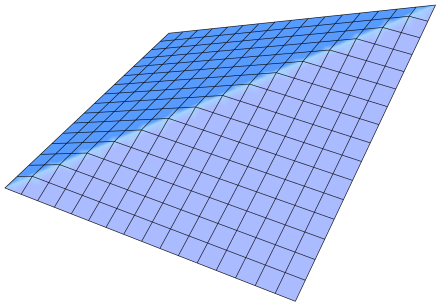

STAT 340: Computer Simulation of Complex Systems

Prof. Marius Hofert • Winter 2015 • University of Waterloo
TR 8:30–9:50AM in MC 2065

Transcribed, LaTeXed, and edited by Rob Zimmerman

Note: These are personal lecture notes, not official course materials. They may contain errors (which by default you should attribute to me, Rob Zimmerman, rather than the course instructor). The permanent home of these — and all of my other lecture notes — is https://rob-zimmerman.github.io/coursenotes.

Contents

1. Discrete Event Simulation

Lecture 1 — Tuesday, January 06, 2015

Review of Basic Concepts in Probability

We start by reviewing some basic concepts in probability theory.

Definition 1.1 A **probability space** is a triple $(\Omega, \mathcal{F}, \mathbb{P})$. Here Ω is the **sample space**, consisting of the possible outcomes of an experiment or states of nature. The collection $\mathcal{F}$

is a **σ-algebra**, so its elements are the subsets of Ω to which probabilities can be assigned. If $|\Omega| < \infty$, we usually take $\mathcal{F} = \mathcal{P}(\Omega)$; if $\Omega = \mathbb{R}^d$, we often take $\mathcal{F} = \mathcal{B}(\mathbb{R}^d)$. Finally, $\mathbb{P} : \mathcal{F} \to [0, 1]$ is a **probability measure**, meaning that $\mathbb{P}(\Omega) = 1$ and

$$\mathbb{P}\left(\bigcup_{i\in\mathbb{N}} A_i\right) = \sum_{i\in\mathbb{N}} \mathbb{P}(A_i)$$

for all pairwise disjoint sets $A_i \in \mathcal{F}$. This property is called **σ-additivity**.

Definition 1.2 Any $B \in \mathcal{F}$ such that $\mathbb{P}(B) = 0$ is called a **null set**. If $\mathbb{P}(A) = 1$, we say that A holds **almost surely** (**a.s.**).

Definition 1.3 If $\mathbb{P}(B) > 0$, then the **conditional probability** of A given B is

$$\mathbb{P}(A \mid B) := \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}.$$

Proposition 1.4 (Law of total probability) If $\{B_i\}_{i\in\mathbb{N}} \subseteq \mathcal{F}$ forms a partition of Ω, so that $\cup_i B_i = \Omega$ and $B_i \cap B_j = \emptyset$ for $i \neq j$, then

$$\mathbb{P}(A) = \sum_{i\in\mathbb{N}} \mathbb{P}(A \cap B_i) = \sum_{i\in\mathbb{N}} \mathbb{P}(A \mid B_i) \cdot \mathbb{P}(B_i),$$

where $\mathbb{P}(A \mid B_i) = 0$ if $\mathbb{P}(B_i) = 0$.

Definition 1.5 The events $A_1, \ldots, A_n$ are **independent** if, for every $k \in \{2, \ldots, n\}$ and every choice of indices $1 \leq i_1 < \cdots < i_k \leq n$,

$$\mathbb{P}(A_{i_1} \cap \cdots \cap A_{i_k}) = \prod_{j=1}^{k} \mathbb{P}(A_{i_j}).$$

Definition 1.6 Any measurable function $X : \Omega \to \mathbb{R}^d$ is called a **random vector**. For $d = 1$, we call X a **random variable**. With inequalities interpreted componentwise when $d > 1$, the function

$$F(x) := \mathbb{P}(X \leq x) = \mathbb{P}(\{\omega : X(\omega) \leq x\})$$

is the **distribution function** (**df**) of X.

Proposition 1.7 Every distribution function of a random variable satisfies the following three properties:

1. F is monotone increasing.
2. F is right-continuous.
3. $\lim_{x\to-\infty} F(x) = 0$ and $\lim_{x\to\infty} F(x) = 1$.

Conversely, these properties characterize distribution functions, and the distribution function determines interval probabilities as follows.

1. Any F satisfying the three properties above is the distribution function of some random variable.
2. If $a < b$, then

$$\begin{aligned}\mathbb{P}(a < X \le b) &= \mathbb{P}(\{\omega \in \Omega : X(\omega) \in (a,b]\}) = \mathbb{P}(X^{-1}((a,b])) \\ &= \mathbb{P}(X^{-1}((-\infty,b]) \setminus X^{-1}(-\infty,a]) \\ &= \mathbb{P}(X^{-1}(-\infty,b]) - \mathbb{P}(X^{-1}(-\infty,a]) \\ &= \mathbb{P}(X \le b) - \mathbb{P}(X \le a) = F(b) - F(a).\end{aligned}$$

3. Taking $a = b - 1/n$ gives

$$\mathbb{P}(X \in (b - \frac{1}{n}, b]) = F(b) - F(b - \frac{1}{n}),$$

and hence

$$\mathbb{P}(X = b) = \lim_{n\to\infty} \left(F(b) - F(b - \frac{1}{n}) \right) = F(b) - F(b^-).$$

In particular, if F is continuous, then $\mathbb{P}(X = b) = 0$ for all $b \in \mathbb{R}$.

Definition 1.8 The distribution function F, or the random variable X itself, is called **discrete** if there exists an at most countable set $S \subseteq \mathbb{R}$ such that $\mathbb{P}(X \in S) = 1$. In this

case, all probability is concentrated on the jumps of F, and

$$f(x) := \mathbb{P}(X = x)$$

is the **probability mass function** (**pmf**).

F (or X) is called **continuous** if F is continuous. F (or X) is called **absolutely continuous** if

$$F(x) = \int_{-\infty}^{x} f(z)\,\mathrm{d}z \quad \text{for all } x \in \mathbb{R}$$

for some $f : \mathbb{R} \to [0, \infty)$ with

$$\int_{-\infty}^{\infty} f(z)\,\mathrm{d}z = 1.$$

In this case, f is called the **density** of F (or of X).

If F is absolutely continuous with density f and moreover, f is bounded by M on the interval between x and $x + h$, then

$$|F(x + h) - F(x)| = \left| \int_{x}^{x+h} f(z)\,\mathrm{d}z \right| \leq M|h| \to 0 \quad \text{as } h \to 0,$$

so F is also continuous. F is even differentiable at each point where f is continuous, with $F'(x) = f(x)$ at such points. However, the converse does not hold: there are continuous distribution functions that are not absolutely continuous, and there are absolutely continuous distribution functions whose densities are not continuous.

Lecture 2 — Thursday, January 08, 2015

The distribution function F can also be of mixed type, with the absolutely continuous and discrete components illustrated in Figure 1:

Definition 2.1 For $y \in (0, 1)$, the **quantile function** of F is

$$F^{-}(y) := \inf\{x \in \mathbb{R} : F(x) \geq y\}.$$

If F is continuous and strictly increasing, then $F^{-} = F^{-1}$, the ordinary inverse of F. Thus F^{-} generalizes the inverse to arbitrary distribution functions.

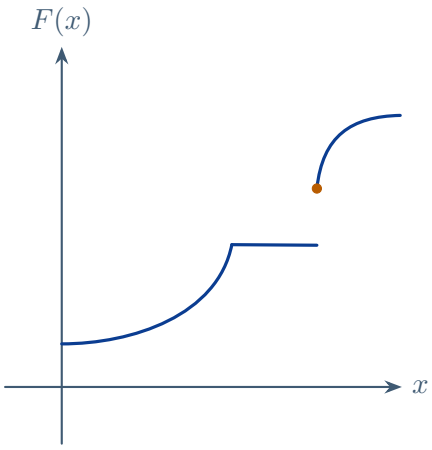

FIGURE 1

Definition 2.2 If

$$\int_{-\infty}^{\infty} |x| \, \mathrm{d}F(x) < \infty,$$

that is, if $\mathbb{E}[|X|] < \infty$ or $X \in L^1(\Omega, \mathcal{F}, \mathbb{P})$, then

$$\mathbb{E}[X] := \int_{-\infty}^{\infty} x \, \mathrm{d}F(x) = \sum_{i \in \mathbb{N}} x_i \cdot f(x_i)$$

in the discrete case. This quantity is called the **expectation**, or **mean**, of X; otherwise, X does not have a **first moment**.

Example 2.3 For the Pareto distribution with density

$$f(x) = \theta(1+x)^{-(\theta+1)}, \quad x \geq 0, \quad \theta > 0,$$

we have $\mathbb{E}[X] = \infty$ for $\theta \leq 1$.

For a Cauchy distribution with density

$$f(x) = \frac{1}{\pi(1+x^2)},$$

the first moment does not exist.

Definition 2.4 If

$$\int_{-\infty}^{\infty} x^2 \, \mathrm{d}F(x) < \infty,$$

that is, if $\mathbb{E}\left[X^2\right] < \infty$ or $X \in L^2(\Omega, \mathcal{F}, \mathbb{P})$, then

$$\mathrm{Var}(X) := \mathbb{E}\left[(X - \mathbb{E}\left[X\right])^2\right] = \mathbb{E}\left[X^2\right] - (\mathbb{E}\left[X\right])^2$$

is the **variance** of X; otherwise, X does not have a **finite second moment**.

Proposition 2.5 Let $X, Y \in L^2$. Then:

1. $aX + bY \in L^2$, with $\mathbb{E}\left[aX + bY\right] = a\mathbb{E}\left[X\right] + b\mathbb{E}\left[Y\right]$.
2. $$\mathbb{E}\left[X\right] = \int_0^{\infty} (1 - F(x)) \, \mathrm{d}x - \int_{-\infty}^{0} F(x) \, \mathrm{d}x.$$
3. If $X \leq Y$ almost surely, then $\mathbb{E}\left[X\right] \leq \mathbb{E}\left[Y\right]$.
4. For $A \in \mathcal{F}$, consider the indicator function
   $$\mathbb{1}_A(\omega) = \begin{cases} 1, & \omega \in A, \\ 0, & \omega \notin A. \end{cases}$$
   Then $\mathbb{E}\left[\mathbb{1}_A\right] = \mathbb{P}(A)$.
5. If $X \geq 0$ almost surely and $\mathbb{E}\left[X\right] = 0$, then $X = 0$ almost surely.
6. (**Jensen's inequality**) If ϕ is convex and $\phi(X)$ is integrable, then $\phi(\mathbb{E}\left[X\right]) \leq \mathbb{E}\left[\phi(X)\right]$. With $\phi(x) = |x|^\beta$ for $\beta \geq 1$, it follows that $(\mathbb{E}\left[|X|\right])^\beta \leq \mathbb{E}\left[|X|^\beta\right]$. Thus the existence of higher moments implies the existence of lower moments.
7. $\mathrm{Var}(aX + b) = a^2 \, \mathrm{Var}(X)$. If $(X_i)_{i \in \mathbb{N}}$ are uncorrelated random variables (that is, $\mathrm{Cov}(X_i, X_j) = 0$ for $i \neq j$) and $\sup_i \mathrm{Var}(X_i) < \infty$, then
   $$\mathrm{Var}\left(\frac{1}{n} \sum_{i=1}^{n} X_i\right) \xrightarrow{n \to \infty} 0.$$

Proof. For (1), note that

$$\int_{-\infty}^{\infty} |ax+by|^2 \,\mathrm{d}F_{X,Y}(x,y) \le 2a^2 \int_{-\infty}^{\infty} |x|^2 \,\mathrm{d}F_X(x) + 2b^2 \int_{-\infty}^{\infty} |y|^2 \,\mathrm{d}F_Y(y) < \infty,$$

so $aX + bY \in L^2$. Moreover,

$$\begin{aligned}\mathbb{E}\left[aX+bY\right] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (ax+by)\,\mathrm{d}F_{X,Y}(x,y) \\ &= a\int_{-\infty}^{\infty} x\,\mathrm{d}F_X(x) + b\int_{-\infty}^{\infty} y\,\mathrm{d}F_Y(y) = a\mathbb{E}\left[X\right] + b\mathbb{E}\left[Y\right].\end{aligned}$$

For (2), note that

$$\begin{aligned}\int_0^{\infty} (1-F(x))\,\mathrm{d}x - \int_{-\infty}^{0} F(x)\,\mathrm{d}x &= \int_0^{\infty} \mathbb{P}(X>x)\,\mathrm{d}x - \int_{-\infty}^{0} \mathbb{P}(X\le x)\,\mathrm{d}x \\ &= \int_0^{\infty}\int_x^{\infty} \mathrm{d}F(z)\,\mathrm{d}x - \int_{-\infty}^{0}\int_{-\infty}^{x} \mathrm{d}F(z)\,\mathrm{d}x.\end{aligned}$$

Interchanging the order of integration gives

$$\begin{aligned}&\int_0^{\infty}\int_x^{\infty} \mathrm{d}F(z)\,\mathrm{d}x - \int_{-\infty}^{0}\int_{-\infty}^{x} \mathrm{d}F(z)\,\mathrm{d}x \\ &= \int_0^{\infty}\int_0^{z} \mathrm{d}x\,\mathrm{d}F(z) - \int_{-\infty}^{0}\int_z^{0} \mathrm{d}x\,\mathrm{d}F(z) \\ &= \int_0^{\infty} z\,\mathrm{d}F(z) + \int_{-\infty}^{0} z\,\mathrm{d}F(z) = \int_{-\infty}^{\infty} x\,\mathrm{d}F(x) = \mathbb{E}\left[X\right].\end{aligned}$$

For (3), we have

$$F_X(x) = \mathbb{P}(X\le x) \ge \mathbb{P}(Y\le x) = F_Y(x) \quad \text{for all } x\in\mathbb{R}.$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[X\right] &= \int_0^{\infty} (1-F_X(x))\,\mathrm{d}x - \int_{-\infty}^{0} F_X(x)\,\mathrm{d}x \\ &\le \int_0^{\infty} (1-F_Y(x))\,\mathrm{d}x - \int_{-\infty}^{0} F_Y(x)\,\mathrm{d}x \\ &= \mathbb{E}\left[Y\right].\end{aligned}$$

For (4), note that $\mathbb{1}_A$ is a random variable with distribution function

$$F(x) = \begin{cases} 0, & x < 0, \\ \mathbb{P}(A^c), & 0 \le x < 1, \\ 1, & x \ge 1. \end{cases}$$

Thus

$$\mathbb{E}\left[\mathbb{1}_A\right] = \int_{-\infty}^{\infty} x \, \mathrm{d}F(x) = 0 \cdot \mathbb{P}(A^c) + 1 \cdot \mathbb{P}(A) = \mathbb{P}(A).$$

For (5), Markov's inequality gives

$$\mathbb{P}(X \ge 1/n) \le n\mathbb{E}[X] = 0 \quad \text{for every } n \in \mathbb{N}.$$

Since $\{X > 0\} = \bigcup_{n=1}^{\infty} \{X \ge 1/n\}$, we have $\mathbb{P}(X > 0) = 0$ and hence $X = 0$ almost surely.

For (6), choose a subgradient m of ϕ at $\mathbb{E}[X]$. The supporting-line inequality gives

$$\phi(X) \ge \phi(\mathbb{E}[X]) + m(X - \mathbb{E}[X]).$$

Taking expectations proves $\mathbb{E}[\phi(X)] \ge \phi(\mathbb{E}[X])$.

For (7), we have

$$\operatorname{Var}(aX + b) = \mathbb{E}\left[(aX + b - a\mathbb{E}\left[X\right] - b)^2\right] = a^2\mathbb{E}\left[(X - \mathbb{E}\left[X\right])^2\right] = a^2 \operatorname{Var}(X).$$

If $X_1, \ldots, X_n$ are uncorrelated and $V := \sup_i \operatorname{Var}(X_i) < \infty$, then

$$\begin{aligned} \operatorname{Var}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) &= \frac{1}{n^2}\sum_{i=1}^{n} \operatorname{Var}(X_i) + \frac{1}{n^2}\sum_{i \ne j} \operatorname{Cov}(X_i, X_j) \\ &= \frac{1}{n^2}\sum_{i=1}^{n} \operatorname{Var}(X_i) \le \frac{V}{n} \xrightarrow{n \to \infty} 0. \end{aligned}$$

□

Expectation and variance summarize properties of F, namely location and scale, using real numbers. There are many other such summary statistics. Two important ones in finance and insurance are given by the following definitions.

Definition 2.6 The **Value-at-Risk** of X at confidence level $\alpha \in (0,1)$ is

$$\mathrm{VaR}_\alpha(X) = F_X^-(\alpha),$$

and the **expected shortfall** of X at confidence level $\alpha \in (0,1)$ is

$$\mathrm{ES}_\alpha(X) = \frac{1}{1-\alpha}\int_\alpha^1 F_X^-(u)\,\mathrm{d}u.$$

These both summarize properties of the tail of F, that is, of $F(x)$ for large x. Typically $\alpha \in \{0.99, 0.995, 0.9997, \dots\}$. In particular, $\mathrm{VaR}_{0.5}(X)$ is the median of X.

We often consider sequences of random variables $(X_n)_{n\in\mathbb{N}}$ and study one of the following types of convergence:

Definition 2.7

1. (X_n) converges to X **almost surely** if $\mathbb{P}(\lim_{n\to\infty} X_n = X) = 1$.
2. (X_n) converges to X **in probability** if
$$\lim_{n\to\infty} \mathbb{P}\left(|X_n - X| > \varepsilon\right) = 0 \quad \text{for every } \varepsilon > 0.$$
3. (X_n) converges to X **in distribution** if $\lim_{n\to\infty} F_{X_n}(x) = F_X(x)$ at every continuity point x of F_X.

We can show that

$$X_n \xrightarrow{\text{a.s.}} X \implies X_n \xrightarrow{\mathbb{P}} X \implies X_n \xrightarrow{d} X.$$

Definition 2.8 The sequence $(X_n)_{n\in\mathbb{N}}$ is **independent and identically distributed** if $X_1, X_2, \dots$ are independent and all X_i have the same distribution function.

Theorem 2.9 (Laws of large numbers) Two important laws of large numbers are the following.

1. **Weak law of large numbers**: if $(X_n)_{n\in\mathbb{N}}$ is an independent sequence of random

variables with common finite mean μ and common finite variance σ^2, then

$$\bar{X}_n := \frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{\mathbb{P}} \mu.$$

2. **Strong law of large numbers**: under the same assumptions, $\bar{X}_n \xrightarrow{\text{a.s.}} \mu$.

Lecture 3 — Tuesday, January 13, 2015

Theorem 3.1 (Central limit theorem) Let $(X_n)_{n\in\mathbb{N}}$ be an independent and identically distributed sequence of random variables with $\mu = \mathbb{E}[X_1]$ and $0 < \sigma^2 = \text{Var}(X_1) < \infty$. Then

$$\frac{\bar{X}_n - \mu}{\sigma}\sqrt{n} = \frac{\sum_{i=1}^{n} X_i - n\mu}{\sigma\sqrt{n}} \xrightarrow{d} \mathcal{N}(0,1).$$

That is,

$$\bar{X}_n \overset{n \text{ large}}{\sim} \mathcal{N}(\mu, \sigma^2/n)$$

and

$$\sum_{i=1}^{n} X_i \overset{n \text{ large}}{\sim} \mathcal{N}(n\mu, n\sigma^2).$$

Systems, Models, and Simulation

A **system** is a collection of **entities**, such as people or machines, that act and interact together to achieve a goal. Entities are characterized by **attributes**. To study a system, we usually make assumptions, often specified by logical relationships. These assumptions constitute a **model** of the system. If the model is too complex to solve analytically, then we study it by **simulation** (i.e., we use a computer to repeatedly evaluate the model and gather data in order to estimate the quantities of interest, also called **performance measures**). The **state** of the system is the collection of random variables needed to describe the system at a particular time.

A system simulation can be **discrete** (if state variables change at only countably many time points), **continuous** (if state variables change continuously with time), or **mixed**, combining discrete and continuous changes. Broadly, we can study a system either by experimenting with the actual system or by experimenting with a model. A model may be physical or mathematical/probabilistic;

a mathematical model may then be studied analytically, when possible, or by simulation.

Remark 3.2

1. If possible and cost-effective, alter the system physically. This provides the most valid results. However, this is rarely feasible and thus requires a model to study. This model must be valid.
2. Besides physical models, most models are mathematical (representing a system with logical or quantitative relationships).
3. If an analytic solution is available (rarely the case), then study the model in this way (no estimation error); otherwise, use simulation.

Simulation has several advantages, drawbacks, and common pitfalls.

1. The main advantages are as follows:
   (a) Simulation can answer questions in cases where models are not analytically tractable.
   (b) Models can be studied under different assumptions.
   (c) Simulations can be used to compare models.
   (d) Simulations allow us to control conditions much better than actual experiments.
   (e) Simulations allow us to study a system over a long period of time.
2. The main drawbacks are as follows:
   (a) A simulation only provides estimates of performance measures.
   (b) Simulation models are often time-consuming to set up.
   (c) If a model is not valid, the simulation results are not valid.
3. Common pitfalls include failing to collect enough data, using software incorrectly, implementation errors, wrong model assumptions such as incorrect distributions or independence assumptions, analyzing data from only one simulation run, wrong seed usage, and undetected numerical issues.

Simulation models (that is, mathematical models to be studied by simulation) can be classified as follows:

1. Static versus dynamic models. In a **static model**, time plays no role; in a **dynamic model**, the system evolves over time.
2. Deterministic versus stochastic models. A **deterministic simulation** model has no random components, whereas a **stochastic simulation** model has at least one random component, as in a bank queue.
3. Continuous versus discrete versus mixed models.

A typical simulation study proceeds according to the flowchart in Figure 2:

Lecture 4 — Thursday, January 15, 2015

Discrete Event Simulation

Discrete event simulation models concern the modeling of dynamic, stochastic, and discrete systems. The time points at which the system may change are known as **events**. The variable which keeps track of time is called the **simulation clock**. After the clock is initialized (typically, $t_0 = 0$), times of occurrence of future events are determined, and the simulation clock is advanced to the first of these future events. Then the system is updated (including performance measures and future events from then on). This process is known as the **next-event time-advance approach**. It is repeated until a stopping criterion is met.

A discrete simulation model is organized around the following components:

1. **System state**: a collection of state variables (that is, variables necessary to describe the system at a particular time).
2. **Simulation clock**.
3. **Event list**: a list containing, for each type of event, the next time of occurrence.
4. **Statistical counters**: variables storing statistical information (typically for estimating performance measures).
5. **Routines**: the **initialization routine** initializes the simulation clock, system state, statistical counters, and event list; the **timing routine** determines the next event from the event list and advances the simulation clock; the **event routine** updates the system state and statistical counters, generates future events, and adds them to the event list; and **library**

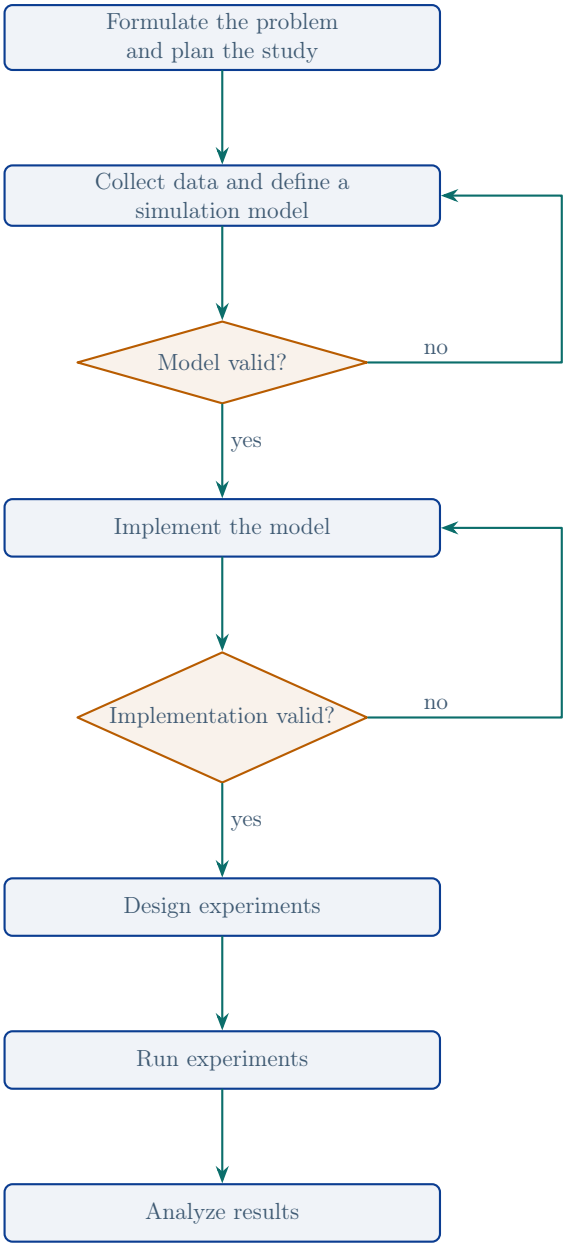
Formulate the problem and plan the study
Collect data and define a simulation model
Model valid?
no
yes
Implement the model
Implementation valid?
no
yes
Design experiments
Run experiments
Analyze results

FIGURE 2

routines provide pseudorandom number generators, optimization procedures, and so on.

6. **Report generator**: computes estimates, reports results.
7. **Main program**: the master program that invokes the timing routine, calls the event routine, checks for termination, and invokes the report generator.

The logical relationships among these components are represented by the control flow in Figure 3:

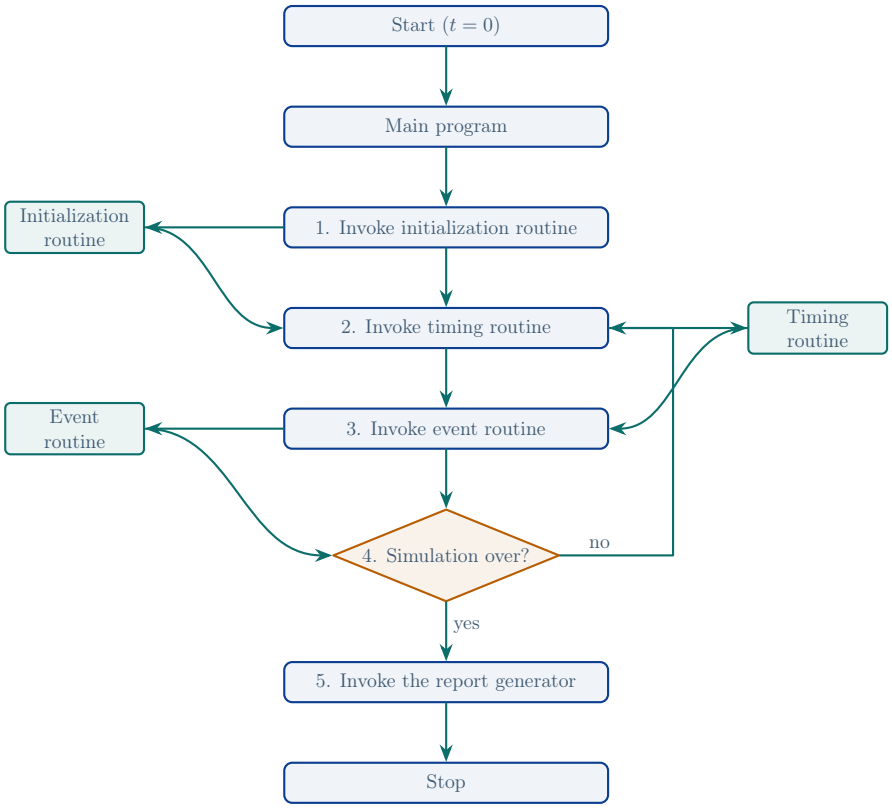

FIGURE 3

This organization of a discrete event simulation is also known as the **event scheduling approach**, because the times of future events are explicitly coded and scheduled to occur in the simulated future.

Queuing Systems

A **queuing system** consists of at least one **server** which provides some service to arriving **customers**. Customers who arrive to find all servers busy join a **queue**. Let t_i be the arrival time of customer i (with $t_0 = 0$). A queuing system is characterized by the following three components:

1. The **arrival process** describes how customers arrive to the system. $A_i := t_i - t_{i-1}$ is the interarrival time between the $(i-1)$st and ith arrival of customers.
2. The **service mechanism** specifies the number s of servers, whether each server has its own queue or whether there is just one queue, and the probability distribution of customers' service times. We write S_i for the service time of the ith arriving customer.
3. The **queue discipline** defines the rules servers use to choose the next customers. Examples include **FIFO** (**first-in, first-out**), **LIFO** (**last-in, first-out**), and **priority service**, where customers are served in order of importance.

A ***GI/G/s* queue** is a queuing system in which there are s parallel servers and one FIFO queue. The interarrival times are iid with distribution F_A, the service times are iid with distribution F_S, and the two sequences are independent. Assuming their means are positive and finite, the utilization factor

$$f = \frac{\mathbb{E}[S_1]}{s \cdot \mathbb{E}[A_1]}$$

measures how heavily the servers are being used. If F_A and F_S are exponential, we write $M/M/s$; the 'M' stands for **Markovian**, meaning memorylessness:

$$X \sim \text{Exp}(\lambda) \qquad \text{implies} \qquad \mathbb{P}(X > s+t \mid X > s) = \mathbb{P}(X > t) \quad \text{for all } s,t > 0.$$

Let D_i be the delay in the queue of customer i, and let $W_i = D_i + S_i$ be the waiting time in the system of customer i. These are both discrete-time customer statistics. Let Q_t be the number of customers in the queue at time t, let L_t be the number of customers in the system at time t, and let

$$\mathbb{1}_{i,t} = \mathbb{1}_{\{\text{server } i \text{ is busy at time } t\}}.$$

These three are continuous-time statistics. Under the usual stability and ergodicity assumptions, the corresponding steady-state quantities can be estimated as follows:

1. The **steady-state average delay** $\mathbb{E}[D]$ is estimated by

$$\bar{D}_n = \frac{1}{n}\sum_{i=1}^{n} D_i.$$

2. The **steady-state average waiting time** $\mathbb{E}[W]$ is estimated by

$$\bar{W}_n = \frac{1}{n}\sum_{i=1}^{n} W_i.$$

3. The **steady-state time-average number in the queue** is

$$Q = \lim_{T\to\infty} \frac{1}{T}\int_0^T Q_t \,\mathrm{d}t$$

estimated by

$$\frac{1}{T(n)}\int_0^{T(n)} Q_t \,\mathrm{d}t,$$

where $T(n)$ is the time of departure of the nth customer.

4. The **steady-state time-average number in the system** is

$$L = \lim_{T\to\infty} \frac{1}{T}\int_0^T L_t \,\mathrm{d}t,$$

estimated by

$$\frac{1}{T(n)}\int_0^{T(n)} L_t \,\mathrm{d}t.$$

5. The **steady-state time-average utilization** is

$$U_i = \lim_{T\to\infty} \frac{1}{T}\int_0^T \mathbb{1}_{i,t} \,\mathrm{d}t,$$

estimated by

$$\frac{1}{T(n)}\int_0^{T(n)} \mathbb{1}_{i,t} \,\mathrm{d}t.$$

Figure 4 summarizes the arrivals, queue, servers, and state variables of a multiserver system.

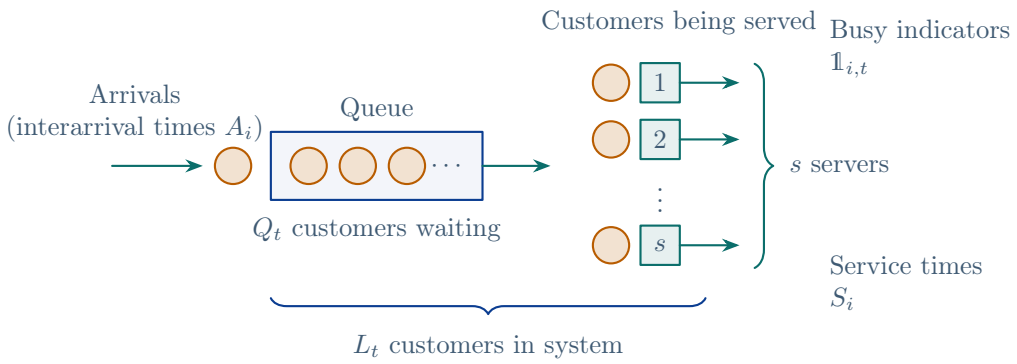

FIGURE 4

Lecture 5 — Tuesday, January 20, 2015

For an $M/M/1$ queuing system, let $c_i = t_i + D_i + S_i$ denote the departure time of customer i, and let e_j denote the time of occurrence of the jth event, whether it is an arrival or a departure. The next-event time-advance approach is illustrated by the timeline in Figure 5:

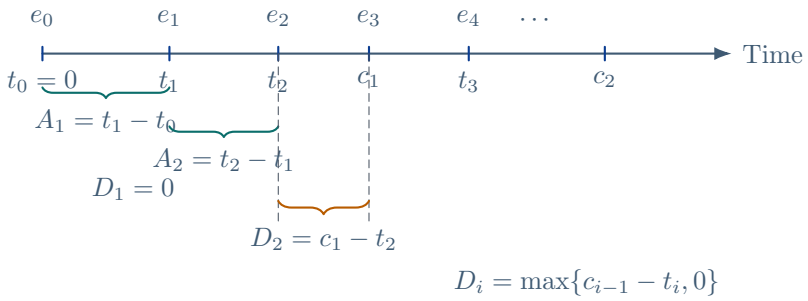

FIGURE 5

Let n be the number of customers whose delays have been recorded, including customers with delay 0, and let

$$TD_n = \sum_{i=1}^{n} D_i$$

be the total delay for the first n customers, let t_a be the time of the next arrival, let t_d be the time of the next departure, and let t_{aq} be the vector of arrival times of the customers currently in the queue. The arrival and departure events are governed by the flow charts in Figure 7.

Example 5.1 Consider an $M/M/1$ queue with interarrival times

$$(A_1, \ldots, A_9) = (0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9)$$

and service times

$$(S_1, \ldots, S_6) = (2.0, 0.7, 0.2, 1.1, 3.7, 0.6).$$

Give a snapshot of the system until D_6 is observed and compute $\bar{D}_6$.

Solution. We have

Simulation clock		System state			Event List		Stats counters	
t	T	U_t	Q_t	$t_{a,q}$	t_a	t_d	n	TD_n
0	∞	0	0	[ ]	0.4	∞	0	0
0.4	∞	1	0	[ ]	1.6	2.4	1	0
1.6	∞	1	1	[1.6]	2.1	2.4	1	0
2.1	∞	1	2	[1.6, 2.1]	3.8	2.4	1	0
2.4	∞	1	1	[2.1]	3.8	3.1	2	0.8
3.1	∞	1	0	[ ]	3.8	3.3	3	1.8
3.3	∞	0	0	[ ]	3.8	∞	3	1.8
3.8	∞	1	0	[ ]	4.0	4.9	4	1.8
4.0	∞	1	1	[4.0]	5.6	4.9	4	1.8
4.9	∞	1	0	[ ]	5.6	8.6	5	2.7
5.6	∞	1	1	[5.6]	5.8	8.6	5	2.7
5.8	∞	1	2	[5.6, 5.8]	7.2	8.6	5	2.7
7.2	∞	1	3	[5.6, 5.8, 7.2]	9.1	8.6	5	2.7
8.6	∞	1	2	[5.8, 7.2]	9.1	9.2	6	5.7

Thus the mean delay based on the first six delayed customers is

$$\bar{D}_6 = \frac{TD_6}{6} = \frac{5.7}{6} = 0.95.$$

□

Lecture 6 — Thursday, January 22, 2015

We now rewrite the logic for scheduling events in a formal algorithmic form.

The simulation clock advances from one event time to the next. A typical sequence is shown in Figure 6:

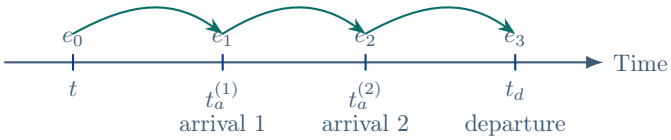

FIGURE 6

The variables are grouped as follows.

1. Count variables: $N = L$ is the number of customers in the system; Q is the number in queue; n is the cumulative number of customers whose delays have been recorded; N_a is the total number of arrivals; N_d is the total number of departures; and U_t indicates whether the server is busy at time t.
2. Time variables: t is the simulation clock; t_a is the next arrival time; t_d is the next departure time (initialized to ∞); and T is the closing time.
3. Queue data and output variables: $t_{a,q}$ stores queued arrival times, D_i is the delay of customer i, and TD is the cumulative delay over recorded customers.

Remark 6.1 A customer served immediately on arrival has delay 0.

The event routines can be summarized by the flowcharts in Figure 7.

The main event loop is as follows: if the next event is an arrival, we call the arrival event routine; if the next event is a departure, we call the departure event routine; and if the next event is after the closing time T, we stop the simulation if there are no customers in the system, or we call the departure event routine until all customers have departed.

The corresponding pseudocode is given next.

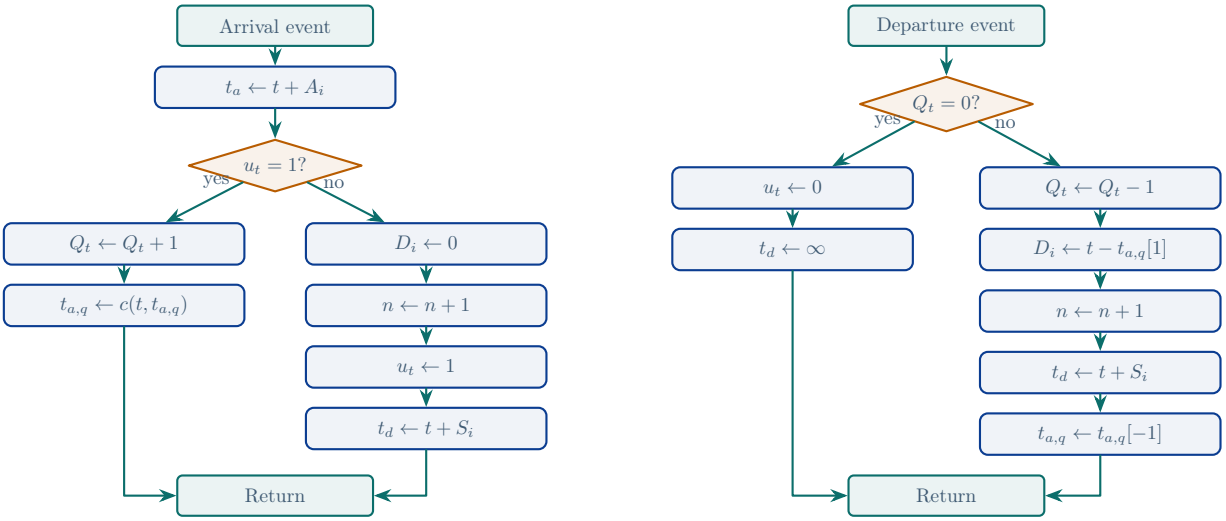

FIGURE 7

Algorithm 1: Event scheduling simulation for an M/M/1 queue on $[0, T]$

Initialize $t \leftarrow 0$, $N \leftarrow 0$, $Q \leftarrow 0$, $U_t \leftarrow 0$, $n \leftarrow 0$, $N_a \leftarrow 0$, $N_d \leftarrow 0$, $TD \leftarrow 0$, $t_{a,q} \leftarrow [\,]$, $t_d \leftarrow \infty$
Generate the first interarrival time A_1 and set $t_a \leftarrow t + A_1$
while $(t < T)$ *or* $(N > 0)$ **do**
    if $(t_a \le t_d)$ *and* $(t_a < T)$ **then**
        Call Algorithm 2
    else
        if $(t_d < t_a)$ *and* $(t_d < T)$ **then**
            Call Algorithm 3
        else
            if $(t_a > T)$ *and* $(t_d > T)$ *and* $(N > 0)$ **then**
                Set $t_a \leftarrow \infty$ and call Algorithm 3
            else
                break
return N_a, N_d, n, TD, *and derived estimates such as* $\bar{D}_n = TD/n$

Algorithm 2: Arrival event update

Set $t \leftarrow t_a$, $N_a \leftarrow N_a + 1$, and $N \leftarrow N + 1$
Generate the next interarrival time A_i and set $t_a \leftarrow t + A_i$
if $N = 1$ **then**
    Set $D_i \leftarrow 0$, $n \leftarrow n + 1$, and $U_t \leftarrow 1$
    Generate service time S_i and set $t_d \leftarrow t + S_i$
else
    Set $Q \leftarrow Q + 1$ and update the queue of arrival times by $t_{a,q} \leftarrow c(t, t_{a,q})$

Algorithm 3: Departure event update

Set $t \leftarrow t_d$, $N_d \leftarrow N_d + 1$, and $N \leftarrow N - 1$
if $N > 0$ **then**
    Set $Q \leftarrow Q - 1$
    Set $D_i \leftarrow t - t_{a,q}[1]$, $TD \leftarrow TD + D_i$, and $n \leftarrow n + 1$
    Generate service time S_i and set $t_d \leftarrow t + S_i$
    Update the queue by removing its first element: $t_{a,q} \leftarrow t_{a,q}[-1]$
else
    Set $t_d \leftarrow \infty$ and $U_t \leftarrow 0$

Lecture 7 — Tuesday, January 27, 2015

We also need the empirical distribution function and the empirical quantile function.

Definition 7.1 Let $X_1, \ldots, X_n$ be an iid sample from a distribution function F. The **empirical distribution function** is

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{X_i \leq x\}}, \quad x \in \mathbb{R}.$$

The empirical distribution function has the following properties:

1. We have $n\hat{F}_n(x) = \sum_{i=1}^{n} \mathbb{1}_{\{X_i \leq x\}}$ and $\mathbb{1}_{\{X_i \leq x\}} \sim \text{Bernoulli}(\mathbb{P}(X_i \leq x)) = \text{Bernoulli}(F(x))$.

Thus $n\hat{F}_n(x) \sim \text{Bin}(n, F(x))$. Taking expectations gives

$$\mathbb{E}\left[\hat{F}_n(x)\right] = \frac{1}{n}\mathbb{E}\left[\text{Bin}(n, F(x))\right] = F(x).$$

Furthermore,

$$\text{Var}(\hat{F}_n(x)) = \frac{1}{n^2}\text{Var}(\text{Bin}(n, F(x))) = \frac{F(x)(1-F(x))}{n}.$$

Thus $\hat{F}_n(x)$ is both unbiased and consistent as an estimator of $F(x)$.

2. For $0 < F(x) < 1$, the central limit theorem gives

$$\sqrt{n}(\hat{F}_n(x) - F(x)) \xrightarrow{d} \mathcal{N}(0, F(x)(1-F(x))).$$

3. We have $\hat{F}_n(x) \xrightarrow{\text{a.s.}} \mathbb{E}\left[\mathbb{1}_{\{X_i \le x\}}\right] = \mathbb{P}(X_i \le x) = F(x)$ for each fixed $x \in \mathbb{R}$ by the strong law of large numbers in (2). We can show even more:

$$\sup_{x\in\mathbb{R}} |\hat{F}_n(x) - F(x)| \xrightarrow{\text{a.s.}} 0.$$

This is the **Glivenko–Cantelli theorem**.

Definition 7.2 Let $X_1, \ldots, X_n$ be an iid sample from a distribution function F. For $p \in (0, 1]$, the **empirical quantile function** is

$$X_{(\lceil np \rceil)} = \hat{F}_n^-(p) = \inf\{x \in \mathbb{R} : \hat{F}_n(x) \ge p\},$$

where

$$\hat{F}_n(x) \ge p \qquad \text{if and only if} \qquad \sum_{i=1}^{n} \mathbb{1}_{\{X_i \le x\}} \ge \lceil np \rceil.$$

The empirical quantile function has the following properties (neither of which are easy to prove!):

1. If $p \in (0, 1)$, F is differentiable at $F^-(p)$, and $f(F^-(p)) > 0$, then

$$\sqrt{n}(\hat{F}_n^-(p) - F^-(p)) \xrightarrow{d} \mathcal{N}\left(0, \frac{p(1-p)}{f(F^-(p))^2}\right).$$

2. If F^- is continuously differentiable on $[a, b] \subseteq (0, 1)$, then for $a < \alpha < \beta < b$, we have

$$\sup_{p\in[\alpha,\beta]} |\hat{F}_n^-(p) - F^-(p)| \xrightarrow{\text{a.s.}} 0.$$

Statistical Analysis of Simulated Data

The statistical analysis of simulated data begins with sample size selection. Performance measures of interest are often expectations. For large n and independent and identically distributed $X_1, \ldots, X_n$ with $\mu = \mathbb{E}[X_i]$ and $\sigma^2 = \mathrm{Var}(X_i) < \infty$, the strong law of large numbers in (2) provides the estimator

$$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i,$$

which is the sample mean. The central limit theorem gives a practical answer to the question of how large n should be.

Lemma 7.3 Let $\alpha \in (0,1)$, $\varepsilon > 0$, and $q_{1-\alpha/2} = \Phi^{-1}(1-\alpha/2)$. For sufficiently large n, the normal approximation and the condition $q_{1-\alpha/2}\sigma/\sqrt{n} \le \varepsilon$ give

$$\mathbb{P}(|\bar{X}_n - \mu| \ge \varepsilon) \lesssim \alpha.$$

Proof. The central limit theorem gives

$$\mathbb{P}(|\bar{X}_n - \mu| \ge \varepsilon) \approx 2\left(1 - \Phi\left(\frac{\varepsilon\sqrt{n}}{\sigma}\right)\right) \le 2\left(1 - \Phi(q_{1-\alpha/2})\right) = \alpha.$$

□

In practical applications, for a pilot sample of size $n_0 \ge 2$, we estimate σ^2 by the sample variance

$$S_{n_0}^2 = \frac{1}{n_0 - 1}\sum_{i=1}^{n_0}(X_i - \bar{X}_{n_0})^2.$$

This is an unbiased estimator of σ^2, since $\mathbb{E}\left[S_{n_0}^2\right] = \sigma^2$. Slutsky's theorem justifies replacing σ by a consistent sample standard deviation in the normal approximation when the final sample is sufficiently large.

The algorithm for data collection is as follows:

Step 1. Fix the error probability $\alpha \in (0,1)$ and absolute error $\varepsilon > 0$.

Step 2. Draw a pilot sample of size $n_0 \ge 2$ and compute S_{n_0}.

Step 3. Compute

$$n' = \left\lceil \left(\frac{q_{1-\alpha/2} S_{n_0}}{\varepsilon} \right)^2 \right\rceil .$$

Step 4. Choose $n = \max\{n_0, n'\}$ and generate the additional $n - n_0$ observations. If the observations are Bernoulli, use $\sqrt{\bar{X}_{n_0}(1 - \bar{X}_{n_0})}$ in place of S_{n_0}.

Confidence intervals provide the corresponding interval estimates. For sufficiently large n, the central limit theorem gives an asymptotic interval in which the true, unknown μ lies with high probability.

Lemma 7.4 Let $X_1, \ldots, X_n$ be iid from F, with $\mu = \mathbb{E}[X_1]$ and $0 < \sigma^2 = \mathrm{Var}(X_1) < \infty$. Then an asymptotic $(1 - \alpha)$-confidence interval is

$$I_n = \left[\bar{X}_n - q_{1-\alpha/2} \frac{S_n}{\sqrt{n}}, \bar{X}_n + q_{1-\alpha/2} \frac{S_n}{\sqrt{n}} \right],$$

meaning that $\mathbb{P}(\mu \in I_n) \to 1 - \alpha$ as $n \to \infty$.

Proof. Slutsky's theorem says that if $Y_n \xrightarrow{d} Y$ and $Z_n \xrightarrow{\mathbb{P}} c \in \mathbb{R}$, then $Y_n Z_n \xrightarrow{d} cY$. Therefore,

$$\begin{aligned} \mathbb{P}\left(-q_{1-\alpha/2} \le \sqrt{n} \frac{\bar{X}_n - \mu}{S_n} \le q_{1-\alpha/2} \right) &= \mathbb{P}\left(-q_{1-\alpha/2} \le \sqrt{n} \frac{\bar{X}_n - \mu}{\sigma} \cdot \frac{\sigma}{S_n} \le q_{1-\alpha/2} \right) \\ &\xrightarrow{n \to \infty} \Phi(q_{1-\alpha/2}) - \Phi(-q_{1-\alpha/2}) \\ &= 1 - \alpha, \end{aligned}$$

since $\sqrt{n}(\bar{X}_n - \mu)/\sigma \xrightarrow{d} \mathcal{N}(0,1)$ by the central limit theorem. Moreover, the law of large numbers applied to X_i and X_i^2 gives $S_n^2 \xrightarrow{\mathbb{P}} \sigma^2$, so $S_n/\sigma \xrightarrow{\mathbb{P}} 1$. □

As before, if $X_i \sim \text{Bernoulli}(p)$, replace S_n by $\sqrt{\bar{X}_n(1 - \bar{X}_n)}$.

Remark 7.5 Before data is observed, the probability that

$$\mu \in \left[\bar{X}_n - q_{1-\alpha/2} \frac{S_n}{\sqrt{n}}, \bar{X}_n + q_{1-\alpha/2} \frac{S_n}{\sqrt{n}} \right] = I_n$$

is approximately $1 - \alpha$. We can check this interpretation experimentally. Suppose we

independently repeat the following procedure 100 times:

Step 1. Draw a random sample of size n from F.

Step 2. Compute the $(1-\alpha)$-confidence interval based on this sample.

Then roughly $(1-\alpha)\cdot 100\%$ of the 100 confidence intervals contain μ. After $\bar{X}_n$ and S_n are observed, the statement $\mu \in I_n$ is either true or false.

Lecture 8 — Thursday, January 29, 2015

To choose a sample size from a target interval length, use the following rule.

Step 1. Fix the significance level $\alpha \in (0,1)$ and target interval length $\ell > 0$.

Step 2. Draw a pilot sample of size $n_0 \geq 2$ and compute S_{n_0}.

Step 3. Compute $n' = \lceil (2q_{1-\alpha/2}S_{n_0}/\ell)^2 \rceil$.

Step 4. Choose $n = \max\{n_0, n'\}$ and generate the additional observations.

This is the same algorithm as we used previously, with ε replaced by $\ell/2$.

Note that if $X_i \sim \mathcal{N}(\mu, \sigma^2)$, then $\sqrt{n}(\bar{X}_n - \mu)/S_n \sim t_{n-1}$. If n is small and X_i is nearly normal for $i \in \{1, \ldots, n\}$, we often replace $q_{1-\alpha/2} = \Phi^{-1}(1-\alpha/2)$ by the corresponding Student quantile $t_{n-1,1-\alpha/2}$.

The Bootstrap

Suppose that $X_1, \ldots, X_n$ are independent and identically distributed with distribution function F, and suppose that F is unknown. We are interested in estimating some quantity $\theta = \theta(F)$. Since F is unknown, we cannot simply simulate from F to improve our approximation of θ. The best we can do from $X_1, \ldots, X_n$ is approximate F by $\hat{F}_n$, and then approximate θ by $\hat{\theta}_n = \theta(\hat{F}_n)$. This gives an estimator of θ, but we still want to know how $\hat{\theta}_n$ is distributed and what $\mathrm{Var}(\hat{\theta}_n)$ is.

The (**nonparametric**) **bootstrap** addresses this problem by treating $\hat{F}_n$ as the true underlying distribution function (the first approximation), and then sampling from $\hat{F}_n$ (i.e., resampling

$X_1, \ldots, X_n$), to approximate the distribution of $\hat{\theta}_n$ (the second approximation). The bootstrap is a powerful method for approximating the distribution of an estimator, and it can be used to construct confidence intervals and perform hypothesis testing.

The nonparametric bootstrap is implemented as follows:

Algorithm 4: Nonparametric bootstrap

Fix a large number $B \in \mathbb{N}$ of bootstrap replications
for $b = 1, \ldots, B$ **do**
    Randomly sample $X_1^{(b)}, \ldots, X_n^{(b)}$ from $X_1, \ldots, X_n$ with replacement
    Form the empirical distribution function $\hat{F}_n^{(b)}(x) = n^{-1} \sum_{i=1}^n \mathbb{1}_{\{X_i^{(b)} \leq x\}}$
    Compute the **bootstrap estimator** $\hat{\theta}_n^{(b)} = \theta(\hat{F}_n^{(b)})$
return *the* ***bootstrap sample*** $\hat{\theta}_n^{(1)}, \ldots, \hat{\theta}_n^{(B)}$ *for approximating the distribution of* $\hat{\theta}_n = \theta(\hat{F}_n)$

Remark 8.1 The final bootstrap sample can be used in several ways.

1. The empirical distribution function

$$\hat{G}_B(t) = \frac{1}{B} \sum_{b=1}^{B} \mathbb{1}_{\{\hat{\theta}_n^{(b)} \leq t\}}$$

   approximates the distribution function of $\hat{\theta}_n$.

2. The bootstrap variance estimator is

$$S_B^2 = \frac{1}{B-1} \sum_{b=1}^{B} (\hat{\theta}_n^{(b)} - \bar{\theta}_n^{(B)})^2$$

   where

$$\bar{\theta}_n^{(B)} = \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}_n^{(b)}.$$

3. Under the regularity conditions that make the percentile bootstrap valid, the interval

$$\left[\hat{\theta}_{n,\alpha/2}^{(B)}, \hat{\theta}_{n,1-\alpha/2}^{(B)}\right]$$

   approximates a $(1-\alpha)$-confidence interval for θ, where $\hat{\theta}_{n,\alpha/2}^{(B)}$ and $\hat{\theta}_{n,1-\alpha/2}^{(B)}$ denote the corresponding empirical quantiles of $\hat{\theta}_n^{(1)}, \ldots, \hat{\theta}_n^{(B)}$.

The bootstrap is useful when F or the distribution of the estimator $\hat{\theta}_n$ is unknown, when n is too small for a central limit theorem approximation to be reliable, or when F is skewed enough that a confidence interval based on the central limit theorem would require a much larger sample size. In particular, a normal-approximation confidence interval for μ is symmetric about the sample mean, whereas a bootstrap interval need not be. The method is also simple and asymptotically consistent under suitable assumptions.

Lecture 9 — Tuesday, February 03, 2015

2. Random Variate Generation

Pseudorandom Number Generation

As we have seen, stochastic simulation models require as input procedures for generating random variates. These are numbers that appear to be realizations of independent random variables, such as independent exponentially distributed random variables for interarrival or service times. We generate **random variates**, not random variables themselves.

Definition 9.1 The deterministic outputs of a generator that are designed to mimic independent Unif$[0, 1]$ variates are called **pseudorandom numbers**. A **pseudorandom number generator** (**PRNG**) is a sequential method in which each number is determined by one or several predecessors according to a fixed mathematical formula.

Pseudorandom numbers play a central role as building blocks for generating random variates from all other distributions, as well as realizations of stochastic processes. One of the first arithmetic PRNGs was von Neumann's middle-square method, presented in 1949 and published in 1951. Since the middle-square method, a large research area has developed involving abstract algebra, number theory, computer programming, and software engineering.

We only have the tools to study relatively suboptimal PRNGs, and in practice we should rely on established implementations. A good PRNG should satisfy the following requirements:

1. It should produce numbers which appear independent and uniformly distributed on $[0, 1]$.
2. It should be fast and not require a lot of storage (fulfilled by most PRNGs).

3. It should allow for a **seed** (that is, setting an initial value for reproducibility) for debugging, verification, or producing identical random numbers when comparing different simulated systems.
4. It should allow for producing subsegments of numbers with one starting where the previous one ends; these **streams** should be sufficiently long (which is good for parallel computing).

Definition 9.2 A **linear congruential generator** (**LCG**) is a PRNG recursively defined by

$$x_n = (ax_{n-1} + c) \pmod{m},$$

where $a \in \{1, 2, \ldots, m-1\}$ is the multiplier, $c \in \{0, 1, \ldots, m-1\}$ is the increment, $m \in \mathbb{N}$ is the modulus, and $x_0 \in \{0, 1, \ldots, m-1\}$ is the seed. To obtain pseudorandom numbers in $[0, 1)$, set

$$u_n = \frac{x_n}{m}, \quad n \in \mathbb{N}.$$

Lehmer (1951) introduced multiplicative congruential generators.

Two immediate drawbacks of LCGs, common to all PRNGs, are as follows.

1. The values x_n are not truly random or independent. We can show by induction that

$$x_n = \left(a^n x_0 + c\frac{a^n - 1}{a - 1}\right) \pmod{m},$$

when $a \neq 1$; if $a = 1$, then $x_n = (x_0 + nc) \pmod{m}$. In either case, every x_n is completely determined by a, c, m, and x_0.

2. We have

$$u_n \in \left\{0, \frac{1}{m}, \frac{2}{m}, \ldots, \frac{m-1}{m}\right\},$$

so values in $(0.1/m, 0.9/m)$ never occur, even though under a true uniform distribution this interval would have probability $0.8/m > 0$.

Note that whenever x_n takes a value for the second time, the same sequence is generated again and the cycle repeats.

Definition 9.3 The **period** of a periodic PRNG sequence is the number of states in its repeating cycle. For an LCG, the period may depend on x_0. Since $x_n \in \{0, 1, \ldots, m-1\}$, the period is at most m. If the period is m, then the PRNG has **full period**.

A full period is desirable, because every integer in $\{0, 1, \ldots, m-1\}$ occurs exactly once, contributing to uniformity of the u_n.

How shall we choose a, c, and m to achieve a full period?

Theorem 9.4 (Hull–Dobell theorem) An LCG has full period if and only if the following conditions hold:

1. c and m are coprime.
2. If q is prime and $q \mid m$, then $q \mid a - 1$.
3. If $4 \mid m$, then $4 \mid a - 1$.

Example 9.5 Determine the period of the LCG

$$x_n = (15x_{n-1} + 4) \pmod 7,$$

and compute the first eight numbers starting from $x_0 = 1$.

Solution. We have

$$a = 15, \qquad c = 4 = 2 \cdot 2, \qquad \text{and} \qquad m = 7.$$

It follows that c and m are coprime, $a - 1 = 14 = 2 \cdot 7$ is divisible by all prime factors of m, and m is not divisible by 4. By the Hull–Dobell theorem, the LCG has full period $m = 7$. Starting from $x_0 = 1$, we obtain

$$x_0 = 1, \qquad x_1 = 5, \qquad x_2 = 2, \qquad x_3 = 6,$$
$$x_4 = 3, \qquad x_5 = 0, \qquad x_6 = 4, \qquad \text{and} \qquad x_7 = 1,$$

after which the cycle repeats. □

Besides a full period, good statistical properties are desirable. In particular, the generator should pass stochastic tests of independence and uniformity, such as Marsaglia's (1995) **diehard tests**.

For LCGs, streams can be produced by generating $x_1, \ldots, x_n$ as stream 1, then taking x_{n+1} as the seed of the next stream $x_{n+1}, \ldots, x_{2n}$, and so on.

The most widely used LCGs are **multiplicative congruential generators** (**MCGs**), which have $c = 0$. By the Hull–Dobell theorem, an MCG with $m > 1$ cannot have full period because $\gcd(0, m) = m \neq 1$. However, if a and m are chosen carefully, the period can be $m-1$. Hutchinson (1966) suggested taking m to be the largest prime less than 2^d, where d is the number of bits used to represent a number, for example $d = 31$ for 32- to 64-bit processors. On such architectures, $m = 2^{31} - 1$ is a Mersenne prime. For prime m, we can show that the period is $m - 1$ if a is primitive modulo m, meaning that the smallest positive integer ℓ for which $m \mid a^\ell - 1$ is $\ell = m - 1$. LCGs with m chosen this way are called **prime modulus MCGs** (**PMMCGs**). A choice of a and m with comparably good properties is $(630360016, 2^{31} - 1)$.

Extensions include more general congruences $x_n = g(x_{n-1}, x_{n-2}, \ldots) \pmod m$. Examples include **quadratic congruential generators**, with $g(x_{n-1}) = a_2 x_{n-1}^2 + a_1 x_{n-1} + c$, **multiple recursive generators**, with $g(x_{n-1}, \ldots, x_{n-q}) = a_1 x_{n-1} + \cdots + a_q x_{n-q}$, and congruences where a and c themselves change with n according to congruential formulae.

Composite generators combine two or more separate PRNGs into a new generator with large period, approximately 2^{191} in the example below, a comparatively small seed, excellent statistical properties, and seed advancement methods for constructing streams and substreams separated by large numbers of steps.

Lecture 10 — Thursday, February 05, 2015

The composite generator suggested by L'Ecuyer (1999) and L'Ecuyer et al. (2002) is given, for $n \geq 3$, by

$$\begin{aligned}
x_{n,1} &= (1403580 x_{n-2,1} - 810728 x_{n-3,1}) \pmod{2^{32} - 209} \\
x_{n,2} &= (527612 x_{n-1,2} - 1370589 x_{n-3,2}) \pmod{2^{32} - 22853} \\
x_n &= (x_{n,1} - x_{n,2}) \pmod{2^{32} - 209} \\
u_n &= \begin{cases} \dfrac{x_n}{2^{32} - 208}, & \text{if } x_n > 0, \\ \dfrac{2^{32} - 209}{2^{32} - 208}, & \text{if } x_n = 0. \end{cases}
\end{aligned}$$

The seed is

$$(x_{n-3,1}, x_{n-2,1}, x_{n-1,1}, x_{n-3,2}, x_{n-2,2}, x_{n-1,2}),$$

where $m_1 = 2^{32} - 209$, $m_2 = 2^{32} - 22853$, $x_{i,1} \in \{0, \ldots, m_1 - 1\}$, and $x_{i,2} \in \{0, \ldots, m_2 - 1\}$. The first three seed components may not all be zero, nor may the last three. The R package `parallel` provides the following functionality:

1. `RNGkind("L'Ecuyer-CMRG")` specifies this PRNG.
2. `nextRNGStream()` advances seed by 2^{127} steps.
3. `nextRNGSubStream()` advances the seed by 2^{76} steps.

R's default PRNG is the **Mersenne Twister** of Matsumoto and Nishimura (1998), which has a very long period of $2^{19937} - 1$. Let

$$w = 32, n = 624, m = 397, r = 31.$$

The construction roughly proceeds as follows:

1. Seed: $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n-1} \in \{0, 1\}^w$.
2. Recurrence (in bits):
$$\boldsymbol{x}_{k+n} = \boldsymbol{x}_{k+m} \oplus (\boldsymbol{x}_k^{(u)} \mid \boldsymbol{x}_{k+1}^{(l)})\boldsymbol{A},$$
where $\boldsymbol{A} = \begin{bmatrix} 0 & \boldsymbol{I}_{w-1} \\ & \boldsymbol{a} \end{bmatrix}$ for $\boldsymbol{a} = 9908B0DF_{16}$ in binary representation. Here, $\oplus$ is bitwise XOR, $\mid$ is bitwise concatenation, and $\boldsymbol{x}_k^{(u)}$ and $\boldsymbol{x}_k^{(l)}$ are the upper $w - r$ bits and lower r bits of $\boldsymbol{x}_k$, respectively. The matrix $\boldsymbol{A}$ performs the **twist**; a separate sequence of bit shifts and masks tempers each output word.

There are also constructions of random numbers for other purposes.

Niederreiter (1978) argues that evenness may be more important than statistical randomness in certain applications, such as Monte Carlo integration, where we want to fill the space more uniformly. Such generators are known as **quasirandom number generators** (**QRNGs**).

In cryptographic applications, **CPRNGs** (**cryptographic pseudorandom number generators**) are designed to pass statistical tests while resisting predictive attacks. Even when part of the output sequence is observed, recovering or predicting subsequent values should remain computationally infeasible.

Nonuniform Random Variate Generation

From now on, we assume that a statistically reliable PRNG is available. Our goal is to derive algorithms for **sampling** (i.e., generating random variates) from any other distribution. We aim for **exact** algorithms (producing random variates with exactly the desired distribution) which are **efficient** (in both setup time and execution time) and **robust** (i.e., efficient for all parameter values).

The **inversion method** is the most important method in nonuniform random variate generation. It is based on the following lemma.

Lemma 10.1 (Quantile transform) Let $U \sim \text{Unif}(0,1)$ and let F be any distribution function. Then $X := F^-(U) \sim F$.

Proof.

$$\mathbb{P}(X \leq x) = \mathbb{P}(F^-(U) \leq x) = \mathbb{P}(U \leq F(x)) = F(x), \quad x \in \mathbb{R},$$

where the middle equality is the defining property of the generalized inverse; see Proposition 1, Part 5 of Embrechts and Hofert (2013). □

The inversion method is therefore:

Algorithm 5: Inversion method

1 Generate $U \sim \text{Unif}(0,1)$
2 **return** $X = F^-(U)$

Remark 10.2

1. The inversion method is the only truly universal method; it applies to any distribution function F. As illustrated in Figure 8, the generated value X is more likely to fall in intervals where F is steep:

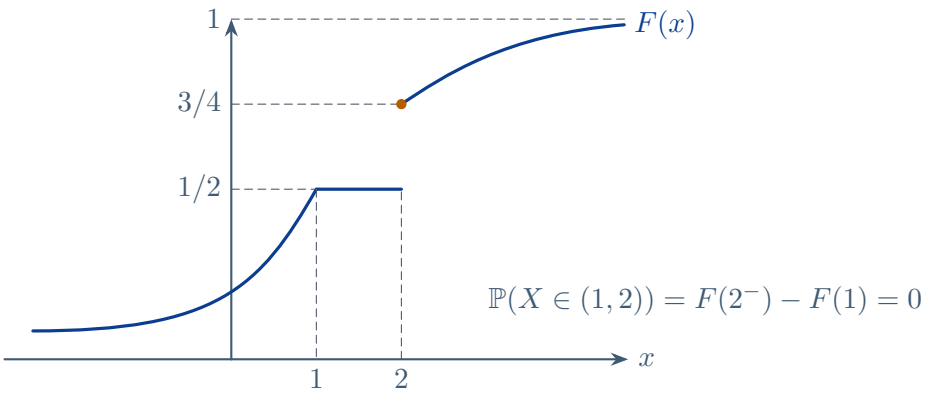

FIGURE 8

2. The R function `RNGkind()` shows that inversion is now used to generate standard normal random variates.

3. If F^- is not explicit, we can solve $F(x) - u = 0$ numerically for x. If $n \gg 1$ variates are required, it can be efficient to first sort $U_{(1)}, \ldots, U_{(n)}$, compute the corresponding ordered values $X_{(i)} = F^-(U_{(i)})$ using root-finding, and then randomly permute $X_{(1)}, \ldots, X_{(n)}$ to obtain $X_1, \ldots, X_n$.

4. For discrete F with jumps at $x_1, x_2, \ldots$ of heights $p_1, p_2, \ldots$, generate $U \sim \text{Unif}(0,1)$ and return

$$X = \begin{cases} x_1, & \text{if } U \in (0, p_1], \\ x_2, & \text{if } U \in (p_1, p_1 + p_2], \\ \vdots & \vdots \\ x_j, & \text{if } U \in \left(\sum_{i=1}^{j-1} p_i, \sum_{i=1}^{j} p_i \right]. \end{cases}$$

Indeed, for all $j \in \mathbb{N}$,

$$\mathbb{P}(X = x_j) = \mathbb{P}\left(\sum_{i=1}^{j-1} p_i < U \leq \sum_{i=1}^{j} p_i \right) = p_j.$$

As a special case, suppose that we have ordered data $x_{(1)} \leq \cdots \leq x_{(n)}$ and want to sample from $\hat{F}_n$. Then all intervals $((j-1)/n, j/n]$ are equally likely, so we return $X = x_{(j)}$ if $U \in ((j-1)/n, j/n]$, or equivalently if $j - 1 < nU \leq j$. Thus

$$X = x_{\lceil nU \rceil}.$$

Repeating this n times corresponds to drawing at random, with replacement, from

$x_1, \ldots, x_n$, which is precisely how we obtain a bootstrap sample from $\hat{F}_n$.

Similarly, to sample $X \sim \text{Unif}\{1, \ldots, n\}$, generate $U \sim \text{Unif}[0, 1)$ and return $X = 1 + \lfloor nU \rfloor$.

Lecture 11 — Tuesday, February 10, 2015

5. If $U_1, \ldots, U_n \overset{\text{iid}}{\sim} \text{Unif}[0, 1]$, then $X_{(i)} = F^-(U_{(i)})$, $i \in \{1, \ldots, n\}$, is a sample of order statistics from F. The ith order statistic can also be sampled by

$$X_{(i)} = F^-(W) \qquad \text{and} \qquad W \sim \text{Beta}(i, n - i + 1),$$

because $U_{(i)} \sim \text{Beta}(i, n - i + 1)$. To see this, let W be a $\text{Beta}(i, n - i + 1)$ random variable. Then

$$\begin{aligned}
\mathbb{P}(U_{(i)} \leq x) &= \mathbb{P}\left(\sum_{j=1}^{n} \mathbb{1}_{\{U_j \leq x\}} \geq i \right) \\
&= \mathbb{P}(\text{Bin}(n, x) \geq i) \\
&= 1 - F_{\text{Bin}(n,x)}(i - 1) \\
&= \int_0^x \frac{1}{B(i, n - i + 1)} z^{i-1} (1 - z)^{n-i} \, \mathrm{d}z \\
&= \mathbb{P}(W \leq x).
\end{aligned}$$

As a special case, when $i = 1$, we have

$$X_{(1)} = F^-(1 - (1 - U)^{1/n}).$$

Since $1 - U \overset{d}{=} U$, we may equivalently return $F^-(1 - U^{1/n})$. Indeed,

$$\mathbb{P}(X_{(1)} > x) = \mathbb{P}(X_1 > x, \ldots, X_n > x) = (1 - F(x))^n.$$

On the other hand, when $i = n$, we have

$$X_{(n)} = F^-(U^{1/n}),$$

since

$$\mathbb{P}(X_{(n)} \leq x) = \mathbb{P}(X_1 \leq x, \ldots, X_n \leq x) = F(x)^n.$$

A "converse" of the quantile transformation is the following:

Lemma 11.1 (Probability integral transform) Let $X \sim F$, where F is continuous. Then $U = F(X) \sim \text{Unif}[0,1]$.

Proof. For $u \in (0,1)$, let $x_u := \sup\{x : F(x) \le u\}$. Continuity of F gives $F(x_u) = u$, and $\{F(X) \le u\}$ agrees with $\{X \le x_u\}$ up to a null set. Therefore

$$\mathbb{P}(U \le u) = \mathbb{P}(F(X) \le u) = \mathbb{P}(X \le x_u) = F(x_u) = u.$$

The endpoint cases are immediate, so $U \sim \text{Unif}[0,1]$. □

Example 11.2 The inversion method gives the following standard samplers.

1. **Weibull distribution**: Here X has distribution function

$$F(x) = 1 - e^{-(x/b)^a}, \quad x \ge 0, \quad a, b \in (0, \infty).$$

Then

$$F(x) = u \iff (x/b)^a = -\log(1-u) \implies F^{-1}(u) = b(-\log(1-u))^{1/a}.$$

Thus we sample $U \sim \text{Unif}(0,1)$ and return

$$X = b(-\log(1-U))^{1/a} = b(-\log U)^{1/a}.$$

Taking $a = 1$ and $b = 1/\lambda$ gives

$$X = -(\log U)/\lambda \sim \text{Exp}(\lambda).$$

2. **Laplace (double exponential) distribution**: Here X has density

$$f(x) = \frac{1}{2b} e^{-|x-\mu|/b}, \quad x \in \mathbb{R}, \quad \mu \in \mathbb{R}, \quad b > 0.$$

Splitting at μ gives

$$F(x) = \begin{cases} \frac{1}{2} e^{(x-\mu)/b}, & x \le \mu, \\ 1 - \frac{1}{2} e^{-(x-\mu)/b}, & x > \mu, \end{cases}$$

and hence

$$F^{-1}(u) = \begin{cases} \mu + b\log(2u), & u \in (0, 1/2], \\ \mu - b\log(2(1-u)), & u \in (1/2, 1). \end{cases}$$

3. Let

$$F(x) = x^2 \frac{e^x - 1}{e - 1}, \quad x \in [0, 1].$$

Then $F(x) = F_1(x)F_2(x)$, where $F_1(x) = x^2$ and $F_2(x) = (e^x - 1)/(e - 1)$. To sample from F, first sample $U_1, U_2 \sim \text{Unif}[0, 1]$, set

$$X_1 = F_1^{-1}(U_1) = \sqrt{U_1} \qquad \text{and} \qquad X_2 = F_2^{-1}(U_2) = \log(1 + U_2(e - 1)),$$

and return $X = \max\{X_1, X_2\}$. Then $X \sim F_1F_2 = F$.

We now consider the generation of random vectors. The goal is to sample $\boldsymbol{X} = (X_1, \ldots, X_d) \sim H$ with margins $X_i \sim F_i$. Some options are:

1. Sample $\boldsymbol{U} = (U_1, \ldots, U_d) \sim \text{Unif}[0, 1]^d$ with independent components, and return

$$\boldsymbol{X} = (F_1^-(U_1), \ldots, F_d^-(U_d)).$$

2. Sample $U \sim \text{Unif}[0, 1]$, set $\boldsymbol{U} = (U, \ldots, U)$, and return

$$\boldsymbol{X} = (F_1^-(U), \ldots, F_d^-(U)).$$

3. For $d = 2$, sample $\boldsymbol{U} = (U, 1 - U)$ and return

$$\boldsymbol{X} = (F_1^-(U), F_2^-(1 - U)).$$

In all three cases, $\boldsymbol{U}$ has Unif$[0, 1]$ margins, so $\boldsymbol{X}$ has the desired margins $F_1, \ldots, F_d$. The dependence structure is different, however: we have independence in the first case, the strongest positive dependence in the second, and the strongest negative dependence in the third.

Definition 11.3 A **copula** is the distribution function of a random vector $\boldsymbol{U}$ whose margins are all Unif$[0, 1]$.

The copulas corresponding to the three cases above are as follows.

For independent components,

$$C(u_1, \ldots, u_d) = \mathbb{P}(U_1 \leq u_1, \ldots, U_d \leq u_d) = \prod_{i=1}^{d} u_i,$$

which is the **independence copula**. For strongest positive dependence,

$$C(u_1, \ldots, u_d) = \mathbb{P}(U_1 \leq u_1, \ldots, U_d \leq u_d) = \min\{u_1, \ldots, u_d\},$$

which is the **upper Fréchet–Hoeffding bound**. For strongest negative dependence in two dimensions,

$$\begin{aligned} C(u_1, u_2) &= \mathbb{P}(U_1 \leq u_1, 1 - U_1 \leq u_2) \\ &= \mathbb{P}(U_1 \leq u_1, U_1 \geq 1 - u_2) \\ &= \max\{0, u_1 - (1 - u_2)\} \\ &= \max\{0, u_1 + u_2 - 1\}, \end{aligned}$$

which is the **lower Fréchet–Hoeffding bound**.

Copulas are widely used in finance and insurance to model dependence between components of random vectors. The central result in the theory of copulas is Sklar's theorem, which dates to 1959.

Theorem 11.4 (Sklar's theorem)

1. Decomposition: for any distribution function H with margins $F_1, \ldots, F_d$, there exists a copula C such that
$$H(\boldsymbol{x}) = C(F_1(x_1), \ldots, F_d(x_d)) \tag{11.1}$$
for every $\boldsymbol{x} \in \mathbb{R}^d$. If $F_1, \ldots, F_d$ are continuous, then C is uniquely determined by
$$C(\boldsymbol{u}) = H(F_1^-(u_1), \ldots, F_d^-(u_d)), \qquad \boldsymbol{u} \in (0,1)^d.$$
Its boundary values are then determined by the copula margins and continuity.
2. Composition: given any copula C and marginals $F_1, \ldots, F_d$, H as defined by (11.1) is a distribution function with marginals $F_1, \ldots, F_d$.

Proof (for continuous margins). For continuous $F_1, \ldots, F_d$, let $\boldsymbol{X} \sim H$, define

$$\boldsymbol{U} := (F_1(X_1), \ldots, F_d(X_d)),$$

and denote the distribution function of $\boldsymbol{U}$ by C. By the probability integral transform, C is a copula. Also,

$$\begin{aligned} H(\boldsymbol{x}) &= \mathbb{P}(X_1 \le x_1, \ldots, X_d \le x_d) \\ &= \mathbb{P}(F_1^-(U_1) \le x_1, \ldots, F_d^-(U_d) \le x_d) \\ &= \mathbb{P}(U_1 \le F_1(x_1), \ldots, U_d \le F_d(x_d)) \\ &= C(F_1(x_1), \ldots, F_d(x_d)) \end{aligned}$$

for every $\boldsymbol{x} \in \mathbb{R}^d$. Hence C is a copula satisfying (11.1). Furthermore,

$$C(\boldsymbol{u}) = H(F_1^-(u_1), \ldots, F_d^-(u_d)), \qquad \boldsymbol{u} \in (0,1)^d.$$

Conversely, let $\boldsymbol{U} \sim C$ and define $\boldsymbol{X} := (F_1^-(U_1), \ldots, F_d^-(U_d))$. Then

$$\begin{aligned} \mathbb{P}(\boldsymbol{X} \le \boldsymbol{x}) &= \mathbb{P}(F_1^-(U_1) \le x_1, \ldots, F_d^-(U_d) \le x_d) \\ &= \mathbb{P}(U_1 \le F_1(x_1), \ldots, U_d \le F_d(x_d)) \\ &= C(F_1(x_1), \ldots, F_d(x_d)) \\ &= H(\boldsymbol{x}). \end{aligned}$$

Thus H defined by (11.1) is a distribution function, namely that of $\boldsymbol{X}$. □

Lecture 12 — Tuesday, February 17, 2015

Remark 12.1 Sklar's theorem decomposes the distribution function

$$H(x_1, \ldots, x_d) = C(F_1(x_1), \ldots, F_d(x_d))$$

into its marginal behaviour and its dependence structure. This has two complementary uses.

1. The probability integral transform lets us study dependence separately from the mar-

gins. This is useful for estimation, goodness-of-fit testing, and reducing the dimension of the parameter space in numerical work.

2. The quantile transform lets us construct multivariate models with prescribed margins and a chosen dependence structure. This is especially useful in stress testing, where the marginal losses and their dependence can be modelled separately.

Example 12.2 Let (Z_1, Z_2) have a bivariate standard normal distribution with correlation ρ. Then

$$\boldsymbol{U} = (\Phi(Z_1), \Phi(Z_2))$$

has uniform margins. The distribution function of $\boldsymbol{U}$ is called the **Gaussian copula**. If F_1 and F_2 are any desired marginal distribution functions, then the quantile transformation

$$\boldsymbol{X} = \left(F_1^-(\Phi(Z_1)), F_2^-(\Phi(Z_2))\right)$$

has margins F_1 and F_2, while retaining the dependence structure induced by the bivariate normal distribution.

The Rejection and Composition Methods

The **rejection method** samples from a target density f by first sampling from a proposal density g that is easier to generate, provided that the proposal dominates the target up to a finite constant.

The proof uses the continuous analogue of the law of total probability: if Y has density g, then

$$\mathbb{P}(X \le x) = \int \mathbb{P}(X \le x \mid Y = y) g(y)\, \mathrm{d}y,$$

whenever the conditional probabilities are well-defined.

Lemma 12.3 Let f and g be densities on $\mathbb{R}$ such that

$$f(x) \le c g(x), \quad x \in \mathbb{R},$$

for some $c \ge 1$. Let $Y \sim g$ and $U \sim \mathrm{Unif}[0,1]$ be independent, and interpret $f(y)/g(y)$ as 0 when $g(y) = 0$; the domination condition ensures that $f(y) = 0$ there. Conditional on the event

$$U c g(Y) \le f(Y),$$

the accepted value $X := Y$ has density f.

Proof.

$$\mathbb{P}(X \le x) = \mathbb{P}(Y \le x \mid Ucg(Y) \le f(Y)) = \frac{\mathbb{P}(Y \le x,\ Ucg(Y) \le f(Y))}{\mathbb{P}(Ucg(Y) \le f(Y))}.$$

The denominator is

$$\mathbb{P}\left(U \le \frac{f(Y)}{cg(Y)}\right) = \int_{-\infty}^{\infty} \mathbb{P}\left(U \le \frac{f(y)}{cg(y)}\right) g(y)\,\mathrm{d}y = \int_{-\infty}^{\infty} \frac{f(y)}{cg(y)} g(y)\,\mathrm{d}y = \frac{1}{c}.$$

The numerator is

$$\int_{-\infty}^{x} \mathbb{P}\left(U \le \frac{f(y)}{cg(y)}\right) g(y)\,\mathrm{d}y = \frac{1}{c}\int_{-\infty}^{x} f(y)\,\mathrm{d}y = \frac{1}{c}F(x).$$

Therefore $\mathbb{P}(X \le x) = F(x)$. □

Algorithm 6: Rejection method

```
repeat
    Generate independent Y ~ g and U ~ Unif[0, 1]
until Ucg(Y) ≤ f(Y)
return X = Y
```

Remark 12.4 Let N be the number of proposals required to obtain one accepted value. Since the acceptance probability is

$$p = \mathbb{P}(Ucg(Y) \le f(Y)) = \frac{1}{c},$$

we have $N \sim \text{Geom}(p)$ and $\mathbb{E}[N] = c$. Thus the constant c measures the expected computational cost per accepted draw.

The rejection method only requires evaluation of f, not direct simulation from it. In some applications, however, evaluating f is itself expensive. Marsaglia's (1977) **squeeze principle** reduces the number of evaluations of f by using fast lower and upper bounds.

Suppose

$$0 \le h_1(x) \le f(x) \le h_2(x) \le cg(x), \quad x \in \mathbb{R},$$

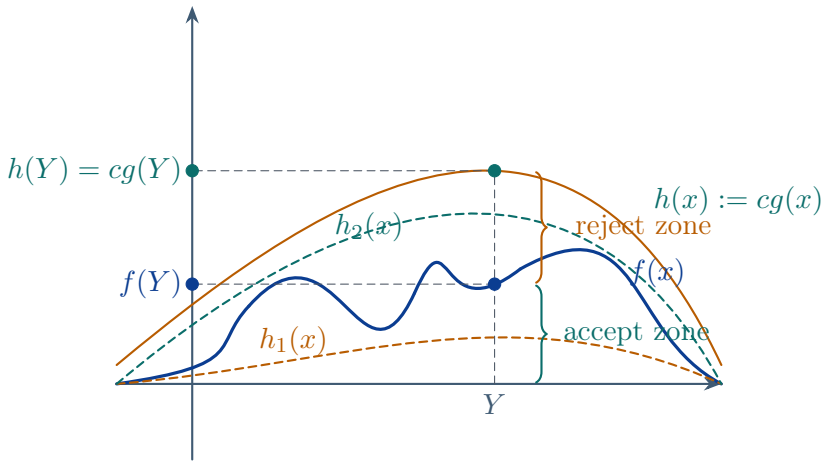

FIGURE 9

where h_1 and h_2 are quick to evaluate. They need not be proper densities or integrate to one. After generating $Y \sim g$ and $U \sim \text{Unif}[0,1]$, immediately accept if

$$Ucg(Y) \le h_1(Y).$$

If this fails, evaluate the target density only when

$$Ucg(Y) \le h_2(Y),$$

and accept precisely when $Ucg(Y) \le f(Y)$. The expected number of proposals remains c, but the expected number of evaluations of f becomes

$$\mathbb{E}[N_f] = \int_{-\infty}^{\infty} (h_2(y) - h_1(y))\,\mathrm{d}y.$$

Indeed,

$$N_f = \sum_{i=1}^{N} \mathbb{1}_{\{h_1(Y_i) < U_i cg(Y_i) \le h_2(Y_i)\}},$$

and Wald's equation gives

$$\mathbb{E}[N_f] = c\,\mathbb{P}(h_1(Y) < Ucg(Y) \le h_2(Y)) = c\int_{-\infty}^{\infty} \frac{h_2(y) - h_1(y)}{cg(y)} g(y)\,\mathrm{d}y = \int_{-\infty}^{\infty} (h_2(y) - h_1(y))\,\mathrm{d}y.$$

If $h_1 = 0$ and $h_2 = cg$, then this reduces to the ordinary rejection method.

Figure 9 depicts the corresponding accept and reject zones for a proposed value Y.

Example 12.5 Suppose we want to sample from a density f. Define

$$A_f = \{(u, v) : 0 \le u \le \sqrt{f(v/u)}\}.$$

If (U, V) is uniformly distributed on A_f, then $X = V/U$ has density f. This is the **ratio-of-uniforms** principle.

Proof. Let $(U, V) \sim \text{Unif}(A_f)$. Then

$$\mathbb{P}(V/U \le x) = \frac{\text{Area}(A_f \cap \{(u, v) : v/u \le x\})}{\text{Area}(A_f)}.$$

Use the change of variables $(u, t) \mapsto (u, v) = (u, tu)$, so that $t = v/u$ and the Jacobian is u. Then

$$\text{Area}(A_f) = \int_{-\infty}^{\infty} \int_0^{\sqrt{f(t)}} u \, \mathrm{d}u \, \mathrm{d}t = \frac{1}{2} \int_{-\infty}^{\infty} f(t) \, \mathrm{d}t = \frac{1}{2}.$$

Similarly,

$$\text{Area}(A_f \cap \{v/u \le x\}) = \int_{-\infty}^{x} \int_0^{\sqrt{f(t)}} u \, \mathrm{d}u \, \mathrm{d}t = \frac{1}{2} F(x).$$

Dividing the two areas gives $\mathbb{P}(V/U \le x) = F(x)$. □

Algorithm 7: Ratio of uniforms

Provided A_f is bounded, find a rectangle $[u_-, u_+] \times [v_-, v_+]$ containing it
repeat
  Generate independent $U \sim \text{Unif}[u_-, u_+]$ and $V \sim \text{Unif}[v_-, v_+]$
until $0 \le U$ *and* $U^2 \le f(V/U)$
return $X = V/U$

Lecture 13 — Thursday, February 19, 2015

How do we design a good rejection algorithm? The proposal density g should be easy to sample from, and the envelope constant c should be as small as possible (there is usually a tradeoff between the simplicity of g and the size of c). Since $\mathbb{E}[N] = c$, lowering c directly lowers the expected number of proposals.

Given acceptance, the residual variable

$$U' := \frac{Ucg(Y)}{f(Y)}$$

is again uniformly distributed on $[0,1]$. In exact arithmetic, this residual uniform can be recycled when generating later random variates. In finite-precision arithmetic this recycling is usually avoided because it can introduce subtle dependence and numerical artefacts.

The same rejection idea applies to discrete distributions by replacing densities with probability mass functions.

Example 13.1 The following are standard uses of the rejection method.

1. If f is a density on $[a,b]$ and $f(x) \le M$ for all $x \in [a,b]$, take g to be the uniform density on $[a,b]$ and take $c = M(b-a)$. Then

   $$cg(y) = M(b-a)\frac{1}{b-a}\mathbb{1}_{[a,b]}(y) = M\mathbb{1}_{[a,b]}(y).$$

   Generate $Y = a + (b-a)U'$ with $U' \sim \text{Unif}[0,1]$, generate $U \sim \text{Unif}[0,1]$, and accept precisely when $UM \le f(Y)$.

2. For the standard normal density, use the double-exponential proposal

   $$g(x) = \frac{1}{2}e^{-|x|}, \quad x \in \mathbb{R}.$$

   Since

   $$0 \le \frac{1}{2}(|x|-1)^2 = \frac{1}{2}x^2 - |x| + \frac{1}{2},$$

   we have

   $$\frac{1}{\sqrt{2\pi}}e^{-x^2/2} \le \frac{1}{\sqrt{2\pi}}e^{1/2}e^{-|x|} = cg(x) \qquad \text{and} \qquad c = \sqrt{\frac{2e}{\pi}}.$$

   Thus we can generate $Y = \text{sign}(2U''-1)(-\log U')$ with $U', U'' \sim \text{Unif}[0,1]$, and accept if

   $$U \le \exp\left(-\frac{1}{2}(|Y|-1)^2\right).$$

   Equivalently, first generate the magnitude $R = -\log U'$, accept it when

   $$U \le \exp\left(-\frac{1}{2}(R-1)^2\right),$$

and only then attach an independent random sign.

3. More generally, if $f(x) = c\psi(x)g(x)$ for some function $0 \le \psi \le 1$, then $f \le cg =: h$ and the acceptance condition $Ucg(Y) \le f(Y)$ is simply $U \le \psi(Y)$. As a simple gamma-type example, consider the density on $(0, 1]$ given by

$$f(x) = c\big(\alpha x^{\alpha-1}\big)e^{-x}, \qquad \alpha > 0,$$

where c is the normalizing constant. Take $g(x) = \alpha x^{\alpha-1}$ and $\psi(x) = e^{-x}$. Generate $U, U' \overset{\text{iid}}{\sim} \text{Unif}[0, 1]$, set $Y = (U')^{1/\alpha}$, and accept Y if $U \le e^{-Y}$. In Văduva's modification, we instead generate $E \sim \text{Exp}(1)$ and accept when $E \ge Y$, which is equivalent because $e^{-E} \sim \text{Unif}[0, 1]$.

The **composition method** is the other general method in this group. It applies when the desired distribution can be written as a mixture of simpler distributions. We use the notation

$$X \mid Z = k \sim F_k.$$

Lemma 13.2 Let $\mathbb{P}(Z = k) = p_k$ for $k \in \mathbb{N}$, and let each F_k be a distribution function. If, conditional on $Z = k$, we have $X \sim F_k$, then X has distribution function

$$F(x) = \sum_{k=1}^{\infty} p_k F_k(x).$$

Proof. By conditioning on Z,

$$\mathbb{P}(X \le x) = \sum_{k=1}^{\infty} \mathbb{P}(X \le x \mid Z = k)\mathbb{P}(Z = k) = \sum_{k=1}^{\infty} p_k F_k(x).$$

□

Algorithm 8: Composition method

1 Generate $Z \in \mathbb{N}$ with probability mass function $(p_k)_{k\in\mathbb{N}}$
2 Given $Z = k$, generate $X \sim F_k$
3 **return** X

Definition 13.3 A distribution of the form

$$F(x) = \sum_{k=1}^{\infty} p_k F_k(x)$$

is called a **mixture**. If F_k has density f_k for every k, then the mixture has density

$$f(x) = \sum_{k=1}^{\infty} p_k f_k(x).$$

Partitions provide a useful source of mixtures. Let $A_1, \ldots, A_K$ be a partition of $\mathbb{R}$ and define

$$p_k = \int_{A_k} f(x)\,\mathrm{d}x = \mathbb{P}(X \in A_k).$$

When $p_k > 0$,

$$f_k(x) = \frac{f(x)\mathbb{1}_{A_k}(x)}{p_k}$$

is the conditional density of X given $X \in A_k$, and

$$f(x) = \sum_{k=1}^{K} p_k f_k(x).$$

This representation is especially helpful when different parts of the support have different numerical difficulties, such as unbounded peaks or heavy tails.

Suppose that, on each stratum A_k, we have an envelope h_k for the unnormalized density $f(x)\mathbb{1}_{A_k}(x)$, and set

$$q_k := \int_{A_k} h_k(x)\,\mathrm{d}x.$$

There are then two natural ways to combine composition and rejection.

Algorithm 9: Composition with rejection

1 Generate $Z \in \{1, \ldots, K\}$ with probabilities $(p_k)_{k=1}^K$
2 **repeat**
3 Given $Z = k$, generate Y from density h_k/q_k on A_k and generate $U \sim \mathrm{Unif}[0,1]$ independently
4 **until** $Uh_Z(Y) \le f(Y)$
5 **return** $X = Y$

Alternatively, we can first use composition to sample from the global envelope

$$H(x) = \sum_{k=1}^{K} h_k(x) \mathbb{1}_{A_k}(x).$$

The normalizing constant of this envelope is $\sum_{k=1}^{K} q_k$.

Algorithm 10: Rejection with a composed envelope

repeat
  Generate $Z \in \{1, \dots, K\}$ with probabilities proportional to $(q_k)_{k=1}^{K}$
  Given $Z = k$, generate Y from density h_k/q_k and generate $U \sim \text{Unif}[0,1]$ independently
until $U h_Z(Y) \le f(Y)$
return $X = Y$

Polynomial densities on $[0,1]$ provide a clean example. If

$$f(x) = \sum_{k=0}^{K} c_k x^k, \quad 0 \le x \le 1,$$

with $c_k \ge 0$ and f normalized, then

$$F(x) = \sum_{k=0}^{K} \frac{c_k}{k+1} x^{k+1} = \sum_{k=1}^{K+1} \frac{c_{k-1}}{k} x^k.$$

Since $F(1) = 1$, the numbers

$$p_k = \frac{c_{k-1}}{k}, \quad k = 1, \dots, K+1,$$

are mixture weights, and the corresponding component distribution functions are $F_k(x) = x^k$.

Lecture 14 — Tuesday, February 24, 2015

Algorithm 11: Polynomial density by composition

Generate $Z \in \{1, \dots, K+1\}$ with probabilities $p_k = c_{k-1}/k$
Generate $U \sim \text{Unif}[0,1]$
return $X = U^{1/Z}$, *which has conditional distribution function* $F_Z(x) = x^Z$

Bignami and de Matteis (1971) also proposed a rejection approach for signed expansions. Suppose

$$f(x) = \sum_{n=1}^{\infty} p_n f_n(x) \qquad \text{and} \qquad \sum_{n=1}^{\infty} p_n = 1,$$

where the coefficients need not all be nonnegative. Writing $p_n^+ = \max\{p_n, 0\}$ gives

$$f(x) \le \sum_{n=1}^{\infty} p_n^+ f_n(x) = c \sum_{n=1}^{\infty} \frac{p_n^+}{c} f_n(x) \qquad \text{and} \qquad c = \sum_{n=1}^{\infty} p_n^+.$$

The expression in parentheses is a proper mixture density and can therefore be used as the proposal in a rejection algorithm.

Example 14.1

1. To sample from
$$F(x) = x\left(\frac{1}{6} + \frac{1}{2}x + \frac{1}{3}x^2\right), \quad 0 \le x \le 1,$$
write
$$F(x) = \frac{1}{6}x + \frac{1}{2}x^2 + \frac{1}{3}x^3.$$
Thus F is a mixture of $F_1(x) = x$, $F_2(x) = x^2$, and $F_3(x) = x^3$ with weights $1/6$, $1/2$, and $1/3$. Generate $Z \in \{1, 2, 3\}$ with these probabilities, generate $U \sim \text{Unif}[0, 1]$, and return $X = U^{1/Z}$.

2. For
$$f(x) = \frac{3}{4}(1 - x^2)\mathbb{1}_{[-1,1]}(x) \le \frac{3}{4}\mathbb{1}_{[-1,1]}(x) = \frac{3}{2}\left(\frac{1}{2}\mathbb{1}_{[-1,1]}(x)\right).$$
rejection with the uniform density on $[-1, 1]$ gives envelope constant $c = 3/2$.

The Acceptance-Complement Method, Table Lookups, and Alias Methods

The **acceptance-complement method** is a hybrid of composition and rejection. It applies when the target density can be split into two parts, one of which is dominated by a tractable proposal density, while the other part is tractable in its own right.

Lemma 14.2 Suppose $f = f_1 + f_2$, where f is a density, f_1 and f_2 are nonnegative, and $f_1(x) \le g(x)$ for some density g. Interpret f_1/g as zero wherever $g = 0$, and let

$$p = \int_{-\infty}^{\infty} f_2(x)\,\mathrm{d}x.$$

Assume $p > 0$. Let $Y \sim g$ and $U \sim \mathrm{Unif}[0,1]$ be independent. If $U \le f_1(Y)/g(Y)$, return $X = Y$; otherwise generate and return X from density f_2/p. Then X has density f.

Proof. The probability of returning an accepted proposal in $(-\infty, x]$ is

$$\int_{-\infty}^{x} \mathbb{P}\left(U \le \frac{f_1(y)}{g(y)}\right) g(y)\,\mathrm{d}y = \int_{-\infty}^{x} f_1(y)\,\mathrm{d}y.$$

The probability of returning a value from the complementary part in $(-\infty, x]$ is

$$\mathbb{P}\left(U > \frac{f_1(Y)}{g(Y)}\right) \int_{-\infty}^{x} \frac{f_2(y)}{p}\,\mathrm{d}y = p \int_{-\infty}^{x} \frac{f_2(y)}{p}\,\mathrm{d}y = \int_{-\infty}^{x} f_2(y)\,\mathrm{d}y.$$

Adding the two contributions gives

$$\mathbb{P}(X \le x) = \int_{-\infty}^{x} f(y)\,\mathrm{d}y.$$

□

Algorithm 12: Acceptance-complement method

```
Generate independent Y ~ g and U ~ Unif[0,1]
if U ≤ f1(Y)/g(Y) then
    return X = Y
else
    Generate X from density f2/p
    return X
```

Remark 14.3

1. For any measurable $A \subseteq \mathbb{R}$ with $q := \int_A f(y)\,\mathrm{d}y \in (0,1)$, we can take

$$f_1(x) = f(x)\mathbb{1}_A(x) \leq \frac{f_1(x)}{q} =: g(x)$$

and $f_2(x) = f(x)\mathbb{1}_{A^c}(x)$, so the acceptance-complement method always applies. The choice of A is a tuning parameter that can be used to optimize the algorithm.

2. Desirable properties: g is easy to sample, f_1/g is easy to evaluate, and f_2/p is easy to sample and evaluate when p is large. The method is most useful when f_1 is close to g, so that the acceptance probability is high and the complementary part is small.

3. Devroye's account of the acceptance-complement method traces the following particularly natural split to Ahrens and Dieter (1981):

$$f_1(x) = \min\{f(x), g(x)\} \qquad \text{and} \qquad f_2(x) = \max\{f(x) - g(x), 0\},$$

where g is chosen to be close to f. As with ordinary rejection, a squeeze step can be used to avoid expensive evaluations of the target density.

The **table lookup method** applies when the support is finite and the probabilities are rational. If X takes values in $\{x_1, \ldots, x_n\}$ and $p_k = m_k/M$ with $m_k \in \mathbb{N}$ and $\sum_k m_k = M$, then sampling can be reduced to a uniform lookup in a table of length M.

Algorithm 13: Table lookup method

1 Create a vector $\boldsymbol{t}$ of length M containing m_k copies of x_k for each k
2 Generate $N \sim \text{Unif}\{1, \ldots, M\}$
3 **return** $\boldsymbol{t}[N]$

The table lookup method was first presented by Marsaglia (1963). The advantage is that sampling is very fast after setup, with $O(1)$ lookup cost. The disadvantages are the setup cost, the fact that X needs to have compact support, the potentially large table, and the need to approximate irrational probabilities in finite-precision arithmetic. For unbounded support, table lookup must be combined with a separate method for the tails. Marsaglia, Tsang, and Wang (2004) introduced the **condensed table lookup method** to reduce storage, which is best described by example.

Example 14.4 Consider $X \in \{x_1, x_2, x_3, x_4\}$ with probabilities

$$p_1 = 0.2245, \qquad p_2 = 0.1221, \qquad p_3 = 0.5452, \qquad \text{and} \qquad p_4 = 0.1082.$$

A direct table lookup with $M = 10\,000$ would contain 2245 copies of x_1, 1221 copies of x_2, 5452 copies of x_3, and 1082 copies of x_4, so it requires storage for all 10 000 entries.

Since the algorithm only indexes the table at random, we can store a condensed run-length representation such as

$$\begin{aligned}\boldsymbol{t} = (&\mathrm{rep}(x_1, 200), \mathrm{rep}(x_2, 100), \mathrm{rep}(x_3, 300), \mathrm{rep}(x_4, 300),\\ &\mathrm{rep}(x_1, 200), \mathrm{rep}(x_2, 200), \mathrm{rep}(x_3, 400), \mathrm{rep}(x_4, 40),\\ &\mathrm{rep}(x_2, 20), \mathrm{rep}(x_3, 50), \mathrm{rep}(x_4, 30), x_1, x_1, x_1, x_1, x_1, x_2, x_3, x_3, x_4, x_4, \dots).\end{aligned}$$

Here $\mathrm{rep}(x, m)$ denotes m consecutive copies of x. The condensed table lookup then chooses a random index by successive digits and resolves that index through the corresponding shorter table.

Lecture 15 — Thursday, February 26, 2015

The condensed table lookup method stores repeated values in several smaller tables, typically corresponding to digits in a chosen base, instead of storing one very long lookup table. For instance, a table with many repeated entries such as

$$t = (\dots, \underbrace{x_1, \dots, x_1}_{5 \text{ times}}, x_2, x_3, x_3, x_4, x_4, x_4)$$

can be represented by shorter tables that are selected according to successive digits of the random index.

Algorithm 14: Condensed table lookup

For the example above, store the table in runs

$$(v_1, m_1), (v_2, m_2), \cdots = (x_1, 200), (x_2, 100), (x_3, 300), (x_4, 300), (x_1, 200), \dots,$$

where each pair means "m_j consecutive copies of v_j"

Precompute cumulative run lengths $s_j = \sum_{i=1}^{j} m_i$

Generate $N \sim \mathrm{Unif}\{1, \dots, 10\,000\}$

Find the first run index J such that $s_J \geq N$

return v_J

Remark 15.1

1. The setup cost is linear in the number of runs, and the storage requirement is proportional to the number of runs rather than the full length M. In this example, each run stores a value and a count, not every individual table entry.
2. Sampling remains exact as long as the run-length representation matches the original lookup table. If the probabilities are rounded to a denominator such as 10 000, then the method samples exactly from that rounded distribution.
3. The run index J can be found either by sequential scan or by binary search over (s_j). Sequential scan is often acceptable when the number of runs is small, while binary search is preferable when many runs are present.

The **alias method** is another method for distributions with finite support. Let $X \sim F$ be discrete with probability mass function $(p_k)_{k=1}^K$. Walker (1974) and Walker (1977) showed that the probabilities can be represented as

$$p_k = \frac{1}{K}\sum_{j=1}^{K}(q_j \mathbb{1}_{\{j=k\}} + (1-q_j)\mathbb{1}_{\{a_j=k\}}),$$

for $k \in \{1,\dots,K\}$, where $a_j \in \{1,\dots,K\}$ are **aliases** and $q_j \in [0,1]$ are **cutoff values**. For a proof, see Devroye (1986, p. 107). Without loss of generality, we may arrange the representation so that the primary outcome in the jth column is j, which gives

$$p_k = \frac{1}{K}q_k + \frac{1}{K}\sum_{j=1}^{K}(1-q_j)\mathbb{1}_{\{a_j=k\}}.$$

Kronmal and Peterson (1979) gave an $O(K)$ algorithm for computing the cutoffs $(q_k)_{k=1}^K$ and aliases $(a_k)_{k=1}^K$.

Algorithm 15: Alias method

In the setup stage, compute $(q_k)_{k=1}^K$ and $(a_k)_{k=1}^K$
Generate $U' \sim \text{Unif}[0,1)$
Set $I = 1 + \lfloor KU' \rfloor$ and $U = KU' - \lfloor KU' \rfloor$
if $U \le q_I$ **then**
    return I
else
    return a_I

Remark 15.2

1. The alias method has the geometric interpretation in Figure 10, which partitions the unit square into K vertical strips:

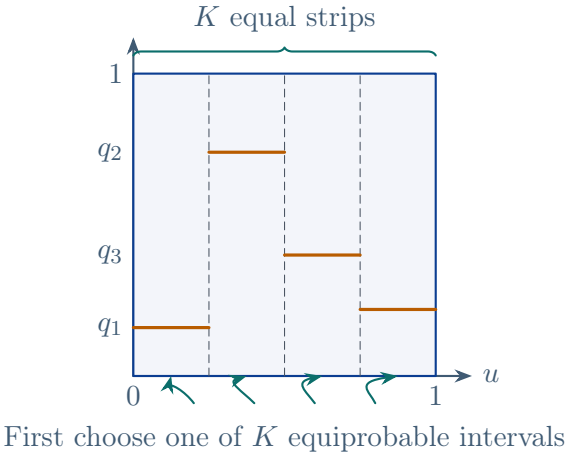

FIGURE 10

   It can also be generalized to other partitions of $[0,1]^2$.

2. If the desired support is $\{x_1, \ldots, x_K\}$ rather than $\{1, \ldots, K\}$, first sample the index by the alias method and then return the corresponding support value.

Sampling Specific Discrete Distributions

Production simulation libraries usually use distribution-specific generation methods. The following list records the main ideas behind the methods used in R for several standard discrete families.

1. A Unif$\{1, \ldots, n\}$ variate can be generated as $1 + \lfloor nU \rfloor$ for $U \sim \text{Unif}[0,1)$. In R, `sample(1:n, size=..., replace=TRUE)` uses the same idea. With the `prob` argument, `sample()` uses a variant of the alias method adapted from Ripley (1987, Algorithm 3.13B).

2. For the hypergeometric distribution, R uses the method of Kachitvichyanukul and Schmeiser (1985): the support is split into three intervals, with rejection steps based on one uniform and two exponential variates.

3. For the Bin(n, p) distribution, R uses an adapted version of Kachitvichyanukul and Schmeiser

(1988). If $np < 30$, the method uses inversion with recursively computed probabilities. If $np \geq 30$, it partitions the support into a left tail, body, and right tail, using exponential envelopes for the tails and a hat envelope with a squeeze step for the body.

4. For the $\mathrm{Geom}(p)$ distribution on $\mathbb{N}_0$, where $0 < p < 1$, a useful representation is

$$X = \lfloor Y \rfloor \qquad \text{and} \qquad Y \sim \mathrm{Exp}(-\log(1-p)).$$

Indeed, for $k \in \mathbb{N}_0$,

$$\mathbb{P}(X = k) = \mathbb{P}(k \leq Y < k+1) = F_Y(k+1) - F_Y(k) = (1-p)^k - (1-p)^{k+1} = p(1-p)^k.$$

5. For the $\mathrm{NegBin}(r, p)$ distribution, where $r > 0$ and $0 < p < 1$, R uses the Gamma–Poisson mixture representation. Let

$$Y \sim \mathrm{Gamma}\left(r, \frac{p}{1-p}\right) \qquad \text{and} \qquad X \mid Y = y \sim \mathrm{Pois}(y).$$

Then

$$\begin{aligned}
\mathbb{P}(X = k) &= \int_0^\infty \frac{e^{-y} y^k}{k!} \frac{1}{\Gamma(r)} \left(\frac{p}{1-p}\right)^r y^{r-1} e^{-py/(1-p)} \, \mathrm{d}y \\
&= \frac{1}{k!\Gamma(r)} \left(\frac{p}{1-p}\right)^r \int_0^\infty y^{k+r-1} e^{-y/(1-p)} \, \mathrm{d}y \\
&= \frac{\Gamma(k+r)}{k!\Gamma(r)} p^r (1-p)^k \\
&= \binom{k+r-1}{k} p^r (1-p)^k,
\end{aligned}$$

with the usual generalized binomial interpretation when r is not an integer. Here the second Gamma parameter is the rate.

6. For the $\mathrm{Pois}(\lambda)$ distribution, R uses an algorithm of Ahrens and Dieter (1982). For $\lambda < 10$, it uses inversion based on table lookup. For $\lambda \geq 10$, it uses rejection with a truncated normal envelope and a squeeze step.

Lecture 16 — Tuesday, March 03, 2015

Sampling Specific Continuous Distributions

For continuous distributions, the methods are more varied and often more complex. The following list records the main ideas behind the methods used in R for several standard continuous families.

1. For the Unif$[a, b]$ distribution, R uses $X = a + (b - a)U$ for $U \sim$ Unif$[0, 1]$ via the Mersenne Twister.
2. For the $\mathcal{N}(\mu, \sigma^2)$ distribution, R uses $X = \mu + \sigma Z$, $Z \sim \mathcal{N}(0, 1)$ via inversion.
3. For the Lognormal(μ, σ^2) distribution, R uses the stochastic representation $X = e^Y$, $Y \sim \mathcal{N}(\mu, \sigma^2)$.
4. For the t_ν distribution, R uses the stochastic representation $X = Z/\sqrt{Y/\nu}$ for independent $Z \sim \mathcal{N}(0, 1)$, $Y \sim \chi^2_\nu$, $\nu > 0$.
5. For the Exp(λ) distribution, via inversion we have $X = (-\log(U))/\lambda$ for $U \sim$ Unif$[0, 1]$. R uses a partition into intervals with table lookup and rejections based on polynomial envelopes. See Ahrens and Dieter (1972, Algorithm 3A).
6. For the Gamma$(\alpha, 1)$ distribution, for $\alpha \in (0, 1)$, R uses rejection with envelope:

$$f(x) = \frac{1}{\Gamma(\alpha)} x^{\alpha-1} e^{-x} \leq \begin{cases} \dfrac{x^{\alpha-1}}{\Gamma(\alpha)}, & x \in [0, 1] \\ \dfrac{e^{-x}}{\Gamma(\alpha)}, & x > 1 \end{cases}$$

   See Ahrens and Dieter (1974). For $\alpha \in [1, \infty)$, R uses rejection based on a normal envelope with a squeeze step; see Ahrens and Dieter (1982).
7. For the χ^2_ν distribution, R uses the stochastic representation $X \sim$ Gamma$(\nu/2, 2)$. Here 2 is a scale parameter.

Simulating Poisson Processes

Definition 16.1 A **homogeneous Poisson process** (**HPP**) $\{N_t\}_{t\geq 0}$ with intensity $\lambda > 0$ is a counting process satisfying the following properties:

1. $N_0 = 0$.
2. For any
$$0 \leq t_0 < t_1 < \cdots < t_n < \infty,$$
the increments $N_{t_1} - N_{t_0}, \ldots, N_{t_n} - N_{t_{n-1}}$ are independent.
3. The increments are Poisson: $N_{t+h} - N_t \sim \text{Pois}(\lambda h)$ for all $t, h \geq 0$.
4. The sample paths are almost surely right-continuous with left limits, and N_t increases only by jumps of size one.

In particular,
$$\mathbb{P}(N_{t+h} - N_t = k) = \frac{(\lambda h)^k}{k!} e^{-\lambda h}, \quad k \in \mathbb{N}_0, \quad h \geq 0.$$
Thus the expected number of events per unit time is λ. Let T_n be the nth jump time, set $T_0 = 0$, and write $\tau_n = T_n - T_{n-1}$ for the interarrival times. Then
$$\mathbb{P}(\tau_1 > t) = \mathbb{P}(N_t = 0) = e^{-\lambda t},$$
and the independent, stationary increments imply that the τ_n are iid $\text{Exp}(\lambda)$ random variables. We thus obtain the following algorithm for sampling an $\text{HPP}(\lambda)$ on $[0, T]$ by interarrival times:

Algorithm 16: Sampling an HPP(λ) by interarrival times

```
Set t = 0 and initialize an empty list of jump times
while t < T do
    Generate ξ ~ Exp(λ) and set t ← t + ξ
    if t ≤ T then
        append t to the list of jump times
return the recorded jump times
```

Remark 16.2 Given jump times $T_1, \dots, T_m$, the path can be reconstructed from

$$N_t = \sum_{i=1}^{m} \mathbb{1}_{[T_i,\infty)}(t), \quad 0 \le t \le T.$$

There is also a construction on a fixed horizon based on conditional order statistics.

Algorithm 17: Sampling an HPP(λ) by counts

Generate $N_T \sim \text{Pois}(\lambda T)$
Draw $U_1, \dots, U_{N_T} \overset{\text{iid}}{\sim} \text{Unif}[0,1]$ and set $T_j = U_{(j)}T$ for $j = 1, \dots, N_T$
return $T_1, \dots, T_{N_T}$ *as the jump times*

Definition 16.3 A **nonstationary Poisson process** (**NPP**) $\{N_t\}_{t\ge 0}$ has $N_0 = 0$, independent increments, and a nonnegative, locally integrable intensity function λ. Its **integrated rate** over $[t, t+h]$ is

$$\Lambda_{t,t+h} = \int_t^{t+h} \lambda(s)\,\mathrm{d}s.$$

Then

$$\mathbb{P}(N_{t+h} - N_t = k) = \frac{\Lambda_{t,t+h}^k}{k!} e^{-\Lambda_{t,t+h}}.$$

Thus $\Lambda_{0,h}$ is the expected number of events in $[0, h]$.

Lewis and Shedler (1979) proposed the rejection method called **thinning**. Suppose $0 < \lambda^* < \infty$ and $\lambda(s) \le \lambda^*$ for all $s \in [0, T]$.

Algorithm 18: Thinning for an NPP

Generate candidate jump times $T'_1, \dots, T'_{K'}$ from an HPP with rate λ^* on $[0, T]$
Initialize an empty list of accepted times
for $k = 1, \dots, K'$ **do**
    Sample $U_k \sim \text{Unif}[0,1]$ independently
    if $U_k \le \lambda(T'_k)/\lambda^*$ **then**
        append T'_k to the list of accepted times
return *the accepted candidate times*

The envelope can be tightened by partitioning $[0, T]$ into intervals $(t_{i-1}, t_i]$ and using a piecewise-

constant upper bound

$$\lambda^*(s) = \sum_i \lambda_i^* \mathbb{1}_{(t_{i-1}, t_i]}(s)$$

with $\lambda(s) \leq \lambda_i^*$ on each interval.

Çinlar (1975) presented an inversion method for an NPP. Let

$$\Lambda(t) := \Lambda_{0,t} = \int_0^t \lambda(s)\, \mathrm{d}s.$$

Assume $\Lambda(t) \to \infty$ as $t \to \infty$, and write Λ^- for its generalized inverse. Given T_{k-1}, with $T_0 = 0$, sample the next jump time as follows.

Algorithm 19: Inversion for an NPP

1 Generate $U \sim \text{Unif}(0, 1)$
2 **return** $T_k = \Lambda^-(\Lambda(T_{k-1}) - \log U)$

To verify the algorithm, for $t \geq T_{k-1}$ observe that

$$\begin{aligned} \mathbb{P}(T_k \leq t \mid T_{k-1}) &= 1 - \mathbb{P}(\text{no event occurs in } (T_{k-1}, t] \mid T_{k-1}) \\ &= 1 - \exp\big(-(\Lambda(t) - \Lambda(T_{k-1}))\big). \end{aligned}$$

Setting this conditional distribution function equal to u gives

$$T_k = \Lambda^-\big(\Lambda(T_{k-1}) - \log(1 - u)\big).$$

Since $1 - U \sim \text{Unif}[0, 1]$, the stated algorithm follows.

Lecture 17 — Thursday, March 05, 2015

3. Monte Carlo Simulation

Crude Monte Carlo Estimation

Our goal is to estimate

$$\mu = \int_A g(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

where $A \subseteq \mathbb{R}^d$ is measurable with $0 < |A| < \infty$ and $g : A \to \mathbb{R}$ is integrable. Usually d is large enough that μ is not analytically available. Observe that if $\boldsymbol{X} \sim \mathrm{Unif}(A)$, then

$$\mu = |A| \int_A g(\boldsymbol{x}) \frac{1}{|A|} \mathbb{1}_A(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = |A|\mathbb{E}[g(\boldsymbol{X})],$$

where $|A|$ is the volume of A and $(1/|A|)\mathbb{1}_A$ is the density of $\boldsymbol{X}$. By the strong law of large numbers in (2), for independent $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \sim \mathrm{Unif}(A)$,

$$\hat{\mu}_n := |A| \frac{1}{n} \sum_{i=1}^{n} g(\boldsymbol{X}_i) \xrightarrow{a.s.} \mu,$$

so $\hat{\mu}_n \approx \mu$ when n is large.

Definition 17.1 For independent $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \sim \mathrm{Unif}(A)$, the **crude Monte Carlo estimator** of μ is

$$\hat{\mu}_n = |A| \frac{1}{n} \sum_{i=1}^{n} g(\boldsymbol{X}_i).$$

When $d = 1$, the integral is the area under g over A, and crude Monte Carlo estimates it by averaging the randomly sampled heights in Figure 11:

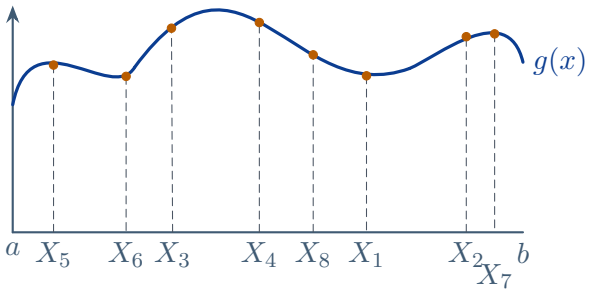

FIGURE 11

The estimator is unbiased:

$$\mathbb{E}[\hat{\mu}_n] = |A|\mathbb{E}[g(\boldsymbol{X})] = \int_A g(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \mu.$$

If $\sigma^2 = \mathrm{Var}(g(\boldsymbol{X})) < \infty$, then

$$\mathrm{Var}(\hat{\mu}_n) = \frac{|A|^2\sigma^2}{n}.$$

Chebyshev's inequality gives consistency, since for every $\varepsilon > 0$,

$$\mathbb{P}(|\hat{\mu}_n - \mu| > \varepsilon) \le \frac{|A|^2\sigma^2}{n\varepsilon^2} \to 0.$$

If $\sigma > 0$, the central limit theorem gives

$$\frac{\sqrt{n}(\hat{\mu}_n - \mu)}{|A|\sigma} \xrightarrow{d} \mathcal{N}(0,1).$$

With

$$S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(g(\boldsymbol{X}_i) - \frac{1}{n}\sum_{j=1}^{n} g(\boldsymbol{X}_j)\right)^2,$$

an asymptotic $(1-\alpha)$ confidence interval for μ is therefore

$$\left[\hat{\mu}_n - \frac{q_{1-\alpha/2}|A|S_n}{\sqrt{n}},\ \hat{\mu}_n + \frac{q_{1-\alpha/2}|A|S_n}{\sqrt{n}}\right], \qquad \text{where} \qquad q_{1-\alpha/2} = \Phi^{-1}(1-\alpha/2).$$

By Slutsky's theorem, the confidence interval is asymptotically valid even if S_n is used in place of σ. The width of the confidence interval is approximately $2q_{1-\alpha/2}|A|S_n/\sqrt{n}$, so to target half-width at most ε, we need

$$n \gtrsim \left(\frac{q_{1-\alpha/2}|A|S_n}{\varepsilon}\right)^2,$$

with S_n estimated from a pilot run.

If n is small and $|A|g(\boldsymbol{X})$ is (nearly) normal, we should replace $q_{1-\alpha/2} = \Phi^{-1}(1-\alpha/2)$ by the corresponding Student t_{n-1} quantile, $t_{n-1}^{-1}(1-\alpha/2)$.

Another elementary estimator is the hit-or-miss estimator, which estimates the area under g by counting the fraction of points in a box that fall under the graph of g.

Definition 17.2 Suppose $0 \le g(\boldsymbol{x}) \le M$ on A. Generate $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n \overset{\text{i.i.d.}}{\sim} \text{Unif}(A)$ and $Y_1, \dots, Y_n \overset{\text{i.i.d.}}{\sim} \text{Unif}[0, M]$ independently, and define the **hit-or-miss estimator** of μ by

$$\hat{\mu}_n^{\text{hm}} = |A| M \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{Y_i \le g(\boldsymbol{X}_i)\}}.$$

Figure 12 illustrates the sampled points and the bounding box used by this estimator.

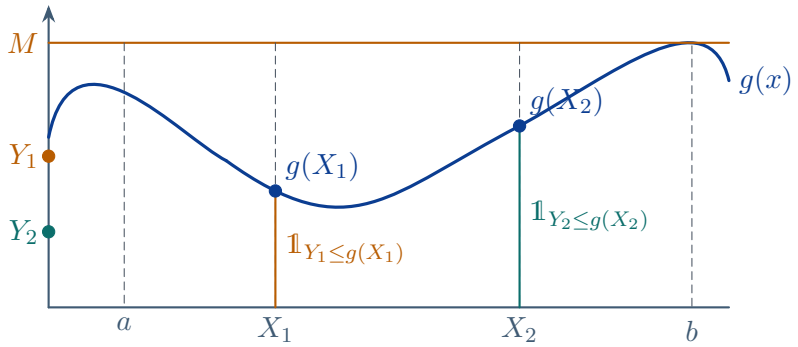

FIGURE 12

Since

$$\begin{aligned}
\mathbb{E}[\hat{\mu}_n^{\text{hm}}] &= |A| M \, \mathbb{P}(Y \le g(\boldsymbol{X})) \\
&= |A| M \int_A \frac{g(\boldsymbol{x})}{M} \frac{1}{|A|} \, \mathrm{d}\boldsymbol{x} \\
&= \int_A g(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \mu,
\end{aligned}$$

this estimator is unbiased. Its variance is

$$\text{Var}(\hat{\mu}_n^{\text{hm}}) = \frac{|A|^2 M^2}{n} p(1-p) = \frac{|A| M \mu - \mu^2}{n},$$

where $p = \mu / (|A| M)$ is the probability that a random point in the box $A \times [0, M]$ falls under the graph of g. The scaled Bernoulli factor $p(1-p)$ is maximized at $p = 1/2$. The hit-or-miss estimator may be easier to compute, but

$$\begin{aligned}
\text{Var}(\hat{\mu}_n) &= \frac{|A|^2}{n} \left(\mathbb{E}[g(\boldsymbol{X})^2] - \mathbb{E}[g(\boldsymbol{X})]^2 \right) \\
&\le \frac{|A|^2}{n} \left(M \mathbb{E}[g(\boldsymbol{X})] - \mathbb{E}[g(\boldsymbol{X})]^2 \right)
\end{aligned}$$

$$= \operatorname{Var}(\hat{\mu}_n^{\mathrm{hm}}).$$

Example 17.3 Using

$$\int_0^1 \sqrt{1-x^2}\,\mathrm{d}x = \frac{\pi}{4},$$

we can estimate π with a hit-or-miss estimator. Take $g(x) = \sqrt{1-x^2}$ and $A = [0,1]$. Then $\mu = \pi/4$ and $M = 1$, so we can estimate π by sampling $X_i, Y_i \sim \mathrm{Unif}[0,1]$ independently and using the hit-or-miss estimator

$$\hat{\pi}_n = \frac{1}{n}\sum_{i=1}^n 4\mathbb{1}_{\{Y_i \le \sqrt{1-X_i^2}\}} = 4\frac{1}{n}\sum_{i=1}^n \mathbb{1}_{\{X_i^2+Y_i^2\le 1\}}.$$

The indicator counts the fraction of sampled points in the unit square that fall inside the quarter unit disk. Under the normal approximation, a target half-width ε at confidence level $1-\alpha$ requires

$$n \gtrsim \left(\frac{q_{1-\alpha/2}S_n}{\varepsilon}\right)^2,$$

where S_n estimates the standard deviation of the summand $4\mathbb{1}_{\{X_i^2+Y_i^2\le 1\}}$ from a pilot run.

We typically change A to be $[0,1]^d$ by a change of variables in order to sample from the $\mathrm{Unif}(A)$ distribution. To reduce $\operatorname{Var}(\hat{\mu}_n)$, we can then replace the pseudorandom number generator by a quasirandom number generator, which produces **low-discrepancy sequences** that are more evenly spread out than independent uniform points. This is the idea behind **quasi-Monte Carlo methods**.

For $d = 1$, we often prefer numerical integration, whose convergence can be of order $O(1/n^k)$ for some $k > 0$ rather than the usual Monte Carlo order $O(1/\sqrt{n})$. In high dimensions, deterministic grids can become infeasible because the number of grid points grows exponentially in d. The Monte Carlo rate $n^{-1/2}$ does not deteriorate with dimension, although its variance constant can still depend strongly on d.

Figure 13 isolates the geometry of a single hit-or-miss draw.

Example 17.4

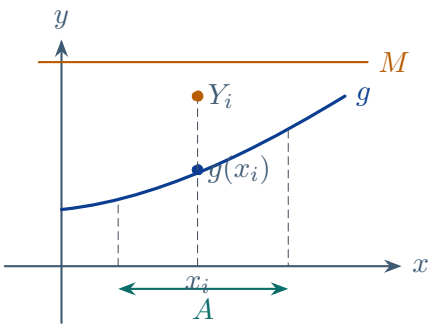

FIGURE 13

1. For an integral over a finite interval,

$$\mu = \int_a^b g(x)\,\mathrm{d}x = (b-a)\int_0^1 g(a+(b-a)u)\,\mathrm{d}u.$$

   Hence, for $U_i \sim \text{Unif}[0,1]$,

$$\hat{\mu}_n = \frac{b-a}{n}\sum_{i=1}^{n} g(a+(b-a)u_i)$$

   estimates μ.

2. For an integral over $[0,\infty)$, use $u = 1/(1+x)$, so $x = (1-u)/u$ and $\mathrm{d}x = -u^{-2}\,\mathrm{d}u$. Then

$$\mu = \int_0^\infty g(x)\,\mathrm{d}x = \int_0^1 g\left(\frac{1-u}{u}\right)\frac{1}{u^2}\,\mathrm{d}u.$$

   Thus

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} g\left(\frac{1-u_i}{u_i}\right)\frac{1}{u_i^2}$$

   for $u_i \sim \text{Unif}[0,1]$.

Lecture 18 — Tuesday, March 10, 2015

3. For an integral over the real line,

$$\mu = \int_{-\infty}^{\infty} g(x)\,\mathrm{d}x = \int_{-\infty}^{0} g(x)\,\mathrm{d}x + \int_{0}^{\infty} g(x)\,\mathrm{d}x.$$

Using $x = (1-u)/u$ on $(0,\infty)$ and $x = (u-1)/u$ on $(-\infty, 0)$ gives

$$\begin{aligned}\mu &= \int_0^1 g\left(\frac{1-u}{u}\right)\frac{1}{u^2}\,\mathrm{d}u + \int_0^1 g\left(\frac{u-1}{u}\right)\frac{1}{u^2}\,\mathrm{d}u \\ &= \int_0^1 \left[g\left(\frac{1-u}{u}\right) + g\left(\frac{u-1}{u}\right)\right]\frac{1}{u^2}\,\mathrm{d}u.\end{aligned}$$

Thus, for $U_i \sim \mathrm{Unif}[0,1]$,

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n}\left[g\left(\frac{1-U_i}{U_i}\right) + g\left(\frac{U_i-1}{U_i}\right)\right]\frac{1}{U_i^2}$$

is a Monte Carlo estimator. If g is even, this reduces to

$$\mu = 2\int_0^{\infty} g(x)\,\mathrm{d}x.$$

Other one-to-one maps from $\mathbb{R}$ to $[0,1]$ include

$$h(x) = \frac{1}{2} + \frac{1}{\pi}\arctan x \qquad \text{and} \qquad h(x) = \frac{e^x}{1+e^x}.$$

4. For integrals over a quadrant, the same transformation is applied coordinatewise. For example,

$$\begin{aligned}\mu &= \int_0^{\infty}\int_0^{x} g(x,y)\,\mathrm{d}y\,\mathrm{d}x \\ &= \int_0^{\infty}\int_0^{\infty} g(x,y)\mathbb{1}_{\{y\le x\}}\,\mathrm{d}y\,\mathrm{d}x \\ &= \int_0^1\int_0^1 g\left(\frac{1-u_1}{u_1}, \frac{1-u_2}{u_2}\right)\frac{1}{u_1^2u_2^2}\mathbb{1}_{\{\frac{1-u_2}{u_2}\le\frac{1-u_1}{u_1}\}}\,\mathrm{d}u_1\,\mathrm{d}u_2.\end{aligned}$$

The corresponding estimator is

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} g\left(\frac{1-U_{i1}}{U_{i1}}, \frac{1-U_{i2}}{U_{i2}}\right) \frac{1}{U_{i1}^2 U_{i2}^2} \mathbb{1}_{\{\frac{1-U_{i2}}{U_{i2}} \le \frac{1-U_{i1}}{U_{i1}}\}}$$

for $U_{i1}, U_{i2} \overset{\text{iid}}{\sim} \text{Unif}[0,1]$.

Depending on how we derive Monte Carlo estimators, we can have quite different variances.

Example 18.1 Let X have the standard Cauchy density

$$f(x) = \frac{1}{\pi(1+x^2)}$$

and let

$$\mu = \mathbb{P}(X > 2) = \mathbb{E}[\mathbb{1}_{\{X>2\}}].$$

The crude estimator

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{X_i>2\}}$$

has variance

$$\text{Var}(\hat{\mu}_n) = \frac{\mu(1-\mu)}{n} \approx \frac{0.1258}{n},$$

since $\mu = 1/2 - \arctan(2)/\pi \approx 0.1476$. Using symmetry, $\mu = \mathbb{P}(|X| > 2)/2$, so if

$$\hat{\mu}'_n = \frac{1}{2n}\sum_{i=1}^{n} \mathbb{1}_{\{|X_i|>2\}},$$

then $\hat{\mu}'_n$ is unbiased for μ, and

$$\text{Var}(\hat{\mu}'_n) = \frac{1}{4n}(2\mu)(1-2\mu) \approx \frac{0.0520}{n}.$$

Define the **efficiency** (or **variance reduction factor**) as the ratio of the variances of the crude estimator to the improved estimator, which here is

$$\frac{\text{Var}(\hat{\mu}_n)}{\text{Var}(\hat{\mu}'_n)} = \frac{0.1258/n}{0.0520/n} \approx 2.42.$$

The complement can do even better. Since

$$1 - 2\mu = 1 - \mathbb{P}(|X| > 2) = \mathbb{P}(|X| \leq 2) = \int_{-2}^{2} f(x)\,\mathrm{d}x = 2\int_{0}^{2} f(x)\,\mathrm{d}x,$$

we have

$$\mu = \frac{1}{2} - \int_{0}^{2} f(x)\,\mathrm{d}x.$$

If $Y_i \sim \text{Unif}[0, 2]$, then

$$\hat{\mu}_n'' = \frac{1}{2} - \frac{2}{n}\sum_{i=1}^{n} f(Y_i)$$

is unbiased for μ, and

$$\text{Var}(\hat{\mu}_n'') = \frac{4\,\text{Var}(f(Y))}{n} \approx \frac{0.0285}{n}.$$

The variance reduction factor relative to the crude one-sided estimator is about 4.41.

Finally, transform the tail integral directly. With the substitution $y = 1/x$, the range $x \in [2, \infty)$ corresponds to $y \in (0, 1/2]$, and

$$\mu = \int_{2}^{\infty} f(x)\,\mathrm{d}x = \int_{0}^{1/2} f\left(\frac{1}{y}\right)\frac{1}{y^2}\,\mathrm{d}y = \int_{0}^{1/2} \frac{1}{\pi(1+y^2)}\,\mathrm{d}y = \int_{0}^{1/2} f(y)\,\mathrm{d}y.$$

Thus, for $Y_i \sim \text{Unif}[0, 1/2]$,

$$\hat{\mu}_n = \frac{1}{2n}\sum_{i=1}^{n} \frac{1}{\pi(1+Y_i^2)}$$

for $Y_i \overset{\text{iid}}{\sim} \text{Unif}[0, 1/2]$ estimates the same tail probability, and

$$\text{Var}(\hat{\mu}_n) \approx \frac{9.55 \cdot 10^{-5}}{n},$$

which leads to an efficiency of about 1317 relative to the crude estimator.

Next, we will discuss techniques for reducing the variance of Monte Carlo estimators. Our goal is to reduce the variance of unbiased and consistent estimators at little computational cost.

4. Variance Reduction Methods

Variance Reduction Overview

The goal of variance reduction is to reduce the variance of a Monte Carlo estimator while preserving unbiasedness or at least consistency. The first techniques exploit dependence. We therefore begin with a short correlation and copula interlude.

If $\mathbb{E}[X_i^2] < \infty$ and $\mathrm{Var}(X_i) > 0$ for $i = 1, 2$, then

$$\mathrm{Cov}(X_1, X_2) = \mathbb{E}[X_1 X_2] - \mathbb{E}[X_1]\mathbb{E}[X_2],$$

and

$$\rho = \mathrm{Corr}(X_1, X_2) = \frac{\mathrm{Cov}(X_1, X_2)}{\sqrt{\mathrm{Var}(X_1)\,\mathrm{Var}(X_2)}}.$$

We have $\rho \in [-1, 1]$, with $|\rho| = 1$ precisely when X_2 is almost surely a linear function of X_1. We now establish an important connection between copulas and $\mathrm{Cov}(X_1, X_2)$ (or ρ). To this end, we first show the following lemma:

Lemma 18.2 (Fréchet–Hoeffding bounds) Let

$$W(u_1, u_2) = \max\{u_1 + u_2 - 1, 0\} \qquad \text{and} \qquad M(u_1, u_2) = \min\{u_1, u_2\}.$$

For every copula C on $[0, 1]^2$,

1. $W(u_1, u_2) \le C(u_1, u_2) \le M(u_1, u_2)$ for all $u_1, u_2 \in [0, 1]$.
2. W and M are copulas.

Proof. Let (U_1, U_2) have copula C. Then

$$\begin{aligned} 1 - C(u_1, u_2) &= \mathbb{P}(U_1 > u_1 \text{ or } U_2 > u_2) \\ &= \mathbb{P}(U_1 > u_1) + \mathbb{P}(U_2 > u_2) - \mathbb{P}(U_1 > u_1 \text{ and } U_2 > u_2) \\ &\le 1 - u_1 + 1 - u_2 \\ &= 2 - u_1 - u_2, \end{aligned}$$

so $C(u_1, u_2) \ge u_1 + u_2 - 1$, and the lower bound 0 is automatic. Hence $C \ge W$. Also,

$$C(u_1, u_2) \le \mathbb{P}(U_1 \le u_1) = u_1 \qquad \text{and} \qquad C(u_1, u_2) \le \mathbb{P}(U_2 \le u_2) = u_2,$$

so $C(u_1, u_2) \le \min\{u_1, u_2\} = M(u_1, u_2)$.

If $U \sim \text{Unif}[0,1]$, then we have seen that $(U, 1-U)$ has copula W and (U, U) has copula M. Hence both bounds are themselves copulas. □

Proposition 18.3 (Hoeffding's identity) Let (X_1, X_2) have joint distribution function H and margins F_1, F_2. If $\mathbb{E}[X_i^2] < \infty$ for $i = 1, 2$, then

$$\text{Cov}(X_1, X_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \big(H(x_1, x_2) - F_1(x_1)F_2(x_2)\big)\, \mathrm{d}x_1\, \mathrm{d}x_2.$$

Proof. Let $Y_1 \sim F_1$ and $Y_2 \sim F_2$ be independent of each other and of (X_1, X_2). Then

$$\text{Cov}(X_1, X_2) = \mathbb{E}[(X_1 - Y_1)(X_2 - Y_2)].$$

For real a and b,

$$a - b = \int_{-\infty}^{\infty} \big(\mathbb{1}_{\{a>x\}} - \mathbb{1}_{\{b>x\}}\big)\, \mathrm{d}x.$$

Applying this identity twice and then Fubini's theorem, justified by the finite second moments, gives

$$\begin{aligned}
\text{Cov}(X_1, X_2) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbb{E}\left[\big(\mathbb{1}_{\{X_1>x_1\}} - \mathbb{1}_{\{Y_1>x_1\}}\big)\big(\mathbb{1}_{\{X_2>x_2\}} - \mathbb{1}_{\{Y_2>x_2\}}\big)\right] \mathrm{d}x_1\, \mathrm{d}x_2 \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \big(\mathbb{P}(X_1 > x_1, X_2 > x_2) - \mathbb{P}(X_1 > x_1)\mathbb{P}(X_2 > x_2)\big)\, \mathrm{d}x_1\, \mathrm{d}x_2 \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \big(H(x_1, x_2) - F_1(x_1)F_2(x_2)\big)\, \mathrm{d}x_1\, \mathrm{d}x_2.
\end{aligned}$$

□

Corollary 18.4 By Sklar's theorem, $H(x_1, x_2) = C(F_1(x_1), F_2(x_2))$. We thus obtain the following formula for the covariance and correlation in terms of the copula:

$$\rho = \frac{\displaystyle\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \big(C(F_1(x_1), F_2(x_2)) - F_1(x_1)F_2(x_2)\big)\, \mathrm{d}x_1\, \mathrm{d}x_2}{\sqrt{\text{Var}(X_1)\,\text{Var}(X_2)}}.$$

For fixed margins, this expression is increasing in the copula: if $C_1(u_1, u_2) \le C_2(u_1, u_2)$ for all $u_1, u_2 \in [0,1]$, then $\rho_{C_1} \le \rho_{C_2}$.

Remark 18.5

1. The Fréchet–Hoeffding bounds imply $\rho_C \in [\rho_W, \rho_M]$ for fixed margins. Corollary 18.4 implies that if $C(u_1, u_2) \geq u_1 u_2$, where $u_1 u_2$ is the independence copula, then $\mathrm{Cov}(X_1, X_2) \geq 0$ (and hence $\rho_C \geq 0$ as well).
2. If $X_1 = g_1(Z)$ and $X_2 = g_2(Z)$ with the same direction of monotonicity, then the variables are comonotone and their covariance is nonnegative. If one function is increasing and the other decreasing, their covariance is nonpositive.

Lecture 19 — Thursday, March 12, 2015

3. By induction on the dimension d, the preceding monotonicity argument extends as follows. If $Z_1, \ldots, Z_d$ are independent, $g_1(Z_1, \ldots, Z_d)$ and $g_2(Z_1, \ldots, Z_d)$ are square-integrable, and g_1, g_2 are componentwise increasing, then

$$\mathrm{Cov}(g_1(Z_1, \ldots, Z_d), g_2(Z_1, \ldots, Z_d)) \geq 0.$$

In particular, if g is componentwise increasing and $\boldsymbol{U} \sim \mathrm{Unif}[0,1]^d$, then

$$\mathrm{Cov}\big(g(\boldsymbol{U}), g(\mathbf{1} - \boldsymbol{U})\big) \leq 0.$$

Common Random Numbers

Here, we are interested in estimating $\mu = \mu_1 - \mu_2$, where $\mu_1 = \mathbb{E}[g_1(\boldsymbol{X})]$, $\mu_2 = \mathbb{E}[g_2(\boldsymbol{X}')]$, and both $\boldsymbol{X}$ and $\boldsymbol{X}'$ have distribution H. Assume $g_1, g_2 : \mathbb{R}^d \to \mathbb{R}$ are square-integrable under H. This problem arises when we compare the performance of two systems. We want the estimator $\hat{\mu}_n = \hat{\mu}_{n,1} - \hat{\mu}_{n,2}$ to have small variance, so that an observed difference is less likely to be simulation noise.

Let the pairs $(\boldsymbol{X}_i, \boldsymbol{X}_i')$ be iid across i, with each component having distribution H. To estimate μ_1 and μ_2 via Monte Carlo, we can then use the unbiased estimator

$$\mathbb{E}[\hat{\mu}_n] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\big(g_1(\boldsymbol{X}_i) - g_2(\boldsymbol{X}_i')\big)\right] = \mu.$$

What about the variance? We have

$$\mathrm{Var}(\hat{\mu}_n) = \frac{1}{n}\left(\mathrm{Var}(g_1(\boldsymbol{X})) + \mathrm{Var}(g_2(\boldsymbol{X}')) - 2\,\mathrm{Cov}(g_1(\boldsymbol{X}), g_2(\boldsymbol{X}'))\right).$$

If $g_1(\boldsymbol{X})$ and $g_2(\boldsymbol{X}')$ are positively correlated, then the variance of $\hat{\mu}_n$ is reduced relative to the independence case. If g_1 and g_2 are componentwise increasing and the input vector has independent components, using the same input vector for both systems gives nonnegative covariance and therefore never increases variance relative to independent inputs, although we need more information to know the size of the reduction.

In general, introducing correlation or dependence *can* reduce variance, but it often remains unclear how much variance is reduced. Furthermore, it can even be dangerous when dependent data is used for further statistical analysis requiring an iid setup.

Using the same random numbers to investigate different problems often reduces variance, but only if the random numbers are used for the same purpose (i.e., if they are **synchronized**). Common ways to preserve synchronization include assigning dedicated streams to specific modelling purposes, using inversion-based sampling so that the same uniforms have comparable meanings, discarding random numbers deliberately when one system needs fewer draws, and designing the simulation study so that corresponding events consume corresponding streams.

Antithetic Variates

Antithetic variates deliberately introduce negative dependence.

Definition 19.1 Suppose

$$\mu = \mathbb{E}\left[g(X)\right] = \mathbb{E}\left[\frac{g(X) + g(X')}{2}\right]$$

where both X and X' have distribution F. Let H be a joint distribution with margins F, and let (X_i, X_i'), $i = 1, \ldots, N$, be independent pairs from H. With total simulation budget $n = 2N$, define

$$\hat{\mu}_n^{\mathrm{av}} = \frac{1}{N}\sum_{i=1}^{N} \frac{g(X_i) + g(X_i')}{2}.$$

This estimator is unbiased for μ, and its variance is

$$\mathrm{Var}(\hat{\mu}_n^{\mathrm{av}}) = \frac{1}{N}\,\mathrm{Var}\left(\frac{g(X) + g(X')}{2}\right)$$

$$= \frac{1}{4N}\left[\mathrm{Var}(g(X)) + \mathrm{Var}(g(X')) + 2\,\mathrm{Cov}(g(X), g(X'))\right].$$

Thus the method is useful when the coupling H makes $\mathrm{Cov}(g(X), g(X')) < 0$, since the variance is then less than $\mathrm{Var}(g(X))/n$ for the crude Monte Carlo estimator $\hat{\mu}_n^{\mathrm{cmc}} = (1/n)\sum_{i=1}^n g(X_i)$, where $X_1, \ldots, X_n$ are independent with distribution F.

Remark 19.2

1. If g is increasing, $X = h(U_1, \ldots, U_d)$ for some componentwise increasing function h and independent $U_1, \ldots, U_d$, and if we take $X' = h(1 - U_1, \ldots, 1 - U_d)$, then $\mathrm{Cov}(g(X), g(X')) \leq 0$ by (3). Thus the antithetic estimator has variance no greater than that of crude Monte Carlo with the same number of function evaluations.
2. Instead of sampling n independent random variables (as in the crude Monte Carlo estimator), we sample $n/2$ pairs of dependent random variables. The basic idea is that negatively correlated random variables "offset" each other and thus their average is closer to μ. For this to work well, g should be strictly monotone.

Example 19.3 The effectiveness of antithetic variates depends strongly on the function g.

1. If $U \sim \mathrm{Unif}[0, 1]$ and
$$X = h(U) = 2U - 1 \sim \mathrm{Unif}[-1, 1],$$
and $g(x) = x^2$, then the antithetic value is
$$X' = 2(1 - U) - 1 = -X \sim \mathrm{Unif}[-1, 1].$$
and $g(X') = g(X)$ almost surely. We therefore gain no additional information about μ from the second evaluation in each pair. With the same total budget of n function evaluations, the antithetic estimator has twice the variance of crude Monte Carlo. (This was a particularly bad choice of g.)
2. In the same setup, if $g(x) = x$, then
$$\mu = \mathbb{E}[X] = 0,$$
and for every pair $(X, X') = (X, -X)$ with $X \sim \mathrm{Unif}[-1, 1]$, we have that
$$\frac{g(X) + g(X')}{2} = 0 = \mu$$

and thus $\mathbb{E}[\hat{\mu}_n^{\text{av}}] = \mu$ and more importantly, $\text{Var}(\hat{\mu}_n^{\text{av}}) = 0$. Simply the best. Thus, how well the method works depends on the monotonicity of g.

Remark 19.4

1. If X has a continuous distribution symmetric about 0, then $F(x) = 1 - F(-x)$, so $F^-(u) = -F^-(1-u)$ and the antithetic pair is $(X, X') = (F^-(U), F^-(1-U)) = (X, -X)$. In this case, the pair has the lower Fréchet–Hoeffding copula W, and $\text{Cov}(g(X), g(X')) \leq 0$ for every increasing function g. Thus, for continuous symmetric distributions, antithetic variates never increase variance for increasing functions.

2. As before, the variance reduction factor is unknown. As an example, we apply the method to estimate $\mu = \int_0^1 e^x \, \mathrm{d}x$ and compare the variance of the antithetic estimator to that of the crude Monte Carlo estimator. The crude estimator is

$$\hat{\mu}_n^{\text{cmc}} = \frac{1}{n} \sum_{i=1}^{n} e^{U_i},$$

and

$$\text{Var}(\hat{\mu}_n^{\text{cmc}}) = \frac{\text{Var}(e^U)}{n} = \frac{\frac{1}{2}(e^2-1) - (e-1)^2}{n} = \frac{(e-1)(3-e)}{2n} \approx \frac{0.2420}{n}.$$

The antithetic estimator, using $n/2$ pairs, is

$$\hat{\mu}_n^{\text{av}} = \frac{1}{n/2} \sum_{i=1}^{n/2} \frac{e^{U_i} + e^{1-U_i}}{2}.$$

Since

$$\text{Cov}(e^U, e^{1-U}) = e - (e-1)^2,$$

we obtain

$$\text{Var}(\hat{\mu}_n^{\text{av}}) = \frac{\text{Var}(e^U) + \text{Cov}(e^U, e^{1-U})}{n} = \frac{10e - 3e^2 - 5}{2n} \approx \frac{0.0078}{n}.$$

The variance reduction factor is therefore approximately $0.2420/0.0078 \approx 30.98$. We thus need a crude Monte Carlo sample roughly 31 times as large to achieve the same precision as the antithetic estimator.

Lecture 20 — Tuesday, March 17, 2015

Control Variates

Let $Y = g(\boldsymbol{X})$, let $\mu = \mathbb{E}[Y]$, and suppose we can compute another random variable $Z = h(\boldsymbol{X})$ whose mean $\mu_Z = \mathbb{E}[Z]$ is known. Then, for every $\beta \in \mathbb{R}$,

$$\mu = \mathbb{E}\left[Y - \beta\left(Z - \mu_Z\right)\right] = \int_{\mathbb{R}^d} \left(g(\boldsymbol{x}) - \beta h(\boldsymbol{x})\right) f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \beta \int_{\mathbb{R}^d} h(\boldsymbol{x}) f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.$$

The random variable Z is called a **control variate**. A coefficient with the same sign as $\mathrm{Cov}(Y, Z)$ adjusts Y in the useful direction, although its magnitude must also be chosen carefully:

- If $\mathrm{Cov}(Y, Z) > 0$ and $Z > \mu_Z$, then Y is likely above μ, so we subtract a positive amount to get closer to μ.
- If $\mathrm{Cov}(Y, Z) > 0$ and $Z < \mu_Z$, then Y is likely below μ, so we subtract a negative amount (i.e., add a positive amount) to get closer to μ.
- If $\mathrm{Cov}(Y, Z) < 0$ and $Z > \mu_Z$, then Y is likely below μ, so we subtract a negative amount (i.e., add a positive amount) to get closer to μ.
- If $\mathrm{Cov}(Y, Z) < 0$ and $Z < \mu_Z$, then Y is likely above μ, so we subtract a positive amount to get closer to μ.

The unbiased and consistent estimator

$$\tilde{\mu}_n(\beta) = \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \beta\left(Z_i - \mu_Z\right)\right)$$

has variance

$$\mathrm{Var}(\tilde{\mu}_n(\beta)) = \frac{1}{n}\left[\mathrm{Var}(Y) + \beta^2 \mathrm{Var}(Z) - 2\beta\,\mathrm{Cov}(Y, Z)\right] =: \frac{f(\beta)}{n}.$$

Assume $\mathrm{Var}(Z) > 0$. Then $\tilde{\mu}_n(\beta)$ improves on crude Monte Carlo exactly when

$$\beta^2 \mathrm{Var}(Z) < 2\beta\,\mathrm{Cov}(Y, Z).$$

For simple unscaled choices such as $\beta \in \{-1, 1\}$, whether this condition holds depends on the joint

distribution of (Y, Z).

Optimizing $f(\beta)$ with respect to β (the **method of optimal control variates**) leads to

$$f'(\beta) = 2\beta \operatorname{Var}(Z) - 2\operatorname{Cov}(Y, Z) = 0$$

which gives a critical point at

$$\beta^* = \frac{\operatorname{Cov}(Y, Z)}{\operatorname{Var}(Z)}.$$

Since $f''(\beta) = 2\operatorname{Var}(Z) > 0$, this critical point is a minimum, and the optimal control variate estimator is

$$\tilde{\mu}_n(\beta^*) = \frac{1}{n}\sum_{i=1}^{n} \left(Y_i - \beta^* \left(Z_i - \mu_Z\right)\right).$$

The optimal adjustment depends on both covariance and scale: β^* is the regression slope of Y on Z. Strong linear association makes the adjusted variable $Y - \beta^*(Z - \mu_Z)$ substantially less variable, while weak association yields little improvement.

The variance of the optimal control variate estimator is

$$\begin{aligned}
\operatorname{Var}(\tilde{\mu}_n(\beta^*)) &= \frac{f(\beta^*)}{n} \\
&= \frac{1}{n}\left[\operatorname{Var}(Y) + \left(\frac{\operatorname{Cov}(Y, Z)}{\operatorname{Var}(Z)}\right)^2 \operatorname{Var}(Z) - 2\frac{\operatorname{Cov}(Y, Z)}{\operatorname{Var}(Z)}\operatorname{Cov}(Y, Z)\right] \\
&= \frac{1}{n}\left[\operatorname{Var}(Y) - \frac{\operatorname{Cov}(Y, Z)^2}{\operatorname{Var}(Z)}\right] \\
&= \operatorname{Var}(\hat{\mu}_n^{\text{cmc}})\left(1 - \frac{\operatorname{Cov}(Y, Z)^2}{\operatorname{Var}(Y)\operatorname{Var}(Z)}\right) \\
&= \operatorname{Var}(\hat{\mu}_n^{\text{cmc}})\left(1 - \rho_{Y,Z}^2\right) \\
&\leq \operatorname{Var}(\hat{\mu}_n^{\text{cmc}}),
\end{aligned}$$

where $\rho_{Y,Z} = \operatorname{Corr}(Y, Z)$ is the correlation between Y and Z. In particular, $\operatorname{Var}(\tilde{\mu}_n(\beta^*)) \ll \operatorname{Var}(\hat{\mu}_n^{\text{cmc}})$ when $|\rho_{Y,Z}|$ is close to 1, and $\operatorname{Var}(\tilde{\mu}_n(\beta^*)) \approx \operatorname{Var}(\hat{\mu}_n^{\text{cmc}})$ when $|\rho_{Y,Z}|$ is close to 0. When $\operatorname{Var}(Y) > 0$ and $|\rho_{Y,Z}| < 1$, the variance reduction factor is

$$\frac{\operatorname{Var}(\hat{\mu}_n^{\text{cmc}})}{\operatorname{Var}(\tilde{\mu}_n(\beta^*))} = \frac{1}{1 - \rho_{Y,Z}^2},$$

which is increasing in $|\rho_{Y,Z}|$ and tends to infinity as $|\rho_{Y,Z}| \to 1$. If $|\rho_{Y,Z}| = 1$, the optimal adjustment has zero variance.

Remark 20.1

1. In practice, we may not know $\mathrm{Var}(Z)$ and certainly not $\mathrm{Cov}(Y, Z)$, so β^* is usually unknown. We can estimate β^* from simulated data by

$$\hat{\beta}_n^* = \frac{\frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y})(Z_i - \bar{Z})}{\frac{1}{n-1}\sum_{i=1}^{n}(Z_i - \bar{Z})^2},$$

   which gives

$$\tilde{\mu}_n = \frac{1}{n}\sum_{i=1}^{n}\Big(Y_i - \hat{\beta}_n^*(Z_i - \mu_Z)\Big) = \bar{Y} - \hat{\beta}_n^*(\bar{Z} - \mu_Z).$$

2. Because $\hat{\beta}_n^*$ is estimated from the same data, the resulting estimator is not necessarily exactly unbiased.

3. The bias can be reduced by splitting the simulated data, using a jackknife correction, or using a bootstrap correction. The bias-corrected **jackknife estimator** of μ is

$$\hat{\mu}_n^{\text{jack}} = n\hat{\mu}_n - (n-1)\bar{\hat{\mu}}_{(-\cdot)},$$

   where

$$\hat{\mu}_{(-i)} = \frac{1}{n-1}\sum_{j\neq i}\Big(Y_j - \hat{\beta}_{(-i)}^*(Z_j - \mu_Z)\Big)$$

   is the control variate estimator computed by leaving out the ith data point, and

$$\bar{\hat{\mu}}_{(-\cdot)} = \frac{1}{n}\sum_{i=1}^{n}\hat{\mu}_{(-i)}.$$

   For a bootstrap correction, let

$$\hat{\mu}_n^{*(b)} = \frac{1}{n}\sum_{i=1}^{n}\Big(Y_i^{(b)} - \hat{\beta}_{(b)}^*(Z_i^{(b)} - \mu_Z)\Big) \qquad \text{and} \qquad \bar{\hat{\mu}}_B^* = \frac{1}{B}\sum_{b=1}^{B}\hat{\mu}_n^{*(b)}.$$

   The bias-corrected bootstrap estimator is

$$\hat{\mu}_{\text{boot}} = 2\hat{\mu}_n - \bar{\hat{\mu}}_B^*.$$

4. The problem of finding β resembles a linear regression problem: $Y = \alpha + \beta Z + \varepsilon$ for some error term ε with $\mathbb{E}[\varepsilon] = 0$. If we fit the regression by least squares, then the least squares slope is $\hat{\beta}_n^*$, and the control variate estimate is the fitted regression line evaluated at $Z = \mu_Z$:
$$\tilde{\mu}_n = \hat{\alpha} + \hat{\beta}_n^* \mu_Z.$$
The fitted residual variance estimates the variance of the adjusted observation $Y - \beta^*(Z - \mu_Z)$, so dividing it by n estimates the variance of the control-variate sample mean. This idea generalizes to multiple control variates $Z_1, \ldots, Z_k$ by fitting the regression
$$Y = \alpha + \beta_1 Z_1 + \cdots + \beta_k Z_k + \varepsilon$$
and using the fitted regression line evaluated at $Z_1 = \mu_{Z_1}, \ldots, Z_k = \mu_{Z_k}$ as the control variate estimator.

Example 20.2 Let us estimate
$$\int_0^1 e^x \, \mathrm{d}x = \mu,$$
using the method of control variates with the helper function $h(x) = x$, and compare it with the crude Monte Carlo estimator.

As in Remark 19.4(2), the crude Monte Carlo estimator has variance
$$\mathrm{Var}(\hat{\mu}_n^{\mathrm{cmc}}) = \frac{(e-1)(3-e)}{2n} \approx \frac{0.2420}{n}.$$
For the method of control variates with $\beta = 1$, we have
$$\mu = \int_0^1 (e^x - h(x)) \, \mathrm{d}x + \int_0^1 h(x) \, \mathrm{d}x = \int_0^1 (e^x - x) \, \mathrm{d}x + \frac{1}{2}.$$

Lecture 21 — Thursday, March 19, 2015

Therefore
$$\hat{\mu}_n = \frac{1}{2} + \frac{1}{n} \sum_{i=1}^{n} (e^{U_i} - U_i)$$

for $U_1, \ldots, U_n \overset{\text{i.i.d.}}{\sim} \text{Unif}[0,1]$, and

$$\begin{aligned}\text{Var}(\hat{\mu}_n) &= \frac{\text{Var}(e^U - U)}{n} \\ &= \frac{\text{Var}(e^U) + \text{Var}(U) - 2\,\text{Cov}(e^U, U)}{n} \\ &= \frac{\frac{1}{2}(e-1)(3-e) + \frac{1}{12} - (3-e)}{n} \approx \frac{0.0437}{n}.\end{aligned}$$

The variance reduction factor is approximately $0.2420/0.0437 = 5.54$.

We can further reduce the variance with optimal control variates. The optimal β is

$$\beta^* = \frac{\text{Cov}(e^U, U)}{\text{Var}(U)} = 6(3-e),$$

and the optimal estimator is

$$\hat{\mu}_n^{\text{ocv}} = \frac{1}{n}\sum_{i=1}^{n}\left(e^{U_i} - \beta^*\left(U_i - \frac{1}{2}\right)\right).$$

The optimal variance reduction factor is

$$\frac{1}{1-\rho^2_{e^U,U}} \approx 61.43,$$

which is a reduction of about 98.37% relative to crude Monte Carlo.

Importance Sampling

Let $\boldsymbol{X}$ have density f, and suppose

$$\mu = \mathbb{E}[g(\boldsymbol{X})].$$

Let h be another density, called the **importance density**, such that $h(\boldsymbol{x}) = 0$ only where $g(\boldsymbol{x})f(\boldsymbol{x}) = 0$. If $\boldsymbol{Y} \sim h$, then

$$\mu = \int_{\mathbb{R}^d} g(\boldsymbol{x})f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_{\mathbb{R}^d} \frac{g(\boldsymbol{x})f(\boldsymbol{x})}{h(\boldsymbol{x})}h(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \mathbb{E}\left[\frac{g(\boldsymbol{Y})f(\boldsymbol{Y})}{h(\boldsymbol{Y})}\right].$$

The idea behind importance sampling is to put more probability mass on ("important") values of $\boldsymbol{x}$ that contribute substantially to the integral, so that we can get a more accurate estimate of μ with a smaller sample size. This can be helpful if $\boldsymbol{X} \sim f$ is inefficient to sample or if $\text{Var}(g(\boldsymbol{X}))$

is large.

An unbiased estimator of μ is the **importance sampling estimator**

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} \frac{g(\boldsymbol{Y}_i)f(\boldsymbol{Y}_i)}{h(\boldsymbol{Y}_i)},$$

where $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n \overset{\text{i.i.d.}}{\sim} h$. If $\mathbb{E}_h[(g(\boldsymbol{Y})f(\boldsymbol{Y})/h(\boldsymbol{Y}))^2] < \infty$, its variance is

$$\operatorname{Var}(\hat{\mu}_n) = \frac{1}{n}\operatorname{Var}\left(\frac{g(\boldsymbol{Y})f(\boldsymbol{Y})}{h(\boldsymbol{Y})}\right),$$

which can be smaller than $\operatorname{Var}(\hat{\mu}_n^{\text{cmc}}) = \operatorname{Var}(g(\boldsymbol{X}))/n$ when h is chosen well. In particular, the variance is small when gf/h is nearly constant under h, so that the importance weights $g(\boldsymbol{Y})f(\boldsymbol{Y})/h(\boldsymbol{Y})$ are nearly constant under h.

Lemma 21.1 Assume $0 < \mathbb{E}[|g(\boldsymbol{X})|] < \infty$. Among all admissible densities h, the one that minimizes $\operatorname{Var}(\hat{\mu}_n)$ is

$$h^*(\boldsymbol{x}) = \frac{|g(\boldsymbol{x})|f(\boldsymbol{x})}{\mathbb{E}[|g(\boldsymbol{X})|]} \propto |g(\boldsymbol{x})|f(\boldsymbol{x}).$$

Proof. For any admissible h, the Cauchy–Schwarz inequality gives

$$\begin{aligned}\left(\int_{\mathbb{R}^d} |g(\boldsymbol{x})|f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\right)^2 &= \left(\int_{\mathbb{R}^d} \frac{|g(\boldsymbol{x})|f(\boldsymbol{x})}{\sqrt{h(\boldsymbol{x})}}\sqrt{h(\boldsymbol{x})}\,\mathrm{d}\boldsymbol{x}\right)^2 \\ &\le \int_{\mathbb{R}^d} \frac{g(\boldsymbol{x})^2 f(\boldsymbol{x})^2}{h(\boldsymbol{x})}\,\mathrm{d}\boldsymbol{x}.\end{aligned}$$

The right-hand side is the second moment of the importance-sampling summand. Equality holds precisely when h is proportional to $|g|f$, which gives h^* after normalization. Subtracting the common squared mean μ^2 proves the result. □

Remark 21.2

1. The **normalized importance sampling estimator** is

$$\hat{\mu}_n^{\text{nis}} = \frac{\sum_{i=1}^n g(\boldsymbol{Y}_i) f(\boldsymbol{Y}_i)/h(\boldsymbol{Y}_i)}{\sum_{i=1}^n f(\boldsymbol{Y}_i)/h(\boldsymbol{Y}_i)}.$$

   It is generally biased but consistent under the usual moment and support conditions. It is useful when $f(\boldsymbol{x}) = c\tilde{f}(\boldsymbol{x})$ is known only up to an unknown normalizing constant c: replacing f by $\tilde{f}$ leaves the ratio unchanged because c cancels.

2. Importance sampling can also estimate conditional expectations:

$$\begin{aligned}\mathbb{E}[g(\boldsymbol{X}) \mid \boldsymbol{X} \in A] &= \int_{\mathbb{R}^d} g(\boldsymbol{x}) \mathbb{1}_{\{\boldsymbol{x}\in A\}} \frac{f(\boldsymbol{x})}{\mathbb{P}(\boldsymbol{X} \in A)} \, \mathrm{d}\boldsymbol{x} \\ &= \frac{\int_{\mathbb{R}^d} g(\boldsymbol{x}) \mathbb{1}_{\{\boldsymbol{x}\in A\}} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}}{\int_{\mathbb{R}^d} \mathbb{1}_{\{\boldsymbol{x}\in A\}} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}} \\ &= \frac{\mathbb{E}_h\left[\mathbb{1}_{\{\boldsymbol{Y}\in A\}} g(\boldsymbol{Y}) f(\boldsymbol{Y})/h(\boldsymbol{Y})\right]}{\mathbb{E}_h\left[\mathbb{1}_{\{\boldsymbol{Y}\in A\}} f(\boldsymbol{Y})/h(\boldsymbol{Y})\right]}.\end{aligned}$$

   The numerator and denominator can be estimated separately, or together through a normalized importance sampling ratio.

3. For $d = 1$, a popular parametric choice of h_θ is the **exponential tilt** of f by a value θ in the interior of the set where $0 < M(\theta) < \infty$:

$$h_\theta(x) = \frac{e^{\theta x} f(x)}{M(\theta)} = e^{\theta x - K(\theta)} f(x),$$

   where $x, \theta \in \mathbb{R}$ and $M(\theta) = \int e^{\theta y} f(y) \, \mathrm{d}y$ is the moment generating function of $X \sim f$ and $K(\theta) = \log M(\theta)$ is the cumulant generating function of X. Positive θ shifts mass to the right tail, while negative θ shifts mass to the left tail.

Example 21.3 Consider

$$\mu = \mathbb{P}\left(\sum_{j=1}^d X_j > q\right),$$

where the X_j are independent with densities f_j. Tilt each density by the same parameter θ:

$$h_{j,\theta}(x) = e^{\theta x - K_j(\theta)} f_j(x),$$

and let $Y_{j,\theta} \sim h_{j,\theta}$ be independent. Under the tilted product density,

$$\begin{aligned}\mu &= \mathbb{E}_{h_\theta}\left[\mathbb{1}_{\{\sum_{j=1}^d Y_{j,\theta}>q\}} \prod_{j=1}^d \frac{f_j(Y_{j,\theta})}{h_{j,\theta}(Y_{j,\theta})}\right] \\ &= \mathbb{E}_{h_\theta}\left[\mathbb{1}_{\{\sum_{j=1}^d Y_{j,\theta}>q\}} \exp\left(-\theta \sum_{j=1}^d Y_{j,\theta} + \sum_{j=1}^d K_j(\theta)\right)\right].\end{aligned}$$

One common saddlepoint choice minimizes the exponential bound

$$\eta(\theta) = \exp\left(-\theta q + \sum_{j=1}^d K_j(\theta)\right)$$

with respect to θ. An interior minimizer θ^* satisfies

$$\frac{\mathrm{d}}{\mathrm{d}\theta}\eta(\theta) = 0 \iff \sum_{j=1}^d K_j'(\theta) = q.$$

When the cumulant generating functions are strictly convex and q lies in the relevant range, this equation has a unique solution because $K_j'(\theta) = \mathbb{E}_{h_{j,\theta}}[Y_{j,\theta}]$ is increasing in θ for each j.

Example 21.4 Let X be standard Cauchy and estimate

$$\mu = \mathbb{P}(X > 2).$$

The crude estimator is

$$\hat{\mu}_n^{\mathrm{cmc}} = \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{\{X_i>2\}}, \quad X_i \sim f.$$

Importance sampling with the helper function $h(y) = (2/y^2)\mathbb{1}_{\{y\geq 2\}}$ gives the estimator

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^n \frac{\mathbb{1}_{\{Y_i>2\}} f(Y_i)}{h(Y_i)} = \frac{1}{n}\sum_{i=1}^n \frac{Y_i^2}{2\pi(1+Y_i^2)}\mathbb{1}_{\{Y_i>2\}},$$

where $Y_1, \ldots, Y_n \overset{\mathrm{iid}}{\sim} h$. With $Y = 1/U \sim h$, we get $U \sim \mathrm{Unif}[0, 1/2]$, and we obtain

$$\mu = \mathbb{E}\left[\mathbb{1}_{\{1/U>2\}} \frac{1}{2\pi}\frac{1}{1+U^2}\right] = \int_0^{1/2} \frac{2}{2\pi(1+u^2)}\,\mathrm{d}u = \int_0^{1/2} f(u)\,\mathrm{d}u,$$

where f is the standard Cauchy density. We saw in Example 18.1 that this leads to an efficiency of 1317.

Lecture 22 — Tuesday, March 24, 2015

Stratified Sampling

Let Y be a discrete variable with

$$p_k = \mathbb{P}(Y = y_k), \quad k = 1, \ldots, K.$$

For $\mu = \mathbb{E}[g(\boldsymbol{X})]$, the law of total expectation gives

$$\mu = \sum_{k=1}^{K} p_k \, \mathbb{E}[g(\boldsymbol{X}) \mid Y = y_k].$$

For each **stratum** k, sample $\boldsymbol{X}_{k1}, \ldots, \boldsymbol{X}_{kn_k}$ from the conditional distribution of $\boldsymbol{X}$ given $Y = y_k$, and define

$$\bar{g}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} g(\boldsymbol{X}_{ki}).$$

The **stratified sampling estimator**

$$\hat{\mu}_n = \sum_{k=1}^{K} p_k \bar{g}_k$$

is unbiased. With independent sampling across strata,

$$\mathrm{Var}(\hat{\mu}_n) = \sum_{k=1}^{K} \frac{p_k^2}{n_k} \mathrm{Var}(g(\boldsymbol{X}) \mid Y = y_k).$$

If the proportional allocations $n_k = np_k$ are positive integers, then

$$\mathrm{Var}(\hat{\mu}_n) = \frac{1}{n} \mathbb{E}[\mathrm{Var}(g(\boldsymbol{X}) \mid Y)].$$

The law of total variance,

$$\mathrm{Var}(g(\boldsymbol{X})) = \mathrm{Var}(\mathbb{E}[g(\boldsymbol{X}) \mid Y]) + \mathbb{E}[\mathrm{Var}(g(\boldsymbol{X}) \mid Y)],$$

shows that proportional stratification reduces variance by

$$\mathrm{Var}(\hat{\mu}_n^{\mathrm{cmc}}) - \mathrm{Var}(\hat{\mu}_n^{\mathrm{strat}}) = \frac{1}{n}\,\mathrm{Var}(\mathbb{E}[g(\boldsymbol{X}) \mid Y]) \geq 0$$

relative to crude Monte Carlo. Although this **proportional stratified sampling estimator** for choosing n_k reduces variance, it is not necessarily optimal. The following result gives the optimal allocation of samples across strata.

Lemma 22.1 (Neyman allocation) Let

$$\sigma_k^2 = \mathrm{Var}(g(\boldsymbol{X}) \mid Y = y_k).$$

Assume $p_k\sigma_k > 0$ for every k. Among positive real allocations n_k satisfying $\sum_{k=1}^K n_k = n$, the variance is minimized by

$$n_k = n\frac{p_k\sigma_k}{\sum_{j=1}^K p_j\sigma_j},$$

with minimum value

$$\mathrm{Var}(\hat{\mu}_n) = \frac{1}{n}\left(\sum_{k=1}^K p_k\sigma_k\right)^2.$$

Proof. For $\boldsymbol{n} = (n_1, \ldots, n_K) \in (0,\infty)^K$, consider the Lagrangian

$$f(\boldsymbol{n}; a) = \sum_{k=1}^K \frac{p_k^2\sigma_k^2}{n_k} + a\left(\sum_{k=1}^K n_k - n\right)$$

and solve $\nabla f(\boldsymbol{n}; a) = \boldsymbol{0}$. This gives

$$-(p_k^2\sigma_k^2)/n_k^2 + a = 0, \quad k \in \{1, \ldots, K\}$$

and

$$\sum_{k=1}^K n_k - n = 0.$$

Solving, we get

$$n_k = \frac{p_k\sigma_k}{\sqrt{a}} \implies \sum_{k=1}^{K} \frac{p_k\sigma_k}{\sqrt{a}} = n \implies \frac{1}{\sqrt{a}} = \frac{n}{\sum_{k=1}^{K} p_k\sigma_k} \implies n_k = n\frac{p_k\sigma_k}{\sum_{j=1}^{K} p_j\sigma_j}.$$

The objective is strictly convex on the positive orthant, so this critical point is the unique global minimum. Substitution gives

$$\sum_{k=1}^{K} \frac{p_k^2\sigma_k^2}{n_k} = \frac{1}{n}\left(\sum_{k=1}^{K} p_k\sigma_k\right)^2.$$

□

In practice, the positive real allocation is rounded to feasible positive integers. Neyman allocation is usually implemented in a pilot-and-allocate step. First, choose a pilot sample size n_0 (for example, 1000) and, for each stratum k, draw $\boldsymbol{X}_{k1}, \ldots, \boldsymbol{X}_{kn_0}$ from the conditional distribution of $\boldsymbol{X}$ given $Y = y_k$. Define

$$\bar{g}_k^{(0)} = \frac{1}{n_0}\sum_{i=1}^{n_0} g(\boldsymbol{X}_{ki}) \qquad \text{and} \qquad s_k = \sqrt{\frac{1}{n_0 - 1}\sum_{i=1}^{n_0}\left(g(\boldsymbol{X}_{ki}) - \bar{g}_k^{(0)}\right)^2}.$$

Then estimate the Neyman allocation by

$$\hat{n}_k = \text{round}\left(n\frac{p_k s_k}{\sum_{j=1}^{K} p_j s_j}\right),$$

with a final adjustment to ensure $\sum_{k=1}^{K} \hat{n}_k = n$. Finally, compute $\hat{\mu}_n$ with these allocations and estimate its variance by

$$\widehat{\text{Var}}(\hat{\mu}_n) = \sum_{k=1}^{K} \frac{p_k^2 s_k^2}{\hat{n}_k}.$$

Remark 22.2

1. As the number of strata K increases and the strata become finer, the standard deviations σ_k within strata typically decrease, which can make the stratified estimator variance $\text{Var}(\hat{\mu}_n)$ very small.
2. Stratification also applies when Y has a continuous distribution function F_Y. We can

construct $Y = F_Y^-(U)$ from $U \sim \text{Unif}(0,1)$. Instead of drawing $U_1, \dots, U_n$ independently and setting $Y_i = F_Y^-(U_i)$, define

$$U_i^* = \frac{i - 1 + U_i}{n} \qquad \text{and} \qquad Y_i = F_Y^-(U_i^*), \quad i = 1, \dots, n,$$

where $U_1, \dots, U_n \overset{\text{i.i.d.}}{\sim} \text{Unif}(0,1)$. Then U_i^* is uniform on the ith interval $I_i = ((i-1)/n, i/n)$, and Y_i is stratified over the corresponding quantile slice. If we sample $\boldsymbol{X}_i$ from the conditional law of $\boldsymbol{X}$ given $Y = Y_i$, then

$$\mathbb{E}[g(\boldsymbol{X}_i)] = \mathbb{E}[g(\boldsymbol{X}) \mid U \in I_i] = n\,\mathbb{E}[g(\boldsymbol{X})\mathbb{1}_{\{U \in I_i\}}].$$

Hence

$$\hat{\mu}_n^{\text{cont-strat}} = \frac{1}{n}\sum_{i=1}^{n} g(\boldsymbol{X}_i)$$

is unbiased, because

$$\begin{aligned}
\mathbb{E}[\hat{\mu}_n^{\text{cont-strat}}] &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[g(\boldsymbol{X}_i)] \\
&= \sum_{i=1}^{n} \mathbb{E}[g(\boldsymbol{X})\mathbb{1}_{\{U \in I_i\}}] \\
&= \mathbb{E}[g(\boldsymbol{X})] = \mu.
\end{aligned}$$

3. The same idea extends to multivariate stratification when Y is a random vector $\boldsymbol{Y}$.

Example 22.3 For estimating π, use

$$\frac{\pi}{4} = \int_0^1 \sqrt{1 - x^2}\,\mathrm{d}x.$$

With continuous stratification on $[0,1]$, the estimator is

$$\hat{\pi}_n^{\text{strat}} = \frac{4}{n}\sum_{i=1}^{n} \sqrt{1 - \left(\frac{i - 1 + U_i}{n}\right)^2}, \quad U_i \overset{\text{i.i.d.}}{\sim} \text{Unif}(0,1).$$

This can be combined with antithetic variates inside each stratum:

$$\hat{\pi}_n^{\text{strat-av}} = \frac{4}{n}\sum_{i=1}^{n}\frac{1}{2}\left[\sqrt{1-\left(\frac{i-1+U_i}{n}\right)^2}+\sqrt{1-\left(\frac{i-U_i}{n}\right)^2}\right].$$

For $n = 10{,}000$, this estimator typically agrees with π to about six decimal places, although any individual randomized run can be less accurate.

How does stratifying on Y compare with using Y as a control variate? For proportional allocation $n_k \approx np_k$,

$$\mathrm{Var}(\hat{\mu}_n^{\text{strat}}) = \frac{1}{n}\mathbb{E}[\mathrm{Var}(g(\boldsymbol{X}) \mid Y)].$$

Using Y as a control variate yields

$$\mathrm{Var}(\hat{\mu}_n^{\text{ocv}}) = \frac{1}{n}\left(\mathrm{Var}(g(\boldsymbol{X})) - \frac{\mathrm{Cov}(g(\boldsymbol{X}),Y)^2}{\mathrm{Var}(Y)}\right).$$

Lemma 22.4 For square-integrable random variables X and Y with $\mathrm{Var}(Y) > 0$,

$$\mathbb{E}[\mathrm{Var}(X \mid Y)] \le \mathrm{Var}(X) - \frac{\mathrm{Cov}(X,Y)^2}{\mathrm{Var}(Y)}.$$

Consequently,

$$\mathrm{Var}(\hat{\mu}_n^{\text{strat}}) \le \mathrm{Var}(\hat{\mu}_n^{\text{ocv}}).$$

Proof. We have

$$\mathrm{Cov}(X,Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] = \mathbb{E}[\mathbb{E}[XY \mid Y]] - \mathbb{E}[\mathbb{E}[X \mid Y]]\mathbb{E}[Y] = \mathrm{Cov}(\mathbb{E}[X \mid Y], Y).$$

Also,

$$\mathbb{E}[\mathrm{Var}(X \mid Y)] = \mathrm{Var}(X) - \mathrm{Var}(\mathbb{E}[X \mid Y]).$$

Therefore it suffices to show

$$\mathrm{Var}(\mathbb{E}[X \mid Y]) \ge \frac{\mathrm{Cov}(X,Y)^2}{\mathrm{Var}(Y)} = \frac{\mathrm{Cov}(\mathbb{E}[X \mid Y], Y)^2}{\mathrm{Var}(Y)}.$$

If $\mathrm{Var}(\mathbb{E}[X \mid Y]) = 0$, then $\mathrm{Cov}(\mathbb{E}[X \mid Y], Y) = 0$ and the desired inequality is immediate.

Otherwise, it follows from

$$\frac{\operatorname{Cov}(\mathbb{E}[X \mid Y], Y)^2}{\operatorname{Var}(\mathbb{E}[X \mid Y]) \operatorname{Var}(Y)} = \operatorname{Corr}(\mathbb{E}[X \mid Y], Y)^2 \leq 1.$$

□

Lecture 23 — Tuesday, March 31, 2015

Latin Hypercube Sampling

Consider

$$\mu = \mathbb{E}\left[g(\boldsymbol{U})\right] = \int_0^1 \cdots \int_0^1 g(u_1, \ldots, u_d) \, \mathrm{d}\boldsymbol{u}.$$

The crude Monte Carlo estimator is

$$\hat{\mu}_n^{\text{cmc}} = \frac{1}{n} \sum_{i=1}^{n} g(\boldsymbol{U}_i),$$

where $\boldsymbol{U}_i \overset{\text{i.i.d.}}{\sim} \text{Unif}[0,1]^d$. Similar to stratification (or quasi-Monte Carlo methods), **Latin hypercube sampling** uniformly fills $[0,1]^d$ in the following way:

Algorithm 20: Latin hypercube sampling

1 Partition each coordinate axis into n intervals $((k-1)/n, k/n]$, $k = 1, \ldots, n$
2 **for** $j = 1, \ldots, d$ **do**
3 ⌊ Choose a random permutation $\boldsymbol{\pi}_j = (\pi_{j,1}, \ldots, \pi_{j,n})$ of $\{1, \ldots, n\}$
4 Generate $U'_{ij} \overset{\text{i.i.d.}}{\sim} \text{Unif}[0,1]$ for all i, j
5 Form $\boldsymbol{U}_i^* = (U_{i1}^*, \ldots, U_{id}^*)$ with $U_{ij}^* = (U'_{ij} + \pi_{j,i} - 1)/n$ for all i, j
6 **return** $\hat{\mu}_n^{lhs} = n^{-1} \sum_{i=1}^n g(\boldsymbol{U}_i^*)$

For each fixed i and j, the marginal distribution of U_{ij}^* is uniform on $[0,1]$. Indeed, for $u \in ((l-1)/n, l/n]$,

$$\begin{aligned}
\mathbb{P}(U_{ij}^* \leq u) &= \mathbb{P}(U_{ij}^* \in (0, u]) \\
&= \mathbb{P}\left(U_{ij}^* \in (0, u] \cap \bigcup_{k=1}^{n} \left(\frac{k-1}{n}, \frac{k}{n}\right]\right)
\end{aligned}$$

$$= \sum_{k=1}^{n} \mathbb{P}\left(U_{ij}^* \in (0,u] \cap \left(\frac{k-1}{n}, \frac{k}{n}\right]\right),$$

where the last step uses the disjointness of the intervals. Moreover,

$$\mathbb{P}\left(U_{ij}^* \in (0,u] \cap \left(\frac{k-1}{n}, \frac{k}{n}\right]\right) = \begin{cases} \frac{1}{n}, & k \in \{1,\dots,l-1\}, \\ u - \frac{l-1}{n}, & k = l, \\ 0, & k \in \{l+1,\dots,n\}. \end{cases}$$

Hence

$$\mathbb{P}(U_{ij}^* \le u) = \sum_{k=1}^{l-1} \frac{1}{n} + \left(u - \frac{l-1}{n}\right) + \sum_{k=l+1}^{n} 0 = u,$$

so $U_{ij}^* \sim \text{Unif}(0,1)$.

Since the permutations $\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_d$ and jitters $U'_{i1}, \dots, U'_{id}$ are independent across coordinates, the components of $\boldsymbol{U}_i^*$ are independent for each fixed i. Therefore $\boldsymbol{U}_i^* \sim \text{Unif}([0,1]^d)$ for every i, and so

$$\mathbb{E}[\hat{\mu}_n^{\text{lhs}}] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[g(\boldsymbol{U}_i^*)] = \mu.$$

Hence $\hat{\mu}_n^{\text{lhs}}$ is unbiased. However, the vectors $\boldsymbol{U}_1^*, \dots, \boldsymbol{U}_n^*$ are not independent, because for each coordinate j, the values $U_{1j}^*, \dots, U_{nj}^*$ are coupled through the permutation $\boldsymbol{\pi}_j$. Typically, but not always, $\text{Var}(\hat{\mu}_n^{\text{lhs}}) \le \text{Var}(\hat{\mu}_n^{\text{cmc}})$; this holds for componentwise monotone functions g.

Other Variance Reduction Techniques

Conditioning replaces a simulated random variable by its conditional expectation. Suppose our aim is to estimate

$$\mu = \mathbb{E}[Y] = \mathbb{E}[g(\boldsymbol{X})],$$

where Y depends on an auxiliary random variable $Z \sim F_Z$. Assume, for this derivation, that (Y, Z) has joint density $f_{Y,Z}$ and marginal densities f_Y and f_Z. For z with $f_Z(z) > 0$, we have

$$\mathbb{E}[Y \mid Z = z] = \int_{\mathbb{R}} y \, f_{Y|Z}(y \mid z) \, \mathrm{d}y \qquad \text{where} \qquad f_{Y|Z}(y \mid z) = \frac{f_{Y,Z}(y,z)}{f_Z(z)}.$$

Then

$$\mu = \mathbb{E}[Y] = \int_{\mathbb{R}} y \, f_Y(y) \, \mathrm{d}y$$

$$
\begin{aligned}
&= \int_{\mathbb{R}} y \left(\int_{\mathbb{R}} f_{Y,Z}(y,z)\,\mathrm{d}z \right) \mathrm{d}y \\
&= \int_{\mathbb{R}} y \left(\int_{\mathbb{R}} \frac{f_{Y,Z}(y,z)}{f_Z(z)} f_Z(z)\,\mathrm{d}z \right) \mathrm{d}y \\
&= \int_{\mathbb{R}} \left(\int_{\mathbb{R}} y\, f_{Y|Z}(y \mid z)\,\mathrm{d}y \right) f_Z(z)\,\mathrm{d}z \\
&= \int_{\mathbb{R}} \mathbb{E}[Y \mid Z = z] f_Z(z)\,\mathrm{d}z \\
&= \mathbb{E}[\mathbb{E}[Y \mid Z]].
\end{aligned}
$$

Hence $\mathbb{E}[Y \mid Z]$ is an unbiased estimator of μ. If $\mathbb{E}[Y \mid Z = z]$ is available in closed form, simulate $Z_1, \ldots, Z_n \overset{\text{i.i.d.}}{\sim} F_Z$ independently and use

$$\hat{\mu}_n^{\text{cond}} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[Y \mid Z = Z_i].$$

Its variance is

$$\mathrm{Var}(\hat{\mu}_n^{\text{cond}}) = \frac{\mathrm{Var}(\mathbb{E}[Y \mid Z])}{n} = \frac{\mathrm{Var}(Y) - \mathbb{E}[\mathrm{Var}(Y \mid Z)]}{n} \leq \frac{\mathrm{Var}(Y)}{n} = \mathrm{Var}(\hat{\mu}_n^{\text{cmc}}),$$

where $\hat{\mu}_n^{\text{cmc}} = n^{-1} \sum_{i=1}^{n} Y_i$ with $Y_1, \ldots, Y_n$ independent and each distributed as Y.

To verify the decomposition above, first compute

$$
\begin{aligned}
\mathrm{Var}(Y \mid Z) &= \mathbb{E}\left[(Y - \mathbb{E}[Y \mid Z])^2 \mid Z \right] \\
&= \mathbb{E}[Y^2 \mid Z] - 2\mathbb{E}[Y\, \mathbb{E}[Y \mid Z] \mid Z] + \mathbb{E}[\mathbb{E}[Y \mid Z]^2 \mid Z] \\
&= \mathbb{E}[Y^2 \mid Z] - \mathbb{E}[Y \mid Z]^2.
\end{aligned}
$$

Taking expectations gives

$$\mathbb{E}[\mathrm{Var}(Y \mid Z)] = \mathbb{E}[Y^2] - \mathbb{E}[\mathbb{E}[Y \mid Z]^2].$$

Also,

$$\mathrm{Var}(\mathbb{E}[Y \mid Z]) = \mathbb{E}[\mathbb{E}[Y \mid Z]^2] - \mathbb{E}[\mathbb{E}[Y \mid Z]]^2.$$

Adding these identities yields

$$\mathrm{Var}(\mathbb{E}[Y \mid Z]) + \mathbb{E}[\mathrm{Var}(Y \mid Z)] = \mathrm{Var}(Y),$$

which is the law of total variance.

Example 23.1 Suppose we want to estimate π using the identity for a quarter disk,

$$\frac{\pi}{4} = \int_0^1 \sqrt{1-x^2}\,\mathrm{d}x.$$

The hit-or-miss estimator from Example 17.3 is

$$\hat{\pi}_n^{\mathrm{cmc}} = \frac{4}{n}\sum_{i=1}^{n} \mathbb{1}_{\{U_i^2+V_i^2\le 1\}},$$

where $U_i, V_i \sim \mathrm{Unif}[0,1]$. Conditioning on U gives

$$\mathbb{E}\left[\mathbb{1}_{\{U^2+V^2\le 1\}} \mid U\right] = \mathbb{E}\left[\mathbb{1}_{\{V\le\sqrt{1-U^2}\}} \mid U\right] = \mathbb{P}\left(V \le \sqrt{1-U^2} \mid U\right) = \sqrt{1-U^2}.$$

Therefore the conditioned estimator is

$$\hat{\pi}_n^{\mathrm{cond}} = \frac{4}{n}\sum_{i=1}^{n} \sqrt{1-U_i^2}.$$

Its variance is

$$\begin{aligned}
\mathrm{Var}(\hat{\pi}_n^{\mathrm{cond}}) &= \frac{16}{n}\,\mathrm{Var}(\sqrt{1-U^2}) \\
&= \frac{16}{n}\left(\mathbb{E}\left[1-U^2\right] - (\mathbb{E}\left[\sqrt{1-U^2}\right])^2\right) \\
&= \frac{16}{n}\left(1-\frac{1}{3}-\left(\frac{\pi}{4}\right)^2\right) \approx \frac{0.7974}{n},
\end{aligned}$$

whereas

$$\mathrm{Var}(\hat{\pi}_n^{\mathrm{cmc}}) = \frac{16}{n}\,\mathrm{Var}(\mathbb{1}_{U^2+V^2\le 1}) = \frac{16}{n}\left(\frac{\pi}{4}\right)\left(1-\frac{\pi}{4}\right) \approx \frac{2.6968}{n}.$$

The variance reduction factor is about 3.38, corresponding to about 70.4% variance reduction. For an even total budget n of function evaluations, antithetic variates give

$$\hat{\pi}_n^{\mathrm{av}} = \frac{4}{n/2}\sum_{i=1}^{n/2} \frac{\sqrt{1-U_i^2}+\sqrt{1-(1-U_i)^2}}{2},$$

with variance approximately $0.2195/n$, so the efficiency relative to crude Monte Carlo is about 12.29 (about 91.86% variance reduction). Using U^2 as an optimal control variate

gives

$$\hat{\pi}_n^{\text{ocv}} = \frac{4}{n}\sum_i \left(\sqrt{1-U_i^2} - \beta^*\left(U_i^2 - \frac{1}{3}\right)\right), \qquad \beta^* = \frac{\text{Cov}(\sqrt{1-U^2}, U^2)}{\text{Var}(U^2)} \approx -0.7363,$$

with variance approximately $0.0260/n$, which corresponds to efficiency about 103.7 (about 99.04% variance reduction).

Another possibility is to combine several unbiased Monte Carlo estimators. Let

$$\hat{\boldsymbol{\mu}}_n = (\hat{\mu}_{n,1}, \ldots, \hat{\mu}_{n,K})^\top \qquad \text{with} \qquad \mathbb{E}[\hat{\mu}_{n,k}] = \mu.$$

For weights satisfying $\boldsymbol{\lambda}^\top \mathbf{1} = 1$, the combined estimator

$$\hat{\mu}_n = \boldsymbol{\lambda}^\top \hat{\boldsymbol{\mu}}_n$$

is unbiased.

Proposition 23.2 Suppose the covariance matrix

$$\boldsymbol{\Sigma} = \big(\text{Cov}(\hat{\mu}_{n,i}, \hat{\mu}_{n,j})\big)_{i,j=1}^K$$

is positive definite. The weight vector minimizing $\text{Var}(\boldsymbol{\lambda}^\top \hat{\boldsymbol{\mu}}_n)$ subject to $\boldsymbol{\lambda}^\top \mathbf{1} = 1$ is

$$\boldsymbol{\lambda}^* = \frac{\boldsymbol{\Sigma}^{-1}\mathbf{1}}{\mathbf{1}^\top \boldsymbol{\Sigma}^{-1}\mathbf{1}},$$

and the resulting variance is

$$\text{Var}\Big((\boldsymbol{\lambda}^*)^\top \hat{\boldsymbol{\mu}}_n\Big) = \frac{1}{\mathbf{1}^\top \boldsymbol{\Sigma}^{-1}\mathbf{1}}.$$

Proof. To find the unbiased linear combination with minimum variance, minimize

$$\boldsymbol{\lambda}^\top \boldsymbol{\Sigma} \boldsymbol{\lambda}$$

subject to

$$\boldsymbol{\lambda}^\top \mathbf{1} = 1.$$

Introduce the Lagrangian

$$L(\boldsymbol{\lambda}, a) = \boldsymbol{\lambda}^\top \boldsymbol{\Sigma} \boldsymbol{\lambda} + a(\boldsymbol{\lambda}^\top \mathbf{1} - 1).$$

The first-order conditions are

$$\nabla_{\boldsymbol{\lambda}} L(\boldsymbol{\lambda}, a) = 2\boldsymbol{\Sigma}\boldsymbol{\lambda} + a\mathbf{1} = \mathbf{0} \qquad \text{and} \qquad \partial_a L(\boldsymbol{\lambda}, a) = \boldsymbol{\lambda}^\top \mathbf{1} - 1 = 0.$$

Since $\boldsymbol{\Sigma}$ is invertible, the first equation gives

$$\boldsymbol{\lambda} = -\frac{a}{2}\boldsymbol{\Sigma}^{-1}\mathbf{1}.$$

Substitute this into the constraint:

$$\left(-\frac{a}{2}\boldsymbol{\Sigma}^{-1}\mathbf{1}\right)^\top \mathbf{1} = 1 \iff -\frac{a}{2}\mathbf{1}^\top\boldsymbol{\Sigma}^{-1}\mathbf{1} = 1 \iff -\frac{a}{2} = \frac{1}{\mathbf{1}^\top\boldsymbol{\Sigma}^{-1}\mathbf{1}}.$$

Therefore

$$\boldsymbol{\lambda}^* = \frac{\boldsymbol{\Sigma}^{-1}\mathbf{1}}{\mathbf{1}^\top\boldsymbol{\Sigma}^{-1}\mathbf{1}}.$$

Now evaluate the variance at $\boldsymbol{\lambda}^*$:

$$\begin{aligned}
\operatorname{Var}\left((\boldsymbol{\lambda}^*)^\top \hat{\boldsymbol{\mu}}_n\right) &= (\boldsymbol{\lambda}^*)^\top \boldsymbol{\Sigma} \boldsymbol{\lambda}^* \\
&= \left(\frac{\mathbf{1}^\top\boldsymbol{\Sigma}^{-1}}{\mathbf{1}^\top\boldsymbol{\Sigma}^{-1}\mathbf{1}}\right)\boldsymbol{\Sigma}\left(\frac{\boldsymbol{\Sigma}^{-1}\mathbf{1}}{\mathbf{1}^\top\boldsymbol{\Sigma}^{-1}\mathbf{1}}\right) \\
&= \frac{\mathbf{1}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}\mathbf{1}}{(\mathbf{1}^\top\boldsymbol{\Sigma}^{-1}\mathbf{1})^2} \\
&= \frac{\mathbf{1}^\top\boldsymbol{\Sigma}^{-1}\mathbf{1}}{(\mathbf{1}^\top\boldsymbol{\Sigma}^{-1}\mathbf{1})^2} \\
&= \frac{1}{\mathbf{1}^\top\boldsymbol{\Sigma}^{-1}\mathbf{1}}.
\end{aligned}$$

□

Appendix: Lecture and Tutorial Index