MTH210a: Statistical Computing

Instructor: Dootika Vats

February 10, 2024

Contents

1 Lecture-wise Summary

Lec No.	Date	Topic
1	Jan 5	FCH and Pseudorandom number generation
2	Jan 8	Pseudorandom numbers
3	Jan 9	Inverse Transform Method
4	Jan 12	Accept-Reject (discrete)
5	Jan 15	Accept-Reject (discrete)
6	Jan 16	Continuous: inverse transform and accept-reject
7	Jan 19	Accept-reject (Continuous)
8	Jan 23	Accept-reject (Continuous)
9	Jan 29	Quiz 1 and Accept-Reject
10	Jan 30	Box-Muller, Ratio-of-Uniforms
11	Feb 2	Ratio-of-Uniforms
12	Feb 5	Composition
13	Feb 6	Miscellaneous, Multivariate normal
14	Feb 10	Simple Monte Carlo

2 Pseudorandom Number Generation

The building block of computational simulation is the generation of uniform random numbers. If we can draw from $U(0,1)$, then we can draw from *most* other distributions. Thus the construction of sampling from $U(0,1)$ requires special attention.

Computers can generate numbers between $(0,1)$, which although are not exactly random (and in fact deterministic), but have the appearance of being $U(0,1)$ random variables. These draws from $U(0,1)$ are *pseudorandom* draws.

The goal in *pseudorandom* generation is to draw

$$X_1, \dots, X_n \overset{\text{approx iid}}{\sim} U(0,1)\,.$$

The resultant sample is as uniformly distributed as possible, and as independent as possible. We will learn about two different pseudorandom generators. These are very basic ones that are actually not really used in real life, but make our point well.

Note: After this lecture, we will always assume that all $U(0,1)$ draws are exactly iid and perfectly random. We will forget that they are infact, pseudorandom. Pseudorandom generation is a whole field in itself; for more on this, checkout CS744 at IITK.

2.1 Multiplicative congruential method

A common algorithm to generate a sequence $\{x_n\}$ is the *multiplicative congruential method*:

1. Set *seed* x_0, and positive integers a, m.
2. Obtain $x_t = a\, x_{t-1} \mod m$
3. Return sequence x_t/m for $t = 1, \dots, n$.

Since $x_t \in \{0, 1, \dots m-1\}$, $x_t/m \in (0,1)$. Also note that after some finite number of steps $< m$, the algorithm will repeat itself, since when a seed x_0 is set, a deterministic sequence of numbers follows. Naturally, to allow for the sequence x_t to mimic uniform and random draws m should be large. Naturally, both a and m should be chosen to be large so as to avoid repetition. Typically m should be a large prime number.

Example 1. Set $a = 123$ and $m = 10$, and let $x_0 = 7$. Then
$x_1 = 123 * 7 \mod 10 = 1$

$$\begin{aligned}
x_2 &= 123 * 1 \mod 10 = 3 \\
x_3 &= 123 * 3 \mod 10 = 9 \\
x_4 &= 123 * 9 \mod 10 = 7 \\
x_5 &= 123 * 7 \mod 10 = 1 \\
&\vdots
\end{aligned}$$

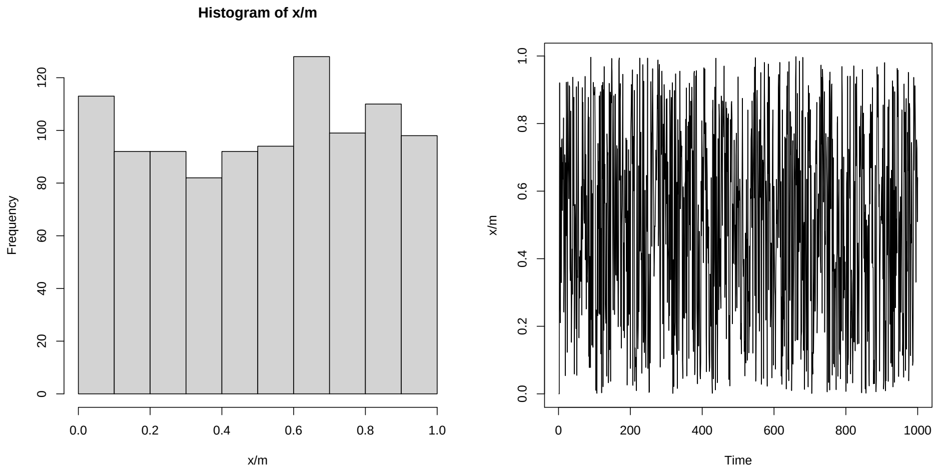
■

Thus, we see that the above choices of a, m, x_0 repeats itself. It is also recommended that a is large to ensure large jumps, and reduce ”dependence” in the sequence. Based on the bits of your machine, it is recommended to set $m = 2^{31} - 1$ and $a = 7^5$. Notice that both are large.

```r
m <- 2^(31) - 1
a <- 7^5
x <- numeric(length = 1e3)
x[1] <- 7

for(i in 2:1e3)
{
  x[i] <- (a * x[i-1]) %% m
}
par(mfrow = c(1,2))
hist(x/m) # looks close to uniformly distributed
plot.ts(x/m) # look like it's jumping around too
```

The histogram shows roughly ”uniform” distribution of the samples and the trace plot shows the lack of dependence between samples.



Any pseudorandom generation method should satisfy:

1. for any initial seed, the resultant sequence has the "appearance" of being IID from Uniform$(0, 1)$.
2. for any initial seed, the number of values generated before repetition begins is large
3. the values can be computed efficiently.

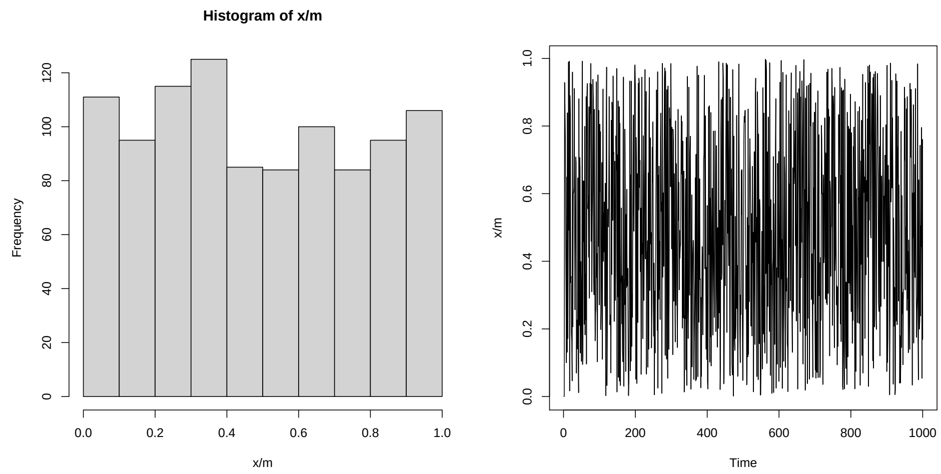
2.2 Mixed Congruential Generator

Notice that in the previous method, if we set the seed to be zero, the algorithms fails! To combat this, there is another method, the *mixed congruential generator*:

1. Set seed x_0, and positive integers a, c, m.
2. $x_t = (a\, x_{t-1} + c) \mod m$
3. Return sequence x_t/m for $t = 1, \dots, n$.

```
m <- 2^(31) - 1
a <- 7^5
c <- 2^(10) - 1
x <- numeric(length = 1e3)
x[1] <- 7

for(i in 2:1e3)
{
  x[i] <- (c + a * x[i-1]) %% m
}
par(mfrow = c(1,2))
hist(x/m) # looks close to uniformly distributed
plot.ts(x/m) # look like it's jumping around too
```

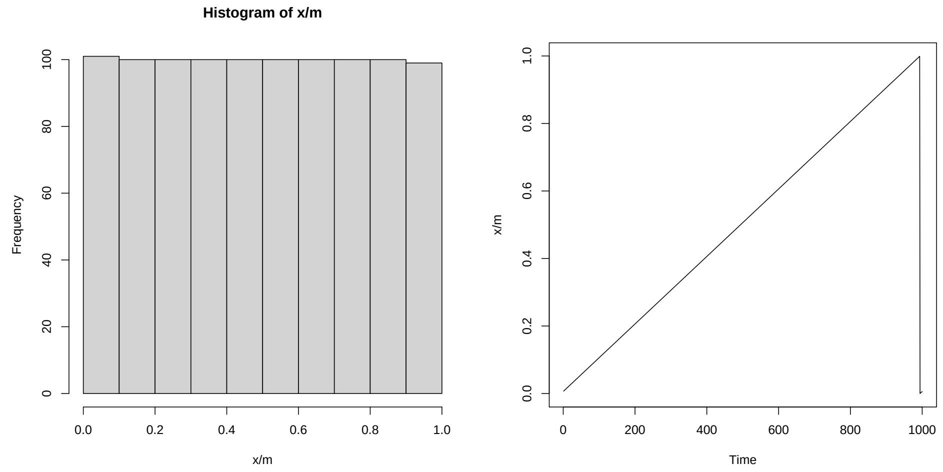


We must be cautious not to be happy with a just a histogram. A histogram shows that the empirical distribution of all samples is uniformly distributed. But we can still get a uniform looking histogram if we set $a = 1$, $m = 1e3$ and $c = 1$.

```r
m <- 1e3
a <- 1
c <- 1
x <- numeric(length = 1e3)
x[1] <- 7

for(i in 2:1e3)
{
  x[i] <- (c + a * x[i-1]) %% m
}
par(mfrow = c(1,2))
hist(x/m) # looks VERY uniformly distributed
plot.ts(x/m) # Clearly "dependent" samples
```

Although a histogram shows an almost perfect uniform distribution, the trace plot shows that the draws don't behave like they are independent.



We could also use

$$x_n = (a_1 x_{n-1} + a_2 x_{n-2} + \cdots + a_k x_{n-k} + c) \mod m\,,$$

but this requires more flops from the computer, and so is not as computationally viable.

We claim that these methods return "good" pseudosamples, in the sense of the three points stated above. There are statistical hypothesis tests, like the Kolmogorov-Smirnov test, one can do to test whether a sample is truly random: independent and identically distributed.

`runif()` in R uses the Mersenne-Twister generator by default (we will not go into this), but there are options to use other generators. After this, we will assume that `runif()` returns truly iid samples from $U(0,1)$.

2.3 Generating $U(a,b)$

Suppose we can draw from $U(a,b)$ for any $a, b \in \mathbb{R}$. But we only know how to draw from $U(0,1)$. Note that if $U \sim U(0,1)$, then for any a, b,

$$(b-a)U + a \sim U(a,b) \quad .$$

That means, we can draw $U \sim U(0,1)$ and set $X = (b-a)U + a$. Then $X \sim U(a,b)$.

```
# Try for yourself

set.seed(1)
repeats <- 1e4
b <- 10
a <- 5
U <- runif(repeats, min = 0, max = 1)
X <- (b - a) * U + a #R is vectorized

hist(X)
```

Questions to think about

- Given a sample of pseudorandom draws from $U(0,1)$ and perfectly IID draws from $U(0,1)$, would you be able to tell the difference?
- Could we obtain uniform samples from $\mathbb{R}$?

2.4 Exercises

1. (Using R) Consider the multiplicative congruential method. For a, m positive integers

$$x_n = ax_{n-1} \mod m\,.$$

   (a) Set seed $x_0 = 5$, $m = 10^4, a = 2$. Generate $n = 10^4$ pseudorandom numbers using the above method. Does this look like a (pseudo) random sample from Uniform$(0,1)$? Maybe plot a histogram to see the empirical distribution.

   (b) Now look at only the first 10 numbers: $x_1, x_2, \ldots, x_{10}$. What is the problem here?

   (c) How can you fix the problem noted in the previous step?

3 Generating Discrete Random Variables

Suppose X is a discrete random variable having probability mass function

$$\Pr(X = x_j) = p_j \quad j = 0, 1, \ldots, \quad \sum p_j = 1\,.$$

Examples of such random variables are: Bernoulli, Poisson, Geometric, Negative Binomial, Binomial, etc. We will learn two methods to draw samples realizations of this discrete random variable:

1. Inverse transform method
2. The acceptance-rejection technique

3.1 Inverse transform method

Let's demonstrate the inverse transform method with an example first.

Example 2 (Bernoulli distribution)**.** If $X \sim \text{Bern}(p)$, then

$$\Pr(X = 1) = p \qquad \text{and} \qquad \Pr(X = 0) = 1 - p := q\,.$$

Let $U \sim U(0, 1)$. Define

$$X = \begin{cases} 0 & \text{if } U \leq q \\ 1 & \text{if } q < U \leq 1 \end{cases}.$$

Then $X \sim \text{Bern}(p)$.

Proof. To show the result we only need to show that $\Pr(X = 1) = p$ and $\Pr(X = 0) = 1 - p$. Recall that by the cumulative distribution function of $U(0, 1)$, for any $0 < t < 1$, $\Pr(U \leq t) = t$. Using this,

$$\Pr(X = 0) = \Pr(U \leq q) = q\,,$$

and also

$$\Pr(X = 1) = \Pr(q < U \leq 1) = 1 - q = p\,.$$

■

■

Algorithm 1 Inverse transform algorithm for Bern(p)

```
Draw $U \sim U(0,1)$
**if** $U < q$ **then** $X = 0$ **else** $X = 1$
**return** X
```

Inverse transform method: The principles used in the above example can be extended to any generic discrete distribution. For a distribution with mass function

$$\Pr(X = x_j) = p_j \qquad \text{for } j = 0, 1, \dots \quad \text{with} \quad \sum_{j=0} p_j = 1\,.$$

Let $U \sim U(0,1)$. Set X to be

$$X = \begin{cases} x_0 & \text{if } U \leq p_0 \\ x_1 & \text{if } p_0 < U \leq p_0 + p_1 \\ x_2 & \text{if } p_0 + p_1 < U \leq p_0 + p_1 + p_2 \\ \vdots & \\ x_j & \text{if } \sum_{i=0}^{j-1} p_i < U \leq \sum_{i=0}^{j} p_i \end{cases}.$$

This works because

$$\Pr(X = x_j) = \Pr\left(\sum_{i=0}^{j-1} p_i < U \leq \sum_{i=0}^{j} p_i\right) = \sum_{i=0}^{j} p_i - \sum_{i=0}^{j-1} p_i = p_j\,.$$

This method is called the *inverse transform method* since the algorithm is essentially looking at the inverse cumulative distribution function of the random variable.

Example 3 (Poisson random variables)**.** The probability mass function for the Poisson random variable is

$$\Pr(X = i) = p_i = \frac{e^{-\lambda}\lambda^i}{i!} \quad i = 0, 1, 2, \dots,$$

Algorithm 2 Inverse transform for Poisson(λ)

```
Draw $U \sim U(0,1)$
if $U \leq p_0$ then
    $X = 0$
else if $U \leq p_0 + p_1$ then
    $X = 1$
    ...
else if $U \leq \sum_{i=1}^{j} p_i$ then
    $X = j$
    ...
```

However, Algorithm 2 outlines a challenge in implementing this algorithm.

Q. What happens when λ is large?

A Poisson(λ) distribution with a large λ will yield p_j to be small when j is small. This implies Algorithm 2 can be quite slow here. A way to make it faster is the following. We know that most likely, a realization from Poisson will be closer to λ, so it will be beneficial to start from around λ. Set $I = \lfloor \lambda \rfloor$, and check whether

$$\sum_{i=0}^{I-1} p_i < U \leq \sum_{i=0}^{I} p_i \,.$$

If it is, then return $X = I$. Else, if $U > \sum_{i=1}^{I} p_i$, then increase I, otherwise, decrease I and check again. ■

Questions to think about

- What other example can you think of where the inverse transform method could take a lot of time?
- Can you try and implement this for a Binomial random variable?

3.2 Accept-Reject for Discrete Random Variables

Although we can draw from any discrete distribution using the inverse transform method, you can imagine that for distributions on countably infinite spaces (like the

Poisson distribution), the inverse transform method may be very expensive. In such situations, and in some other situations as well, acceptance-rejection sampling may be more reliable.

Let $\{p_j\}$ denote the pmf of the target distribution with $\Pr(X = a_j) = p_j$ and let $\{q_j\}$ denote the pmf of another distribution with $\Pr(Y = a_j) = q_j$. Suppose you can efficiently draw from $\{q_j\}$ and you want to draw from $\{p_j\}$. Let c be a constant such that

$$\frac{p_j}{q_j} \leq c < \infty \quad \text{for all } j \text{ such that } p_j > 0\,.$$

That is,

$$c \geq \sup_{j : p_j > 0} \frac{p_j}{q_j}\,.$$

If we can find such a $\{q_j\}$ and c, then we can implement an *Acceptance-Rejection* algorithm also known as *Accept-Reject* sampler. Here, distribution $\{q_j\}$ is called the proposal distribution. The idea is to draw samples from $\{q_j\}$ and accept these samples if they seem likely to be from $\{p_j\}$.

Note: When $\{p_j\}$ has a finite set of states, c is always finite (since the maximum exists). However, when target distribution does not have a finite set of states, then c need not be finite, and accept-reject is not possible.

Algorithm 3 Acceptance-Rejection sampler to draw 1 sample from $\{p_j\}$ using proposal $\{q_j\}$

```
1: Draw $U \sim U(0,1)$
2: Simulate $Y = y$ from proposal, independent of $U$. Let $q_y = \Pr(Y = y)$ and let $p_y = \Pr(X = y)$.
3: if $U \leq \frac{p_y}{cq_y}$ then
4:     Return $X = y$ and stop
5: else
6:     Goto step 1
```

Theorem 1. When c is finite, the Accept-Reject method generates a random variable with probability

$$\Pr(X = a_j) = p_j\,.$$

Further, the number of iterations needed to generate an acceptance is distributed as Geometric$(1/c)$.

Proof. First, we look at the second statement. We note that the number of iterations required to stop the algorithm is clearly geometrically distributed by the definition of the geometric distribution – the distribution of the number of Bernoulli trials needed to get one success (with support $1, 2, 3...$).

We will show that the probability of success is $1/c$. "Success" here is an acceptance. First, see that in any iteration of the algorithm we have

$$\begin{aligned}\Pr(Y = a_j, \text{accept}) &= \Pr(Y = a_j)\ \Pr(\text{accept} \mid Y = a_j)\\ &= q_j \Pr\left(U \leq \frac{p_j}{cq_j} \mid Y = a_j\right)\\ &= q_j \frac{p_j}{cq_j} = \frac{p_j}{c}\,.\end{aligned}$$

Using this we can calculate the marginal pmf of accepting:

$$\Pr(\text{accept}) = \sum_j \Pr(Y = a_j, \text{accept}) = \sum_j \frac{p_j}{c} = \frac{1}{c}\,.$$

Thus, the second statement is proved. We will now use this to show the main statement. Note that

$$\begin{aligned}\Pr(X = a_j) &= \sum_{n=1}^{\infty} \Pr(a_j \text{ accepted on iteration } n)\\ &= \sum_{n=1}^{\infty} \Pr(\text{No acceptance until iteration } n-1)\ \Pr\left(Y = a_j, \text{accept}\right)\\ &= \underbrace{\sum_{n=1}^{\infty} \left(1 - \frac{1}{c}\right)^{n-1}}_{c}\ \frac{p_j}{c}\\ &= p_j\,.\end{aligned}$$

This completes the proof. □

Note: Since the probability of acceptance in any loop is $1/c$, the expected number of loops for one acceptance is c. The larger c is, the more expensive the algorithm.

One important thing to note is that within the support $\{a_j\}$ of $\{p_j\}$, the proposal distribution must always be positive. That is, for all a_j in the support of $\{p_j\}$, $\Pr(Y = a_j) = q_j > 0$. That is, a proposal distribution must have support *larger* than the target distribution.

Example 4 (Sampling from Binomial using AR)**.** The binomial distribution has pmf

$$\Pr(X = x) = \binom{n}{x}(1-p)^{n-x}p^x \qquad \text{for } x = 0, 1, \dots, n\,.$$

We will use AR to simulate draws from Binomial(n, p). The first task is to choose a proposal distribution. We could use any of Poisson, negative-binomial, or geometric distributions. We cannot use Bernoulli, since the support of Bernoulli does not contain the support of Binomial.

We choose to use the geometric distribution, but we must be a little careful. We use the version of geometric distribution that is defined as the number of failures before the first success, so that the support of the geometric distribution has 0 in it. The pmf of the geometric distribution is

$$\Pr(X = x) = (1-p)^x p \qquad x = 0, 1, \dots\,.$$

We will first find c. Note that

$$\begin{aligned}\frac{p(x)}{q(x)} &= \frac{\binom{n}{x}(1-p)^{n-x}p^x}{(1-p)^x p}\\ &= \binom{n}{x}(1-p)^{n-2x}p^{x-1}\,.\end{aligned}$$

Set

$$c = \max_{x=0,1,\dots,n} \binom{n}{x}(1-p)^{n-2x}p^{x-1}\,.$$

For $n = 10, p = 0.25$, we yield $c = 2.373\dots$.

To be safe (since we don't know all the decimal points), we may set c to be slightly larger (say $c = 2.5$) as c just needs to be an upper bound. Once c is known, the AR algorithm can be implemented simply as described. Now here, we would expect, on average, 2.5 values of Geometric random variables to be proposed until *one* acceptance.

Note, that c depends on both n and p. Particularly, if n is large, then c increases drastically. A way to understand this is then the mean of the target distribution (np) can be much larger than the mean of the proposal, $(1-p)/p$. In this case, this implies that the bulk of the mass of the target distribution is far away from the bulk of the mass of the proposal distribution. This is not ideal. We want the pmf of the proposal and target to match each other as much as possible, so that c is close to 1. This suggests, that we may **not** want to choose the same p in the proposal distribution!

A possible fix, is to consider a Geometric(p^*) proposal where p^* is such that the mean of the target and the mean of the proposal matches:

$$np = \frac{1-p^*}{p^*} \Rightarrow p^* = \frac{1}{np+1}$$

In this case, we have

$$\frac{p(x)}{q(x)} = \frac{\binom{n}{x}(1-p)^{n-x}p^x}{(1-p^*)^x p^*}$$

the maximum over $\{0, 1, \ldots, n\}$ can be determined on the computer. For $n = 100$ and $p = .25$, the old bound is 1028.497 and the new one is 6.0455, which is much more efficient! ■

Example 5. (Geometric Random Variable) We consider the Geometric random variable with pmf (trails until x failures)

$$\Pr(X = x) = (1-p)^x p \qquad x = 0, 1, 2, \ldots$$

We cannot use Binomial as a proposal, but we can use Poisson. Let us consider the Poisson(λ) proposal. The Poisson random variable has pmf

$$\Pr(X = x) = \frac{e^{-\lambda}\lambda^x}{x!} \qquad x = 0, 1, 2, \ldots.$$

First step is to find c, if it exists

$$\begin{aligned}\frac{p(x)}{q(x)} &= \frac{(1-p)^x p}{\frac{e^{-\lambda}\lambda^x}{x!}} \\ &= \frac{p}{e^{-\lambda}}\left(\frac{1-p}{\lambda}\right)^x x!\,.\end{aligned}$$

For small values of λ ($< 1-p$), the above clearly diverges as x increases, thus the maximum doesn't exist. This is true for large values of λ as well. To see, this (intuitively), consider the Stirling's approximation of the factorial:

$$\log(x!) \approx x\log(x) - x \Rightarrow x! \approx e^{x\log x - x}\,.$$

Using this:

$$\begin{aligned}\frac{p(x)}{q(x)} &= \frac{p}{e^{-\lambda}}\left(\frac{1-p}{\lambda}\right)^x x! \\ &\approx \frac{p}{e^{-\lambda}}\left(\frac{1-p}{\lambda}\right)^x e^{x\log x - x} \\ &= \frac{p}{e^{-\lambda}}\left(\frac{(1-p)e^{\log(x)}}{e\lambda}\right)^x\end{aligned}$$

Thus, no matter how large λ is, eventually as x increases $e^{\log(x)}$ will be larger than λ and the ratio will diverge. Thus, this proposal does not allow an AR for the Geomtric distribution. ■

Question to think about

- Why is c always greater than 1?
- What happens when c is large or small?

3.3 Exercises

1. Show that if $U \sim U(0,1)$, then for any a, b,

$$(b-a) * U + a \sim U(a,b)$$

2. Use the inverse transform method to sample from a geometric distribution, where for $0 < p < 1$ and $q = 1-p$,

$$\Pr(X = i) = pq^{i-1}, \quad i \geq 1, \quad \text{where } q = 1-p\,.$$

3. We want to draw a sample from the random vector $(X,Y)^\top$, that follows the distribution with joint probability mass function

$$P(X=i, Y=j) = \theta_{i,j} \qquad \text{where } \; i,j \in \{0,1\}\,.$$

Here $\sum_{i,j} \theta_{i,j} = 1$. Write an inverse-transform algorithm to draw realizations of $(X, Y)^\top$.

4. List as many appropriate proposal distributions as you can think of for the following target distributions:

   - Binomial
   - Bernoulli
   - Geometric
   - Negative Binomial
   - Poisson

5. (Using R) Draw 10,000 draws from a Binomial$(20, .75)$ distribution using an accept-reject sampler.

6. In an accept-reject algorithm, we need to find c such that

$$\frac{p_j}{q_j} \leq c \quad \text{forall } j \text{ for which } p_i > 0\,.$$

   And, the probability of accepting in any iteration is $1/c$. Why is c guaranteed to be more than 1?

7. Simulate from a Negative Binomial(n, p) using the inverse transform and accept-reject methods. Implement in R with $n = 10$ successes and $p = .30$.

8. Simulate from the following "truncated Poisson distribution" with pmf:

$$\Pr(X = i) = \frac{e^{-\lambda}\lambda^i/i!}{\sum_{j=0}^{m} e^{-\lambda}\lambda^j/j!} \quad i = 0, 1, 2, \dots, m\,.$$

   Implement in R with $m = 30$ and $\lambda = 20$.

9. Suppose we want to obtain samples from a discrete distribution with pmf $\{p_i\}$. We use accept-reject with proposal distribution with pmf $\{q_i\}$, such that for some $\alpha \in \mathbb{R}$:

$$\frac{p_i}{q_i} \propto i^\alpha \qquad i = 1, 2, \dots, ,$$

   For what values of α would this AR algorithm work?

10. Suppose we want to obtain samples from a discrete distribution with pmf $\{p_i\}$. Two possible proposal distributions are $\{q_i^{(1)}\}$ and $\{q_2^{(2)}\}$, yielding AR bounds c_1 and c_2 such that $c_1 > c_2$. Which proposal distribution is better?

11. Implement a an algorithm to sample from a Zero Inflated Binomial distribution. Can you think of an application of such a distribution?

4 Generating continuous random variables

Similar to generating discrete random variables, there are various methods for generating continuous random variables. We will discuss three main methods:

1. Inverse transform
2. The accept-reject method
3. Ratio of uniforms

We will also discuss a few special samplers.

4.1 Inverse transform

The principles of the inverse transform method for discrete distributions, apply similarly to continuous random variables. Consider a random variable X with probability density function $f(x)$ so that $f(x) \geq 0$, $\int_{-\infty}^{\infty} f(x) = 1$ with distribution function

$$F(x) = \int_{-\infty}^{x} f(x)\, dx\,.$$

The following theorem will be the foundation for the inverse transform method.

Theorem 2. Let $U \sim U(0,1)$. For any continuous distribution F, a random variable $X = F^{-1}(U)$ has distribution F.

Proof. Let F_X be the distribution function of $X = F^{-1}(U)$. We need to show that $F_X = F$. Note that for any $x \in \mathbb{R}$,

$$\begin{aligned}
F_X(x) &= \Pr(X \leq x)\\
&= \Pr(F^{-1}(U) \leq x)\\
&= \Pr(F(F^{-1}(U)) \leq F(x)) \qquad \text{(Since F is non-decreasing)}\\
&= \Pr(U \leq F(x))\\
&= F(x)\,.
\end{aligned}$$

□

The above theorem then implies that if we can invert the CDF function, then we can obtain random draws from that random variable.

Example 6. Exponential(1)**:** For the Exponential(1) distribution, the cdf is $F(x) = 1 - e^{-x}$. Thus,

$$F^{-1}(u) = -\log(1-u)\,.$$

To generate $X \sim \text{Exp}(1)$ we can thus use the following algorithm:

Algorithm 4 Exponential(1) Inverse transform

```
Generate $U \sim U(0,1)$
Set $X = -\log(1-U) \sim \text{Exp}(1)$
```

Similarly, we can draw from an Exponential(λ) distribution. ■

Example 7. Cauchy distribution: Cauchy distribution has pdf

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

and

$$u = F(x) = \int_{-\infty}^{x} f(y)dy = \frac{1}{\pi}\arctan(x) + \frac{1}{2}\,.$$

So, $F^{-1}(u) = \tan(\pi(u - .5))$.

Algorithm 5 Cauchy distribution

```
Generate $U \sim U(0,1)$
Set $X = \tan(\pi(U - .5) \sim \text{Cauchy}$
```

■

Example 8. Gamma distribution: The CDF of a Gamma(n, λ) distribution is

$$F(x) = \int_0^x \frac{\lambda e^{-\lambda y}(\lambda y)^{n-1}}{\Gamma(n)}dy\,.$$

Here, we don't know the CDF in closed form and thus cannot analytically find the inverse. This is an example where the inverse transform method cannot work in practice (even though it works theoretically). Thus, unlike the discrete case, this genuinely motivates the need for another method to sample from a distribution.

■

Questions to think about

- The CDF $F(x)$ is a deterministic function, so how is $F^{-1}(U)$ a random quantity?
- Can we use the inverse transform method to generate sample from a normal distribution?

4.2 Accept-reject method

Suppose we cannot generate from distribution F with pdf $f(x)$, like the Gamma distribution example in inverse transform. We can use accept-reject in a similar way as the discrete case. That is, we choose an appropriate *proposal distribution* with density $g(x)$, and accept or reject it based on certain probabilities.

Let the support of F be $\mathcal{X}$ and choose a proposal distribution G with density $g(x)$ *whose support is larger or the same as the support of F.* That is, if $\mathcal{Y}$ is the support of G then, $\mathcal{X} \subseteq \mathcal{Y}$. If we can fine c such that

$$\sup_{x \in \mathcal{X}} \frac{f(x)}{g(x)} \leq c \,,$$

then an accept-reject sampler can be implemented.

Algorithm 6 Accept-reject for continuous random variables

```
Draw $U \sim U(0,1)$
Draw proposal $Y \sim G$, independently
if $U \leq \frac{f(Y)}{c\,g(Y)}$ then
    Return $X = Y$
else
    Go to Step 1.
```

Theorem 3. Algorithm 6 returns $X \sim F$. Further, the number of loops the AR algorithm takes to return X is distributed Geometric$(1/c)$.

Proof. Let F_X denote the CDF of the random variable draw returned by the algoritihm. For an arbitrary $x \in \mathcal{X}$. We will show that

$$F_X(x) = F(x)\,.$$

First, we consider the probability of acceptance:

$$\begin{aligned}
\Pr(\text{accept}) &= \Pr\left(U \leq \frac{f(Y)}{cg(Y)}\right) \\
&= \mathrm{E}\left[I\left(U \leq \frac{f(Y)}{cg(Y)}\right)\right] \\
\\
&= \mathrm{E}\left[\mathrm{E}\left[I\left(U \leq \frac{f(Y)}{cg(Y)}\right) \mid Y\right]\right] \qquad\qquad \text{using iterated expectations} \\
&= \mathrm{E}\left[\Pr\left(U \leq \frac{f(Y)}{cg(Y)} \mid Y\right)\right] \\
&= \mathrm{E}\left[\frac{f(Y)}{cg(Y)}\right] \\
&= \int_{\mathcal{Y}} \frac{f(y)}{cg(y)} g(y) dy \\
&= \frac{1}{c}\int_{\mathcal{Y}} f(y) dy \\
&= \frac{1}{c}\int_{\mathcal{X}} f(y) dy + \frac{1}{c}\int_{\mathcal{Y}/\mathcal{X}} f(y) dy \qquad \text{since } \mathcal{Y} \subseteq \mathcal{X} \\
&= \frac{1}{c}\,.
\end{aligned}$$

Now that we have this established, consider

$$\begin{aligned}
F_X(x) = \Pr(X \leq x) &= \Pr(Y \leq x \mid \text{accept}) \\
&= \frac{\Pr\left(Y \leq x, U \leq \frac{f(Y)}{cg(Y)}\right)}{\Pr(\text{accept})} \\
&= c \cdot \mathrm{E}\left[\mathrm{E}\left[I\left(Y \leq x, U \leq \frac{f(Y)}{cg(Y)}\right) \mid Y\right]\right] \\
&= c \cdot \mathrm{E}\left[I\left(Y \leq x\right) \mathrm{E}\left[I\left(U \leq \frac{f(Y)}{cg(Y)}\right) \mid Y\right]\right] \\
&= c \cdot \mathrm{E}\left[I\left(Y \leq x\right) \frac{f(Y)}{cg(Y)}\right] \\
&= c \cdot \int_{-\infty}^{x} \frac{f(y)}{cg(y)} g(y) dy \\
&= \int_{-\infty}^{x} f(y) dy \\
&= F(x)\,.
\end{aligned}$$

□

From the proof, we know that $\Pr(accept) = 1/c$, and so, just like the discrete example, the number of attempts it takes to generate an acceptance is distributed Geometric$(1/c)$. Thus

$$\textit{Expected number of loops for an acceptance is} = c\,.$$

Accept-reject method: intuition

At a proposed value y:

- if $f(y)$ is large but $g(y)$ is small means this value will not be proposed often and is a good value to accept for f, so higher probability of accepting it.
- if $f(y)$ is small but $g(y)$ is large, then this value will be proposed often but is unlikely for f, so accept this value less often.

We can choose any g we want as long its support is larger than other the support of f, and the resulting c is finite. However, some gs will be better than other gs, based on the expected number of iterations, c.

Example 9. Beta distribution: Consider the beta distribution Beta$(4, 3)$, where

$$f(x) = \frac{\Gamma(7)}{\Gamma(4)\Gamma(3)} x^{4-1}(1-x)^{3-1} \quad 0 < x < 1; \quad .$$

Consider a uniform proposal distribution. So that $G = U(0, 1)$ and

$$g(x) = 1 \qquad \text{for } x \in (0,1)\,.$$

Note that, $\mathcal{X} = \mathcal{Y}$ in this case. For this choice of g,

$$\sup_{x\in(0,1)} \frac{f(x)}{g(x)} = \sup_{x\in(0,1)} f(x)$$

We can show that maximum of $f(x)$ occurs at $x = 3/5$ and

$$\sup_{x\in(0,1)} \frac{f(x)}{g(x)} = \sup_{x\in(0,1)} f(x) = 60\left(\frac{3}{5}\right)^3\left(\frac{2}{5}\right)^2 = 2.0736 = c\,.$$

Algorithm 7 Accept-reject for Beta$(4, 3)$

```
1: Draw $U \sim U(0,1)$
2: Draw proposal $Y \sim U(0,1)$
3: if $U \leq \frac{f(Y)}{c\,g(Y)}$ then
4:     Return $X = Y$
5: else
6:     Go to Step 1.
```

■

In order to choose a good proposal distribution (that yields a finite c), it is important to choose a g so that it has "fatter tails" than f. This ensures that as $x \to \pm\infty$, g dominates f, so that $c \to 0$ in the extremes, rather than blows up.

Example 10. Normal distribution

The target density function is

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} .$$

We know that the t-distribution has the right support and fatter tails and the "fattest" t distribution is with degrees of freedom 1, which is Cauchy. The pdf of a Cauchy distribution is

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

(We know we can sample from Cauchy using inverse transform, so that is easy.) We will need to find the supremum of the ratio of the densities. Consider

$$\frac{f(x)}{g(x)} = \frac{\pi}{\sqrt{2\pi}} (1 + x^2) e^{-x^2/2} .$$

When $x \to \infty, -\infty$, $e^{-x^2/2}$ decreases more rapidly than x^2 increases, the ratio tends to zero. This can be shown more formally using L'Hopital's rule.

Taking a derivative of the above ratio and setting it to 0 (and checking the second derivative condition), yields that the supremum above occurs at $x = -1, 1$, so

$$\sup_{x \in \mathbb{R}} \frac{f(x)}{g(x)} = \frac{f(1)}{g(1)} = \sqrt{2\pi} e^{-1/2} \approx 1.746 \Rightarrow c = 1.80\,.$$

The actual algorithm can now be implemented similarly as before.

Note: Consider if the target distribution was Cauchy, and the proposal was $N(0, 1)$? The ratio would clearly diverge as $x \to -\infty, \infty$, and thus an accept-reject sampler would not be possible. ■

Example 11. Sampling from a uniform circle Consider a unit circle centered at $(0, 0)$:

$$x^2 + y^2 < 1 \qquad -1 < x, y < 1\,.$$

We are interested in sampling uniformly from within this circle. Since the area of the circle is π, the target density is

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

Consider the uniform distribution on the square as a proposal distribution

$$g(x, y) = \frac{1}{4} I(-1 < x < 1) I(-1 < y < 1)\,.$$

First, we will find c. For x, y such that $x^2 + y^2 < 1$,

$$\frac{f(x, y)}{g(x, y)} = \frac{4}{\pi} I(x^2 + y^2 < 1) \leq \frac{4}{\pi} := c\,.$$

Next, note that

$$\frac{f(x, y)}{cg(x, y)} = \frac{4}{\pi} I(x^2 + y^2 < 1) \frac{\pi}{4} = I(x^2 + y^2 < 1)\,.$$

So for any (x, y) drawn from within the square, the ratio will be either 1 or 0, thus, no need to draw a uniform at all!

Note: How do we draw uniformly from within the box? Note that

$$g(x, y) = \left[\frac{1}{2} I(-1 < x < 1)\right] \left[\frac{1}{2} I(-1 < y < 1)\right] = g_1(x) \cdot g_1(y)\,,$$

where g_1 is a density of a $U(-1, 1)$ random variable. Thus, with two independent draws $U_1, U_2 \overset{\text{iid}}{\sim} U(-1, 1)$, we have $U_1 \times U_2$ being a draw from the uniform box.

Algorithm 8 Accept-reject for Uniform distribution on a circle

```
Draw proposal $(U_1, U_2) \sim U(-1, 1) \times U(-1, 1)$
if $U_1^2 + U_2^2 \leq 1$ then
    Return $(X, Y) = (U_1, U_2)$
else
    Go to Step 1.
```

$\blacksquare$

Questions to think about

- In A-R, do we want c to be large or small?
- Why is c guaranteed to be more than 1?
- How can you decide whether one proposal distribution is better than another proposal distribution?
- Try implementing the circle/square example in 3 dimension, 4 dimensions, and a general p dimensions. What happens to c?
- Can a similar A-R algorithm be implemented for Beta(m, n) for all $m, n \in \mathbb{Z}$?

4.2.1 Choosing a proposal

Sometimes it is difficult to find a good proposal or even one that works! That is, for a target density $f(x)$ it can sometimes be challenging to find a proposal density $g(x)$ such that

$$\sup_x \frac{f(x)}{g(x)} < \infty$$

Here are certain examples of when it may be difficult / impossible to implement accept-reject.

Example 12 (Beta)**.** Consider a Beta(m, n)

$$f(x) = \frac{\Gamma(m + n)}{\Gamma(m)\Gamma(n)} x^{m-1}(1 - x)^{n-1}$$

Depending on m and n, the Beta distribution can behave quite differently. Particularly, note that when both $m, n < 1$ the Beta density function is unbounded!

When $m, n < 1$, if we use a uniform proposal distribution

$$\sup_{x \in (0,1)} \frac{f(x)}{g(x)} = \sup_{x \in (0,1)} \frac{\Gamma(m+n)}{\Gamma(m)\Gamma(n)} x^{m-1}(1-x)^{n-1} = \infty \,.$$

So a Uniform distribution will not work. In fact, any proposal distribution with a bounded density will not work. So this is an example of a distribution where it is difficult to find a good proposal distribution.

However, when say $n \geq 1$, then

$$\begin{aligned} f(x) &= \frac{\Gamma(m+n)}{\Gamma(m)\Gamma(n)} x^{m-1}(1-x)^{n-1} \\ &\leq \frac{\Gamma(m+n)}{\Gamma(m)\Gamma(n)} x^{m-1} \,. \end{aligned}$$

If we look at the upper bound, the function x^{m-1} on $x \in (0,1)$ can define a valid distribution if normalized. So, consider $g(x) = mx^{m-1}$, which is a proper density on $0 \leq x \leq 1$, and

$$\frac{f(x)}{g(x)} \leq m \frac{\Gamma(m+n)}{\Gamma(m)\Gamma(n)} := c$$

This Accept-Reject sampler can be implemented easily. Similarly if $m \geq 1$. Thus, an AR sampler is easier to implement here if one of m or n is more than (or equal to) 1. ■

Example 13 (Accept-Reject for Cauchy target)**.** Consider the target density

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

The Cauchy distribution is known to have "fat tails" so that as $x \to \pm\infty$, the density function reduces to zero slowly. This means that it is very challenging to find $g(x)$ that "dominates" the density in the tails.

For example, let the proposal be $N(0,1)$. As discussed before, we will see the ratio of the densities be

$$\frac{f(x)}{g(x)} = \frac{2\pi}{\pi} \frac{e^{x^2/2}}{1+x^2} \to \infty \text{ as } x \to \pm\infty!$$

In fact, as far we know, there are no possible standard accept-reject algorithms possible here. ■

4.2.2 Choosing parameters for a fixed proposal family

If you have chosen a family of proposal distributions that you know gives a finite c, it may be unclear what the best parameters for that proposal distribution is. That is, if the target $f(x)$ and the proposal density is $g(x \mid \theta)$ (where θ is a parameter you can change, to change the behaviour of the proposal, then you want to find a value of the parameter θ so that the resulting proposal is the "best".

Notice that the upper bound will be a function of θ, so that

$$\sup_x \frac{f(x)}{g(x|\theta)} \leq c(\theta)\,.$$

The value $c(\theta)$ is the expected number of loops for the accept-reject algorithm. Since we want this to be small, the best proposal density within this family would be the one that minimizes $c(\theta)$, so set

$$\theta^* := \arg\min_\theta c(\theta)\,.$$

Example 14 (Gamma distribution)**.** Consider the target distribution Gamma(α, β)

$$f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}\,.$$

Further, suppose we want to use an Exp(λ) proposal. Then

$$g(x|\lambda) = \lambda e^{-\lambda x}\,.$$

We can now find $c(\lambda)$,

$$\frac{f(x)}{g(x)} = \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{x^{\alpha-1}e^{-\beta x}}{\lambda e^{-\lambda x}}$$

$$= \frac{\beta^\alpha}{\lambda\Gamma(\alpha)} x^{\alpha-1} e^{-x(\beta-\lambda)} .$$

First note that no matter what λ is, if $0 < \alpha < 1$, then $f(x)/g(x) \to \infty$ as $x \to 0$. So accept-reject with this proposal won't work!

However, when $\alpha \geq 1$, then $x^{\alpha-1}$ increases, so we want to choose λ such that $e^{-x(\beta-\lambda)}$ decreases (since exponential decay is more powerful than polynomial increase) (of course, you should show this more mathematically). Thus we want $\beta > \lambda$!

Thus, we restriction attention to $\alpha \geq 1$, $\lambda < \beta$,

$$c(\lambda) = \sup_x \frac{f(x)}{g(x)} = \sup_{x>0} \frac{\beta^\alpha}{\lambda\Gamma(\alpha)} x^{\alpha-1} e^{-x(\beta-\lambda)}$$

which you can show, occurs at

$$x = \frac{\alpha - 1}{\beta - \lambda} ,$$

for which

$$c(\lambda) = \frac{\beta^\alpha}{\lambda\Gamma(\alpha)} \left(\frac{\alpha-1}{\beta-\lambda}\right)^{\alpha-1} e^{1-\alpha} ,$$

which is minimized for

$$\lambda = \beta/\alpha.$$

Thus, the optimal exponential proposal for the Gamma$(\alpha, \beta), \alpha > 1$ is Exp(β/α). ■

Questions to think about

- How would you implement accept-reject for Gamma(α, β) for $0 < \alpha < 1$?

4.3 The Box-Muller transformation for $N(0, 1)$.

A classical method to generate samples from $N(0, 1)$ is the Box-Muller transformation method. Here, we will draw random variables (R^2, Θ) from a certain distribution in the polar coordinate system, and then use a transformation h, so that $h(R^2, \Theta) \sim N(0, 1)$. To find the h, we will need some theory for this.

Let X and $Y \overset{\text{iid}}{\sim} N(0, 1)$. The joint density of (X, Y) is

$$f(x, y) = \frac{1}{2\pi} e^{-x^2/2} e^{-y^2/2} \quad x \in \mathbb{R}, y \in \mathbb{R}.$$

Let (R^2, Θ) denote the polar coordinates of (X, Y) so that, $X = R\cos\Theta$ and $Y =$

$R\sin\Theta$; here the support of R is $(0,\infty)$ and the support of Θ is $(0,2\pi)$. Then,

$$R^2 = X^2 + Y^2 \qquad \tan\Theta = \frac{Y}{X}\,.$$

Notationally, we denote a realization from (R^2,Θ) as (d,θ) and find the joint density of $f(d,\theta)$. Thus, let $d = x^2+y^2$ and $\theta = \tan^{-1}(y/x)$. We know that the density for (d,θ) can be found by

$$f(d,\theta) = |J| f(x,y) \qquad \text{where } J = \begin{vmatrix} \frac{\partial x}{\partial d} & \frac{\partial y}{\partial d} \\ \frac{\partial x}{\partial \theta} & \frac{\partial y}{\partial \theta} \end{vmatrix}$$

Solving for J,

$$J = \begin{vmatrix} \frac{\partial\sqrt{d}\cos\theta}{\partial d} & \frac{\partial\sqrt{d}\sin\theta}{\partial d} \\ \frac{\partial\sqrt{d}\cos\theta}{\partial\theta} & \frac{\partial\sqrt{d}\sin\theta}{\partial\theta} \end{vmatrix} = \begin{vmatrix} \frac{1}{2}\frac{\cos\theta}{\sqrt{d}} & \frac{1}{2}\frac{\sin\theta}{\sqrt{d}} \\ -\sqrt{d}\sin\theta & \sqrt{d}\cos\theta \end{vmatrix} = \frac{1}{2}\,.$$

Since $d = x^2+y^2$, the joint density of (R^2,Θ) is $f(d,\theta)$ with

$$\begin{aligned} f(d,\theta) &= \frac{1}{2}\frac{1}{2\pi}e^{-d/2} \qquad\qquad 0<d<\infty, 0<\theta<2\pi \\ &= \underbrace{\frac{1}{2\pi}I(0<\theta<2\pi)}_{U(0,2\pi)}\ \underbrace{\frac{1}{2}e^{-d/2}I(0<d<\infty)}_{\text{Exp}(2)} \end{aligned}$$

This is a separable density, so R^2 and Θ are independent, and $\Theta \sim U[0,2\pi]$ and $R^2 \sim \text{Exp}(2)$.

To generate from Exp(2), we can use an inverse transform method. If $U \sim U(0,1)$, then by the inverse transform method, $-2\log U \sim \text{Exp}(2)$ (verify for yourself). To generate from $U(0,2\pi)$, we know if $U \sim U(0,1)$, then $2\pi U \sim U(0,2\pi)$. The Box-Muller algorithm then is given in Algorithm 9 which produces X and Y from $N(0,1)$ indendently.

Algorithm 9 Box-Muller algorithm for $N(0,1)$

```
1: Generate $U_1$ and $U_2$ from $U(0,1)$ independently
2: Set $R^2 = -2\log U_1$ and $\Theta = 2\pi U_2$
3: Set $X = R\cos(\Theta) = \sqrt{-2\log U_1}\cos(2\pi U_2)$
4: and $Y = R\sin(\Theta) = \sqrt{-2\log U_1}\sin(2\pi U_2)$.
```

4.4 Ratio-of-Uniforms

Ratio-of-uniforms is a powerful, however not so popular method to generate samples for a continuous random variables. When it works, it can work really well. The method is based critically on the following theorem.

Theorem 4. Let $f(x)$ be a target density with support $\mathcal{X}$ and distribution function F. Define the set

$$D = \left\{ (u,v) : 0 \le u \le \sqrt{f\left(\frac{v}{u}\right)} \right\} .$$

If D bounded, let (U,V) be uniformly distributed over the set D; then $V/U \sim F$.

Proof. We will show that the density of $Z = V/U$ is $f(z)$. Note that by definition, the joint density of (U,V) is

$$g_{(U,V)}(u,v) = \frac{1}{\int\int_D du\,dv} I\left\{(u,v) \in D\right\} .$$

Consider transformation $(U,V) \mapsto (U,Z)$ with $Z = V/U$. Then $U = U$ and $V = UZ$. It's easy to see that the Jacobian for this transformation is U. So

$$g_{(U,Z)}(u,z) = \frac{u}{\int\int_D du\,dv} I\left\{0 \le u \le f^{1/2}(z)\right\} .$$

Now that we have the joint distribution of (U,Z), all we need to show is that the marginal distribution of Z is F. Finding the marginal density of $Z = V/U$, we integrate out U,

$$\begin{aligned} g_Z(z) &= \int \frac{u}{\int\int_D du\,dv} I\left\{0 \le u \le f^{1/2}(z)\right\} du \\ &= \frac{1}{\int\int_D du\,dv} \int_0^{f^{1/2}(z)} udu \\ &= \frac{f(z)}{2\int\int_D du\,dv} . \end{aligned}$$

Since $g_Z(z)$ and $f(z)$ are both densities, this implies that

$$1 = \int g_Z(z)dz = \frac{\int f(z)dz}{2\int\int_D du\,dv} = \frac{1}{2\int\int_D du\,dv} \Rightarrow \int\int_D du\,dv = \frac{1}{2}$$

This implies $f_Z(z) = f(z)$. Thus, $Z = V/U$ has the desired distribution. □

So if we can draw $(U, V) \sim \text{Unif}(D)$, then $V/U \sim F$. But D looks quite complicated, so how do we uniformly draw from D?

Think back to the AR technique used to draw uniformly from a circle! If D is a bounded set, then if we enclose D in a rectangle, we can use accept-reject to draw uniform draws from D! So, the task is to find $[0, a] \times [b, c]$ such that

$$0 \leq u \leq a \quad b \leq v \leq c \qquad \text{for all } (u, v) \in D\,.$$

We just need to find any such a, b, c. First, note that if $\sup_x f^{1/2}(x)$ exists, then

$$0 \leq u \leq f^{1/2}\left(\frac{v}{u}\right) \leq \sup_{x \in \mathcal{X}} f^{1/2}(x) =: a\,.$$

Note now that inside D, if $x = v/u \Rightarrow v/x = u \leq f^{1/2}(x)$. This implies that

$$\frac{v}{x} \leq f^{1/2}(x)\,.$$

Now for:

$$x \geq 0: \quad v \leq x f^{1/2}(x) \leq \sup_{x \in \mathcal{X}} x f^{1/2}(x) =: c$$
$$x \leq 0: \quad v \geq x f^{1/2}(x) \geq \inf_{x \in \mathcal{X}} x f^{1/2}(x) =: b\,.$$

Note that if $\sqrt{f(x)}$ or $x^2 f(x)$ are unbounded, then D is unbounded, and the method cannot work. Now that we have found the rectangle: $[0, a] \times [b, c]$, we can propose from the rectangle, check if the proposed value is in the region D; if it is, we accept it and return V/U. This leads to the following algorithm:

Algorithm 10 Ratio-of-Uniforms

```
1: Generate $(U, V) \sim U[0, a] \times U[b, c]$
2: If $U \leq \sqrt{f(V/U)}$, then set $X = V/U$.
3: Else go to 1.
```

Steps 1 and 2 in Algorithm 10 are implementing an Accept-Reject to sample uniformly from D. To understand how effective this algorithm will be, we can calculate the probability of acceptance for the AR. First, note that

$$\sup_{(u,v)\in D} \frac{f(u,v)}{g(u,v)} = \sup_{(u,v)\in D} \frac{\frac{I((u,v)\in D)}{\int_C dudv}}{\frac{1}{a*(c-b)}} = 2a(c-b)$$

Thus,

$$\Pr\left(\text{Accepting for AR in RoU}\right) = \frac{1}{2a(c-b)}\,.$$

So if a is large and/or $(c-b)$ is large, the probability is small, and thus the algorithm will take a large number of loops to yield one acceptance.

Example 15 (Exponential(1))**.**

$$f(x) = e^{-x} \quad x \geq 0$$

Here,

$$D = \left\{(u,v) : 0 \leq u \leq e^{-v/2u}\right\}.$$

Since $e^{-x/2}$ is a decreasing function, $a = \sup_x e^{-x/2} = 1$. Additionally,

$$b = \inf_{x\leq 0} xe^{-x/2} = 0 \qquad \text{(since support is } x \geq 0\text{)}$$

and

$$c = \sup_{x\geq 0} xe^{-x/2} \Rightarrow c = 2e^{-1} \qquad \text{(show for yourself)}\ .$$

So we sample from $U[0,1] \times [0, 2/e]$ and then implement accept-reject. ■

Example 16 (Normal(θ, σ^2))**.** The target density is:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\theta)^2/2\sigma^2}\,.$$

The set D is

$$D = \left\{(u,v) : 0 \leq u \leq \left(\frac{1}{2\pi\sigma^2}\right)^{1/4} e^{-(v-u\theta)^2/4\sigma^2u^2}\right\}$$

In order to draw the region later, we need to rearrange the bound above, which gives us (by taking log):

$$(v-\theta u)^2 \leq 4\sigma^2 u^2 \left(\log u + \frac{1}{2}\log(2\pi\sigma^2)\right)\,.$$

The above defines the region D. Now, in order to bound the region D, we find the limits a, b, c:

$$a = \sup_{x \in \mathbb{R}} (2\pi\sigma^2)^{-1/4} e^{-(x-\theta)^2/4\sigma^2} = (2\pi\sigma^2)^{-1/4}$$

$$b = \inf_{x \leq 0} \left(\frac{1}{2\pi\sigma^2}\right)^{1/4} x e^{-(x-\theta)^2/4\sigma^2} \qquad \text{and} \qquad c = \sup_{x \geq 0} \left(\frac{1}{2\pi\sigma^2}\right)^{1/4} x e^{-(x-\theta)^2/4\sigma^2}$$

First, we find b and then c will follow similarly. Note that b will be non-positive, and thus, to find the infimum, we first take negative and then log. That is, let for $x < 0$, let

$$A(x) = \left(\frac{1}{2\pi\sigma^2}\right)^{1/4} (-x) e^{-(x-\theta)^2/4\sigma^2}$$

Then $A(x)$ is non-negative, and we want to find the supremem of $A(x)$ for $x \leq 0$. Taking log:

$$\begin{aligned} \log(A(x)) &= -\frac{1}{4}\log(2\pi\sigma^2) + \log(-x) - \frac{(x-\theta)^2}{4\sigma^2} \\ \Rightarrow \frac{d\,\log(A(x))}{dx} &= \frac{1}{x} - \frac{(x-\theta)}{2\sigma^2} \overset{set}{=} 0 \\ \Rightarrow x &= \frac{\theta \pm \sqrt{\theta^2 + 8\sigma^2}}{2} \end{aligned}$$

Now, we need to decide which of ± would be choose. Note that $\sqrt{\theta^2 + 8\sigma^2} > \theta$. Hence, since we are taking $\sup_{x \geq 0} A(x)$, we obtain

$$x_b := \frac{\theta - \sqrt{\theta^2 + 8\sigma^2}}{2}$$

Thus,

$$b = x_b f^{1/2}(x_b)$$

Similarly, we obtain

$$x_c := \frac{\theta + \sqrt{\theta^2 + 8\sigma^2}}{2}$$

with

$$c = x_c f^{1/2}(x_c)\,.$$

All that needs to be done now is to implement Algorithm 10 with these values of a, b, c, given the values of θ and σ^2

■

Questions to think about

1. Construct a similar RoU sampler for Cauchy distribution.
2. Why does RoU fail when D is unbounded?
3. For $N(0,1)$ between RoU and AR using Cauchy proposal, which is more efficient, in terms of the expected number of uniforms required for one acceptance?

4.5 The Composition Method

We have now learned many algorithm for sampling distributions. For certain special distributions, it is easier to use a *composition method* for sampling.

Suppose we have an efficient way of simulating random variables from two pmfs $\{p_j^{(1)}\}$ and $\{p_j^{(2)}\}$, and we want to simulate from

$$\Pr(X = j) = \alpha p_j^{(1)} + (1-\alpha)p_j^{(2)} \quad j \geq 0 \;\; \text{where } 0 < \alpha < 1 \,.$$

First you should note that the above *composition pmf* is a valid pmf since $\sum_j \Pr(X = j) = 1$. How would we sample in such a situation?

Let $X_1 \sim P^{(1)}$ and $X_2 \sim P^{(2)}$. Set

$$X = \begin{cases} X_1 & \text{with probability } \;\alpha \\ X_2 & \text{with probability } \;1-\alpha \end{cases} .$$

Algorithm 11 Composition method

```
1: Draw $U \sim U(0,1)$
2: **if** $U \leq \alpha$ **then** simulate $X_1 \sim P^{(1)}$ **else** simulate $X_2$ and stop
```

Proof. Consider

$$\begin{aligned} &\Pr(X = j) \\ &= \Pr(X = j, U \leq \alpha) + \Pr(X = j, \alpha < U \leq 1) \qquad \text{(by law of total probability)} \end{aligned}$$

$$
\begin{aligned}
&= \Pr(X = j \mid U \le \alpha)\Pr(U \le \alpha) + \Pr(X = j \mid \alpha < U \le 1)\Pr(\alpha < U \le 1) \\
&= \Pr(X_1 = j)\Pr(U \le \alpha) + \Pr(X_2 = j)\Pr(\alpha < U \le 1) \qquad \text{(by independence of } U \text{ and } X_1, X_2) \\
&= \alpha p_j^{(1)} + (1-\alpha)p_j^{(2)}\,.
\end{aligned}
$$

□

We can set this up more generally for k different distributions. In general, $F_i, i = 1, \dots, k$ are distribution functions, and α_i are such that $0 < \alpha_i < 1$ for all i and $\sum_i \alpha_i = 1$. The composition (or mixture) distribution is

$$F(x) = \sum_{i=1}^{k} \alpha_i F_i(x)\,.$$

If each of the F_j are continuous distributions with densities f_j, then the composition or mixture density is

$$f(x) = \sum_{i=1}^{k} \alpha_i f_i(x)\,.$$

Let $X_i \sim F_i$. To simulate from the composition F, set

$$X = \begin{cases} X_1 & \text{with probability } \alpha_1 \\ X_2 & \text{with probability } \alpha_2 \\ \vdots & \\ X_k & \text{with probability } \alpha_k \end{cases}\,.$$

Example 17 (Zero inflated Poisson distribution)**.** A Poisson(λ) distribution usually has a small mass at 0. But sometimes, we need a counting distribution with large mass at 0. For example, consider the random variable X being the number of COVID-19 patients tested positive every hour. Many hours of the day this number may be 0, and then this number can be quite high for some hours.

In such a case, we may use the *zero inflated Poisson distribution* (ZIP). Recall that if $X \sim \text{Poisson}(\lambda)$

$$\Pr(X = k) = e^{-\lambda}\frac{\lambda^k}{k!} \quad k = 0, 1, \dots\,.$$

If $X \sim \text{ZIP}(\delta, \lambda)$ for $\delta > 0$

$$\Pr(X = k) = \begin{cases} \delta + (1-\delta)e^{-\lambda} & \text{if } k = 0 \\ (1-\delta)e^{-\lambda}\dfrac{\lambda^k}{k!} & \text{if } k = \{1, 2, \dots\} \end{cases} .$$

Note that the mean of a ZIP is $(1-\delta)\lambda < \lambda$ since more mass is given at 0. We will use the composition method to sample from the ZIP distribution. To sample from a ZIP, first $p_j^{(1)}$ be defined as

$$\Pr(X_1 = 0) = 1 \quad \text{and} \quad \Pr(X_1 \neq 0) = 0 \,,$$

and let $X_2 \sim \text{Poisson}(\lambda)$. Define the pmf:

$$\Pr(X = k) = \delta p_k^{(1)} + (1-\delta)p_k^{(2)} .$$

Then $X \sim \text{ZIP}(\delta, \lambda)$. To see this, plug in $k = 0$ and $k = 1, 2, \dots$ above:

Algorithm 12 Zero inflated Poisson distribution

```
1: Draw $U \sim U(0,1)$
2: if $U \leq \delta$ then $X = 0$ else simulate $X \sim \text{Poisson}(\lambda)$
```

■

Other composition or mixture distributions are also possible. Think about Zero-inflated Binomial, Zero-inflated Geometric, 2-inflated Poisson, etc.

Example 18 (Mixture of normals)**.** Consider two normal distributions $N(\mu_1, \sigma_1^2)$ and $N(\mu_1, \sigma_2^2)$. For some $0 < p < 1$, the mixture density is

$$\begin{aligned} f(x) &= pf_1(x; \mu_1, \sigma_1^2) + (1-p)f_2(x; \mu_2, \sigma_2^2) \\ &= p\frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left\{-\frac{1}{2}\left(\frac{x-\mu_1}{\sigma_1}\right)^2\right\} + (1-p)\frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left\{-\frac{1}{2}\left(\frac{x-\mu_2}{\sigma_2}\right)^2\right\} \end{aligned}$$

Mixture distributions are particularly useful for clustering problems and we will come back to them again in the data analysis part of the course. If we want to sample from this distribution

Algorithm 13 Sampling from a Gaussian mixture

```
1: Generate $U \sim U[0,1]$
2: If $U < p$, generate $N(\mu_1, \sigma_1^2)$
3: Otherwise, generate $N(\mu_2, \sigma_2^2)$.
```

■

Example 19 (Zero-inflated gamma distribution)**.** Just like the zero-inflated Poisson distribution, there are zero-inflated normal and Gamma distributions. Let's motivate the zero-inflated Gamma distribution:

Suppose you are an auto-insurance company and you want to study the cost of claims associated with each customer. That is, each customer, if they have an accident, will come to you and claim insurance money reimbursement for the accident. So

$$\text{Let } X = \text{insurance money asked for by a customer in a month.}$$

However, most customers will not enter into any accidents, so they will claim Rs 0. But when they do, they will claim reimbursement for some amount of money that, say, will follow a Gamma distribution.

The density function can be defined as follows for $0 < p < 1$

$$f(x) = p\mathbb{I}(x = 0) + (1-p)\frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-x\beta}\,.$$

Algorithm 14 Sampling from a zero-inflated Gamma

```
1: Generate $U \sim U[0,1]$
2: If $U < p$, return $X = 0$
3: Otherwise, generate $X \sim \text{Gamma}(\alpha, \beta)$.
```

■

4.6 Miscellaneous methods in sampling

4.6.1 Known relationships

It is always useful to remember the relationships between different distributions.

1. **Binomial distribution**: We know that if $Y_1, Y_2, \dots, Y_n \overset{iid}{\sim} \text{Bern}(p)$, then

$$X = Y_1 + Y_2 + \dots Y_n \sim \text{Bin}(n, p)\,.$$

So, we can simulate n Bernoulli variables, add them up, and we have a realization from a Binomial(n, p).

2. **Negative binomial distribution**: Number of failures until the rth success. So possibly related to geometric! If $Y_1, Y_2, \dots, Y_r \overset{iid}{\sim} \text{Geom}(p)$ (on failures), then

$$X = Y_1 + Y_2 + \dots + Y_r \sim NB(r, p)\,.$$

3. **Beta distribution** If $X \sim \text{Gamma}(a, 1)$ and $Y \sim \text{Gamma}(b, 1)$, then

$$\frac{X}{X+Y} \sim \text{Beta}(a, b)\,.$$

4. **Dirichlet distribution :** The Dirichlet distribution is a distribution over pmf.

$$f(x_1, x_2, \dots, x_k) = \frac{\Gamma(\alpha_1 + \dots + \alpha_k)}{\prod_{i=1}^k \Gamma(\alpha_i)} \prod_{i=1}^k x_i^{\alpha_i - 1} \quad 0 \leq x_i \leq 1, \sum_{i=1}^k x_i = 1\,.$$

The Dirichlet distribution is a generalization of the Beta distribution. Similarly,

$$\begin{aligned} Y_1 &\sim \text{Gamma}(\alpha_1, 1) \\ Y_2 &\sim \text{Gamma}(\alpha_2, 1) \\ &\vdots \\ Y_k &\sim \text{Gamma}(\alpha_k, 1) \end{aligned}$$

Let

$$X_i = \frac{Y_i}{\sum_{i=1}^k Y_i}\,.$$

Then $(X_1, \dots, X_k) \sim \text{Dir}(\alpha_1, \alpha_2, \dots, \alpha_k)$.

5. **Chi-squared distribution**: If $Y_1, Y_2, \dots, Y_k \overset{iid}{\sim} N(0,1)$, then

$$X = Y_1^2 + Y_2^2 + \cdots + Y_k^2 \sim \chi_k^2 \,.$$

   This way we can simulate χ^2 distributions with integer degrees of freedom.

6. **t-distribution** Let $Z \sim N(0,1)$ and $Y \sim \chi_k^2$, then

$$X = \frac{Z}{\sqrt{\dfrac{Y}{k}}} \sim t_k \,.$$

7. **Location-scale family**: Let F be a distribution in the location-scale family. Then, if Z has CDF $F_Z(z)$ in the sense that it doesn't have any parameters. Then for $\mu \in \mathbb{R}$ and $\sigma > 0$,

$$Y = \mu + \sigma Z \text{ has CDF } F_Y(y) = F_Z\left(\frac{z-\mu}{\sigma}\right) \,.$$

   If Z has pdf $f(z)$ then Y has pdf $\sigma^{-1} f((z-\mu)/\sigma)$.

   So, if $Z \sim N(0,1)$, then $Y = \mu + \sigma Z \sim N(\mu, \sigma^2)$.

4.6.2 Multidimensional target

We have almost entirely focused on univariate densities, but most often interest is in multivariate/multidimensional target distribution.

- **Conditional Distribution:** Consider a variable $\mathbf{X} = (X_1, X_2, \dots, X_k)$, with a joint pdf

$$f(\mathbf{x}) = f(x_1, x_2, \dots, x_k) \,.$$

  We can use conditional distribution properties:

$$f(\mathbf{x}) = f_{X_1}(x_1) f_{X_2|X_1}(x_2) \dots f_{X_k|X_1,\dots,X_{k-1}}(x_k) \,.$$

Algorithm 15 Sampling $\mathbf{X}$ using conditional distributions

```
1: Generate $X_1 \sim f_{X_1}(x_1)$
2: Generate $X_2 \sim f_{X_2|X_1}(x_2)$
3: Generate $X_3 \sim f_{X_3|X_2,X_1}(x_3)$
4: ⋮
5: Generate $X_n \sim f_{X_k|X_{k-1},\dots,X_1}(x_k)$
6: Return $\mathbf{X} = (X_1, \dots, X_k)$
```

- **Multivariate normal:** Consider sampling from a $N_k(\mu, \Sigma)$ where Σ is positive definite. Then for $|\cdot|$ denoting determinant,

$$f_{\mathbf{X}}(\mathbf{x}) = \left(\frac{1}{2\pi}\right)^{k/2} |\Sigma|^{-1/2} \exp\left\{-\frac{(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)}{2}\right\} ,$$

is the density of a multivariate normal distribution with mean μ and covariance Σ. First, note that since Σ is a positive-definite (symmetric) matrix, we can use the eigenvalue decomposition

$$\Sigma = Q\Lambda Q^{-1}$$

where Q is the matrix of eigenvectors and since Σ is symmetric, Q is guaranteed to be an orthongal matrix so that $Q^{-1} = Q^T$ and Λ is a diagonal matrix of eigenvalues. Then, we can define the *square-root* of Σ as

$$\Sigma^{1/2} := Q\Lambda^{1/2}Q^{-1} ,$$

so that

$$\Sigma^{1/2}\Sigma^{1/2} = Q\Lambda^{1/2}Q^{-1}Q\Lambda^{1/2}Q^{-1} = Q\Lambda Q^{-1} .$$

Similarly, the inverse square-root is

$$\Sigma^{-1/2} = Q\Lambda^{-1/2}Q^{-1} ,$$

Set $\mathbf{Z} = \Sigma^{-1/2}(\mathbf{X} - \mu)$. Then

$$\mathbf{Z} \sim N_k(0, I_k) .$$

That is, $\mathbf{Z}$ is a k-dimensional multivariate normal distribution with an identity covariance matrix. Which implies if $\mathbf{Z} = (Z_1, \dots, Z_k)$, then $\text{Cov}(Z_i, Z_j) = 0$ for all $i \neq j$.

For the normal distribution, if the covariance is zero, then the random variables are independent! This isn't true in general but is true for normal random variables.

So, to sample from $N_k(\mu, \Sigma)$, we can sample $Z_1, Z_2, \dots, Z_k \overset{iid}{\sim} N(0,1)$, and set $\mathbf{Z} = (Z_1, \dots, Z_k)$. Then

$$\mathbf{X} := \mu + \Sigma^{1/2}\mathbf{Z} \sim N_k(\mu, \Sigma)\,.$$

Then $\mathbf{Z} \sim N_k(\mu, \Sigma)$.

Questions to think about

- Can you construct a zero-inflated normal distribution and find a suitable application of it?

4.7 Exercises

1. Using the inverse transform method, simulate from $\text{Exp}(\lambda)$ for any $\lambda > 0$. Implement this for $\lambda = 5$.

2. Use the inverse transform method to obtain samples from the Weibull(α, λ)

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

3. (Ross 5.1) Give a method for generating a random variable having density function

$$f(x) = \frac{e^x}{e-1} \qquad 0 \leq x \leq 1\,.$$

4. (Ross 5.2) Give a method for generating a random variable having density function

$$f(x) = \begin{cases} \dfrac{x-2}{2} & \text{if } 2 \leq x \leq 3 \\ \dfrac{2-x/3}{2} & \text{if } 3 \leq x \leq 6 \end{cases}$$

5. (Ross 5.3) Use the inverse transform method to generate a random variable having

distribution function

$$F(x) = \frac{x^2 + x}{2} \qquad 0 \leq x \leq 1\,.$$

6. Sample following the following distribution using two different methods:

$$f(x) = \begin{cases} \frac{3}{4}(1 - x^2) & \text{if } x \in (-1, 1) \\ 0 & \text{otherwise} \end{cases}$$

7. (Ross 5.6) Let X be an Exp(1). Provide an efficient algorithm for simulating a random variable whose distribution is the conditional distribution of X given that $X < 0.05$. That is, its density function is

$$f(x) = \frac{e^{-x}}{1 - e^{-0.05}} \qquad 0 < x < 0.05\,.$$

Using `R` generate 1000 such random variables and use them to estimate $\mathrm{E}[X \mid X < 0.05]$.

8. (Ross 5.7) Suppose it is relatively easy to generate random variables from any of the distributions F_i, $i = 1, \ldots, k$. How could we generate a random variable from the distribution function

$$F(x) = \sum_{i=1}^{n} p_i F_i(x)\,,$$

where $p_i \geq 0$ and $\sum p_i = 1$.

9. (Ross 5.8) Using the previous exercise, provide algorithms for generating random variables from the following distributions:

   (a) $F(x) = \frac{x + x^3 + x^5}{3}, 0 \leq x \leq 1$.

   (b) $F(x) = \begin{cases} \frac{1 - e^{-2x} + 2x}{3} & \text{if } x \in (0, 1) \\ \frac{3 - e^{-2x}}{3} & \text{if } x \in [1, \infty) \end{cases}$

10. (Ross 5.9) Give a method to generate a random variable with distribution function

$$F(x) = \int_0^\infty x^y e^{-y} dy \qquad 0 \leq x \leq 1$$

11. (Ross 5.15) Give two methods for generating a random variable with density function

$$f(x) = xe^{-x}, 0 \leq x < \infty\,.$$

12. (Ross 5.18) Give an algorithm for generating a random variable having density function

$$f(x) = 2xe^{-x^2}\,, \qquad x > 0\,.$$

13. (Ross 5.19) Show how to generate a random variable who distribution function is

$$F(x) = \frac{x + x^2}{2}\,, \qquad 0 \leq x \leq 1$$

using the inverse transform, accept-reject, composition method.

14. (Ross 5.20) Use the AR method to find an efficient way to generate a random variable having density function

$$f(x) = \frac{(1+x)e^{-x}}{2} \qquad 0 < x < \infty\,.$$

15. (Ross 5.21) Consider the target density to be a truncated Gamma$(\alpha, 1)$, $\alpha < 1$ defined on (a, ∞) for some $a > 0$. Suppose the proposal distribution is a truncated exponential(λ), defined on the same (a, ∞). What is the best λ to use?

16. (Using R)

    (a) Implement an accept-reject sampler to sample uniformly from the circle $\{x^2 + y^2 \leq 1\}$ and obtain 10000 samples and estimate the probability of acceptance. Does it approximately equal $\pi/4$?

    (b) Now consider sampling uniformly from a p-dimensional sphere (a circle is $p = 2$). Consider a p-vector $\mathbf{x} = (x_1, x_2, \dots, x_p)$ and let $\|\cdot\|$ denote the Euclidean norm. The pdf of this distribution is

    $$f(\mathbf{x}) = \frac{\Gamma\left(\frac{p}{2}+1\right)}{\pi^{p/2}} I\{\|\mathbf{x}\| \leq 1\}\,.$$

    Use a uniform p-dimensional hypercube to sample uniformly from this sphere. Implement this for $p = 3, 4, 5,$ and 6. What happens as p increases?

17. (Using R)

(a) Using accept-reject and a standard normal proposal, obtain samples from a truncated standard normal distribution with pdf:

$$f(x) = \frac{1}{\Phi(a) - \Phi(-a)} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} I(-a < x < a)\,,$$

where $\Phi(\cdot)$ is the CDF of a standard normal distribution. Run for $a = 4$ and $a = 1$. What are the differences between the two settings.

(b) Now consider a multivariate truncated normal distribution, where for $\mathbf{x} = (x_1, x_2, \dots, x_p)$, the pdf is

$$f(\mathbf{x}) = \left(\frac{1}{\Phi(a) - \Phi(-a)}\right)^p \left(\frac{1}{\sqrt{2\pi}}\right)^p e^{-\mathbf{x}^T\mathbf{x}/2} I(-a < \mathbf{x} < a)\,.$$

Implement an accept-reject sampler with proposal distribution $N_p(0, I)$ with $a = 4$ and $p = 3, 10$ and with $a = 1$ and $p = 3, 10$. Describe the differences between these settings.

18. Implement an accept-reject sampler to draw from a Gamma$(\alpha, 1)$ for $\alpha > 1$. Using the above method, can you draw samples from Gamma(α, β), for any β?

19. In accept-reject sampling, why is $c \geq 1$?

20. Use ratio-of-uniforms method to sample from a truncated exponential distribution with density

$$f(x) = \frac{e^{-x}}{1 - e^{-a}} \quad 0 < x < a\,.$$

How efficient is this algorithm?

21. Use ratio-of-uniforms method to sample from the distribution with density

$$f(x) = \frac{1}{x^2} \quad x \geq 1\,.$$

22. Use ratio-of-uniforms method to draw samples from a t_ν distribution for $\nu \geq 1$.

23. (Zero-inflated Gamma distribution) Suppose you are an auto-insurance company and you want to study the cost of claims associated with each customer. That is, each customer, if they have an accident, will come to you and claim insurance money reimbursement for the accident. So

Let X = insurance money asked for by a customer in a month.

However, most customers will not enter into any accidents, so they will claim Rs 0. But when they do, they will claim reimbursement for some amount of money that, say, will follow a Gamma distribution.

The density function can be defined as follows for $0 < p < 1$

$$f(x) = p\mathbb{I}(x = 0) + (1 - p)\frac{\beta^\alpha}{\Gamma(\alpha)}x^{\alpha-1}e^{-x\beta}\,.$$

Provide an algorithm to sample this random variable.

5 Importance Sampling

We have so far learned many (many!) ways of sampling from different distributions. These sampling methodologies are particularly useful when we want to estimate characteristic of F. Using computer simulated samples from F to estimate characteristics of F is broadly termed as *Monte Carlo*

5.1 Simple Monte Carlo

Suppose F is a distribution with density f. We are interested in estimating the expectation of a function $h : \mathcal{X} \to \mathbb{R}$ with respect to F. That is, we want to estimate

$$\theta := \mathrm{E}_F[h(X)] = \int_{\mathcal{X}} h(x) f(x)\, dx < \infty\,,$$

we assume that θ is finite. We also assumed that

$$\sigma^2 = \mathrm{Var}_F(h(X)) < \infty\,.$$

Note: there is no "data" here, there is just an integral! We are just interested in estimating an annoying integral.

Note: notation $E_F[X]$ means the expectation is with respect to F. From now on, it is very important to keep track of what the expectation is with respect to.

Suppose we can draw iid samples $X_1, \dots, X_N \overset{\text{iid}}{\sim} F$ (this we can do using the many methods we have learned). Then, by the weak law of large numbers, as $N \to \infty$,

$$\hat{\theta} = \frac{1}{N} \sum_{t=1}^{N} h(X_t) \xrightarrow{p} \theta\,.$$

In addition, we can find the variance of the estimator:

$$\begin{aligned}
\mathrm{Var}(\hat{\theta}) &= \mathrm{Var}\left(\frac{1}{N} \sum_{t=1}^{N} h(X_t) \right) \\
&= \frac{1}{N^2} \sum_{t=1}^{N} \mathrm{Var}_F(h(X_t)) \qquad \text{because of independence} \\
&= \frac{\mathrm{Var}_F(h(X_1))}{N} \qquad \text{because of identical}
\end{aligned}$$

$$= \frac{\sigma^2}{N} \,.$$

Naturally, a central limit theorem also holds if $\sigma^2 < \infty$, so that as $N \to \infty$

$$\sqrt{N}(\hat{\theta} - \theta) \overset{d}{\to} N(0, \sigma^2) \,,$$

This central limit theorem gives us an expected behavior of $\hat{\theta}$ for large values of n.

Q. But is there a way we can obtain a better estimator of θ?

A. Possibly by using importance sampling.

5.2 Simple importance sampling

Our goal is the same. For $h : \mathcal{X} \to \mathbb{R}$, we want to estimate $\theta = \mathrm{E}_F[h(X)]$. Similar to the the accept-reject sampler, we will choose a proposal distribution. Let G be a distribution with density g defined on $\mathcal{X}$ so that,

$$\begin{aligned}
\mathrm{E}_F[h(X)] &= \int_{\mathcal{X}} h(x) f(x) dx \\
&= \int_{\mathcal{X}} \frac{h(x)f(x)}{g(x)} g(x)\, dx \\
&= \mathrm{E}_G\left[\frac{h(Z)f(Z)}{g(Z)}\right], \qquad Z \sim G
\end{aligned}$$

If $Z_1, \dots, Z_N \overset{\text{iid}}{\sim} G$, then an estimator of θ is

$$\hat{\theta}_g = \frac{1}{N} \sum_{t=1}^{N} \frac{h(Z_t)f(Z_t)}{g(Z_t)} \,.$$

The estimator $\hat{\theta}_g$ is the *importance sampling estimator*, the method is called *importance sampling* and G is the *importance distribution.*

Let

$$w(Z_t) = \frac{f(Z_t)}{g(Z_t)}$$

be the weights assigned to each point Z_t. Then $\hat{\theta}_g$ is a weighted average of of $h(Z_t)$. Intuitively, this means that depending on how likely a sampled value is for f and g, a weight is assigned to that value.

Example 20 (Moments of Gamma distribution)**.** Suppose we want to estimate the kth moment of a Gamma distribution. That is, let F be the density of a Gamma(α, β) distribution. Then

$$\theta = \int_0^\infty x^k \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} dx\,.$$

Suppose we set G to be also an Exponential(λ) distribution. Let $Z_1, \dots, Z_N \sim \text{Exp}(\lambda)$

$$\begin{aligned}\hat{\theta}_g &= \frac{1}{N}\sum_{t=1}^{N}\left[\frac{h(Z_t)f(Z_t)}{g(Z_t)}\right] \\ &= \frac{1}{N}\sum_{t=1}^{N}\left[\frac{\beta^\alpha}{\Gamma(\alpha)}\frac{Z_t^k Z_t^{\alpha-1}e^{-\beta Z_t}}{\lambda e^{-\lambda Z_t}}\right]\,.\end{aligned}$$

■

So we have now constructed an alternative estimator of θ. In fact, a different choice of G will yield a different estimator. It is now important to study the properties of this importance sampling estimator. We study a sequence of properties.

Theorem 5 (Unbiasedness)**.** The importance sampling estimator $\hat{\theta}_g$ is unbiased for θ.

Proof. To show an estimator is unbiased, we need to show that $\text{E}[\hat{\theta}_g] = \theta$. Consider

$$\begin{aligned}\text{E}\left[\hat{\theta}_g\right] &= \text{E}_G\left[\frac{1}{N}\sum_{t=1}^{N}\frac{h(Z_t)f(Z_t)}{g(Z_t)}\right] \\ &= \frac{1}{N}\sum_{t=1}^{N}\text{E}_G\left[\frac{h(Z_t)f(Z_t)}{g(Z_t)}\right] \\ &= \frac{1}{N}\sum_{t=1}^{N}\text{E}_G\left[\frac{h(Z_1)f(Z_1)}{g(Z_1)}\right] \\ &= \int_{\mathcal{X}}\frac{h(z)f(z)}{g(z)}g(z)\,dz \\ &= \int_{\mathcal{X}} h(z)f(z)dz \\ &= \theta\,.\end{aligned}$$

□

Theorem 6. *The importance sampling estimator is consistent for θ. That is, as $N \to \infty$,*

$$\hat{\theta}_g \xrightarrow{p} \theta\,.$$

Proof. Note that $\hat{\theta}_g$ is just a sample average:

$$\hat{\theta}_g = \frac{1}{N}\sum_{t=1}^{N} \frac{h(Z_t)f(Z_t)}{g(Z_t)}\,.$$

The law of large numbers applies to any sample average whose expectation is finite. So by the law of large numbers, as $N \to \infty$,

$$\hat{\theta}_g \xrightarrow{p} \mathrm{E}[\hat{\theta}_g] = \theta\,.$$

□

This means that as we get more and more samples from G, our estimator will get increasingly closer to the truth.

However, we should never be happy with a point estimator!

It is essential to quantify the variability in our estimator $\hat{\theta}_g$ in order to ascertain how "erratic" or "stable" the estimator is. We also want to establish expected behavior for $\hat{\theta}_g$, but *does a central limit theorem hold?*. Notice that a simple Monte Carlo is just a sample average, so we should be able to directly apply the CLT result, if the variance is finite. Note that, the variance of $\hat{\theta}_g$ is

$$\mathrm{Var}(\hat{\theta}_g) = \mathrm{Var}_g\left(\frac{1}{N}\sum_{t=1}^{N}\frac{h(Z_t)f(Z_t)}{g(Z_t)}\right) = \frac{1}{N}\mathrm{Var}_g\left(\frac{h(Z_1)f(Z_1)}{g(Z_1)}\right) =: \frac{\sigma_g^2}{N}\,.$$

A central limit theorem will hold if $\sigma_g^2 = \mathrm{Var}_g\left(\frac{h(Z_1)f(Z_1)}{g(Z_1)}\right) < \infty$.

So the question is, when is this finite?

The following theorem provides a sufficient condition.

Theorem 7. Suppose $\sigma^2 = \text{Var}_F(h(X)) < \infty$. If g is chosen such that

$$\sup_{z \in \mathcal{X}} \frac{f(z)}{g(z)} \leq M < \infty$$

then

$$\sigma_g^2 < \infty \,.$$

Proof. First note that if the variance of a random variable is finite, this is equivalent to saying that the second moment of that variable is finite. So, consider the second moment of $\frac{h(Z)f(Z)}{g(Z)}$ where $Z \sim G$.

$$\begin{aligned}
\text{E}_G\left[\left(\frac{h(Z)f(Z)}{g(Z)}\right)^2\right] &= \int_{\mathcal{X}} \frac{h(z)^2 f(z)^2}{g(z)^2} g(z)dz \\
&= \int_{\mathcal{X}} h(z)^2 \, \frac{f(z)}{g(z)} \, f(z)dz \\
&\leq M \int_{\mathcal{X}} h(z)^2 f(z)dz \\
&= M \, \text{E}_F(h(X)^2) < \infty \quad \text{by assumption} \,.
\end{aligned}$$

□

Thus, if an accept-reject is possible for the propogal G, then a simple importance sampling estimator of θ, with a finite variance, is also possible. Now, we have a central limit theorem that can hold. Recall

$$\sigma_g^2 = \text{Var}_G\left(\frac{h(Z)f(Z)}{g(Z)}\right) \,. \tag{1}$$

By the CLT, if $\sigma_g^2 < \infty$, then as $N \to \infty$,

$$\sqrt{N}(\hat{\theta}_g - \theta) \xrightarrow{d} N(0, \sigma_g^2) \,. \tag{2}$$

Further, an estimator of σ_g^2 is easily available since we have N samples of $h(Z)f(Z)/g(Z)$ available. Thus, an estimator of σ_g^2 is the sample variance from all the samples:

$$\hat{\sigma}_g^2 := \frac{1}{N-1}\left(\frac{h(Z_t)f(Z_t)}{g(Z_t)} - \hat{\theta}_g\right)^2 \,.$$

Example 21 (Gamma continued)**.** Recall from the accept-reject example for Gamma(α, β) with $\alpha \geq 1$ and Exponential(λ) proposal for an accept-reject sampler will work only if $\lambda < \beta$. That means, when $\lambda < \beta$, there exists a finite M, and the importance sampling estimator will have a finite variance. ■

Questions to think about

1. Can we construct G so that its support, $\mathcal{Y}$ is larger than $\mathcal{X}$?
2. Check what happens with $\beta = \lambda$ in this simulation.
3. Why would a CLT be useful here?
4. How would we check whether this importance sampler is better than IID Monte Carlo?

5.2.1 Optimal proposals

How do we choose the importance distribution g? The proposal g should be chosen so that:

- Sampling from G is relatively easy
- $\text{Var}_g(\hat{\theta}_g) = \sigma_g^2/N$ is smaller than regular Monte Carlo variance estimator.

Note that, one reason to use importance sampling would be to obtain smaller variance estimators than the original. So, if we can choose g such that σ_g^2 is minimized that would be ideal!

Let's see this term:

$$\sigma_g^2 = \text{Var}_G\left(\frac{h(Z)f(Z)}{g(Z)}\right) = \text{E}_G\left[\frac{h(Z)^2 f(Z)^2}{g(Z)^2}\right] - \theta^2 = \underbrace{\int_{\mathcal{X}} \frac{h(z)^2 f(z)^2}{g(z)} dz}_{A} - \theta^2$$

For the above to be small, term A should be close to θ^2. This logic leads to the following theorem.

Theorem 8. If $\int_{\mathcal{X}} |h(x)| f(x) dx \neq 0$, the importance density g^* that minimizes σ_g^2 is

$$g^*(z) = \frac{|h(z)| f(z)}{\text{E}_F\left[|h(x)|\right]} .$$

Proof. Consider the above importance density. The second moment of the importance sampling estimator with this density is:

$$
\begin{aligned}
&\theta^2 + \sigma^2_{g^*} \\
&= \mathrm{E}_{G^*}\left[\left(\frac{h(Z)f(Z)}{g^*(Z)}\right)^2\right] \\
&= \int_{\mathcal{X}} \frac{h(z)^2 f(z)^2}{g^*(z)^2} g^*(z) dz \\
&= \int_{\mathcal{X}} \frac{h(z)^2 f(z)^2}{|h(z)| f(z)} \cdot \mathrm{E}_F\left[|h(x)|\right] dz \\
&= \mathrm{E}_F\left[|h(x)|\right] \int_{\mathcal{X}} |h(z)| f(z) dz \\
&= \left[\int_{\mathcal{X}} |h(z)| f(z) dz\right]^2 \\
&= \left[\int_{\mathcal{X}} \frac{|h(z)| f(z)}{g(z)} g(z) dz\right]^2 \quad \text{for any other } g \text{ defined on } \mathcal{X} \\
&= \left(\mathrm{E}_G\left[\frac{|h(z)| f(z)}{g(z)}\right]\right)^2 \\
&\leq \mathrm{E}_G\left[\frac{h(z)^2 f(z)^2}{g^2(z)}\right] \quad \text{By Jensen's inequality: for a convex function } \phi,\ \phi(E[x]) \leq E(\phi(x)) \\
&= \theta^2 + \sigma^2_g .
\end{aligned}
$$

Thus, for any generic proposal g defined on $\mathcal{X}$, we have

$$\sigma^2_{g^*} \leq \sigma^2_g .$$

Since this is true for all g, this implies that g^* produces the smallest $\sigma^2_{g^*}$. □

Note that, with this choice of proposal,

$$
\begin{aligned}
\sigma^2_{g^*} &= \mathrm{Var}_{g^*}\left(\frac{h(Z)f(Z)}{g^*(Z)}\right) \\
&= \mathrm{E}_F\left[|h(x)|\right]^2 \mathrm{Var}_{G^*}\left(\frac{h(Z)f(Z)}{|h(Z)|f(Z)}\right) \\
&= \mathrm{E}_F\left[|h(z)|\right]^2 \mathrm{Var}_{G^*}\left(\frac{h(Z)f(Z)}{|h(Z)|f(Z)}\right) .
\end{aligned}
$$

If on the support $\mathcal{X}$, $h(Z) = |h(Z)|$, then the variance of the importance sampling estimator is zero!

Example 22 (Gamma distribution)**.** Consider estimating moments of a Gamma(α, β) distribution. We actually know the optimal importance distribution here! For estimating the kth moment

$$\begin{aligned} g^*(z) &\propto |h(z)|f(z) \\ &= |x^k| x^{\alpha-1} \exp\{-\beta x\} \\ &= x^{\alpha+k-1} \exp\{-\beta x\} \; . \end{aligned}$$

So the optimum importance distribution is Gamma$(\alpha+k, \beta)$. The variance in this case of the estimator will be 0. ■

Example 23 (Mean of standard normal)**.** Let $h(x) = x$ and let $f(x)$ be the density of a standard normal distribution. So we are interested in estimating the mean of the standard normal distribution. The universally optimal proposal in this case is

$$g^*(x) = \frac{|x|e^{-x^2/2}}{\int |x|e^{-x^2/2}dx}$$

But it may be quite challenging to draw samples from the above distribution! In order for importance sampling to be useful, we need not find the optimal proposal, as long as we can find a *more* efficient proposal than sampling from the target.

Consider an importance distribution of $N(0, \sigma^2)$ for some $\sigma^2 > 0$. The variance of the importance estimator is

$$\begin{aligned} \sigma_g^2 &= \int_{-\infty}^{\infty} \frac{h(x)^2 f(x)^2}{g(x)} dx \\ &= \int_{-\infty}^{\infty} x^2 \frac{\sigma}{\sqrt{2\pi}} \exp\left\{ \frac{x^2}{2\sigma^2} - x^2 \right\} dx \\ &= \sigma \int_{-\infty}^{\infty} x^2 \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{x^2}{2}\left(2 - \frac{1}{\sigma^2}\right) \right\} dx \\ &= \frac{\sigma}{\sqrt{2-\sigma^{-2}}} \int_{-\infty}^{\infty} x^2 \cdot \underbrace{\sqrt{\frac{2-\sigma^{-2}}{2\pi}} \exp\left\{ -\frac{x^2}{2}\left(2 - \frac{1}{\sigma^2}\right) \right\}}_{\text{density of } N(0,(2-\sigma^{-2})^{-1}) \text{ if } \sigma^2 > 1/2} dx \\ &= \frac{\sigma}{\sqrt{2-\sigma^{-2}}} \frac{1}{2-\sigma^{-2}} \\ &= \frac{\sigma}{(2-\sigma^{-2})^{3/2}} \qquad \text{if } \sigma^2 > 1/2 \end{aligned}$$

else if $\sigma^2 < 1/2$, the integral diverges and the variance is infinite. Also, minimizing the variance:

$$\arg \min_{\sigma > \sqrt{1/2}} \frac{\sigma}{(2 - \sigma^{-2})^{3/2}} = \sqrt{2}\,.$$

Thus the optimal proposal has standard deviation $\sigma = \sqrt{2}$, not 1! Also, at $\sigma^2 = 2$, the variance is .7698 which is less than 1. ■

5.2.2 Questions to think about

- Does this mean that $N(0, 2)$ is the optimal proposal for estimating the mean of a standard normal?
- What is the optimal proposal within the class of *Beta* proposals for estimating the mean of a Beta distribution?