, \ldots, g(X_n)\frac{f(X_n)}{q(X_n)}$. The Monte Carlo standard error estimate would then be $\frac{\hat{\sigma}}{\sqrt{n}}$.

However, estimating sample variances for the ratio importance sampling estimator (5.5) is more complicated. We describe the approach, starting with the central limit theorem for this estimator.

Theorem 4. *Define $w(x) = f(x)/q(x)$. If $E_q(w(x)^2) < \infty$ and $E_q g^2(x)w^2(x) < \infty$, then,*

- *A C.L.T. (bivariate version) holds with*

$$\sqrt{n}\begin{bmatrix}\frac{1}{n}\sum_{i=1}^{n} g(x_i)w(x_i) - \mu \\ \frac{1}{n}\sum_{i=1}^{n} w(x_i) - 1\end{bmatrix} \xrightarrow{L} N(0, \Sigma) \tag{5.9}$$

  where Σ is a 2×2 variance-covariance matrix.

- *The method of moments estimate of the asymptotic variance (σ^2) is consistent.*

Proof. See (Sen and Singer, 1993), Theorem 3.3.9 for the proof of the bivariate Central Limit Theorem. □

To derive the variance of the ratio estimator ($\tilde{\mu}_n$), we appeal to the delta method, where if $Q(x, y)$ is a real valued function, then

$$\sqrt{n}\left(Q\left(\frac{1}{n}\sum_{i=1}^{n} g(x_i)w(x_i), \frac{1}{n}\sum_{i=1}^{n} w(x_i)\right) - Q(\mu, 1)\right) \xrightarrow{L} N(0, Q'(\mu,1)^T \Sigma Q'(\mu,1))$$

where $Q'(\mu, 1)$ is a 2×1 matrix. If we set $Q(a, b) = a/b$ (where a and b are the numerator and denominator of the ratio estimator respectively)

$$\begin{aligned} Q'(a,b) =& (1/b, -a/b^2)^T, \text{ so } Q'(\mu,1) = (1, -\mu)^T \text{ and} \\ \Sigma =& \begin{bmatrix}\text{Var}(a) & \text{Cov}(a,b) \\ \text{Cov}(a,b) & \text{Var}(b)\end{bmatrix} \end{aligned}$$

Let the asymptotic variance be denoted by σ^2, so $\sigma^2 = Q'(\mu,1)^T \Sigma Q'(\mu,1)$.

$$\begin{aligned} \sigma^2 =& \text{Var}\left(\frac{1}{n}\sum_{i=1}^{n} g(x_i)w(x_i)\right) - 2\mu\text{Cov}\left(\frac{1}{n}\sum_{i=1}^{n} g(x_i)w(x_i), \frac{1}{n}\sum_{i=1}^{n} w(x_i)\right) \\ &+\mu^2\text{Var}\left(\frac{1}{n}\sum_{i=1}^{n} w(x_i)\right) \\ =& \frac{1}{n}\left\{\text{Var}_q(g(x)w(x)) - 2\mu\text{Cov}(g(x)w(x), w(x)) + \mu^2\text{Var}(w(x))\right\} \end{aligned} \tag{5.10}$$

We can estimate σ^2 by using a method of moments estimator, that is, simply replacing each of the variances and covariances above with sample variances and covariances. However, we need to be very careful here since we are assuming that normalizing constants are known. In the general case where we only have $h(x)$ where $f(x) = h(x)/c$, we would like our estimator to use the weights $\tilde{w}(x) = \frac{h(x)}{q(x)} = \frac{cf(x)}{q(x)} = cw(x)$. Since this estimator does not rely on knowing the normalizing constant, neither should the variance estimator. After simplification, the variance from (5.10), can be estimated by:

$$\begin{aligned} \hat{\sigma}^2 = n\left(\frac{\sum_{i=1}^{n} g(x_i)\tilde{w}(x_i)}{\sum_{i=1}^{n}\tilde{w}(x_i)}\right)^2 \Bigg(&\frac{\sum_{i=1}^{n} g(x_i)^2\tilde{w}(x_i)^2}{(\sum_{i=1}^{n} g(x_i)\tilde{w}(x_i))^2} + \frac{\sum_{i=1}^{n}\tilde{w}(x_i)^2}{(\sum_{i=1}^{n}\tilde{w}(x_i))^2} \\ &-2\frac{\sum_{i=1}^{n} g(x_i)\tilde{w}(x_i)^2}{(\sum_{i=1}^{n} g(x_i)\tilde{w}(x_i))(\sum_{i=1}^{n}\tilde{w}(x_i))}\Bigg) \end{aligned}$$

For an estimate of Monte Carlo standard error, we would, as usual, use $\hat{\sigma}/\sqrt{n}$. Note that we do not have to retain all the samples generated to estimate the standard error; we would simply keep track of the following terms:

$$\sum_{i=1}^{n} g(x_i)\tilde{w}(x_i), \sum_{i=1}^{n}\tilde{w}(x_i), \sum_{i=1}^{n} g(x_i)^2\tilde{w}(x_i)^2, \sum_{i=1}^{n} g(x_i)\tilde{w}(x_i))^2, \text{ and } \sum_{i=1}^{n}\tilde{w}(x_i)^2.$$

The above variance estimate is consistent under the assumptions of Theorem 4.

5.6.3 Comparison to rejection sampling

While importance sampling is an approach that involves reweighting existing samples (from the importance function q) to estimate expectations of interest, and rejection sampling actually produces draws from the target f, the two approaches can be put into a common theoretical framework (see, for instance, Chen (2005)). It is instructive to compare importance sampling to rejection sampling. Here are some major points of comparison:

1. Importance sampling, when viewed as a generalization of rejection sampling (importance sampling with weights set to 0 whenever a sample is rejected by the rejection sampler), can be shown to be always at least as efficient in terms of reduced asymptotic variance (Chen, 2005).

2. Importance sampling does not require that the rejection sampling condition (5.3) be satisfied. In other words, it is not necessary for the importance function q to be heavier tailed than the target density f. However, this is very desirable. If (5.3) is satisfied, and $E_q(g(x)^2) < \infty$, then the Central Limit theorm holds and the method of moments estimate of the asymptotic variance is consistent. Hence, this provides a relatively simple way of ensuring that Theorem 4 holds. Also, when (5.3) is not satisfied, it is likely that the standard error of the importance sampling estimate is infinite.

3. It is not necessary to find the bounding constant K in (5.3). This is a major advantage over rejection sampling.

4. In importance sampling, it may not always be desirable to match the importance function q to f (unlike rejection sampling). As explained earlier, it may be more important to find q that satisfies other criteria, such as matching $g(x)f(x)$ well for tail probability calculations.

5. Variance estimates for importance sampling estimators are less reliable than for rejection sampling based estimates.

6. Importance sampling estimators may have numerical stability issues. Specifically, we need to exponentiate $\log h - \log q$ to calculate weights h/q while this is unnecessary in a rejection sampler, where everything can be done on log-scale.

7. It is more natural to estimate densities using samples from f (as produced by the rejection sampler) than via an importance sampling estimate. However, it is possible to use importance sampling to estimate the cdf $F(x)$ at many points x by noting that

$$F(x) = \int 1(X \leq x) f(x) dx = E_f(1(X \leq x)).$$

   And, of course, once it is written as an expectation, it is easy to use importance sampling to estimate it. In principle, it is then possible to use the estimate of the cdf to derive the corresponding density estimate.