Monte Carlo Simulation Notes

Lecturer: Shane Henderson Cornell ORIE
Scribe: Kevin Kircher, Cornell MAE

These notes cover a subset of the material from ORIE 6580, Simulation, as taught by Prof. Shane Henderson at Cornell University in the spring of 2016. They cover the basics of Monte Carlo simulation, *i.e.*, of analyzing stochastic systems by generating samples of the underlying random variables. Much course material, including some entire topics, has been omitted. Knowledge of calculus-based probability, and of stochastic processes at the level of *Stochastic Processes* by Sheldon Ross [1], is assumed.

These notes also draw material from *Simulation* by Sheldon Ross [2], *Handbooks in Operations Research and Management Science: Simulation* edited by Shane Henderson and Barry Nelson [3], and the wonderful lecture notes of Prof. Martin Haugh at Columbia University.

Any errors in these notes are the fault of the scribe, not the lecturer. If you find any, feel free to let Kevin know.

Contents

Chapter 1

Introduction

1.1 Notation

1.1.1 Vectors and scalars

Let $\mathbb{R}$ be the set of real numbers and $\mathbb{N} = \{1, 2, 3, \ldots\}$ be the set of natural numbers. For any $n \in \mathbb{N}$, we denote the set of real n-dimensional column vectors by $\mathbb{R}^n$. We indicate vectors with bold font. For vectors $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$, we write $(\mathbf{x}, \mathbf{y})$ to indicate the column vector in $\mathbb{R}^{n+m}$ obtained by stacking $\mathbf{x}$ above $\mathbf{y}$:

$$(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}.$$

This notation extends to any finite list of elements. In particular, we represent a vector $\mathbf{x} \in \mathbb{R}^n$ with components $x_1, \ldots, x_n \in \mathbb{R}$ with any of the following, interchangeably:

$$\mathbf{x} = (x_1, \ldots, x_n) = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 & \ldots & x_n \end{bmatrix}^\top.$$

1.1.2 Random vectors and variables

We indicate random variables and random vectors with capital letters. With some abuse of notation[1], we write $\mathbf{X} \in \mathbb{R}^n$ to mean that $\mathbf{X}$ is a real n-dimensional random vector with components $X_1, \ldots, X_n$. We indicate particular realizations of $\mathbf{X}$ by $\mathbf{x}$, and of X_i by x_i. Nonbold capital letters may represent sets, matrices, or scalar random variables. When not clear from context, the precise meaning will be made clear in the text.

In Monte Carlo simulation, we often consider samples from the distribution of a random vector $\mathbf{X}$. We denote such samples by $\mathbf{X}_1, \mathbf{X}_2, \ldots$. The notation $\mathbf{X}_i$, which indicates the i^{th} sample (a random vector), should not be confused with X_i, which indicates the i^{th} component of $\mathbf{X}$. When necessary, we indicate the j^{th} component of $\mathbf{X}_i \in \mathbb{R}^n$ by $(\mathbf{X}_i)_j$.

[1] To be precise, an n-dimensional random vector is a function from a sample space Ω to $\mathbb{R}^n$.

1.1.3 Functions

We use nonstandard notation for functions. Traditionally, a function f that maps real n-dimensional vectors into real scalars is written $f : D \to \mathbb{R}$, where $D \subseteq \mathbb{R}^n$ is the domain of f. Instead of this notation, we write $f : \mathbb{R}^n \to \mathbb{R}$ to declare the input-output syntax of f, with the understanding that f may not be defined on all of $\mathbb{R}^n$. When necessary, we indicate the domain of f by $\operatorname{dom} f$. We indicate vector-valued functions with bold font, *e.g.*, $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$.

When discussing algorithms' performance, we occasionally use 'big O' notation. For functions $f, g : \mathbb{R} \to \mathbb{R}$, we say that f is big O of g as $x \to \infty$, denoted $f(x) = O(g(x))$, if there exist some constants $M > 0$ and $x_0 \in \mathbb{R}$ such that

$$|f(x)| \leq M\,|g(x)| \text{ for all } x \geq x_0.$$

Put another way (and assuming $g(x) \neq 0$ for all sufficiently large x), $f(x) = O(g(x))$ as $x \to \infty$ if the ratio $|f(x)/g(x)|$ remains bounded as $x \to \infty$. Similarly, we say that $f(x) = O(g(x))$ as $x \to a \in \mathbb{R}$ if there exists some constants $M, \delta > 0$ such that

$$|x - a| < \delta \quad \Longrightarrow \quad |f(x)| \leq M\,|g(x)|\,.$$

1.1.4 Probability, expectation, variance, and covariance

We denote the probability of an event A by $\mathbb{P}\,A$. For example, the probability that a random variable X exceeds a threshold x is written $\mathbb{P}\,\{X > x\}$. We denote expectation by $\mathbb{E}$. For a random variable $X \in \mathbb{R}$, the variance of X is

$$\operatorname{Var} X := \mathbb{E}[X - \mathbb{E}\,X]^2 = \mathbb{E}\,X^2 - \mathbb{E}[X]^2.$$

Here the symbol $:=$ means 'is defined as.' The covariance between random variables X and Y is

$$\operatorname{Cov}(X, Y) := \mathbb{E}[(X - \mathbb{E}\,X)(Y - \mathbb{E}\,Y)],$$

so $\operatorname{Cov}(Y, X) = \operatorname{Cov}(X, Y)$ and $\operatorname{Cov}(X, X) = \operatorname{Var} X$. The correlation between X and Y is

$$\rho := \frac{\operatorname{Cov}(X, Y)}{\sqrt{\operatorname{Var} X \operatorname{Var} Y}}.$$

For a random vector $\mathbf{X} = (X_1, \ldots, X_n) \in \mathbb{R}^n$, we define the $n \times n$ covariance matrix

$$\begin{aligned} \operatorname{Cov} \mathbf{X} &:= \mathbb{E}\left[(\mathbf{X} - \mathbb{E}\,\mathbf{X})(\mathbf{X} - \mathbb{E}\,\mathbf{X})^\top\right] \\ &= \begin{bmatrix} \operatorname{Var} X_1 & \operatorname{Cov}(X_1, X_2) & \ldots & \operatorname{Cov}(X_1, X_n) \\ \operatorname{Cov}(X_2, X_1) & \operatorname{Var} X_2 & \ldots & \operatorname{Cov}(X_2, X_n) \\ \vdots & & \ddots & \vdots \\ \operatorname{Cov}(X_n, X_1) & \operatorname{Cov}(X_n, X_2) & \ldots & \operatorname{Var} X_n \end{bmatrix}. \end{aligned}$$

Since $\operatorname{Cov}(X_j, X_i) = \operatorname{Cov}(X_i, X_j)$, $\operatorname{Cov} \mathbf{X}$ is symmetric for any $\mathbf{X}$. The covariance between random vectors $\mathbf{X} \in \mathbb{R}^n$ and $\mathbf{Y} \in \mathbb{R}^m$ is the $n \times m$ matrix

$$\begin{aligned}\operatorname{Cov}(\mathbf{X}, \mathbf{Y}) &:= \mathbb{E}[(\mathbf{X} - \mathbb{E}\,\mathbf{X})(\mathbf{Y} - \mathbb{E}\,\mathbf{Y})^\top] \\ &= \begin{bmatrix} \operatorname{Cov}(X_1, Y_1) & \dots & \operatorname{Cov}(X_1, Y_m) \\ \vdots & \ddots & \vdots \\ \operatorname{Cov}(X_n, Y_1) & \dots & \operatorname{Cov}(X_n, Y_m) \end{bmatrix},\end{aligned}$$

so $\operatorname{Cov}(\mathbf{Y}, \mathbf{X}) = \operatorname{Cov}(\mathbf{X}, \mathbf{Y})^\top$ and $\operatorname{Cov}(\mathbf{X}, \mathbf{X}) = \operatorname{Cov} \mathbf{X}$.

1.2 The basic problem

We consider the problem of estimating a parameter

$$\theta := \mathbb{E}\, h(\mathbf{X}).$$

Here $\mathbf{X}$ is a random vector in $\mathbb{R}^m$ and $h : \mathbb{R}^m \to \mathbb{R}$. The random variables $X_1, \dots, X_m$ may be continuous, discrete, or mixed. They can represent static quantities, or they may be the values of some stochastic processes over time. The dimension m could be very large. The function h may be nonlinear, nonsmooth, or not even analytically representable. For example, h could be the indicator function $\mathcal{I}_A : \mathbb{R}^m \to \mathbb{R}$ of some bizarre set $A \subset \mathbb{R}^m$, in which case $\theta = \mathbb{E}\, \mathcal{I}_A(\mathbf{X}) = \mathbb{P}\{\mathbf{X} \in A\}$. We assume only that $\mathbb{E}\,|h(\mathbf{X})|$ and $\operatorname{Var} h(\mathbf{X})$ are finite.

In principle, the parameter θ could be computed analytically or by deterministic methods for numerical integration. If $\mathbf{X}$ is continuous with probability density function (pdf) f defined on all of $\mathbb{R}^m$, for example, then the expected value of $h(\mathbf{X})$ is the multiple integral

$$\mathbb{E}\, h(\mathbf{X}) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h(\mathbf{x}) f(\mathbf{x}) \mathrm{d}x_1 \cdots \mathrm{d}x_m.$$

In practice, however, this multiple integral is usually too complex to evaluate analytically. Furthermore, the computational costs of deterministic methods for numerical integration typically increase exponentially quickly with the dimension m of $\mathbf{X}$. By contrast, Monte Carlo methods for computing $\mathbb{E}\, h(\mathbf{X})$ converge at a rate that is *independent* of m. This makes Monte Carlo methods attractive tools for complex, high-dimensional systems.

1.3 The Monte Carlo framework

Rather than computing expectation integrals analytically or by deterministic numerical methods, Monte Carlo methods generate independent, identically distributed (iid) random samples $\mathbf{X}_1, \dots, \mathbf{X}_n$ from the distribution of $\mathbf{X}$ and estimate $\mathbb{E}\, h(\mathbf{X})$ by the corresponding sample average,

$$\bar{\theta}_n := \frac{1}{n} \sum_{i=1}^{n} h\left(\mathbf{X}_i\right).$$

1.3.1 Statistical properties of Monte Carlo estimators

Since $\mathbf{X}_1, \ldots, \mathbf{X}_n$ have the same distribution as $\mathbf{X}$, the sample average $\bar{\theta}_n$ is an unbiased estimator of θ:

$$\mathbb{E}\,\bar{\theta}_n = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\,h\left(\mathbf{X}_i\right) = \frac{n\,\mathbb{E}\,h(\mathbf{X})}{n} = \theta.$$

We assume that $\mathbb{E}\,|h(\mathbf{X})| < \infty$, so the strong Law of Large Numbers implies that $\bar{\theta}_n$ is also a consistent estimator of θ:

$$\bar{\theta}_n \to \theta \text{ almost surely as } n \to \infty.$$

We also assume that

$$\sigma^2 := \operatorname{Var} h(\mathbf{X}) < \infty,$$

so $h(\mathbf{X}_1), h(\mathbf{X}_2), \ldots$ is a sequence of iid random variables with finite mean θ and finite variance σ^2. Therefore, the Central Limit Theorem gives the asymptotic distribution of $\bar{\theta}_n$:

$$\frac{\sqrt{n}(\bar{\theta}_n - \theta)}{\sigma} \Rightarrow \mathcal{N}(0,1) \text{ as } n \to \infty. \tag{1.1}$$

Here $\Rightarrow$ denotes convergence in distribution and $\mathcal{N}(0,1)$ is a standard normal random variable.

1.3.2 Selecting a sample size

How big must our sample be before we can be confident that our Monte Carlo estimator is accurate? To make the question precise, let's define a confidence level[2] $\alpha \in (0,1)$ and an error tolerance $\varepsilon > 0$. With these definitions, a precise statement of our goal is: Find a value of n for which

$$\mathbb{P}\left\{\left|\bar{\theta}_n - \theta\right| \le \varepsilon\right\} = 1 - \alpha.$$

The Central Limit Theorem (1.1) gives us an approximate[3] answer. Because $\sqrt{n}(\bar{\theta}_n - \theta)/\sigma$ is asymptotically distributed as a standard normal random variable, for large n we have

$$\begin{aligned}
\mathbb{P}\left\{\left|\bar{\theta}_n - \theta\right| \le \varepsilon\right\} &= \mathbb{P}\left\{\left|\frac{\sqrt{n}(\bar{\theta}_n - \theta)}{\sigma}\right| \le \frac{\sqrt{n}\varepsilon}{\sigma}\right\} \\
&\approx \mathbb{P}\left\{|\mathcal{N}(0,1)| \le \frac{\sqrt{n}\varepsilon}{\sigma}\right\} \\
&= \mathbb{P}\left\{\mathcal{N}(0,1) \le \frac{\sqrt{n}\varepsilon}{\sigma}\right\} - \mathbb{P}\left\{\mathcal{N}(0,1) \le -\frac{\sqrt{n}\varepsilon}{\sigma}\right\} \\
&= 2\,\mathbb{P}\left\{\mathcal{N}(0,1) \le \frac{\sqrt{n}\varepsilon}{\sigma}\right\} - 1,
\end{aligned}$$

[2] Typical values of α are 0.05 and 0.01.

[3] It's approximate because the Central Limit Theorem only gives the *asymptotic* distribution of $\bar{\theta}_n$ in the limit of large n. How large must n be before $\bar{\theta}_n$ starts to 'look normal'? It depends on 'how normal' the distribution of $h(\mathbf{X})$ is, but the typical rule of thumb is that n should be at least 30.

where the last line follows by symmetry and normalization of the standard normal pdf.

To guarantee that $\mathbb{P}\left\{\left|\bar{\theta}_n - \theta\right| \leq \varepsilon\right\} \approx 1 - \alpha$, therefore, we should choose n such that

$$\begin{aligned} & 2\,\mathbb{P}\left\{\mathcal{N}(0,1) \leq \frac{\sqrt{n}\varepsilon}{\sigma}\right\} - 1 = 1 - \alpha, \\ \iff \quad & \Phi\left(\frac{\sqrt{n}\varepsilon}{\sigma}\right) = 1 - \alpha/2, \end{aligned}$$

where Φ is the standard normal cumulative distribution function (cdf). With the definition

$$z_{1-\alpha/2} := \Phi^{-1}(1 - \alpha/2),$$

the required sample size is

$$n = \left(\frac{\sigma z_{1-\alpha/2}}{\varepsilon}\right)^2.$$

In practice, we usually don't know the variance σ^2 of $h(\mathbf{X})$. It can be estimated, however, by the sample variance

$$\bar{\sigma}_n^2 := \frac{1}{n-1}\sum_{i=1}^{n}(h\left(\mathbf{X}_i\right) - \bar{\theta}_n)^2.$$

The sample variance is an unbiased and consistent estimator of σ^2. By the Converging Together Theorem, therefore, we can replace σ by $\bar{\sigma}_n$ in the Central Limit Theorem:

$$\frac{\sqrt{n}(\bar{\theta}_n - \theta)}{\bar{\sigma}_n} \Rightarrow \mathcal{N}(0,1) \text{ as } n \to \infty. \tag{1.2}$$

This gives a two-stage procedure for deciding how large a sample size to use:

1. Choose a pilot sample size n_0 (say, 50 or 100), generate $\mathbf{X}_1, \ldots, \mathbf{X}_{n_0}$ iid from the distribution of $\mathbf{X}$, and estimate σ by $\bar{\sigma}_{n_0}$.
2. Set $n = (\bar{\sigma}_{n_0} z_{1-\alpha/2}/\varepsilon)^2$.

1.3.3 Constructing confidence intervals

In the previous section, we asked,

> Given α and ε, what sample size $n \in \mathbb{N}$ is required to be $100(1-\alpha)\%$ confident that θ lies in the interval $[\bar{\theta}_n - \varepsilon, \bar{\theta}_n + \varepsilon]$?

A closely related question is,

> Given α and n, for what error tolerance $\varepsilon > 0$ can we be $100(1-\alpha)\%$ confident that θ lies in the interval $[\bar{\theta}_n - \varepsilon, \bar{\theta}_n + \varepsilon]$?

By the same reasoning applied in §1.3.2, the answer to the second question is

$$\varepsilon = \frac{\sigma z_{1-\alpha/2}}{\sqrt{n}}.$$

Because σ is usually unknown, we replace σ by $\bar{\sigma}_n$ when constructing confidence intervals.

1.3.4 The Monte Carlo algorithm

Algorithm 1 (Monte Carlo estimation of $\theta = \mathbb{E}\, h(\mathbf{X})$). Given confidence level $\alpha \in (0, 1)$ and sample size $n \in \mathbb{N}$:

1. generate $\mathbf{X}_1, \ldots, \mathbf{X}_n$ iid from the distribution of $\mathbf{X}$
2. estimate $\theta = \mathbb{E}\, h(\mathbf{X})$ and $\sigma^2 = \operatorname{Var} h(\mathbf{X})$ by

$$\bar{\theta}_n = \frac{1}{n}\sum_{i=1}^{n} h\left(\mathbf{X}_i\right), \quad \bar{\sigma}_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(h\left(\mathbf{X}_i\right) - \bar{\theta}_n\right)^2$$

3. conclude with approximately $100(1-\alpha)\%$ confidence that

$$\theta \in \left[\bar{\theta}_n - \frac{\bar{\sigma}_n z_{1-\alpha/2}}{\sqrt{n}},\ \bar{\theta}_n + \frac{\bar{\sigma}_n z_{1-\alpha/2}}{\sqrt{n}}\right]$$

1.3.5 Convergence rate

The confidence interval half-width $\varepsilon = \sigma z_{1-\alpha/2}/\sqrt{n}$ gives a measure of the Monte Carlo convergence rate. For fixed α, ε is directly proportional to the standard deviation σ of $h(\mathbf{X})$, meaning that Monte Carlo simulation works better on problems with less variability.

The confidence interval half-width also decreases as $n \to \infty$, but only at the rate of $O(1/\sqrt{n})$. Put another way, the sample size required to achieve a confidence interval half-width ε is $O(1/\varepsilon^2)$ as $\varepsilon \to 0$. This is *slow*: Suppose we have achieved a confidence interval half-width ε_1 using a sample size n_1, and we'd like to reduce the confidence interval half-width to $\varepsilon_2 = \varepsilon_1/10$. The sample size n_2 required to achieve this reduction satisfies

$$\frac{n_2}{n_1} \approx \frac{1/\varepsilon_2^2}{1/\varepsilon_1^2} = \left(\frac{\varepsilon_1}{\varepsilon_2}\right)^2 = 100.$$

In other words, for every order of magnitude decrease in estimation error, we must increase the sample size by two orders of magnitude. By contrast, most deterministic numerical methods, *e.g.*, for integration or optimization, are $O(1/\sqrt{\varepsilon})$ or faster. But as discussed in §1.2, the Monte Carlo convergence rate is the same *regardless of the dimension of* $\mathbf{X}$.

1.3.6 What's hard about Monte Carlo simulation?

The Monte Carlo algorithm 1 seems simple to implement, and it often is. This is one of the nice things about Monte Carlo simulation. However, there are two sticking points:

1. How can we generate the random samples $\mathbf{X}_1, \ldots, \mathbf{X}_n$ from the distribution of $\mathbf{X}$?

2. As discussed in §1.3.5, the Monte Carlo convergence rate of $O(1/\sqrt{n})$ is *slow*. How can it be improved?

We discuss the first question in Chapter 2, and the second question in Chapter 3. Chapter 4 covers methods for gradient estimation, also known as sensitivity analysis.

Chapter 2

Generating random variables

In order to implement the Monte Carlo algorithm 1, we need to be able to generate iid samples from the distribution of $\mathbf{X}$. Computers (at least today's non-quantum variety) are fundamentally deterministic machines. So how can we generate random numbers in software?

This question is the focus of this chapter. The answer: First, we use a deterministic algorithm to generate a stream of numbers between zero and one that 'look like' iid Uni$(0,1)$ random variables (see §2.1). Then we (again deterministically) transform these numbers into sequences that 'look like' iid realizations of $\mathbf{X}_1, \ldots, \mathbf{X}_n$ (see §2.2).

2.1 Uniform random variables

The goal of uniform random number generation is to produce a stream of numbers $u_1, u_2, \ldots$ that 'looks like' a sequence of iid Uni$(0,1)$ random variables. Most uniform random number generators are based on the deterministic linear recurrence

$$x_i = (a_1 x_{i-1} + \cdots + a_k x_{i-k}) \mod m. \tag{2.1}$$

Here $\mathbf{x}_i := (x_{i-k+1}, \ldots, x_i) \in S$ is the state of the random number generator at stage i and $S \subset \mathbb{R}^k$ is the state space. The initial state $\mathbf{x}_0$ is called the seed of the random number generator. The order $k \in \mathbb{N}$, modulus $m \in \mathbb{N}$, and coefficients $a_1, \ldots, a_k \in \mathbb{R}$ are chosen by the generator designer.

The 'modulo' operation $a \mod b$ is the remainder after dividing a by b. A basic fact of modular arithmetic is that $a \mod b \in [0, b)$ for any $a \in \mathbb{R}$. Therefore, $x_i \in [0, m)$ for all i, so we set $u_i = x_i/m \in [0, 1)$. Algorithms that use uniform random generators to generate nonuniform random numbers often require u_i to be strictly greater than zero, so it is customary to modify the mapping from x_i to u_i slightly so that $u_i \neq 0$.

Because computers represent real numbers only to finite precision, the state space S is necessarily a finite set. Therefore, there must exist integers $\ell \geq 0$ and $j \in (0, |S|]$ such that $\mathbf{x}_{\ell+j} = \mathbf{x}_\ell$. Since the recurrence (2.1) is deterministic, the stream of states is periodic: For all $i \geq \ell$, $\mathbf{x}_{i+j} = \mathbf{x}_i$ and $u_{i+j} = u_i$. The smallest j for which this holds is called the period of the random number generator. A sequence of truly iid Uni$(0,1)$ random variables would

not be periodic, so good random number generators have very long periods. It can be shown that when the modulus m is a prime number, the coefficients $a_1, \ldots, a_k$ can be chosen such that the period is $m^k - 1$, the longest possible. A typical choice of m is $2^{31} - 1$, the largest prime number representable on a 32-bit machine.

Most software packages have built-in random number generators, so we discuss this topic no further except to state some desirable properties. Ideally, a uniform random number generator should

1. have a very long period,
2. be fast and use little memory,
3. be able to generate the same sequence of numbers, repeatably (so that, for example, different simulation algorithms can be compared under the same input samples), and
4. be able to jump ahead efficiently (so that, for example, the random number stream can be broken into substreams that can be passed to processors running in parallel).

2.2 Nonuniform random variables

2.2.1 Inversion

Algorithm 2 (Generating X from cdf F by inversion). Given inverse cdf F^{-1} such that $F^{-1}(u) = \inf\{x \mid F(x) \geq u\}$:

1. generate $U \sim \text{Uni}(0, 1)$
2. set $X = F^{-1}(U)$

Why does inversion work?

For continuous X, the cdf of the returned variable, evaluated at any $x \in \mathbb{R}$, is

$$\mathbb{P}\left\{F^{-1}(U) \leq x\right\} = \mathbb{P}\left\{U \leq F(x)\right\} = F(x).$$

The first equality follows from the fact that F^{-1} is nondecreasing. The second follows from the facts that $F(x) \in [0, 1]$ for all $x \in \mathbb{R}$, and that $\mathbb{P}\{U \leq u\} = u$ for $U \sim \text{Uni}(0, 1)$ and $u \in [0, 1]$.

The proof also works for discrete or mixed X, provided we define

$$F^{-1}(u) = \inf\{x \mid F(x) \geq u\}.$$

(The infimum is attained since F_x is right-continuous.) With this definition, it is not hard to show that $F^{-1}(U) \leq x$ if and only if $U \leq F(x)$, so the above proof works.

Computing F^{-1}

If X is continuous, then we can (in principle, at least) compute F^{-1} by solving the equation $F(F^{-1}(x)) = x$ for x. For example, if $X \sim \text{Exp}(\lambda)$, then $F(x) = 1 - e^{-\lambda x}$, and

$$F(F^{-1}(x)) = x \quad \iff \quad 1 - e^{-\lambda F^{-1}(x)} = x \quad \iff \quad F^{-1}(x) = -\frac{1}{\lambda}\ln(1-x).$$

For some continuous distributions, such as the normal distribution, an analytical expression for F^{-1} cannot be found. In these cases, $F^{-1}(U)$ can be numerically approximated.

If X is discrete, then F^{-1} typically cannot be computed analytically. One exception is the geometric distribution: If $X \sim \text{Geo}(p)$, then for $u \in (0,1)$,

$$F^{-1}(u) = \left\lceil \frac{\ln(1-u)}{\ln(1-p)} \right\rceil.$$

2.2.2 Acceptance/rejection

Suppose X is continuous with pdf f, and that we can sample from a distribution g. If there exists a constant c such that $\sup\{f(x)/g(x) \mid x \in \text{dom}\, f\} \leq c$, then we can generate samples from f by acceptance/rejection.

Algorithm 3 (Generating X from pdf f by acceptance/rejection). Given pdf g and constant c such that $\sup\{f(x)/g(x) \mid x \in \text{dom}\, f\} \leq c$:

1. generate $Y \sim g$
2. generate $U \sim \text{Uni}(0,1)$
3. if $U \leq f(Y)/cg(Y)$, set $X = Y$; otherwise, go to step 1

The expected number of iterations before Y is accepted is exactly c, so g should be chosen to make c is as close to one as possible.

Algorithm 4 (Generating X from pmf p by acceptance/rejection). Given pmf q and constant c such that $p(x_i)/g(x_i) \leq c$ for all i:

1. generate $Y \sim q$
2. generate $U \sim \text{Uni}(0,1)$
3. if $U \leq p(Y)/cq(Y)$, set $X = Y$; otherwise, go to step 1

Chapter 3

Improving the Monte Carlo algorithm

3.1 Comparing estimation algorithms

Suppose we have two candidate algorithms for estimating θ. Which algorithm is better? To answer this question, we need a way to quantify each algorithm's performance. Intuitively, this performance metric should include both how much computational effort the algorithm requires, and how good of an estimator it produces. If we take the former to be the algorithm's *cost* and the latter to be its *benefit*, then it is reasonable to prefer the algorithm with the smaller cost/benefit ratio.

Definition 1 (Cost and benefit). *For $\alpha \in (0,1)$ and $\varepsilon_\alpha > 0$, let $\mathcal{A}$ be an algorithm that produces an unbiased estimator $\hat{\theta}$ of θ such that with approximately $100(1-\alpha)\%$ confidence,*

$$\theta \in \left[\hat{\theta} - \varepsilon_\alpha,\ \hat{\theta} + \varepsilon_\alpha\right].$$

The cost of $\mathcal{A}$ is the time required to compute $\hat{\theta}$. The benefit of $\mathcal{A}$ is $(z_{1-\alpha/2}/\varepsilon_\alpha)^2$.

Why do we choose this definition of benefit? The definition includes $z_{1-\alpha/2}$ so that the benefit of $\mathcal{A}$ is independent of the confidence level α. The benefit being inversely proportional to ε_α makes sense: A useless estimator ($\varepsilon_\alpha \to \infty$) gives zero benefit, and the benefit increases monotonically as the estimator gets better ($\varepsilon_\alpha \to 0$). We square the ratio $z_{1-\alpha/2}/\varepsilon_\alpha$ in order to get a simple expression for the cost/benefit ratio of the Monte Carlo algorithm 1.

3.1.1 Cost/benefit ratio of the Monte Carlo algorithm

The Monte Carlo algorithm 1 produces an estimator of $\theta = \mathbb{E}\, h(\mathbf{X})$ by generating samples $\mathbf{X}_1, \ldots, \mathbf{X}_n$ from the distribution of $\mathbf{X}$, then computing $\bar{\theta}_n = (h(\mathbf{X}_1) + \cdots + h(\mathbf{X}_n))/n$. What is the cost/benefit ratio of this algorithm?

Let t be the time required to generate $\mathbf{X}_1$ from the distribution of $\mathbf{X}$ and compute $h(\mathbf{X}_1)$. In general, these operations are much more computationally expensive than the simple arithmetic operations required to compute $\bar{\theta}_n$ and $\bar{\sigma}_n$ from $h(\mathbf{X}_1), \ldots, h(\mathbf{X}_n)$. Therefore, the cost

of algorithm 1 is approximately nt. The $100(1-\alpha)\%$ confidence interval half-width associated with $\bar{\theta}_n$ is

$$\varepsilon_\alpha = \frac{\sigma z_{1-\alpha/2}}{\sqrt{n}},$$

so the benefit of algorithm 1 is n/σ^2. The approximate cost/benefit ratio is

$$\frac{nt}{n/\sigma^2} = t\sigma^2.$$

This simple expression – the product of the single-sample computation time and the variance constant – is the standard metric for comparing Monte Carlo algorithms.

3.1.2 Goal of this chapter

The goal of this chapter is to reduce the cost/benefit ratio of Monte Carlo algorithm 1. We discuss five methods that aim to accomplish this by reducing the estimator variance without unduly increasing computation time. The methods cleverly construct a new random variable Y such that $Y \neq h(\mathbf{X})$ but $\mathbb{E}\, Y = \theta$. For example, Y can be computed by applying some function $f : \mathbb{R}^m \to \mathbb{R}$ to $\mathbf{X}$, where $f \neq h$ but $\mathbb{E}\, f(\mathbf{X}) = \mathbb{E}\, h(\mathbf{X})$. Or, Y can be another function $g : \mathbb{R}^d \to \mathbb{R}$ of some other random vector $\mathbf{Z} \in \mathbb{R}^d$. In either case, we require the estimator

$$\hat{\theta}_n := \frac{1}{n}\sum_{i=1}^{n} Y_i$$

to be asymptotically distributed according to

$$\frac{\sqrt{n}(\hat{\theta}_n - \theta)}{\sigma_y} \Rightarrow \mathcal{N}(0,1) \text{ as } n \to \infty$$

for some variance constant σ_y^2. We want to choose Y such that the cost/benefit ratio of estimating θ by $\hat{\theta}_n$ is smaller than that of estimating θ by $\bar{\theta}_n$. In other words, we want

$$t_y\sigma_y^2 < t\sigma^2,$$

where t_y is the time required to generate one sample from the distribution of Y.

3.2 Conditional Monte Carlo

Suppose we discover some random vector $\mathbf{Z} \in \mathbb{R}^d$ such that

1. it is easy to generate samples from the distribution of $\mathbf{Z}$, and
2. we can compute $\mathbb{E}\,[h(\mathbf{X}) \mid \mathbf{Z} = \mathbf{z}]$ *exactly* for any realization $\mathbf{z}$ of $\mathbf{Z}$.

Can we use this new information to reduce the cost/benefit ratio of the Monte Carlo algorithm 1?

Conditional Monte Carlo methods do exactly that. The basic idea is to use iterated expectation:

$$\mathbb{E}\, h(\mathbf{X}) = \mathbb{E}\left[\mathbb{E}\left[h(\mathbf{X}) \mid \mathbf{Z}\right]\right].$$

Here the inner expectation is taken with respect to the conditional distribution of $h(\mathbf{X})$ given $\mathbf{Z}$, and the outer expectation is taken with respect to the distribution of $\mathbf{Z}$. Put another way, the expected value (with respect to the distribution of $\mathbf{Z}$) of the random variable

$$Y := \mathbb{E}\left[h(\mathbf{X}) \mid \mathbf{Z}\right]$$

is exactly $\mathbb{E}\, h(\mathbf{X})$. Therefore, we can estimate $\theta = \mathbb{E}\, h(\mathbf{X})$ by

$$\hat{\theta}_n := \frac{1}{n}\sum_{i=1}^{n} Y_i,$$

where $Y_1, \ldots, Y_n$ are iid samples from the distribution of Y.

To find the asymptotic distribution of $\hat{\theta}_n$, we apply iterated variance:

$$\operatorname{Var} h(\mathbf{X}) = \mathbb{E} \operatorname{Var}\left(h(\mathbf{X}) \mid \mathbf{Z}\right) + \operatorname{Var} \mathbb{E}\left[h(\mathbf{X}) \mid \mathbf{Z}\right].$$

Because $\operatorname{Var} h(\mathbf{X}) = \sigma^2$, $Y = \mathbb{E}\left[h(\mathbf{X}) \mid \mathbf{Z}\right]$, and $\mathbb{E} \operatorname{Var}\left(h(\mathbf{X}) \mid \mathbf{Z}\right) \geq 0$, we have

$$\begin{aligned}\sigma_y^2 := \operatorname{Var} Y &= \sigma^2 - \mathbb{E} \operatorname{Var}\left(h(\mathbf{X}) \mid \mathbf{Z}\right) \\ &\leq \sigma^2.\end{aligned}$$

Since $\sigma^2 < \infty$ by assumption, $Y_1, Y_2, \ldots$ is a sequence of iid random variables with finite expectation θ and finite variance σ_y^2. By the Central Limit Theorem, therefore, $\hat{\theta}_n$ is asymptotically distributed according to

$$\frac{\sqrt{n}(\hat{\theta}_n - \theta)}{\sigma_y} \Rightarrow \mathcal{N}(0,1) \text{ as } n \to \infty.$$

When constructing confidence intervals, we estimate the unknown variance constant σ_y^2 by

$$\hat{\sigma}_{y,n}^2 := \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \hat{\theta}_n)^2.$$

3.2.1 The Conditional Monte Carlo algorithm

Algorithm 5 (Conditional Monte Carlo estimation of $\theta = \mathbb{E}\, h(\mathbf{X})$). Given confidence level $\alpha \in (0,1)$, sample size $n \in \mathbb{N}$, and function $g : \mathbb{R}^d \to \mathbb{R}$ such that $g(\mathbf{z}) = \mathbb{E}\,[h(\mathbf{X}) \mid \mathbf{Z} = \mathbf{z}]$:

1. generate $\mathbf{Z}_1, \ldots, \mathbf{Z}_n$ iid from the distribution of $\mathbf{Z}$
2. compute $Y_1 = g(\mathbf{Z}_1), \ldots, Y_n = g(\mathbf{Z}_n)$
3. estimate $\theta = \mathbb{E}\, Y$ and $\sigma_y^2 = \operatorname{Var} Y$ by

$$\hat{\theta}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i, \quad \hat{\sigma}_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \hat{\theta}_n)^2$$

4. conclude with approximately $100(1-\alpha)\%$ confidence that

$$\theta \in \left[\hat{\theta}_n - \frac{z_{1-\alpha/2}\hat{\sigma}_{y,n}}{\sqrt{n}},\ \hat{\theta}_n + \frac{z_{1-\alpha/2}\hat{\sigma}_{y,n}}{\sqrt{n}}\right]$$

3.2.2 Cost/benefit ratio

What is the cost/benefit ratio of the conditional Monte Carlo algorithm 5? Let t_y be the time required to generate $\mathbf{Z}_1$ from the distribution of $\mathbf{Z}$ and compute $Y_1 = g(\mathbf{Z}_1)$. The time required to generate the samples and compute the $g(\mathbf{Z}_i)$ is usually much larger than the time required to execute the arithmetic operations in step 3 of algorithm 5, so the approximate cost is nt_y. The conditional Monte Carlo $100(1-\alpha)\%$ confidence interval half-width is $\varepsilon_\alpha = z_{1-\alpha/2}\sigma_y/\sqrt{n}$, so the benefit of algorithm 5 is $(z_{1-\alpha/2}/\varepsilon_\alpha)^2 = n/\sigma_y^2$. Therefore, the approximate cost/benefit ratio is

$$\frac{nt_y}{n/\sigma_y^2} = t_y\sigma_y^2.$$

We have shown that $\sigma_y^2 \le \sigma^2$ (where $\sigma^2 = \operatorname{Var} h(\mathbf{X})$), so the conditional Monte Carlo algorithm 5 improves the cost/benefit ratio of the Monte Carlo algorithm 1 if $t_y \le t$ (where t is the time required to generate $\mathbf{X}_1$ from the distribution of $\mathbf{X}$ and compute $h(\mathbf{X}_1)$). In particular, this requires that the subroutine g be reasonably efficient.

3.3 Control variates

Suppose that in the process of generating $\mathbf{X}_i$ from the distribution of $\mathbf{X}$ and computing $h(\mathbf{X}_i)$, we also generate a sample $\mathbf{Z}_i$ from the distribution of an intermediate random vector $\mathbf{Z} \in \mathbb{R}^d$. Suppose as well that

1. $\boldsymbol{\mu}_z := \mathbb{E}\,\mathbf{Z}$ is *known exactly*,
2. $\Sigma_{zz} := \operatorname{Cov}\mathbf{Z} \in \mathbb{R}^{d\times d}$ is nonsingular, and
3. $\mathbb{E}[Z_1^2 + \cdots + Z_d^2] < \infty$.

The second and third assumptions are technical conditions, but the assumption that $\boldsymbol{\mu}_z$ is known exactly is crucial. In fact, it's the fundamental reason that the method of control variates works: As a byproduct of the Monte Carlo algorithm 1, we get additional information *for free*. The components $Z_1, \ldots, Z_d$ of $\mathbf{Z}$ are called *control variates*.

How can we use control variates to construct an estimator of $\theta = \mathbb{E}\,h(\mathbf{X})$ that has smaller variance than the standard Monte Carlo estimator $\bar{\theta}_n$? Consider the random variable

$$Y(\boldsymbol{\lambda}) := h(\mathbf{X}) - \boldsymbol{\lambda}^\top(\mathbf{Z} - \boldsymbol{\mu}_z),$$

where $\boldsymbol{\lambda} \in \mathbb{R}^d$ is an arbitrary parameter vector over which we will eventually optimize. If $Y_1(\boldsymbol{\lambda}), \ldots, Y_n(\boldsymbol{\lambda})$ are generated iid from the distribution of $Y(\boldsymbol{\lambda})$, then the estimator

$$\hat{\theta}_n(\boldsymbol{\lambda}) := \frac{1}{n}\sum_{i=1}^{n} Y_i(\boldsymbol{\lambda})$$

of θ is unbiased and consistent for any $\boldsymbol{\lambda}$. Furthermore, assumption 3 implies that the variance of $Y(\boldsymbol{\lambda})$ is finite:

$$\begin{aligned}\sigma_y^2(\boldsymbol{\lambda}) := \operatorname{Var} Y(\boldsymbol{\lambda}) &= \operatorname{Var} h(\mathbf{X}) - 2\boldsymbol{\lambda}^\top \operatorname{Cov}(\mathbf{Z}, h(\mathbf{X})) + \boldsymbol{\lambda}^\top(\operatorname{Cov}\mathbf{Z})\boldsymbol{\lambda} \\ &= \sigma^2 - 2\boldsymbol{\lambda}^\top\boldsymbol{\Sigma}_{zy} + \boldsymbol{\lambda}^\top\Sigma_{zz}\boldsymbol{\lambda} \\ &< \infty,\end{aligned}$$

where

$$\boldsymbol{\Sigma}_{zy} := \operatorname{Cov}(\mathbf{Z}, h(\mathbf{X})) \in \mathbb{R}^d.$$

Because $Y_1(\boldsymbol{\lambda}), Y_2(\boldsymbol{\lambda}), \ldots$ is a sequence of iid random variables with finite mean θ and finite variance $\sigma_y^2(\boldsymbol{\lambda})$, the Central Limit Theorem gives the asymptotic distribution of $\hat{\theta}_n(\boldsymbol{\lambda})$:

$$\frac{\sqrt{n}(\hat{\theta}_n(\boldsymbol{\lambda}) - \theta)}{\sigma_y(\boldsymbol{\lambda})} \Rightarrow \mathcal{N}(0,1) \text{ as } n \to \infty.$$

3.3.1 Choosing the parameter $\boldsymbol{\lambda}$

To maximize the variance reduction achieved by the control variates method, we minimize $\sigma_y^2(\boldsymbol{\lambda}) = \operatorname{Var} Y(\boldsymbol{\lambda})$ over $\boldsymbol{\lambda}$. Because $\sigma_y^2(\cdot)$ is a convex function, the first-order condition $\nabla\sigma_y^2(\boldsymbol{\lambda}^\star) = \mathbf{0}$ is necessary and sufficient for the the global optimality of $\boldsymbol{\lambda}^\star$. Thus,

$$\nabla\sigma_y^2(\boldsymbol{\lambda}^\star) = -2\boldsymbol{\Sigma}_{zy} + 2\Sigma_{zz}\boldsymbol{\lambda}^\star = \mathbf{0} \quad \Longrightarrow \quad \boldsymbol{\lambda}^\star = \Sigma_{zz}^{-1}\boldsymbol{\Sigma}_{zy}.$$

By assumption 2, the inverse Σ_{zz}^{-1} exists.

In practice, the components of Σ_{zz} and $\boldsymbol{\Sigma}_{zy}$, which are built from variances and correlation coefficients, are usually unknown. Therefore, $\boldsymbol{\lambda}^\star$ is usually not computable. We can estimate $\boldsymbol{\lambda}^\star$, however, by

$$\hat{\boldsymbol{\lambda}}_n := \hat{\Sigma}_{zz}^{-1}\hat{\boldsymbol{\Sigma}}_{zy},$$

where

$$\hat{\Sigma}_{zz} := \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{Z}_i - \boldsymbol{\mu}_z)(\mathbf{Z}_i - \boldsymbol{\mu}_z)^\top, \quad \hat{\boldsymbol{\Sigma}}_{zy} := \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{Z}_i - \boldsymbol{\mu}_z)(h(\mathbf{X}_i) - \bar{\theta}_n).$$

This gives the approximately optimal control variates estimator of θ:

$$\hat{\theta}_n(\hat{\boldsymbol{\lambda}}_n) = \frac{1}{n}\sum_{i=1}^{n} Y_i(\hat{\boldsymbol{\lambda}}_n).$$

An unbiased and consistent estimator of $\sigma_y^2(\boldsymbol{\lambda}^\star) = \operatorname{Var} Y(\boldsymbol{\lambda}^\star)$ is

$$\hat{\sigma}_{y,n}^2(\hat{\boldsymbol{\lambda}}_n) := \frac{1}{n-1}\sum_{i=1}^{n}\left(Y_i(\hat{\boldsymbol{\lambda}}_n) - \hat{\theta}_n(\hat{\boldsymbol{\lambda}}_n)\right)^2.$$

By the Converging Together and Central Limit Theorems, therefore, the estimator $\hat{\theta}_n(\hat{\boldsymbol{\lambda}}_n)$ of θ is asymptotically distributed according to

$$\frac{\sqrt{n}(\hat{\theta}_n(\hat{\boldsymbol{\lambda}}_n) - \theta)}{\hat{\sigma}_{y,n}(\hat{\boldsymbol{\lambda}}_n)} \Rightarrow \mathcal{N}(0,1) \text{ as } n \to \infty.$$

3.3.2 Choosing the control variate Z

Plugging $\boldsymbol{\lambda}^\star$ into $\sigma_y^2(\cdot)$, the minimum variance is

$$\begin{aligned}\sigma_y^2(\boldsymbol{\lambda}^\star) &= \sigma^2 - 2(\Sigma_{zz}^{-1}\boldsymbol{\Sigma}_{zy})^\top\boldsymbol{\Sigma}_{zy} + (\Sigma_{zz}^{-1}\boldsymbol{\Sigma}_{zy})^\top\Sigma_{zz}(\Sigma_{zz}^{-1}\boldsymbol{\Sigma}_{zy}) \\ &= \sigma^2 - \boldsymbol{\Sigma}_{zy}^\top\Sigma_{zz}^{-1}\boldsymbol{\Sigma}_{zy} \\ &= (1 - R^2)\sigma^2,\end{aligned}$$

where

$$R^2 := \frac{\boldsymbol{\Sigma}_{zy}^\top\Sigma_{zz}^{-1}\boldsymbol{\Sigma}_{zy}}{\sigma^2} \in [0,1].$$

To maximize the variance reduction, we want to choose the control variates $Z_1, \ldots, Z_d$ so that R^2 is as close to 1 as possible. In the worst-case scenario, where $R = 0$, we achieve no variance reduction. By assumption 2, Σ_{zz} (hence Σ_{zz}^{-1}) is positive definite, so $R = 0$ occurs only if $\boldsymbol{\Sigma}_{zy} = \mathbf{0}$, meaning every control variate is uncorrelated with $h(\mathbf{X})$.

In the case of a single control variate ($d = 1$), R reduces to the correlation coefficient between $h(\mathbf{X})$ and Z. In the scalar case, therefore, we want to choose the control variate Z to be as strongly correlated (either positively or negatively) with $h(\mathbf{X})$ as possible.

3.3.3 The control variates algorithm

Algorithm 6 (Control variates estimation of $\theta = \mathbb{E}\, h(\mathbf{X})$). Given confidence level $\alpha \in (0, 1)$, sample size $n \in \mathbb{N}$, and control variate expectation $\boldsymbol{\mu}_z$:

1. generate $(\mathbf{X}_1, \mathbf{Z}_1), \ldots, (\mathbf{X}_n, \mathbf{Z}_n)$ iid from the joint distribution of $(\mathbf{X}, \mathbf{Z})$
2. estimate $\boldsymbol{\lambda}^\star$ by $\hat{\boldsymbol{\lambda}}_n = \hat{\Sigma}_{zz}^{-1} \hat{\boldsymbol{\Sigma}}_{zy}$, where

$$\hat{\Sigma}_{zz} = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{Z}_i - \boldsymbol{\mu}_z)(\mathbf{Z}_i - \boldsymbol{\mu}_z)^\top, \quad \hat{\boldsymbol{\Sigma}}_{zy} = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{Z}_i - \boldsymbol{\mu}_z)(h(\mathbf{X}_i) - \bar{\theta}_n),$$

   and $\bar{\theta}_n = (h(\mathbf{X}_1) + \cdots + h(\mathbf{X}_n))/n$ is the usual Monte Carlo estimator of θ

3. compute

$$Y_i(\hat{\boldsymbol{\lambda}}_n) = h(\mathbf{X}_i) - \hat{\boldsymbol{\lambda}}_n^\top (\mathbf{Z}_i - \boldsymbol{\mu}_z), \quad i = 1, \ldots, n$$

4. estimate $\theta = \mathbb{E}\, Y(\hat{\boldsymbol{\lambda}}_n)$ and $\sigma_y^2(\hat{\boldsymbol{\lambda}}_n) = \operatorname{Var} Y(\hat{\boldsymbol{\lambda}}_n)$ by

$$\hat{\theta}_n(\hat{\boldsymbol{\lambda}}_n) = \frac{1}{n} \sum_{i=1}^{n} Y_i(\hat{\boldsymbol{\lambda}}_n), \quad \hat{\sigma}_{y,n}^2(\hat{\boldsymbol{\lambda}}_n) = \frac{1}{n-1} \sum_{i=1}^{n} \Big(Y_i(\hat{\boldsymbol{\lambda}}_n) - \hat{\theta}_n(\hat{\boldsymbol{\lambda}}_n) \Big)^2$$

5. conclude with approximately $100(1-\alpha)\%$ confidence that

$$\theta \in \left[\hat{\theta}_n(\hat{\boldsymbol{\lambda}}_n) - \frac{\hat{\sigma}_{y,n}(\hat{\boldsymbol{\lambda}}_n) z_{1-\alpha/2}}{\sqrt{n}},\ \hat{\theta}_n(\hat{\boldsymbol{\lambda}}_n) + \frac{\hat{\sigma}_{y,n}(\hat{\boldsymbol{\lambda}}_n) z_{1-\alpha/2}}{\sqrt{n}} \right]$$

3.3.4 Cost/benefit ratio

The computational cost of the control variates algorithm 6 is greater than that of the Monte Carlo algorithm 1. Indeed, step 3 of the control variates algorithm requires computing the

Monte Carlo estimator $\bar{\theta}_n$. The control variates method also requires inverting $\hat{\Sigma}_{zz} \in \mathbb{R}^{d\times d}$, which may become costly if d is large, *i.e.*, if many control variates are used.

For large n, however, the time required to compute $\hat{\theta}_n(\hat{\boldsymbol{\lambda}}_n)$ is approximately nt, where t is the time required to generate $\mathbf{X}_1$ and compute $h(\mathbf{X}_1)$. Therefore, the approximate cost/benefit ratio of the control variates algorithm 6 is

$$t\sigma_y^2(\boldsymbol{\lambda}^\star) = (1 - R^2)t\sigma^2.$$

Recall that the cost/benefit ratio of the Monte Carlo algorithm 1 is $t\sigma^2$. Therefore, the method of control variates reduces the cost/benefit ratio by

$$t\sigma^2 - (1 - R^2)t\sigma^2 = R^2 t\sigma^2.$$

The cost/benefit ratio reduction can be large when the control variates $Z_1, \ldots, Z_d$ are chosen such that

$$R^2 = \frac{\operatorname{Cov}(\mathbf{Z}, h(\mathbf{X}))^\top (\operatorname{Cov}\mathbf{Z})^{-1} \operatorname{Cov}(\mathbf{Z}, h(\mathbf{X}))}{\operatorname{Var} h(\mathbf{X})}$$

is close to 1.

3.4 Evil twins

In the Monte Carlo algorithm 1, we generate samples $\mathbf{X}_1, \ldots, \mathbf{X}_n$ *independently* from the distribution of $\mathbf{X}$. This means that $h(\mathbf{X}_1), \ldots, h(\mathbf{X}_n)$ are also independent. But suppose that every time we generated $\mathbf{X}_i$ from the distribution of $\mathbf{X}$, we generated another sample $\mathbf{X}_i'$ from the distribution of $\mathbf{X}$ such that $h(\mathbf{X}_i)$ and $h(\mathbf{X}_i')$ were *dependent*. Could we design an algorithm to take advantage of this dependence?

The method of evil twins, also known as antithetic variates, does exactly that. We call $h(\mathbf{X}_i')$ the *evil twin* or *antithesis* of $h(\mathbf{X}_i)$. Let's define

$$Y_i := \frac{h(\mathbf{X}_i) + h(\mathbf{X}_i')}{2}, \quad \hat{\theta}_n := \frac{1}{n}\sum_{i=1}^n Y_i.$$

The evil twins estimator $\hat{\theta}_n$ of θ is unbiased and consistent. Furthermore, $Y_1, Y_2, \ldots$ is a sequence of iid random variables with mean $\theta < \infty$ and variance

$$\begin{aligned}
\sigma_y^2 := \operatorname{Var} Y &= \operatorname{Var}\left(\frac{h(\mathbf{X}_1) + h(\mathbf{X}_1')}{2}\right) \\
&= \frac{\operatorname{Var} h(\mathbf{X}_1) + \operatorname{Var} h(\mathbf{X}_1') + 2\operatorname{Cov}(h(\mathbf{X}_1), h(\mathbf{X}_1'))}{4} \\
&= \frac{(1+\rho)\sigma^2}{2} \\
&< \infty,
\end{aligned}$$

where $\rho \in [-1, 1]$ is the correlation between $h(\mathbf{X}_1)$ and $h(\mathbf{X}'_1)$. By the Central Limit Theorem, therefore, $\hat{\theta}_n$ is asymptotically distributed according to

$$\frac{\sqrt{n}(\hat{\theta}_n - \theta)}{\sigma_y} \Rightarrow \mathcal{N}(0, 1) \text{ as } n \to \infty. \tag{3.1}$$

When constructing confidence intervals, we estimate the unknown variance constant σ_y^2 by

$$\hat{\sigma}_{y,n}^2 := \frac{1}{n-1} \sum_{i=1}^{n} (Y_i - \hat{\theta}_n)^2.$$

3.4.1 Cost/benefit ratio

What is the cost/benefit ratio of the evil twins algorithm? Recall our definition that t is the time required to generate $\mathbf{X}_1$ from the distribution of $\mathbf{X}$ and compute $h(\mathbf{X}_1)$. Computing $Y_1 = (h(\mathbf{X}_1)+h(\mathbf{X}'_1))/2$ requires time $2t$, so the time required to compute $\hat{\theta}_n$ is approximately $2tn$. The asymptotic distribution (3.1) gives an approximate $100(1-\alpha)\%$ confidence interval half-width of $\varepsilon_\alpha = z_{1-\alpha/2}\sigma_y/\sqrt{n}$, so the benefit of the evil twins algorithm is $(z_{1-\alpha/2}/\varepsilon_\alpha)^2 = n/\sigma_y^2$. Therefore, the evil twins cost/benefit ratio is

$$\frac{2tn}{n/\sigma_y^2} = 2t\sigma_y^2 = 2t\left(\frac{(1+\rho)\sigma^2}{2}\right) = t(1+\rho)\sigma^2.$$

Do evil twins have a lower cost/benefit ratio than the Monte Carlo algorithm 1? The approximate cost/benefit ratio reduction is

$$t\sigma^2 - t(1+\rho)\sigma^2 = -\rho t\sigma^2.$$

Therefore, evil twins reduce the cost/benefit ratio whenever $\rho < 0$, meaning $h(\mathbf{X}_i)$ and $h(\mathbf{X}'_i)$ are negatively correlated. If $h(\mathbf{X}'_i)$ is a perfect evil twin of $h(\mathbf{X})$, meaning $\rho = -1$, then the cost/benefit ratio is reduced to *zero*. If $h(\mathbf{X}_i)$ and $h(\mathbf{X}'_i)$ are positively correlated, however, then evil twins *increase* the cost/benefit ratio.

3.4.2 Inducing a negative correlation between $h(\mathbf{X}_i)$ and $h(\mathbf{X}'_i)$

As discussed in §3.4.1, the evil twins algorithm hinges on the ability to make $h(\mathbf{X}'_i)$ negatively correlated with $h(\mathbf{X}_i)$. But h can be a very complicated function, and we have control only over the inputs $\mathbf{X}_i$ and $\mathbf{X}'_i$. So how can we induce this negative correlation?

In this section, we provide a method to guarantee that the correlation ρ between $h(\mathbf{X}_i)$ and $h(\mathbf{X}'_i)$ is negative. We assume that

1. $h : \mathbb{R}^m \to \mathbb{R}$ is monotonic in each of its arguments,

2. the components $X_1, \ldots, X_m$ of $\mathbf{X}$ are independent, and

3. the function $\mathbf{F}^{-1} : \mathbb{R}^m \to \mathbb{R}^m$, defined by

$$\mathbf{F}^{-1}(\mathbf{u}) = \begin{bmatrix} F_1^{-1}(u_1) \\ \vdots \\ F_m^{-1}(u_m) \end{bmatrix},$$

is well-defined on $(0,1)^m$, where F_j is the cdf of X_j.

Assumptions 2 and 3 guarantee that we can generate $\mathbf{X}_i$ and $\mathbf{X}_i'$ from the distribution of $\mathbf{X}$ using the inversion method (see §2.2.1). We do this by first generating $\mathbf{U}_i \sim \text{Uni}(0,1)^m$, then setting

$$\mathbf{X}_i = \mathbf{F}^{-1}(\mathbf{U}_i), \quad \mathbf{X}_i' = \mathbf{F}^{-1}(\mathbf{1} - \mathbf{U}_i),$$

where $\mathbf{1} = (1, \ldots, 1) \in \mathbb{R}^m$. Because $\mathbf{1} - \mathbf{U}_i$ is uniformly distributed on $(0,1)^m$, $\mathbf{X}_i'$ has the same distribution as $\mathbf{X}_i$ (hence $\mathbf{X}$).

We claim that $(\mathbf{X}_i)_j$ and $(\mathbf{X}_i')_j$ are negatively correlated for each $j = 1, \ldots, m$. To see this, observe that each F_j is a cdf, hence nondecreasing. It can be shown that F_j^{-1} is also nondecreasing. This means that $(\mathbf{X}_i)_j = F_j^{-1}((\mathbf{U}_i)_j)$ is small when $(\mathbf{U}_i)_j$ is small, and $(\mathbf{X}_i)_j$ is large when $(\mathbf{U}_i)_j$ is large. By contrast, $(\mathbf{X}_i')_j = F_j^{-1}(1 - (\mathbf{U}_i)_j)$ is *large* when $(\mathbf{U}_i)_j$ is small (hence $1 - (\mathbf{U}_i)_j$ is large), and $(\mathbf{X}_i')_j$ is *small* when $(\mathbf{U}_i)_j$ is large. This suggests that the correlation between $(\mathbf{X}_i)_j$ and $(\mathbf{X}_i')_j$ is negative for each j.

Does the negative correlation between each $(\mathbf{X}_i)_j$ and $(\mathbf{X}_i')_j$ also mean that $h(\mathbf{X}_i)$ and $h(\mathbf{X}_i')$ are negatively correlated? Not necessarily, but assumption 1 guarantees that this is the case. A formal proof of this fact, along with a more rigorous statement of the argument in the previous paragraph, can be found in the appendix of Chapter 8 of [2].

The componentwise monotonicity of h is a sufficient but not necessary condition for evil twins to improve efficiency. If this condition doesn't hold (or holds with respect to some but not all arguments of h), then evil twins may still reduce variance – but there are no guarantees.

3.4.3 The evil twins algorithm

Algorithm 7 (Evil twins estimation of $\theta = \mathbb{E}\, h(\mathbf{X})$). Given confidence level $\alpha \in (0,1)$, sample size $n \in \mathbb{N}$, and inverse cdf $\mathbf{F}^{-1} : \mathbb{R}^m \to \mathbb{R}^m$ defined such that if $\mathbf{U} \sim \text{Uni}(0,1)^m$, then $\mathbf{F}^{-1}(\mathbf{U})$ has the distribution of $\mathbf{X}$:

1. generate $\mathbf{U}_1, \ldots, \mathbf{U}_n$ iid from the $\text{Uni}(0,1)^m$ distribution
2. compute $Y_1, \ldots, Y_n$, where
$$Y_i = \frac{h(\mathbf{F}^{-1}(\mathbf{U}_i)) + h(\mathbf{F}^{-1}(\mathbf{1} - \mathbf{U}_i))}{2}$$
3. estimate $\theta = \mathbb{E}\, Y$ and $\sigma_y^2 = \text{Var}\, Y$ by
$$\hat{\theta}_n := \frac{1}{n}\sum_{i=1}^{n} Y_i, \quad \hat{\sigma}_{y,n}^2 := \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \hat{\theta}_n)^2$$
4. conclude with approximately $100(1-\alpha)\%$ confidence that
$$\theta \in \left[\hat{\theta}_n - \frac{\hat{\sigma}_{y,n} z_{1-\alpha/2}}{\sqrt{n}},\ \hat{\theta}_n + \frac{\hat{\sigma}_{y,n} z_{1-\alpha/2}}{\sqrt{n}}\right]$$

3.5 Importance sampling

3.5.1 Motivation: Rare-event simulation

Importance sampling is a technique that's mainly used for *rare-event simulation*, where we want to estimate very small, but positive, probabilities. For example, an electric power system operator might be interested in the probability that a power outage will happen tomorrow, under two different generator dispatch scenarios. In a well-designed system, this probability should be small in either scenario. However, the difference between the two probabilities is a key reliability consideration.

Standard Monte Carlo methods are bad at rare-event simulation. Suppose, for example, we want to estimate $\mathbb{P}\{\mathbf{X} \in A\}$ for some random vector $\mathbf{X} \in \mathbb{R}^m$ and set $A \subset \mathbb{R}^m$. We can do this with the Monte Carlo algorithm 1 by setting $h = \mathcal{I}_A$, where $\mathcal{I}_A : \mathbb{R}^m \to \mathbb{R}$ is the indicator function of the set A. If $\mathbb{E}\, h(\mathbf{X}) = \mathbb{P}\{\mathbf{X} \in A\}$ is small, say 10^{-10}, then we expect to generate about 10^{10} samples $\mathbf{X}_i$ from the distribution of $\mathbf{X}$ before observing $\mathbf{X}_i \in A$ even once. If generating $\mathbf{X}_i$ is computationally expensive, then we probably can't afford to generate this many replications – let alone enough replications to estimate $\mathbb{P}\{\mathbf{X} \in A\}$ with

any degree of accuracy. Loosely speaking, importance sampling circumvents this difficulty by preferentially sampling from the *important* values of the support of $\mathbf{X}$.

3.5.2 The importance sampling estimator

Suppose the random vector $\mathbf{X} \in \mathbb{R}^m$ is continuous with pdf f. Let g be another pdf with the same support as f, *i.e.*,

$$f(\mathbf{x}) \neq 0 \quad \Longrightarrow \quad g(\mathbf{x}) \neq 0$$

for all $\mathbf{x} \in \mathbb{R}^m$. (We say that any g satisfying this property is an *importance sampling density*.) Then

$$\begin{aligned} \theta = \mathbb{E}\, h(\mathbf{X}) &= \int_{\mathbb{R}^m} h(\mathbf{x}) f(\mathbf{x}) \mathrm{d}\mathbf{x} = \int_{\mathbb{R}^m} \left(\frac{h(\mathbf{x}) f(\mathbf{x})}{g(\mathbf{x})} \right) g(\mathbf{x}) \mathrm{d}\mathbf{x} \\ &= \mathbb{E}\, \tilde{h}(\tilde{\mathbf{X}}). \end{aligned}$$

Here the random vector $\tilde{\mathbf{X}}$ has pdf g, and $\tilde{h} : \mathbb{R}^m \to \mathbb{R}$ is defined by

$$\tilde{h}(\mathbf{x}) = \begin{cases} 0 & \text{if } f(\mathbf{x}) = 0 \\ h(\mathbf{x}) f(\mathbf{x}) / g(\mathbf{x}) & \text{otherwise.} \end{cases}$$

If we can generate iid samples $\tilde{\mathbf{X}}_1, \ldots, \tilde{\mathbf{X}}_n$ from g, then we can estimate θ by

$$\hat{\theta}_n := \frac{1}{n} \sum_{i=1}^{n} Y_i,$$

where $Y_i = \tilde{h}(\tilde{\mathbf{X}}_i)$. The estimator $\hat{\theta}_n$ of θ is unbiased and consistent. If $\mathbb{E}[\tilde{h}(\tilde{\mathbf{X}})^2] < \infty$, then

$$\sigma_y^2 := \operatorname{Var} \tilde{h}(\tilde{\mathbf{X}}) = \mathbb{E}[\tilde{h}(\tilde{\mathbf{X}})^2] - \theta^2 < \infty,$$

so $Y_1, Y_2, \ldots$ is a sequence of iid random variables with finite expectation θ and finite variance σ_y^2. By the Central Limit Theorem, therefore, $\hat{\theta}_n$ is asymptotically distributed according to

$$\frac{\sqrt{n}(\hat{\theta}_n - \theta)}{\sigma_y} \Rightarrow \mathcal{N}(0, 1) \text{ as } n \to \infty.$$

When constructing confidence intervals, we estimate the unknown variance constant σ_y^2 by

$$\hat{\sigma}_{y,n}^2 := \frac{1}{n-1} \sum_{i=1}^{n} (Y_i - \hat{\theta}_n)^2.$$

3.5.3 The importance sampling algorithm

Algorithm 8 (Importance sampling estimation of $\theta = \mathbb{E}\, h(\mathbf{X})$). Given confidence level $\alpha \in (0,1)$, sample size $n \in \mathbb{N}$, and importance sampling density g:

1. generate $\tilde{\mathbf{X}}_1, \ldots, \tilde{\mathbf{X}}_n$ iid from g
2. compute $Y_1 = \tilde{h}(\tilde{\mathbf{X}}_1), \ldots, Y_n = \tilde{h}(\tilde{\mathbf{X}}_n)$, where

$$\tilde{h}(\mathbf{x}) = \begin{cases} 0 & \text{if } f(\mathbf{x}) = 0 \\ h(\mathbf{x}) f(\mathbf{x}) / g(\mathbf{x}) & \text{otherwise,} \end{cases}$$

   and f is the pdf of $\mathbf{X}$
3. estimate $\theta = \mathbb{E}\, \tilde{h}(\tilde{\mathbf{X}})$ and $\sigma_y^2 = \operatorname{Var} \tilde{h}(\tilde{\mathbf{X}})$ by

$$\hat{\theta}_n = \frac{1}{n} \sum_{i=1}^{n} Y_i, \quad \hat{\sigma}_{y,n}^2 = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{\theta}_n)^2$$

4. conclude with approximately $100(1-\alpha)\%$ confidence that

$$\theta \in \left[\hat{\theta}_n - \frac{\hat{\sigma}_{y,n} z_{1-\alpha/2}}{\sqrt{n}},\ \hat{\theta}_n + \frac{\hat{\sigma}_{y,n} z_{1-\alpha/2}}{\sqrt{n}} \right]$$

3.5.4 Cost/benefit ratio

The cost/benefit ratio of the importance sampling algorithm 8 need not be smaller than that of the Monte Carlo algorithm 1. In fact, it can be far larger if the importance sampling density g is chosen poorly. To see this, let's consider the cost and benefit of the importance sampling algorithm 8.

For simplicity, we assume that the time required to generate $\tilde{\mathbf{X}}_1$ from g and compute $\tilde{h}(\tilde{\mathbf{X}}_1)$ is similar to t, the time required to generate $\mathbf{X}_1$ from the distribution of $\mathbf{X}$ and compute $h(\mathbf{X})$. Under this assumption, the cost of the importance sampling algorithm 8 is approximately nt.

The importance sampling variance constant is

$$\begin{aligned} \sigma_y^2 = \operatorname{Var} \tilde{h}(\tilde{\mathbf{X}}) &= \int_{\mathbb{R}^m} \tilde{h}(\mathbf{x})^2 g(\mathbf{x}) \mathrm{d}\mathbf{x} - \theta^2 \\ &= \int_{\mathbb{R}^m} \left(\frac{h(\mathbf{x}) f(\mathbf{x})}{g(\mathbf{x})} \right)^2 g(\mathbf{x}) \mathrm{d}\mathbf{x} - \theta^2. \end{aligned}$$

The approximate $100(1-\alpha)\%$ confidence interval half-width is $\varepsilon_\alpha = \hat{\sigma}_{y,n} z_{1-\alpha/2}/\sqrt{n}$, so the benefit of the importance sampling algorithm 8 is $(z_{1-\alpha/2}/\varepsilon_\alpha)^2 = n/\sigma_y^2$. The approximate cost/benefit ratio is

$$\frac{nt}{n/\sigma_y^2} = t\sigma_y^2 = t\left(\int_{\mathbb{R}^m} \left(\frac{h(\mathbf{x})f(\mathbf{x})}{g(\mathbf{x})}\right)^2 g(\mathbf{x})\mathrm{d}\mathbf{x} - \theta^2\right).$$

In the current problem setting, the Monte Carlo variance constant is

$$\sigma^2 = \operatorname{Var} h(\mathbf{X}) = \int_{\mathbb{R}^m} h(\mathbf{x})^2 f(\mathbf{x})\mathrm{d}\mathbf{x} - \theta^2.$$

Therefore, the cost/benefit ratio reduction (or increase, if negative) of the importance sampling algorithm 8 relative to the Monte Carlo algorithm 1 is

$$\begin{aligned} t\sigma^2 - t\sigma_y^2 &= t\left(\int_{\mathbb{R}^m} h(\mathbf{x})^2 f(\mathbf{x})\mathrm{d}\mathbf{x} - \int_{\mathbb{R}^m} \left(\frac{h(\mathbf{x})f(\mathbf{x})}{g(\mathbf{x})}\right)^2 g(\mathbf{x})\mathrm{d}\mathbf{x}\right) \\ &= t\int_{\mathbb{R}^m} \left(h(\mathbf{x})^2 f(\mathbf{x}) - \frac{h(\mathbf{x})^2 f(\mathbf{x})^2}{g(\mathbf{x})}\right)\mathrm{d}\mathbf{x} \\ &= t\int_{\mathbb{R}^m} \left(1 - \frac{f(\mathbf{x})}{g(\mathbf{x})}\right) h(\mathbf{x})^2 f(\mathbf{x})\mathrm{d}\mathbf{x}. \end{aligned}$$

The importance sampling algorithm 8 improves upon the Monte Carlo algorithm 1 only if this integral is positive. But this is far from guaranteed – in fact, for some choices of g the integral can approach $-\infty$, making the importance sampling estimator useless. In the next section, we discuss heuristics for choosing g in order to (hopefully) keep this from happening.

3.5.5 Choosing an importance sampling density

To maximize the reduction in cost/benefit ratio achieved by the importance sampling algorithm 8, we want to choose the importance sampling density g in order to make the integral

$$\int_{\mathbb{R}^m} \left(1 - \frac{f(\mathbf{x})}{g(\mathbf{x})}\right) h(\mathbf{x})^2 f(\mathbf{x})\mathrm{d}\mathbf{x}$$

as large as possible.

Recall that f and g are densities, so f/g is nonnegative and $1 - f/g \leq 1$. To maximize the integral, therefore, we should choose g such that $1 - f(\mathbf{x})/g(\mathbf{x}) \approx 1$ when $h(\mathbf{x})^2 f(\mathbf{x})$ is large. This gives one rule of thumb:

Choose g such that $g(\mathbf{x}) \gg f(\mathbf{x})$ when $h(\mathbf{x})^2 f(\mathbf{x})$ is large.

Because the pdf g must normalize, making $g(\mathbf{x})$ large in some places requires making it small in others. This is risky business: If for some $\mathbf{x}$, $g(\mathbf{x})$ is close to zero but $f(\mathbf{x})$ is large, then $1 - f(\mathbf{x})/g(\mathbf{x})$ could be much less than zero. If this happens when $h(\mathbf{x})^2 f(\mathbf{x})$ is also large, then the value of the integral could be greatly reduced, or even made negative. To avoid this, we should follow another rule of thumb:

Choose g such that $g(\mathbf{x}) < f(\mathbf{x})$ only when $h(\mathbf{x})^2 f(\mathbf{x})$ is small.

What is the best possible choice of the importance sampling density g? Suppose we knew θ *a priori* and chose $g = g^\star := hf/\theta$. If $h > 0$, then $g^\star$ is nonnegative, normalizes, and has the same support as f, so $g^\star$ is an importance sampling density. With this choice of g,

$$\begin{aligned}
\sigma_y^2 &= \int_{\mathbb{R}^m} \left(\frac{h(\mathbf{x}) f(\mathbf{x})}{g^\star(\mathbf{x})} \right)^2 g^\star(\mathbf{x}) \mathrm{d}\mathbf{x} - \theta^2 \\
&= \int_{\mathbb{R}^m} \left(\frac{h(\mathbf{x}) f(\mathbf{x}) \theta}{h(\mathbf{x}) f(\mathbf{x})} \right)^2 g^\star(\mathbf{x}) \mathrm{d}\mathbf{x} - \theta^2 \\
&= \theta^2 \int_{\mathbb{R}^m} g^\star(\mathbf{x}) \mathrm{d}\mathbf{x} - \theta^2 \\
&= 0.
\end{aligned}$$

That is, $g^\star = hf/\theta$ gives an estimator with the best possible variance, zero. Of course, θ is what we set out to estimate, so it is never known *a priori*. But it does give a heuristic that can lead to good importance sampling estimators:

Choose g 'similar to' hf.

One concrete interpretation of what it means for g and hf to be 'similar' is that their maximum values coincide. This interpretation motivates the *maximum principle*:

Choose g such that $\sup\{g(\mathbf{x}) \mid \mathbf{x} \in D\} = \sup\{h(\mathbf{x}) f(\mathbf{x}) \mid \mathbf{x} \in D\}$.

The maximum principle can work well when g is selected from the same family of densities that f belongs to.

3.6 Stratification

Suppose we discover some random vector $\mathbf{Z} \in \mathbb{R}^d$ such that for any subset A of the range of $\mathbf{Z}$,

1. we can easily compute $\mathbb{P}\{\mathbf{Z} \in A\}$, and
2. we can sample from the conditional distribution of $h(\mathbf{X})$, given that $\mathbf{Z} \in A$.

Can we use this new random vector $\mathbf{Z}$ to reduce the cost/benefit ratio of the Monte Carlo algorithm 1? Stratification aims to do exactly that. The idea is to divide the range of $\mathbf{Z}$ into separate regions, generate samples of $\mathbf{Z}$ from each region, and combine the results to estimate $\theta = \mathbb{E}\, h(\mathbf{X})$.

Let's make the idea of 'dividing the range of $\mathbf{Z}$ into separate regions' precise. For $s \in \mathbb{N}$, let $A_1, \ldots, A_s$ be disjoint subsets of $\mathbb{R}^d$. We say that

$$\mathcal{P} := \{A_1, \ldots, A_s\}$$

is a *partition* if

$$p_j := \mathbb{P}\{\mathbf{Z} \in A_j\} > 0, \quad j = 1, \ldots, s,$$

and

$$\sum_{j=1}^{s} p_j = 1.$$

If $\mathcal{P}$ is a partition, then we call each A_j a *stratum* (plural *strata*). By assumption 1, we can easily compute the probabilities $p_1, \ldots, p_s$.

3.6.1 Stratum estimators

Let's define random variables $Y_1, \ldots, Y_s \in \mathbb{R}$ such that Y_j has the conditional distribution of $h(\mathbf{X})$, given that $\mathbf{Z} \in A_j$. We also define a random variable I that indicates which stratum $\mathbf{Z}$ falls in:

$$I = \begin{cases} 1 & \text{if } \mathbf{Z} \in A_1 \\ \vdots & \vdots \\ s & \text{if } \mathbf{Z} \in A_s. \end{cases}$$

Applying iterated expectation, we have

$$\theta = \mathbb{E}\, h(\mathbf{X}) = \mathbb{E}\left[\mathbb{E}\left[h(\mathbf{X}) \mid I\right]\right] = \sum_{j=1}^{s} p_j\, \mathbb{E}\left[h(\mathbf{X}) \mid I = j\right] = \sum_{j=1}^{s} p_j\, \mathbb{E}\, Y_j = \sum_{j=1}^{s} p_j \theta_j,$$

where

$$\theta_j := \mathbb{E}\, Y_j.$$

Therefore, we can estimate θ by averaging estimates of $\theta_1, \ldots, \theta_s$, weighted by $p_1, \ldots, p_s$.

Each θ_j can be estimated by choosing a sample size n_j, generating iid samples $Y_{j,1}, \ldots, Y_{j,n_j}$ from the distribution of Y_j (by assumption 2, this is doable), and computing

$$\hat{\theta}_{j,n_j} := \frac{1}{n_j} \sum_{i=1}^{n_j} Y_{j,i}.$$

The estimator $\hat{\theta}_{j,n_j}$ of θ_j is unbiased and consistent. If $\operatorname{Var} Y_j$ is finite, then for all j, $Y_{j,1}, Y_{j,2}, \ldots$ is a sequence of iid random variables with finite expectation θ_j and finite variance

$$\sigma_j^2 := \operatorname{Var} Y_j.$$

By the Central Limit Theorem, therefore,

$$\hat{\theta}_{j,n_j} \Rightarrow \mathcal{N}\left(\theta_j, \frac{\sigma_j^2}{n_j}\right) \text{ as } n_j \to \infty. \tag{3.2}$$

The variances $\sigma_1^2, \ldots, \sigma_s^2$ are typically unknown, but σ_j^2 can be estimated by

$$\hat{\sigma}_{j,n_j}^2 := \frac{1}{n_j - 1} \sum_{i=1}^{n_j} (Y_{j,i} - \hat{\theta}_{j,n_j})^2.$$

3.6.2 Overall estimators

Having estimated $\theta_1, \ldots, \theta_s$ by $\hat{\theta}_{1,n_1}, \ldots, \hat{\theta}_{s,n_s}$, we can estimate θ by

$$\hat{\theta}_n = \sum_{j=1}^{s} p_j \hat{\theta}_{j,n_j}, \tag{3.3}$$

where

$$n := n_1 + \cdots + n_s$$

is the total sample size. Combining equations (3.2) and (3.3) gives the asymptotic distribution of $\hat{\theta}_n$ as $n_1 \to \infty, \ldots, n_s \to \infty$:

$$\begin{aligned}
\hat{\theta}_n &\Rightarrow \sum_{j=1}^{s} p_j \mathcal{N}\left(\theta_j, \ \frac{\sigma_j^2}{n_j}\right) \\
&= \mathcal{N}\left(\sum_{j=1}^{s} p_j \theta_j, \ \sum_{j=1}^{s} \frac{p_j^2 \sigma_j^2}{n_j}\right) \\
&= \mathcal{N}\left(\theta, \ \frac{\sigma_{\text{strat}}^2}{n}\right).
\end{aligned}$$

We have defined the variance constant

$$\sigma_{\text{strat}}^2 := \sum_{j=1}^{s} \frac{p_j^2 \sigma_j^2}{q_j}, \tag{3.4}$$

which depends on both the partition $\mathcal{P}$ and the vector

$$\mathbf{q} := \frac{1}{n} \begin{bmatrix} n_1 \\ \vdots \\ n_s \end{bmatrix}$$

of stratum sample fractions. When constructing confidence intervals, we estimate the unknown variance constant σ_{strat}^2 by

$$\hat{\sigma}_{\text{strat},n}^2 := \sum_{j=1}^{s} \frac{p_j^2 \hat{\sigma}_{j,n_j}^2}{q_j}.$$

3.6.3 The stratification algorithm

Algorithm 9 (Stratification estimation of $\theta = \mathbb{E}\, h(\mathbf{X})$). Given confidence level $\alpha \in (0,1)$, stratum sample sizes $n_1, \ldots, n_s \in \mathbb{N}$, and stratum probabilities $p_1, \ldots, p_s$:

1. for $j = 1, \ldots, s$
   (a) generate $Y_{j,1}, \ldots, Y_{j,n_j}$ iid from the conditional distribution of $h(\mathbf{X})$, given $\mathbf{Z} \in A_j$
   (b) estimate $\theta_j = \mathbb{E}\, Y_j$ and $\sigma_j^2 = \operatorname{Var} Y_j$ by

$$\hat{\theta}_{j,n_j} = \frac{1}{n_j} \sum_{i=1}^{n_j} Y_{j,i}, \quad \hat{\sigma}^2_{j,n_j} = \frac{1}{n_j - 1} \sum_{i=1}^{n_j} (Y_{j,i} - \hat{\theta}_{j,n_j})^2$$

2. estimate $\theta = \mathbb{E}\, h(\mathbf{X})$ and $\sigma^2_{\text{strat}} = n \sum_{j=1}^{s} p_j^2 \sigma_j^2 / n_j$ by

$$\hat{\theta}_n = \sum_{j=1}^{s} p_j \hat{\theta}_{j,n_j}, \quad \hat{\sigma}^2_{\text{strat},n} = n \sum_{j=1}^{s} \frac{p_j^2 \hat{\sigma}^2_{j,n_j}}{n_j}$$

3. conclude with approximately $100(1-\alpha)\%$ confidence that

$$\theta \in \left[\hat{\theta}_n - \frac{\hat{\sigma}_{\text{strat},n} z_{1-\alpha/2}}{\sqrt{n}},\ \hat{\theta}_n + \frac{\hat{\sigma}_{\text{strat},n} z_{1-\alpha/2}}{\sqrt{n}} \right]$$

3.6.4 Cost/benefit ratio

The computational effort required to compute the stratification estimator $\hat{\theta}_n$ is typically dominated by the time required to generate $Y_{j,1}, \ldots, Y_{j,n_j}$ for each $j = 1, \ldots, s$. Therefore, the cost of the stratification algorithm 9 is approximately

$$\sum_{j=1}^{s} n_j t_j,$$

where t_j is the time required to generate $Y_{j,1}$ from the conditional distribution of $h(\mathbf{X})$, given $\mathbf{Z} \in A_j$.

The stratification $100(1-\alpha)\%$ confidence interval half-width is $\varepsilon_\alpha = \sigma_{\text{strat}} z_{1-\alpha/2} / \sqrt{n}$, giving a benefit of $(z_{1-\alpha/2}/\varepsilon_\alpha)^2 = n/\sigma^2_{\text{strat}}$. Therefore, the approximate cost/benefit ratio of the stratification algorithm 9 is

$$\frac{\sum_{j=1}^{s} n_j t_j}{n/\sigma^2_{\text{strat}}} = \sigma^2_{\text{strat}} \sum_{j=1}^{s} q_j t_j.$$

Recalling that the cost/benefit ratio of the Monte Carlo algorithm 1 is $t\sigma^2$, the cost/benefit ratio reduction (or increase, if negative) from stratification is

$$t\sigma^2 - \sigma^2_{\text{strat}} \sum_{j=1}^{s} q_j t_j.$$

The reduction may not be positive: Although it is not difficult to ensure that $\sigma^2_{\text{strat}} \leq \sigma^2$ (see §3.6.6), it is not always the case that $\sum_{j=1}^{s} q_j t_j \leq t$. In some cases, sampling from the conditional distribution of $h(\mathbf{X})$, given $\mathbf{Z} \in A_j$, takes considerably longer than sampling directly from the distribution of $\mathbf{X}$ and evaluating h.

Given a total sample size n, we have two degrees of freedom in implementing the stratification algorithm 9:

1. the partition $\mathcal{P} = \{A_1, \ldots, A_s\}$, and
2. the stratum sample fractions $\mathbf{q} = (q_1, \ldots, q_s)$.

The next two sections discuss methods for choosing $\mathcal{P}$ and $\mathbf{q}$ in order to minimize the cost/benefit ratio of the stratification algorithm 9.

3.6.5 Choosing a partition

In order for the stratification algorithm 9 to be implementable, the partition $\mathcal{P}$ must be chosen such that $\mathbb{P}\{\mathbf{Z} \in A_j\}$ can be computed for all j. Assuming $\mathcal{P}$ is admissible in this sense, we are free to choose both the number of strata s, and the 'shape' of the strata (by choice of the A_j). These choices influence both the approximate stratification runtime $\sum_{j=1}^{s} q_j t_j$ and the variance constant σ^2_{strat}.

In order to reduce runtime, $\mathcal{P}$ should be chosen so that for all j, the time required to sample from the conditional distribution of $h(\mathbf{X})$, given $\mathbf{Z} \in A_j$, is not too much larger than the time required to sample from the distribution of $\mathbf{X}$ directly and evaluate h.

In order to reduce variance (for fixed $\mathbf{q}$), $\mathcal{P}$ should be chosen such that the stratum probabilities p_j and variances σ_j^2 make

$$\sigma^2_{\text{strat}} = \sum_{j=1}^{s} \frac{p_j^2 \sigma_j^2}{q_j}$$

small. Unfortunately, the variances σ_j^2 are generally not known *a priori*. A good rule of thumb is to choose $\mathcal{P}$ such that the stratum means θ_j vary strongly with j, but the variance σ_j^2 within any particular stratum is small.

3.6.6 Allocating samples

Once the partition $\mathcal{P}$ is fixed, we are free to choose any sample size fractions $q_1, \ldots, q_s$ that satisfy

$$q_j \geq 0, \quad \sum_{j=1}^{s} q_j = 1.$$

This choice influences both the stratification runtime $\sum_{j=1}^{s} q_j t_j$ and the variance constant σ^2_{strat}.

Suppose t_j, the time required to generate one sample from the conditional distribution of $h(\mathbf{X})$, given $\mathbf{Z} \in A_j$, is disproportionately large for some j. Then we can reduce runtime by taking fewer samples from that stratum, *i.e.*, by making that q_j small.

To reduce the variance constant σ^2_{strat}, we can choose $\mathbf{q}$ such that q_j is large whenever $p_j^2\sigma_j^2$ is large. It can be shown that (for fixed $\mathcal{P}$) the sample size fractions that minimize σ^2_{strat} are

$$q_j^\star := \frac{p_j\sigma_j^2}{p_1\sigma_1^2 + \cdots + p_s\sigma_s^2}, \quad j = 1, \ldots, s.$$

Unfortunately, the variances σ_j^2 are generally not known *a priori*. One way to handle this is to run some initial pilot simulations with modest sample sizes to estimate the σ_j^2. Another way is to use a suboptimal allocation heuristic called *proportional sampling*.

Proportional sampling

In proportional sampling, we set

$$q_j = q_{j,\text{prop}} := p_j$$

for all j. Proportional sampling is easy to implement. It also guarantees that the variance constant σ^2_{prop} is no larger than σ^2. To see this, observe that

$$\sigma^2_{\text{prop}} = \sum_{j=1}^{s} \frac{p_j^2\sigma_j^2}{q_{j,\text{prop}}} = \sum_{j=1}^{s} p_j\sigma_j^2.$$

How does σ^2_{prop} compare to the variance constant σ^2 in the Monte Carlo algorithm 1? By iterated variance,

$$\sigma^2 = \operatorname{Var} h(\mathbf{X}) = \mathbb{E} \operatorname{Var} \left(h(\mathbf{X}) \mid I\right) + \operatorname{Var} \mathbb{E} \left[h(\mathbf{X}) \mid I\right].$$

But $\operatorname{Var} \mathbb{E} \left[h(\mathbf{X}) \mid I\right] \geq 0$ (all variances are nonnegative), so

$$\begin{aligned}
\sigma^2 &\geq \mathbb{E} \operatorname{Var} \left(h(\mathbf{X}) \mid I\right) \\
&= \sum_{j=1}^{s} p_j \operatorname{Var} \left(h(\mathbf{X}) \mid I = j\right) \\
&= \sum_{j=1}^{s} p_j\sigma_j^2 \\
&= \sigma^2_{\text{prop}}.
\end{aligned}$$

3.6.7 Post-stratification

We conclude our discussion of stratification with a variant called post-stratification, also known as *something for nothing*. The basic idea is that in some situations, sampling from the conditional distribution of $h(\mathbf{X})$, given that $\mathbf{Z} \in A_j$, is hard. In these situations, we can still get a variance reduction by estimating θ using the Monte Carlo algorithm 1, then applying stratification after the fact.

How is this possible? First, we generate iid realizations $(\mathbf{X}_1, \mathbf{Z}_1), \ldots, (\mathbf{X}_n, \mathbf{Z}_n)$ from the joint distribution of $(\mathbf{X}, \mathbf{Z})$. We then compute the number[1] of realizations of $\mathbf{Z}$ in each stratum:

$$N_j := \sum_{i=1}^{n} \mathcal{I}_{A_j}(\mathbf{Z}_i),$$

where $\mathcal{I}_{A_j}$ is the indicator function of the stratrum A_j. We can then estimate the stratum mean $\theta_j = \mathbb{E}\left[h(\mathbf{X}) \mid \mathbf{Z} \in A_j\right]$ by

$$\hat{\theta}_{j,N_j} := \frac{1}{N_j} \sum_{i=1}^{n} h(\mathbf{X}_i)\mathcal{I}_{A_j}(\mathbf{Z}_i).$$

Similarly, we estimate the stratum variance σ_j^2 by

$$\hat{\sigma}_{j,N_j} := \frac{1}{N_j - 1} \sum_{i=1}^{n} (h(\mathbf{X}_i)\mathcal{I}_{A_j}(\mathbf{Z}_i) - \hat{\theta}_{j,N_j})^2.$$

The post-stratification estimator of θ is

$$\hat{\theta}_{\text{post},n} := \sum_{j=1}^{s} p_j \hat{\theta}_{j,N_j}.$$

An estimator of the post-stratification variance constant is

$$\hat{\sigma}^2_{\text{post},n} := n \sum_{j=1}^{s} \frac{p_j^2 \hat{\sigma}^2_{j,N_j}}{N_j}.$$

Amazingly, the post-stratification variance constant is asymptotically equal to the proportional stratification variance constant. When sampling from the conditional distribution of $h(\mathbf{X})$, given $\mathbf{Z} \in A_j$, is slow for some j, the post-stratification algorithm 10 can have a lower cost/benefit ratio than the stratification algorithm 9 (for the same partition, sample size, and sample allocations).

[1] Note that $N_1, \ldots, N_s$ are random variables.

3.6.8 The post-stratification algorithm

Algorithm 10 (Post-stratification estimation of $\theta = \mathbb{E}\, h(\mathbf{X})$). Given confidence level $\alpha \in (0,1)$, stratum probabilities $p_1, \ldots, p_s$, and sample size $n \in \mathbb{N}$:

1. generate $(\mathbf{X}_1, \mathbf{Z}_1), \ldots, (\mathbf{X}_n, \mathbf{Z}_n)$ iid from the joint distribution of $(\mathbf{X}, \mathbf{Z})$
2. for $j = 1, \ldots, s$
   (a) compute the number of $\mathbf{Z}_i$ in stratum A_j:
   $$N_j = \sum_{i=1}^{n} \mathcal{I}_{A_j}(\mathbf{Z}_i)$$
   (b) estimate $\theta_j = \mathbb{E}\left[h(\mathbf{X}) \mid \mathbf{Z} \in A_j\right]$ and $\sigma_j^2 = \mathrm{Var}\left(h(\mathbf{X}) \mid \mathbf{Z} \in A_j\right)$ by
   $$\hat{\theta}_{j,N_j} = \frac{1}{N_j} \sum_{i=1}^{n} h(\mathbf{X}_i)\mathcal{I}_{A_j}(\mathbf{Z}_i), \quad \hat{\sigma}_{j,N_j} = \frac{1}{N_j - 1} \sum_{i=1}^{n} (h(\mathbf{X}_i)\mathcal{I}_{A_j}(\mathbf{Z}_i) - \hat{\theta}_{j,N_j})^2$$
3. estimate $\theta = \mathbb{E}\, h(\mathbf{X})$ and σ^2_{post} by
   $$\hat{\theta}_{\text{post},n} = \sum_{j=1}^{s} p_j \hat{\theta}_{j,N_j}, \quad \hat{\sigma}^2_{\text{post},n} = n \sum_{j=1}^{s} \frac{p_j^2 \hat{\sigma}^2_{j,N_j}}{N_j}$$
4. conclude with approximately $100(1-\alpha)\%$ confidence that
   $$\theta \in \left[\hat{\theta}_{\text{post},n} - \frac{\hat{\sigma}_{\text{post},n} z_{1-\alpha/2}}{\sqrt{n}},\ \hat{\theta}_{\text{post},n} + \frac{\hat{\sigma}_{\text{post},n} z_{1-\alpha/2}}{\sqrt{n}}\right]$$

Chapter 4

Gradient estimation

In this chapter, we extend the basic problem in §1.2 to allow θ to depend on a deterministic parameter vector $\mathbf{u} \in \mathbb{R}^d$. This dependence may be *structural* (meaning the function h depends on $\mathbf{u}$), *distributional* (meaning the distribution of $\mathbf{X}$ depends on $\mathbf{u}$), or both:

$$\theta(\mathbf{u}) := \mathbb{E}^{\mathbf{u}}\, h(\mathbf{X}, \mathbf{u}). \tag{4.1}$$

Here the notation $\mathbb{E}^{\mathbf{u}}$ emphasizes the dependence of the distribution of $\mathbf{X}$ on $\mathbf{u}$. To make this notation explicit, suppose $\mathbf{X}$ is continuous. In this case, the pdf f of $\mathbf{X}$ depends on $\mathbf{u}$, so the expectation is

$$\mathbb{E}^{\mathbf{u}}\, h(\mathbf{X}, \mathbf{u}) = \int_{\mathbb{R}^m} h(\mathbf{x}, \mathbf{u}) f(\mathbf{x}, \mathbf{u}) \mathrm{d}\mathbf{x}.$$

When θ depends on $\mathbf{u}$, a natural question to ask is how θ responds to small perturbations in $\mathbf{u}$ about its nominal value. For example, suppose $u_1, \ldots, u_d$ represent the generation levels of d power plants, $X_1, \ldots, X_m$ represent the (random) demand for electricity at m nodes in the power network, and $\theta(\mathbf{u})$ represents the probability of a power outage. The grid operator would no doubt be interested to know that, say, θ increases wildly in response to a small decrease in one of the u_j.

The study of how θ changes with $\mathbf{u}$ is called *sensitivity analysis*. It leads directly to the topic of this chapter: Estimating the gradient of θ at some point $\mathbf{u}_0 \in \mathbb{R}^d$,

$$\mathbf{g}(\mathbf{u}_0) := \nabla \theta(\mathbf{u})|_{\mathbf{u}_0} = \begin{bmatrix} \frac{\partial \theta}{\partial u_1}\Big|_{\mathbf{u}_0} \\ \vdots \\ \frac{\partial \theta}{\partial u_d}\Big|_{\mathbf{u}_0} \end{bmatrix}.$$

Gradient estimation is also used in numerical algorithms for choosing $\mathbf{u}$ in order to minimize or maximize $\theta(\mathbf{u})$. This field is called *simulation-based optimization*.

Throughout this chapter, we assume that for $j = 1, \ldots, d$,

1. θ is differentiable with respect u_j at $\mathbf{u}_0$, and
2. $\operatorname{Var} \partial h / \partial u_j|_{\mathbf{u}_0}$ is finite in a neighborhood of $\mathbf{u}_0$.

4.1 Finite difference approximation

4.1.1 Forward difference

Recall that the partial derivative of θ with respect to u_j is defined by

$$g_j(\mathbf{u}_0) = \left.\frac{\partial \theta}{\partial u_j}\right|_{\mathbf{u}_0} := \lim_{\varepsilon \to 0} \frac{\theta(\mathbf{u}_0 + \varepsilon \mathbf{e}_j) - \theta(\mathbf{u}_0)}{\varepsilon},$$

where $\mathbf{e}_j$ is the j^{th} standard basis vector in $\mathbb{R}^d$. Truncating the limit at some small but finite ε gives the forward difference approximation of $g_j(\mathbf{u}_0)$:

$$\Delta_{\text{f}}(j, \mathbf{u}_0, \varepsilon) := \frac{\theta(\mathbf{u}_0 + \varepsilon \mathbf{e}_j) - \theta(\mathbf{u}_0)}{\varepsilon}.$$

We can't evaluate θ exactly, so the forward difference approximation is not computable. However, we can estimate it using Monte Carlo simulation. To do this, we generate $\mathbf{X}_1, \ldots, \mathbf{X}_n$ iid from the distribution of $\mathbf{X}$, then compute

$$\hat{\Delta}_{\text{f},n}(j, \mathbf{u}_0, \varepsilon) := \frac{\bar{\theta}_n(\mathbf{u}_0 + \varepsilon \mathbf{e}_j) - \bar{\theta}_n(\mathbf{u}_0)}{\varepsilon},$$

where

$$\bar{\theta}_n(\mathbf{u}) := \frac{1}{n} \sum_{i=1}^{n} h(\mathbf{X}_i, \mathbf{u})$$

is the usual Monte Carlo estimator of $\theta(\mathbf{u})$.

Bias-Variance trade-off

There are two sources of error in the forward difference estimator. The first is the truncation of the limit, which introduces a bias. The second is the approximation of θ by $\bar{\theta}_n$, which introduces some variance. Both the bias and variance depend on the perturbation ε. We now explore the trade-off between the two.

Taylor expanding θ about $\mathbf{u}_0$, we have

$$\begin{aligned}
\mathbb{E}\, \hat{\Delta}_{\text{f},n}(j, \mathbf{u}_0, \varepsilon) &= \Delta_{\text{f}}(j, \mathbf{u}_0, \varepsilon) \\
&= \frac{\theta(\mathbf{u}_0 + \varepsilon \mathbf{e}_j) - \theta(\mathbf{u}_0)}{\varepsilon} \\
&= \frac{1}{\varepsilon} \left(\left(\theta(\mathbf{u}_0) + \frac{\varepsilon}{1!} \left.\frac{\partial \theta}{\partial u_j}\right|_{\mathbf{u}_0} + \frac{\varepsilon^2}{2!} \left.\frac{\partial^2 \theta}{\partial u_j^2}\right|_{\mathbf{u}_0} + \ldots \right) - \theta(\mathbf{u}_0) \right) \\
&= g_j(\mathbf{u}_0) + O(\varepsilon)
\end{aligned}$$

as $\varepsilon \to 0$. To make the bias small, therefore, we want to make ε small. However, the variance of the forward difference estimator is

$$\begin{aligned}
\operatorname{Var} \hat{\Delta}_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon) &= \frac{1}{\varepsilon^2} \operatorname{Var} \left(\bar{\theta}_n(\mathbf{u}_0 + \varepsilon \mathbf{e}_j) - \bar{\theta}_n(\mathbf{u}_0) \right) \\
&= \frac{1}{\varepsilon^2} \left(\operatorname{Var} \bar{\theta}_n(\mathbf{u}_0 + \varepsilon \mathbf{e}_j) + \operatorname{Var} \bar{\theta}_n(\mathbf{u}_0) - 2\operatorname{Cov}(\bar{\theta}_n(\mathbf{u}_0 + \varepsilon \mathbf{e}_j), \bar{\theta}_n(\mathbf{u}_0)) \right) \\
&\approx \frac{1}{\varepsilon^2} \left(2 \operatorname{Var} \bar{\theta}_n(\mathbf{u}_0) - 2 \operatorname{Cov}(\bar{\theta}_n(\mathbf{u}_0 + \varepsilon \mathbf{e}_j), \bar{\theta}_n(\mathbf{u}_0)) \right),
\end{aligned}$$

provided $\operatorname{Var} \bar{\theta}_n(\cdot)$ is continuous at $\mathbf{u}_0$. If $\bar{\theta}_n(\mathbf{u}_0 + \varepsilon \mathbf{e}_j)$ and $\bar{\theta}_n(\mathbf{u}_0)$ are computed using independent samples, then the covariance term vanishes and

$$\operatorname{Var} \hat{\Delta}_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon) \approx \frac{2 \operatorname{Var} h(\mathbf{X}, \mathbf{u})|_{\mathbf{u}_0}}{n\varepsilon^2} = O(1/\varepsilon^2)$$

as $\varepsilon \to 0$. To make the variance small, therefore, we want to make ε large.

The bias-variance trade-off appears explicitly in the mean squared error of the forward difference estimator:

$$\operatorname{MSE} \hat{\Delta}_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon) = \underbrace{\operatorname{Bias}(\hat{\Delta}_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon))^2}_{O(\varepsilon^2)} + \underbrace{\operatorname{Var} \hat{\Delta}_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon)}_{O(1/\varepsilon^2)}.$$

The bias-variance trade-off can be alleviated by computing $\bar{\theta}_n(\mathbf{u}_0 + \varepsilon \mathbf{e}_j)$ and $\bar{\theta}_n(\mathbf{u}_0)$ using the *same* samples $\mathbf{X}_1, \ldots, \mathbf{X}_n$ (*i.e.*, using common random numbers). This is because for small ε, we expect $h(\mathbf{X}, \mathbf{u}_0 + \varepsilon \mathbf{e}_j)$ to be positively correlated with $h(\mathbf{X}, \mathbf{u}_0)$, making $\operatorname{Cov}(\bar{\theta}_n(\mathbf{u}_0 + \varepsilon \mathbf{e}_j), \bar{\theta}_n(\mathbf{u}_0))$ large.

Asymptotic distribution

We can write the forward difference estimator as

$$\begin{aligned}
\hat{\Delta}_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon) &= \frac{\bar{\theta}_n(\mathbf{u}_0 + \varepsilon \mathbf{e}_j) - \bar{\theta}_n(\mathbf{u}_0)}{\varepsilon} \\
&= \frac{1}{\varepsilon} \left(\frac{1}{n} \sum_{i=1}^{n} h(\mathbf{X}_i, \mathbf{u}_0 + \varepsilon \mathbf{e}_j) - \frac{1}{n} \sum_{i=1}^{n} h(\mathbf{X}_i, \mathbf{u}_0) \right) \\
&= \frac{1}{n} \sum_{i=1}^{n} Z_{\mathrm{f},i}(j, \mathbf{u}_0, \varepsilon),
\end{aligned}$$

where

$$Z_{\mathrm{f},i}(j, \mathbf{u}, \varepsilon) := \frac{h(\mathbf{X}_i, \mathbf{u} + \varepsilon \mathbf{e}_j) - h(\mathbf{X}_i, \mathbf{u})}{\varepsilon}.$$

Because $\mathbf{X}_1, \mathbf{X}_2, \ldots$ are iid, $Z_{\mathrm{f},1}(j, \mathbf{u}_0, \varepsilon), Z_{\mathrm{f},2}(j, \mathbf{u}_0, \varepsilon), \ldots$ is a sequence of iid random variables with expectation $g_j(\mathbf{u}_0) + O(\varepsilon)$ and finite variance. By the Central Limit and Converging

Together theorems, therefore, $\hat{\Delta}_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon)$ is asymptotically distributed according to

$$\frac{\sqrt{n}\left(\hat{\Delta}_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon) - g_j(\mathbf{u}_0) + O(\varepsilon)\right)}{\hat{\sigma}_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon)} \Rightarrow \mathcal{N}(0, 1) \text{ as } n \to \infty,$$

where

$$\hat{\sigma}^2_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon) := \frac{1}{n-1} \sum_{i=1}^{n} \left(Z_{\mathrm{f},i}(j, \mathbf{u}_0, \varepsilon) - \hat{\Delta}_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon)\right)^2$$

is an estimator of the variance of $Z_{\mathrm{f}}(j, \mathbf{u}_0, \varepsilon) := (h(\mathbf{X}, \mathbf{u}_0 + \varepsilon \mathbf{e}_j) - h(\mathbf{X}, \mathbf{u}_0))/\varepsilon$. This asymptotic distribution lets us compute confidence intervals for $g_j(\mathbf{u}_0)$ (up to an unknown $O(\varepsilon)$ bias).

The forward difference algorithm

Algorithm 11 (Forward difference estimation of $\mathbf{g}(\mathbf{u}) = \nabla \mathbb{E}^{\mathbf{u}} h(\mathbf{X}, \mathbf{u})$ at $\mathbf{u}_0$ using common random numbers). Given confidence level $\alpha \in (0,1)$, sample size $n \in \mathbb{N}$, and perturbations $\varepsilon_1, \ldots, \varepsilon_d$:

1. generate $\mathbf{X}_1, \ldots, \mathbf{X}_n$ iid from the distribution of $\mathbf{X}$
2. compute and store $h(\mathbf{X}_1, \mathbf{u}_0), \ldots, h(\mathbf{X}_n, \mathbf{u}_0)$
3. for $j = 1, \ldots, d$
   (a) for $i = 1, \ldots, n$, compute
   $$Z_{\mathrm{f},i}(j, \mathbf{u}_0, \varepsilon_j) = \frac{h(\mathbf{X}_i, \mathbf{u}_0 + \varepsilon_j \mathbf{e}_j) - h(\mathbf{X}_i, \mathbf{u}_0)}{\varepsilon_j}$$
   (b) estimate $g_j(\mathbf{u}_0)$ by
   $$\hat{\Delta}_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon_j) = \frac{1}{n} \sum_{i=1}^{n} Z_{\mathrm{f},i}(j, \mathbf{u}_0, \varepsilon_j)$$
   (c) estimate $\operatorname{Var} Z_{\mathrm{f}}(j, \mathbf{u}_0, \varepsilon_j)$ by
   $$\hat{\sigma}^2_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon_j) = \frac{1}{n-1} \sum_{i=1}^{n} \Big(Z_{\mathrm{f},i}(j, \mathbf{u}_0, \varepsilon_j) - \hat{\Delta}_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon_j) \Big)^2$$
   (d) conclude with approximately $100(1-\alpha)\%$ confidence that
   $$g_j(\mathbf{u}_0) + O(\varepsilon_j) \in \left[\hat{\Delta}_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon_j) - \frac{\hat{\sigma}_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon_j) z_{1-\alpha/2}}{\sqrt{n}},\ \hat{\Delta}_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon_j) + \frac{\hat{\sigma}_{\mathrm{f},n}(j, \mathbf{u}_0, \varepsilon_j) z_{1-\alpha/2}}{\sqrt{n}} \right]$$

4.1.2 Central difference

An equivalent expression for the partial derivative of θ with respect to u_j, evaluated at $\mathbf{u}_0 \in \mathbb{R}^d$, is

$$g_j(\mathbf{u}_0) = \lim_{\varepsilon \to 0} \frac{\theta(\mathbf{u}_0 + (\varepsilon/2)\mathbf{e}_j) - \theta(\mathbf{u}_0 - (\varepsilon/2)\mathbf{e}_j)}{\varepsilon}.$$

This gives rise to the central difference approximation of $g_j(\mathbf{u}_0)$:

$$\Delta_{\mathrm{c}}(j, \mathbf{u}_0, \varepsilon) := \frac{\theta(\mathbf{u}_0 + (\varepsilon/2)\mathbf{e}_j) - \theta(\mathbf{u}_0 - (\varepsilon/2)\mathbf{e}_j)}{\varepsilon}.$$

As with forward differences, we use Monte Carlo simulation to estimate $\Delta_{\mathrm{c}}(j, \mathbf{u}_0, \varepsilon)$ by

$$\hat{\Delta}_{\mathrm{c},n}(j, \mathbf{u}_0, \varepsilon) = \frac{\bar{\theta}_n(\mathbf{u}_0 + (\varepsilon/2)\mathbf{e}_j) - \bar{\theta}_n(\mathbf{u}_0 - (\varepsilon/2)\mathbf{e}_j)}{\varepsilon}.$$

Bias reduction

The main advantage of central difference approximation is that it reduces bias relative to forward difference approximation. To see this, we again Taylor expanding θ about $\mathbf{u}_0$:

$$\begin{aligned}
\mathbb{E}\,\hat{\Delta}_{\mathrm{c},n}(j, \mathbf{u}_0, \varepsilon) &= \Delta_{\mathrm{c}}(j, \mathbf{u}_0, \varepsilon) \\
&= \frac{\theta(\mathbf{u}_0 + (\varepsilon/2)\mathbf{e}_j) - \theta(\mathbf{u}_0 - (\varepsilon/2)\mathbf{e}_j)}{\varepsilon} \\
&= \frac{1}{\varepsilon}\Bigg(\left(\theta(\mathbf{u}_0) + \frac{\varepsilon/2}{1!}\frac{\partial\theta}{\partial u_j}\bigg|_{\mathbf{u}_0} + \frac{(\varepsilon/2)^2}{2!}\frac{\partial^2\theta}{\partial u_j^2}\bigg|_{\mathbf{u}_0} + \frac{(\varepsilon/2)^3}{3!}\frac{\partial^3\theta}{\partial u_j^3}\bigg|_{\mathbf{u}_0} + \dots \right) \\
&\quad - \left(\theta(\mathbf{u}_0) + \frac{-\varepsilon/2}{1!}\frac{\partial\theta}{\partial u_j}\bigg|_{\mathbf{u}_0} + \frac{(-\varepsilon/2)^2}{2!}\frac{\partial^2\theta}{\partial u_j^2}\bigg|_{\mathbf{u}_0} + \frac{(-\varepsilon/2)^3}{3!}\frac{\partial^3\theta}{\partial u_j^3}\bigg|_{\mathbf{u}_0} + \dots \right) \Bigg) \\
&= \frac{1}{\varepsilon}\left(\varepsilon\frac{\partial\theta}{\partial u_j}\bigg|_{\mathbf{u}_0} + \frac{\varepsilon^3}{24}\frac{\partial^3\theta}{\partial u_j^3}\bigg|_{\mathbf{u}_0} + \dots \right) \\
&= g_j(\mathbf{u}_0) + O(\varepsilon^2).
\end{aligned}$$

As $\varepsilon \to 0$, the $O(\varepsilon^2)$ central difference bias vanishes much faster than the $O(\varepsilon)$ forward difference bias.

The variance of the central difference estimator is

$$\begin{aligned}
\operatorname{Var}\hat{\Delta}_{\mathrm{c},n}(j, \mathbf{u}_0, \varepsilon) &= \frac{1}{\varepsilon^2}\operatorname{Var}\left(\bar{\theta}_n(\mathbf{u}_0 + (\varepsilon/2)\mathbf{e}_j) - \bar{\theta}_n(\mathbf{u}_0 - (\varepsilon/2)\mathbf{e}_j)\right) \\
&\approx \frac{1}{\varepsilon^2}\left(2\operatorname{Var}\bar{\theta}_n(\mathbf{u}_0) - 2\operatorname{Cov}(\bar{\theta}_n(\mathbf{u}_0 + (\varepsilon/2)\mathbf{e}_j), \bar{\theta}_n(\mathbf{u}_0 - (\varepsilon/2)\mathbf{e}_j))\right),
\end{aligned}$$

provided $\operatorname{Var}\bar{\theta}_n(\cdot)$ is continuous at $\mathbf{u}_0$. If we compute $\bar{\theta}_n(\mathbf{u}_0 + (\varepsilon/2)\mathbf{e}_j)$ and $\bar{\theta}_n(\mathbf{u}_0 - (\varepsilon/2)\mathbf{e}_j)$ using common random numbers, then $h(\mathbf{X}, \mathbf{u}_0 + (\varepsilon/2)\mathbf{e}_j)$ should be positively correlated with $h(\mathbf{X}, \mathbf{u}_0 - (\varepsilon/2)\mathbf{e}_j)$, making $\operatorname{Cov}(\bar{\theta}_n(\mathbf{u}_0 + (\varepsilon/2)\mathbf{e}_j), \bar{\theta}_n(\mathbf{u}_0 - (\varepsilon/2)\mathbf{e}_j))$ large.

Asymptotic distribution

We can write the central difference estimator as

$$\begin{aligned}
\hat{\Delta}_{\mathrm{c},n}(j, \mathbf{u}_0, \varepsilon) &= \frac{\bar{\theta}_n(\mathbf{u}_0 + (\varepsilon/2)\mathbf{e}_j) - \bar{\theta}_n(\mathbf{u}_0 - (\varepsilon/2)\mathbf{e}_j)}{\varepsilon} \\
&= \frac{1}{\varepsilon}\left(\frac{1}{n}\sum_{i=1}^{n} h(\mathbf{X}_i, \mathbf{u}_0 + (\varepsilon/2)\mathbf{e}_j) - \frac{1}{n}\sum_{i=1}^{n} h(\mathbf{X}_i, \mathbf{u}_0 - (\varepsilon/2)\mathbf{e}_j) \right) \\
&= \frac{1}{n}\sum_{i=1}^{n} Z_{\mathrm{c},i}(j, \mathbf{u}_0, \varepsilon),
\end{aligned}$$

where

$$Z_{\mathrm{c},i}(j, \mathbf{u}, \varepsilon) := \frac{h(\mathbf{X}_i, \mathbf{u} + (\varepsilon/2)\mathbf{e}_j) - h(\mathbf{X}_i, \mathbf{u} - (\varepsilon/2)\mathbf{e}_j)}{\varepsilon}.$$

Because $\mathbf{X}_1, \mathbf{X}_2, \ldots$ are iid, $Z_{\mathrm{c},1}(j, \mathbf{u}_0, \varepsilon), Z_{\mathrm{c},2}(j, \mathbf{u}_0, \varepsilon), \ldots$ is a sequence of iid random variables with expectation $g_j(\mathbf{u}_0) + O(\varepsilon^2)$ and finite variance. By the Central Limit and Converging Together theorems, therefore, $\hat{\Delta}_{\mathrm{c},n}(j, \mathbf{u}_0, \varepsilon)$ is asymptotically distributed according to

$$\frac{\sqrt{n}\left(\hat{\Delta}_{\mathrm{c},n}(j, \mathbf{u}_0, \varepsilon) - g_j(\mathbf{u}_0) + O(\varepsilon^2)\right)}{\hat{\sigma}_{\mathrm{c},n}(j, \mathbf{u}_0, \varepsilon)} \Rightarrow \mathcal{N}(0, 1) \text{ as } n \to \infty,$$

where

$$\hat{\sigma}^2_{\mathrm{c},n}(j, \mathbf{u}_0, \varepsilon) := \frac{1}{n-1} \sum_{i=1}^{n} \left(Z_{\mathrm{c},i}(j, \mathbf{u}_0, \varepsilon) - \hat{\Delta}_{\mathrm{c},n}(j, \mathbf{u}_0, \varepsilon)\right)^2$$

is an estimator of the variance of $Z_{\mathrm{c}}(j, \mathbf{u}_0, \varepsilon) := (h(\mathbf{X}, \mathbf{u}_0 + (\varepsilon/2)\mathbf{e}_j) - h(\mathbf{X}, \mathbf{u}_0 - (\varepsilon/2)\mathbf{e}_j))/\varepsilon$. This asymptotic distribution lets us compute confidence intervals for $g_j(\mathbf{u}_0)$ (up to an unknown $O(\varepsilon^2)$ bias).

The central difference algorithm

Algorithm 12 (Central difference estimation of $\mathbf{g}(\mathbf{u}) = \nabla \mathbb{E}^{\mathbf{u}} h(\mathbf{X}, \mathbf{u})$ at $\mathbf{u}_0$ using common random numbers). Given confidence level $\alpha \in (0,1)$, sample size $n \in \mathbb{N}$, and perturbations $\varepsilon_1, \ldots, \varepsilon_d$:

1. generate $\mathbf{X}_1, \ldots, \mathbf{X}_n$ iid from the distribution of $\mathbf{X}$
2. for $j = 1, \ldots, d$
   (a) for $i = 1, \ldots, n$, compute

$$Z_{\mathrm{c},i}(j, \mathbf{u}_0, \varepsilon_j) = \frac{h(\mathbf{X}_i, \mathbf{u}_0 + (\varepsilon_j/2)\mathbf{e}_j) - h(\mathbf{X}_i, \mathbf{u}_0 - (\varepsilon_j/2)\mathbf{e}_j)}{\varepsilon_j}$$

   (b) estimate $g_j(\mathbf{u}_0)$ by

$$\hat{\Delta}_{\mathrm{c},n}(j, \mathbf{u}_0, \varepsilon_j) = \frac{1}{n} \sum_{i=1}^{n} Z_{\mathrm{c},i}(j, \mathbf{u}_0, \varepsilon_j)$$

   (c) estimate $\operatorname{Var} Z_{\mathrm{c}}(j, \mathbf{u}_0, \varepsilon_j)$ by

$$\hat{\sigma}^2_{\mathrm{c},n}(j, \mathbf{u}_0, \varepsilon_j) = \frac{1}{n-1} \sum_{i=1}^{n} \left(Z_{\mathrm{c},i}(j, \mathbf{u}_0, \varepsilon_j) - \hat{\Delta}_{\mathrm{c},n}(j, \mathbf{u}_0, \varepsilon_j) \right)^2$$

   (d) conclude with approximately $100(1-\alpha)\%$ confidence that

$$g_j(\mathbf{u}_0) + O(\varepsilon_j^2) \in \left[\hat{\Delta}_{\mathrm{c},n}(j, \mathbf{u}_0, \varepsilon_j) - \frac{\hat{\sigma}_{\mathrm{c},n}(j, \mathbf{u}_0, \varepsilon_j) z_{1-\alpha/2}}{\sqrt{n}},\ \hat{\Delta}_{\mathrm{c},n}(j, \mathbf{u}_0, \varepsilon_j) + \frac{\hat{\sigma}_{\mathrm{c},n}(j, \mathbf{u}_0, \varepsilon_j) z_{1-\alpha/2}}{\sqrt{n}} \right]$$

4.1.3 Comparison

Costs. Both the forward difference algorithm 11 and the central difference algorithm 12 require generating n iid samples from the distribution of $\mathbf{X}$ and computing d sample averages and sample standard deviations. Therefore, the two algorithms' computational effort differs only in the number of evaluations of h. The forward difference algorithm evaluates h a total of $(d+1)n$ times, compared to $2dn$ evaluations in the central difference algorithm. If d is large and evaluating h is slow, then forward differences can be much faster than central differences. For large d, it can be beneficial to use *simultaneous perturbations*, where

$u_1, \ldots, u_d$ are perturbed simultaneously rather than one at a time; see §3.2 of Chapter 9 of [3] for details.

Benefits. The the forward difference algorithm 11 and central difference algorithm 12 yield estimators of $g_j(\mathbf{u}_0)$ with similar variance. However, the forward difference bias is $O(\varepsilon_j)$, so the $O(\varepsilon_j^2)$ central difference bias vanishes much faster as $\varepsilon_j \to 0$. This makes central differences more attractive than forward differences for most applications.

4.2 Infinitesimal perturbation analysis

While the forward and central difference algorithms are general and can work very well, they both produce biased estimates of $\mathbf{g}(\mathbf{u}_0)$. They also require the user to specify the perturbations $\varepsilon_1, \ldots, \varepsilon_d$. It is not always obvious how these perturbations should be chosen, because they influence both the bias and variance of the gradient estimators. A third drawback of the forward and central difference algorithms is that they are computationally intensive, requiring $(d+1)n$ or $2dn$ evaluations of h, respectively.

Infinitesimal perturbation analysis (IPA) is an alternative to finite difference methods. IPA produces unbiased estimates, is usually simple to implement, and is more computationally efficient than finite difference methods. IPA only applies, however, if the following conditions hold:

1. The distribution of $\mathbf{X}$ *does not depend on* $\mathbf{u}$, *i.e.*,

$$\theta(\mathbf{u}) := \mathbb{E}\, h(\mathbf{X}, \mathbf{u}). \tag{4.2}$$

   (Contrast this with the general definition (4.1), which includes both structural and distributional dependence.)

2. For fixed $\mathbf{X}$, $\nabla_{\mathbf{u}} h(\mathbf{X}, \mathbf{u})$ is *known exactly* at $\mathbf{u}_0$.

3. The structure of h and the distribution of $\mathbf{X}$ allow us to swap the order of integration and differentiation:

$$\nabla\, \mathbb{E}\, h(\mathbf{X}, \mathbf{u}) = \mathbb{E}\, \nabla_{\mathbf{u}} h(\mathbf{X}, \mathbf{u}).$$

If these conditions hold, then

$$\hat{\mathbf{g}}_n(\mathbf{u}_0) := \frac{1}{n} \sum_{i=1}^{n} \nabla_{\mathbf{u}} h(\mathbf{X}_i, \mathbf{u})|_{\mathbf{u}_0}$$

is an unbiased estimator of $\mathbf{g}(\mathbf{u}_0)$, provided $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are sampled from the distribution of $\mathbf{X}$. If $\mathbf{X}_1, \mathbf{X}_2, \ldots$ are also independent, then $\partial h(\mathbf{X}_1, \mathbf{u})/\partial u_j|_{\mathbf{u}_0}, \partial h(\mathbf{X}_2, \mathbf{u})/\partial u_j|_{\mathbf{u}_0}, \ldots$ is a sequence of iid random variables with expectation $g_j(\mathbf{u}_0)$ and finite variance. By the Central

Limit and Converging Together Theorems, therefore, $\hat{g}_{n,j}(\mathbf{u}_0)$ is asymptotically distributed according to

$$\frac{\sqrt{n}(\hat{g}_{n,j}(\mathbf{u}_0) - g_j(\mathbf{u}_0))}{\hat{\sigma}_{n,j}(\mathbf{u}_0)} \Rightarrow \mathcal{N}(0,1) \text{ as } n \to \infty,$$

where

$$\hat{\sigma}_n^2(j, \mathbf{u}_0) := \frac{1}{n-1} \sum_{i=1}^{n} \left(\left. \frac{\partial h(\mathbf{X}_i, \mathbf{u})}{\partial u_j} \right|_{\mathbf{u}_0} - \hat{g}_{n,j}(\mathbf{u}_0) \right)^2$$

is an estimator of $\operatorname{Var} \partial h(\mathbf{X}, \mathbf{u}) / \partial u_j|_{\mathbf{u}_0}$. This allows us to construct confidence intervals for the partial derivatives of θ with respect to $u_1, \ldots, u_d$.

4.2.1 The IPA algorithm

Algorithm 13 (IPA estimation of $\mathbf{g}(\mathbf{u}) = \nabla \mathbb{E}\, h(\mathbf{X}, \mathbf{u})$ at $\mathbf{u}_0$). Given confidence level $\alpha \in (0,1)$ and sample size $n \in \mathbb{N}$:

1. generate $\mathbf{X}_1, \ldots, \mathbf{X}_n$ iid from the distribution of $\mathbf{X}$
2. estimate $\mathbf{g}(\mathbf{u}_0) = \nabla \mathbb{E}\, h(\mathbf{X}, \mathbf{u}_0) = \mathbb{E}\, \nabla h(\mathbf{X}, \mathbf{u}_0)$ by

$$\hat{\mathbf{g}}_n(\mathbf{u}_0) = \frac{1}{n} \sum_{i=1}^{n} \nabla h(\mathbf{X}_i, \mathbf{u}_0)$$

3. for $j = 1, \ldots, d$
   (a) estimate $\operatorname{Var} \partial h(\mathbf{X}, \mathbf{u}) / \partial u_j|_{\mathbf{u}_0}$ by

$$\hat{\sigma}_{n,j}^2(\mathbf{u}_0) = \frac{1}{n-1} \sum_{i=1}^{n} \left(\left. \frac{\partial h(\mathbf{X}_i, \mathbf{u})}{\partial u_j} \right|_{\mathbf{u}_0} - \hat{g}_{n,j}(\mathbf{u}_0) \right)^2$$

   (b) conclude with approximately $100(1-\alpha)\%$ confidence that

$$\frac{\partial \theta}{\partial u_j} \in \left[\hat{g}_{n,j}(\mathbf{u}_0) - \frac{\hat{\sigma}_{n,j}(\mathbf{u}_0) z_{1-\alpha/2}}{\sqrt{n}},\ \hat{g}_{n,j}(\mathbf{u}_0) + \frac{\hat{\sigma}_{n,j}(\mathbf{u}_0) z_{1-\alpha/2}}{\sqrt{n}} \right]$$

4.2.2 When does IPA work?

IPA can be applied if assumptions 1–3 hold. The first assumption depends on the model and simulator. The second assumption can be relaxed. If we don't have an analytical expression

for $\partial h(\mathbf{X}_i, \mathbf{u})/\partial u_j|_{\mathbf{u}_0}$, we can approximate it by the central difference

$$\frac{h(\mathbf{X}_i, \mathbf{u}_0 + (\varepsilon/2)\mathbf{e}_j) - h(\mathbf{X}_i, \mathbf{u}_0 - (\varepsilon/2)\mathbf{e}_j)}{\varepsilon}.$$

Since $h(\mathbf{X}_i, \cdot)$ can be evaluated exactly, we do not need Monte Carlo techniques to evaluate this central difference. If the perturbation ε is sufficiently small, then the $O(\varepsilon^2)$ bias is negligible.

The third assumption,

$$\nabla \mathbb{E}\, h(\mathbf{X}, \mathbf{u}) = \mathbb{E}\, \nabla_{\mathbf{u}} h(\mathbf{X}, \mathbf{u}),$$

is the most restrictive. A sufficient condition for it to hold is that both of the following are satisfied:

- $h(\mathbf{X}, \mathbf{u})$ is differentiable with respect to each u_j at $\mathbf{u}_0$ with probability 1, and
- there exist $\delta > 0$ and $\ell : \mathbb{R}^m \to \mathbb{R}$ such that $\mathbb{E}\, \ell(\mathbf{X})$ is finite, and for almost every $\mathbf{X}$,

$$\max\left\{\|\mathbf{v} - \mathbf{u}_0\|, \|\mathbf{w} - \mathbf{u}_0\|\right\} \le \delta \quad \Longrightarrow \quad |h(\mathbf{X}, \mathbf{v}) - h(\mathbf{X}, \mathbf{w})| \le \ell(\mathbf{X}) \|\mathbf{v} - \mathbf{w}\|.$$

The second condition says that $h(\mathbf{X}, \mathbf{u})$ is almost surely Lipschitz continuous on a δ-neighborhood of $\mathbf{u}_0$, and that the Lipschitz constant $\ell(\mathbf{X})$ has finite expected value.

4.3 The likelihood ratio method

IPA is unbiased, simple to implement, and computationally efficient. However, it can only be applied if all $\mathbf{u}$-dependence is in the function h. By contrast, the likelihood ratio method applies only when h *is independent of* $\mathbf{u}$, *i.e.*, when

$$\theta(\mathbf{u}) := \mathbb{E}^{\mathbf{u}}\, h(\mathbf{X}).$$

(Contrast this with the general definition (4.1), which includes both structural and distributional dependence, and with the IPA definition (4.2), which includes only structural dependence.) The basic idea of the likelihood ratio method is to use the importance sampling trick (see §3.5) to reduce the problem to one with only structural dependence, then apply IPA.

Suppose $\mathbf{X}$ is continuous with pdf $f(\cdot, \mathbf{u})$ and $\tilde{f}$ is an importance sampling density. (Recall that this means $\tilde{f}$ has the same support as $f(\cdot, \mathbf{u})$, *i.e.*,

$$f(\mathbf{x}, \mathbf{u}) \neq 0 \quad \Longrightarrow \quad \tilde{f}(\mathbf{x}) \neq 0$$

for all $\mathbf{x} \in \mathbb{R}^m$.) Then

$$\begin{aligned}\theta(\mathbf{u}) = \mathbb{E}^{\mathbf{u}}\, h(\mathbf{X}) &= \int_{\mathbb{R}^m} h(\mathbf{x}) f(\mathbf{x}, \mathbf{u}) \mathrm{d}\mathbf{x} = \int_{\mathbb{R}^m} \left(\frac{h(\mathbf{x}) f(\mathbf{x}, \mathbf{u})}{\tilde{f}(\mathbf{x})}\right) \tilde{f}(\mathbf{x}) \mathrm{d}\mathbf{x} \\ &= \mathbb{E}\, \tilde{h}(\tilde{\mathbf{X}}, \mathbf{u}).\end{aligned}$$

Here the random vector $\tilde{\mathbf{X}}$ has pdf $\tilde{f}$, and $\tilde{h} : \mathbb{R}^m \times \mathbb{R}^d \to \mathbb{R}$ is defined by

$$\tilde{h}(\mathbf{x}) = \begin{cases} 0 & \text{if } f(\mathbf{x}) = 0 \\ h(\mathbf{x}, \mathbf{u}) f(\mathbf{x}, \mathbf{u}) / \tilde{f}(\mathbf{x}) & \text{otherwise.} \end{cases}$$

By working with $\tilde{\mathbf{X}} \sim \tilde{f}(\cdot)$ instead of $\mathbf{X} \sim f(\cdot, \mathbf{u})$ and $\tilde{h}$ instead of h, we have found that

$$\theta(\mathbf{u}) = \mathbb{E}^{\mathbf{u}} h(\mathbf{X}) = \mathbb{E}\, \tilde{h}(\tilde{\mathbf{X}}, \mathbf{u}).$$

In other words, we have transformed all distributional dependence on $\mathbf{u}$ into structural dependence on $\mathbf{u}$. If $\tilde{h}$ and $\tilde{\mathbf{X}}$ satisfy the conditions in §4.2.2, therefore, we can estimate $\nabla\theta(\mathbf{u})$ using IPA.

4.3.1 The likelihood ratio algorithm

Algorithm 14 (Likelihood ratio estimation of $\mathbf{g}(\mathbf{u}) = \nabla\, \mathbb{E}^{\mathbf{u}} h(\mathbf{X})$ at $\mathbf{u}_0$). Given confidence level $\alpha \in (0, 1)$, sample size $n \in \mathbb{N}$, and importance sampling density $\tilde{f}$:

1. generate $\tilde{\mathbf{X}}_1, \ldots, \tilde{\mathbf{X}}_n$ iid from $\tilde{f}$
2. estimate $\mathbf{g}(\mathbf{u}_0) = \nabla\, \mathbb{E}\, \tilde{h}(\tilde{\mathbf{X}}, \mathbf{u}_0) = \mathbb{E}\, \nabla \tilde{h}(\tilde{\mathbf{X}}, \mathbf{u}_0)$ by

$$\hat{\mathbf{g}}_n(\mathbf{u}_0) = \frac{1}{n} \sum_{i=1}^{n} \nabla \tilde{h}(\tilde{\mathbf{X}}_i, \mathbf{u}_0)$$

3. for $j = 1, \ldots, d$
   (a) estimate $\operatorname{Var} \partial h(\mathbf{X}, \mathbf{u}) / \partial u_j |_{\mathbf{u}_0}$ by

$$\hat{\sigma}^2_{n,j}(\mathbf{u}_0) = \frac{1}{n-1} \sum_{i=1}^{n} \left(\left. \frac{\partial h(\mathbf{X}_i, \mathbf{u})}{\partial u_j} \right|_{\mathbf{u}_0} - \hat{g}_{n,j}(\mathbf{u}_0) \right)^2$$

   (b) conclude with approximately $100(1-\alpha)\%$ confidence that

$$g_j(\mathbf{u}_0) \in \left[\hat{g}_{n,j}(\mathbf{u}_0) - \frac{\hat{\sigma}_{n,j}(\mathbf{u}_0) z_{1-\alpha/2}}{\sqrt{n}},\ \hat{g}_{n,j}(\mathbf{u}_0) + \frac{\hat{\sigma}_{n,j}(\mathbf{u}_0) z_{1-\alpha/2}}{\sqrt{n}} \right]$$

4.3.2 Choosing an importance sampling density

The only requirements on the density $\tilde{f}$ are that

1. we can sample from $\tilde{f}$, and
2. $\tilde{f}(\mathbf{x}) \neq 0$ whenever $f(\mathbf{x}, \mathbf{u}) \neq 0$.

The typical choice of $\tilde{f}$ is $\tilde{f}(\mathbf{x}) = f(\mathbf{x}, \mathbf{u}_0)$.

4.4 Summary

Algorithm	**u**-dependence	# h calls	Advantages	Disadvantages
FFD	general	$(d+1)n$	model-free	choose $\varepsilon_1, \ldots, \varepsilon_d$ $O(\varepsilon_j)$ bias
CFD	general	$2dn$	model-free	choose $\varepsilon_1, \ldots, \varepsilon_d$ $O(\varepsilon_j^2)$ bias
IPA	structural	0	unbiased low variance	prove $\nabla \mathbb{E}\, h(\mathbf{X}, \mathbf{u}) = \mathbb{E}\, \nabla h(\mathbf{X}, \mathbf{u})$ express $\nabla_{\mathbf{u}} h(\cdot, \mathbf{u})$ analytically
LR	distributional	0	unbiased	choose $\tilde{f}$ prove $\nabla \mathbb{E}\, \tilde{h}(\tilde{\mathbf{X}}, \mathbf{u}) = \mathbb{E}\, \nabla \tilde{h}(\tilde{\mathbf{X}}, \mathbf{u})$ express $\nabla_{\mathbf{u}} \tilde{h}(\cdot, \mathbf{u})$ analytically often high variance

Bibliography

[1] Ross, Sheldon M. *Stochastic processes*. Vol. 2. New York: John Wiley & Sons, 1996.

[2] Ross, Sheldon M. *Simulation*. 5$^{\text{th}}$ edition. San Diego: Academic Press, 2002.

[3] Henderson, Shane G., and Barry L. Nelson, eds. *Handbooks in Operations Research and Management Science: Simulation*. Vol. 13. Elsevier, 2006.