

Chapter 3

The Central Limit Theorem, Law of Large Numbers and Monte Carlo Methods

3.1 Computer Experimentation and Simulation

As we shall see, computation plays a key role in the analysis of stochastic systems. A number of models that we shall study are amenable to computational formulations that involve solving:

- systems of linear equations
- systems of linear ordinary differential equations
- partial differential equations

However, perhaps the most widely used computational tool in studying stochastic systems is the use of computer experimentations (i.e. “simulations”).

Example 3.1: Suppose that we wish to compute the probability of a “straight” in a hand of four cards that are randomly dealt from a deck of 52 cards. (A “straight” is a hand in which the cards are consecutively ordered, regardless of the suit: e.g. 3,4,5,6 or 5,6,7,8). This can be computed analytically. But a conceptually more straightforward means of solving this problem would be to deal a large number of hands (e.g. a thousand hands), and to estimate the probability of a straight with the fraction of the hands on which straights were dealt.

Of course, this could take hours or days to complete. An alternative is to engage the power of the computer to simulate this situation. In other words, use the computer to simulate thousands of hands, followed by computing the fraction of hands on which straights occur. For example, suppose that the computer has available a “uniform random number generator” that generates a sequence of U_1, U_2, \dots of iid uniform random variables on $[0, 1]$. (We will return to the question of uniform random number generation later.) If we generate 52 such uniform random numbers and rank order the 52 uniform random variables, we obtain a permutation of the integers: $1, 2, \dots, 52$. To illustrate the idea, suppose that we generate three uniform random variables U_1, U_2, U_3 say, 0.61, 0.23, 0.47. This corresponds to the random permutation 1, 3, 2 (0.61 is the largest, 0.47 is the second largest, and 0.23 is the third largest).

Once we have the permutation of the 52 integers, we can immediately identify this with a random shuffle of a

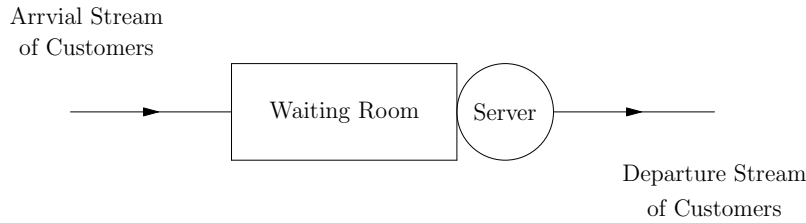


Figure 3.1: Single Server Queue

52 card deck, after which it is straightforward to determine whether the first four cards dealt correspond to a straight or not. By repeatedly generating new sequences of 52 iid uniform random variables, the computer is generating new random shuffles of the deck. In a matter of seconds, the computer can conduct an enormous number of statistically independent computer experiments, thereby permitting an accurate computation of the probability of a straight.

As we shall see, this approach to numerical computation for stochastic analysis is enormously powerful and widely used.

3.2 Performance Engineering: The Single Server Queue

One important application of stochastic modeling lies in the domain of “performance engineering.” Performance engineering relates to the design and analysis of complex man-made systems in which the principal determinant of performance tends to be the algorithms of policies used to govern the system (as opposed to physical laws). Many such man-made systems exhibit congestion effects:

- packets of a router on the Internet
- requests for a web-page on a server farm
- information requests from a database
- satisfying customer demand from a warehouse

As an illustration of the use of simulation, consider the simplest possible example of a model exhibiting congestion effects: the single-server queue (see Figure 3.1).

In the simplest version of this model, we assume:

- infinite capacity waiting room (i.e. buffer)
- single server
- first in / first out (FIFO) queue discipline
- iid inter-arrival times: χ_1, χ_2, \dots
- iid service time requirements: V_0, V_1, \dots

Remark 3.1: While the FIFO queue discipline is the one that most frequently arises, others come up as well (e.g. LIFO (last in / first out), processor sharing (PS), priority queues (different customer classes have different priorities), shortest remaining processing time first (SRPT), etc).

Remark 3.2: The queue described above is called the G/G/1 queue (first G = generalized inter-arrival time distribution, second G = generalized service time requirement distribution, 1 = single server).

Time	Time of Next Arrival	Time of Next Departure	$Q(\cdot)$
0	2.1	-	0
2.1	2.7	3.9	1
2.7	6.9	3.9	2
3.9	6.9	8.0	1
6.9	8.8	8.0	2
8.0	8.8	10.3	1
8.8	9.5	10.3	2
9.5	10.0	10.3	3
10.0	17.1	10.3	4

Table 3.1: Event Schedule

3.3 Discrete-Event Simulations

The single-server queue is a simple example of a discrete-event system. Discrete-event systems are systems in which the state transitions occur at discrete epochs. For example, the state transitions of a single-server queue occur at the arrival and departure times.

Simulation of discrete-event systems uses the principles of “future event scheduling” and “asynchronous simulation.” We illustrate these ideas in the setting of the single-server queue.

As for Example 1, we assume the existence of a uniform random number generator (RNG). Suppose that the common distribution of the inter-arrival times is F_χ , and that the common distribution of the service times is F_V . Suppose that there exist algorithms for transforming uniform random variables into non-uniform random variables having the distributions F_χ and F_V . (Such algorithms will be discussed later.) Assume that the first seven inter-arrival times generated are:

$$2.1, 0.6, 4.2, 1.9, 0.7, 0.5, 3.1$$

and the first six service time requirements are:

$$1.8, 4.1, 2.3, 0.5, 7.1, 3.4$$

One can then simulate the process $Q = (Q(t) : t \geq 0)$ via Table 3.1 (which tracks future events to be “scheduled”).

Note that time is not incremented in fixed steps (“asynchronous simulations”). The key to this approach is to track the “next most imminent event”. While this is trivial for the single-server queue model just described, it becomes time-consuming for models involving more complex queue disciplines or involving networks of queues (e.g. the Internet or call centers). For more complex such discrete-event systems, the choice of data structures to maintain the future event schedule can have a significant impact on computational speed.

References

Law, A.M. and W.D. Kelton, Simulation Modeling and Analysis, McGraw Hill: 2000. Chapters 1-3.

3.4 Generating Non-Uniform Random Variables

Our discrete-event simulation example illustrates the need for algorithms capable of transforming uniform random numbers into a given non-uniform distribution. The general problem can be stated as follows:

Given: A sequence U_1, U_2, \dots of iid uniform $(0, 1)$ random variables

Goal: A sequence X_1, X_2, \dots of iid random variables with distribution function $F(\cdot)$.

We will describe two methods here, each with its own advantages and disadvantages.

3.4.1 Method 1 (Inversion)

Note that the function $F(\cdot)$ is non-decreasing. Such a non-decreasing function possesses a “generalized function inverse”, given by:

$$F^{-1}(x) = \inf\{y : F(y) \geq x\}$$

If F is continuous and strictly increasing, the generalized inverse is a function for which:

$$F(F^{-1}(x)) = F^{-1}(F(x)) = x$$

Algorithm A1: Inversion Algorithm

1	Generate a uniform $(0,1)$ random variable u
2	Return $X = F^{-1}(u)$

The random variable X returned at Step 2 has distribution F .

Example 3.2: Suppose that we wish to generate a Bernoulli random variable with parameter p so that:

$$F(x) = \begin{cases} 0 & x < 0 \\ 1 - p & 0 \leq x < 1 \\ 1 & x \geq 1 \end{cases}$$

Here,

$$F^{-1}(x) = \begin{cases} 0 & x \leq 1 - p \\ 1 & x > 1 - p \end{cases}$$

for $0 \leq x \leq 1$. To generate a Bernoulli random variable with parameter p , generate a uniform $(0, 1)$ random variable U , and set

$$X = \begin{cases} 0 & U \leq 1 - p \\ 1 & \text{o.w.} \end{cases}$$

Example 3.3: Suppose that we wish to generate a random variable X with probability mass function

$$P\{X = x_i\} = p_i, \quad 1 \leq i \leq n$$

where $x_1 < \dots < x_n$ with $p_1 + \dots + p_n = 1$. The method of inversion reduces here to generating a uniform $(0, 1)$ U and setting

$$X = \begin{cases} x_1 & U \leq p_1 \\ x_2 & p_1 < U \leq p_1 + p_2 \\ x_3 & p_1 + p_2 < U \leq p_1 + p_2 + p_3 \\ \vdots & \\ x_n & U > p_1 + \dots + p_{n-1} \end{cases}$$

Example 3.4: To generate an exponentially distributed random variable X having parameter $\lambda > 0$ recall that:

$$F(x) = \begin{cases} 0 & x \leq 0 \\ 1 - e^{-\lambda x} & x \geq 0 \end{cases}$$

If $y = F^{-1}(x)$, then $F(y) = x$ so that :

$$1 - e^{-\lambda y} = x \quad \text{i.e. } y = -\frac{1}{\lambda} \log(1 - x)$$

Hence, $F^{-1}(x) = -\lambda^{-1} \log(1 - x)$ for $0 < x < 1$. To generate the required exponential random variable, generate a uniform $(0, 1)$, random variable U and return

$$X = -\frac{1}{\lambda} \log(1 - U)$$

Remark 3.3: Note that $1 - U \stackrel{D}{=} U$ (here, $\stackrel{D}{=}$ denotes “has the same distribution as”) so that the modified algorithm:

$$X = -\frac{1}{\lambda} \log(U)$$

also returns an exponentially distributed random variable with parameter $\lambda > 0$.

Remark 3.4: Inversion is a particularly efficient algorithm when $F^{-1}(\cdot)$ can be efficiently computed. This usually requires that F and F^{-1} can be computed in “closed form”. (Note that such closed forms do not exist for certain important distributions e.g. normally distributed random variables.)

Remark 3.5: The use of inversion asserts that the required random variable X is a root of the equation

$$F(X) = U$$

One potential implementation of inversion therefore involves the use of numerical root-finding (e.g. Newton’s method) to compute the (random) root X to the above equation.

Our second algorithm for generating non-uniform random variables is the method of “acceptance-rejection”.

3.4.2 Method 2 (Acceptance-Rejection)

Suppose that the required random variable X has a density $f(\cdot)$. We assume that there exists a random variable Z (that can be cheaply generated) having a density g for which

$$c \triangleq \sup\{f(x)/g(x) : x \in \mathbb{R}\} < \infty$$

Algorithm A2

Algorithm A2: Acceptance-Rejection Algorithm

1	Generate the random variable Z
2	Generate an independent uniform $(0,1)$ random variable U .
3	Is $U \leq f(Z)/(c \cdot g(Z))$?
4	Yes. Return $X = Z$ (‘‘accept Z ’’)
5	No. Go to 1. (‘‘reject Z ’’)

Example 3.5: Suppose that we wish to generate a random variable X having distribution function

$$F(x) = \begin{cases} 0 & x \leq 0 \\ \frac{1}{2}x^5 + \frac{1}{2}x^4 & 0 \leq x \leq 1 \\ 1 & x \geq 1 \end{cases}$$

Note that use of inversion requires computing an (approximate) root X of :

$$\frac{1}{2}X^5 + \frac{1}{2}X^4 - U = 0$$

where U is a uniform $(0, 1)$ random variable. Such a root would need to be computed numerically.

An alternative is to use acceptance-rejection. We start by computing the density f . Recall that:

$$f(x) = \frac{d}{dx}F(x) = \frac{5}{2}x^4 + 2x^3$$

Note that $f(x) \leq c \cdot g(x)$ where $c = 9/2$ and $g(\cdot)$ is the density of a uniform $(0, 1)$ random variable U . Hence, the acceptance-rejection algorithm takes the form

1. Generate a uniform random variable U_1 on $(0, 1)$
 2. Generate an independent uniform $(0, 1)$ random variable U_2
 3. Is $U_2 \leq \frac{2}{9}(\frac{5}{2}U_1^4 + 2U_1^3)$?
- Yes. Return $X = U_1$.
No. Go to 1.

Remark 3.6: Let W be the number of times steps 1 through 3 are executed in order to return an X . Note that W is geometric with parameter $1/c$, so $E[W] = c$. Hence, the smaller the value of c , the more efficient the algorithm.

Remark 3.7: Generating uniform $(0, 1)$ random variables is (typically) very cheap. The computer time required to generate a uniform $(0, 1)$ random variable is generally (much) smaller than the time required to execute a floating point operation (FLOP), perhaps 20% of a FLOP.

Remark 3.8: The method of acceptance-rejection generalizes easily to the multivariate setting. Suppose that we wish to generate a random vector X having density function $f(\cdot)$. Assume that there exists a random vector Z (having the same dimension, d , as X) having a density g for which $f(x) \leq cg(x)$ for all $x \in \mathbb{R}^d$. Algorithm A2 applies without change in this multidimensional setting.

Before concluding this section, we justify our two algorithms.

Justification for Algorithm A1 (when F is continuous and F is strictly increasing)

$$P\{F^{-1}(U) \leq x\} = P\{F(F^{-1}(U)) \leq F(x)\} = P\{U \leq F(x)\} = F(x)$$

And now for Algorithm A2, Note that:

$$P\{Z \in dx | \text{acceptance}\} = \frac{P\{Z \in dx, \text{acceptance}\}}{P\{\text{acceptance}\}}$$

But

$$P\{Z \in dx, \text{acceptance}\} = g(x)dx \cdot P\{\text{acceptance} | Z \in dx\} = g(x)dx \cdot \frac{f(x)}{cg(x)} = \frac{f(x)dx}{c}$$

So,

$$P\{\text{acceptance}\} = \int_{\mathbb{R}} P\{Z \in dx, \text{acceptance}\} = \int_{\mathbb{R}} \frac{f(x)}{c} dx = \frac{1}{c}$$

yielding

$$P\{Z \in dx | \text{acceptance}\} = f(x)dx$$

References

Ripley, B.D. Stochastic Simulation. John Wiley (1987).

3.5 Generating Uniform Random Variables

Conceptually speaking, there are two main approaches to uniform random variables:

- physical generators
- mathematical algorithms

Here is one approach to “physical generation” of uniform random numbers. Suppose that we flip a fair coin n times, and encode heads and tails as 1 and 0, respectively. The n outcomes Z_1, Z_2, \dots, Z_n are n iid Bernoulli random variables with parameter $1/2$. Set:

$$\tilde{U} = \sum_{i=1}^n Z_i 2^{-i}$$

Note that \tilde{U} is a random “dyadic number” for which

$$P(\tilde{U} = j2^{-n}) = 2^{-n}$$

for $0 \leq j \leq 2^n$. As $n \rightarrow \infty$, evidently \tilde{U} converges to

$$\sum_{i=1}^{\infty} Z_i 2^{-i}$$

which is uniformly distributed on $[0, 1)$. Hence, \tilde{U} is an approximation to a uniform random variable. By increasing n we can generate a random variable arbitrarily close to an uniform random variable. By induction mn coin flips, we obtain m (approximately) uniform random variables. Of course, this approach to generating uniform random variables is (enormously) time consuming.

A faster means of generating random numbers is to record charged particle emissions in a vacuum tube. Under reasonable physical postulates, the times at which charged particle emissions occur should follow a Poisson process (to be discussed later in the course). A Poisson process is a process in which the event inter-occurrence times are iid exponentially distributed random variables. Call the inter-event times τ_1, τ_2, \dots . Then:

$$U_i = 1 - e^{-\lambda\tau_i}$$

defines a sequence of iid uniform $(0, 1)$ random variables (assuming the exponential parameter for τ_i is $\lambda > 0$). There are several problems with using such physical generators to generate uniform random numbers:

1. The question of whether such sequences are truly random can not be fully resolved with scientific or logical means. In particular, if one’s scientific world view is that the universe is governed by deterministic physical laws (and that randomness is merely a manifestation of our ignorance of the fundamental organizing principles of the universe), then truly random physical generators can not exist.

2. Physical generators always have hidden biases that may be invisible to the user. For example, if the coin is not fair, the random numbers that are produced are not truly uniform. Regarding vacuum tube technology, the recording devices invariably have an associated “locking period”. During the locking period, particle registrations can not be registered. Hence, extremely short inter-event emission times are not registered by the device. So, the distribution of inter-event times is not exponential (even if the underlying physical process is Poisson), leading to a systematic (and subtle) bias.
3. Certain simulation algorithms are based on re-use of random number streams. One could accomplish this by storing the stream of physical random numbers generated. Depending on the number of random numbers needed and fast memory available, this may (or may not) be feasible.
4. Physical generators tend to be slow. For example, vacuum tube based physical generators generate (at best) hundreds of random numbers per second. This is slow, relative to modern computers.

Remark 3.9: A famous (and early) listing of such random number can be found in:

Rand Corporation. A Million Random Digits with 100,000 Normal Deviates. Free Press, Glencoe, Illinois (1955).

3.5.1 Linear Congruential Generators

The most common means of generating uniform random variables on a computer is to use mathematical algorithms. The most widely used such algorithms are “linear congruential generators”. Such generators follow recursions of the form:

$$u_{i+1} = (au_i + b) \pmod{m}$$

where:

- u_0 = initial (seed) value (chosen as an integer between 0 and $n-1$)
- a = integer (called the “multiplier”)
- b = integer (called the “increment”)
- m = integer (called the “modulus”)

Remark 3.10: The above recursion uses “modular arithmetic”. If z and m are non-negative integers, $z \pmod{m}$ is the integer remainder that remains after z is divided by m (e.g. $11 \pmod{7} = 4$).

Given the above recursion, we obtain our uniform random numbers via:

$$U_i = \frac{u_i}{n}$$

Sequences $(U_i : i \geq 0)$ can (obviously) not be truly random. The hope is that if a , b and m are carefully chosen, then $(U_i : i \geq 1)$ can behave (in a statistical sense) like a truly random iid uniform sequence. In other words, the U_i 's pass a carefully chosen battery of statistical tests that are consistent with the hypothesis that the U_i 's are iid uniform $(0, 1)$ random variables.

Linear congruential generators (LCGs) have some structural properties that are well understood. It seems reasonable to choose a , b and m so that the recursion exhibits a full period prior to repeating itself. For a multiplicative LCG in which $b = 0$, full period means that the recursion cycles through each integer belonging

to $\{1, 2, \dots, m - 1\}$ once and only once before repeating itself. (If the generator does not have full period, “gaps” in the set of integers visited by the recursion may exist, creating potential non-uniformities in the U_i s).

Definition 3.1: An integer a is said to be a primitive element modulo m if the smallest integer l for which $a^l - 1$ is divisible

Result 3.1: Consider a multiplicative LCG with m prime. The, the LCG achieves full period if a is a primitive element modulo m .

It is fortuitous that $2^{31} - 1$ is prime. (On 32 bit computers, this is essentially the largest machine representable integer.) So, a common choice of $m = 2^{31} - 1$. Common choices for a are $a = 16807$ and $a = 630360016$.

Remark 3.11: Since $2^{31} - 1$ is close to a power of 2, division by $2^{31} - 1$ can be efficiently implemented through “simulated division”.

Remark 3.12: The most commonly used statistical tests are χ^2 tests (on the U_i ’s, the (U_i, U_{i+1}) ’s, etc.) and run tests (to test the high-dimensionally independence of the u_i ’s).

Remark 3.13: An LCG with $m = 2^{31} - 1$ has a period on the order of about 2 billion. This is not a very long random number stream, given that some applications demand far more random variates. In addition, some parallel computing implementations demand a large number of independent random number streams. For random number generators that address these issues, visit the website of Pierre L’Ecuyer at Universite de Montreal.

Remark 3.14: Despite the obvious non-randomness inherent in such algorithmically determined random number streams, carefully tested generators tend to work (very) well in practice.

3.6 Convergence of the Monte Carlo Method

We have proposed a class of algorithms, based on computer experimentation / simulation, for computing probabilities (or, more generally, expectations). Algorithms that depend on streams of random numbers are frequently called Monte Carlo algorithms. (Monte Carlo was a “code word” used World War II to describe sampling-based methods for neutron scattering computations associated with Manhattan project.) The first order of business in studying such a scientific computing algorithm is to establish convergence.

The generic problem we wish to address is:

Goal: Compute $\alpha = E[X]$, where X is a random variable that can be generated in finite time (on a computer)

Algorithm A3: Sample Mean

1	Generate n iid copies X_1, X_2, \dots, X_n of the random variable X
2	Estimate α via:
3	$\alpha_n = \frac{1}{n} (X_1 + \dots + X_n)$

We wish to show that α_n converges to α as $n \rightarrow \infty$. This convergence is implied by the “law of large numbers”.

Definition 3.2: Let $(Z_n : 1 \leq n \leq \infty)$ be a sequence of random variables. Then, Z_n converges in probability to Z_∞ as $n \rightarrow \infty$ (written $Z_n \xrightarrow{P} Z_\infty$ as $n \rightarrow \infty$) if for each $\epsilon > 0$,

$$P\{|Z_n - Z_\infty| > \epsilon\} \rightarrow 0$$

as $n \rightarrow \infty$.

3.6.1 Weak Law of Large Numbers (WLLN)

Let X_1, X_2, \dots be an iid sequence of random variables for which $E[X_1] < \infty$. Then,

$$\frac{1}{n} (X_1 + \dots + X_n) \xrightarrow{P} E[X_1]$$

as $n \rightarrow \infty$.

It follows from the WLLN that:

$$\alpha_n \xrightarrow{P} \alpha$$

as $n \rightarrow \infty$. Hence, the WLLN establishes that the Monte Carlo method converges.

Because of its importance, we will probe the WLLN under the (stronger) hypothesis that $\text{var}[X_1] < \infty$. The proof will illustrate some more widely applicable ideas. It depends on some (easy) probability inequalities.

Markov’s Inequality Let W be a non-negative random variable. Then,

$$P(W > w) \leq \frac{E[W]}{w}$$

Proof.

$$P(W > w) = E[I(W > w)]$$

since $W/w \geq 1$ on $\{W > w\}$

$$\begin{aligned} &\leq E\left[\frac{W}{w}\right] I(W > w) \\ &\leq \frac{E[W]}{w} \end{aligned}$$

□

Chebyshev’s Inequality Let Γ be a random variable for which $\text{var}[\Gamma] < \infty$. Then,

$$P\{|\Gamma - E[\Gamma]| > \epsilon\} \leq \frac{\text{var}(\Gamma)}{\epsilon^2}$$

Proof. Put $W = (\Gamma - E[\Gamma])^2$ and $w = \epsilon^2$ and use Markov’s Inequality.

□

We now prove the WLLN when $\text{var}[X_1] < \infty$.

Proof (of the WLLN). Put $\Gamma = X_1 + \dots + X_n$. Chebyshev's Inequality implies that:

$$P\{|\Gamma - E[\Gamma]| > n\epsilon\} \leq \frac{\text{var}[\Gamma]}{n^2\epsilon^2}$$

i.e

$$P\left\{\left|\frac{\Gamma}{n} - E[X_1]\right| > \epsilon\right\} \leq \frac{\text{var}[X_1]}{n\epsilon^2}$$

so that

$$P\{|n^{-1}(X_1 + \dots + X_n) - E[X_1]| > \epsilon\} \leq \frac{\text{var}[X_1]}{n\epsilon^2} \rightarrow 0$$

as $n \rightarrow \infty$. □

3.7 The Strong Law of Large Numbers

The Monte Carlo method is actually convergent in a stronger sense than that describe by the WLLN.

Definition 3.3: Let $(Z_n : 1 \leq n \leq \infty)$ be a sequence of random variables. Then, Z_n converges almost surely to Z_∞ as $n \rightarrow \infty$, (written $Z_n \xrightarrow{\text{a.s.}} Z_\infty$ as $n \rightarrow \infty$) if $P\{A\} = 1$, where A is the event defined by:

$$A = \{\omega : Z_n(\omega) \rightarrow Z_\infty(\omega) \text{ as } n \rightarrow \infty\}$$

Remark 3.15: There are a number of equivalent terminologies for almost sure convergence:

- convergence with probability one
- convergence almost everywhere
- convergence almost certainly

They all mean the same thing.

Strong Law of Large Numbers (SLLN)

Let X_1, X_2, \dots be an iid sequence of random variables for which $E[X_1] < \infty$. Then

$$n^{-1}(X_1 + \dots + X_n) \xrightarrow{\text{a.s.}} E[X_1]$$

as $n \rightarrow \infty$.

It follows from the SLLN that:

$$\alpha_n \xrightarrow{\text{a.s.}} \alpha$$

as $n \rightarrow \infty$. Hence, the SLLN establishes that the Monte Carlo method converges almost surely.

The Strong Law of Large Numbers implies the WLLN, and is a more sophisticated result. Note that

the event A described by the SLLN is “infinite dimensional”. The formulation and proof of the SLLN requires a mathematical foundation capable of computing probabilities of infinite-dimensional events. This is an example of a result for which a complete understanding requires the use of “measure theoretic probability”.

To obtain a sense of the issues involved, consider the special case in which the X_i 's are iid Bernoulli random variables with parameter $1/2$. The natural sample space here is

$$\Omega = \{0, 1\}^{\mathbb{Z}_+}$$

where $\mathbb{Z}_+ = \{0, 1, 2, 3, \dots\}$. In other words, Ω is the set of 0-1 valued sequences of the form:

$$\omega = (\omega_0, \omega_1, \dots)$$

where $\omega_i \in \{0, 1\}$. For $i \geq 0$ put:

$$X_i(\omega) = \omega_i$$

The event A is the infinite-dimensional event consisting of the sequences $\omega \in \Omega$ for which:

$$n^{-1} \sum_{i=0}^{\infty} X_i(\omega) = n^{-1} \sum_{i=0}^{n-1} \omega_i \rightarrow \frac{1}{2}$$

as $n \rightarrow \infty$.

One approach one might follow in computing $P\{A\}$ for such an infinite-dimensional event is to approximate it by corresponding finite-dimensional computations. Note that:

$$P\{X_0 = i_0, \dots, X_{n-1} = i_{n-1}\} = 2^{-n}$$

for $i_j \in \{0, 1\}$, so it is easy to compute the probability of any finite-dimensional event A_n involving X_0, \dots, X_{n-1} namely:

$$P\{(X_0, X_1, \dots, X_{n-1}) \in A_n\} = 2^{-n} |A_n|$$

where $|A_n|$ is the number of sequences (i_0, \dots, i_{n-1}) lying in A_n . This approach to computing probabilities fails when A is infinite-dimensional, because for every infinite-dimensional sequence (i_0, i_1, i_2, \dots) ,

$$P\{(X_0, X_1, X_2, \dots) = (i_0, i_1, \dots)\} = 0$$

In fact, it is not even a priori clear that the probability P can be extended from finite-dimensional sequences to infinite-dimensional sequences. Kulmugoruv proved that such infinite-dimensional extensions exist (in great generality).

Let $\Omega = \mathbb{R}^\infty$. For each $\omega = (\omega_0, \omega_1, \dots) \in \Omega$, let $X_i(\omega) = \omega_i$ for $i \geq 0$.

Definition 3.4: For $n \geq 0$, let P_n be a probability on \mathbb{R}^{n+1} . The sequence $(P_n : n \geq 1)$ is said to have the consistency property if

$$P_n\{(X_0, X_1, \dots, X_m) \in \cdot\} = P_m\{(X_0, X_1, \dots, X_m) \in \cdot\}$$

for $n \geq m$.

Theorem 3.1 ((Kulmugoruv's Extension Theorem)). *Let $(P_n : n \geq 1)$ be a sequence of probabilities having the consistency property. Then there exists a probability P on Ω for which:*

$$P\{(X_0, \dots, X_m) \in \cdot\} = P_m\{(X_0, \dots, X_m) \in \cdot\}$$

for $m \geq 0$. Furthermore, P is unique.

We can give a hint of one approach to constructing P in the Bernoulli setting described earlier. Let $\tilde{\Omega} = [0, 1]$ and let \tilde{P} be the uniform distribution on $\tilde{\Omega}$. For $\tilde{\omega} \in \tilde{\Omega}$, let $Y(\tilde{\omega}) = \tilde{\omega}$. Then,

$$\tilde{P}(Y \in B) = \int_B dy$$

For $\tilde{\omega} \in [0, 1]$, there exists a dyadic expansion:

$$\tilde{\omega} = \sum_{j=1}^{\infty} i_j(\tilde{\omega})2^{-j}$$

where $i_j(\tilde{\omega}) \in \{0, 1\}$. We can then construct the probability P as follows:

$$P\{(X_0, X_1, \dots) \in \cdot\} \triangleq \tilde{P}\{(i_0, i_1, \dots) \in \cdot\}$$

Hence, in this specific context, the extension can be identified with the uniform distribution on $[0, 1]$.

Reference:

Breiman, L. Probability. Addison-Wesley:1968. Chapters 1 to 3.

3.8 Rate of Convergence in the Monte Carlo Method

If one is to use the Monte Carlo method as a computational tool, a study of its rate of convergence is appropriate. The rate of convergence is described by the central limit theorem (CLT). The CLT is a limit theorem that uses the notion of “convergence in distribution”.

Definition 3.5: Let $(Z_n : 1 \leq n \leq \infty)$ be a sequence of random variables. Then Z_n converges in distribution to Z_∞ as $n \rightarrow \infty$ (written, $Z_n \Rightarrow Z_\infty$ as $n \rightarrow \infty$) if

$$P\{Z_n \leq z\} \rightarrow P\{Z_\infty \leq z\}$$

as $n \rightarrow \infty$ at each continuity point z of $P\{Z_\infty \leq \cdot\}$.

Remark 3.16: Convergence in distribution is equivalently known as “weak convergence”.

Remark 3.17: The caveat about continuity points is included, so that the definition covers cases like: $Z_n = 1 + n^{-1}$, $Z_\infty = 1$. ($Z_n, 1 \leq n \leq \infty$, is a deterministic sequence that clearly converges as a deterministic sequence). Then:

$$P\{Z_n \leq z\} = \begin{cases} 0 & z < 1 + n^{-1} \\ 1 & z \geq 1 + n^{-1} \end{cases}$$

$$P\{Z_\infty \leq z\} = \begin{cases} 0 & z < 1 \\ 1 & z \geq 1 \end{cases}$$

Note that $0 = P\{Z_n \leq 1\} \not\rightarrow P\{Z_\infty \leq 1\} = 1$, so $P\{Z_n \leq z\}$ does not converge to $P\{Z_\infty \leq z\}$ at the discontinuity point $z = 1$.

Central Limit Theorem

Let $(X_n : n \geq 1)$ be a sequence of iid random variables with $0 < \sigma^2 = \text{var}[X_1] < \infty$. Then,

$$\frac{X_1 + \dots + X_n - n \text{E}[X_1]}{\sqrt{n}\sigma} \Rightarrow N(0, 1)$$

as $n \rightarrow \infty$, where $N(0, 1)$ is a normal random variable with mean zero and unit variance.

Remark 3.18: The CLT asserts that for each $x \in \mathbb{R}$,

$$P \left\{ \frac{X_1 + \dots + X_n - n \text{E}[X_1]}{\sqrt{n}\sigma} \leq x \right\} \rightarrow \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt$$

Remark 3.19: An equivalent formulation of the CLT is the following:

$$\frac{n^{\frac{1}{2}}}{\sigma} \left(n^{-1} \sum_{i=1}^n X_i - \text{E}[X_1] \right) \Rightarrow N(0, 1)$$

as $n \rightarrow \infty$.

Returning to the analysis of the Monte Carlo method, the CLT asserts that:

$$n^{\frac{1}{2}} (\alpha_n - \alpha) \Rightarrow \sigma N(0, 1)$$

as $n \rightarrow \infty$. This mathematically rigorous limit theorem can be re-formulated as the following (formal) approximation:

$$\alpha_n \stackrel{D}{\approx} \alpha + \frac{\sigma}{\sqrt{n}} N(0, 1)$$

where $\stackrel{D}{\approx}$ denotes “has approximately the same distribution as”. The above approximation yields the following insights:

- the rate of convergence is of order $n^{-\frac{1}{2}}$ (so adding one significant figure of accuracy requires increasing n by a factor of 100). So, the Monte Carlo method is not suitable for computations in which high accuracy (large numbers of significant figures) is required.
- the error is normally distributed for large n
- the complexity of the computation depends on the problem instance (at least asymptotically) through only one feature of the distribution of X_1 , namely its variance.

Given the slow rate of convergence we will wish to attach “error bars” to any Monte Carlo computation.

3.9 Characteristic Functions

A key tool in the analysis of sums of independent random variables (and the most common means of proving the CLT) is the concept of characteristic functions.

Definition 3.6: Given an \mathbb{R}^d -valued random vector \vec{Z} , the characteristic function of \vec{Z} is given by

$$c(\theta) = \text{E} \left[e^{i\theta \vec{Z}} \right]$$

for $\theta \in \mathbb{R}^d$.

Remark 3.20: The quantity $\theta \vec{Z}$ is the inner product of $\theta = (\theta_1, \dots, \theta_d)$ with $\vec{Z} = (Z_1, \dots, Z_d)^T$, namely

$$\theta \vec{Z} \triangleq \sum_{i=1}^d \theta_i Z_i$$

Remark 3.21: Recall that

$$e^{i\omega} = \cos \omega + i \sin \omega$$

for $\omega \in \mathbb{R}$.

Remark 3.22: When \vec{Z} has a probability density function f , the characteristic function is essentially its Fourier transform, namely:

$$c(\theta) = \int_{\mathbb{R}^d} e^{i\theta \vec{Z}} f(\vec{Z}) d\vec{Z}$$

Result 3.2: There is a 1-1 correspondence between characteristic functions and probability distributions of random vectors. In particular, if $\vec{Z}_1 \stackrel{D}{=} \vec{Z}_2$, then (obviously) $c_{\vec{Z}_1}(\cdot) = c_{\vec{Z}_2}(\cdot)$. Also, if $c_{\vec{Z}_1}(\cdot) = c_{\vec{Z}_2}(\cdot)$, then $\vec{Z}_1 \stackrel{D}{=} \vec{Z}_2$. (This latter result depends on the Fourier inversion theorem.)

Theorem 3.2 (Fourier Inversion Theorem (for real-valued random variables)). *Suppose that $c(\cdot)$ is the characteristic function of a distribution F , so that:*

$$c(\theta) = \int_{\mathbb{R}} e^{i\theta z} F(dz)$$

Then, for $z_1 < z_2$,

$$F(z_2) - F(z_1) + \frac{\Delta F(z_1) - \Delta F(z_2)}{2} = \lim_{T \rightarrow \infty} \frac{1}{2\pi} \int_{-T}^T \frac{e^{-i\theta z_1} - e^{-i\theta z_2}}{i\theta} c(\theta) d\theta$$

where $\Delta F(z) = F(z) - F(z-)$ for $z \in \mathbb{R}$. If $c(\cdot)$ is integrable, then F has a continuous density f given by:

$$f(z) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-i\theta z} c(\theta) d\theta$$

Remark 3.23: $F(z-) = \lim_{y \rightarrow z} F(y)$, so $\Delta F(z)$ is the size of the jump in F at z (and can be interpreted as the probability that the associated random variable takes on value z).

A key property of characteristic functions is that the characteristic function of a sum of independent random variables can easily be computed. In particular, put

$$c_j(\theta) = \mathbb{E} \left[e^{i\theta \vec{Z}_j} \right]$$

If $\vec{Z}_1, \dots, \vec{Z}_n$ are n independent random variables, then

$$\mathbb{E} \left[e^{i\theta(\vec{Z}_1 + \dots + \vec{Z}_n)} \right] = \mathbb{E} \left[\prod_{j=1}^n e^{i\theta \vec{Z}_j} \right]$$

by independence we have

$$\begin{aligned} &= \prod_{j=1}^n \mathbb{E} \left[e^{i\theta \vec{Z}_j} \right] \\ &= \prod_{j=1}^n c_j(\theta) \end{aligned}$$

This suggests the following approach to computing the exact distribution of $\vec{Z}_1 + \dots + \vec{Z}_n$:

1. Compute $c_j(\cdot)$ for $1 \leq j \leq n$.
2. Compute the product

$$\prod_{j=1}^n c_j(\theta)$$

3. Invert the product to compute the distribution of $\vec{Z}_1 + \dots + \vec{Z}_n$.

Remark 3.24: Steps 1 and 3 can be implemented numerically (via the Fast Fourier Transform)

Example 3.6: Suppose that $Z_i \stackrel{D}{=} N(\mu_i, \sigma_i^2)$ for $1 \leq i \leq n$ and that Z_1, \dots, Z_n are independent random variables.

- 1.

$$\mathbb{E} [e^{i\theta Z_j}] = e^{i\theta\mu_j - \frac{1}{2}\theta^2\sigma_j^2}$$

- 2.

$$\begin{aligned} \prod_{j=1}^n c_j(\theta) &= \prod_{j=1}^n e^{i\theta\mu_j - \frac{1}{2}\theta^2\sigma_j^2} \\ &= e^{i\theta \sum_{j=1}^n \mu_j - \frac{1}{2}\theta^2 \sum_{j=1}^n \sigma_j^2} \\ c(\theta) &= e^{i\theta\mu - \frac{1}{2}\theta^2\sigma^2} \end{aligned}$$

where

$$\mu = \mu_1 + \dots + \mu_n \quad \text{and} \quad \sigma^2 = \sigma_1^2 + \dots + \sigma_n^2$$

- 3.

$$c(\theta) = e^{i\theta\mu - \frac{1}{2}\theta^2\sigma^2}$$

is the characteristic function of a $N(\mu, \sigma^2)$ random variable, so

$$Z_1 + \dots + Z_n \stackrel{D}{=} N(\mu, \sigma^2)$$

Characteristic functions have other important properties. For example, one can compute the moments of a random variable through successive differentiation of its characteristic function. To see the relationship, note that

$$\frac{d^k}{d\theta^k} \mathbb{E} [e^{i\theta Z}] = \mathbb{E} \left[\frac{d^k}{d\theta^k} e^{i\theta Z} \right] = \mathbb{E} [i^k Z^k e^{i\theta Z}] = i^k \mathbb{E} [Z^k e^{i\theta Z}]$$

(assuming that the interchange of derivative and expectation / integral is valid). Setting $\theta = 0$ we conclude that:

$$\left. \frac{d^k}{d\theta^k} c(\theta) \right|_{\theta=0} = i^k \mathbb{E} [Z^k]$$

A rigorously stated version of this result follows:

Result 3.3: Suppose that Z is a real-valued random variable and let $c(\cdot)$ be its characteristic function. If $\mathbb{E} [|Z|^k] < \infty$, then $c(\cdot)$ is k -times continuously differentiable and

$$c^{(k)}(0) = i^k \mathbb{E} [Z^k]$$

Conversely, if $c(\cdot)$ has a finite derivative of even order k at $\theta = 0$, then $\mathbb{E} [|Z|^k] < \infty$ and $\mathbb{E} [Z^k] = -i^k c^{(k)}(0)$.

Characteristic functions are also a useful tool in establishing convergence in distribution.

Result 3.4: Let $(Z_n : 1 \leq n < \infty)$ be a sequence of real-valued random variables and put $c_n(\theta) = \mathbb{E} [e^{i\theta Z_n}]$.

1. If $Z_n \Rightarrow Z_\infty$ as $n \rightarrow \infty$, then $c_n(\theta) \rightarrow \mathbb{E} [e^{i\theta Z_\infty}]$ and $n \rightarrow \infty$.
2. Suppose that for each $\theta \in \mathbb{R}$, there exists a limit $\gamma(\theta)$ for which:

$$c_n(\theta) \rightarrow \gamma(\theta)$$

as $n \rightarrow \infty$, where $\gamma(\cdot)$ is continuous at $\theta = 0$. Then $\gamma(\cdot)$ is the characteristic function of a random variable Z_∞ and $Z_n \Rightarrow Z_\infty$ as $n \rightarrow \infty$.

Reference

Chung, K.L. A Course in Probability Theory. Academic Press, New York: 1974. Chapter 6.

3.10 Proof of the Central Limit Theorem

Given the importance of the CLT, we give here a brief indication of its proof.

Suppose that X_1, X_2, \dots is a sequence of iid random variables with $0 < \sigma^2 = \text{var}[X_1] < \infty$. Put:

$$\tilde{X}_i = \frac{X_i - \mathbb{E}[X_1]}{\sigma}$$

and note that $\mathbb{E}[\tilde{X}_i] = 0$ and $\text{var}[\tilde{X}_i] = \mathbb{E}[\tilde{X}_i^2] = 1$. Hence,

$$\mathbb{E} [e^{i\theta \tilde{X}_1}] = 1 - \frac{\theta^2}{2} + O(\theta^2)$$

as $\theta \rightarrow 0$. Then:

$$\begin{aligned} \mathbb{E} \left[e^{i\theta \frac{X_1 + \dots + X_n - n \mathbb{E}[X_1]}{\sqrt{n}\sigma}} \right] &= \mathbb{E} \left[e^{i\theta \sum_{j=1}^n \frac{\tilde{X}_j}{\sqrt{n}}} \right] \\ &= \mathbb{E} \left[\prod_{j=1}^n e^{i \frac{\theta}{\sqrt{n}} \tilde{X}_j} \right] \end{aligned}$$

by independence we find:

$$= \prod_{j=1}^n \mathbb{E} \left[e^{i \frac{\theta}{\sqrt{n}} \tilde{X}_j} \right]$$

and by identical distribution:

$$\begin{aligned} &= \left(\mathbb{E} \left[e^{i \frac{\theta}{\sqrt{n}} \tilde{X}_1} \right] \right)^n \\ &= \left(1 - \frac{\theta^2}{2n} + O(n^{-1}) \right)^n \rightarrow e^{-\frac{\theta^2}{2}} \end{aligned}$$

But $e^{-\frac{\theta^2}{2}}$ is the characteristic function of a $N(0, 1)$ random variable, so:

$$\frac{X_1 + \dots + X_n - n \mathbb{E}[X_1]}{\sqrt{n}\sigma} \Rightarrow N(0, 1)$$

as $n \rightarrow \infty$. (See section 9)

3.11 More on the Monte Carlo Method

As discussed earlier, the Monte Carlo method exhibits a rate of convergence of order $n^{-\frac{1}{2}}$ in the number n of experiments undertaken. Given the slow rate of convergence, one may question widespread usage.

But the Monte Carlo method has important advantages relative to competing numerical methods:

1. Conceptual simplicity: Conceptually speaking, it can easily be applied to stochastic modeling problems of almost arbitrary complexity.
2. Versatility: The method applies to computation in virtually all stochastic modeling contexts.
3. Flexibility: Code can be easily modified to reflect changes in the model (e.g. changing the service time requirement distribution in the single-server queue).
4. Animation: Simulations are well-suited to animation of the stochastic dynamics of the system under consideration (leading to greater intuition / insight than numbers alone).

In addition, we will now argue that the rate of convergence of the Monte Carlo method is competitive in many numerical settings.

Note that virtually all Monte Carlo computations involve what are essentially numerical integrations (typically in a high-dimensional space). Consider, for example, the problem of computing $E[Q(t)]$ where $Q(t)$ is the number-in-system at time t in the single-server queue. Note that:

$$E[Q(t)] = \sum_{n=0}^{\infty} E[Q(t)I(N(t) = n)]$$

where $N(t)$ is the number of arrivals to the system in the interval $[0, t]$. If the system starts empty, then $Q(t) \leq N(t)$, so

$$E[Q(t)] = \sum_{n=0}^{\infty} \sum_{j=0}^n jP\{Q(t) = j, N(t) = n\}$$

But $\{N(t) = n, Q(t) = j\}$ depends on the first $n + 1$ inter-arrival times, $\chi_1, \chi_2, \dots, \chi_{n+1}$ and the first $n + 1$ service time requirements V_0, V_1, \dots, V_n . So,

$$\{N(t) = n, Q(t) = j\} = \{(\chi_1, \dots, \chi_{n+1}, V_0, \dots, V_n) \in B_{nj}\}$$

where B_{nj} is a subset of \mathbb{R}_+^{2n+2} . For example

$$\{N(t) = n, Q(t) = n\} = \{\chi_1 + \dots + \chi_n \leq t < \chi_1 + \dots + \chi_{n+1}, \chi_1 + V_0 > t\}$$

so

$$P\{N(t) = n, Q(t) = n\} = \int_{B_{nn}} \prod_{i=1}^{n+1} f_1(x_i) \prod_{i=1}^{n+1} f_2(v_{i-1}) dx_1 \dots dx_{n+1} dv_0 \dots dv_n$$

where $f_1(\cdot)$ is the common density of the χ_i 's, $f_2(\cdot)$ is the common density of the V_i 's and

$$B_{nn} = \{(\chi_1, \dots, \chi_{n+1}, V_0, \dots, V_n) = (x_1, \dots, x_{n+1}, v_0, \dots, v_n) : \\ x_1 + \dots + x_n \leq t < x_1 + \dots + x_{n+1}, x_1 + v_0 > t\}$$

Consequently, $E[Q(t)]$ can be expressed as:

$$E[Q(t)] = \sum_{n=0}^{\infty} \sum_{j=0}^n \int_{B_{nj}} \prod_{i=1}^{n+1} f_1(x_i) \prod_{i=1}^n f_2(v_{i-1}) dx_1 \dots dx_{n+1} dv_0 \dots dv_n$$

In other words, $E[Q(t)]$ can, in principal, be expressed in terms of a sum of high-dimensional integrals.

This suggests the possibility of computing $E[Q(t)]$ by taking advantage of non-sampling based numerical integration methods. Unfortunately, non-sampling based methods typically exhibit the “curse of dimensionality” (so that the complexity of the algorithm increases rapidly as a function of dimension). To appreciate the difficulties, consider the simplest possible scheme for numerical integration of a function $h(\cdot)$ over the d -dimensional unit hypercube:

$$I = \int_{[0,1]^d} h(\vec{x}) d\vec{x} = \int_0^1 \cdots \int_0^1 h(x_1, \dots, x_d) dx_1 \dots dx_d$$

The d -dimensional analog of the well known “rectangular integration rule” in dimension one involves splitting $[0, 1]^d$ into n sub hypercubes of equal volume n^{-1} . Call the hypercubes H_1, \dots, H_n . Choose a representative point $\vec{x}_i \in H_i$ (say, the point closest to the origin) and approximate I via:

$$I_n = \sum_{i=1}^n h(\vec{x}_i) n^{-1}$$

Then

$$I - I_n = \sum_{i=1}^n \int_{H_i} h(\vec{x}) d\vec{x} - \sum_{i=1}^n h(\vec{x}_i) n^{-1} = \sum_{i=1}^n \int_{H_i} [h(\vec{x}) - h(\vec{x}_i)] d\vec{x}$$

If $h(\cdot)$ is smooth,

$$h(\vec{x}) - h(\vec{x}_i) = \nabla h(\vec{x}_i)(\vec{x} - \vec{x}_i) + o(\|\vec{x} - \vec{x}_i\|)$$

Assuming that $\nabla h(\cdot)$ is bounded over $[0, 1]^d$,

$$|h(\vec{x}) - h(\vec{x}_i)| = C \|\vec{x} - \vec{x}_i\|$$

But for $\vec{x} \in H_i$, the average order of magnitude of $\|\vec{x} - \vec{x}_i\|$ is of order $n^{-\frac{1}{d}}$ (since the sides of a hypercube of volume n^{-1} are equal to $n^{-\frac{1}{d}}$). Hence, I_n converges to I at rate $n^{-\frac{1}{d}}$. This rate of convergence is exceptionally slow when d is large (and slower than the Monte Carlo rate when $d \geq 3$). This is an example of the “curse of dimensionality”.

Definition 3.7: An n -point integration rule over $[0, 1]^d$ is a set of points $\vec{x}_1, \dots, \vec{x}_n$ in $[0, 1]^d$ and weights w_1, \dots, w_n . The corresponding approximation to I is:

$$I_n = \sum_{j=1}^n w_j h(\vec{x}_j)$$

Definition 3.8: C_M^r is the class of functions h on $[0, 1]^d$, all of whose r^{th} (mixed) partial derivatives exist, are continuous on $[0, 1]^d$, and are $\leq M$ in absolute value there.

Result 3.5: There is a constant $k = k(M, r)$ such that for any n -point integration rule, there exists a function $\tilde{h} \in C_M^r$ for which:

$$\left| \sum_{j=1}^n w_j \tilde{h}(\vec{x}_j) - \int_{[0,1]^d} \tilde{h}(\vec{x}) d\vec{x} \right| > kn^{-\frac{r}{d}}$$

This result is a more careful statement of the “curse of dimensionality”.

On the other hand, the Monte Carlo method exhibits a rate of convergence that is insensitive to the size of the dimension d in which the sampling is occurring. Consider, for example, the problem of computing $\alpha = P\{A\}$ via the Monte Carlo method. The method involves generating n iid random variables I_1, I_2, \dots, I_n , where I_j is one or zero depending on whether or not A occurred on the j^{th} simulation. The estimator is:

$$\alpha_n = n^{-1} \sum_{j=1}^n I_j$$

Chebyshev's inequality implies that:

$$P\{|\alpha_n - \alpha| > \epsilon n^{-\frac{1}{2}}\} \leq \frac{\alpha(1-\alpha)}{\epsilon^2} \leq \frac{1}{4\epsilon^2}$$

regardless of the dimension d associated with the event A . Hence, for this class of "problem instances", the $n^{-\frac{1}{2}}$ rate of convergence in the number n of simulations conducted holds universally across all dimensions d . This suggests that the Monte Carlo method is a superior method for computing high-dimensional probabilities (and, more generally, expectations).

Remark 3.25: When the dimension is small (say $d \leq 3$) and the integrand h is smooth, one typically should use quadrature methods to numerically integrate the function. For example, for functions h that are four times continuously differentiable over $[0, 1]$, Simpson's rule yields a convergence rate of order n^{-4} (which is much faster than the Monte Carlo rate). For dimensions of moderate size ($4 \leq d \leq 50$), quasi-random integration rules are commonly recommended (see the references below for details). For high dimensional integration ($d > 50$), the standard recommendation is the use of Monte Carlo methods. So, the Monte Carlo method is a fundamental numerical tool in the scientific computing environment.

References

Davis, P.J and P. Rabinowitz. Methods of Numerical Integration. Academic Press, New York (1984). Chapter 5.

Evens, M. and T Swartz. Approximating Integrals via Monte Carlo and Deterministic Methods. Oxford University Press (2000). Chapter 5.

3.12 Error Bars for the Monte Carlo Method

Given the slow rate of convergence associated with the Monte Carlo method, it is clearly of value to assess the error associated with a given Monte Carlo computation

The CLT offers one means of accomplishing this. Assume that our goal is to compute $\alpha = E[X]$, where $0 < \text{var}(X) = \sigma^2 < \infty$. Suppose we run n independent computer experiments, thereby generating iid copies X_1, X_2, \dots, X_n of the random variables X . Our estimator is:

$$\alpha_n = n^{-1}(X_1 + \dots + X_n)$$

The CLT asserts that for any $x \geq 0$,

$$P\left\{-z \leq \frac{n^{\frac{1}{2}}}{\sigma}(\alpha_n - \alpha) \leq z\right\} \rightarrow P\{-z \leq N(0, 1) \leq z\}$$

as $n \rightarrow \infty$. Suppose we choose z so that $P\{-z \leq N(0, 1) \leq z\} = 1 - \delta$. (Standard values of δ are $\delta = 0.1, 0.05$, and 0.01) Such a z -value can be found in statistical tables for the normal distribution (e.g. for $\delta = 0.1, z = 1.64$ where as $z = 1.96$ for $\delta = 0.05$). The event

$$\left\{ -z \leq \frac{n^{\frac{1}{2}}}{\sigma}(\alpha_n - \alpha) \leq z \right\}$$

is identical to

$$\left\{ \alpha \in \left[\alpha_n - \frac{\sigma z}{\sqrt{n}}, \alpha_n + \frac{\sigma z}{\sqrt{n}} \right] \right\}$$

Hence we may conclude that

$$P \left\{ \alpha \in \left[\alpha_n - \frac{\sigma z}{\sqrt{n}}, \alpha_n + \frac{\sigma z}{\sqrt{n}} \right] \right\} \Rightarrow 1 - \delta$$

as $n \rightarrow \infty$. In other words, the random interval

$$\left[\alpha_n - \frac{\sigma z}{\sqrt{n}}, \alpha_n + \frac{\sigma z}{\sqrt{n}} \right]$$

contains the parameter α that we are computing with (approximately) a probability of $1 - \delta$. Such a random interval is called an (approximate) $100(1 - \delta)\%$ confidence interval for α . Hence, if $\delta = 0.1$,

$$\left[\alpha_n - (1.64) \frac{\sigma}{\sqrt{n}}, \alpha_n + (1.64) \frac{\sigma}{\sqrt{n}} \right]$$

is an (approximate) 90% confidence interval for α . Informally, our estimate for α is

$$\alpha_n \pm (1.64) \frac{\sigma}{\sqrt{n}}$$

Thus, the “error bars” associated with Monte Carlo computations have their basis in the statistical notion of confidence intervals.

While the above methodology is appealing, it actually can not be operationally implemented because σ^2 is almost never known a priori. Furthermore, σ can be estimated from the simulated data via

$$s_n = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \alpha_n)^2}$$

We will argue later that $s_n \Rightarrow \sigma$ as $n \rightarrow \infty$, thereby suggesting that

$$P \left\{ \alpha \in \left[\alpha_n - z \frac{s_n}{\sqrt{n}}, \alpha_n + z \frac{s_n}{\sqrt{n}} \right] \right\} \rightarrow 1 - \delta$$

as $n \rightarrow \infty$. This justifies the confidence interval algorithm below.

Algorithm A4: Computation of an approximate $100(1 - \delta)\%$ confidence interval for $\alpha = E[X]$

- | | |
|---|---|
| 1 | Select δ and choose n (the number of computer runs) |
| 2 | Generate n iid copies X_1, \dots, X_n of the random variable X . |
| 3 | Compute: |
| 4 | $\alpha_n = n^{-1}(X_1 + \dots + X_n)$ |
| 5 | $s_n = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \alpha_n)^2}$ |
| 6 | Find z so that $P\{-z \leq N(0, 1) \leq z\} = 1 - \delta$. |
| 7 | Then $\left[\alpha_n - z \frac{s_n}{\sqrt{n}}, \alpha_n + z \frac{s_n}{\sqrt{n}} \right]$ is an $100(1 - \delta)\%$ confidence interval for α . |

How is n typically selected in practice? One first simulates a small number n_0 of “trial runs”, thereby yielding X_1, \dots, X_{n_0} . One then computes:

$$\tilde{\alpha}_{n_0} = \frac{1}{n_0}(X_0 + \dots + X_{n_0}) \quad \text{and} \quad \tilde{s}_{n_0} = \sqrt{\frac{1}{n_0 - 1} \sum_{i=1}^{n_0} (X_i - \tilde{\alpha}_{n_0})^2}$$

Assuming that one desires a final confidence interval with half-width ϵ , one should choose n so that

$$n \approx \frac{z^2 \tilde{s}_{n_0}^2}{\epsilon^2}$$

Having now compute n , one then executes Algorithm A4 with this value of n . Usually, the trial runs are not used in producing the final confidence interval for α . (Re-use of the trial runs can adversely affect the statistical validity of the final confidence interval.)

Remark 3.26: If we desire a final confidence interval with half-width $\epsilon|\alpha|$ (so the relative error is ϵ), we should choose

$$n \approx \frac{z^2 \tilde{s}_{n_0}^2}{\epsilon^2 \tilde{\alpha}_{n_0}^2}$$

Remark 3.27: Why does the formula for s_n contain $n - 1$ instead of n (in the denominator)? One explanation is that the sample variance should clearly be undefined when one has a sample of only one observation. This occurs with $n - 1$ in the formula (but not with n). Secondly, the sample variance s_n^2 is itself a random variable. With $n - 1$ appearing in the formula

$$\text{E} [s_n^2] = \sigma^2$$

where as $\text{E} [s_n^2] < \sigma^2$ with n appearing in the formula rather than $n - 1$. In other words, the expectation of this sample variance then equals the “population variance” σ^2 , so that s_n^2 is an “unbiased estimator” of σ^2 .

We now justify the asymptotic validity of Algorithm A4. This serves to illustrate various convergence concepts for random variables. Note first that:

$$s_n^2 = \frac{1}{n-1} \sum_{i=1}^n X_i^2 - \left(\frac{n}{n-1} \right) \alpha_n^2$$

Because $\text{E} [X^2] < \infty$, the strong law of large numbers ensures that:

$$\alpha_n \xrightarrow{\text{a.s.}} \alpha$$

and

$$\frac{1}{n} \sum_{i=1}^n X_i^2 \xrightarrow{\text{a.s.}} \text{E} [X_1^2]$$

as $n \rightarrow \infty$.

Proposition 3.1: Suppose that $Z_n \xrightarrow{\text{a.s.}} Z_\infty$ as $n \rightarrow \infty$ and $W_n \xrightarrow{\text{a.s.}} W_\infty$ as $n \rightarrow \infty$. If $h : \mathbb{R}^2 \rightarrow \mathbb{R}$ is continuous, then $h(Z_n, W_n) \xrightarrow{\text{a.s.}} h(Z_\infty, W_\infty)$ as $n \rightarrow \infty$.

Proof. Let

$$A = \{\omega : Z_n(\omega) \rightarrow Z_\infty(\omega) \text{ as } n \rightarrow \infty\}$$

$$B = \{\omega : W_n(\omega) \rightarrow W_\infty(\omega) \text{ as } n \rightarrow \infty\}$$

$$C = \{\omega : h(Z_n(\omega), W_n(\omega)) \rightarrow h(Z_\infty(\omega), W_\infty(\omega)) \text{ as } n \rightarrow \infty\}$$

By assumption, $P\{A\} = P\{B\} = 1$. By standard real variable arguments for deterministic sequences, any $\omega \in A \cap B$ also lies in C . In other words, $C \supseteq A \cap B$. But

$$P\{A \cap B\} = P\{A\} + P\{B\} - P\{A \cup B\} = 1 + 1 - P\{A \cup B\} \geq 1 + 1 - 1 = 1$$

It follows that $P\{C\} = 1$ □

The proposition implies that $\alpha_n^2 \xrightarrow{\text{a.s.}} \alpha^2$ and $n \rightarrow \infty$. Hence, $(n/(n-1))\alpha_n^2 \xrightarrow{\text{a.s.}} \alpha^2$ as $n \rightarrow \infty$ (applying the proposition again). Also, $(1/(n-1))\sum_{i=1}^n X_i^2 = (n/(n-1))(n^{-1}\sum_{i=1}^n X_i^2) \xrightarrow{\text{a.s.}} E[X_1^2]$ (applying the proposition a third time). Finally

$$s_n^2 = \left(\frac{n}{n-1}\right) \left(\frac{1}{n} \sum_{i=1}^n X_i^2\right) - \left(\frac{n}{n-1}\right) \alpha_n^2 \xrightarrow{\text{a.s.}} E[X_1^2] - (E[X_1])^2 = \sigma^2$$

(applying the proposition a fourth time).

Exercise 3.1:

1. Prove that if $Z_n \xrightarrow{\text{a.s.}} Z_\infty$ as $n \rightarrow \infty$, then $Z_n \xrightarrow{P} Z_\infty$ as $n \rightarrow \infty$.

2. Let $a \in \mathbb{R}$ be deterministic. Prove that $Z_n \xrightarrow{P} a$ as $n \rightarrow \infty$ if and only if $Z_n \Rightarrow a$ as $n \rightarrow \infty$.

(Hint for 1): Note that $\{|Z_n - Z_\infty| \leq \epsilon\} \supseteq \{|Z_m - Z_\infty| \leq \epsilon \text{ for } m \geq n\}$.

As a consequence, $s_n^2 \Rightarrow \sigma^2$ as $n \rightarrow \infty$.

Result 3.6: Let $(Z_n : 1 \leq n \leq \infty)$ be a sequence of real-valued random variables. Then, $Z_n \Rightarrow Z_\infty$ as $n \rightarrow \infty$ if and only if

$$E[f(Z_n)] \rightarrow E[f(Z_\infty)]$$

for each bounded continuous $f : \mathbb{R} \rightarrow \mathbb{R}$.

Definition 3.9: Let $(Z_n : 1 \leq n \leq \infty)$ be a sequence of \mathbb{R}^d -valued random vectors. We say that $Z_n \Rightarrow Z_\infty$ as $n \rightarrow \infty$ (i.e. “ Z_n converges to Z_∞ in distribution”) if

$$E[f(Z_n)] \rightarrow E[f(Z_\infty)]$$

as $n \rightarrow \infty$ for each bounded continuous $f : \mathbb{R}^d \rightarrow \mathbb{R}$.

Remark 3.28: Convergence in distribution is equivalently called “weak convergence”.

Remark 3.29: Weak convergence extends to (much) more abstract (and general) settings. Let $(Z_n : 1 \leq n \leq \infty)$ be a sequence of S -valued random variables, where S is a complete separable metric space (e.g. $S = C[0, 1]$, the space of continuous function on $[0, 1]$ equipped with the “sup norm” metric, is such a space). We say that $Z_n \Rightarrow Z_\infty$ as $n \rightarrow \infty$ (i.e. Z_n converges weakly to Z_∞) if

$$E[f(Z_n)] \rightarrow E[f(Z_\infty)]$$

as $n \rightarrow \infty$ for each bounded continuous $f : S \rightarrow \mathbb{R}$.

Result 3.7: Let $((Z_n, W_n) : n \geq 1)$ be a sequence of \mathbb{R}^2 -valued random vectors. Suppose

$$Z_n \Rightarrow Z_\infty$$

as $n \rightarrow \infty$ and

$$W_n \Rightarrow a$$

as $n \rightarrow \infty$, where a is deterministic. Then $(Z_n, W_n) \Rightarrow (Z_\infty, a)$ as $n \rightarrow \infty$.

Remark 3.30: The fact that a is deterministic is important. In general, weak convergence of the marginal distribution (i.e. $Z_n \Rightarrow Z_\infty$ and $W_n \Rightarrow W_\infty$) does not imply weak convergence of the joint distribution (i.e. $(Z_n, W_n) \Rightarrow (Z_\infty, W_\infty)$).

The above result implies that

$$\left(\frac{n^{\frac{1}{2}}}{\sigma} (\alpha_n - \alpha), \frac{s_n}{\sigma} \right) \Rightarrow (N(0, 1), 1)$$

as $n \rightarrow \infty$.

Result 3.8: Suppose that $(Z_n : 1 \leq n \leq \infty)$ is a sequence of \mathbb{R}^d -valued random variables satisfying $Z_n \Rightarrow Z_\infty$ as $n \rightarrow \infty$. If $h : \mathbb{R}^d \rightarrow \mathbb{R}^l$ is continuous, then $h(Z_n) \Rightarrow h(Z_\infty)$ as $n \rightarrow \infty$.

Put $h(x, y) = x/y$. Then

$$h \left(\frac{n^{\frac{1}{2}}}{\sigma} (\alpha_n - \alpha), s_n \right) \Rightarrow h(N(0, 1), 1)$$

i.e.

$$\frac{n^{\frac{1}{2}}}{s_n} (\alpha_n - \alpha) \Rightarrow N(0, 1)$$

as $n \rightarrow \infty$.

Remark 3.31: The above result is known as the “continuous mapping principle”.

Proof of the Continuous Mapping Principle

We need to show that $E[f(h(Z_n))] \rightarrow E[f(h(Z_\infty))]$ for all bounded and continuous f . But $f \circ h$ is bounded and continuous, so this is implied by the definition of weak convergence of Z_n to Z_∞ .

Remark 3.32: Note that $h(x, y) = x/y$ is not continuous everywhere.

Result 3.9: Suppose that $(Z_n : 1 \leq n \leq \infty)$ is a sequence of \mathbb{R}^d -valued random variables satisfying $Z_n \Rightarrow Z_\infty$ as $n \rightarrow \infty$. Let $C_h = \{x : x \text{ is a point at which } h \text{ is continuous}\}$ for some mapping $h : \mathbb{R}^d \rightarrow \mathbb{R}^l$. If $P\{Z_\infty \in C_h\} = 1$, then $h(Z_n) \Rightarrow h(Z_\infty)$.

Since $\{(x, 1) : x \in \mathbb{R}\} \subseteq C_h$ for $h(x, y) = x/y$, the “extended” version of the continuous mapping principle guarantees that

$$\frac{n^{\frac{1}{2}}}{s_n} (\alpha_n - \alpha) \Rightarrow N(0, 1)$$

as $n \rightarrow \infty$. It follows that if z is selected so that $P\{-x \leq N(0, 1) \leq z\} = 1 - \delta$, then

$$P \left\{ \alpha \in \left[\alpha_n - z \frac{s_n}{\sqrt{n}}, \alpha_n + z \frac{s_n}{\sqrt{n}} \right] \right\} \rightarrow 1 - \delta$$

as $n \rightarrow \infty$, justifying the use of

$$\left[\alpha_n - z \frac{s_n}{\sqrt{n}}, \alpha_n + z \frac{s_n}{\sqrt{n}} \right]$$

as an (approximate) $100(1 - \delta)\%$ confidence interval for α .

Reference:

Billingsley, P. Weak Convergence of Probability Measures. John Wiley, New York: 1968.

3.13 The Bootstrap

We describe here an alternative method to computing a confidence interval for α that is representative of a very important class of methods with broad statistical applicability.

The key to producing a confidence interval for α , based on the estimator α_n , is the computation of z_1 and z_2 having the property that:

$$P\{z_1 \leq \alpha_n = \alpha \leq z_2\} = 1 - \delta$$

Given such z_1, z_2 an (exact) $100(1 - \delta)\%$ confidence interval for α can be obtained as:

$$[\alpha_n - z_2, \alpha_n - z_1]$$

So, the key to providing a confidence interval for α is the computation of the distribution of the random variable $\alpha_n - \alpha$.

Note that the distribution of $\alpha_n - \alpha$ depends on the distribution function $F(\cdot)$ underlying the X_i 's, as does α . To denote the dependence of $P\{\alpha_n - \alpha \leq \cdot\}$ and α on F , we write it as $P_F\{\alpha_n - \alpha \leq \cdot\}$.

Suppose that we knew the exact distribution F in closed form, as well as its mean α_F . One way to compute $P_F\{\alpha_n - \alpha_F \leq x\}$ would be to draw an iid sample X_{11}, \dots, X_{1n} from the distribution F (by simulating the X_{ij} 's) and to compute:

$$\alpha_n(1) = n^{-1} \sum_{j=1}^n X_{ij}$$

If we repeat this process m independent times, we obtain $\alpha_n(1), \dots, \alpha_n(m)$. The Law of Large Number ensures that

$$m^{-1} \sum_{i=1}^m I(\alpha_n(i) - \alpha_F \leq x) \rightarrow P_F\{\alpha_n - \alpha_F \leq x\}$$

as $n \rightarrow \infty$. The problem with this Monte Carlo based approach to computing $P_F(\alpha_n - \alpha_F \leq \cdot)$ is that it requires complete knowledge of F (as well as α_F). Given that our original purpose here is to produce a confidence interval for the unknown α_F , this is clearly unrealistic.

The idea underlying the “bootstrap” is to use the original n computer experiments, X_1, \dots, X_n as a surrogate to the distribution F . In particular, note that if n is large, then the “sample distribution function” F_n (also known as the “empirical distribution function”) defined by

$$F_n(x) = n^{-1} \sum_{j=1}^n I(X_j \leq x)$$

is close to $F(\cdot)$, in the sense that the Law of Large Numbers guarantees that

$$F_n(x) \xrightarrow{\text{a.s.}} F(x)$$

as $n \rightarrow \infty$. Note that $F_n(\cdot)$ corresponds to a probability distribution that attaches probability mass n^{-1} to each of the n points X_1, \dots, X_n . Furthermore, the mean, α_{F_n} of the distribution F_n is easily computable:

$$\alpha_{F_n} = \int_{\mathbb{R}} x F_n(dx) = n^{-1} \sum_{i=1}^n X_i (= \alpha_n)$$

because F_n is close to F , it should follow that:

$$P_F\{\alpha_n - \alpha_F \leq \cdot\} \approx P_{F_n}\{\alpha_n^* - \alpha_{F_n} \leq \cdot\}$$

where α_n^* is the average of n observations drawn from the distribution F_n . We now apply the same idea as described earlier to compute $P_{F_n}\{\alpha_n^* - \alpha_{F_n} \leq \cdot\}$ via the Monte Carlo method. Specifically, we draw an iid sample $X_{11}^*, X_{12}^*, \dots, X_{mn}^*$ from the distribution F_n (by sampling each of the n original experiments X_1, \dots, X_n with probability n^{-1}), and compute:

$$\alpha_n^*(1) = n^{-1} \sum_{j=1}^n X_{1j}^*$$

If we repeat this process m independent times, we obtain $\alpha_n^*(1), \dots, \alpha_n^*(m)$. The Law of Large Number ensures that:

$$m^{-1} \sum_{i=1}^m I(\alpha_n^*(i) - \alpha_n \leq x) \rightarrow P_{F_n}\{\alpha_n^* - \alpha_{F_n} \leq x\}$$

as $n \rightarrow \infty$. We now compute z_1^* and z_2^* so that:

$$m^{-1} \sum_{i=1}^m I(z_1^* \leq \alpha_n^*(i) - \alpha_n \leq z_2^*) \approx 1 - \delta$$

and use

$$[\alpha_n - z_2^*, \alpha_n - z_1^*]$$

as our approximate $100(1 - \delta)\%$ confidence interval for $\alpha = E[X_1]$. This confidence interval is called a “bootstrap confidence interval” for α , and each sample $X_{11}^*, \dots, X_{1n}^*$ from the distribution F_n is called a “bootstrap sample”.

Example 3.7: To illustrate this confidence interval procedure, suppose that our goal is to produce a confidence interval for $\alpha = E[X]$. We first do $n = 3$ computer experiments yielding 0.6, 1.9, 1.1. Then $\alpha_3 = 1.2$. We then generate $m = 4$ bootstrap samples from $\{0.6, 1.9, 1.1\}$:

$$\begin{aligned} 1.1, 1.1, 0.6 & \quad (\alpha_3^*(1) = 0.9) \\ 0.6, 1.1, 1.9 & \quad (\alpha_3^*(2) = 1.2) \\ 1.9, 0.6, 1.9 & \quad (\alpha_3^*(3) = 1.5) \\ 0.6, 0.6, 1.9 & \quad (\alpha_3^*(4) = 1.0) \end{aligned}$$

So, the bootstrap approximation to (for example) $P_F\{\alpha_n - \alpha_F \leq 0\}$ is $3/4$.

Remark 3.33: The confidence interval procedure of Section 12 (based on the normal approximation) yields an approximate confidence interval, not an exact confidence interval. How good an approximation is it? It can be shown that:

$$\left| P \left\{ \alpha < \alpha_n - z \frac{s_n}{\sqrt{n}} \right\} - P \{ N(0, 1) > z \} \right| = O \left(\frac{1}{\sqrt{n}} \right)$$

so the “coverage error” decreases at rate $n^{-\frac{1}{2}}$. (This has to do with the fact that the rate of convergence in the CLT is of order $n^{-\frac{1}{2}}$.) The above bootstrap procedure has the same accuracy:

$$|P_F\{\alpha < \alpha_n - z\} - P_{F_n}\{\alpha_{F_n} \leq \alpha_n^* - z\}| = O\left(\frac{1}{\sqrt{n}}\right)$$

A modified version of the above bootstrap confidence interval procedure can yield a confidence interval with even better coverage characteristics than a confidence interval based on the normal approximation. The idea is to compute the sample standard deviation $s_n^*(i)$ from each bootstrap sample $X_{i1}^*, \dots, X_{in}^*$, namely

$$s_n^*(i) = \sqrt{\frac{1}{n-1} \sum_{j=1}^n (X_{ij}^* - \alpha_n^*(i))^2}$$

Then,

$$\left| P_F \left\{ \alpha < \alpha_n - z \frac{\sigma}{\sqrt{n}} \right\} - P_{F_n} \left\{ \alpha_{F_n} < \alpha_n^* - z \frac{s_n^*}{\sqrt{n}} \right\} \right| = O(n^{-1})$$

as $n \rightarrow \infty$. In this setting, the bootstrap approximation has a convergence error of order n^{-1} (rather than $n^{-\frac{1}{2}}$). This can be used to construct bootstrap confidence intervals that have higher accuracy than those based on the normal approximation ideas of Section 12. The bootstrap is correcting for non-normality in the distribution of X , thereby yielding more accurate confidence intervals.

References

Hall, P. The Bootstrap and Edgeworth Expansion. Springer-Verloy: 1992.

3.14 More Complex Monte Carlo Computations

We have previously described the use of the Monte Carlo method to compute $\alpha = E[X]$, where X is a random variable that can be simulated in finite time (i.e. is “simulatable”). However, there are certain problems in which more complicated computations are of interest:

Example 3.8: Suppose that we wish to compute the standard deviation of a random variable Y via the Monte Carlo method, where Y is simulatable. Note that:

$$\sigma = (E[Y^2] - (E[Y])^2)^{\frac{1}{2}}$$

can not be expressed as the mean of a simulatable random variable.

Example 3.9: The median of a continuous random variable Y is defined as the smallest value of m for which:

$$P\{Y \leq m\} = \frac{1}{2}$$

Note that m is defined as the root of an equation (and is not of the form $\alpha = E[X]$ for some simulatable random variable X).

Example 3.10: Assume that we wish to compute the density $f(y)$ of a continuous random variable Y at the point y . Again, the quantity $f(y)$ is not of the form $\alpha = E[X]$.

Example 3.11: Suppose that we wish to use the Monte Carlo method to maximize an objective function $\alpha(\theta) = E[X(\theta)]$ over a decision variable $\theta \in \mathbb{R}^d$. Given a grid $\theta_1, \dots, \theta_m$, perform n independent simulations $X_1(\theta_i), \dots, X_n(\theta_i)$ at each point θ_i , and estimate the maximum $\alpha(\theta^*)$ via:

$$\max_{1 \leq i \leq m} \alpha_n(\theta_i)$$

where

$$\alpha_n(\theta_i) = n^{-1} \sum_{j=1}^n X_j(\theta_i)$$

In each of the above problems it is easy to propose an estimator for the quantity of interest.

Example 8 (continued) Suppose that we perform n iid simulations, thereby generating n iid random variables Y_1, \dots, Y_n that are copies of the random variable Y . The obvious estimator for σ is

$$\sigma_n = \left(n^{-1} \sum_{i=1}^n Y_i^2 - \left(n^{-1} \sum_{j=1}^n Y_j \right)^2 \right)^{\frac{1}{2}}$$

Example 9 (continued) Again, we generate n iid copies Y_1, \dots, Y_n of the random variables Y . We estimate m via:

$$m_n = \sup \left\{ x : n^{-1} \sum_{j=1}^n I(Y_j \leq x) < \frac{1}{2} \right\}$$

the “sample median” of the observations Y_1, \dots, Y_n .

Example 10 (continued) To estimate the density of the random variables Y at the point y , we need to “smooth” the observation that are collected in a neighborhood of y . Let $\phi(\cdot)$ be the density of a $N(0, 1)$ random variable, and consider (for $h > 0$):

$$f_n(y) = \sum_{j=1}^n \frac{1}{nh} \phi\left(\frac{y - Y_j}{h}\right)$$

as an estimator of the density f at y . This estimator smooths the discrete sample distribution that attaches a probability n^{-1} to each point Y_1, \dots, Y_n into a continuous distribution having density $f_n(\cdot)$. The parameter h is a “bandwidth” parameter that has a big influence on the nature of the density estimator f_n . For example, when h is large, $f_n(\cdot)$ tends to be very smooth and $f_n(y)$ effectively averages over all the observation that are roughly within distance h of y . Of course, since $f(y)$ measures the likelihood only at the point y , h must somehow shrink to zero as the number of observations n tends to infinity. This type of density estimator is called a “kernel estimator”, and $\phi(\cdot)$ is the corresponding kernel.

Example 11 (continued) Here, the estimator is self-evident, namely:

$$\max_{1 \leq i \leq m} \alpha_n(\theta_i)$$

A key issue in these problems is the determination of error bars (or equivalently, confidence intervals).. We turn to the question next.

3.15 The Delta Method and Small Noise Approximations

Given a random variables vector W , many applications call for computing the distribution of $g(W)$, where g is a (deterministic) smooth function. We say that W has “small noise” if $W - E[W]$ is mall. In this case, Taylor expansions can be quite informative:

$$g(W) \approx g(E[W]) + \nabla g(E[W])(W - E[W])$$

In other words, if g is smooth and W has “small noise”, then $g(W)$ is (approximately) an affine combination of the components of W . In other words, $g(W)$ is essentially linear in W .

This is of particular importance in the case that W has both “small noise” and is approximately multivariate normally distributed.

Definition 3.10: Let μ be a (deterministic) column d -vector, and let C be a (deterministic) non-negative definite symmetric $d \times d$ matrix. Then, the d -dimensional random vector W is said to have a multivariate normal distribution with mean μ and covariance matrix C if for each (column vector) $\theta \in \mathbb{R}^d$,

$$E \left[e^{i\theta^T W} \right] = e^{i\theta^T \mu - \theta^T C \theta / 2}$$

We use the notation $N(\mu, C)$ to denote this multivariate normal random vector.

When C is positive definite, the density of W is just

$$(2\pi)^{-\frac{d}{2}} |\det C|^{-\frac{1}{2}} e^{-(W-\mu)^T C^{-1} (W-\mu) / 2}$$

Exercise 3.2: Suppose that

$$W \stackrel{D}{=} N(\mu, C)$$

1. Prove that $E[W] = \mu$
2. Prove that

$$E \left[(W - E[W])(W - E[W])^T \right] = C$$

3. Prove that if B is a (deterministic) $d \times n$ matrix then:

$$BW \stackrel{D}{=} N(B\mu, BCB^T)$$

It follows from part (3) that an affine function of a multivariate normal random variable is again a multivariate normal random variable. This makes the “small noise approximation” particularly easy to apply when W is multivariate, normally distributed.

Example 3.12: Suppose that a surveyor maps an area that is described as a rectangle with height h and width w . Measurement errors occur and the measured height and width are random variables H and W , respectively. Suppose that $(H, W)^T$ are bivariate normally distributed with mean $(h, w)^T$ and

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

The purchaser of the land is particularly interested in the measurement error for the total area. The true area is $a = hw$, there the measured area is $A = HW$, so the measurement error for the area is $A - a = HW - hw$. Note that:

$$A = g(H, W)$$

where $g(x_1, x_2) = x_1x_2$. If $\|C\|$ is small, the “small noise approximation” yields

$$A - a \approx w(H - h) + h(W - w) \stackrel{D}{=} N(0, w^2C_{11} + 2hwC_{12} + h^2C_{22})$$

In other words, the measurement error for the area is approximately normally distributed with mean zero and variance:

$$w^2C_{11} + 2hwC_{12} + h^2C_{22}$$

Example 3.13: Suppose that we have a population that is growing exponentially in time, so that:

$$x(t) = be^{at}$$

for some (deterministic) constants a and b , where $x(t)$ is the population at time t . Both the initial population size b and the rate constant a have been measured with error. We view the measurement $(A, B)^T$ as a bivariate normal random vector with mean $(a, b)^T$ and

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

where $\|C\|$ is small. The estimated population size at time t is

$$X(t) = Be^{At}$$

so that the “small noise approximation” asserts that

$$X(t) - x(t) \stackrel{D}{=} N(0, \sigma^2(t))$$

where

$$\sigma^2(t) = x(t)^2 \left(a^2C_{11} + 2\frac{a}{b}C_{12} + \frac{1}{b^2}C_{22} \right)$$

Exercise 3.3: Suppose that $x(\cdot)$ is the solution to a deterministic differential equation

$$\frac{d}{dt}x(t) = \phi(\theta, x(t))$$

such that

$$x(t) = x_0$$

where ϕ is deterministic and θ represents a vector of parameters. (For example, $\phi(\theta, x) = e^{\theta x}$ in Example 13.) Assume that x_0 and θ are measured with error, and that $(X_0, \hat{\theta})^T$ are multivariate normally distributed with mean $(x_0, \theta)^T$.

1. Compute the small noise approximations for the solution $X(t)$ to

$$\frac{d}{dt}X(t) = \phi(\hat{\theta}, X(t))$$

such that

$$X(t) = X_0$$

2. Discuss the computational issues that arise in computing the variance of your small noise approximation

The small noise approximation plays an important role in dealing with our “error bar” problem. However, we need one more definition and result.

Definition 3.11: Let W be a random d -vector (written as a column vector), satisfying $E\|W\|^2 < \infty$. The $d \times d$ matrix

$$E[(W - E[W])(W - E[W])^T]$$

is called the covariance matrix of W .

The following is the multivariate generalization of the central limit theorem described in Section 6 (and can be proved via an identical characteristic function argument as in Section 10)

Result 3.10: Suppose that X_1, X_2, \dots is an iid sequence of random d -vectors with $E[\|X_1\|^2] < \infty$. Let $\mu = E[X_1]$ and C be the covariance matrix of X_1 . Then

$$n^{\frac{1}{2}} \left(n^{-1} \sum_{j=1}^n X_j - \mu \right) \Rightarrow N(0, C)$$

as $n \rightarrow \infty$.

This suggests the approximation

$$n^{-1} \sum_{j=1}^n X_j \stackrel{D}{\approx} \mu + n^{-\frac{1}{2}} N(0, C)$$

where n is large.

Suppose now that we wish to use the Monte Carlo method to compute $\alpha = g(E[X])$, where g is smooth and X is a random d -vector. The obvious estimator is:

$$\alpha_n = g(\bar{X}_n)$$

where $\bar{X}_n = n^{-1} \sum_{l=1}^n X_l$. If n is large, the multivariate CLT asserts that:

$$\bar{X}_n \stackrel{D}{\approx} N(E[X], n^{-1}C)$$

so that \bar{X} is approximately multivariate normal with “small noise”. The small noise approximation that yields

$$\alpha_n - \alpha \stackrel{D}{\approx} N(0, n^{-1}\sigma^2)$$

where

$$\sigma^2 = \nabla g(E[X])C\nabla g(E[X])^T$$

(Here, we follow the convention that all gradients are written as row vectors.) In other words, we have established that our estimator α_n has, for large n , a normal distribution with mean α and variance $\sigma^2 n^{-1}$.

This can be made rigorous by appealing to the ideas described in Section 12.

Result 3.11: Suppose that $E[\|X_1\|^2] < \infty$ and that $g(\cdot)$ is continuously differentiable in a neighborhood of $E[X_1]$. Then

$$n^{\frac{1}{2}}(\alpha_n - \alpha) \Rightarrow N(0, \sigma^2)$$

as $n \rightarrow \infty$, where $\sigma^2 = \nabla g(E[X])C\nabla g(E[X])^T$.

It follows that if $\sigma^2 > 0$, then

$$\left[\alpha_n - z \frac{\sigma}{\sqrt{n}}, \alpha_n + z \frac{\sigma}{\sqrt{n}} \right]$$

is an (approximate) $100(1 - \delta)\%$ confidence interval for $\alpha = g(\mathbb{E}[X])$, provided that z is selected so that $P\{-z \leq N(0, 1) \leq z\} = 1 - \delta$.

Of course, σ^2 is unknown, so it must be estimated from the simulated data. Put

$$\sigma_n^2 = \nabla g(\bar{X}_n) C_n \nabla g(\bar{X}_n)^T$$

where

$$C_n = \frac{1}{n-1} \sum_{j=1}^n (X_j - \bar{X}_n)(X_j - \bar{X}_n)^T$$

Result 3.12: Suppose that $\mathbb{E}[\|X_1\|^2] < \infty$ and that $g(\cdot)$ is continuously differentiable in a neighborhood of $\mathbb{E}[X_1]$.

1.

$$\sigma_n^2 \xrightarrow{\text{a.s.}} \sigma^2$$

as $n \rightarrow \infty$.

2. If $\sigma^2 > 0$, then

$$P \left\{ \alpha \in \left[\alpha_n - z \sqrt{\frac{\sigma_n^2}{n}}, \alpha_n + z \sqrt{\frac{\sigma_n^2}{n}} \right] \right\} \rightarrow P\{-z \leq N(0, 1) \leq z\}$$

as $n \rightarrow \infty$.

This provides theoretical justification for the following algorithm.

Algorithm A5: Computation of an approximate $100(1 - \delta)\%$ confidence interval for $\alpha = g(\mathbb{E}[X])$

1	Select δ and choose n (the number of computer runs).
2	Generate n iid copies X_1, \dots, X_n of the random variable X .
3	Compute
4	$\bar{X}_n = n^{-1}(X_1 + \dots + X_n)$
5	$C_n = \frac{1}{n-1} \sum_{j=1}^n (X_j - \bar{X}_n)(X_j - \bar{X}_n)^T$
6	Compute
7	$\alpha_n = g(\bar{X}_n)$
8	$\sigma_n^2 = \nabla g(\bar{X}_n) C_n \nabla g(\bar{X}_n)^T$
9	Select z so that $P\{-z \leq N(0, 1) \leq z\} = 1 - \delta$.
10	Then $\left[\alpha_n - z \sqrt{\frac{\sigma_n^2}{n}}, \alpha_n + z \sqrt{\frac{\sigma_n^2}{n}} \right]$ is an $100(1 - \delta)\%$ confidence interval for α .

Remark 3.34: The use of the multivariate CLT and small noise approximation to develop approximate confidence intervals for $\alpha = g(\mathbb{E}[X])$ is called the “delta method” by statisticians.

Let us now apply the delta method to Example 8.

Example 8 (continued) Note that

$$\sigma = g(\mathbb{E}[X])$$

where $X = (Y^2, Y)$ and $g(x_1, x_2) = (x_1 - x_2^2)^{\frac{1}{2}}$. Here

$$\nabla g(x_1, x_2) = \left(\frac{1}{2}(x_1 - x_2^2)^{-\frac{1}{2}}, -x_2(x_1 - x_2^2)^{-\frac{1}{2}} \right)$$

and

$$\begin{aligned} C_{11} &= E [Y^4] - (E [Y^2])^2 \\ C_{12} &= E [Y^3] - (E [Y^2])(E [Y]) \\ C_{22} &= E [Y^2] - (E [Y])^2 \end{aligned}$$

Algorithm A5 can then be implemented in a straightforward fashion for this example.

Example 9 (continued) A similar approach, based on the delta method, can also be attempted here. Let:

$$F_n(x) = n^{-1} \sum_{j=1}^n I(Y_j \leq x)$$

be the sample distribution function of the Y_j s. The sample median approximately satisfies the equation

$$F_n(m_n) = \frac{1}{2}$$

(In fact, it exactly satisfies the equation for n odd.) Then

$$F_n(m_n) - F_n(m) = \frac{1}{2} - F_n(m)$$

It is easy to argue that when $P\{Y \leq \cdot\}$ is strictly increasing and continuous, then $m_n \xrightarrow{\text{a.s.}} m$ as $n \rightarrow \infty$. When n is large, one might hope that:

$$F_n(m_n) - F_n(m) \approx F(m_n) - F(m) \approx F'(m)(m_n - m) = f(m)(m_n - m)$$

where $f(m)$ is the density of the random variable Y at the median m . (Mathematically, this step (while correct) is problematic because $F_n(\cdot)$ is non-differentiable, so care must be taken to justify this step (assuming a rigorous proof is desired).) On the other hand, the ordinary CLT of Section 8 asserts that

$$\frac{1}{2} - F_n(m) \stackrel{D}{\approx} N\left(0, \frac{1}{4n}\right)$$

for large n . We conclude that

$$m_n - m \stackrel{D}{\approx} N\left(0, \frac{1}{4nf(m)^2}\right)$$

for large n . Hence,

$$\left[m_n - z\sqrt{\frac{1}{4nf(m)^2}}, m_n + z\sqrt{\frac{1}{4nf(m)^2}} \right]$$

is an approximate $100(1 - \delta)\%$ confidence interval for the median m , provided that z is selected so that $P\{-z \leq N(0, 1) \leq z\} = 1 - \delta$.

Exercise 3.4: Develop a corresponding approximation confidence interval for $q(p)$, where $q(p)$ is the “ p^{th} quantile” of the random variable Y defined as the smaller root of the equation

$$P\{Y \leq q(p)\} = p$$

Such quantile computations are of interest on “value at risk” calculations in the finance setting.

Note that the above confidence interval procedure can not be implemented operationally, since $f(m)$ is typically unknown. Hence, one needs some means of estimating the density $f(m)$ from the observed data Y_1, \dots, Y_n . We discussed this issue in the context of Example 9; more will be added in Section 16.

We conclude this section with a rigorous statement of the limit theorem that justifies the above confidence interval.

Result 3.13: Let Y be a continuous random variable with a density $f(\cdot)$ that is continuous and positive at m . Then

$$n^{\frac{1}{2}}(m_n - m) \Rightarrow N\left(0, \frac{1}{4f(m)^2}\right)$$

as $n \rightarrow \infty$.

3.16 Kernel-based Density Estimation

We now discuss kernel-based density estimation in greater detail, both because of the problem's intrinsic importance and because the ideas discussed here are useful in many other settings.

Recall that our kernel-based density estimator takes the form:

$$f_n(y) = \sum_{j=1}^n \frac{1}{nh} \phi\left(\frac{y - Y_j}{h}\right)$$

A key issue is the determination of a good value for the bandwidth (or “smoothing parameter”) h .

Definition 3.12: Suppose that Γ is an estimator of γ . The mean square error of the estimator Γ is

$$E[(\Gamma - \gamma)^2]$$

Definition 3.13: Suppose that Γ is an estimator of γ . The bias of Γ is $E[\Gamma] - \gamma$. If the bias is zero, the estimator Γ is said to be unbiased.

Note that

$$\begin{aligned} E[(\Gamma - \gamma)^2] &= E[(\Gamma - E[\Gamma] + E[\Gamma] - \gamma)^2] \\ &= E[(\Gamma - E[\Gamma])^2] + 2E[(\Gamma - E[\Gamma])(E[\Gamma] - \gamma)] + (E[\Gamma] - \gamma)^2 \\ &= \text{var}(\Gamma) + (\text{bias of } \Gamma)^2 \end{aligned}$$

so the mean squared error decomposes into the sum of the variance and the squared bias. The mean square error is therefore easily computable for many estimators. In particular, this can be done for our kernel-based density estimator.

Note that when h is small,

$$\begin{aligned}
\mathbb{E}[f_n(y)] &= \frac{1}{h} \mathbb{E} \left[\phi \left(\frac{y - Y_1}{h} \right) \right] \\
&= \int_{-\infty}^{\infty} \frac{1}{h} \phi \left(\frac{y - z}{h} \right) f(z) dz \\
&= \int_{-\infty}^{\infty} \phi(u) f(y - uh) du \\
&\approx \int_{-\infty}^{\infty} \phi(u) \left[f(y) - uh f'(y) + \frac{u^2 h^2}{2} f''(y) \right] du \\
&= f(y) + \frac{h^2}{2} f''(y)
\end{aligned}$$

so the bias of $f_n(y)$ (for h small) is $(h^2/2)f''(y)$ approximately.

As for the variance,

$$\begin{aligned}
\text{var}(f_n(y)) &= \frac{1}{nh^2} \text{var} \left(\phi \left(\frac{y - Y_1}{h} \right) \right) \\
&= \frac{1}{nh^2} \mathbb{E} \left[\phi^2 \left(\frac{y - Y_1}{h} \right) \right] - \frac{1}{n} (\mathbb{E}[f_n(y)])^2 \\
&= \frac{1}{nh^2} \mathbb{E} \left[\phi^2 \left(\frac{y - Y_1}{h} \right) \right] - \frac{f^2(y)}{n} + O \left(\frac{h^2}{n} \right)
\end{aligned}$$

But

$$\begin{aligned}
\frac{1}{h} \mathbb{E} \left[\phi^2 \left(\frac{y - Y_1}{h} \right) \right] &= \int_{-\infty}^{\infty} \frac{1}{h} \phi^2 \left(\frac{y - z}{h} \right) f(z) dz \\
&= \int_{-\infty}^{\infty} \phi^2(u) f(y - uh) du
\end{aligned}$$

since h is small

$$\begin{aligned}
&\approx \int_{-\infty}^{\infty} \phi^2(u) f(y) du \\
&= f(y) \int_{-\infty}^{\infty} \phi^2(u) du \\
&= f(y) \int_{-\infty}^{\infty} \frac{1}{2\pi} e^{-u^2} du \\
&= \frac{1}{2\sqrt{\pi}} f(y)
\end{aligned}$$

So, for h small,

$$\text{var}(f_n(y)) \approx \frac{1}{2\sqrt{\pi}} \frac{1}{nh} f(y)$$

A reasonable criterion to use for choosing h is:

Choose h to minimize the mean squared error of $f_n(y)$.

Our analysis above therefore suggests choosing h to minimize:

$$\frac{1}{2\sqrt{\pi}} \frac{1}{nh} f(y) + \frac{h^4}{4} f''(y)^2$$

The minimizing h^* is

$$h^* = n^{-\frac{1}{5}} \left(\frac{f(y)}{2\sqrt{\pi} f''(y)^2} \right)^{\frac{1}{5}}$$

To get a sense of the rate of convergence of $f_n(y)$ to $f(y)$ with this choice of h , note that because $f_n(y)$ is an average of a large number n of iid random variables, the CLT suggests that

$$f_n(y) \stackrel{D}{\approx} N(Ef_n(y), \text{var}(f_n(y))) \approx f(y) + \frac{h^2}{2} f''(y) + \frac{1}{\sqrt{nh}} N\left(0, \frac{f(y)}{2\sqrt{\pi}}\right)$$

Consequently, with $h = h^*$, the corresponding rate of convergence of $f_n(y)$ to $f(y)$ is of order $n^{-\frac{2}{5}}$. This, of course, is a slower rate of convergence than the order $n^{-\frac{1}{2}}$ rate associated with our previous Monte Carlo estimators. This is not surprising, given that very few observations fall close enough to y to be useful in computing the density estimate at that point.

The optimal choice h^* of the smoothing parameter h depends on the values of $f(y)$ and $f''(y)$, both of which are unknown. Practically implementable methods for computing good choices for h are the subject of an enormous literature; “cross validation” is one widely recommended method.

Remark 3.35: The analysis developed above can be made rigorous.

The above analysis offers some useful general insights:

1. Many statistical problems require local “smoothing” of the data. For example, suppose that we wish to estimate some function (in the presence of “noise” in the function evaluation). All the function evaluations that are gathered in a neighborhood of a point y should contribute to estimating the value of the function there. In other words, we wish to “smooth” all the “local data” to produce an estimate at that point. Use of “smoothing kernels” (like $\phi(\cdot)$) are a standard means of attacking this problem. Such smoothing problems arise in many different areas of engineering (e.g. imaging) and computer science (e.g. learning algorithms).
2. Many estimator procedures contain statistical parameters (like h) that must be “tuned”. Mean square error (MSE) analysis is a standard tool for developing insight into such issues.
3. Computation of error bars (i.e. confidence intervals) for smoothing problems is challenging. While our analysis developed a normal approximation for $f_n(y)$ development of a confidence interval procedure based on this normal approximation is problematic (because of the presence of $f(y)$ and $f''(y)$ in the approximation). Error bar computation is even more challenging if the estimator procedure includes some statistically complicated techniques for choosing the smoothing parameter h .

Point 3 makes clear that while confidence intervals can, in principle, be based on analytically derived limit theorems (like normal approximations based on the CLT), it can be useful to have an alternative methodology (that is easily implemented by users) to solve this problem.

3.17 Return to the Bootstrap

Consider Examples 8 through 11 introduced earlier, and the development of associated error bars for the corresponding Monte Carlo estimators. Recall that in Example 8, Algorithm A5 was applied. Of course it required computing $\nabla g(\cdot)$. This can be non-trivial on some problems. In Example 9, a confidence interval procedure for m , based on the normal approximation was developed, provided that $f(m)$ could be estimated. Estimation of $f(m)$ is possible (see Example 8), but is not easy. For Examples 10 and 11 no operationally useful confidence interval procedure was proposed.

It turns out that the bootstrap offers a solution to all four of these problems, as well as a general means of attacking an enormous spectrum of other such problems. The bootstrap idea is to apply Monte Carlo to produce the required intervals. We now describe how the bootstrap can be applied to Examples 6 through 9.

Example 8 (continued) We approximate

$$P_F\{\sigma_n - \sigma \leq \cdot\}$$

by

$$P_{F_n}\{\sigma_n^* - \sigma_n \leq \cdot\}$$

In view of this approximation, we draw $Y_{11}^*, \dots, Y_{1n}^*$ independently, from $\{Y_1, \dots, Y_n\}$ (each with probability n^{-1}), and compute

$$\sigma_n^*(1) = \left(n^{-1} \sum_{i=1}^n Y_{1i}^{*2} - \left(\frac{1}{n} \sum_{j=1}^n Y_{1j}^* \right)^2 \right)^{\frac{1}{2}}$$

If we repeat this process m independent times, we obtain $\sigma_n^*(1), \dots, \sigma_n^*(m)$. We now compute z_1^* and z_2^* so that

$$m^{-1} \sum_{i=1}^m I(z_1^* < \sigma_n^*(i) - \sigma_n \leq z_2^*) \approx 1 - \delta$$

and use

$$[\sigma_n - z_2^*, \sigma_n - z_1^*]$$

as an approximate $100(1 - \delta)\%$ confidence interval for σ .

Example 9 (continued) We approximate

$$P_F\{m_n - m \leq \cdot\}$$

by

$$P_{F_n}\{m_n^* - m_n \leq \cdot\}$$

Again, in view of this approximation, we draw $Y_{11}^*, \dots, Y_{1n}^*$ independently from $\{Y_1, \dots, Y_n\}$ (each with probability n^{-1}). Compute the sample median $m_n^*(1)$ of the observations $\{Y_{11}^*, \dots, Y_{1n}^*\}$. Repeat this process m independent times to obtain $m_n^*(1), \dots, m_n^*(m)$. Finally, compute z_1^* and z_2^* so that

$$m^{-1} \sum_{i=1}^m I(z_1^* < m_n^*(i) - m_n \leq z_2^*) \approx 1 - \delta$$

and use

$$[m_n - z_2^*, m_n - z_1^*]$$

as an approximate $100(1 - \delta)\%$ confidence interval for m .

Example 10 (continued) We approximate

$$P_F\{f_n(y) - f(y) \leq \cdot\}$$

by

$$P_{F_n}\{f_n^*(y) - f_n(y) \leq \cdot\}$$

Here, $f_n(y)$ is a kernel estimator for $f(y)$ that uses bandwidth parameter h_n . The parameter h_n is obtained by running an algorithm (that we shall denote H) on $\{Y_1, \dots, Y_n\}$. We do not specify the algorithm here. (It could be any of many that have been proposed in the literature, or one of your own making!).

To compute the bootstrap confidence interval, we draw $Y_{11}^*, \dots, Y_{n1}^*$ independently from $\{Y_1, \dots, Y_n\}$ (each with probability n^{-1}). Run algorithm H on $\{Y_{11}^*, \dots, Y_{n1}^*\}$ to compute $h_n^*(1)$. Now, calculate $f_n^*(y, 1)$ from $\{Y_{11}^*, \dots, Y_{n1}^*\}$ using the smoothing parameter $h_n^*(1)$. Repeating this process m independent times, we obtain $\{f_n^*(y, 1), \dots, f_n^*(y, m)\}$. We now compute z_1^* and z_2^* so that

$$m^{-1} \sum_{i=1}^m I(z_1^* < f_n^*(y, i) - f_n(y) \leq z_2^*) \approx 1 - \delta$$

and use

$$[f_n(y) - z_2^*, f_n(y) - z_1^*]$$

as an approximate $100(1 - \delta)\%$ confidence interval for $f(y)$.

Example 11 (continued) We use here the notation F to characterize the distribution of the $X(\theta)$'s (as a function of θ). In view of this notation, we approximate

$$P_F \left\{ \max_{1 \leq i \leq m} \alpha_n(\theta_i) - \alpha(\theta^*) \leq \cdot \right\}$$

by

$$P_{F_n} \left\{ \max_{1 \leq i \leq m} \alpha_n^*(\theta_i) - \max_{1 \leq i \leq m} \alpha_n(\theta_i) \leq \cdot \right\}$$

At each point θ_i , draw $X_{11}^*(\theta_i), \dots, X_{1n}^*(\theta_i)$ independently from $\{X_1(\theta_i), \dots, X_n(\theta_i)\}$ (each with probability n^{-1}), and compute

$$\max_{1 \leq i \leq m} \alpha_n^*(\theta_i, 1)$$

where $\alpha_n^*(\theta_i, 1) = n^{-1} \sum_{j=1}^n X_{1j}^*(\theta_i)$. If we repeat this process m independent times, we obtain

$$\left\{ \max_{1 \leq i \leq m} \alpha_n^*(\theta_i, j) : 1 \leq j \leq m \right\}$$

We now compute z_1^* and z_2^* so that

$$m^{-1} \sum_{j=1}^m I \left(z_1^* < \max_{1 \leq i \leq m} \alpha_n^*(\theta_i, j) - \max_{1 \leq i \leq m} \alpha_n(\theta_i) \leq z_2^* \right) \approx 1 - \delta$$

and use

$$\left[\max_{1 \leq i \leq m} \alpha_n(\theta_i) - z_2^*, \max_{1 \leq i \leq m} \alpha_n(\theta_i) - z_1^* \right]$$

as an approximate $100(1 - \delta)\%$ confidence interval for $\alpha(\theta^*)$.

These four examples serve to illustrate the power of the bootstrap idea across a variety of different problem domains.

References

Devroye, L. and L. Gyurfi. Non-parametric Density Estimation: The L_1 View. John Wiley, New York: 1985.

Hall, P. The Bootstrap and Edgeworth Expansion. Springer-Verloy: 1992.

Serfling, R.J. Approximation Theorems of Mathematical Statistics. John Wiley, New York: 1980.