Approximating Probabilities and Expectations by Repeated Simulation, or, "Monte Carlo"

36-402, Spring 2025

28 January 2025

Contents

1 "Math Is Hard; Let's Go Simulate"

1.1 Calculating Probabilities Means Integrating, and Integration Is Hard

Suppose I ask you the probability that a standard Gaussian falls in the interval $[0.8, 0.9]$. This is easy, you think, the standard Gaussian pdf is

$$\frac{1}{\sqrt{2\pi}} e^{-x^2/2} \tag{1}$$

so the probability is just

$$\int_{0.8}^{0.9} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx \tag{2}$$

how hard can that be to integrate?[1]

[1] Do you remember why $\int_{-\infty}^{\infty} e^{-x^2/2} dx = \sqrt{2\pi}$ anyway? (Be honest with yourself, if not with me.)

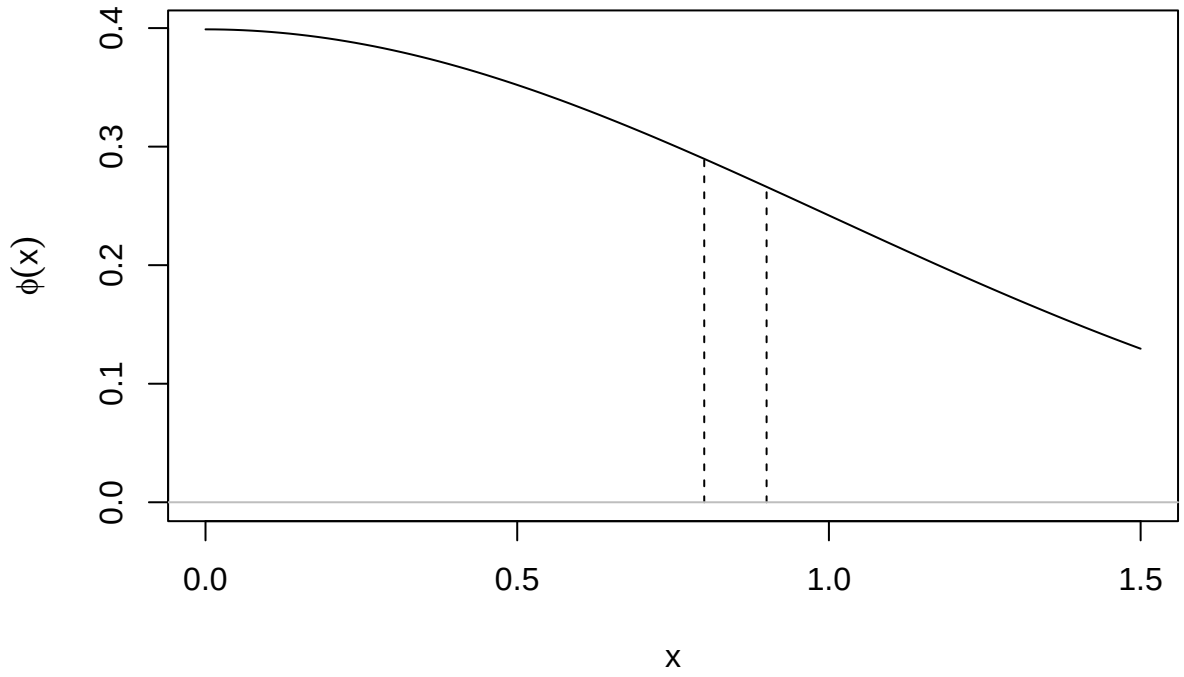


Figure 1: The probability a standard Gaussian is between 0.8 and 0.9 is the area under the curve and betwen the dashed lines — how hard could that integral be?

Well, it turns out that $\Phi(a) = \frac{1}{2\pi}\int_{-\infty}^{a} e^{-x^2/2} dx$ is a transcendental function, with no closed-form algebraic expression. And what I want is $\Phi(0.9) - \Phi(0.8)$ and no that's not going to be a straightforward integral after all.

Now, because of literal centuries of work by very dedicated (and very strange) people, we have extremely accurate approximations to $\Phi(a)$ for arbitrary a. It's easy to use those to calculate both $\Phi(a_1)$ and $\Phi(a_2)$ to (say) 11 decimal places, and so get $\Phi(a_2) - \Phi(a_1)$ to at least 10 decimal places if that's what we need. Those accurate approximations are what are coded up in R behind the name `pnorm()`, so *we* don't even really have to know what those approximations are, just trust the R development team. But suppose we didn't have those available, or were interested in some other distribution which hadn't been obsessively investigated for over 200 years. Would we *really* have to do some kind of raw, painful numerical integration?!?

1.2 Gaussian Probabilities from Gaussian Simulations

R will give us any number of simulated Gaussians using `rnorm()`. Let's try something:

```
big.Z <- rnorm(n=1e6) # Draws from standard Gaussian by default
mean((big.Z <= 0.9) & (big.Z >= 0.8))
```

```
## [1] 0.027535
```

This draws a large number of simulated Gaussians, and then finds the proportion of them which are both ≥ 0.8 and ≤ 0.9. Intuitively, it should seem pretty sensible to think that this is close to the probability that a random Gaussian from this distribution falls between those limits. How close is this to the probability we'd get from $\Phi(0.9) - \Phi(0.8)$?

```
pnorm(0.9) - pnorm(0.8)
```

```
## [1] 0.02779527
```

The simulation-based answer is within 0.9% of the one based on (nearly) exact calculation.

I will let you experiment with changing the interval, changing which Gaussian distribution we're using, etc. Generally, you should convinced yourself that this approach — simulate from the distribution you care about, and see how often your simulations hit the region of interest — gives you decent approximations to the probabilities of intervals, in situations where you have another, reliable way of calculating those probabilities. This should then give you confidence that the method can work in situations where you *don't* know the answer.

2 Approximating Probabilities by Repeated Simulation

Here's the general approach: There's some random variable X, distributed according to some probability distribution P. We want to know the probability that X falls in to some set A, or has some property A. When that's true, we write $X \in A$, and when it's false, we write $X \notin A$. We want to know $P(X \in A)$, a probability I'll abbreviate as p.

We suppose two things:

1. We have a way of **simulating** an unlimited number of random copies of X, $X_1, X_2, \ldots X_b, \ldots$, all with the distribution P, and all statistically independent of each other.
2. For each t, we have some way of telling whether or not $X_t \in A$. This is some mechanically-assessable property which the computer can check, not a human judgment call.

As a final piece of notation, write $I_t = 1$ if $X_t \in A$, and $I_t = 0$ if $X_t \notin A$. (I_t is called the **indicator variable** for the event $X_t \in A$.) Set

$$\hat{p} \equiv \frac{1}{b}\sum_{t=1}^{b} I_t \tag{3}$$

Claim 1: $\hat{p}$ is an *unbiased* approximation to p.

Claim 2: $\hat{p}$ is a *consistent* approximation of p.

Claim 3: For large b, $\hat{p}$ has Gaussian fluctuations around p.

2.1 Unbiasedness (claim 1)

$$\begin{aligned}
\mathbb{E}\left[\hat{p}\right] &= \mathbb{E}\left[\frac{1}{b}\sum_{t=1}^{b} I_t\right] && (4)\\
&= \frac{1}{b}\sum_{t=1}^{b}\mathbb{E}\left[I_t\right] && (5)\\
&= \frac{1}{b}\sum_{t=1}^{b} 1\cdot P(X\in A) + 0\cdot P(X\notin A) && (6)\\
&= \frac{1}{b}\sum_{t=1}^{b} p = \frac{bp}{b} = p && (7)
\end{aligned}$$

2.2 Consistency (claim 2)

Consistency, remember, means that converging on the truth as (here) $b \to \infty$. This is just the law of large numbers, but it won't hurt to remember how to prove that.

$$\begin{aligned}
\mathbb{E}\left[(p-\hat{p})^2\right] &= \mathbb{E}\left[\left(p - \frac{1}{b}\sum_{t=1}^{b} I_t\right)^2\right] && (8)\\
&= \left(p - \mathbb{E}\left[\frac{1}{b}\sum_{t=1}^{b} I_t\right]\right)^2 + \mathrm{Var}\left[\frac{1}{b}\sum_{t=1}^{b} I_t\right] && (9)\\
&= (p-p)^2 + \frac{1}{b^2}\sum_{t=1}^{b}\mathrm{Var}\left[I_t\right] && (10)\\
&= \frac{1}{b^2} bp(1-p) && (11)\\
&= \frac{p(1-p)}{b} && (12)
\end{aligned}$$

Since the expected squared distance[2] between p and $\hat{p}$ is going to 0, $\hat{p} \to p$.

2.3 Variance and standard error of the approximation

Notice that we've just proved that

$$\mathrm{Var}\left[\hat{p}\right] = \frac{p(1-p)}{b} \tag{13}$$

The standard error of an estimator is its standard deviation, i.e., the square root of its variance, so

$$SE(\hat{p}) = \sqrt{\frac{p(1-p)}{b}} \tag{14}$$

This involves the unknown probability we're trying to estimate, but a feasible approximation is

$$\widehat{SE}(\hat{p}) = \sqrt{\frac{\hat{p}(1-\hat{p})}{b}} \tag{15}$$

2.4 Gaussian fluctuations (Claim 3)

$\hat{p}$ is an average of independent, identically-distributed random variables, so the central limit theorem applies. In this case, it takes the form

$$\sqrt{b}\frac{\hat{p}-p}{p(1-p)} \rightsquigarrow \mathcal{N}(0,1) \tag{16}$$

More practically, if more likely to make mathematicians upset,

[2] If you've taken more advanced probability courses, you're familiar with the idea that there are different sorts or "modes" of convergence for random variables. (If you don't know what that means, skip the rest of this footnote, which will read like so much "bar, bar, bar" in any case.) When $\mathbb{E}\left[(X_n - Y_n)^2\right] \to 0$, we say that X_n and Y_n "converge in mean square", or "converge in L_2". This implies "convergence in probability", i.e., for all $\epsilon > 0$, $\mathbb{P}\left[|X_n - Y_n| > \epsilon\right] \to 0$ (use Chebyshev's inequality), but not vice versa. Mean-square convergence is a little unusual at this level of statistics course, but as you can tell from this example, it often involves more elementary mathematical tools than convergence in probability.

$$\hat{p} \rightsquigarrow \mathcal{N}\left(p, \frac{p(1-p)}{b}\right) \tag{17}$$

We can use this, in the usual way, to calculate confidence intervals, etc., etc.

2.5 How many simulation runs?

We saw above that the standard error of $\hat{p}$ is

$$SE(\hat{p}) = \sqrt{\frac{p(1-p)}{b}} \tag{18}$$

Suppose we want the standard error of our approximation to be at most δ. We can use that to work out how many simulations we need:

$$\begin{aligned}
\delta &\leq \sqrt{\frac{p(1-p)}{b}} && (19)\\
\delta^2 &\leq \frac{p(1-p)}{b} && (20)\\
b &\geq \frac{p(1-p)}{\delta^2} && (21)
\end{aligned}$$

This still involve the unknown p, but we can bound this conservatively, by noticing that $p(1-p) \leq 1/4$. So if want an accuracy of $\pm\delta$, $\frac{1/4}{\delta^2}$ simulations will definitely do the job. For instance, if we want to ensure the standard error is at most 10^{-2}, we need at most 2500 simulations. — If p is far from $1/2$ in either direction, we might be able to get away with considerably fewer, so a "pilot run" to get a rough approximation to p can be a good idea.

Often, especially when the probability we're interested in is small, we care about *relative* accuracy, accuracy as a fraction of the value in question. The relative error we're looking at is

$$\frac{SE(\hat{p})}{p} = \sqrt{\frac{1-p}{bp}} \tag{22}$$

If we want to ensure that the relative error is at most r, we can solve for b:

$$\begin{aligned}
r &\leq \sqrt{\frac{1-p}{bp}} && (23)\\
r^2 &\leq \frac{1-p}{bp} && (24)\\
b &\geq \frac{1-p}{pr^2} && (25)
\end{aligned}$$

Notice that $\frac{1-p}{p}$ *doesn't* have a finite upper bound that holds for all p (unlike $p(1-p)$), so we can't give a conservative formula here.

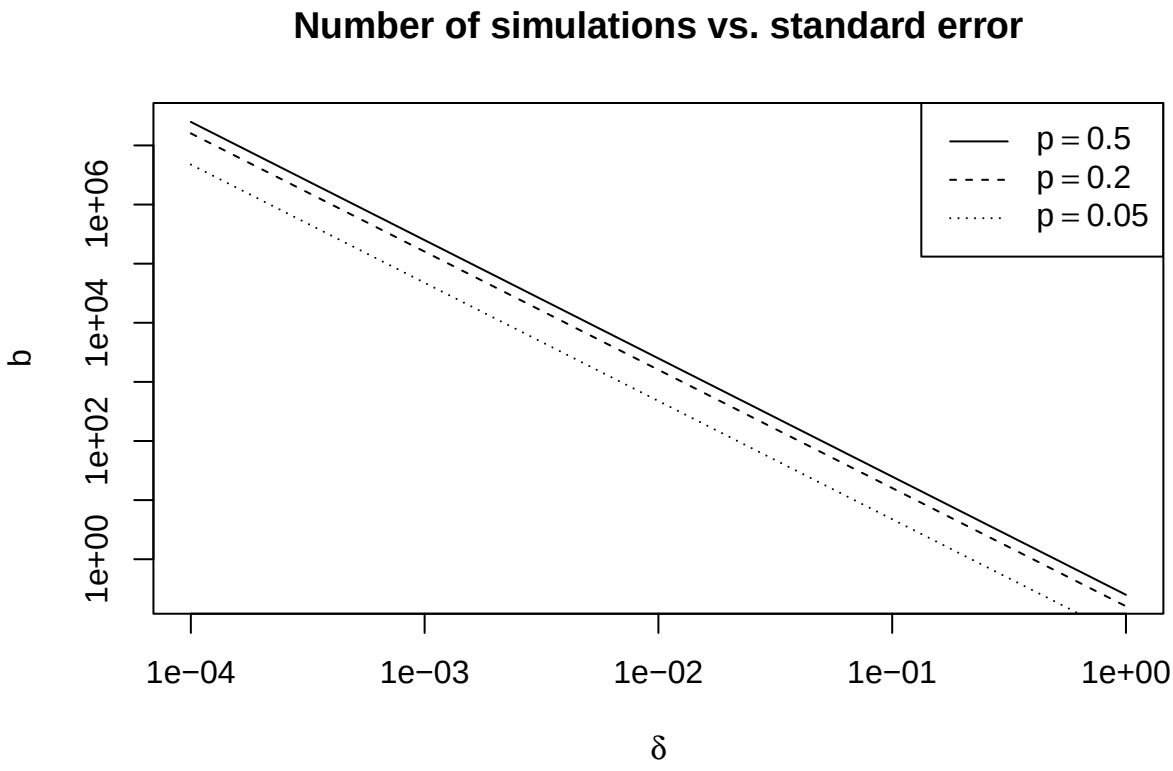


Figure 2: Number of simulations b required to guarantee the standard error of the approximate probability $\hat{p}$ is at most δ, as a function of the true probability p. (The same bound would hold for $q = 1 - p$, so, for example, the $p = 0.05$ line is also the one for $q = 0.95$.) Note the log scale on both axes.

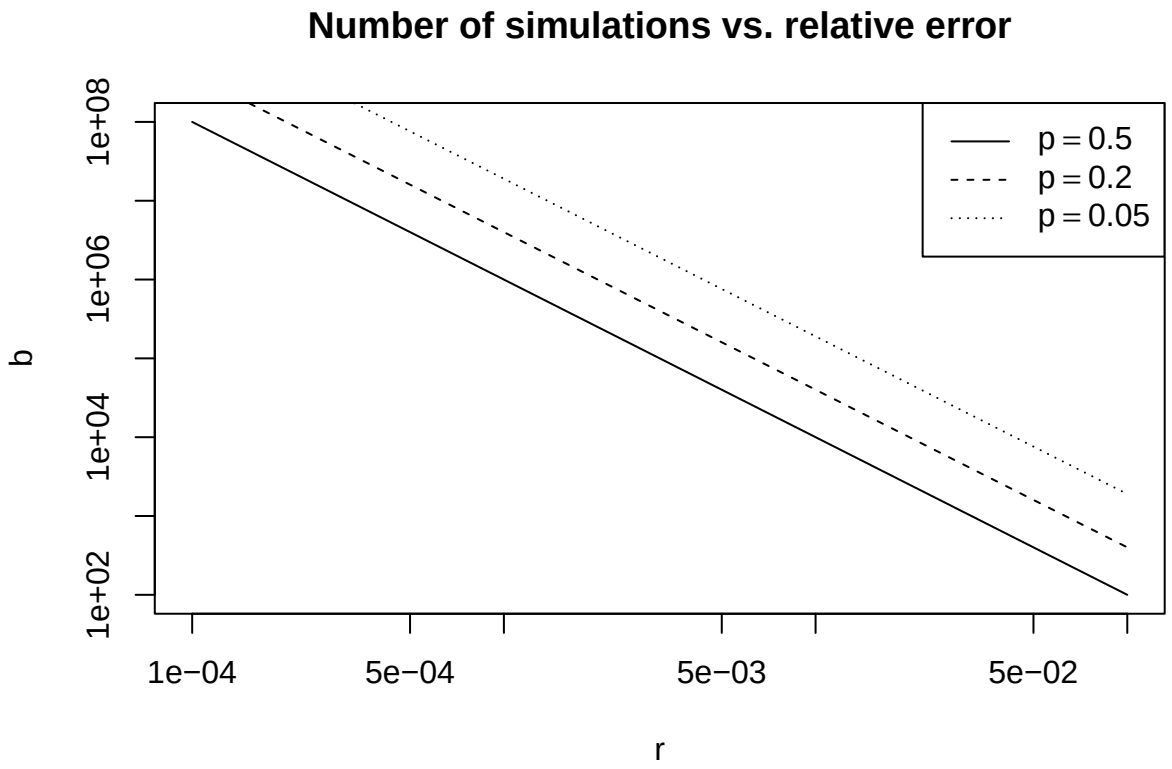


Figure 3: Number of simulations b required to guarantee the *relative* error of the approximate probability $\hat{p}$ is at most r, as a function of the true probability p. (Unlike the standard error, this is *not* symmetric around $p = 1/2$, and probabilities closer to 1 would require even smaller values of b.) Note the log scale on both axes.

2.6 Simulation vs. Numerical Integration (more seriously this time)

The probability of any region A is the integral of the probability density function $p(x)$ over A, $\mathbb{P}[X \in A] = \int_A p(x)dx$. When A is just an interval, $A = [a_1, a_2]$, this is the definite integral $\int_{a_1}^{a_2} p(x)dx$. There are, and long have been, extremely well-established and reliable techniques for numerically evaluating integrals[3]. It's only reasonable to ask how well they compare to the simulation-based approximation I've been describing.

The truth is that if all you need is $(2\pi)^{-1/2}\int_{a_1}^{a_2} e^{-x^2/2}dx$, you're almost certainly better off using a good numerical integration routine rather than simulating. The error in the simulation approximation, as we've just seen, goes down like $1/\sqrt{b}$, so it can require a *lot* of simulations to get an accurate approximation. The simulation method shows its advantages in four circumstances:

1. *Ugly integrands*: If the distribution we're interested in is *not* as nice and smooth as $e^{-x^2/2}$, numerical integrators will usually need to evaluate it at lots of points. The error of the simulation approach, however, will remain $\sqrt{\frac{p(1-p)}{b}}$. This can reduce, or even reverse, the computational advantages of numerical integration.
2. *Ugly domains*: If the domain of integration isn't just a simple interval (or box, in higher dimensions), but something complicated, because we're looking at a complicated event, again, numerical integrators will require lots and lots of computational work. But the error of the simulation approximation is (still) just $\sqrt{\frac{p(1-p)}{b}}$. So for getting the probabilities of complicated events, the simulation approach is better.
3. *Higher dimensions*: If X isn't one dimension but (say) d dimensional, numerical integration will require a computational effort that typically grows exponentially in d. But (to repeat) the error of the simulation approach is $\sqrt{\frac{p(1-p)}{b}}$, *independent of the dimension d*. So for high-dimensional distributions, the simulation approach is better.
4. *Implicitly defined probabilities*: A lot of the time in scientific practice, the probability distribution we really care about isn't anything that's given explicitly in terms of a formula. Rather, it's given *implicitly*, in terms of a complicated scientific model of some process. If this is a **generative** model, then we can "run it forward" to get simulated data drawn from this distribution, but good luck actually writing out the probability density[4]. This is a situation where numerical integration is just hopeless, but approximation by repeated simulation works just fine.

3 Approximating Expectation Values

Recall that the expectations of functions are also integrals:

$$\mathbb{E}[f(X)] = \int f(x)p(x)dx \tag{26}$$

We can approximate expectation values in exactly the same way we approximated probabilities, by

1. Simulating independent runs $X_1, X_2, \ldots X_b$, and
2. Applying the function f to each of them, getting $f(X_1), f(X_2), \ldots f(X_b)$, and then
3. Averaging.

That is,

$$\widehat{\mathbb{E}[f(X)]} = \frac{1}{b}\sum_{t=1}^{b} f(X_t) \tag{27}$$

Just as with approximating probabilities:

[3] I won't explain here how numerical integrators work, but they're covered in any good book on scientific computing or numerical analysis, e.g., Press et al. (1992).

[4] Lots of weather and climate models are like this, but situations like this also arise in branches of genetics, neuroscience, economics, geology, and I dare say other fields I'm not aware of.

- $\widehat{\mathbb{E}[f(X)]}$ is an unbiased approximation to $\mathbb{E}[f(X)]$
- $\widehat{\mathbb{E}[f(X)]}$ is a consistent approximation to $\mathbb{E}[f(X)]$, so $\widehat{\mathbb{E}[f(X)]} \to \mathbb{E}[f(X)]$ as $b \to \infty$
- The standard error of the approximation is $\sqrt{\mathrm{Var}[f(X)]/b}$
  - We can consistently approximate $\mathrm{Var}[f(X)]$ by the sample variance of the $f(X_t)$
- For large b, $\widehat{\mathbb{E}[f(X)]} \rightsquigarrow \mathcal{N}(\mathbb{E}[f(X)], \mathrm{Var}[f(X)]/b)$.
  - If that expression makes you uncomfortable, you know enough to re-write it so the limiting distribution on the right-hand side is constant in b.

The arguments for all of these points are exactly parallel to the ones for approximating probabilities, so I won't rehearse them here.

4 Dependent Simulations

I've worked out everything above assuming that each run is a statistically independent draw from the distribution we want. There are, however, many situations where it's easier to generate *dependent* draws from the same complex distribution. They're identically-distributed, but not independent[5]. This can still be analyzed in a *very* similar manner.

For simplicity, let's pretend that $X_1, X_2, \ldots X_b$ are all one-dimensional. (Often they're big complicated objects, but what we're really interested in is $Y_t = f(X_t)$ and the function f maps down to one dimension.) We'll say that they all have the same distribution, so that in particular they share a common expectation $\mathbb{E}[X]$ and variance $\mathrm{Var}[X]$. We'll also assume that they are (as the jargon says) **weakly stationary**, or **second-order stationary**, so that $\mathrm{Cov}[X_t, X_{t+h}] = \mathrm{Var}[X]\rho(h)$ for some **autocorrelation function**[6] ρ. Clearly $\rho(0) = 1$, $\rho(h) = \rho(-h)$, and in general $|\rho(h)| \leq 1$. In the independent-runs case we've been thinking about, $\rho(0) = 1$, $\rho(h) = 0$ for $h \geq 1$. Following our previous notation, we'll say

$$\widehat{\mathbb{E}[X]} = \frac{1}{b}\sum_{t=1}^{b} X_t \tag{28}$$

As a final bit of notation, we'll decree that the **integrated autocorrelation time**[7] τ is

$$\tau = \sum_{h=-\infty}^{\infty} \rho(h) \tag{29}$$

with the understanding that τ might be infinite. (For the independent-runs case, $\tau = 1$.)

Now we're in business:

1. $\widehat{\mathbb{E}[X]}$ is an unbiased approximation to $\mathbb{E}[X]$. (This follows from linearity of expectations.)
2. If $\tau < \infty$, then as $b \to \infty$,
$$\mathrm{Var}\left[\widehat{\mathbb{E}[X]}\right] \approx \frac{\mathrm{Var}[X]}{b/\tau} \tag{30}$$
   - One shows[8] this by, essentially,repeatedly applying the rule $\mathrm{Var}[U+W] = \mathrm{Var}[U] + \mathrm{Var}[W] + 2\mathrm{Cov}[U,W]$ to the definition of $\widehat{\mathbb{E}[X]}$, and then noticing that for large b, the resulting sum-of-covariances can be approximated in terms of τ.

[5] I won't go into what's called "Markov chain Monte Carlo" (MCMC) here, but if you're really curious, look at [https://www.stat.cmu.edu/~cshalizi/statcomp/13/lectures/15/markov.pdf] and [https://www.stat.cmu.edu/~cshalizi/statcomp/13/lectures/16/mcmc.pdf] from the 2013 statistical computing course.

[6] The correlation coefficient between any two random variables U and W is $\frac{\mathrm{Cov}[U,W]}{\sqrt{\mathrm{Var}[U]\mathrm{Var}[W]}}$. The correlation between X_t and X_{t+h} is therefore $\frac{\mathrm{Var}[X]\rho(h)}{\sqrt{\mathrm{Var}[X]\mathrm{Var}[X]}} = \rho(h)$, hence the name "autocorrelation function".

[7] Some people define the integrated autocorrelation time as $\tau' = \sum_{h=1}^{\infty}\rho(h)$. Since $\tau = 1 + 2\tau'$, the two definitions are equivalent, but this one leads to slightly simpler formulas. Just be careful you understand which convention an author is using. Also, some alternative names include "autocorrelation time", "covariance time" and "correlation time".

[8] For an elementary proof of a slightly more general result, see Shalizi (2022).

- Notice that this looks like what we had with the independent-runs case, only with a smaller number of effectively-independent runs, b/τ instead of b.

3. If $\tau < \infty$, then $\widehat{\mathbb{E}[X]}$ is a consistent approximation to $\mathbb{E}[X]$.
   - This follows from unbiasedness and the bound on the variance we just saw.
   - This is a sufficient condition for consistency, not a necessary one. (The necessary-and-sufficient condition is complicated.)
4. If $\tau < \infty$ and some other, more technical conditions also hold, then $\widehat{\mathbb{E}[X]}$ has Gaussian fluctuations around $\mathbb{E}[X]$, with the variance from item (2) above.
   - The extra conditions are *really* unilluminating at this level.

In short, basically everything that's true of the independent-runs case is still true if we only have dependent runs but $\tau < \infty$. This is because, under that assumption, b correlated simulations are equivalent to b/τ uncorrelated ones.

Exponential decay of correlations: Suppose that $\rho(h) = e^{-h/T}$. Show that τ is finite, and express it in terms of T. (In this situation, some people call T itself the "correlation time".)

5 "Monte Carlo"

What I've been calling "approximating by repeated simulation" is usually known, in the jargon, as "Monte Carlo approximation". I've been deliberately avoiding that phrase so far, because I find it tends to confuse newcomers[9]. But it *is* an established part of the jargon of the mathematical world, and it's worth taking a minute or two in order to appreciate where the technique, and the phrase, came from.

There's a long history of *occassionally* using simulations to approximate integrals. The most famous early example was the "Buffon's needle" problem in the 18th century, with a handful of 19th century successors. These were all isolated efforts, "one-offs", of no real influence, and only retrospectively seem to be important or part of something bigger[10].

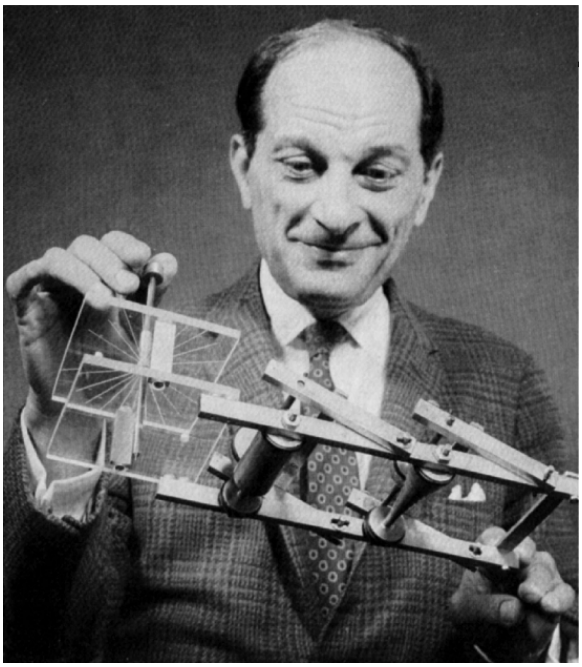
Modern Monte Carlo goes back, basically, to Los Alamos and the building of the atomic and the hydrogen bombs in the 1940s — or rather just a little before. In 1935, the great physicist Enrico Fermi started simulating neutron-nuclei interactions with pencil, paper, and random number tables (Schwartz 2017, 124). Moreover, he realized that what he was doing was a (potentially) very general way of evaluating complex integrals under complex distributions. In the early 1940s, while working on the Manhattan Project, Fermi needed to calculate the behavior of chain reactions in different potential designs for atom bombs (cf. Serber (1992)), so he turned to his simulation technique, designing and building a mechanical simulator to trace over 2D blueprints of bombs. In 1946, the mathematician Stanislaw Ulam realized that you could do simulations on the new digital "electronic computing machines". He developed the idea with polymath John von Neumann under the code name of "the Monte Carlo method" (which was suggested by Nicholas Metropolis[11]). During the period 1946–1953, the Monte Carlo method was intensively developed by Los Alamos scientists, culminating in Metropolis et al. (1953), which gave a general method for doing Monte Carlo approximations for potentially *extremely* complicated distributions, and demonstrated its feasibility with 1953-vintage computing hardware.[12]

[9] Also, it's at least somewhat more common to say "Monte Carlo *estimation*" rather than "Monte Carlo *approximation*", but when we statisticians say "estimation" we usually mean something we do to *data*, and here everything is based on simulations. So I am sticking to "approximation".

[10] This is a very common pattern in the history of science, and the history of ideas more generally. ("Everything was first said by someone else, who meant something very different.")

[11] After the famous casino at Monte Carlo in Monaco on the French Riviera.

[12] I suspect there's a professional history-of-science study of the birth of Monte Carlo, but if so I haven't found it. There are many biographies of von Neumann, who was one of the most important scientists and mathematicians of the 20th century; all of those books cover this episode in more or less detail. (I am fond of Poundstone (1992) and have not yet had a chance to read the latest, Bhattacharya (2021).) There doesn't seem to be a biography of Ulam, but his memoir (Ulam 1976) is extremely entertaining.



Stanislaw Ulam, shown holding the "Fermiac" mechanical simulator (via).

From that point onward, the Monte Carlo method spread outward, partly through publications but still more from scientists who worked at, or visited, Los Alamos and then later RAND (which has a lot of Los Alamos "alumni"), etc. At first it was more extensively used in physics and chemistry than in statistics, but it *was* showing up in statistics by the 1960s (e.g., Bartlett (1960), or the once-influential Hammersley and Handscomb (1964)). Statisticians only really embraced it in the 1980s, however, with the arrival of desktop personal computers.

6 Further Reading

See the references at the end of chapter 5 ("Simulation") in the textbook.

References

Bartlett, M. S. 1960. *Stochastic Population Models in Ecology and Epidemiology*. London: Methuen.

Bhattacharya, Ananyo. 2021. *The Man from the Future: The Visionary Life of John von Neumann*. New York: W. W. Norton.

Hammersley, J. M., and D. C. Handscomb. 1964. *Monte Carlo Methods*. London: Chapman; Hall.

Metropolis, Nicholas, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. "Equations of State Calculations by Fast Computing Machines." *Journal of Chemical Physics* 21:1087–92. https://doi.org/10.1063/1.1699114.

Poundstone, William. 1992. *Prisoner's Dilemma*. New York: Doubleday.

Press, William H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 1992. *Numerical Recipes in C: The Art of Scientific Computing*. 2nd ed. Cambridge, England: Cambridge University Press. http://www.nrbook.com/.

Schwartz, David N. 2017. *The Last Man Who Knew Everything: The Life and Times of Enrico Fermi, Father of the Nuclear Age*. New York: Basic Books.

Serber, Robert. 1992. *The Los Alamos Primer: The First Lectures on How to Build the Atomic Bomb*. Berkeley: University of California Press.

Shalizi, Cosma Rohilla. 2022. "A Simple Non-Stationary Mean Ergodic Theorem, with Bonus Weak Law of Large Numbers." arxiv:2203.09085. https://arxiv.org/abs/2203.09085.

Ulam, Stanislaw M. 1976. *Adventures of a Mathematician*. New York: Scribner.