Chapter 6

Random Variable Generation

Definition 6.1 (Informal)**.** *A* **uniform pseudorandom number generator** *(UPRNG) is an algorithm which starting from an initial value u_0 and a transformation D, produces a sequence $u_i = D(u_{i-1})$ in $[0,1]$ for $i = 1, \ldots$. For all n, $u_1, \ldots, u_n$ approximate the behavior of an i.i.d. sequence of* Uniform$([0,1])$ *random numbers.*

We could provide a mathematical definition of a UPRNG.

Definition 6.2 (Formal)**.** *Let $u_0 \in [0,1]$ and let $D : [0,1] \to [0,1]$, define the dynamical system*

$$u_i = D(u_{i-1}), \quad i = 1, 2, \ldots$$

For a set $A \subset [0,1]$, define $N_n(A)$ as the number of $u_i \in A$ for $i = 0, 1, 2, \ldots, n-1$. We call D a UPRNG if and only if for every $u_0 \in [0,1]$ and every Borel set $A \subset [0,1]$

$$\frac{N_n(A)}{n} \to \int_A dx.$$

In words, no matter the starting point u_0 the long term relative frequency of the event $u_i \in A$ approaches the probability of that event for a uniform random RV as $n \to \infty$.

But before we get to the UPRNG let us define a pseudorandom sequence

Definition 6.3 (pseudorandom)**.** *Consider the finite set $\mathcal{M} = \{0, 1, \ldots, M-1\}$ and consider the sequence $u_0, u_1, \ldots \in \mathcal{M}$. For every $a \in \mathcal{M}$, define $N_n(a)$ as the number of $u_i = a$ for $i = 0, 1, 2, \ldots, n-1$. We call the sequence $u_0, u_1, \ldots$* **pseudorandom** *on $\mathcal{M}$ if and only if for every $a \in \mathcal{M}$*

$$\frac{N_n(a)}{n} \to \frac{1}{M}.$$

6.1 Congruential Generators

Definition 6.4. *Let u_0 be fixed and let D be a map, define the dynamical system*

$$u_i = D(u_{i-1}), \quad i = 1, \ldots$$

We call T_0 the period of D started at u_0 the smallest positive integer such that

$$u_{i+T_0} = u_i, \text{ for some } i.$$

The smallest period T for all admissible starting points u_0 is called the period for D.

Exercise 6.5. *If we start at a fixed point u_0, and let T_0 be the period of D w.r.t u_0, then if*

$$u_{i+T_0} = u_i$$

holds for some i, then it holds for all i.

Definition 6.6. *A* **congruential generator** *with parameters (a, b, M) on $\{0, 1, \ldots, M-1\}$ is defined by the function*

$$D(x) = (ax + b) \mod M.$$

Example 6.7. *The congruential generator $(3, 0, 16)$ on $\{0, 1, \ldots, 16\}$, has many different periods. For instance if $u_0 = 0$ then the period for 0 is 1. If we instead start at 1 then the period is 4. If we start at 2 the period is 2. etc.*

Is it possible for a congruential generator to generate something pseudo-random?

Lemma 6.8. *Consider a congruential generator D on $\mathcal{M} = \{0, 1, \ldots, M-1\}$ with period M, then for any starting point $u_0 \in \mathcal{M}$, the sequence $u_i = D(u_{i-1})$ is pseudorandom on $\mathcal{M}$.*

Exercise 6.9. *Prove the above lemma.*

The problem with a congruential generator on $\mathcal{M}$ is that the period is as long as the number of unique values, this will be problematic if M is small. What we can do is to use the congruential generator for a larger set and restrict it to a smaller to get a better generator.

Lemma 6.10. *Consider a congruential generator D on $\mathcal{M} = \{0, 1, \ldots, M-1\}$ with period M, then for any starting point $u_0 \in \mathcal{M}$, define $u_i = D(u_{i-1})$ then the sequence $v_i = u_i \mod K$ for $1 \leq K \leq M$ is pseudorandom on $\{0, 1, \ldots, K-1\}$ if M is a multiple of K.*

Exercise 6.11. *Prove the above lemma.*

One issue with the method in the above lemma, is that the period is K for v_i. Thus we have essentially thrown away the good thing about our congruential generator, i.e. the long period. This can be fixed by instead of taking the modulus we divide:

Lemma 6.12. *Consider a congruential generator D on $\mathcal{M} = \{0, 1, \ldots, M-1\}$ with period M, then for any starting point $u_0 \in \mathcal{M}$, define $u_i = D(u_{i-1})$ then the sequence $v_i = \lfloor (u_i/M) * K \rfloor$ for $1 \leq K \leq M$ is pseudorandom on $\mathcal{K} = \{0, 1, \ldots, K-1\}$ if M is a multiple of K, moreover the period of v_i is M.*

Proof. Consider the sequence v_i as defined above, and lets calculate

$$\begin{aligned} N_n(a) &= \#\{v_i = a, i = 1 \ldots, n\} \\ &= \#\{\lceil Ma/K \rceil \leq u_i \leq \lfloor Ma/K + M/K \rfloor, i = 1, \ldots, n\} \end{aligned}$$

We know that $u_i \in \mathcal{M}$, thus for every $a \in \mathcal{K}$, we just need to count how many numbers of $\mathcal{M}$ will produce a single number in $\mathcal{K}$, and this should be independent of a, i.e. if the following does not depend on a

$$\lfloor Ma/K + M/K \rfloor - \lceil Ma/K \rceil.$$

If now M is divisible by K then Ma/K is an integer and M/K is an integer, thus

$$\lfloor Ma/K + M/K \rfloor - \lceil Ma/K \rceil = Ma/K + M/K - Ma/K = M/K,$$

which proves our first claim. The second claim follows from realizing that

$$\begin{aligned} v_{i+T_0} = v_i &\Leftrightarrow \lfloor (u_{i+T_0}/M) * K \rfloor = \lfloor (u_i/M) * K \rfloor \\ &\Leftrightarrow u_{i+T_0}/M * K = u_i/M * K \\ &\Leftrightarrow u_{i+T_0} = u_i. \end{aligned}$$

That is, the period of v_i is the same as the period of u_i which is M. □

The following number theoretical theorem tells us exactly when we can expect period M.

Theorem 6.13 (Hull–Dobell Theorem)**.** *The congruential generator* (a, b, M) *has period* M iff

- $\gcd(b, M) = 1$,
- p *divides* $a - 1$ *for every prime* p *that divides* M
- 4 *divides* $a - 1$ *if* 4 *divides* M.

See [HD, K].

Let us check the moments of the pseudorandom numbers generated

Lemma 6.14. *Let* $u_0, u_1, \ldots$ *be a psuedo random sequence over* $\mathcal{M} = \{0, 1, \ldots, M-1\}$. *Then the empirical mean and variance has limits as follows*

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} u_i = \frac{M-1}{2}$$

and

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} u_i^2 - \left(\frac{1}{n} \sum_{i=1}^{n} u_i \right)^2 = \frac{M^2 - 1}{12}.$$

Proof. From Definition 6.3

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} u_i = \lim_{n\to\infty} \sum_{i=0}^{M-1} i \frac{N_n(i)}{n} = \sum_{i=0}^{M-1} \frac{i}{M} = \frac{M-1}{2}$$

The empirical variance follows similarly. □

The conclusion is that the long term empirical moments converge to the discrete uniform over $\mathcal{M}$.

We saw earlier that rescaling the result by dividing (see Lemma 6.12) gives us a generator over a smaller set. Our initial problem of generating number from the uniform distribution can be partially solved by a generator of large period.

Corollary 6.15. *Let* $u_0, u_1, \ldots$ *be a psuedo random sequence over* $\mathcal{M} = \{0, 1, \ldots, M-1\}$. *Then* $v_i = u_i/M$ *has the empirical mean and variance limits as follows*

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} v_i = \frac{1}{2} - \frac{1}{2M}$$

and

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} v_i^2 - \left(\frac{1}{n} \sum_{i=1}^{n} v_i \right)^2 = \frac{1}{12} - \frac{1}{12M^2}.$$

From the above we see that if we have a generator with a large period M, which could be over 64-bit integers, i.e. we could have period 2^{64}, then the resulting u_i/M will have mean $\frac{1}{2} - 2^{-65}$ and variance $\frac{1}{12} - \frac{1}{3}2^{-130}$. Will such a generator be an UPRNG? Actually no, but we can achieve the following

Lemma 6.16. *Let $v_0, v_1, \ldots$ be a pseudorandom sequence in $\mathcal{M} = \{0, 1, \ldots, M-1\}$, define $u_i = v_i/M$. For any interval $A = (a, b) \subset [0, 1]$, define $N_n(A)$ as the number of $u_i \in A$ for $i = 0, 1, 2, \ldots, n-1$. We have*

$$\left| \lim_{n\to\infty} \frac{N_n(A)}{n} - \int_A dx \right| \leq \frac{1}{M}.$$

For reasons a little bit beyond this course, it turns out that we cannot represent a UPRNG on a finite machine, we can however generate almost an UPRNG as seen in the lemma above.

Now all of what we have done so-far is to check whether the sequence has the right limiting distribution and the right moments. But, this doesn't tell us anything about the randomness, for instance the sequence $0, 1, 0, 1, 0, 1, \ldots$ is a pseudorandom sequence over $\{0, 1\}$, it is however very structrured and thus not random. There is a series of tests one usually does to verify if a pseudorandom sequence is good, we will see this a bit more in the computer exercises.

6.2 Sampling

The previous section was just to give you a flavor of what random in a computer represents, namely it is not random but a deterministic sequence that "looks" random. We will leave the pseudorandom part for now and instead attack the problem of sampling from a generic distribution given a random sample from Uniform($[0, 1]$).

> Recall from Theorem 5.38 that given a distribution function F and $X \sim \text{Uniform}([0, 1])$, then $F^{-1}(X) \sim F$.

However, sometimes finding the quantile function F^{-1} can be analytically impossible or if done numerically, very expensive. There are other ways to sample that are more costly than inversion sampling (given the inverse) but sometimes cheaper than computing the inverse.

As you can see in Algorithm 1, the updated variable is a conditional random variable on the event $U \leq r(X)$, the distribution of this conditional random variable is the distribution we are after, namely F. This is contained in the lemma below:

Lemma 6.17. *Let $X \sim G$ and let $U \sim \text{Uniform}([0, 1])$, then if we define the RV $I = \mathbb{1}_{U \leq r(X)}$, the random variable $Y = X \mid (I = 1)$ satisfies $Y \sim F$.*

Algorithm 1 Accept-Reject Sampler

1: *input:*

(1) a target density $f(x)$,

(2) a sampling density $g(x)$ that satisfies $f(x) \leq Mg(x)$.

2: *output:* a sequence of samples $x_0, \ldots$ with distribution f
3: Sample initial state $X^{(0)}$ from g.
4: **repeat**
5: At iteration t,
6: Generate x from g and compute the ratio $r(x) = \frac{f(x)}{Mg(x)}$
7: Draw $U \sim \text{Uniform}([0,1])$ and set $X^{t+1} = x$, if $U \leq r(x)$, otherwise **goto** 6?
8: **until** desired number of samples are obtained.

Proof. By the properties of conditional densities we have the equality (Bayes)

$$f_{X|I}(x \mid I = 1) = \frac{f_{I|X}(I = 1 \mid X = x) f_X(x)}{f_I(1)}.$$

Let us compute the constituents of the RHS of the above. Now since I is discrete we know that

$$f_{I|X}(I = 1 \mid X = x) = \mathbb{P}(I = 1 \mid X = x) = \mathbb{P}(U \leq r(x)) = r(x)$$

and from the law of total probability (Theorem 1.16)

$$f_I(1) = \mathbb{P}(I = 1) = \int \mathbb{P}(I = 1 \mid X = x) g(x) dx = \frac{1}{M} \int p(x) dx = \frac{1}{M}.$$

Finally we achieve

$$f_{X|I}(x \mid I = 1) = \frac{r(x)g(x)}{1/M} = \frac{f(x)g(x)M}{Mg(x)} = f(x)$$

□

In some cases there is not really simple density g to the density f that you wish to sample from, so the accept-reject does not work. An example of this is the Normal distribution, which basically requires us to have a density g which is Gaussian like if we want the algorithm to not reject way too much. So how do we do it? We have a few options

1. Since the distribution function for the Gaussian does not have a closed form, the inverse is hard to compute and requires a lot of computation.

There does however exist approximations of the inverse that we can use.

$$\Phi^{-1}(\alpha) \approx t - \frac{a_0 + a_1 t}{1 + b_1 t + b_2 t}.$$

You will find the constants in the notebooks.

2. Use a transformation method, i.e. find some kind of function $h(x)$ and a simpler distribution F such that $h(X)$ is Gaussian or close to it.

Theorem 6.18 (Box-Muller)**.** *Suppose that* $U_1, U_2 \overset{\text{IID}}{\sim} \text{Uniform}([0,1])$*, then*

$$Z_0 = \sqrt{-2\ln(U_1)}\cos(2\pi U_2)$$
$$Z_1 = \sqrt{-2\ln(U_1)}\sin(2\pi U_2)$$

are independent random variables, and $Z_0, Z_1 \sim \text{Normal}(0,1)$*.*

Proof. Consider bivariate normal RV. Z, then the distribution of $Y = |Z|^2$ is χ^2 distributed with 2 degrees of freedom. Furthermore $W = Z/|Z|$, is uniformly distributed on the unit circle. We know that Y, W are independent (see Exercise 6.19). Thus to generate a bivariate normal it is enough to generate from a χ^2 distribution with 2 degrees of freedom and a point from the uniform distribution on the circle. The χ^2 with 2 degrees of freedom is just the exponential distribution with parameter $1/2$, as such we can generate it using the inversion sampling method (Theorem 5.38). The rest of the proof is left as an exercise. □

Exercise 6.19. *First show that* W, Y *in the proof above are independent. Then show that* W *generated using* $(\cos(2\pi U_2), \sin(2\pi U_2))$ *is uniform on the unit circle. Finally to show that* Z_0, Z_1 *are independent, since they are Gaussian it suffices to show that their covariance is zero.*

6.3 Practice exercises

Exercise 6.20.

- *Implement your own congruential generator* (a, b, M)*.*
- *Use the congruential generator to generate pseudo random numbers from* $\text{Uniform}([0,1])$*. Test out a combination* (a, b, M) *that seems to work well when tested with for instance, Kolmogorov Smirnoff. This is easy to do, let* $\hat{F}_n$ *denote the empirical distribution function according*

to the n samples drawn, compare this to the distribution function for the uniform distribution F and consider

$$\sup_{x\in[0,1]} |\hat{F}_n(x) - F(x)|.$$

Derive a statistical test based on Theorem 5.27 and test whether your sampler passes.

Exercise 6.21.

- *Now that you can sample from the uniform distribution, generate samples from $N(10, 5)$ using the Box-Muller method, Theorem 6.18.*
- *Repeat the testing from Exercise 6.20 but now compare to F being the normal distribution.*
- *What did you actually assume for the test above? Is it accurate? Does it satisfy the conditions of Theorem 5.27?*

Exercise 6.22. *Consider the continuous distribution with density*

$$p(x) = \frac{1}{2}\cos(x), \quad -\frac{\pi}{2} < x < \frac{\pi}{2}.$$

- *Plot the distribution function $F(x)$.*
- *Find the inverse distribution function F^{-1}.*
- *Implement an inversion sampler to sample from F.*
- *Implement an Accept-Reject Sampler, Algorithm 1 with sampling density* Uniform$([-\pi/2, \pi/2])$. *On average, how many samples get rejected?*