MA3227 Numerical Analysis II

Lecture 23: Simulation of Random Variables

Simon Etter

2019/2020

Simulation of Random Variables

Introduction

In Lecture 22, we introduced random variables $X \sim \mathcal{X}$ as functions $X : \Omega \to \Xi$ defined on some unspecified probability space (Ω, P) such that

$$P(X \in A) = P\big(X^{-1}(A)\big) = \mathcal{X}(A) \quad \text{for all } A \subset \Xi. \tag{1}$$

In order to work with such random variables, we hence need two things:

- A probability space (Ω, P).
- A function $X(\omega)$ such that (1) is satisfied.

On a computer, the probability space (Ω, P) is almost always given by $\Omega = [0,1]^n$ and $P(A) = \text{volume}(A)$. The reason for this is that uniformly distributed $\omega_k \in [0,1]$ can be easily generated by generating a string of bits (e.g. 10110) where each bit is equally likely to be 0 or 1, and then mapping these strings onto equally-spaced points in $[0,1]$. This process is known as *random number generation*, and I will provide a bit more detail on the next slide.

Once we have a sequence of uniformly distributed numbers $\omega_k \in [0,1]$, the next task is find a function $X\big((\omega_k)\big) \to \Xi$ such that (1) is satisfied. This is called *simulation* or *sampling* of random variables, and the main aim of this lecture is to introduce several techniques for doing so.

Simulation of Random Variables

Def: Random number generator (RNG)

Any algorithm / piece of hardware which produces a sequence $u_k \in [0,1]$ which looks as if the u_k were independent samples from a random variable $U \sim \text{Uniform}[0,1]$.

Remark

For the most part, the u_k produced by RNGs will play the role of the ω_k on the previous slide. Nevertheless, it is common practice to write u_k instead of ω_k to emphasise that the u_k are uniformly distributed in $[0,1]$.

Discussion of random number generators

There is no rigorous definition of "u_k which look as if they were independent samples of $U \sim \text{Uniform}[0,1]$". Instead, there are long lists of tests which you can use to measure how close your RNG is to producing "true" samples of $U \sim \text{Uniform}[0,1]$.
See e.g. `https://en.wikipedia.org/wiki/Diehard_tests`.

Simulation of Random Variables

Discussion of random number generators (continued)

RNGs come in two varieties:

- "True" RNGs use noise in your hardware to produce truly unpredictable u_k.
- Pseudo RNGs (pRNGs) take in a seed s and return a sequence u_k which is fully deterministic but which looks like a sequence of independent samples of $U \sim \text{Uniform}[0, 1]$.

True RNG are important for applications like cryptography, but for Monte Carlo purposes they have two important drawbacks:

- They are much slower than pRNGs, see `rng_benchmark()`.
- They by definition make it impossible to reproduce results, which is a nuisance when you want to debug your code.

For these reasons, we will exclusively consider pRNGs in this module.

Designing a fast and high-quality pRNG is highly non-trivial. Luckily, you will almost surely never have to do this yourself since most programming languages come with a pre-installed pRNG.

Simulation of Random Variables

pRNGs in Julia (and most other programming languages)

The pRNG functionality in Julia is provided by the `rand()` function. This function implicitly defines a sequence u_k and keeps an index k pointing to the current element. Each call to `rand()` returns the current u_k and then increments $k \leftarrow k + 1$.

The state of the pRNG can be reset using `Random.seed!()`.

Example

```
julia> Random.seed!(42);
julia> rand()
0.5331830160438613
julia> rand()
0.4540291355871424
julia> Random.seed!(42);
julia> rand()
0.5331830160438613
```

Note that the argument to `Random.seed!()` is not the index k.

```
julia> Random.seed!(43); rand()
  0.18097523182192754
```

(not `0.4540291355871424`)

Simulation of Random Variables

Discussion

The above concludes our discussion of random number generation.
We now move on to simulation of random variables, i.e. the problem of finding $X : [0,1]^n \to \Xi$ such that

$$P(X \in A) = P\big(X^{-1}(A)\big) = \mathcal{X}(A) \quad \text{for all } A \subset \Xi.$$

Let us begin on with a simple example.

Simulation of Random Variables

Example

Task: Given $U \sim \text{Uniform}[0,1]$, construct $X(U) \sim \text{Uniform}[a,b]$.

Uniform$[a,b]$ is the uniform distribution on the interval $[a,b]$, i.e. we want $P(X \in [c,d]) = \frac{d-c}{b-a}$ for all c, d such that $a \leq c \leq d \leq b$.

Solution: A simple solution is

$$X(U) = a + (b-a)\,U \qquad \Longleftrightarrow \qquad X^{-1}(x) = \tfrac{x-a}{b-a}$$

since then

$$\begin{aligned} P(X(U) \in [c,d]) &= P(U \in X^{-1}[c,d]) \\ &= P(U \in [\tfrac{c-a}{b-a}, \tfrac{d-a}{b-a}]) \\ &= \tfrac{d-a}{b-a} - \tfrac{c-a}{b-a} \\ &= \tfrac{d-c}{b-a}. \end{aligned}$$

Alternatively, we could set $X(U) = b + (a-b)\,U$, or we could construct $X(U)$ by piecing together several linear functions whose ranges partition $[a,b]$, etc.

Simulation of Random Variables

Discussion

Constructing a random variable X with a desired target distribution $\mathcal{F}$ was easy in the above example because $\mathcal{F} = \text{Uniform}[a, b]$ was just a linear transformation of the initial distribution $\text{Uniform}[0, 1]$.

In general, finding $X(U)$ such that $U \sim \text{Uniform}[0, 1] \implies X \sim \mathcal{F}$ can be quite difficult. The following slides present two strategies for constructing random variables which work for fairly general distributions $\mathcal{F}$ but which may not be very efficient.

Remember that there are many ways how we can construct $X(U)$ such that $X \sim \mathcal{F}$. Of course, it is well possible that some $X(U)$ are easier to evaluate than others.

Simulation of Random Variables

Thm: Transformation sampling

Let $\mathcal{F}$ be a distribution on $\mathbb{R}$ with cumulative distribution function $F(x)$, and assume $U \sim \text{Uniform}[0, 1]$. Then,

$$X = F^{-1}(U) \sim \mathcal{F}.$$

Proof. We have

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

where in the second step I used the monotonicity of $F(x)$ and in the third step I used $U \sim \text{Uniform}[0, 1]$.

Example

Consider the distribution $\mathcal{F}$ with density function $f(x) = 2x$ on $[0, 1]$ and cumulative distribution function

$$F(x) = \int_0^x 2x'\, dx' = x^2 \qquad \text{for } x \in [0, 1].$$

If $U \sim \text{Uniform}[0, 1]$, then $X = \sqrt{U} \sim \mathcal{F}$.

See `transformation_sampling()`.

Simulation of Random Variables

Pros and cons of transformation sampling

- Pro: Easy to implement and fast if $F^{-1}(u)$ can be easily computed.
- Con: Only works for random variables $\Omega \to \mathbb{R}$.
- Con: $F^{-1}(u)$ may not be easy to compute.

To illustrate the last point, consider the normal distribution $\mathcal{N}(\mu, \sigma^2)$. There is no known direct formula for the CDF

$$F(x) = \tfrac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{x} \exp\left(-\tfrac{(x-\mu)^2}{2\sigma^2}\right) dx;$$

hence the only way to compute $F^{-1}(u)$ is to evaluate the above integral using quadrature and then apply a root finder to determine $x \in \mathbb{R}$ such that $F(x) = u$.

Simulation of Random Variables

Thm: Rejection sampling

Let $\mathcal{F}, \mathcal{G}$ be two distributions on $\mathbb{R}^n$ with probability density functions $f(x)$ and $g(x)$, respectively, and let

$$U_k \sim \text{Uniform}[0,1] \qquad \text{and} \qquad G_k \sim \mathcal{G} \qquad \text{be independent.}$$

Assume there exists $M > 0$ such that

$$f(x) \leq M\, g(x) \qquad \text{for all } x \in \mathbb{R}^n.$$

Consider the random variable F defined through the following algorithm.

Algorithm 1 Rejection sampling

```
for k = 1, 2, ... do
    if U_k(ω) ≤ f(G_k(ω)) / (M g(G_k(ω))) then
        Return F(ω) = G_k(ω)
    end if
end for
```

We then have $F \sim \mathcal{F}$.

Simulation of Random Variables

Proof (not examinable). Let us introduce the abbreviation

$$C_k(\omega) = \begin{cases} 1 & \text{if } U_k(\omega) \le \frac{f(G_k(\omega))}{M\, g(G_k(\omega))}, \\ 0 & \text{otherwise.} \end{cases}$$

Using the law of total probability, we obtain
(`https://en.wikipedia.org/wiki/Law_of_total_probability`)

$$\begin{aligned} P(C_k = 0) &= \int_{\mathbb{R}^n} g(x)\, P(C_k = 0 \mid G_k = x)\, dx \\ &= \int_{\mathbb{R}^n} g(x) \left(1 - \frac{f(x)}{M\, g(x)}\right) dx \\ &= \int_{\mathbb{R}^n} g(x)\, dx - \frac{1}{M} \int_{\mathbb{R}^n} f(x)\, dx \\ &= 1 - \frac{1}{M}. \end{aligned}$$

Simulation of Random Variables

Proof (not examinable, continued).

Using that

- $P(A \cup B) = P(A) + P(B)$ if A, B are disjoint, and
- $p(x, y) = p_X(x)\, p_Y(y)$ if $p(X, Y), p_X(x), p_Y(y)$ are the PDFs of two independent random variables X, Y, respectively,

we obtain

$$\begin{aligned}
P(F \in A) &= P(G_1 \in A, C_1 = 1) + P(G_2 \in A, C_1 = 0, C_2 = 1) + \dots \\
&= \int_A g(x)\, \tfrac{f(x)}{M\, g(x)}\, dx + (1 - \tfrac{1}{M}) \int_A g(x)\, \tfrac{f(x)}{M\, g(x)}\, dx + \dots \\
&= \tfrac{1}{M} \int_A f(x)\, dx \left(\sum_{k=0}^{\infty} (1 - \tfrac{1}{M})^k \right) \\
&= \tfrac{1}{M} \int_A f(x)\, dx\, \frac{1}{1 - (1 - \frac{1}{M})} \\
&= \int_A f(x)\, dx,
\end{aligned}$$

i.e. $F \sim \mathcal{F}$ as claimed.

Simulation of Random Variables

Example

Consider again the distribution $\mathcal{F}$ with density function $f(x) = 2x$ on $[0,1]$, and set $\mathcal{G} = \text{Uniform}[0,1]$ with density function $g(x) = 1$.

We have $f(x) \leq 2\, g(x)$, i.e. $M = 2$ in the notation of the rejection sampling theorem. We can hence generate a sample f according to $\mathcal{F}$ by generating samples g_k according to $\mathcal{G}$ and u_k according to Uniform$[0,1]$, and setting $f = g_k$ where k is the smallest integer such that

$$u_k \leq \frac{f(g_k)}{M\, g(g_k)} = \frac{2g_k}{2 \times 1} = g_k.$$

See `rejection_sampling()` for numerical demonstration.

Moreover, we have seen in the proof of the rejection sampling theorem that the probability for accepting a proposal g_k is $P(C_k = 1) = \frac{1}{M}$. Hence,

$$\mathbb{E}[\text{number of tries until accepted}] = \tfrac{1}{P(C_k=1)} = M$$

This is the expectation value of a geometrically distributed random variable with success probability $P(C_k = 1) = \frac{1}{M}$.
See `https://en.wikipedia.org/wiki/Geometric_distribution`.

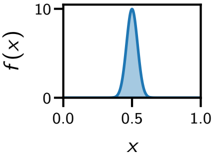
Simulation of Random Variables

Pros and cons of rejection sampling

- Pro: Works for fairly general distributions $\mathcal{F}$. All we need is a sampleable proposal distribution $\mathcal{G}$ such that $M = \sup_x \frac{f(x)}{g(x)} < \infty$.
- Con: Can be very inefficient: we have seen that $M = \sup_x \frac{f(x)}{g(x)}$ is the expected number of samples of $\mathcal{G}$ required to generate a single sample of $\mathcal{F}$; thus if M is very large, we generate and discard many samples of $\mathcal{G}$ to generate a single sample of $\mathcal{F}$.

Example

Consider the distribution



If we choose $\mathcal{G} = \text{Uniform}[0,1]$, we need $M = 10$ samples of $\mathcal{G}$ to generate a single sample of $\mathcal{F}$.

Simulation of Random Variables

Discussion

The goal of the transformation and rejection sampling is to generate samples of some random variable F which we cannot sample otherwise.

However, the aim of Monte Carlo algorithms is to compute $\mathbb{E}[F]$, and sampling F is only a means towards this end. The following result shows that it is possible to compute $\mathbb{E}[F]$ even if we can only sample some other random variable G.

Thm: Importance sampling

Let $\mathcal{F}, \mathcal{G}$ be distributions on $\mathbb{R}^n$ with probability densities $f(x)$ and $g(x)$, respectively, and let $F \sim \mathcal{F}$, $G \sim \mathcal{G}$. Furthermore, assume

$$x\, f(x) \neq 0 \quad \implies \quad g(x) \neq 0.$$

Then,

$$\mathbb{E}[F] = \mathbb{E}[G\, \tfrac{f(G)}{g(G)}].$$

Proof.

$$\mathbb{E}[F] = \int_{\mathbb{R}^n} x\, f(x)\, dx = \int_{\mathbb{R}^n} x\, \tfrac{f(x)}{g(x)}\, g(x)\, dx = \mathbb{E}\Big[G\, \tfrac{f(G)}{g(G)}\Big].$$

Simulation of Random Variables

Example

Consider again the distribution $\mathcal{F}$ with density function $f(x) = 2x$ on $[0, 1]$, and set $\mathcal{G} = \text{Uniform}[0, 1]$ with density function $g(x) = 1$. Assuming $F \sim \mathcal{F}$ and $G \sim \mathcal{G}$, we then have

$$\mathbb{E}[F] = \mathbb{E}[G \, \tfrac{f(G)}{g(G)}] = \mathbb{E}[G \, \tfrac{2G}{1}] = \mathbb{E}[2G^2].$$

This is easily confirmed analytically,

$$\mathbb{E}[F] = \int_0^1 x \, 2x \, dx = \tfrac{3}{2} = \int_0^1 2x^2 \, dx = \mathbb{E}[2G],$$

and demonstrated numerically in `importance_sampling()`.

Simulation of Random Variables

Discussion

Recall from Lecture 22 that

$$\mathbb{E}\Big[\tilde{\mathbb{E}}_N[F] - \mathbb{E}[F]\Big] = \sqrt{\tfrac{1}{N}\,\mathrm{Var}[F]}.$$

After applying the importance sampling trick, we hence obtain

$$\mathbb{E}\Big[\tilde{\mathbb{E}}_N[G\,\tfrac{f(G)}{g(G)}] - \mathbb{E}[F]\Big] = \sqrt{\tfrac{1}{N}\,\mathrm{Var}[G\,\tfrac{f(G)}{g(G)}]},$$

which shows that the error becomes larger if we choose $\mathcal{G}$ such that

$$\mathrm{Var}[G\,\tfrac{f(G)}{g(G)}] > \mathrm{Var}[F].$$

Surprisingly, it is sometimes also possible to reduce the variance using the importance sampling trick as demonstrated in the example on the next slide.

Simulation of Random Variables

Example

Consider the random variables $F \sim \text{Uniform}[0,1]$ and $G \sim \mathcal{G}$ where $\mathcal{G}$ has probability density $g(x) = 2x$ on $[0,1]$.

We then have

$$\text{Var}[F] = \mathbb{E}[F^2] - \mathbb{E}[F]^2 = \int_0^1 x^2\,dx - \left(\int_0^1 x\,dx\right)^2 = \tfrac{1}{3} - \tfrac{1}{4} = \tfrac{1}{12}$$

but

$$\text{Var}[G\,\tfrac{f(G)}{g(G)}] = \text{Var}[G\,\tfrac{1}{2G}] = \text{Var}[\tfrac{1}{2}] = 0,$$

that is

$$\mathbb{E}\Big[\tilde{\mathbb{E}}_N[F] - \mathbb{E}[F]\Big] = \sqrt{\tfrac{1}{12N}} \qquad \text{but} \qquad \mathbb{E}\Big[\tilde{\mathbb{E}}_N[G\,\tfrac{f(G)}{g(G)}] - \mathbb{E}[F]\Big] = 0.$$

Simulation of Random Variables

Remark

Throughout this lecture, we focused on constructing a single random variable $X : \Omega \to \Xi$ such that $X \sim \mathcal{X}$. However, the Monte Carlo estimator

$$\tilde{\mathbb{E}}_N[X] = \tfrac{1}{N} \sum_{k=1}^{N} X_k$$

requires a sequence $X_1, \dots X_N \overset{\text{iid}}{\sim} \mathcal{X}$ of such random variables. Such a sequence can be easily constructed using the following result.

Thm: Sequence of iid random variables

Assume $X : \Omega \to \Xi$ is a random variable with distribution $X \sim \mathcal{X}$. Then, the sequence of random variables

$$X_k : \quad \Omega^N \to \Xi, \quad (\omega_1, \dots, \omega_N) \mapsto X(\omega_k)$$

satisfies $X_1, \dots X_N \overset{\text{iid}}{\sim} \mathcal{X}$, assuming the probability measure on Ω^N is defined through

$$P\big(A_1 \times \ldots \times A_N\big) = P(A_1) \, \ldots \, P(A_N).$$

Simulation of Random Variables

Proof. We compute

$$\begin{aligned}
P(X_1 \in A_1) &= P\big(X_1^{-1}(A_1)\big) \\
&= P\big(X^{-1}(A_1) \times \underbrace{\Omega \times \ldots \times \Omega}_{N-1 \text{ times}}\big) \\
&= P\big(X^{-1}(A_1)\big) \times \underbrace{1 \times \ldots \times 1}_{N-1 \text{ times}} \\
&= \mathcal{X}(A_1)
\end{aligned}$$

and hence conclude that $X_1 \sim \mathcal{X}$. Showing $X_k \sim \mathcal{X}$ for all other k can be done analogously.

We further have

$$\begin{aligned}
P(X_1 \in A_1, \ldots, X_N \in A_N) &= P\big(X_1^{-1}(A_1) \cap \ldots \cap X_N^{-1}(A_N)\big) \\
&= P\big(X_1^{-1}(A_1) \times \ldots \times X_N^{-1}(A_N)\big) \\
&= P\big(X_1^{-1}(A_1)\big) \times \ldots \times P\big(X_N^{-1}(A_N)\big),
\end{aligned}$$

which shows that $X_1, \ldots, X_N$ are independent.

Simulation of Random Variables

Discussion

The practical implication of the above theorem is as follows.

Assume we have a function `randX() -> x` which generates a sample `x` of a random variable $X \sim \mathcal{X}$ by making one or more calls to `rand()` and then transforming the resulting sample $\mathtt{u} \in [0,1]^n$ into $\mathtt{x} = X(\mathtt{u})$.

Since a sequence of samples from the underlying pRNG are assumed to be independent, it follows that we can think of the result of N calls to `randX()` as a single sample of the sequence of random variables $X_1, \ldots X_N \overset{\text{iid}}{\sim} \mathcal{X}$.

Simulation of Random Variables

Summary

- Pseudo random number generator (pRNG): sequence $u_k \in [0, 1]$ such that u_k "looks like samples of $U \sim \text{Uniform}[0, 1]$.
- Transformation sampling: $X = F^{-1}(U)$ with $U \sim \text{Uniform}[0, 1]$ is distributed according to the CDF $F(x)$.
- Rejection sampling: propose samples according to a proposal distribution $\mathcal{G}$ and then reject with probability $\frac{f(x)}{M\,g(x)}$ to produce samples according to $\mathcal{F}$.
- Importance sampling: $\mathbb{E}[F] = \mathbb{E}[G\,\frac{f(G)}{g(G)}]$.